Chapter 5

Model Predictive Control

EDWARD S. MEADOWS
Departamento de Eletrônica
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, RS BRAZIL

JAMES B. RAWLINGS
Department of Chemical Engineering
The University of Wisconsin
Madison, WI USA

5.1 Introduction

Model predictive control (MPC) is an optimal-control based method to select control inputs by minimizing an objective function. The objective function is defined in terms of both present and predicted system variables and is evaluated using an explicit model to predict future process outputs. This chapter discusses the current practice in MPC, computational and design issues, performance and stability and future research directions. Model predictive control incorporates ideas from systems theory, system identification and optimization. The application of these ideas is illustrated using simulation results from two chemical reactor models.

Because MPC is implemented with digital computers, much of this chapter deals with discrete-time formulations. In discrete time, a nonlinear, deterministic process model can be represented by

$$x_{k+1} = f(x_k, u_k) \tag{5.1}$$

$$y_k = g(x_k) \tag{5.2}$$

The states x_k, controls u_k and outputs y_k are vectors.

The MPC control problem is as follows: With knowledge of the current output y_k, we seek a control that minimizes the objective function

$$J = \phi(y_{k+N|k}) + \sum_{j=0}^{N-1} L(y_{k+j|k}, u_{k+j|k}, \Delta u_{k+j|k}) \tag{5.3}$$

Of the N-move control sequence that minimizes the above objective, only the first is implemented. When another measurement becomes available, the parameters of the problem are updated and a new optimization problem is formulated whose solution provides the next control. This repeated optimization using an objective function that is modified through process feedback is one of the principal defining features of MPC.

The objective function that appears in Equation 5.3 illustrates some of the commonly used terminology in the literature of MPC. States, outputs and controls are usually doubly indexed as $x_{k+j|k}$, $y_{k+j|k}$ and $u_{k+j|k}$ to indicate values at time $k+j$ given information up to and including time k. Since MPC requires prediction, the double subscript also carries the connotation of prediction when the first index is larger than the second. Variables that are singly indexed may be used to represent controls, states or outputs that have already been implemented, computed or measured, up to the current time k. An optimal sequence of controls is often indicated using an asterisk ($u^*_{k+j|k}$). The corresponding state and output values then become $x^*_{k+j|k}$ and $y^*_{k+j|k}$, respectively.

Another feature of MPC illustrated by Equation 5.3 is the presence of $\Delta u_{k+j|k}$ in the objective, in which $\Delta u_{k+j|k} = u_{k+j|k} - u_{k+j-1|k}$. In chemical engineering applications, the speed of control action may be limited. For example, a large valve may require several seconds to change position. Rather than model the dynamics of control actuators, it is common practice to include limitations on actuator speed through penalties or constraints on $\Delta u_{k+j|k}$. Aside from the issue of actuator limitations, inclusion of $\Delta u_{k+j|k}$ offers performance advantages in that control action can be limited through restrictions or penalties on $\Delta u_{k+j|k}$ without introducing permanent offset in the output. This issue is discussed further in Section 5.7.

The objective function stage-cost L in Equation 5.3 may be chosen to meet a variety of process objectives including maximization of profits, minimization of operating costs or energy usage, or may be a measure of deviation of the process output from a reference trajectory. In subsequent discussion, we will emphasize the case of regulation to a setpoint using a quadratic error criterion. In this case, the objective function of Equation 5.3 may be written

$$J = \|y_{k+N|k} - y_{\text{ref}}\|_Q^2 + \sum_{j=0}^{N-1} \Big(\|y_{k+j|k} - y_{\text{ref}}\|_Q^2 + \|u_{k+j|k} - u_{\text{ref}}\|_R^2 + \|\Delta u_{k+j|k}\|_S^2 \Big) \tag{5.4}$$

In Equation 5.4, Q, R and S indicate weighting matrices in vector norms. When penalties or constraints on $\Delta u_{k+j|k}$ are used, we admit the special notation $u_{k-1|k}$ to indicate the most recently implemented control value, which is *not* a decision variable.

If the process model is linear, MPC with the objective of Equation 5.4 reduces to the linear-quadratic optimal control problem. LQ optimal control theory can be viewed as a (very important) special case of MPC. Many of the key theoretical results from LQ optimal control have analogs or extensions to the nonlinear case. For example, it is possible to guarantee nominal stability of an LQ optimal control in the closed loop based on properties of solutions of the Riccati difference equation [5]. In nonlinear MPC, this concept may be extended to provide stability results for nonlinear MPC based on iterations using the theory of dynamic programming [48].

Another key feature of MPC is the ability to handle constraints directly, including constraints on the control inputs, outputs or internal states. The constraint handling property has been one of the most significant contributors to its success in industrial applications. Some typical constraint formulations are discussed in Section 5.4.

5.2 Some Examples

Before proceeding with more mathematical developments, we wish to introduce two example problems that will be used to illustrate some of the principles of nonlinear MPC. Our examples are both taken from the chemical engineering literature and represent chemical reactors commonly used in industry.

5.2.1 Isothermal CSTR

Here we consider control of a continuous stirred-tank reactor (CSTR) using dilution rate as the manipulated variable. The volume of the reactor is constant. The dynamics and control of this reactor have been studied by several researchers, including Van de Vusse [75], Kantor [31], Kravaris and Daoutidis [38], and Sistu and Bequette [69]. The reactor model is given by

$$\dot{x}_1 = -k_1 x_1 - k_3 x_1^2 + (x_F - x_1)u \tag{5.5}$$

$$\dot{x}_2 = k_1 x_1 - k_2 x_2 - x_2 u \tag{5.6}$$

which models the reactions

$$\mathrm{A} \xrightarrow{k_1} \mathrm{B} \qquad \mathrm{B} \xrightarrow{k_2} \mathrm{C} \qquad 2\mathrm{A} \xrightarrow{k_3} \mathrm{D}$$

The state variables x_1 and x_2 represent the concentrations of species A and B, respectively, x_F represents the concentration of A in the feed and u represents the dilution (feed) rate. The parameters $\{k_1, k_2, k_3, x_F\}$ have nominal values $\left\{50\ \mathrm{hr}^{-1}, 100\ \mathrm{hr}^{-1}, 10\ \mathrm{liter/(mol\ hr)}, 10\ \mathrm{mol/liter}\right\}$. The primary control objective is to maintain the output variable $y = x_2$ at a setpoint of 1.0.

Assuming a perfect model, we can solve Equations 5.5 and 5.6 at steady state to find values of $\{x_1, x_2, u\}$ that give the desired output. Because of the quadratic term in Equation 5.5, there are two solutions: $\{2.5, 1.0, 25\}$ and $\{6.6667, 1.0, 233.33\}$. According to Sistu and Bequette [69], lower feed rates are preferable to achieve higher conversion; therefore as a secondary control objective, we seek to maintain a low dilution rate.

Assuming that the steady-state model solution provides setpoints for $\{x_1, x_2, u\}$, we can form an objective function

$$J = \int_t^{t+T} \gamma_1 \left[x_1(\tau) - 2.5\right]^2 + \gamma_2 \left[x_2(\tau) - 1.0\right]^2 + \gamma_3 \left[u(\tau) - 25\right]^2 d\tau \tag{5.7}$$

in which $\{\gamma_1, \gamma_2, \gamma_3\}$ represent weight factors that may be used for scaling or weighting the relative importance of each contribution.

Equations 5.5, 5.6 and the objective function of Equation 5.7 are defined for continuous time systems. In order to solve with a computer, the differential equations must be discretized and the controller must be expressed in a finite parameterization. We discretized the dynamic equations and the objective function using orthogonal collocation on finite elements. Following the example of Sistu and Bequette [69], each finite element was 0.002 hours (7.2 seconds) long. To solve the model equations, we used three interior collocation points and a point on the element boundaries. (More details on the orthogonal collocation technique are provided in Section 5.4.) In the nonlinear program that solved the MPC problem, we implemented constraints on control and state variables to ensure that they remained non-negative.

Using $\gamma_1 = \gamma_2 = 1000$ and $\gamma_3 = 1$, MPC results for the nominal system are shown in Figure 5.1 for horizon length 5 (five finite elements of 7.2 seconds each). The initial condition was $\{2.9997, 1.11697\}$, which was the same initial condition used by Sistu and Bequette [69]. The points in the diagrams show the sampling times. Between sampling times the curves were computed by integrating the nominal model. Both states and the control converge quickly to the desired target values. Similar results were obtained for simulations with horizon lengths from 1 to 7.

In most processes, all of the internal model states are not measured. If we use only the output variable x_2 in the objective ($\gamma_1 = \gamma_3 = 0$), we find that the MPC controller is able to bring the output to the setpoint, but the controller drives the system to the steady-state values at the higher (undesirable) dilution rate. These results are shown in Figure 5.2. Similar results were also obtained for horizon lengths from 1 to 7.

Since using only the output variable failed to satisfy the secondary control objective of maintaining low dilution rate, a logical attempt to satisfy the secondary objective might be to place an upper limit on the control input. Although not provided in the problem description supplied by Sistu and Bequette [69], we can reasonably assume that the nominal steady-state feed rate (25) lies near the middle of the operating range and we therefore apply a maximum feed rate of 50. The results are provided in Figure 5.3, which shows that the steady-state value of the output x_2 has offset. The same effect is seen with controllers with horizon lengths 1 through 7. The length of the horizon has an effect on the steady-state values observed in the simulations. The steady-state value of the output versus horizon length is shown in Figure 5.4. We can see that longer horizons produce less offset. With "well-behaved" systems, steady-state offset will disappear in the limit as the horizon increases to infinity.

Since none of the closed-loop control values in Figure 5.3 are at the upper limit, it is not obvious why the input constraint causes steady-state offset. To see why this occurs, we must examine the open-loop control profile computed at each time step. For each of the horizon lengths tested in simulation, we found that the input constraint is active for *some* portion of the prediction horizon. For $N = 1$ to 4, the input constraint is active for the initial open-loop control; therefore, the steady-state value of x_2 corresponds to $u = 50$. For $N = 5$, the initial open-loop control is not constrained, but

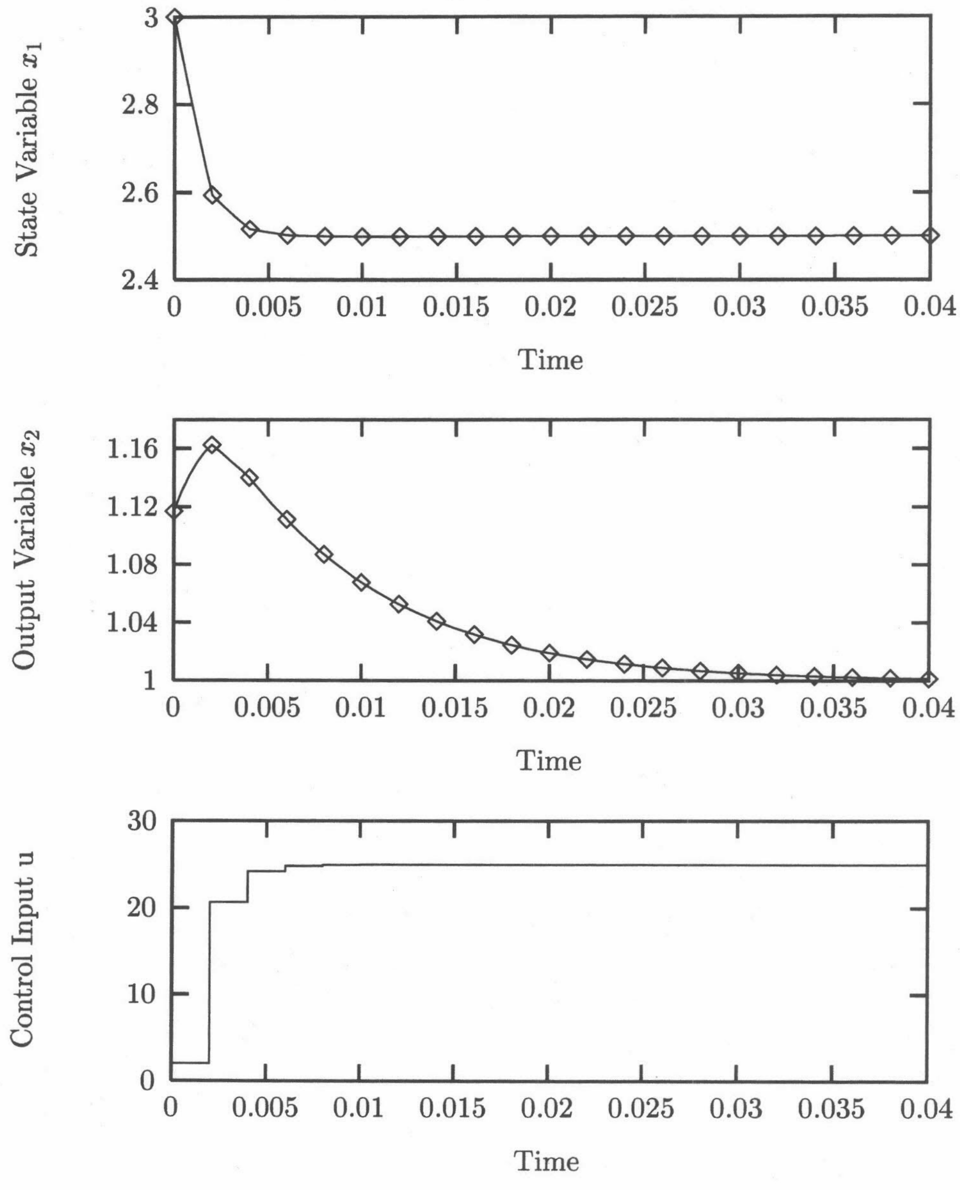

Figure 5.1: Isothermal CSTR: Nominal system with horizon length 5.

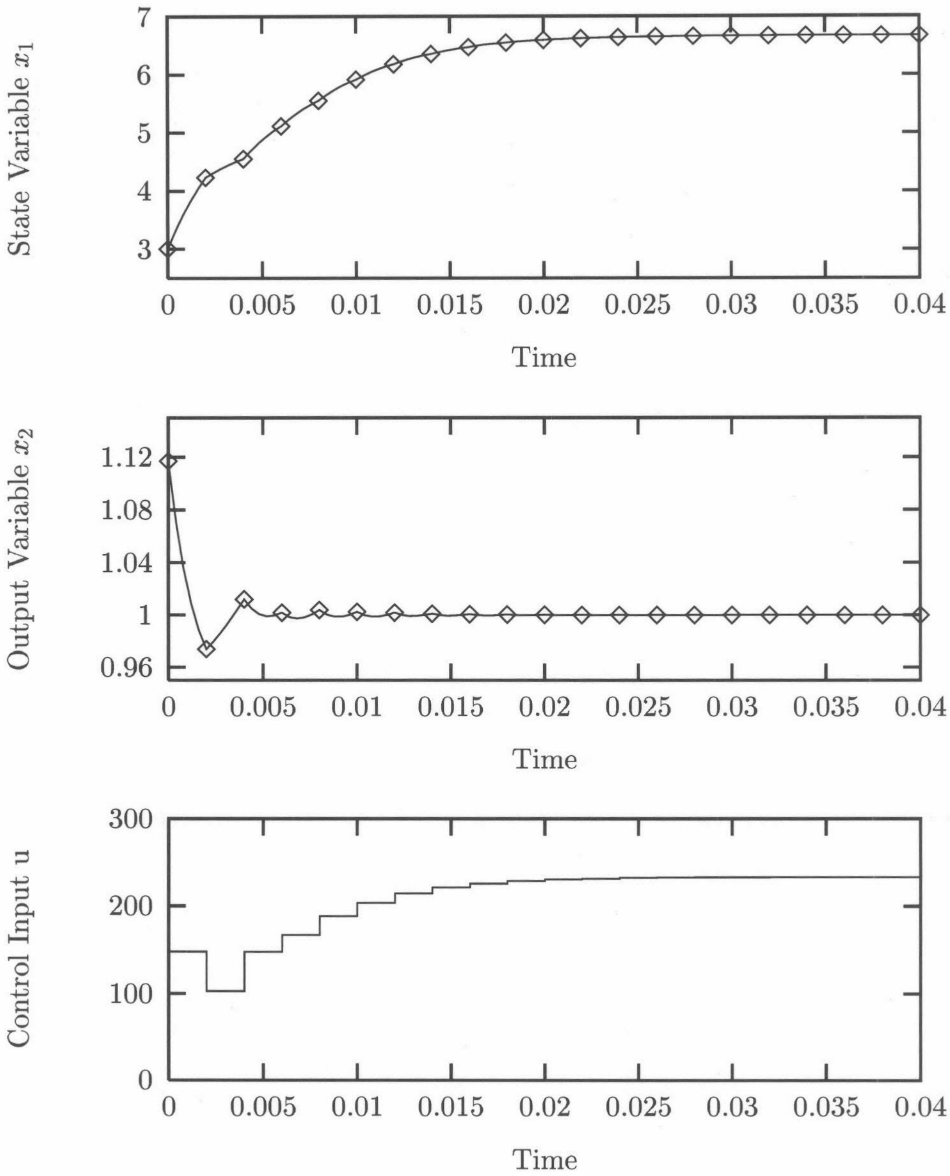

Figure 5.2: Isothermal CSTR: Nominal system with horizon length 5 using only the output.

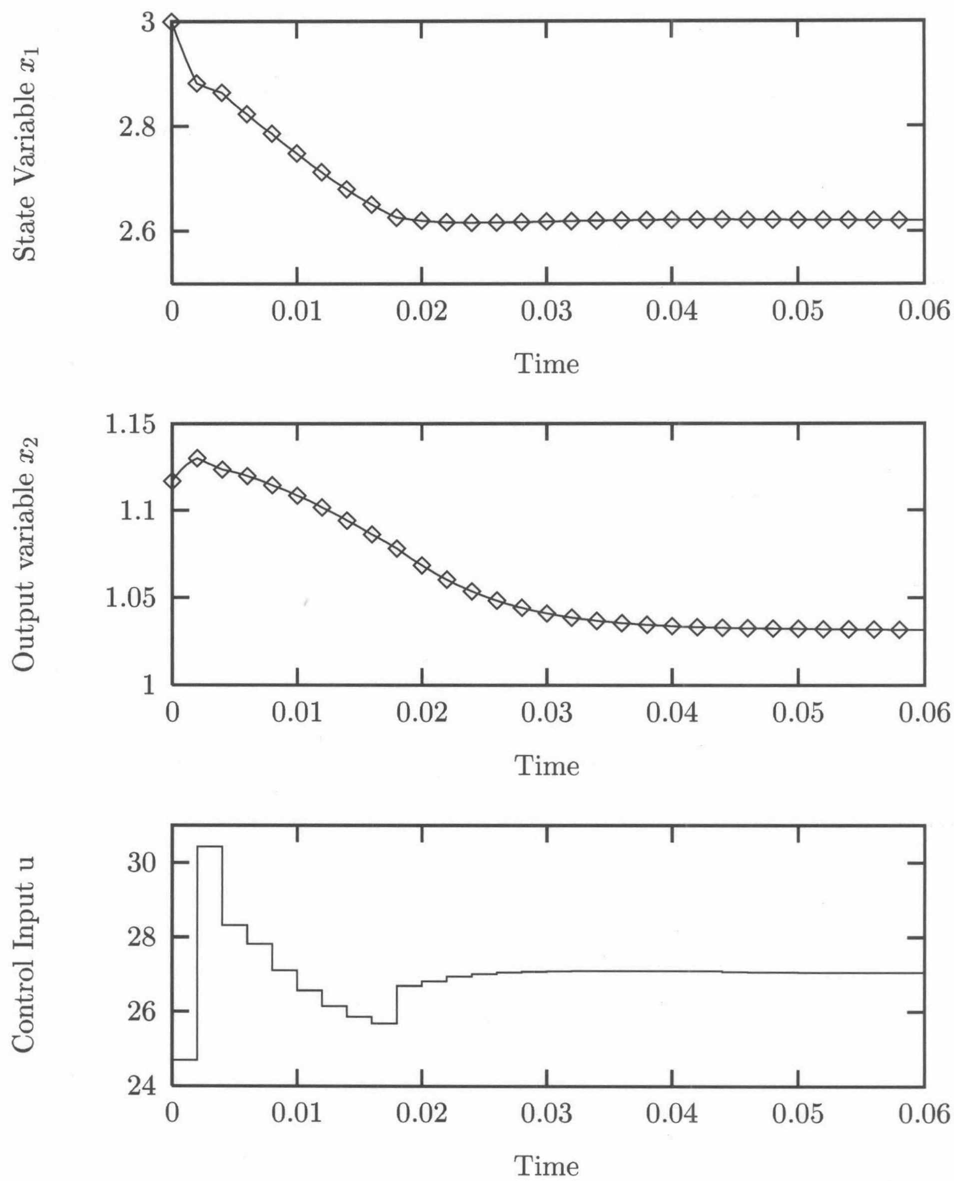

Figure 5.3: Isothermal CSTR: Nominal system with horizon length 5 using only the output with input constraints.

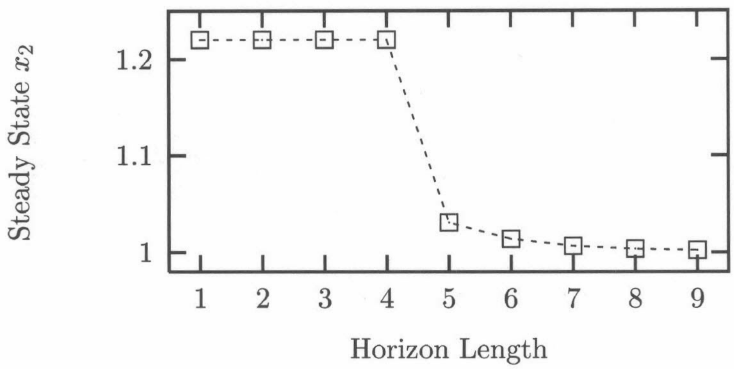

Figure 5.4: Isothermal CSTR: Steady-state output versus horizon length.

subsequent values in the prediction horizon are constrained. With the initial state used in the examples, the initial open-loop control sequence for $N = 5$ is $\{24.7, 50, 50, 50, 50\}$. For horizon lengths greater than 5, the open-loop control profiles start at the *lower* constraint $u_{\min} = 0$ and reach the upper constraint subsequently within the control horizon. For example, with $N = 6$, the open-loop control profile is $\{0, 46.4, 50, 50, 50, 50\}$. To minimize the objective function while satisfying the input constraints, the optimal open-loop control sequence begins at the lower bound and increases during the prediction horizon until it reaches the upper bound.

It should be emphasized that the observed offset is *not* due to model error or unmeasured disturbances. The controller and the simulation both use the nominal model and no noise or disturbances were included. The observed offset is the result of using a controller formulation that is asymptotically stable only in the limit as $N \to \infty$. The control values and corresponding states reflect the best *finite horizon* solution to each constrained optimization problem, but do not necessarily result in zero offset. In the limit as the horizon length approaches infinity, we would expect the closed-loop output to approach its setpoint with zero offset, since steady-state offset would result in an unbounded objective function. This effect can be seen in Figure 5.4.

Since imposing control bounds failed to provide a satisfactory controller, we look for other strategies to improve the performance of the nominal system in closed loop. Figure 5.2 shows that, without the control penalty, the control initially jumps to almost 148 on its way to the final undesirable steady-state value at the undesirable high dilution rate. To prevent excessive jumps in the control, we can include a penalty on $\Delta u_{k+j|k}$. Figure 5.5 shows the effect of including a 1 percent penalty on $\Delta u_{k+j|k}$.

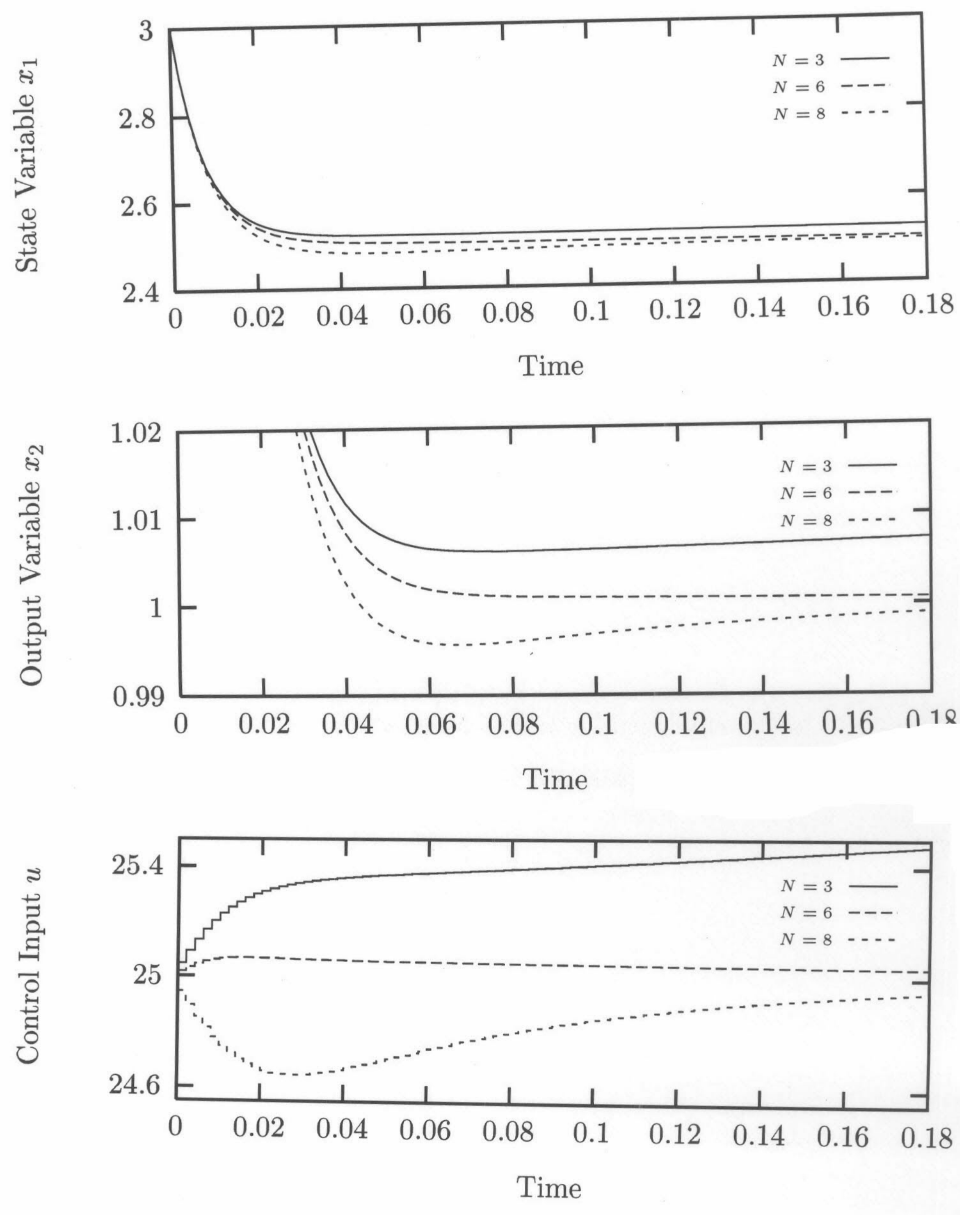

Figure 5.5: Isothermal CSTR: System response with Δu penalty.

For this case, qualitative differences appear in the responses for different horizon lengths. Figure 5.5 shows the results for horizon lengths of 3, 6 and 8. Notice that the speed of response has been greatly slowed due to the penalty on Δu, compared to Figures 5.1 or 5.2. The time scale on the figures has increased by a factor of three from the previous figure. (The use of symbols to identify the sampling instants obscures the results and has been dropped.) We find that for horizons less than 6, the desired steady-state operating point $y = 1.0$ is not asymptotically stable. The case $N = 3$, illustrated in Figure 5.5, shows a continuous increase in the control following the initial transient. The simulation ended before a steady-state value was reached. All horizons shorter that 6 showed similar unstable behavior.

In the other cases illustrated, $N = 6$ and $N = 8$, the controller brings the system to the desired steady state. With $N = 6$, there is no overshoot, while $N = 8$ shows overshoot, both in the control and in the output. The vertical axis of the graph of the output variable, x_2, has been expanded to more clearly show the different behaviors near the setpoint.

When the output variable is x_2 only and the flow rate is low, the linearized reactor model contains a right half-plane zero and therefore exhibits nonminimum phase characteristics. For linear systems, it is known that model predictive control can induce unstable closed-loop behavior for non-minimum phase systems, even if the open-loop system is stable [23]. Some of the linear systems results can be used as a guide to explain the observed behavior of the nominal system under MPC that was examined in this section.

In Figure 5.1, two output were assumed, i.e., $y = [x_1, x_2]^T$. In this case, the system cannot be described as non-minimum phase. Figure 5.2 shows the results of using $y = x_2$, illustrating the non-minimum phase effect. Although the desired setpoint was reached, the control continued to increase until it reached the second steady state at the higher setpoint. This is unstable behavior. A similar non-minimum phase linear system would not have reached an upper steady state, but would have demonstrated control action that continued to increase without limit.

To prevent MPC from returning an unstable controller for non-minimum phase linear systems, it is necessary to penalize control action or decrease the control horizon relative to the prediction horizon [23, 25]. We chose the first course and the results are provided in Figure 5.5.

5.2.2 Fluidized Bed Reactor

In this section, we demonstrate model predictive control through simulation of the fluidized-bed reactor described by Elnashaie, *et al.* [22]. The model describes the catalytic exothermic consecutive reactions

$$\mathrm{A} \xrightarrow{k_1} \mathrm{B} \xrightarrow{k_2} \mathrm{C}$$

in a freely bubbling fluidized-bed reactor. After a series of simplifying assumptions, the dimensionless reactor model is given by the following:

$$\frac{1}{\mathrm{Le}_A}\frac{dx_A}{dt} = \bar{\beta}\left(x_{A,F} - x_A\right) - r_1 \tag{5.8}$$

$$\frac{1}{\mathrm{Le}_B}\frac{dx_B}{dt} = \bar{\beta}\left(x_{B,F} - x_B\right) + r_1 - r_2 \tag{5.9}$$

$$\frac{dT}{dt} = \bar{\beta}\left(T_F - T + u\right) + \beta_1 r_1 + \beta_2 r_2 \tag{5.10}$$

$$r_1 = \alpha_1 \exp\left(-\frac{\gamma_1}{T}\right) x_A \tag{5.11}$$

$$r_2 = \alpha_2 \exp\left(-\frac{\gamma_2}{T}\right) x_B \tag{5.12}$$

In the model, Le_A and Le_B represent Lewis numbers, $\bar{\beta}$ the residence time, α_1, α_2 pre-exponential factors for rate constants, γ_1, γ_2 activation energies, and β_1, β_2 heats of reaction. Concentrations of A and B and the temperature are given by x_A, x_B and T, respectively; $x_{A,F}$, $x_{B,F}$ and T_F represent their values in the inlet stream. For details concerning derivation of the reactor model and the numerical values of the constants, the reader is referred to the original publications [21, 22].

In our version of the model, the control enters the equations in a slightly different way than in the original work of Elnashaie *et al.* [22]. Their paper was concerned with chaotic behavior of this reactor using proportional control. Their original model can be recovered by setting $u = K\left(T_{\mathrm{set}} - T\right)$.

The measured output of the system is the temperature of the reactor effluent T. The control objective is to maintain the effluent temperature at the setpoint T_{set} that corresponds to the maximum yield of species B. Using the nominal model, we computed this optimal (dimensionless) effluent temperature of 0.9213, which will be used as the setpoint for our simulation studies. At this temperature, the nominal steady state for the reactor is $x_A =$

0.2355, $x_B = 0.6448$, which is an unstable steady state. The eigenvalues of the linearized system matrix at this point are $\{-0.2706, -0.0507, 0.6323\}$.

No sampling interval was specified by Elnashaie *et al.*, since their study was not concerned with digital control. We selected a sampling interval of 1.09 dimensionless time units, slightly shorter than the doubling time of the unstable mode. Using the objective function

$$J = 100 \sum_{k=0}^{N-1} \int_{k\,\Delta T}^{(k+1)\,\Delta T} [T(\tau) - 0.9213]^2 \, d\tau + \beta (\Delta u_k)^2$$

with prediction and control horizon equal to 4, simulation results for the nominal system are provided in Figure 5.6. As in the previous example, we employed the orthogonal collocation technique in a simultaneous model solution/optimization approach. Both cases $\beta = 0$ and $\beta = 0.1$ are shown. Without any penalty on control action, Figure 5.6 shows a control range of approximately -2 to 3.5 with a maximum $|\Delta u|$ of approximately 5.5. Based on the values of the numerical results reported by Elnashaie *et al.* [22], we can infer that this control is probably more aggressive than the system can support. Using a 0.1 percent penalty on Δu ($\beta = 0.1$) moderates the control action and yields a slower response by the internal states and the output variable.

Another common approach that dampens excessive control action is to select a control horizon M that is shorter than the prediction horizon N. After the M-th interval, the control is held constant through the end of the prediction horizon. Figure 5.7 shows the results of this approach for $N = 4$ without control penalty ($\beta = 0$), using control horizons $M = 1$ and $M = 2$. Using $M = 1$ greatly dampens the control input and slows the response of the system. Note the increase in scale in Figure 5.7 necessary to illustrate the transient of x_A and x_B. Increasing the prediction horizon from $M = 1$ to $M = 2$ increases both the aggressiveness of the control action and the speed of the system response. With $M = 2$ we observed system behavior very similar to that shown in Figure 5.6 using the 0.1 percent penalty on Δu.

5.3 Models for MPC

The heart of model predictive control is the process model itself. Models can be classified by various features. Since MPC requires the solution of a model to predict future process outputs, the form of the model selected has

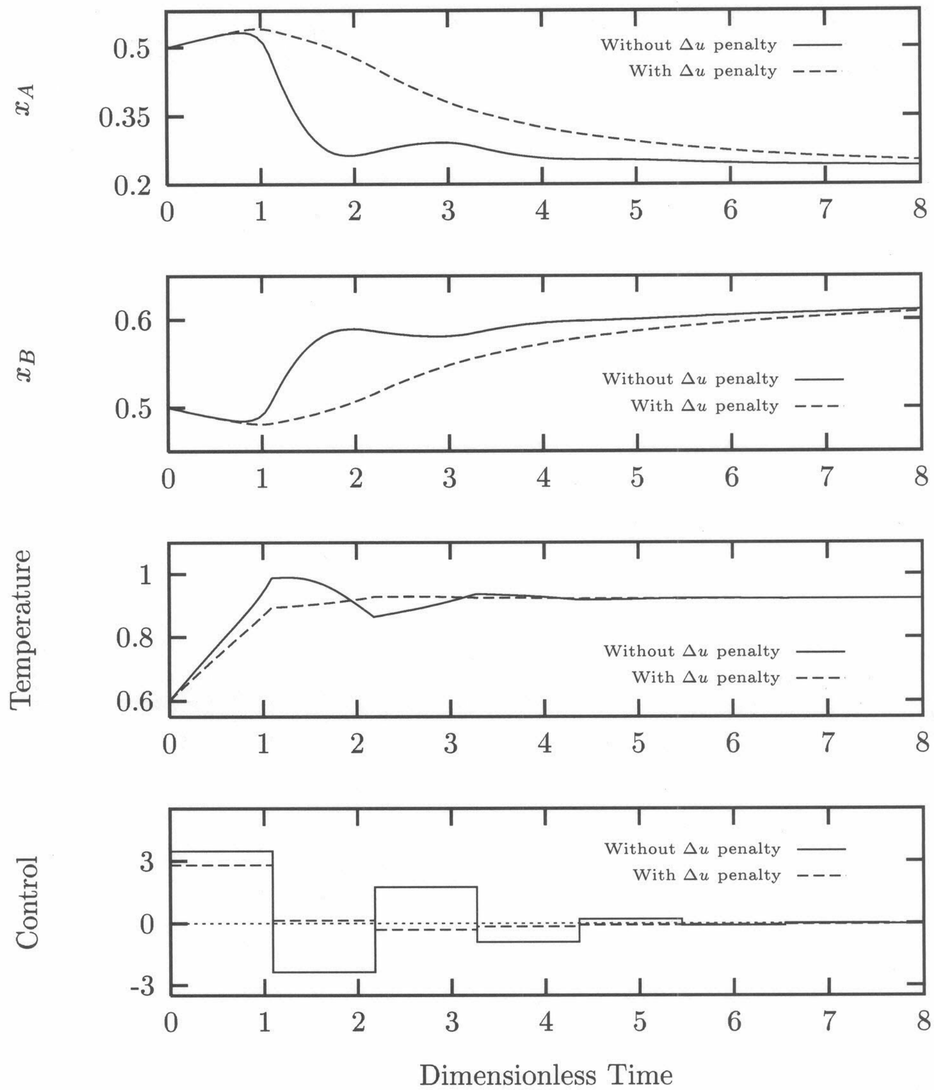

Figure 5.6: Fluidized-bed reactor: Effect of penalty on Δu.

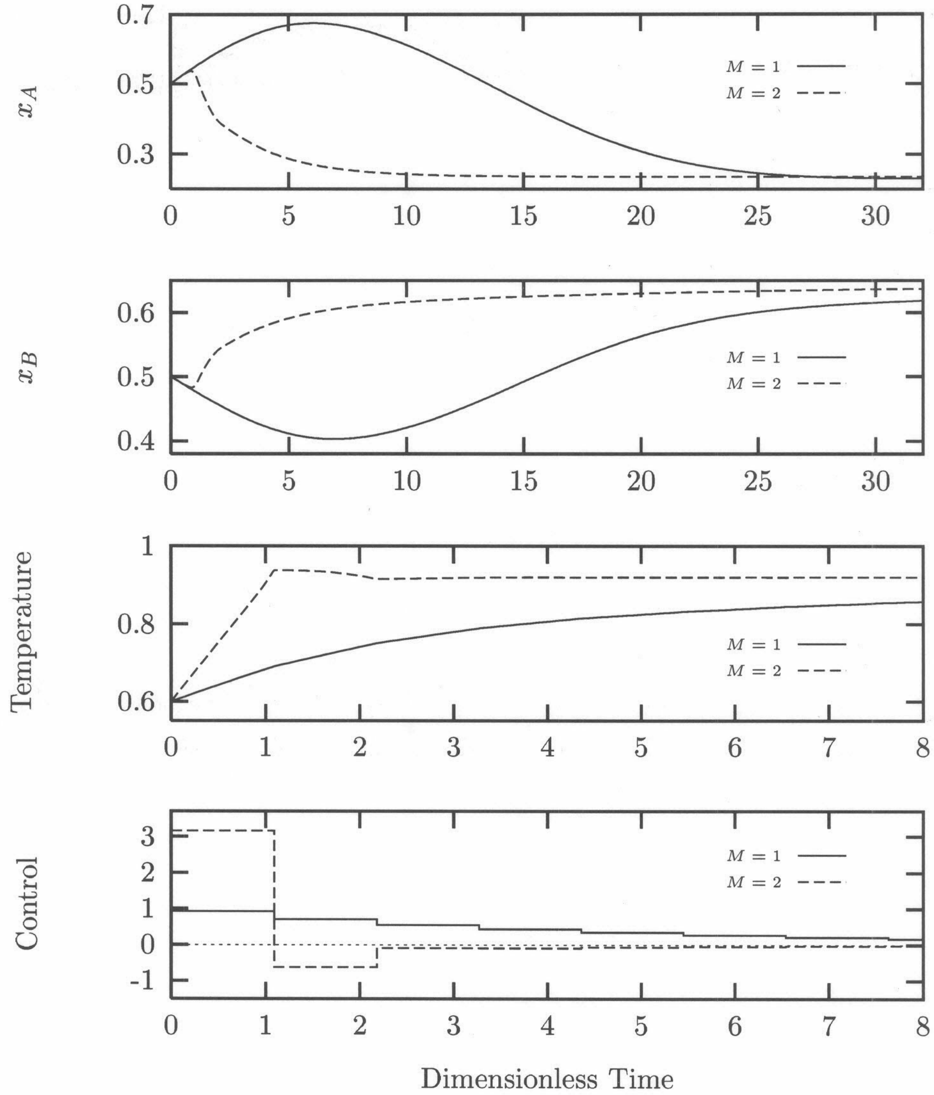

Figure 5.7: Fluidized-bed reactor: Effect of control horizon.

a large impact on our ability to implement MPC. Some specific categories of models are discussed below:

- Linear or nonlinear: The response of linear dynamic systems obeys the principle of superposition, i.e., the response of the system to a linear combination of inputs is a linear combination of their responses to each input separately. Many (perhaps most) systems of engineering interest approximate this behavior for small inputs, which accounts for the universal study and application of linear control theory. In almost any control application, linear design techniques are usually the first to be attempted and are completely satisfactory for many engineering applications, especially those involving regulation about a steady-state operating point. Linear models are used extensively in the industrial practice of MPC.

  Nonlinear models have no specific characteristics except that they don't include the linear case. This makes it difficult to generalize, since nonlinear models can be devised to have almost any characteristic or demonstrate any strange behavior. For many engineering applications, it is usually appropriate to assume that the models are continuous with respect to model parameters and that differential or difference equation models satisfy growth conditions that permit solutions to be computed.

- Continuous-time or discrete-time: Most of the physical laws that are used by engineers to develop models are presented as differential equations with time as the independent variable. A typical representation is of the form $\dot{x} = f(x, u)$. Before the widespread availability of digital computers, differential equation models were the central tool of researchers and control engineers for the study of dynamic systems. If simulations were needed, analog computers could be used. Prediction was rarely used as an element of control engineering, since complex models could not be solved using paper-and-pencil methods.

  With the advent of digital computers, the study of difference equations, which had been relegated to a minor role previously, assumed a new significance. We can write a nonlinear difference equation as $x_{k+1} = f(x_k, u_k)$. Since MPC is universally implemented via digital computer, difference equations are the model of choice. Differential equation models (as in Section 5.2) must be discretized for computer solution. In this chapter, our examples are based on continuous-time models that are discretized using orthogonal collocation.

- Distributed parameter or lumped parameter: A distributed parameter model involves partial differential equations, instead of ordinary differential equations. An example of a distributed parameter model would be that for a plug flow reactor, in which the changes in chemical concentrations in the reactor are subject to both spatial variation and variations in time. Although we do not specifically consider distributed parameter models in this chapter, the basic MPC concept is completely applicable to systems described by distributed parameter models, with a corresponding increase in computations necessary to solve the model.

- Deterministic or stochastic: All physical processes are subject to unpredictable disturbances. Disturbances can affect MPC control design and operation in at least two distinct ways:

  - In the process of model identification, a model is often selected from some class based on experimental results. The selection process implicitly or explicitly uses assumptions about the disturbances to select and evaluate the best model of the class. These assumptions have a direct impact on the model selected.
  - After the model identification phase is complete, the assumptions about the disturbances are sometimes discarded, and control design may be based on this "nominal" model. If the model allows us to predict the statistics of process variables based on assumptions about random effects on the models, we say that it is a stochastic model. We can represent discrete-time stochastic models using w and v to represent state and output disturbances as follows:

$$x_{k+1} = f(x_k, u_k, w_k) \quad (5.13)$$

$$y_k = g(x_k, v_k) \quad (5.14)$$

  In general, better control can be achieved by using explicit stochastic models [16, 17, 51] in the prediction phase of MPC, but with a much higher computational burden. The traditional approach to stochastic MPC is through dynamic programming [2, 3], which suffers from combinatorial growth in the number of optimization variables with increasing horizon length (Bellman's "curse of dimensionality"). Better ways to incorporate stochastic models in MPC are active areas of research.

- Input-output or state-space: As indicated by the name, input-output models provide a relation between the process input and the output, without reference to variables internal to the process. An input-output model of a distillation column, for example, might relate the temperature of a side-product stream to reboiler duty, without consideration of individual tray temperatures or compositions; whereas, a state-space model might include equations relating all internal compositions and flow rates to the reboiler duty, with the side product temperature being provided as a function of the composition. Because most nonlinear state-space models are based on heat, mass and momentum balances, each state has a physical interpretation. States may also be generated as mathematically convenient intermediate variables of an input-output process model. In the examples discussed in this chapter, the states have definite physical significance; however, the principles of MPC are applicable in either case.

  As a special class of input-output model, we should mention that artificial neural network models are becoming important in many engineering applications, including model predictive control, with a number of researchers providing important new results [39, 40, 45, 59, 61, 20]. This chapter does not specifically discuss neural networks in MPC, although the above cited works make clear that artificial neural networks may be successfully used in MPC. Chapter 7 by McAvoy in this book contains further discussion of the use of artificial neural networks in control.

- Frequency domain or time domain: Frequency domain models in continuous time are based on the Laplace transform of continuous linear systems and are not used for model predictive control, except for the linear unconstrained problem. For frequency domain descriptions in discrete time, it is not possible to be so definitive, since the shift operation (q or q^{-1}) is often interchanged with the z transform operator. Depending on the context, these models may be described as input-output or discrete-time frequency domain models. Since few of the concepts of frequency domain analysis are applicable for nonlinear systems, frequency domain concepts play little or no role in nonlinear MPC.

- First-principles or "black box": Models that are derived from heat, mass and momentum balances are frequently called "first-principles"

> or "fundamental" models, in contrast to other modeling schemes that fit a data set to an arbitrary form. Both approaches have been used successfully in MPC applications. An example of the "black box" approach is that used in model identification using step or impulse response models, in which model coefficients are fitted to process data using statistical methods without regard to underlying physical principles. The first principles approach is reflected in the models used in Section 5.2. Basic principles of chemistry were applied to arrive at a model of the chemical reactions involved and mass balances were used to create an overall reactor model.
>
> The clearest tradeoff between the first-principles and black-box models lies in our confidence in predictions made by the model. A first-principles model presumably can be used to predict over a wide range of conditions, even without prior operating experience, provided that the basic assumptions of the model remain valid. On the other hand, a model based on (for example) artificial neural networks has almost no predictive value outside the range of operating conditions where data has been collected.
>
> The best features of both approaches can perhaps be incorporated into a hybrid approach in which fundamental modeling is used for those portions of a process where the physical phenomena are well understood, and an input-output model used for the remainder of the process model.

Each of these model characteristics has an impact on the implementation of MPC. Since MPC is an on-line control method, the speed of the computer algorithm used is essential. Linear models are very well suited to MPC because they may be solved quickly and the optimization problem may be posed as linear or quadratic programming problems, for which robust and reliable software is available. Using continuous models in MPC also impacts the speed of solution because it usually requires significantly more variables to represent a discretized model. Distributed parameter models require more time to solve because of the additional difficulty in solving partial differential equation models.

From the class of input-output models, step or impulse response models have been favored for industrial application. In view of recent results concerning the use of linear state-space models [55], there appears to be little advantage in continuing with this practice. An equivalent state-space model is no less easy to identify, contains fewer unknown coefficients, can be evaluated more quickly by computer and can represent a wider range of processes.

For the nonlinear case, the decision to use first-principles or input-output models is less clear. For small systems with well-understood physical phenomena, fundamental modeling is preferable, because of the ability of the model to predict beyond the range of existing operating data. On the other hand, model identification is easier for larger systems using black-box models.

In conventional industrial MPC, installation of an MPC controller usually begins with identification of a linear input-output model in discrete time. Little or no effort is made to determine a first-principles model based on mass, energy or momentum balances. The objective of the modeling process is to determine a model that can be numerically evaluated quickly and that adequately describes the process dynamics in a neighborhood of some desired steady-state operating point. MPC based on these models has been extraordinarily successful in industry, but it is only applicable for stable processes that operate continuously near the nominal steady-state values. Although recent research results have extended the use of linear models for unstable processes [55], the range of operating conditions remains limited to those near the nominal steady state.

MPC using nonlinear models is especially suited for batch or semi-batch operations in which process conditions can vary significantly over the course of a batch, or for continuous processes that are expected to experience widely varying operating conditions.

In the past, the expense of computation often prevented even the consideration of the use of nonlinear models. As the power of computers available for on-line computations continues to increase, it has become feasible to consider using more complex models of various kinds. Today, the principle limitations in applying nonlinear models for on-line process control are model identification and robustness of control software and algorithms. A discussion of current research issues is contained in Section 5.9.

5.4 Computational Issues

The MPC control depends on finding solutions to a nonlinear programming problem at each sample time. This section discusses solutions techniques that are readily implementable using current optimization software. Since the complexity of the nonlinear program (NLP) is directly related to the model form, we consider several of the most common forms.

To solve the model predictive control problem, it is necessary to both solve an optimization problem and solve the system model. These two procedures may be implemented either sequentially or simultaneously. For sequential solutions, the model is solved at each iteration of the optimization routine. The controls are the decision variables, which are supplied to a routine that computes the model solution. The model solution is then used to compute the objective function, and the computed value is returned to the optimization code. Since gradient information is used by most modern optimization routines, it can be obtained through numerical differentiation (not recommended, see below), involving repeated calls to the routine that solves the model, or the sensitivity equations can be evaluated along with the model equations using, for example, the software routine DDASAC [7]. Our experience using numerical derivatives obtained through finite differences in the sequential approach has been strongly negative. To obtain gradients using finite difference typically involves differencing the output of an integration routine with adaptive step size. Although tightly bounded by the integration algorithm, the integration error is unpredictable. Differencing such an inexact quantity greatly degrades the quality of the finite difference derivatives. For this reason, using finite difference derivatives in the sequential approach cannot be recommended. See Gill *et al.* [27] for a further discussion of this issue.

In contrast, simultaneous model solution and optimization includes both the model states and controls as decision variables and the model equations are appended to the optimization problem as equality constraints. This can greatly increase the size of the optimization problem, leading to a tradeoff between the two approaches. For small problems with few states and a short prediction horizon, the sequential method using sensitivity equations for derivatives is probably the better approach. For larger problems, we have found that the simultaneous approach generally is more robust, i.e., the optimization software is less likely to fail. For problems with state or output variable constraints in continuous time, the sequential approach is more complicated. An approach to incorporating state variable constraints in continuous time was suggested by Sargent and Sullivan [66] that requires an additional state variable to be defined for each state or output variable constraint.

The remainder of this section discusses implementation issues for MPC using discrete-time and continuous-time models and the simultaneous optimization/model solution approach.

5.4.1 Discrete-Time Models

In this section, we consider the solution of MPC problems posed solely in discrete time. Since the computational issues discussed here are primarily concerned with solving the open-loop, constrained control problem, we will temporarily abandon the double time index. The model is given by Equations 5.1 and 5.2 and the objective function is given by

$$J = \|g(x_N) - y_{\text{ref}}\|_Q^2 + \sum_{j=0}^{N-1} \left(\|g(x_j) - y_{\text{ref}}\|_Q^2 + \|u_j - u_{\text{ref}}\|_R^2 + \|\Delta u_j\|_S^2\right) \tag{5.15}$$

For the unconstrained problem, or problems with only control constraints, it is possible to substitute the model equations directly into Equation 5.15 and to use only the controls as optimization variables (sequential approach). As noted previously, we have observed ***significantly*** better performance from optimization codes if each state and control is considered an independent optimization variable, with the model equations appended as nonlinear equality constraints:

$$\begin{aligned}
x_1 - f(x_0, u_0) &= 0 \\
x_2 - f(x_1, u_1) &= 0 \\
\vdots \quad &\vdots \quad \vdots \\
x_N - f(x_{N-1}, u_{N-1}) &= 0
\end{aligned}$$

Considering states and controls as independent decision variables, the size of the vector of decision variables is $n(N+1) + Nm$. Most optimization software requires the vector to be passed as a single argument. We have found it most convenient to partition the entire vector of decision variables as $\left[x_0^T \; x_1^T \cdots x_N^T \; u_0^T \; u_1^T \; \cdots u_{N-1}\right]$.

MPC is characterized in part by its ability to incorporate constraints. In chemical process applications, constraints are most commonly expressed as simple bounds on the problem variables:

$$\begin{aligned}
y_{\min} &\le y_j \le y_{\max} \\
u_{\min} &\le u_j \le u_{\max} \\
\Delta u_{\min} &\le \Delta u_j \le \Delta u_{\max}
\end{aligned} \tag{5.16}$$

The control constraints of Equation 5.16 can be expressed as a set of linear inequality constraint

$$\begin{bmatrix} C \\ -C \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} \leq \begin{bmatrix} c_{\text{ub}} \\ c_{\text{lb}} \end{bmatrix}$$

in which C, c_{ub} and c_{lb} are given by

$$C = \begin{bmatrix} I & & & & \\ -I & I & & & \\ & -I & I & & \\ & & \ddots & \ddots & \\ & & & -I & I \end{bmatrix}$$

$$c_{\text{ub}} = \begin{bmatrix} u_{-1} + \Delta u_{\max} \\ \Delta u_{\max} \\ \vdots \\ \Delta u_{\max} \end{bmatrix} \qquad c_{\text{lb}} = \begin{bmatrix} -u_{-1} - \Delta u_{\min} \\ -\Delta u_{\min} \\ \vdots \\ -\Delta u_{\min} \end{bmatrix}$$

The form of the above constraint illustrates an example of another commonly used constraint, the general linear inequality constraint of the form

$$\begin{aligned} Dx_k &\leq d \\ Hu_k &\leq h \end{aligned}$$

These constraints are applied to each predicted state or control in the horizon. The complete set of constraints may then be expressed using the Kronecker matrix product

$$(I_N \otimes D) \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \leq \mathbf{1}_N \otimes d \qquad (I_N \otimes H) \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} \leq \mathbf{1}_N \otimes h$$

in which I_N is the N-dimensional identity matrix and $\mathbf{1}_N$ is an N-dimensional vector whose elements are equal to 1.

Sometimes it provides better closed-loop performance to use a control horizon M that is less than the prediction horizon N, as demonstrated in the simulation results for the fluidized-bed reactor. For $M < N$, we add additional linear equality constraints $u_{N-1-M} = u_{N-M} = \cdots = u_{N-1}$.

The linear bounds discussed above are sufficiently general to include simple bounds on the decision variables. Even so, it is usually preferable to avoid including them in the general formulation for linear constraints, for two reasons: Since most optimization software permits simple bounds to be included in the argument list of the calling routines, it simplifies the programming task to use the available program interface. More importantly, some implementations include special considerations for simple bounds [27] that can reduce the computational effort necessary to solve the nonlinear program.

Most optimization software contain routines to use finite differences to compute gradients. We strongly recommend against use of finite difference gradients except perhaps for the smallest and simplest problems. Gradients (and Jacobians) should be provided if possible to optimization routines to assure good results. For the objective function of Equation 5.15, the gradient may be expressed by

$$\nabla J = \begin{bmatrix} 2G_0^T Q\left[g(x_0) - y_{\text{ref}}\right] \\ 2G_1^T Q\left[g(x_1) - y_{\text{ref}}\right] \\ \vdots \\ 2G_N^T Q\left[g(x_N) - y_{\text{ref}}\right] \\ 2R\,(u_0 - u_{\text{ref}}) + 2S(2u_0 - u_{-1} - u_1) \\ 2R\,(u_1 - u_{\text{ref}}) + 2S(2u_1 - u_0 - u_2) \\ \vdots \\ 2R\,(u_{N-1} - u_{\text{ref}}) + 2S\,(u_{N-1} - u_{N-2}) \end{bmatrix}$$

in which G_k is the Jacobian matrix of $g(x)$ evaluated at (x_k, u_k).

The Jacobian of the equality constraints representing the system equations is sparse and has the following general form:

$$\left[\begin{array}{ccccccccc}
I & & & & & & & & \\
-F_{x,0} & I & & & & -F_{u,0} & & & \\
 & -F_{x,1} & \mathrm{I} & & & & -F_{u,1} & & \\
 & & -F_{x,2} & \mathrm{I} & & & & -F_{u,2} & \\
 & & & -F_{x,3} & \mathrm{I} & & & & -F_{u,3}
\end{array}\right]$$

The Jacobian of the inequality constraints is also sparse with the following structure:

$$\left[\begin{array}{cccccccc}
D & & & & & & & \\
 & D & & & & & & \\
 & & D & & & & & \\
 & & & D & & & & \\
 & & & & H & & & \\
 & & & & & H & & \\
 & & & & & & H & \\
 & & & & & & & H \\
 & & & & I & & & \\
 & & & & -I & I & & \\
 & & & & & -I & I & \\
 & & & & & & -I & I \\
 & & & & -I & & & \\
 & & & & I & -I & & \\
 & & & & & I & -I & \\
 & & & & & & I & -I
\end{array}\right]$$

The above examples represent an MPC problem using a prediction horizon of 5, with linear constraints on state, control variable and control increments at each time step.

5.4.2 Continuous-Time Models

For continuous-time systems, we consider here nonlinear models represented as differential equations of the general form

$$\dot{x} = f(x, u, t) \tag{5.17}$$

A controller is sought which minimizes some performance index

$$J = \phi\left[x\left(t_f\right), t_f\right] + \int_{t_0}^{t_f} L\left[x(t), u(t), t\right] dt \tag{5.18}$$

We can find necessary conditions for a minimum using variational techniques, but in general these do not form a practical basis for implementation. We seek an approximate solution using piecewise constant controls with intervals that correspond to some appropriate sampling frequency of the process. Optimal controls for the system of Section 5.2 were obtained using the method of orthogonal collocation on finite elements, a technique based on numerical integration using Gaussian quadrature. We present here a brief tutorial description of the method. See the following for a more detailed theoretical discussion: [4, 15, 12, 77, 78].

Consider the solution of minimizing the objective function of Equation 5.18 over a single time element. In the isothermal CSTR of Section 5.2, the time interval was 0.002 hours or 7.2 seconds. At the beginning of the time interval the state (or state estimate) is known and we seek a single control, constant over the interval, that minimizes the objective function.

The orthogonal collocation technique assumes that the solution to Equation 5.17 can be approximated by an interpolating polynomial, expressed as the sum of Lagrange polynomials [14], with nodes located at the roots of an orthogonal family of polynomials. By this choice of nodes, integration using the orthogonal collocation technique is exact for polynomials of degree $2n-1$, although only a polynomial of degree n is used for the interpolation. See Davis and Rabinowitz [15] for a complete explanation of this remarkable result.

In Section 5.2, all computations were performed using either two or three internal collocation points and one on the boundaries of each finite element. Three internal collocation points are illustrated in Figure 5.8. If the element is scaled so that $t_1 = 0$ and $t_5 = 1$ then the node points are given by $\{t_1, t_2, t_3, t_4, t_5\} = \{0, 0.1127, 0.5, 0.8873, 1.0\}$, which are the roots of the third order Legendre polynomial, augmented by the element endpoints. If $t_1 \neq 0$ or $t_5 \neq 1$, then a linear change of variable is necessary to transform the time interval to $[0, 1]$. When the dynamic model is time-invariant, the only correction necessary is to scale the right-hand side of Equation 5.17 by $t_5 - t_1$.

With these choices of interpolating polynomials and node points, derivatives and integrals can be represented by linear combinations of the state values at the node points. Let the jth state at node i be represented by x_{ij}. We admit the notation $x_{i,*}^T$ to indicate the entire state at node i. With constant control, an approximate solution to the differential equation can be obtained by solving the following nonlinear equation:

$$A_0 X = F(X, u) \tag{5.19}$$

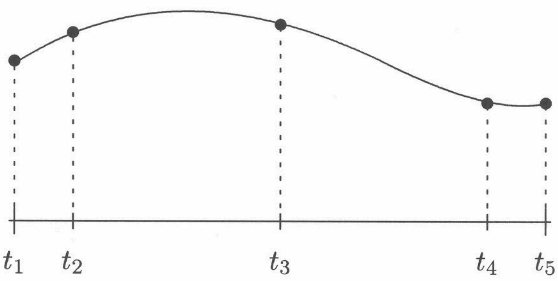

Figure 5.8: Collocation element.

in which

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ x_{3,1} & x_{3,2} & \cdots & x_{3,n} \\ x_{4,1} & x_{4,2} & \cdots & x_{4,n} \\ x_{5,1} & x_{5,2} & \cdots & x_{5,n} \end{bmatrix} \qquad F(X,u) = \begin{bmatrix} x_{\text{init}}^T \\ f^T(x_{2,*}^T, u, t_2) \\ f^T(x_{3,*}^T, u, t_3) \\ f^T(x_{4,*}^t, u, t_4) \\ f^T\left(x_{5,*}^T, u, t_5\right) \end{bmatrix}$$

The matrix A_0 contains the first derivative collocation weights:

$$A_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -5.32 & 3.87 & 2.07 & -1.29 & 0.68 \\ 1.5 & -3.23 & 0 & 3.23 & -1.5 \\ -0.68 & 1.29 & -2.07 & -3.87 & 5.32 \\ 1 & -1.88 & 2.67 & -14.79 & 13 \end{bmatrix}$$

(The subscript 0 is included to emphasize that the first line of the collocation weight matrix is added to account for the initial condition.) The objective function of Equation 5.18 is approximated by

$$J \approx \phi\left(x_{5*}^T, t_5\right) + \sum_{k=1}^{5} c_k\, L\left(x_{k*}^T, u, t_k\right) \tag{5.20}$$

in which the coefficients c_k are the quadrature weights $\left\{0, \frac{5}{18}, \frac{4}{9}, \frac{5}{18}, 0\right\}$ corresponding to the third order Legendre polynomial. For other choices of orthogonal polynomials and node points, the collocation weight matrices and quadrature weights may be computed using the FORTRAN programs of Villadsen and Michelsen [77] or using root-finding techniques using the orthogonal polynomials [15]. In the examples of this chapter, collocation and quadrature weights were computed using the `octave` [18] interface to the Villadsen and Michelson routines.

With these choices, the control is obtained as the solution to the following nonlinear program

$$\min_{u,X} J$$

subject to:

$$A_0 X = F(X, u) \tag{5.21}$$

which represents a discretized form of the MPC problem with horizon length 1.

For horizon lengths greater than 1, we can extend the orthogonal collocation method to a set of finite elements in time, with the solution of the problem approximated by polynomials defined on each element. This situation is illustrated in Figure 5.9. Note that we have chosen an indexing scheme that consecutively numbers nodes across finite element boundaries. It would perhaps be more natural to adopt an additional index to indicate to which finite element a state or control belongs; however, this would result in triple indices for states, controls and outputs, which are not supported in most programming languages or any known optimization software implementations. The structure shown in Figure 5.9 corresponds to the collocation scheme used for the isothermal CSTR example of Section 5.2. The fixed-bed reactor example used one fewer interior collocation point. The matrix A_0 in

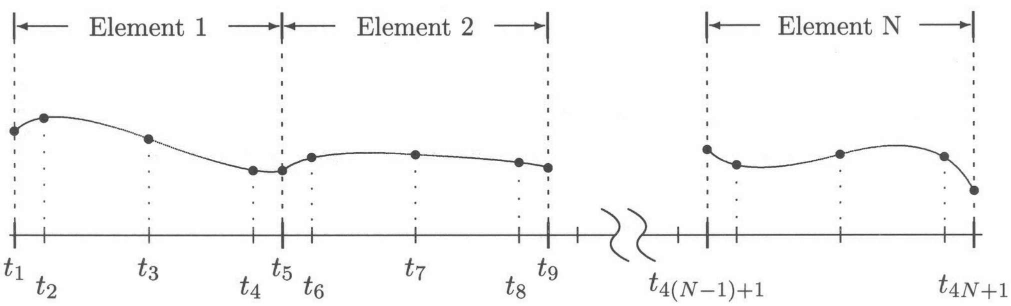

Figure 5.9: Collocation on finite elements.

Equation 5.21 become a partitioned matrix that is almost block diagonal:

$$\tilde{A} = \begin{bmatrix} \boxed{A_0} & & & & & \\ & \boxed{A} & & & & \\ & & \boxed{A} & & & \\ & & & \ddots & & \\ & & & & \boxed{A} & \\ & & & & & \boxed{A} \end{bmatrix} \tag{5.22}$$

The overlap is due to the presence of the points on the boundaries of two adjacent finite elements. The state vector includes contributions from states in every multiple finite element and so becomes

$$\tilde{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ x_{3,1} & x_{3,2} & \cdots & x_{3,n} \\ x_{4,1} & x_{4,2} & \cdots & x_{4,n} \\ x_{5,1} & x_{5,2} & \cdots & x_{5,n} \\ \vdots & \vdots & & \vdots \\ x_{4N-2,1} & x_{4N-2,2} & \cdots & x_{4N-2,n} \\ x_{4N-1,1} & x_{4N-1,2} & \cdots & x_{4N-1,n} \end{bmatrix}$$

The $\tilde{F}$ vector is constructed as in the one-element case, except that the value of the control must change across element boundaries:

$$\tilde{F}(\tilde{X}, u_1, u_2, \ldots, u_{N-1}) = \begin{bmatrix} x_{\text{init}}^T \\ f^T(x_{2*}^T, u_0, t_2) \\ f^T(x_{3*}^T, u_0, t_3) \\ f^T(x_{4*}^T, u_0, t_4) \\ f^T\left(x_{5*}^T, u_0, t_5\right) \\ f^T(x_{6*}^T, u_1, t_6) \\ f^T(x_{7*}^T, u_1, t_7) \\ f^T(x_{8*}^T, u_1, t_8) \\ f^T\left(x_{9*}^T, u_1, t_9\right) \\ f^T(x_{10*}^T, u_2, t_{10}) \\ \vdots \\ f^T(x_{(4N-2)*}^T, u_{N-1}, t_{4N-2}) \\ f^T(x_{(4N-1)*}^T, u_{N-1}, t_{4N-1}) \end{bmatrix}$$

Taking $\tilde{c} = \mathbf{1}_N \otimes c$ (the Kronecker product of $\mathbf{1}_N$ and c) the objective function of Equation 5.18 is approximated by

$$J \approx \phi\left(x^T_{(4N-1)*}, t_{4N-1}\right) + \sum_{j=0}^{N-1} \sum_{k=4j+1}^{4j+5} \tilde{c}_k \, L\left(x^T_{k*}, u_j, t_k\right)$$

Taking $U = \left[u_1^T, u_2^T, \ldots, u_{N-1}^T\right]^T$, we can write the orthogonal collocation approximation as

$$\tilde{A}\tilde{X} = \tilde{F}\left(\tilde{X}, U\right) \tag{5.23}$$

Since most optimization codes pass the optimization variables as a single matrix, the matrix equation of Equation 5.23 must be appropriately indexed. In our examples, we were more interested in coding convenience that in optimal performance, so we reconstructed the state matrix $\tilde{X}$ as shown above within the objective function function. Naturally, choices of indexing schemes are at the discretion of the programmer. For on-line application, we would seek the fastest evaluation method possible and would probably include such items as sparse matrix multiplication and optimization of sparse systems.

Since optimization codes require a single vector of decision variables, the task of re-indexing can become tedious. The Jacobian matrix of the equality constraints is quite sparse, becoming more sparse with larger problems, and the overlapped, quasi-block diagonal structure of $\tilde{A}$ is repeated in the Jacobian. This is illustrated more clearly in Figure 5.10, which shows the non-zero elements for the fluidized-bed reactor example with prediction horizon 3, using 5 internal collocation points in each finite element.

5.5 Feedback and State Observers

Like the LQ-optimal controller, the state-space formulation of nonlinear MPC discussed in this chapter uses state feedback. The current state (or state estimate) is used as an initial condition at each iteration to predict the future behavior and select an optimal control sequence. Model predictive control does not prescribe a specific form for the state observer. This section discusses several possibilities and their effects on closed-loop system behavior, including steady-state tracking in the presence of modeling errors and disturbance rejection.

If the full state is measured at each sampling point, an obvious possibility for feedback would be simply to reset the states to the current observations.

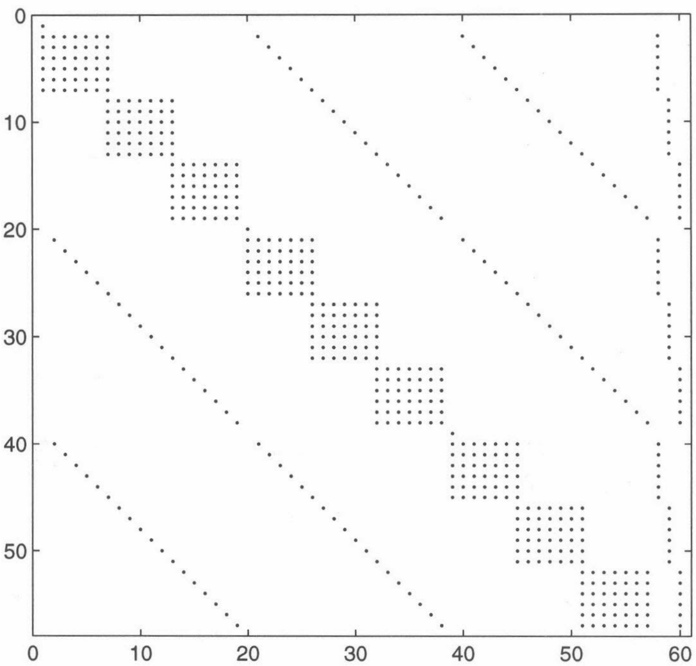

Figure 5.10: Sparse Matrix Structure of Constraint Jacobian.

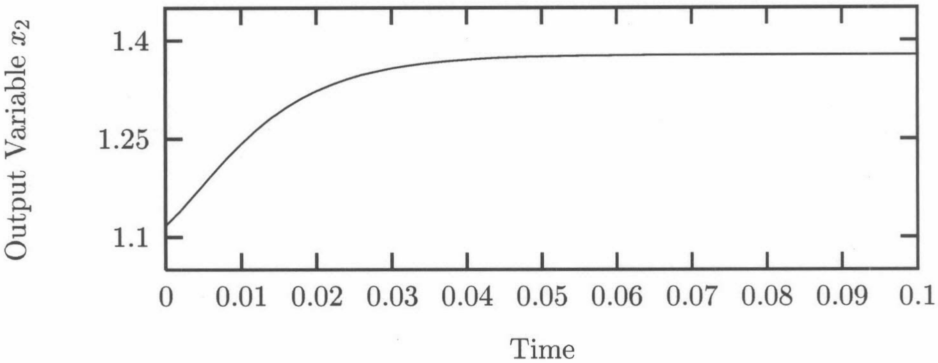

Figure 5.11: Isothermal CSTR: Feedback through resetting of the state.

The updated states then become the initial conditions for the model predictions. For the case without model error, this method works well as shown in Figure 5.1. In the presence of modeling error, simply resetting the state usually produces steady-state offset. We simulated this controller using the model of Section 5.2 and a +50 percent error in the concentration of species A in the the feedstream, x_F. The resulting system output is shown in Figure 5.11. The setpoint was 1.0 and the objective was the same is in Figure 5.5 with horizon length 5. We noted the presence of steady-state offset for horizon lengths 6 through 9 as well. This controller produces offset for *any* finite horizon. Since resetting the state provides such poor results, it is generally not used in practice. We next describe two other approaches with better properties: conventional MPC feedback and MPC with steady-state target optimization.

5.5.1 Conventional MPC Feedback

By far the most common feedback method in MPC is to compare the measured output of the process to the model prediction at time k to generate a disturbance estimate $\hat{d}_k = y_k - \tilde{y}_k$, in which y_k and $\tilde{y}_k$ represent the process measurement and model prediction, respectively. In forming the MPC objective function, the disturbance term is then added to the output prediction over the entire prediction horizon to produce the modified objective

$$J = \sum_{j=0}^{N-1} \|y_{\text{ref}} - \left(y_{k+j|k} - \hat{d}_k\right)\|_Q^2 + \|\Delta u_{k+j|k}\|_S^2$$

This procedure assumes that differences observed between the process output and the model prediction are due to additive step disturbances in the output that persist throughout the prediction horizon. Although simplistic, this error model offers several related practical advantages:

- It accurately models setpoint changes which often enter feedback loops as step disturbances.
- It approximates slowly varying disturbances. Since errors in the model can appear as slowly varying output disturbances, it provide robustness to model error.
- It provides zero offset for step changes in setpoint.

Coupled with a linear step response model, feedback through the estimation of a step disturbance has been extensively applied in industrial applications. The stability and robustness have been analyzed for linear, unconstrained systems by García and Morari [23]. More details for linear systems with an emphasis on robust design using linear models may be found in the the monograph by Morari and Zafiriou [53].

Figure 5.12 illustrates the effect of feedback using the disturbance term with the isothermal CSTR model. The simulation used the same model mismatch, and the same objective function weights as in Figure 5.11. The results for horizon lengths 6 through 9 are included in Figure 5.12. The MPC controller was not stabilizing for the nominal system with horizons less than 6. Instability was observed in the simulations using these horizon lengths for both cases with and without model mismatch.

Feedback through differencing model prediction and measurement does not require a state-space description. *Linear* MPC methods used in industry do not use a state-space model and incorporate the disturbance directly into the MPC objective function. The feedback mechanism is sometimes obscured by this treatment. A state-space description, even if not used in the controller, allows a more natural interpretation of the conventional MPC feedback procedure. Using an augmented state-space model, conventional MPC feedback can be shown to be a particular form of a state observer for the resulting system:

$$x_{k+1} = f(x_k, u_k) \tag{5.24}$$

$$d_{k+1} = d_k \tag{5.25}$$

$$y_k = g(x_k) + d_k \tag{5.26}$$

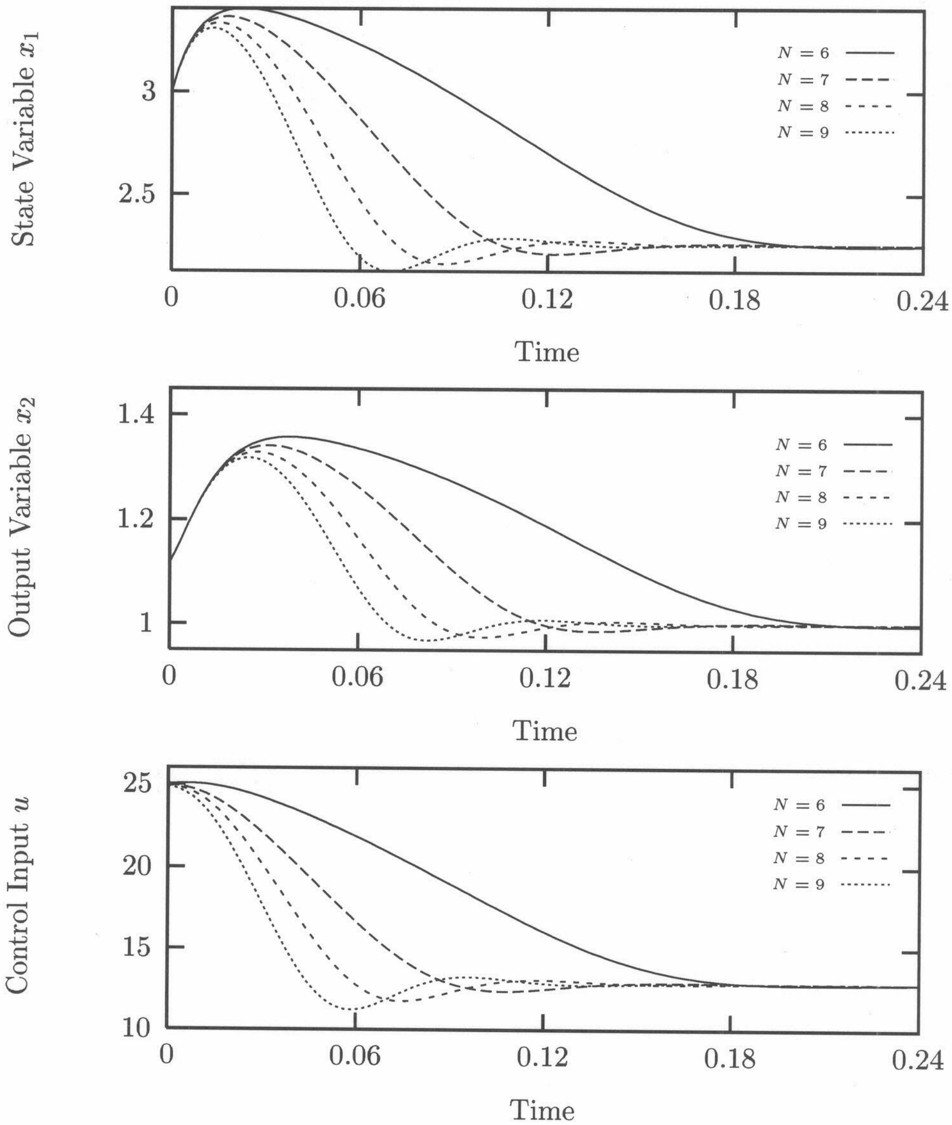

Figure 5.12: Isothermal CSTR: Conventional MPC feedback with model mismatch.

Taking the augmented state to be $\left[x_k^T \;\; d_k^T\right]^T$, we can use a variety of techniques, including the extended Kalman filter or a moving horizon observer [55, 57, 56, 65], to estimate both the state and the disturbance. A general state observer for the augmented system of Equation 5.24 through 5.26 has the form

$$\begin{aligned} \hat{x}_{k+1} &= f(\hat{x}_k, u_k) + h_x\left(\hat{x}_k, \hat{d}_k, y_k, u_k\right) \\ \hat{d}_{k+1} &= \hat{d}_k + h_d\left(\hat{x}_k, \hat{d}_k, y_k, u_k\right) \end{aligned}$$

in which the $\hat{x}_k$ and $\hat{d}_k$ represent the observer estimates. To obtain the usual MPC observer, choose

$$\begin{aligned} h_x\left(\hat{x}_k, \hat{d}_k, y_k, u_k\right) &= 0 \\ h_d\left(\hat{x}_k, \hat{d}_k, y_k, u_k\right) &= y_k - g(\hat{x}_k) - \hat{d}_k \end{aligned}$$

This leaves an open-loop state observer for the original states x_k and a deadbeat observer for the disturbance d_k. In the absence of information to construct a disturbance model, this approach is a good choice in view of its favorable features. With additional information and engineering judgment, it is possible to obtain better state estimates using an observer that updates $\hat{x}_k$ as well as $\hat{d}_k$. Whether the extra effort leads to better control depends upon the specific observer design and knowledge about the disturbances. Several researchers have presented results for linear systems showing better control using a full state estimator [26, 46, 55, 64] in the presence of noisy output measurements and model mismatch.

Since the conventional MPC feedback uses a deadbeat observer of the output disturbance, it is sensitive to random fluctuations in the output. This is illustrated in Figure 5.13, which reproduces the situation of Figure 5.12 except that a zero-mean Gaussian disturbance with variance 0.09 has been added to the output measurement. Figure 5.13 show the system response for horizon lengths 7 and 9. The initial portion of the figures shows the expected faster response of the longer horizon lengths. The faster response comes at the expense of poorer disturbance rejection, i.e. noise in the input.

5.5.2 MPC with Steady-State Target Optimization

We next investigate an alternative feedback method that does not directly incorporate the disturbance term in the MPC objective and can penalize

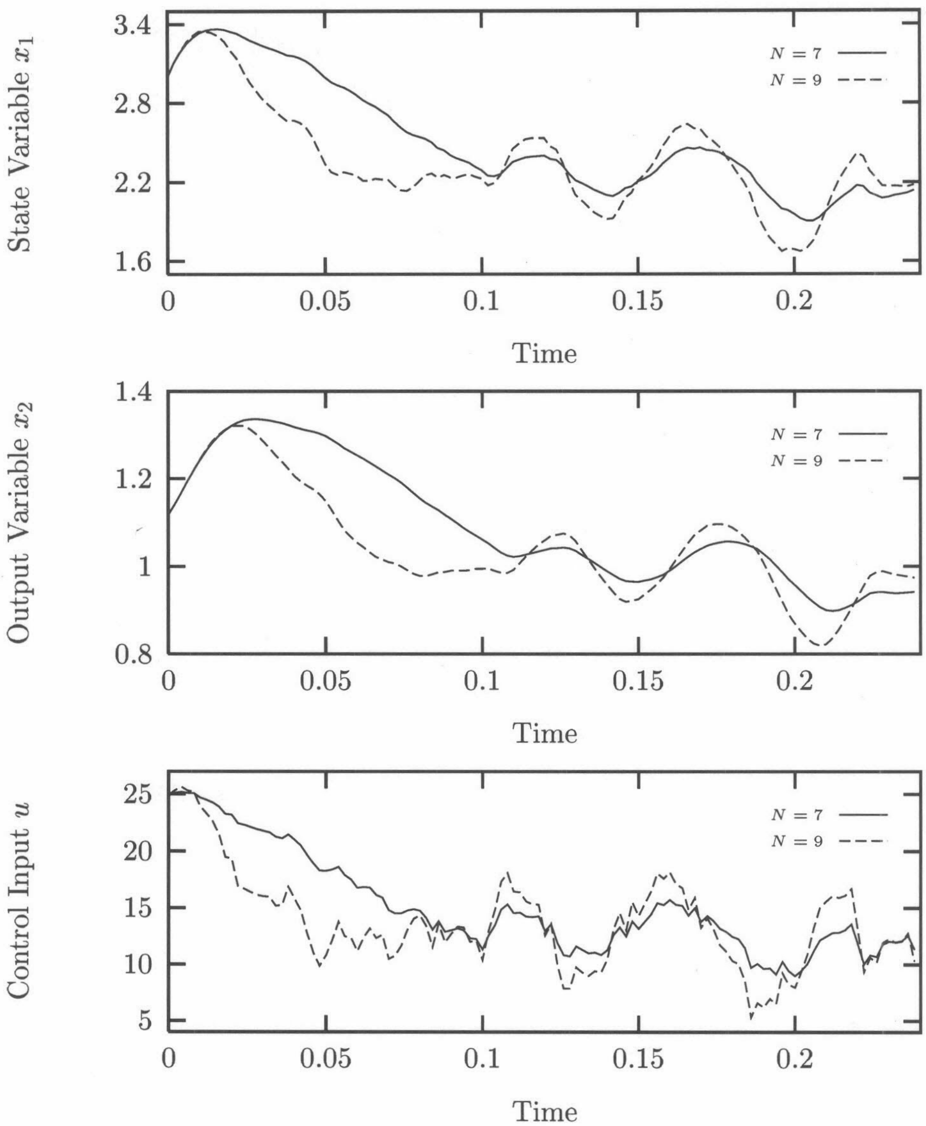

Figure 5.13: Isothermal CSTR: Response of system to Gaussian noise using conventional MPC feedback.

both u and Δu. In this approach, we use a preprocessing step in which new steady-state targets for the state and control, x_t and u_t, are computed based on the disturbance estimate, $\hat{d}_k$, and the output setpoint, y_{ref}. The new targets for the state and control are then incorporated directly into the objective as a penalty on $(x - x_t)$ and $(u - u_t)$. This method is discussed by Muske and Rawlings [55] in the context of linear systems. The method presented here is a direct nonlinear analog.

To apply this method we solve the augmented system of Equations 5.24 through 5.26 at steady state to obtain target values for the internal states, controls or output variable. These are used as setpoints in the MPC objective function, exactly as we did in the first simulation result of Figure 5.1. Figure 5.1 (no model error, no disturbances) shows excellent speed of response without offset. We can duplicate some aspects of that excellent performance using the alternate feedback method.

Using the conventional MPC observer (open-loop state observer with deadbeat disturbance estimator), we can choose steady-state target values by solving the following nonlinear program at each iteration:

$$\min_{u_t, x_t} \|u_{\text{ref}} - u_t\|_R^2 \tag{5.27}$$

subject to:

$$\begin{aligned} x_t &= f(x_t, u_t) \\ y_{\text{ref}} &= g(x_t) + \hat{d}_k \\ x_t &\in \mathcal{X} \\ u_t &\in \mathcal{U} \end{aligned}$$

In the above nonlinear program u_{ref} represents the input setpoint, for problems in which there are extra degrees of freedom in the input vector, i.e. more inputs than output setpoints. The input setpoint is selected with the understanding the the controller will enforce it as a setpoint only in a minimum-norm sense. The solution $\{u_t, x_t\}$ provides target values for the states and controls that account for the current estimate for the output disturbance $\hat{d}_k$. At each iteration, a new estimate of $\hat{d}_k$ is obtained from the state estimator and the target values are readjusted accordingly.

If the above nonlinear programming problem does not have a feasible solution, then the output setpoint cannot be reached exactly; an alternative

NLP may be used to compute the control:

$$\min_{u_t, x_t} \|y_{\text{ref}} - (g(x_t) + \hat{d}_k)\|_Q^2 \tag{5.28}$$

subject to:

$$\begin{aligned} x_t &= f(x_t, u_t) \\ x_t &\in \mathcal{X} \\ u_t &\in \mathcal{U} \end{aligned}$$

In this case, we cannot simultaneously find values of $\{x_t, u_t\}$ to satisfy the steady-state system equations at the desired setpoint. This can happen, for example, if an especially large disturbance enters the system, or if the desired setpoint minus the disturbance does not correspond to a steady state of the model equations. We always assume that the model has at least one real steady-state solution that satisfies the input and output constraints, so Equation 5.28 always has a feasible point.

This technique can give excellent results. Figure 5.14 shows the response of the isothermal CSTR with model mismatch using a 1 percent weight on $(u - u_t)^2$ in relation to the weight on the output. The value of u_t was adjusted at each sampling time using the nonlinear program of Equation 5.27 with $u_{\text{ref}} = 25$, the nominal steady-state value. Unlike the previous examples that used a penalty on Δu, the response is rapid. It satisfies both the primary and secondary control objectives: It brings the system to the setpoint $y = 1$ without offset and it does so at a low feed rate. These results indicate that this is the clear method of choice for control of the reactor of Section 5.2. The curves shown in Figure 5.14 were computed with $N = 5$. Slightly slower response was observed with shorter horizon lengths, but the differences were very small and would not appear as distinct curves if plotted in Figure 5.14. Even with horizon length 1, the response was very similar, and the time necessary to compute the MPC control was much less. Recall that this system was unstable for short horizons using conventional MPC feedback.

Up to this point, all of the isothermal CSTR examples shown have started from the same initial conditions. It is necessary to check the behavior of the controller over a range of initial conditions before deciding on a specific controller design. Figure 5.15 shows trajectories of the system from various initial conditions. The system was controlled using feedback through steady-state target optimization with horizon length 3. The initial conditions were selected to lie along the perimeter of the rectangular region of

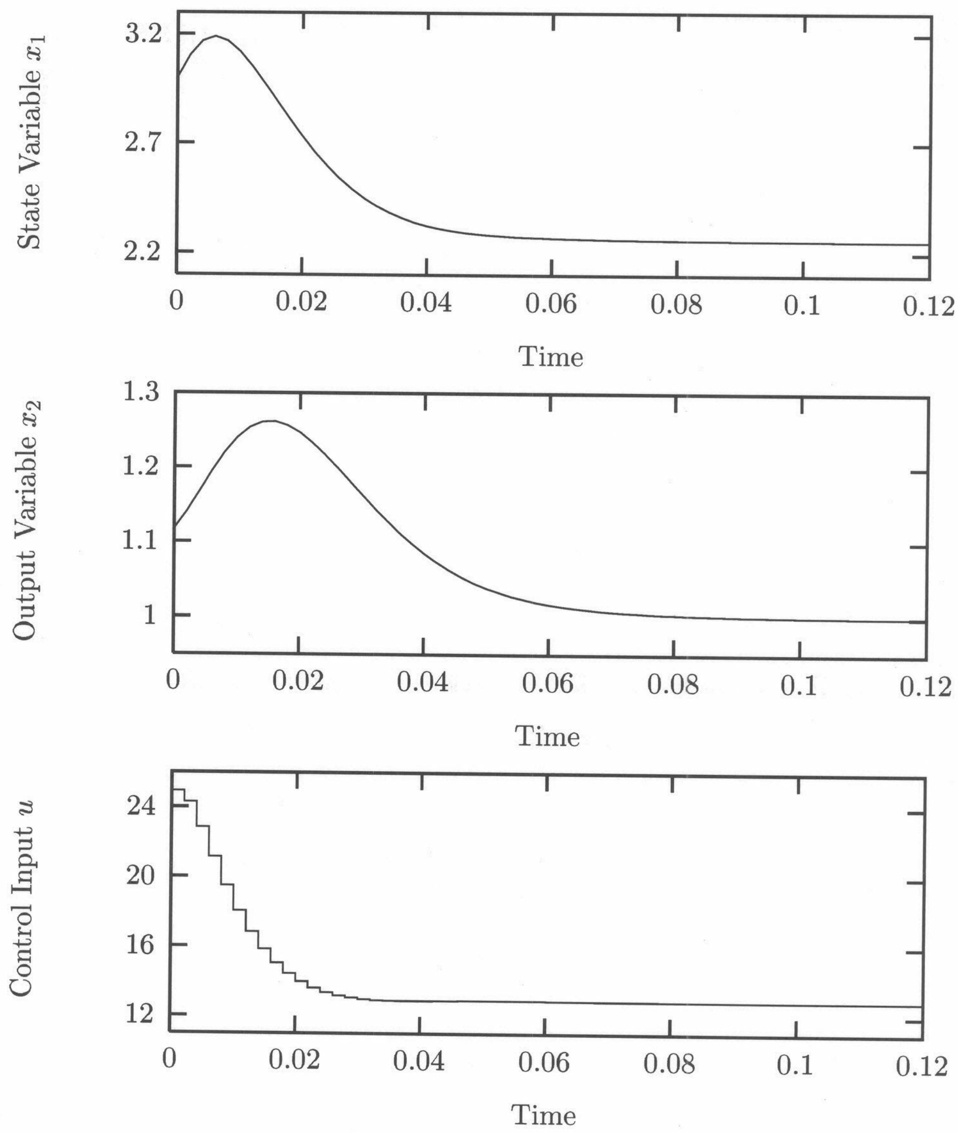

Figure 5.14: Isothermal CSTR: System response using target optimization.

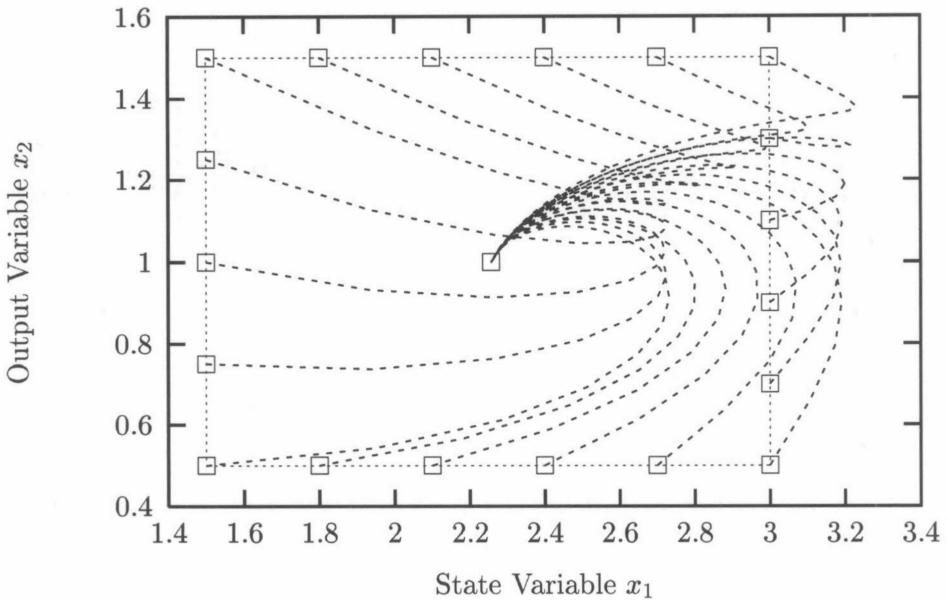

Figure 5.15: Isothermal CSTR: System response from various initial conditions.

the state-space shown in the diagram. The initial conditions and the final value of the controlled system are indicated with boxes. Since the controller is implemented at discrete points in time, the system trajectories can cross, unlike the situation with continuous-time feedback systems in two dimensions. This figure shows that the region of attraction for the MPC controller in the presence of model error is not negligible and that the controller would be expected to perform well from a variety of initial conditions.

Using both the conventional MPC feedback and feedback through target optimization, we compared the ability of the MPC controller to reject zero-mean Gaussian disturbances in the output. As in Figure 5.13, we used discrete, additive, zero-mean Gaussian noise in the output of the process with variance 0.09. To allow direct comparison of various methods, each simulation used the same noise sequence, which we obtained by re-seeding the random number generator with the same seed value.

Figure 5.16 shows the response of the two feedback methods. The target optimization feedback method with horizon length 3 is compared to the conventional DMC controller using horizon length 7. The target optimization feedback is insensitive to the horizon length and provides virtually the same response for horizon lengths from 1 to 5. During the course of the simulations, about 15 percent of the initial nonlinear programming problems of Equation 5.27 had no feasible point (which *never* occurred without the random disturbances). In these cases, the NLP of Equation 5.28 was used to determine the reference values. Figure 5.16 shows that feedback through target optimization with no weighting on Δu is more sensitive to disturbances than the conventional MPC feedback method with the weighting on Δu. This effect clearly shows up in the input. To attenuate the effect of the zero-mean disturbances, one can choose to weight Δu in the controller optimization, but there are other choices as well. For example, we can incorporate a moving average filter to estimate the disturbance term: $\hat{d}_k = 0.9\,\hat{d}_{k-1} + 0.1\,(y_k - g\,(\hat{x}_k))$. This reduces the occurrence of infeasibility in the initial NLP to about 5 percent. The effect of this filter is illustrated in Figure 5.17. As in the previous figure, the conventional MPC feedback controller with horizon length 7 is included for reference. From Figure 5.17, we can see that feedback through target optimization combined with the filter for the disturbance term has greatly improved the disturbance rejection ability of the controller, while retaining the ability of the feedback method to provide robustness and fast response.

Figure 5.17 shows that an MPC controller with a short horizon using target optimization feedback can perform as well as or better than conventional MPC feedback with longer horizons in terms of robustness, speed of response and disturbance rejection; however, the feedback through the target optimization solution is *much* faster to implement. Using horizon length 9 and conventional MPC feedback, the simulations took about 20 hours to simulate 120 time steps on an IBM-compatible 80486/33 MHz computer. At each sampling time, this method required the solution of a nonlinear programming problem involving 99 decision variables. In contrast, using feedback through target optimization, the initial NLP needed to compute the targets involves 3 decision variables and requires an insignificant amount of time to compute. The control computation itself involved as few as 6 decision variables (for $N = 1$) to 18 variable (for $N = 3$) and could easily have been computed in real time using a sampling interval of 7.2 seconds.

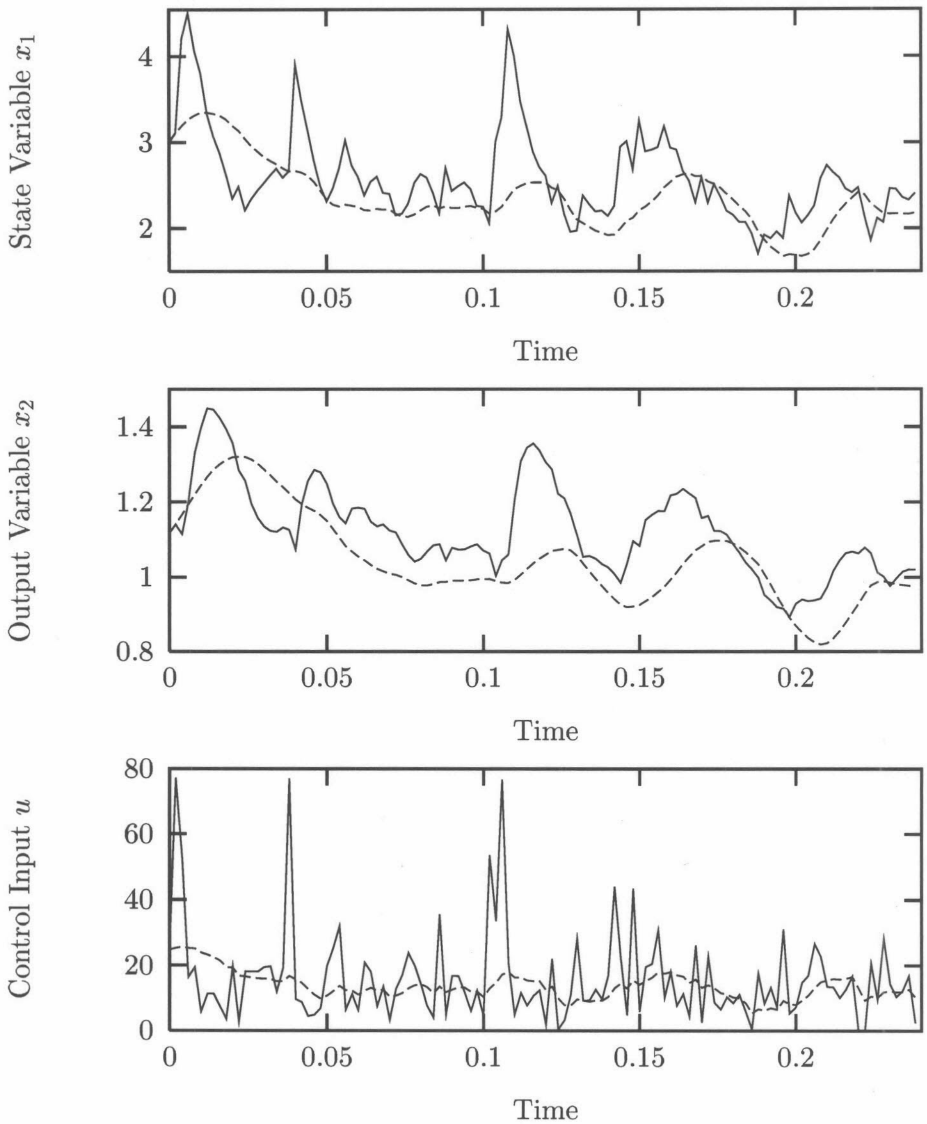

Figure 5.16: Isothermal CSTR: Effect of output disturbances using feedback through target optimization. Solid line: feedback through target optimization. Dashed line: conventional MPC feedback.

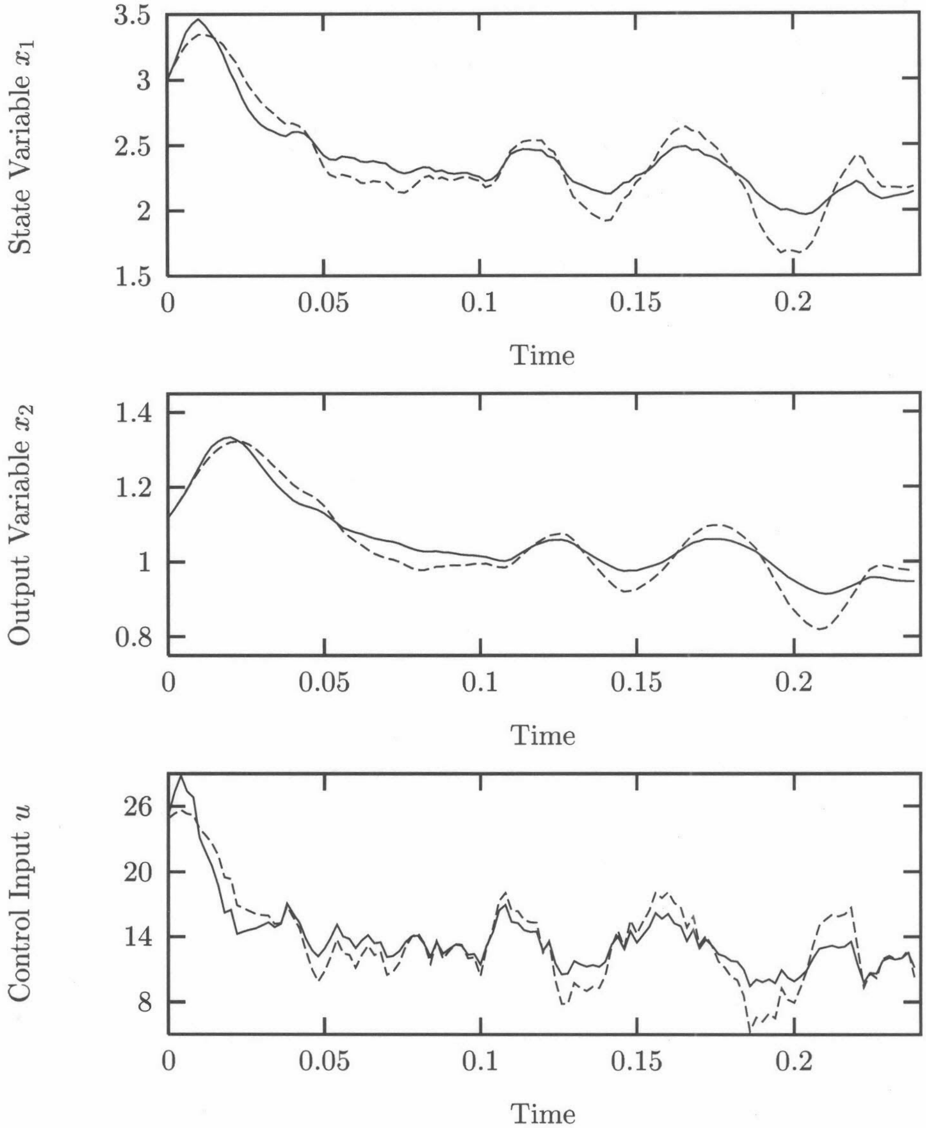

Figure 5.17: Isothermal CSTR: Effect of output disturbances using feedback through target optimization. Solid line: feedback through target optimization. Dashed line: conventional MPC feedback.

5.6 Selection of Tuning Parameters

As illustrated by some of the previous examples, the choices of control horizon, weight matrices and feedback methods have a profound effect on MPC nominal stability and robustness. Ideally, we would prefer to know *a priori* what range of tuning parameters would provide nominal stability and robustness. We would then be able to select the tuning parameters to meet specific process objectives, without being concerned that a poor choice could drive the system to instability. Implementable sufficient conditions that guarantee nominal stability [55] and robustness [1] of MPC are known for linear systems. Unfortunately, the known sufficient conditions for nonlinear systems, such as those described in Section 5.7 (infinite horizon, zero final-state constraint), are usually much too strong to offer practical implementation guidance, and we must resort to a set of heuristics based on the extrapolation of linear systems results, numerical simulations and experiments.

The most significant tuning parameters that must be selected are prediction horizon, control horizon, sampling interval, and penalty weight matrices. In addition, the control engineer must decide whether to penalize Δu or whether to use a filter in the feedback loop. As part of the numerical implementation, the discretization scheme and software must be selected. Some guidelines for selection of tuning parameters are provided below. In the following discussion, we assume that an explicit model is available and that the model has been acceptably verified.

- Sampling Interval: For stable, minimum phase systems, stability does not depend on the sampling interval; however, to ensure good closed-loop performance, the sampling interval should be small enough to capture adequately the dynamics of the process, yet large enough to permit the on-line computations necessary for implementation. Large sampling interval can result in ringing (excessive oscillations) between sample points. An example of this phenomenon is provided by García and Morari [23] using a linear system.

  For unstable systems, robustness depends critically on the sampling interval chosen. Recall that the choice of sampling interval for the fluidized-bed reactor problem of Section 5.2 was based on the doubling time of the unstable mode of the nominal system. The nominal system showed stable behavior under MPC, yet the MPC controller is unstable with model mismatch. We applied a +20 percent error in the pre-exponential factor of the first reaction. Simulation results are presented in Figure 5.18. Using the original sampling interval of 1.09, we see that the reactor experiences a temperature runaway near the end of the third sampling period. This constitutes a catastrophic failure. (Although our integrator bravely attempted to integrate the model past the runaway, we have little confidence in either the model or the integrator beyond this point. Results shown after the runaway must be viewed skeptically.)

After receiving these simulation results, we cut the sampling interval in half to 0.545 and observed similar unstable behavior, with the onset of the runaway occurring slightly later. Halving the sampling interval again to 0.2725 provided the second curve indicated in Figure 5.18, which demonstrates good control without any tendency toward runaway.

For unstable systems, there exists an inverse relationship between the model error and the maximum allowable size of the sampling interval. In the example of Figure 5.18, with the true pre-exponential factor α_1 20 percent higher than the model, we observe a temperature runaway between sampling periods. Since feedback is only incorporated at the sampling points, the sampling interval must be chosen sufficiently small to detect any unexpected increase in temperature before it becomes a runaway situation. This is a difficult design issue that has not been fully explored for nonlinear systems. For stable linear SISO systems, Zafiriou and Morari [79] have provided some criteria for sampling interval selection that later appeared in their book [53]. The interested reader is referred to these works as well as to standard texts on digital control and digital signal processing for further discussion of sampling intervals for linear systems.

- Prediction Horizon: For linear systems, several researchers have provided selection criteria for the prediction horizon that assure stability [23, 62, 67]. The question has not been resolved for nonlinear systems, but we can use the simulation results of Section 5.2 to illustrate some general principles.

  With conventional MPC feedback, Figure 5.12 shows that longer horizons tend to produce more aggressive control action, more overshoot, and faster response. From Figure 5.13, we see also that longer prediction horizons are more sensitive to disturbances, although this effect can be partially mitigated by including a filter in the feedback loop.

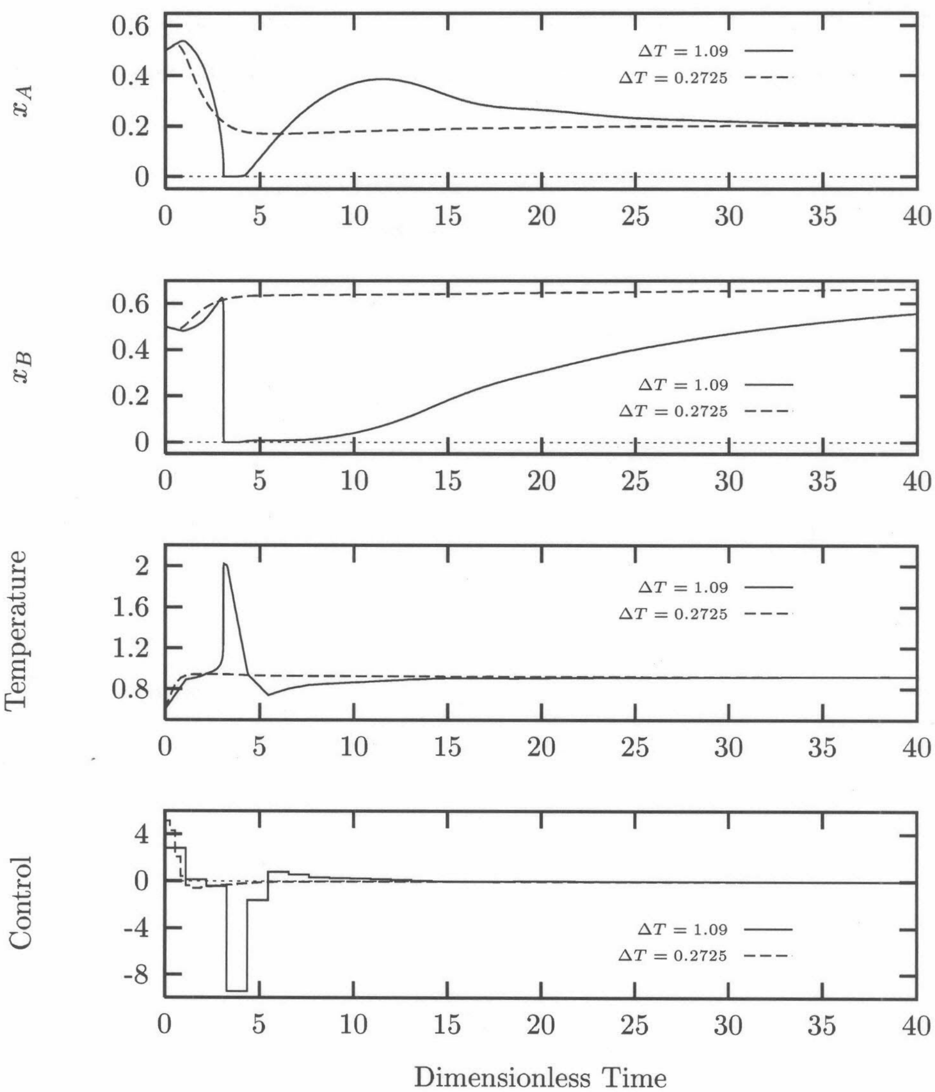

Figure 5.18: Fluidized-bed reactor: Effect of sampling interval.

Nominal stability is strongly affected by the horizon length, as illustrated by Figure 5.5, in which we observed a critical horizon length. Prediction horizons shorter than the critical value produced an unstable closed-loop system.

Based on linear systems results [23] and our simulations, the effect of increasing prediction horizon diminishes as N becomes large, and the advantages of longer horizons are outweighed by the increase in computations required.

With feedback through the nonlinear programming technique, we observed that prediction horizon had almost no effect. Whether this applies more generally is unknown and is topic of current research.

- Control Horizon: Linear systems results [23] indicate that shortening the control horizon relative to the prediction horizon tends to produce less aggressive controllers, slower system response and less sensitivity to disturbances. The effect is very similar to that of increasing the penalty on control action in the MPC objective function. We investigated this effect using the fluidized-bed reactor model of Section 5.2. For the nominal system, Figure 5.7 showed the effects of using prediction horizon $N = 4$ and control horizons $M = 1$ and $M = 2$. Comparing Figure 5.7 with Figure 5.6, we observe that the time scale of the system response is much longer for $M = 1$ or $M = 2$ compared to $M = 4$ with less aggressive control action. Unfortunately, in the case with model mismatch, this is exactly what is *not* needed, as indicated in Figure 5.19. Slowing the control action permits the reactor temperature to run away for $M = 1$. For $M = 2$, the control action is quick enough to prevent the runaway.

- Penalty Weights: For choice of weighting matrices, Bryson and Ho [6] suggest applying weights that are inversely related to the maximum acceptable range of the variables being penalized. We have generally found that this technique rarely provides good results because it over-penalizes control action. To provide good performance by the optimization algorithm, we recommend applying a penalty to setpoint deviations that places the objective in the interval 1-100 for the range of expected conditions, and then applying small penalties (less than 10 percent of the output penalty) on control or control increments, to achieve good closed-loop performance.

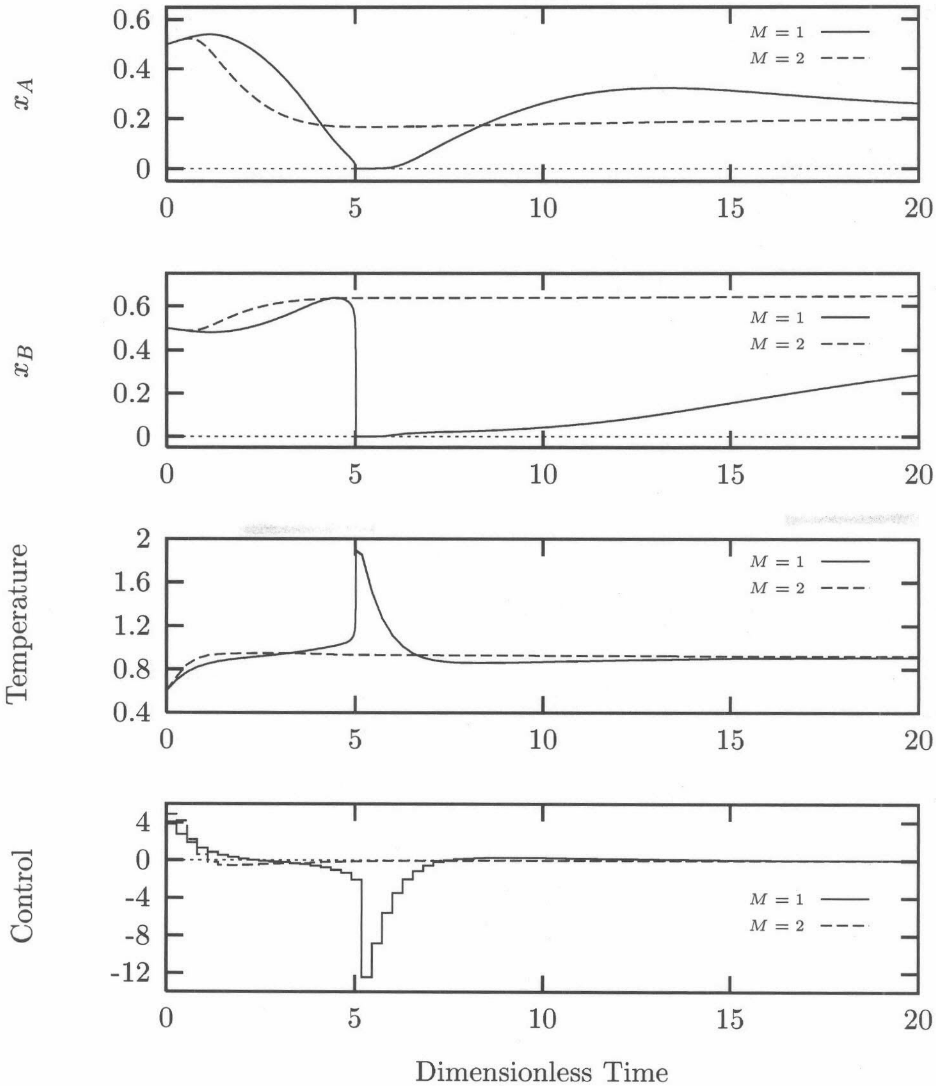

Figure 5.19: Fluidized-bed reactor: Effect of control horizon.

In the simulations results presented in this chapter, we typically applied a penalty weight of 100 on the integral square deviation of the output from the setpoint. Relative to the penalty on setpoint deviations, in our simulation studies we applied a 0.1 to 1 percent penalty to the control or control increments. Naturally, these weight must be adjusted if the terms which contribute to the objective have widely differing magnitudes arising from dimensional differences. See Gill, *et al.* [27] for a discussion of scaling issues in optimization.

- Collocation approximation: For the simulations using the isothermal CSTR model, we used 3 internal collocation points in the interior of each element, as well as points on the element boundaries. For the fixed-bed reactor, we generally used two interior collocation points to reduce the computation time. In one case, we repeated the fixed-bed reactor simulation in the presence of model mismatch using both 1 and 4 interior collocation points. The higher order approximation gave virtually identical results as the case with 2 interior collocation points. The lower order approximation resulted in a reactor runaway.

  Clearly, there are relationships between the size of the sampling interval, model error, the degree of the interpolating polynomial, and the performance and stability of MPC using orthogonal collocation. These relationships have not been quantified, so we can provide only a set of heuristic guidance based on our own experience. We suggest at least two interior collocation points be used, and no more than about 5. Unless the model predicts rapidly varying solutions within one finite element, a smaller number is desirable to reduce computation time.

- Filter in feedback loop: In Section 5.5, we showed that filtering the feedback signal when using steady-state target optimization provided good disturbance rejection and fast system response. The choice and effect of the disturbance filter is strongly system dependent. In the example of Section 5.5, we (naively) chose a very simple moving average filter with relatively slow response and observed excellent results. Applying the same procedure to the unstable fixed-bed reactor problem, the filter can produce catastrophic failure, depending on the speed of the linear filter chosen.

  As in Section 5.5, we used the filter $\hat{d}_k = \alpha_f \hat{d}_{k-1} + (1-\alpha_f)\left(y_k - g\left(\hat{x}_k\right)\right)$ in which $\alpha_f = 0.9$. The same filter applied to the fixed-bed reactor problem produced a runway in simulations that included a zero-mean

Gaussian disturbance in the output. Figure 5.20 shows the effect of the disturbance, both with and without the filter. The variance of the Gaussian disturbance is 0.01. As before, we do not place any confidence in the post-runaway simulation results; however, the computed temperature following the runaway illustrates that the algorithm with the filter slows and smooths the computed control input. We previously observed that increasing the control penalty or decreasing the control horizon relative to the prediction horizon also slowed control action and also resulted in reactor runaway.

We attempted to avoid the runaway situation by using a faster filter. Using $\alpha_f = 0.5$ produced the simulation results shown in Figure 5.21. We have succeeded in avoiding the runaway, the filter is effective in reducing noise sensitivity in the control signal, and slightly less effective in reducing variation in the temperature, which is the controlled output. If we consider data in the final portion of the time interval of Figure 5.21, the controller is essentially in a regulator mode to maintain the setpoint in the presence of disturbances. A rough measure of the effectiveness of the filter is the ratio of standard deviation with the filter to the standard deviation without the filter. After subtracting linear trends, the filter with $\alpha_f = 0.5$ produced standard deviation ratios of 0.538, 0.720 and 0.577 for the control, temperature and feedback ($\hat{d}_k$) signals, respectively. (For comparison, the same filter used to attenuate a Gaussian noise signal alone produces a ratio of 0.578.)

These results indicate that a filter in the feedback loop can be effective in reducing the controller sensitivity to output disturbances; however, like other methods that slow system response, the filter can induce unstable closed-loop behavior with unstable processes in the presence of uncertainty. Simulation studies are essential to detect such behavior before implementation.

5.7 Stability and Performance

5.7.1 Nominal Stability

The first property of a control system that should be satisfied is nominal stability, i.e., stability for systems free from modeling errors or disturbances. For linear systems without constraints, stability can be verified by checking the eigenvalues of the closed-loop system. For nonlinear systems, there

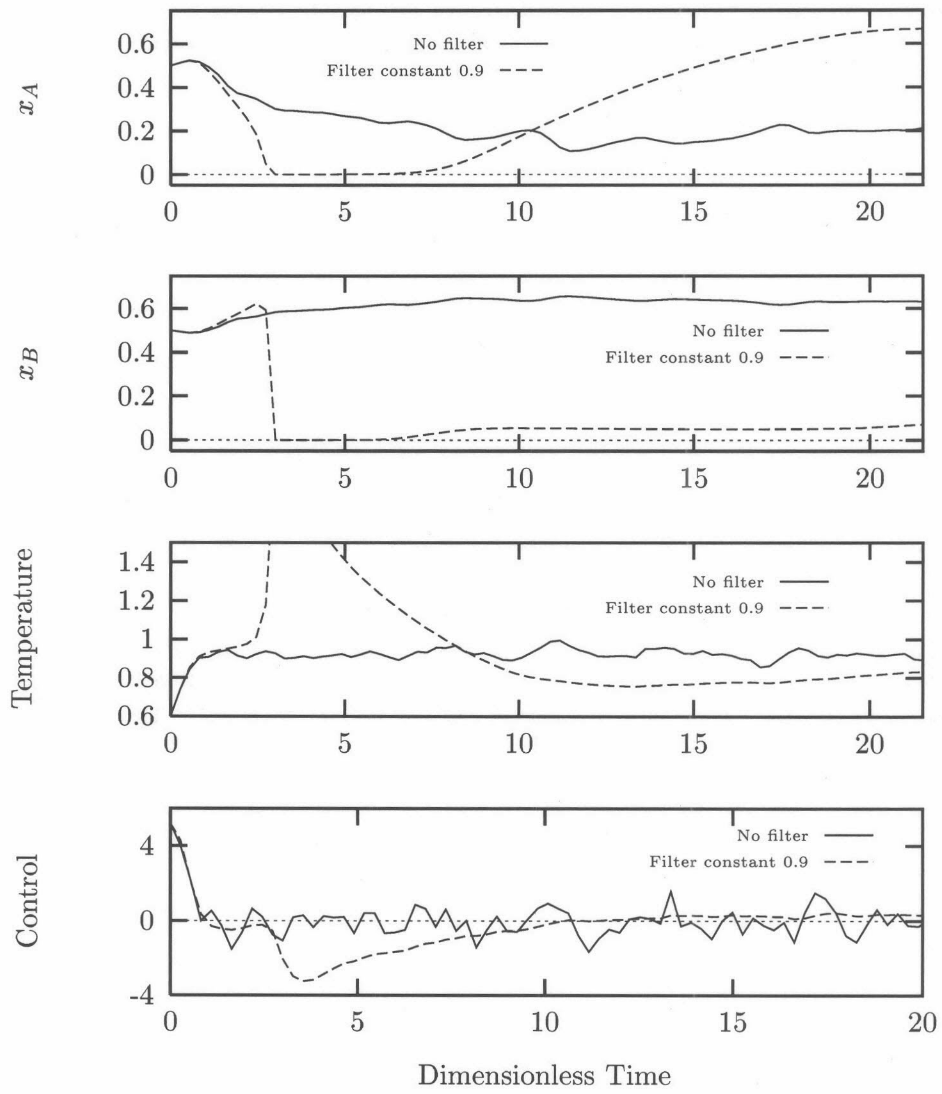

Figure 5.20: Fluidized-bed reactor: Disturbances and filtering effects.

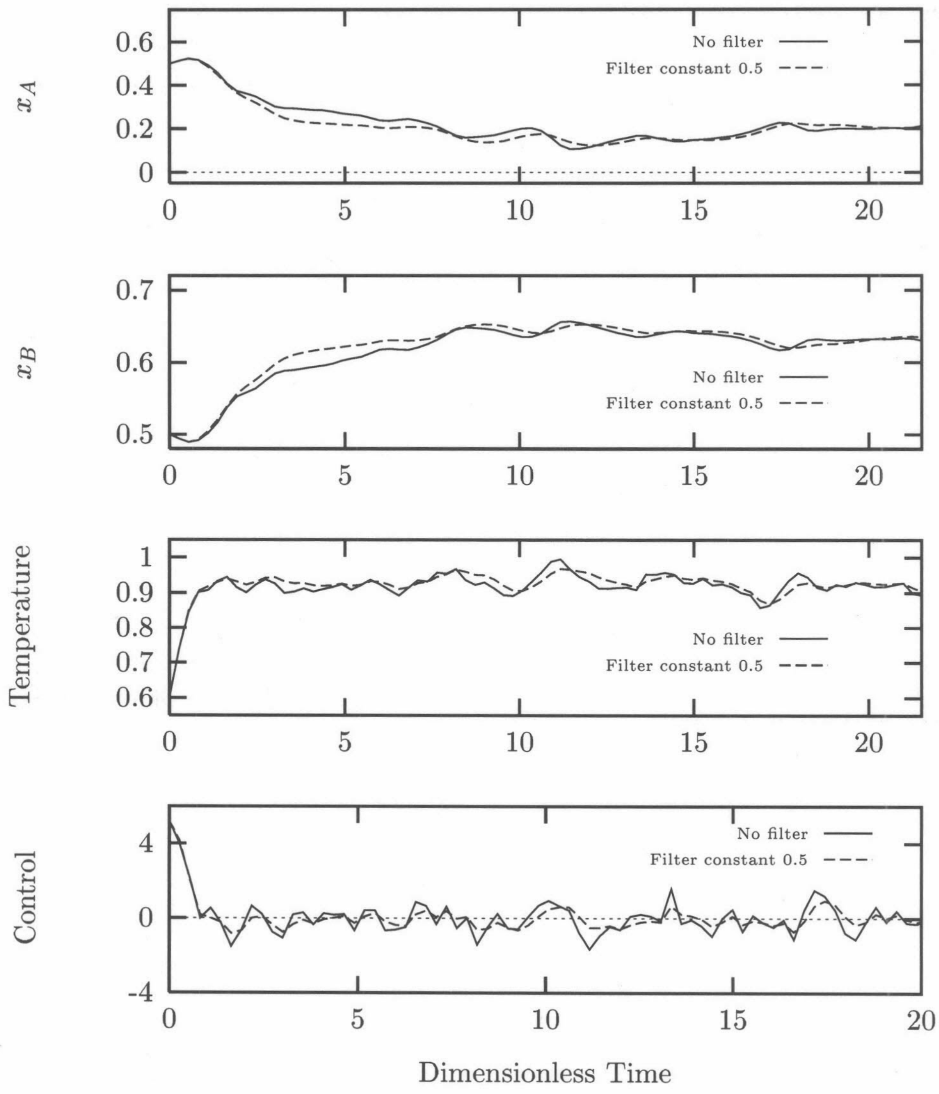

Figure 5.21: Fluidized-bed reactor: Disturbances and filtering effects.

is no simple equivalent criterion. Since closed-loop solutions to MPC control problems for nonlinear systems are almost never available, the Second Method of Lyapunov has been extensively used to analyze the stability of MPC. With sufficiently strong conditions on the nonlinear program that provides the MPC controller, we can view MPC as a method for generating Lyapunov functions for nonlinear systems.

Since Lyapunov analysis assumes such a prominent role, we repeat the basic theorem below. The following is adapted from Vidyasagar [76]:

Theorem 5 (Lyapunov Stability Theorem) *Let x_k be a sequence in $\Re^n$ indexed by $k \in \{0, 1, 2 \ldots\}$. The dynamic process x_k is asymptotically stable if there exist a function $V : \Re^n \rightarrow \Re_+$ and three functions α, β and γ of class $\mathcal{K}$ such that*

$$\alpha(\|x_k\|) \leq V(x_k) \leq \beta(\|x_k\|) \tag{5.29}$$

$$\Delta V(x_k) \triangleq V(x_{k+1}) - V(x_k) \leq -\gamma(\|x_k\|) \tag{5.30}$$

Class $\mathcal{K}$ functions are a specific class of positive definite functions. Using the definition of Vidyasagar [76], a class $\mathcal{K}$ function $\alpha : \Re_+ \rightarrow \Re_+$ is continuous and strictly increasing and satisfies $\alpha(0) = 0$. If α, β and γ have the properties of class $\mathcal{K}$ functions only in a neighborhood the origin, then the Lyapunov stability theorem provides only a local result.

Model predictive control provides an objective function that satisfies the sufficient conditions of Theorem 5 if we include the final state constraint $x_{k+N|k} = 0$. For simplicity, we will use a quadratic stage cost $\|x_{k+j|k}\|_Q^2 + \|u_{k+j|k}\|_R^2$ with Q and R positive definite. Without loss of generality, we can assume that $f(0, 0) = 0$ through a simple change of coordinates. Using the MPC objective function

$$J^*(x_k) = \min_{u_{k+j|k}} \sum_{j=0}^{N-1} \left(\|x_{k+j|k}\|_Q^2 + \|u_{k+j|k}\|_R^2 \right) + \|x_{k+N|k}\|_Q^2$$

we see that $J^*(x_k) \geq \|x_k\|_Q^2$ and the quadratic function

$$\alpha(\|x_k\|) = \sigma_{\min}(Q) \, \|x_k\|^2$$

(in which $\sigma_{\min}(Q)$ is the smallest singular value of Q), satisfies the lower bound of Equation 5.29. Note that x_k and u_k are equivalent to $x_{k|k}$ and $u_{k|k}$, respectively.

Equation 5.30 requires a bound for the difference between values of the Lyapunov Function for subsequent states. Using the MPC objective function as our candidate Lyapunov function, Equation 5.30 is also satisfied by including the final state constraint $x_{k+N|k} = 0$. If we represent a sequence of controls that corresponds to the optimal $J^*(x_k)$ by

$$\left\{u^*_{k|k}, u^*_{k+1|k}, \dots u^*_{k+N-1|k}\right\}$$

and let J' be defined by

$$\begin{aligned} J' &= J^*(x_k) - (\|x_k\|^2_Q + \|u^*_k\|^2_R) \qquad (5.31)\\ &= \sum_{j=1}^{N-1} \left(\|x_{k+j|k}\|^2_Q + \|u^*_{k+j|k}\|^2_R\right) \end{aligned}$$

The value of J' is simply $J^*(x_k)$ without the contribution of the first stage. In this case there is also no contribution from the final state to the summation since it is constrained to be zero.

After the control $u^*_{k|k}$ is implemented, the state moves to $x_{k+1|k} = f\left(x_k, u^*_{k|k}\right)$. Since we are considering the nominal system, the state moves exactly as predicted. At the new time $k+1$, the optimal control moves computed at time k still constitute a feasible sequence to which we can append a zero control without affecting the value of J'. This feasible sequence $\left\{u^*_{k+1|k}, \dots u^*_{k+N-1|k}, 0\right\}$ may or may not be optimal at time $k+1$, but it provides an upper bound for the optimal objective $J^*(x_{k+1})$ so that we have $J' \geq J^*(x_{k+1})$. Combining this inequality with Equation 5.31, we obtain

$$\begin{aligned} J^*(x_k) - J^*(x_{k+1}) &\geq \|x_k\|^2_Q \qquad (5.32)\\ &\geq \sigma_{\min}(Q)\,\|x_k\|^2 \qquad (5.33) \end{aligned}$$

which satisfies the inequality of Equation 5.30 of Theorem 5 after a change in sign. Notice that it was not necessary to consider the effects of constraints. For the nominal system, the existence of a solution to the nonlinear program at some initial time is sufficient for the existence of subsequent feasible solutions. Figure 5.22 provides an intuitive basis for the stability arguments above. After the control at time k is selected and implemented, a new problem is defined over the subsequent "window" as shown in the figure. The new objective includes state and control contributions on the interval $[k+1, k+N+1]$. The value of the optimal objective at $k+1$ must be less

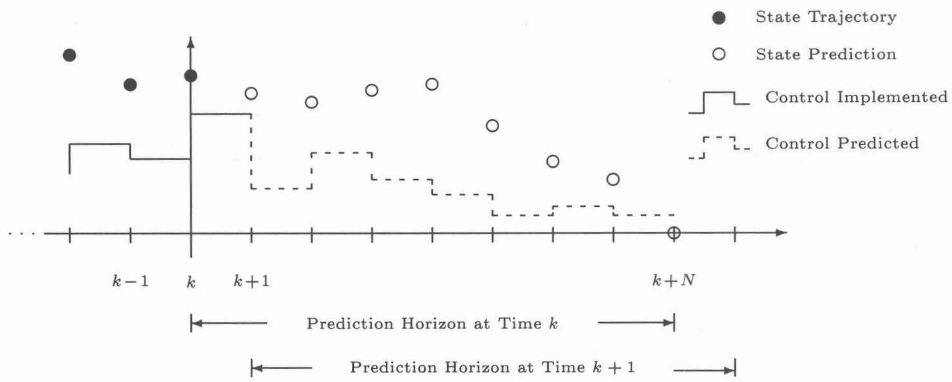

Figure 5.22: Model predictive control concept.

than that at k, because of the final state constraint. The decrease is lower bounded by $\|x_k\|_Q^2$, since the contribution of the state at k is no longer included.

To apply the Lyapunov Stability Theorem to MPC, it remains to be shown that there exist an upper bound $\beta(\|x\|)$ on the MPC objective. The upper bound is closely related to the null-controllability of the system. In general, *a priori* evaluation of null controllability is difficult for nonlinear systems; the case of constrained nonlinear systems is even more difficult. Suppose, however, that we knew that the origin could be reached using feasible controls from any state in some specific subset $\mathcal{S}$ of $\Re^n$ and the state trajectory under this sequence of controls remained in the feasible set. Then the MPC objective function could then be evaluated using the feasible control sequence and would provide an upper bound on the optimal MPC objective. The argument proceeds as follows: Let x_k be some state in $\mathcal{S}$ such that the control sequence $\left\{u_{k|k}, u_{k+1|k}, \ldots u_{k+N-1|k}\right\}$ produces a final state $x_{k+N|k} = 0$. We can construct a (non-optimal) MPC objective

$$J(x_k) = \sum_{j=0}^{N-1} \|x_{k+j|k}\|_Q^2 + \|u_{k+j|k}\|_R^2 \tag{5.34}$$

Again, there is no contribution of the final state because of the zero end-state. If $J(x_k)$ is continuous, then it is a class $\mathcal{K}$ function, since it is lower bounded by $\|x_k\|_Q^2$ which is a class $\mathcal{K}$ function. The arguments above may be formalized in the following theorem:

Theorem 6 (Stability of MPC) *For the nominal system*

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) \\ f(0,0) &= 0 \end{aligned}$$

if the MPC objective function is continuous, then the origin is an asymptotically stable fixed point under MPC with a region of attraction that consists of the subset of $\Re^n$ that is constrained null controllable.

A proof of the theorem follows from the previous discussion.

It is possible to generalize the previous discussion by substituting a general class $\mathcal{K}$ stage cost for the quadratic functions used above. In this case, asymptotic stability is retained. It is also possible to arrive at a similar stability result using an infinite horizon in which the final state constraint may be viewed as a finite parameterization of the more general infinite horizon problem. Details are provided by Meadows *et al.* [49]. Similar results were also reported by Keerthi and Gilbert, who considered both the infinite horizon and the final state constraint [34]. Stronger results are available. If the stage cost is quadratic, as discussed above, then MPC produces exponential stability. The result follows from the fact that the bounding class $\mathcal{K}$ functions α, β and γ are quadratic [68]. See also Vidyasagar [76] for a continuous time proof.

We have not directly addressed the question of constrained null controllability. As a practical matter, null controllability is usually verified in the process of solving the nonlinear program for the MPC controller. If no feasible point can be found, then for practical purposes, the system is not null-controllable. A numerical algorithm is not equivalent to a mathematical proof, but as a practical matter, if optimization software cannot find a feasible point for some initial state of interest, the system is a poor candidate for control with MPC. This is an aesthetically unsatisfactory approach to the verification of nonlinear controllability; however, it must be noted that the same problem exists with respect to linear MPC controllers subject to constraints, and has not prevented their widespread implementation.

The topic of nonlinear constrained controllability is an active research area which has not yet produced a complete theory. For further discussion of nonlinear controllability concepts see the following: [28, 29, 58, 70, 71, 76].

5.7.2 Steady-State Performance

In addition to nominal stability, an evaluation of a control method should consider robustness to model error. In this section we state some steady-state results that hold for MPC controllers applied to **arbitrary plants** rather than the nominal plant. The price we pay for the generality of considering arbitrary plants is that the results are in the form of **alternative conditions**, i.e. **either** the closed loop does not approach a steady state **or** the plant goes to a steady state with certain guaranteed properties, such as zero offset. The value of these results is that they characterize the steady state that the closed loop exhibits for any plant, in terms of the nominal model only. The steady-state properties can therefore be examined without knowledge of the plant.

We are concerned in this section with the target optimization feedback form of MPC. Some of the methods we use subsequently can also be applied to develop analogous results for the conventional MPC feedback scheme, but we do not investigate that here. Our opinion is that the target optimization provides a more general framework to handle a larger variety of process models than can be addressed with the conventional MPC feedback, so we focus our attention there. First we summarize the three parts of the controller discussed in Section 5.5.2

State estimator. The state estimator takes the current control and plant measurement and updates the state and disturbance estimates via

$$\begin{aligned} \hat{x}_{k+1} &= f(\hat{x}_k, u_k) \\ \hat{d}_{k+1} &= y_k - g(\hat{x}_k) \end{aligned}$$

Steady-state target optimization. The steady-state target optimization takes $\hat{d}_k$ and y_{ref} and produces as output x_t, u_t through solution of

$$\min_{u_t, x_t} \|u_{\text{ref}} - u_t\|_R^2 \tag{5.35}$$

subject to:

$$\begin{aligned} x_t &= f(x_t, u_t) \\ y_{\text{ref}} &= g(x_t) + \hat{d}_k \\ x_t &\in \mathcal{X} \\ u_t &\in \mathcal{U} \end{aligned}$$

Notice that we consider explicitly any controller input and state (output) constraints in the target optimization as well. If this optimization problem does not have a feasible point, the following nonlinear program is used to compute the targets

$$\min_{u_t, x_t} \|y_{\mathrm{ref}} - (g(x_t) + \hat{d}_k)\|_Q^2 \tag{5.36}$$

subject to:

$$\begin{aligned} x_t &= f(x_t, u_t) \\ x_t &\in \mathcal{X} \\ u_t &\in \mathcal{U} \end{aligned}$$

We always assume that this optimization problem has a feasible point, i.e. the model has at least one steady-state solution satisfying the constraints.

MPC control law. Finally, the MPC controller takes x_t, u_t and $\hat{x}_k$ and produces the control input u_k from the solution of

$$\min_{u_{k+j|k}} \sum_{j=0}^{N-1} \|x_{k+j|k} - x_t\|_Q^2 + \|u_{k+j|k} - u_t\|_R^2 \tag{5.37}$$

subject to:

$$\begin{aligned} x_{k|k} &= \hat{x}_k \\ x_{k+N|k} &= x_t \\ x_{k+j|k} &\in \mathcal{X} \\ u_{k+j|k} &\in \mathcal{U} \end{aligned}$$

The first optimal input is then applied to the plant, $u_k = u^*_{k|k}$.

The first result states that if the closed-loop system goes to steady state, then the steady-state plant output and input satisfy the target optimization problem.

Theorem 7 (Steady-state properties) *If*

1. *The steady-state model, $x = f(x, u)$, has a solution, $x \in \mathcal{X}$, $u \in \mathcal{U}$.*

2. *The MPC control law is nominally asymptotically stabilizing.*

3. The closed-loop goes to a steady state, $y_k \to y_s$, $u_k \to u_s$, $x_t \to x_{ts}$, $u_t \to u_{ts}$.

Then either $\|u_{\text{ref}} - u_s\|_R$ is minimized and $y_s = y_{\text{ref}}$ if Equation 5.35 has a feasible solution at steady state or $\|y_{\text{ref}} - y_s\|_Q$ is minimized.

Proof: At steady state (Assumption 3), the estimator and target optimization yield

$$
\begin{aligned}
\hat{x}_s &= f(\hat{x}_s, u_s) \\
\hat{d}_s &= y_s - g(\hat{x}_s) \\
x_{ts} &= f(x_{ts}, u_{ts}) \\
x_{ts} &\in \mathcal{X} \\
u_{ts} &\in \mathcal{U}
\end{aligned}
$$

The target optimization is well defined because of Assumption 1. We next show that because the controller is nominally asymptotically stabilizing (Assumption 2), $\hat{x}_s = x_{ts}$, i.e. the state estimate converges to the steady-state target. Consider the control problem with steady-state input and state targets (Assumption 3). The state estimate evolves according to

$$\hat{x}_{k+1} = f(\hat{x}_k, u_s)$$

Because the controller is nominally asymptotically stabilizing and the state estimator evolves according to the nominal model, the stage cost in Equation 5.37 goes to zero, which implies $\hat{x}_s = x_{ts}$ and $u_s = u_{ts}$. From the disturbance estimate equation

$$y_s = g(x_{ts}) + \hat{d}_s$$

Therefore, if at the steady state Equation 5.35 is being solved then the steady input minimizes $\|u_{\text{ref}} - u_s\|_R$ and $y_s = y_{\text{ref}}$. If this target problem has no feasible solution at steady state, then Equation 5.36 applies and the steady plant output minimizes $\|y_{\text{ref}} - y_s\|_Q$.

Because of the constraints, plant-model mismatch, and plants with more setpoints than inputs, it may not be possible to satisfy the conditions necessary for zero steady-state offset. We can add further restrictions to provide conditions that do ensure zero steady-state offset.

Corollary 1 (Zero steady-state offset) *If the conditions of Theorem 7 are satisfied and there exists x_t, u_t satisfying*

$$\begin{aligned} x_t &= f(x_t, u_t) \\ y_{\text{ref}} &= g(x_t) + \hat{d}_k \\ x_t &\in \mathcal{X} \\ u_t &\in \mathcal{U} \end{aligned}$$

for every y_{ref} and $\hat{d}_k$, then there is zero steady-state offset, $y_s = y_{\text{ref}}$.

Proof: The added conditions of the corollary ensure that the target optimization in Equation 5.35 has a feasible point. Therefore, zero steady-state offset follows from Theorem 7.

5.8 Discussion

MPC has its roots in the development of optimal control theory in the late 1950's, in which the "state-space" approach was considered a modern alternative to the frequency-domain techniques that had dominated control theory and practice since the 1930's. Kalman's landmark paper [30] consolidated and extended much of the then current work in the field, and provided rigorous definitions for the now-common concepts of controllability and observability. Although Kalman reported work for linear systems, many of the key concepts may be extended to nonlinear systems.

It was recognized early that optimality in the linear-quadratic problem is not sufficient for stability. Kalman [30] provided sufficient conditions for stability for this case. Several other workers considered formulations that could provide a stabilizing LQ optimal controller. One of these was Kleinman [36], whose "easy way" to stabilize a linear system was equivalent to a finite-horizon LQ optimal control that is subject to the final state constraint $x(T) = 0$. Kleinman followed up his continuous-time result with a discrete-time analog [37] in which he provided a criterion for selection of an initial condition for the matrix Riccati Difference Equation (RDE) that arises from the LQ optimal control problem in discrete time.

Thomas and coworkers[72, 73, 74] recognized that the zero final state constraint could be incorporated into the LQ optimal regulator by inverting the Riccati Differential (or Difference) Equation and integrating (or iterating) the result from an initial condition of zero. Conceptually, this is equivalent to an infinite final state weighing matrix in a finite-horizon LQ optimal control problem. Kwon and Pearson provided a firm theoretical basis for using the zero final-state constraint to provide stability for time-varying and invariant linear systems, including results for continuous time [42] and discrete time systems [43]. To implement MPC in continuous time requires that a solution of the optimization be available continuously. For linear systems, this means that the solution of matrix Riccati differential equation must be available. Kwon *et. al* [41] investigated the time-varying case and showed that a stabilizing solution to the matrix Riccati difference equation could be updated based on the system equations, without the need for a new solution at each time. Although not presented, the authors pointed out that the results are also applicable to discrete-time systems.

By the late 1970's, a substantial body of research results had been developed that provided a solid theoretical foundation for implementation of model predictive control. At about the same time, the price instability of feedstocks and energy in the process industries demanded higher efficiency from existing plants. Rather than control processes in midrange of operations where traditional linear controllers would suffice, it became necessary to operate closer to limits using models that could be generated quickly with a minimum of modeling effort. In 1978, Richalet et. al [63] presented IDCOM and in 1979, Cutler and Ramaker [13] presented Dynamic Matrix Control (DMC), both of which responded to these new demands. These approaches used either step or impulse response models in discrete time whose coefficients can be determined from plant step test data. Since processes are assumed to be time-invariant, the plant dynamics can be described by a matrix of constant coefficients, hence the acronym DMC. The DMC controller used the linear system represented by the dynamic matrix to predict future process output based on current and future control action. The control was selected that minimized a quadratic objective based on control and process variables predicted by the model. Constraints on the control variable were explicitly incorporated by the optimization algorithm in subsequent formulations [24].

In the chemical engineering literature, model predictive control came to be identified with the methods of DMC, including linear step response models, finite prediction horizons, and the use of the DMC disturbance model. DMC assumes that the difference between current processes measurements and the model prediction is due to an unmodeled disturbance that maintains the same value throughout the prediction horizon; to obtain a prediction, one must use that model equations and add to the predicted variables the value of the current disturbance estimate. The effect is extremely useful, since it eliminates steady-state offset for step changes in setpoint, as discussed in Section 5.5.

The basic DMC algorithm has undoubtedly been a phenomenal success by any measure; however, its origins as a somewhat *ad hoc* method necessarily left some questions unanswered. For example, with the introduction of output variable constraints, some stable processes can become closed-loop unstable. The multivariable nature of DMC allows for a large number of tuning variables in the objective function. The length of the prediction horizon is a tuning parameter. Often, the control horizon is different from the prediction horizon–yet another tuning parameter is available.

The influential paper by García and Morari [23] strengthened the trend toward identification of DMC and its descendants with MPC in general. For the unconstrained case, García and Morari provided some stability results and suggested tuning parameters for the DMC algorithm.

The step response model used in DMC admits only stable processes. Seeking to extend the applicability of the step response models and to incorporate the large body of existing results for state space models, Li and coworkers [46] and Lee *et al.* [44] produced MPC algorithms that converted the step response models into a state-space form. The new formulations were based on the step response models and incorporated the step response coefficients. Although Eaton and Rawlings [19] and Lee *et al.* extended the basic method to include integrating processes, formulations that depend on step response coefficients are unsuitable for general unstable processes.

In parallel, Clarke and coworkers opted for a transfer function model of the process and formulated the approach known as generalized predictive control (GPC) [9, 10, 11]. Part of the aim of this development was to be able to handle unstable plants.

Rawlings and Muske [54, 55, 62] departed from the step response models common in chemical engineering practice to use the standard linear state space model to formulate a stabilizing constrained MPC method. Their control formulation guaranteed nominal stability using any valid tuning parameters.

Inequality constraints in any MPC method lead to a non-linear feedback control law. A natural extension of existing results for linear systems involves MPC with explicitly nonlinear models. In the chemical engineering literature, Patwardhan and coworkers [60] and Eaton and Rawlings [19] propose a simultaneous optimization-model solution approach that provides a continuous-time MPC controller. Although no stability proofs were provided, numerical results indicated stability using chemical reactor example problems. Patwardhan, *et al.*, also demonstrate robustness to parameter error.

Chen and Shaw [8] provide an early discussion of model predictive control with the end point constraint for control of nonlinear systems. A rigorous and comprehensive approach to the nonlinear problem in discrete time was presented by Keerthi and Gilbert in a series of papers in the mid-1980's [32, 33, 34, 35]. They consider a general nonlinear, time-varying dynamic system with a general cost function defined on either a finite or infinite horizon. Finite horizon problems include the zero final state constraint. Sufficient conditions are provided that ensure asymptotic stability of the nonlinear system using an MPC control law. These include a modified controllability condition on the nonlinear system. Keerthi and Gilbert also provided sufficient conditions for convergence of finite horizon results to the infinite horizon case. Meadows and coworkers [49] present similar results for the time-invariant case and show that any feedback linearizable system can be stabilized using MPC. In the same work, they also show that there exist discrete-time systems that require discontinuous feedback laws for stability and that nonlinear MPC can generate such control laws.

The work of Keerthi and Gilbert and Meadows *et al.* provides sufficient conditions for stability of nonlinear MPC. The key difficulty in implementation arises from the final state constraint; it is difficult or even impossible to find the set of states for which the final state constraint is feasible *a priori*. As we argued previously, we can shift the burden of verifying feasibility to the optimization routines, but even so, the range of feasible states may be unacceptably small.

With these limitations in mind, Mayne and Michalska [47, 52] offer an approach that retains the stability property of MPC using the final state constraint, while expanding the set of initial feasible states. Rather than adopt the strict final state equality constraint $x_N = 0$, Mayne and Michalska propose using a "dual-mode" controller whose final state is constrained to lie within a region of the state space $W \subset \Re^n$ that contains the origin. Outside the region, a nonlinear MPC controller is used that includes the final state constraint $x_N \in W$. Within W, a stabilizing controller is used whose design is based on the system's linearization at the origin. The region W is defined as the largest elliptical region in the state space in which the linearized controller produces a decreasing (and therefore asymptotically stable) Lyapunov function. It can be determined off-line through the solution of a single initial nonlinear programming problem. Mayne and

Michalska presented detailed analyses of stability and robustness, with special emphasis on providing an implementable algorithm. A disadvantage of the Mayne-Michalska algorithm is that it requires a system model that has a stabilizable linearization at the origin, but this condition is usually satisfied in systems of engineering interest.

5.9 Future Directions

Nonlinear MPC offers a number of interesting challenges. Among these we can list a few:

- Robustness: We have seen that MPC used with the deadbeat disturbance estimator is robust to model structural and parameter uncertainty. To date, most results in this area indicate good performance in the presence of model uncertainty; however, a rigorous analysis that can provide sufficient conditions for robustness has not been completed.

- Estimator/Controller Interactions: Unlike the linear systems result, for general nonlinear systems there is no separation principle. Controller performance is intimately linked to estimator design. In the CSTR example, without state feedback it was difficult to achieve the desired steady state even though the setpoint was satisfied. Meadows *et al.* [50] and Scokaert and Rawlings [68] showed that with constrained linear systems, if the closed-loop dynamic system satisfies a Lipschitz condition, a weak form of separability can be demonstrated. More work is needed for the general nonlinear case.

- Nominal Stability: There continues to exist a gap between observed results and theory in nonlinear MPC. Experimental results, in terms of both simulations and applications, continue to indicate good performance of MPC on a wide class of problems, but tuning the algorithm to obtain stability is usually necessary. A theory to guide the tuning process is not yet available. Current theoretical results require infinite horizons or the final state constraint $x_N = 0$ to assure stability, conditions that can be much too strong (the final state constraint) or unachievable in practice (infinite horizon).

  As an alternative to these two approaches, it is possible to guarantee stability by using a final state penalty function rather than a constraint. Consider the case in which an infinite horizon is used. If this problem could be solved, it would provide a stabilizing controller. If we had an analytical expression for the optimal infinite horizon objective function $J_\infty^*(x)$, we could devise an equivalent finite horizon function that used $J_\infty^*(x)$ as a final state penalty function:

$$\Psi^*(x_k) = \min_{u_{k+j|k}} \sum_{j=1}^{N-1} \left(\|x_{k+j|k}\|_Q^2 + \|u_{k+j|k}\|_R^2 \right) + J_\infty^*(x_N)$$

Provided that the stage cost used to compute J_∞^* is the same as in Ψ^*, the principle of optimality guarantees that $\Psi^*(x) = J_\infty^*(x)$; therefore, MPC using the objective function $J_\infty^*(x_k)$ is stabilizing.

It is probably difficult or impossible to provide a closed-form expression for J_∞^*; however, if we are able to compute a bound on $J_\infty^*(x) \leq \tilde{J}^*(x)$ such that $\tilde{J}^*(0) = 0$, then the MPC controller computed using $\tilde{J}^*$ is also stabilizing, subject to some technical conditions. Some preliminary results in this area have already been presented [48] and an article containing more details concerning this approach is in preparation.

Acknowledgments

The authors wish to thank Dr. Pierre O. M. Scokaert for helpful discussions. The financial support of NSF, through grant CTS–9311420, and the industrial members of the Texas–Wisconsin Modeling and Control Consortium (TWMCC) is gratefully acknowledged. The first author was financially supported during the writing of this chapter from a fellowship from the Brazilian government, acting through the Conselho Nacional de Desenvolvimento Científico e Tecnológico (National Council of Scientific and Technological Development). This support is gratefully acknowledged.

5.10 Notation

A	matrix of first derivative collocation weights without initial condition
$\tilde{A}$	matrix of first derivative collocation weight in orthogonal collocation approximation with multiple finite elements
A_0	matrix of first derivative collocation weight, modified to include the initial condition equation

C	matrix used to form linear inequality constraints on Δu
c_k	vector of quadrature weights for integration in orthogonal collocation approximation
$\tilde{c}_k$	vector of quadrature weights for integration in orthogonal collocation approximation
$c_{\text{lb}}, c_{\text{ub}}$	upper and lower bounding vectors for linear inequality constraints on control increments
D, H	matrices used to express linear constraints on state and control variables
d, h	right-hand sides of linear inequality constraints on state and control variables
d_k	disturbance term used in conventional MPC feedback
$\hat{d}_k$	disturbance estimate obtained from state observer
F	right-hand side of matrix equation representing a differential equation discretized by orthogonal collocation approximation with a single finite element
$\tilde{F}$	right-hand side of matrix equation representing a differential equation discretized by orthogonal collocation approximation with multiple finite elements
$F_{x,k}, F_{u,k}$	Jacobian matrix of $f(x, u)$ with respect to x or u, evaluated at (x_k, u_k)
f	state transition function in dynamic system equation
G_k	Jacobian matrix of $g(x)$ evaluated at (x_k, u_k)
g	function relating state vector to output vector
h_x, h_d	correction terms for state update in state observer
J	MPC objective function
J_∞	infinite horizon optimal MPC objective function
J^*	optimal MPC objective function
$\tilde{J}^*_\infty$	upper bound on infinity horizon optimal MPC objective function
j	indexing variable for time within prediction horizon for discrete system

K	constant for proportional control in fluidized-bed reactor model
$\mathcal{K}$	set of functions used in Lyapunov stability theorem
k	independent variable representing time in discrete system
k_1, k_2	reaction rate constants in fluidized-bed reactor model
k_1, k_2, k_3	reaction rate constants in isothermal CSTR model
L	stage cost, penalty incurred at each time in prediction horizon as function of outputs, controls or control increments
$\mathrm{Le}_A, \mathrm{Le}_B$	Lewis number in fluidized-bed reactor model
M	MPC control horizon, when different from N
m	dimension of control vector
N	MPC prediction horizon
n	dimension of state vector
Q, R, S	weighting matrices for vector norms used in MPC objective function
q, q^{-1}	shift operators in discrete-time systems
$\mathcal{S}$	set or null-controllable initial states
T	temperature in fluidized-bed reactor model
T	prediction horizon in continuous-time MPC
T_F	feed temperature in fluidized-bed reactor model
T_{set}	setpoint temperature in fluidized-bed reactor model
ΔT	sampling interval
t	independent variable time in continuous-time systems
t_f	final time in prediction horizon for continuous-time MPC problems
t_0	initial time in prediction horizon for continuous-time MPC problems
u_k	control vector at time k
u_t	input target vector obtained from target optimization
u^*	solution of nonlinear program for feedback through steady-state target optimization

Δu_k	control increments, i.e., $u_k - u_{k-1}$
$\mathcal{U}$	subset of $\Re^m$ used as general control constraint
V	Lyapunov function
W	in the algorithm of Mayne and Michalska, the region near the origin in which a linear control law is stabilizing
w_k	random input variable in discrete-time stochastic model
X	matrix containing elements $x_{i,j}$
$\tilde{X}$	matrix of states $x_{i,j}$ in orthogonal collocation approximation with multiple finite elements
$\mathcal{X}$	subset of $\Re^n$ used as general state constraint
x_1, x_2	state variables in the isothermal CSTR model
x_A, x_B	concentrations in fluidized-bed reactor model
$x_{A,F}, x_{B,F}$	concentrations of species A and B in fluidized-bed reactor model
x_F	feed concentration of species A in isothermal CSTR model
$x_{i,j}$	in problems discretized with orthogonal collocation, the j-th element of the state vector at discretization point i
$x_{i,*}$	in problems discretized with orthogonal collocation, the entire state vector at discretization point i
x_{init}	initial condition for state in orthogonal collocation approximation
$\hat{x}_k$	state estimate obtained from state observer
x_k	state vector at time k
x_t	state target vector obtained from target optimization
y_k	output vector at time k
y_{ref}	reference trajectory in MPC objective
$\tilde{y}_k$	model output, used when necessary to distinguish model output from true process output

Greek Symbols

α_f	filter constant in experiments with fluidized-bed reactor

α_1, α_2	pre-exponential factors in fluidized-bed reactor model
α, β, γ	class $\mathcal{K}$ function used in Lyapunov Stability Theorem
β	control weight in MPC objective function
$\bar{\beta}$	residence time in fluidized-bed reactor model
β_1, β_2	heats of reaction in fluidized-bed reactor model
$\gamma_1, \gamma_2, \gamma_3$	weighting constants in MPC objective function of Equation 5.7
γ_1, γ_2	activation energies in fluidized-bed reactor model
μ	feedback control law resulting from MPC
ϕ	final state penalty function
Ψ^*	finite horizon objective function equivalent to an infinite horizon objective
$\sigma_{\min}$	smallest singular value
τ	independent variable time in MPC objective function of Equation 5.7

Subscripts

A, B	refers to species A and B in fluidized-bed reactor model
F	value in feed stream in example models
k	value at time k
$j\|k$	value at time j, given information up to and including time k
i, j	in combination, refers to j-th element of vector at collocation point i
$i, *$	in combination, refers to entire vector at collocation point i
init	initial condition in orthogonal collocation approximation
lb, ub	upper and lower bounding vector for linear constraints
max, min	used to indicate upper and lower limits in linear inequality constraints
min	used in conjunction with $\sigma_{\min}$ to indicate the smallest singular value
ref	reference values, often indicates setpoint

set	setpoint in fluidized-bed reactor model
x, d	in state observer, refers to terms which account for the state and disturbance estimate, respectively
$+$	only used in conjunction with $\Re_+$ to indicate the subset of real numbers $[0, \infty)$
∞	infinite horizon value

Operators and Special Symbols

$\|\cdot\|$	vector norm
$\|\cdot\|_Q$	Q-weighted vector norm
$\Re^n$	the n-th Cartesian product of the real numbers
$\Re_+$	the semi-infinite subset of the real numbers $[0, \infty\}$
$\otimes$	Kronecker matrix product

References

[1] T. A. Badgwell. Robust stability conditions for a SISO model predictive control algorithm. Submitted for publication in *Automatica*, 1995.

[2] R. E. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, New Jersey, 1957.

[3] R. E. Bellman and S. E. Dreyfus. *Applied Dynamic Programming.* Princeton University Press, Princeton, New Jersey, 1962.

[4] L. T. Biegler. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Comput. Chem. Eng.*, 8(3/4):243–248, 1984.

[5] R. R. Bitmead, M. Gevers, and V. Wertz. *Adaptive Optimal Control, The Thinking Man's GPC.* Prentice-Hall, Englewood Cliffs, New Jersey, 1990.

[6] A. E. Bryson and Y. Ho. *Applied Optimal Control.* Hemisphere Publishing, New York, 1975.

[7] M. Caracotsios. *Model Parametric Sensitivity Analysis and Nonlinear Parameter Estimation. Theory and Applications.* PhD thesis, University of Wisconsin, Madison, 1986.

[8] C. C. Chen and L. Shaw. On receding horizon control. *Automatica*, 16(3):349–352, 1982.

[9] D. W. Clarke and C. Mohtadi. Properties of generalized predictive control. *Automatica*, 25(6):859–875, 1989.

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

[11] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized predictive control—Part II. Extensions and interpretations. *Automatica*, 23(2):149–160, 1987.

[12] J. E. Cuthrell and L. T. Biegler. On the optimization of differential-algebraic systems. *AIChE J.*, 33(8):1257–1270, 1987.

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

[14] P. J. Davis. *Interpolation and Approximation.* Dover Books on Advanced Mathematics. Dover Publications, Inc., New York, 1975.

[15] P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration.* Academic Press, Orlando, Florida, 2nd edition, 1984.

[16] S. E. Dreyfus. *Dynamic Programming and the Calculus of Variations.* Academic Press, New York, 1965.

[17] S. E. Dreyfus and A. M. Law. *The Art and Theory of Dynamic Programming.* Academic Press, New York, 1977.

[18] J. W. Eaton. Octave—a high-level interactive language for numerical computations. Technical Report CCSR-93-003, The University of Texas Center for Control and Systems Research, Austin, Texas 78712, 1993.

[19] J. W. Eaton and J. B. Rawlings. Model-predictive control of chemical processes. *Chem. Eng. Sci.*, 47(4):705–720, 1992.

[20] J. W. Eaton, J. B. Rawlings, and L. H. Ungar. Stability of neural net based model predictive control. In *Proceedings of the 1994 American Control Conference*, pages 2481–2485, 1994.

[21] S. S. E. H. Elnashaie. Multiplicity of the steady states in fluidized-bed reactors. III. yield of the consecutive reaction, A→B→C. *Chem. Eng. Sci.*, 32:295–301, 1977.

[22] S. S. E. H. Elnashaie, M. E. Abashar, and F. A. Teymour. Chaotic behaviour of fluidized-bed catalytic reactors with consecutive exothermic chemical reactions. *Chem. Eng. Sci.*, 50(1):49–67, 1995.

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

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

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

[26] G. Gattu and E. Zafiriou. Nonlinear quadratic dynamic matrix control with state estimation. *Ind. Eng. Chem.*, 31(4):1096–1104, 1992.

[27] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* Academic Press, London, 1981.

[28] R. Hermann and A. J. Krener. Nonlinear controllability and observability. *IEEE Trans. Auto. Cont.*, 22(5):728–740, 1977.

[29] A. Isidori. *Nonlinear Control Systems.* Springer-Verlag, Berlin, 2nd edition, 1989.

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

[31] J. C. Kantor. Stability of state feedback transformations for nonlinear systems–some practical considerations. In *Proceedings of the 1986 American Control Conference*, page 1014, Seattle, Washington, June 1986.

[32] S. S. Keerthi. *Optimal Feedback Control of Discrete-Time Systems with State-Control Constraints and General Cost Functions.* PhD thesis, University of Michigan, 1986.

[33] S. S. Keerthi and E. G. Gilbert. An existence theorem for discrete-time infinite-horizon optimal control problems. *IEEE Trans. Auto. Cont.*, 30(9):907–909, September 1985.

[34] S. S. Keerthi and E. G. Gilbert. Moving-horizon approximations for a general class of optimal nonlinear infinite-horizon discrete-time systems. In *Proceedings of the 20th Annual Conference on Information Sciences and Systems*, pages 301–306, Princeton, NJ, 1986. Princeton University.

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

[36] D. L. Kleinman. An easy way to stabilize a linear constant system. *IEEE Trans. Auto. Cont.*, 15(12):692, December 1970.

[37] D. L. Kleinman. Stabilizing a discrete, constant, linear system with application to iterative methods for solving the riccati equation. *IEEE Trans. Auto. Cont.*, 19:252–254, June 1974.

[38] C. Kravaris and P. Daoutidis. Nonlinear state feedback control of second-order nonminimum-phase nonlinear systems. *Comput. Chem. Eng.*, 14(4/5):439–449, 1990.

[39] C.-C. Ku and K. Y. Lee. Diagonal recurrent neural networks for dynamic systems control. *IEEE Trans. Neural Net.*, 6(1):144–156, January 1995.

[40] J. G. Kuschewski, S. Hui, and S. H. Zak. Application of feedforward neural networks to dynamical system identification and control. *IEEE Trans. Cont. Sys. Tech.*, 1(1):37–49, March 1993.

[41] W. H. Kwon, A. M. Bruckstein, and T. Kailath. Stabilizing state-feedback design via the moving horizon method. *Int. J. Control*, 37(3):631–643, 1983.

[42] W. H. Kwon and A. E. Pearson. A modified quadratic cost problem and feedback stabilization of a linear system. *IEEE Trans. Auto. Cont.*, 22(5):838–842, October 1977.

[43] W. H. Kwon and A. E. Pearson. On feedback stabilization of time-varying discrete linear systems. *IEEE Trans. Auto. Cont.*, 23(3):479–481, June 1978.

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

[45] A. U. Levin and K. S. Narendra. Control of nonlinear dynamical systems using neural networks: controllability and stabilization. *IEEE Trans. Neural Net.*, 4(2):192–206, March 1993.

[46] S. Li, K. Y. Lim, and D. G. Fisher. A state space formulation of model predictive control. *AIChE J.*, 35(2):241–249, February 1989.

[47] D. Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Trans. Auto. Cont.*, 35(7):814–824, July 1990.

[48] E. S. Meadows. *Stability and Continuity of Nonlinear Model Predictive Control.* PhD thesis, The University of Texas at Austin, 1994.

[49] E. S. Meadows, M. A. Henson, J. W. Eaton, and J. B. Rawlings. Receding horizon control and discontinuous state feedback stabilization. To appear in *International Journal of Control*.

[50] E. S. Meadows, K. R. Muske, and J. B. Rawlings. Constrained state estimation and discontinuous feedback in model predictive control. In *Proceedings of the 1993 European Control Conference*, pages 2308–2312. European Automatic Control Council, June 1993.

[51] E. S. Meadows and J. B. Rawlings. Topics in model predictive control. In R. Berber, editor, *NATO Advanced Study Institute on Methods of Model Based Control Proceedings*, Antalya, Turkey, July 1994. Kluwer Academic Publishers.

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

[53] M. Morari and E. Zafiriou. *Robust Process Control.* Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[54] K. R. Muske and J. B. Rawlings. Linear model predictive control of unstable processes. *J. Proc. Cont.*, 3(2):85–96, 1993.

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

[56] K. R. Muske and J. B. Rawlings. Nonlinear moving horizon state estimation. In R. Berber, editor, *NATO Advanced Study Institute on Methods of Model Based Control Proceedings*, Antalya, Turkey, July 1994. Kluwer Academic Publishers.

[57] K. R. Muske, J. B. Rawlings, and J. H. Lee. Receding horizon recursive state estimation. In *Proceedings of the 1993 American Control Conference*, pages 900–904, June 1993.

[58] H. Nijmeijer and A. J. van der Schaft. *Nonlinear Dynamical Control Systems.* Springer-Verlag, New York, 1990.

[59] R. E. Nordgrenn and P. H. Merckl. An analytical comparison of a neural network and a model-based adaptive controller. *IEEE Trans. Neural Net.*, 4(4):685–694, July 1993.

[60] A. A. Patwardhan, J. B. Rawlings, and T. F. Edgar. Nonlinear model predictive control. *Chem. Eng. Commun.*, 87:123–141, 1990.

[61] S.-Z. Qin, H.-T. Su, and T. J. McAvoy. Comparison of four neural net learning methods for dynamic system identification. *IEEE Trans. Neural Net.*, 3(1):122–130, January 1992.

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

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

[64] N. L. Ricker. Use of quadratic programming for constrained internal model control. *Ind. Eng. Chem. Proc. Des. Dev.*, 24:925–936, 1985.

[65] D. G. Robertson, J. H. Lee, and J. B. Rawlings. A moving horizon-based approach for least squares state estimation. Submitted for publication in *AIChE J.*, 1994.

[66] R. W. H. Sargent and G. R. Sullivan. The development of an efficient optimal control package. In J. Stoer, editor, *Lecture Notes in Control and Information Sciences 7*, pages 158–168, Berlin, 1978. 8th IFIP Conference on Optimization Techniques, Springer-Verlag.

[67] R. Scattolini and S. Bittanti. On the choice of the horizon in long-range predictive control—some simple criteria. *Automatica*, 26(5):915–917, 1990.

[68] P. O. Scokaert and J. B. Rawlings. Stability of model predictive control under perturbations. In *IFAC Symposium on Nonlinear Control System Design, Tahoe City, California*, 1995.

[69] P. B. Sistu and B. W. Bequette. Model predictive control of processes with input multiplicities. *Chem. Eng. Sci.*, 6(6):921–936, 1995.

[70] E. D. Sontag. An algebraic approach to bounded controllability of nonlinear systems. *Int. J. Control*, 39:181–188, 1984.

[71] H. J. Sussmann and V. Jurdjevic. Controllability of nonlinear systems. *J. Diff. Eq.*, 12:95–115, 1972.

[72] Y. Thomas and A. Barraud. Commande optimale a horizon fuyant. *Revue RAIRO*, J(1):126–140, April 1974.

[73] Y. Thomas, D. Sarlat, and L. Shaw. Receding-horizon approach to the synthesis of nonlinear multivariable regulators. *Electron. Lett.*, 13(11):329–331, May 1977.

[74] Y. A. Thomas. Linear quadratic optimal estimation and control with receding horizon. *Electron. Lett.*, 11:19–21, January 1975.

[75] J. G. Van de Vusse. Plug-flow type reactor versus tank reactor. *Chem. Eng. Sci.*, 19:964, 1964.

[76] M. Vidyasagar. *Nonlinear Systems Analysis.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 2nd edition, 1993.

[77] J. Villadsen and M. L. Michelsen. *Solution of Differential Equation Models by Polynomial Approximation.* Prentice-Hall, Englewood Cliffs New Jersey, 1978.

[78] J. V. Villadsen and W. E. Stewart. Solution of boundary-value problems by orthogonal collocation. *Chem. Eng. Sci.*, 22:1483–1501, 1967.

[79] E. Zafiriou and M. Morari. Design of robust digital controllers and sampling-time selection for SISO systems. *Int. J. Control*, 44(3):711–735, 1986.