CHAPTER 9

Model Predictive Control of Chemical Processes: A Tutorial

Victoria M. Ehlinger, Ali Mesbah

University of California, Berkeley, CA, United States

In this chapter, we will illustrate the ability of model predictive control (MPC) in dealing with the multivariable nature of process dynamics, process constraints, and multiple control objectives in chemical processes. The mathematical formulation of MPC is presented for a general class of processes. The receding-horizon implementation of MPC is demonstrated for a batch crystallization process and a continuous fermentation process. We will discuss the importance of state estimation for output-feedback implementation of MPC when the knowledge of the process state is not fully available. This chapter will conclude with a brief introduction to advanced topics in MPC that are of relevance to chemical process control.

9.1 Why MPC?

Chemical processes commonly exhibit nonlinear, multivariable dynamics subject to input and output constraints. Since its introduction in the 1920s, proportional-integral-derivative (PID) control has been the workhorse of process control.[1] Despite its widespread use, PID control lacks the ability to handle processes with multiple inputs and multiple outputs (i.e., multivariable process dynamics) and process constrais.[1] Hence, in practice, PID controllers are typically implemented in control structures such as cascade, split range, selectors, and overrides to deal with the multivariable nature of process dynamics, making the control structure selection a key consideration in PID control.[3,4] In the 1970s, MPC emerged in the process industry to address the shortcomings of PID control in coping with multivariable process dynamics and constraints.[5] Engineers at Shell published some of the first research papers on MPC and demonstrated its application to a fluid catalytic cracker. Their algorithm became known as dynamic matrix control (DMC), named for the dynamic matrix used to relate predicted future output changes to future inputs moves (i.e., a process model).[6] Although the DMC algorithm performed well for multivariable process control, output constraints could not

[1] Actuation constraints can be accounted for in PID control by adding antiwindup effect.[2]

Coulson and Richardson's Chemical Engineering. http://dx.doi.org/10.1016/B978-0-08-101095-2.00009-6

be handled systematically. Engineers at Shell improved upon DMC by explicitly incorporating the input and output constraints into the online optimization problem; this technique became known as quadratic dynamic model control.[7] These techniques were taken to academia in the early 1980s and triggered extensive research activities over the next three decades to develop MPC techniques for systems with complex dynamics (e.g., nonlinear, uncertain, distributed parameter, hybrid, etc.) and constraints in a wide range of applications. Nowadays in the process industry, MPC is widely used for advanced control of petrochemicals, specialty chemicals, pharmaceuticals, and biochemical processes.[5]

To describe the benefits of MPC, consider a crude unit in a petroleum refinery. The crude unit, the first process unit in a refinery, performs distillation of the crude oil into several fractions, which then undergo a series of reaction and separation processes to produce high value-added products (e.g., gasoline, diesel, and jet fuel).[8] Crude oil commonly contains dozens of chemical compounds, each of which with its own set of physical and thermodynamic properties, making the process dynamics complex. MPC can handle multiple control objectives, such as set point tracking for the several product streams in a crude unit (typically light gases, naphtha, kerosene, diesel, light oil, and heavy gas oil/bottoms). These set points are often adjusted based on the current pricing and market conditions for gasoline, diesel, and jet fuel.[9] While a PID controller can only control a single process output with a single process input, MPC, in theory, does not have any limitations to the number of inputs and outputs it can handle. This is a key advantage for processes with complex dynamics and multiple inputs and multiple outputs, and a primary motivator for the development of MPC in industry. In many chemical processes, maximizing the process productivity can be in conflict with maintaining the desired product quality. For example, maximum production from a crude unit occurs when the amount of light gas products is minimal; however, the light and heavy ends that are now part of the naphtha and diesel fractions result in lower quality products with a lower market value. In addition, a refinery may switch between different feeds with different compositions every few days, but the product specifications remain the same. MPC enables accounting for the range of acceptable product specifications as state or output constraints.[8] Other constraints due to the physical limitations of the crude unit, for example, environmental regulations and safety limitations, can also be accounted for in MPC. Additionally, MPC allows for operating the process close to the process constraints (e.g., operating conditions that correspond to the highest profitability of the process).[10] Overall, the improved process operation achieved by MPC in the process industry can be significant, while the payout time of MPC is typically less than a year.[9]

The key notion in MPC is to use a process model, which can be physics based or empirical, to optimize the process inputs over forecasts of process behavior made over a finite time horizon.[11] MPC offers several advantages over PID control including the ability to handle processes with multiple inputs and multiple outputs, input and output (similarly state) constraints, and multiple control objectives.[12] To illustrate the advantages of MPC, consider the

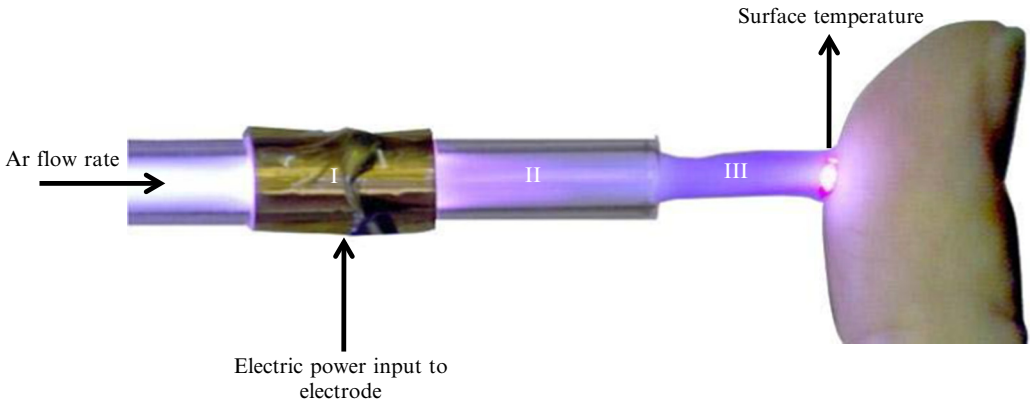

Fig. 9.1
Diagram of an atmospheric-pressure plasma jet (APPJ).[13]

Argon atmospheric-pressure plasma jet (APPJ) shown in Fig. 9.1 (see Ref. 13 for further details). APPJs are used in materials processing and biomedical applications for treatment of materials, such as biological tissues and organic materials, that are heat sensitive or unable to withstand vacuum pressures.[14] In the APPJ shown in Fig. 9.1, the ions and reactive species that are generated in Region I and II, as a result of applying electric field to Argon, extend through Region III to the target surface, which is cooled by the flow of a coolant (e.g., blood in a tissue) and by heat transfer with the surrounding air. An MPC controller is designed to regulate the thermal effects of the plasma on the target surface by driving the temperature of the target toward a desired set point. This is done by manipulating the two inputs of the APPJ device: the electric power input to the electrode and the inlet gas velocity. The APPJ under study is an example of a system with multiple inputs and multiple outputs (i.e., surface temperature and gas temperature in the jet). The ability of the MPC controller in handling the multivariable dynamics of the system alleviates the need to pair the different inputs and outputs of the system, as is the case in PID control.

The control objective of the APPJ is to maintain the surface temperature at a set point of 317 K while the APPJ is held at a prespecified distance from the surface. To ensure safe operation of the APPJ, the gas temperature in the plasma plume is constrained to be less than or equal to 319 K so that no significant changes occur in the plasma characteristics. In the MPC controller, both the velocity and power inputs are used as the system inputs to realize the control objective, whereas the PI controller can only adjust the electric power input to follow the desired surface temperature set point. In addition, the PI controller cannot enforce any constraint on the gas temperature. The closed-loop simulation results for both the MPC controller and the PI controller are shown in Fig. 9.2. At time 500 s, a step disturbance is applied to the distance of

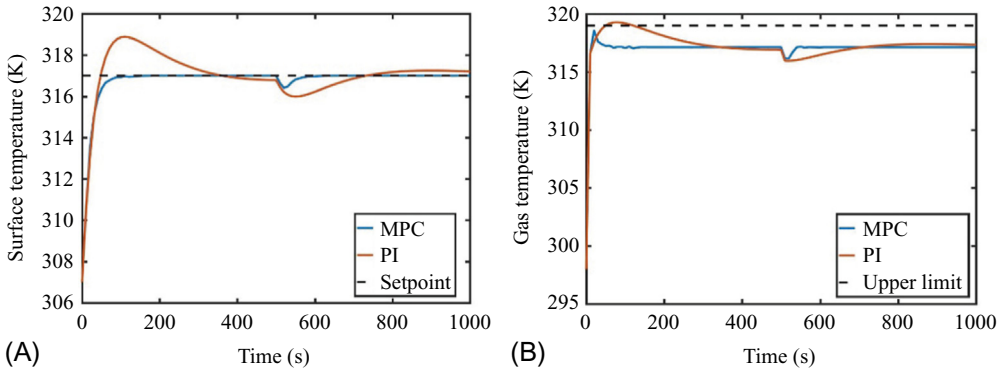

Fig. 9.2

Closed-loop simulation results of the MPC and PI control of the Argon atmospheric-pressure plasma jet shown in Fig. 9.1. The control objective is to follow a desired set point for the surface temperature, while maintaining the gas temperature below a certain limit. A step disturbance in the distance between the jet and the target surface is introduced at time 500 s. (A) Surface temperature set point tracking. The MPC controller shows minimal overshoot and faster set point tracking. (B) Gas temperature constraint handling. The MPC controller maintains the gas temperature within the constraint while the PI controller does not.

the APPJ from the target surface. Because the APPJ is now farther away from the target surface, the surface temperature decreases and the controller must adjust the inputs in order to bring the surface temperature back to its set point of 317 K. Fig. 9.2A shows that both the MPC and PI controllers are able to bring the surface temperature back to its set point after the disturbance is introduced. However, the MPC controller enables reaching the set point more quickly, without overshooting the set point. This is due to the fact that the MPC controller can systematically account for the multivariable nature of the jet dynamics and simultaneously actuate both system inputs toward achieving the control objective. Further, as shown in Fig. 9.2B, the PI controller violates the constraint imposed on the gas temperature, while the MPC controller stays within the constraint limit due to explicit incorporation of the gas temperature constraint in the control framework.

To demonstrate the flexibility of MPC in handling multiple control objectives, the objective function of the MPC controller in the previous example is modified to drive the surface temperature to the set point of 317 K while maintaining the input power level at the desired level of 12 W. The closed-loop simulation results are shown in Fig. 9.3. The MPC controller is able to simultaneously follow the set points for the surface temperature and power level, and therefore realize both control objectives. In contrast, when the PI controller is used for regulating the surface temperature, the (open-loop) power profile cannot meet its respective set point. This motivating example clearly shows the ability of MPC in dealing with the multivariable process dynamics, process constraints, and multiple control objectives.

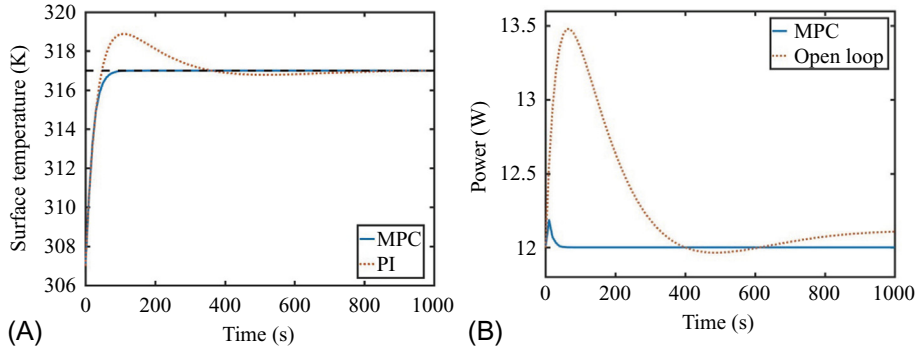

Fig. 9.3

Closed-loop simulation results of MPC and PI control of the Argon atmospheric-pressure plasma jet shown in Fig. 9.1. The MPC controller has multiple control objectives of maintaining the surface temperature and the power at the desired set points, whereas the PI controller can only be tasked to realize one control objective. (A) Surface temperature set point tracking. (B) Power set point tracking. The open-loop profile corresponds to the case in which the PI controller is used for regulating the surface temperature only.

9.2 Formulation of MPC

MPC involves solving an optimal control problem (OCP) in a receding-horizon fashion. The key components of an OCP, which include a process model, an objective function, and input and state constraints, as well as the receding-horizon implementation in MPC are discussed in this section.

9.2.1 Process Model

The first step in developing an MPC controller is to obtain a model of the process. Process models provide a description of the process dynamics in terms of process inputs, states, and outputs. Inputs, also known as manipulated variables, are process variables that can be adjusted by the controller in order to cause a change in the process dynamics. Examples of process inputs, depending on the process configuration, include the flow rate, temperature, and concentration of inlet streams to a unit operation. States are process variables that change in response to input changes and define the state of the process at different times. Examples of states include concentration, temperature, liquid level in a tank, and pressure in a gas drum. Outputs consist of the measured process variables that change in response to input variations. Process outputs are often defined as some function of states. In many chemical processes, process outputs are a subset of states because all states may not be measured due to various technical and economic limitations.

Process models are linear or nonlinear and can be described in discrete- or continuous-time form. The quality of a process model is critical to the performance of an MPC controller. Process models can be derived from data (i.e., empirical) or based on first principles. Empirical models usually have a limited range of validity, whereas first-principle models can typically predict the process behavior over a wider range of process conditions since they rely on mass, momentum, and energy conservation laws, along with constitutive equations (e.g., phase equilibria and chemical kinetics).[15] In its most general form, a process model can be represented by the following (continuous-time) nonlinear state-space representation

$$\begin{aligned} \frac{dx(t)}{dt} &= f(x(t),u(t),\theta), \quad x(t_0)=x_0 \\ y(t) &= h(x(t),u(t),\theta), \end{aligned} \tag{9.1}$$

where $x \in \mathbb{R}^n$ is the vector of states; $u \in \mathbb{R}^{n_u}$ is the vector of (manipulated) process inputs; $y \in \mathbb{R}^{n_y}$ is the vector of (measurable) process outputs; $\theta \in \mathbb{R}^{n_p}$ is the vector of model parameters; t is time; x_0 is the initial conditions of the states; and f and h are some nonlinear functions of the inputs, states, and parameters. $\mathbb{R}$ denotes the set of real numbers.

First-principle models of chemical processes are typically nonlinear due to the nonlinearity of conservation balances and constitutive equations.[16] Nonlinear first-principle models can seldom be solved analytically. Further, their numerical solution can be computationally expensive for real-time control applications. When a process model is intended to describe the system dynamics around a desired operating point (e.g., steady-state operating conditions), nonlinear first-principle models can be linearized around the operating point of interest. In this case, Eq. (9.1) will reduce to a linear state-space model

$$\begin{aligned} \frac{dx(t)}{dt} &= Ax(t)+Bu(t), \quad x(t_0)=x_0 \\ y(t) &= Cx(t)+Du(t), \end{aligned} \tag{9.2}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times n_u}$, $C \in \mathbb{R}^{n_y\times n}$, and $D \in \mathbb{R}^{n_y\times n_u}$, with $D=0$ in many processes.

The state-space model (9.1) can also be written in the discrete-time form, which is convenient for programming of an MPC controller using digital computers. The discrete-time (nonlinear) state-space model takes the form

$$\begin{aligned} x(k+1) &= f_k(x(k),u(k),\theta), \quad x(k=0)=x_0 \\ y(k) &= h_k(x(k),u(k),\theta), \end{aligned} \tag{9.3}$$

where k is the time index; and the functions f_k and h_k are obtained based on the discretization of the functions f and h in Eq. (9.1).

Although first-principle models offer the advantage of a wider operating range and provide physics-based insights into the process dynamics, empirical models are frequently used for

designing MPC controllers in industry because they are generally less expensive to develop and are often computationally more efficient than first-principle models for real-time control applications. Empirical models are derived by perturbing the process (e.g., using a series of step tests), collecting the output response of the process, and subsequently fitting the input-output data to a prespecified (often simple) model parameterization or a state-space model of the form Eq. (9.2).[17,17a] A critical consideration in empirical modeling is the choice of the input perturbations that must sufficiently excite the system dynamics to generate informative data for data-driven model development while being process friendly (e.g., avoid pushing the process outside of its safety limits or leading to an unacceptable amount of off-specification product).[18] Empirical models are only adequate for the operating range over which the data has been collected, whereas first-principle models can be used for a wider operating range. Generally speaking, the modeling approach to be adopted is application specific and critically depends on the process complexity as well as the computational requirements of the MPC application.

9.2.2 Objective Function

The objective function is some function of the process inputs and outputs (or states) that describes the control objective. Control objectives vary depending on the type of process (batch versus continuous) and various process considerations. Typical control objectives include disturbance rejection, set point tracking, batch time minimization, etc. In its general form, the objective function J consists of two parts

$$J(x(k),u(k)) = J_{\mathrm{f}}\left(x\left(N_{\mathrm{p}}\right)\right) + \sum_{i=0}^{N_{\mathrm{p}}-1} J_{\mathrm{c}}(x(i),u(i)), \tag{9.4}$$

where N_{p} is the prediction horizon, the time over which the model predicts the process states or outputs; J_{f} is the terminal cost term; and J_{c} is the running cost term. The running cost describes the control objective over the prediction horizon of the MPC controller, while the terminal cost describes the cost function at the end of the prediction horizon.

In MPC, the objective function is commonly defined in terms of a quadratic set point tracking function

$$J(x(k),u(k)) = \sum_{i=0}^{N_{\mathrm{p}}} \left(y_i - y_{\mathrm{sp}}\right)^{\top} Q \left(y_i - y_{\mathrm{sp}}\right) + \sum_{i=0}^{N_{\mathrm{c}}} \left(u_i - u_{\mathrm{sp}}\right)^{\top} R \left(u_i - u_{\mathrm{sp}}\right), \tag{9.5}$$

where y_{sp} and u_{sp} are the desired set points for the outputs and inputs, respectively; Q and R are symmetric and positive symmetric weight matrices, respectively; and N_{c} is the control horizon, the time over which the controller can adjust the inputs in order to optimize the objective function. Q, R, N_{p}, and N_{c} are generally the tuning parameters of an MPC controller and must be chosen by the user.

9.2.3 State and Input Constraints

One of the key features of MPC is the ability to systematically handle input and state constraints. Input constraints are defined by

$$u \in U, \tag{9.6}$$

which indicates that the process inputs u must lie in a compact set U. Linear input constraints are represented by

$$Du \leq d, \tag{9.7}$$

where $D \in \mathbb{R}^{n_u \times m}$ is the input constraint matrix; $d \in \mathbb{R}^m$ is a positive vector; and m denotes the number of input constraints. Input constraints represent physical limitations of the process such as valve positions and ranges of flow rates.

Like process models, state constraints are categorized as linear and nonlinear. Nonlinear state constraints can generally be defined by

$$g_k(x(k), u(k), \theta) \leq 0, \tag{9.8}$$

where g_k is the state constraint function. In the case of linear state constraints, Eq. (9.8) simplifies to

$$Hx \leq h, \tag{9.9}$$

where $H \in \mathbb{R}^{n \times p}$ is the constraint matrix; $h \in \mathbb{R}^p$ is a positive vector; and p denotes the number of state constraints. Output constraints are defined in a similar manner as state constraints. State or output constraints typically represent limits on species concentration or temperature and pressure of the process. These constraints can be used to systematically account for the safety, regulatory, and product quality requirements of the process.

9.2.4 Optimal Control Problem

The process model, objective function, and input and state constraints defined above can now be used to formulate the OCP, which is the cornerstone of an MPC controller. For the case of full-state feedback $x(k) = y(k)$ (i.e., all states are measured at the sampling time k), MPC solves the following OCP at each sampling time k

$$\min_{\boldsymbol{u}} \; J(x(k), \boldsymbol{u}) \tag{9.10}$$

subject to:

$$\begin{aligned}
&\overline{x}(i+1) = f_k(\overline{x}(i), u(i), \theta), && i = 0 \ldots N_\mathrm{p} - 1 \\
&u(i) \in U, && i = 0 \ldots N_\mathrm{c} - 1 \\
&g_k(\overline{x}(i), u(i), \theta) \leq 0, && i = 1 \ldots N_\mathrm{p} \\
&\overline{x}(0) = x(k),
\end{aligned}$$

where $u = [u(0), \ldots, u(N_c - 1)]^\top$ is the vector of process inputs over the control horizon N_c and $\overline{x}$ denotes the model predictions. $\boldsymbol{u}$ comprises the vector of decision variables of the optimization problem for which the OCP (Eq. 9.10) is solved. The optimal solution to the OCP is denoted by $\boldsymbol{u}^*$ and is referred to as the (open-loop) optimal control policy. When the process model used in the OCP (Eq. 9.10) is nonlinear, the control approach is commonly referred to as nonlinear MPC (NMPC).

9.2.5 Receding-Horizon Implementation

In practice, the performance of the open-loop optimal control policy $\boldsymbol{u}^*$ computed by Eq. (9.10) will degrade due to the inevitable plant-model mismatch (i.e., inaccuracies of the process model) and process disturbances. To mitigate the performance degradation of the (open-loop) optimal control policies, the OCP must be solved recursively when new process measurements $x(k)$ become available at every measurement sampling time instant k. This is known as the receding-horizon implementation of MPC[15] and is illustrated in Fig. 9.4. The idea is as follows. At the sampling time instant k, the OCP (Eq. 9.10) is solved over the control horizon N_c. The solution of the OCP is a sequence of optimal input values, for which the objective function is minimized over the prediction horizon N_p. The first element of the optimal input sequence, $u^*(0)$, is then applied to the process. Subsequently, the prediction horizon shifts one sample ahead and the procedure is repeated as soon as new measurements become available at the next sampling time instant $k + 1$. In the case of full-state feedback, a closed-loop control system with an MPC controller is shown in Fig. 9.5. The optimal inputs from the MPC controller are applied to the plant and the measured outputs of the plant (in this case the full state

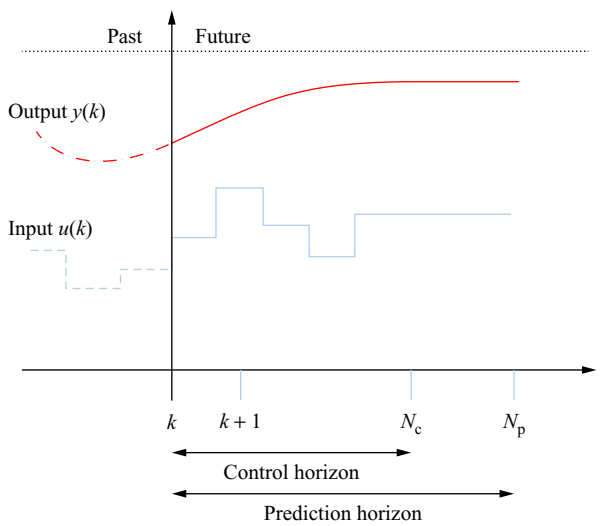

Fig. 9.4
The receding-horizon principle in MPC.

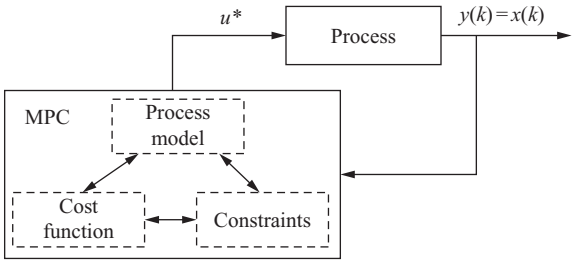

Fig. 9.5
Model predictive control in a closed-loop control system when all the state variables are measured (i.e., the case of full-state feedback).

vector) are fed to the MPC controller. The new measurements are used to reinitialize the process model at every sampling time k (see the OCP (Eq. 9.10)).

For processes with a fixed end time, the receding-horizon implementation of the OCP can be adapted to shrinking horizon. In this case, when the prediction horizon is shifted to the next sampling time instant, the prediction horizon shrinks as the process approaches the end time of the process.[19] The shrinking-horizon implementation is particularly useful for batch processes, whereas the receding-horizon implementation is typically used for continuous processes.

9.2.6 Optimization Solution Methods

The OCP (Eq. 9.10) is in fact a dynamic optimization problem. There exists various direct optimization strategies, namely, single shooting, multiple shooting, and simultaneous strategies, for solving the dynamic optimization problem. Generally speaking, these strategies convert the OCP into a finite dimensional nonlinear programming problem (NLP) through parameterizing the vector of decisions variables $\boldsymbol{u}$. In the single shooting strategy,[20] after reducing the infinitely many degrees of freedom of the control vector through parameterization, an initial value problem encompassing the model equations is numerically solved in each iteration step of the optimization procedure. The solution is obtained for the current values of the parameterized control vector. Hence, model simulation and optimization are carried out sequentially, guaranteeing the solution feasibility even in the case of premature optimization terminations. By contrast, the model equations are discretized along with the control vector in the simultaneous optimization strategy.[21] The discretized differential equations are included in the optimization problem as nonlinear constraints, typically leading to very large NLPs. Simultaneous model simulation and optimization can result in faster computations compared to the single shooting strategy. However, feasible state trajectories are obtained only after successful termination of the optimization since the discretized model equations are violated during the optimization procedure.

Table 9.1 Comparison of the direct optimization strategies[24]

	Single Shooting	Multiple Shooting	Simultaneous
Use of integration solvers	Yes	Yes	No
Size of nonlinear programming problem	Small	Intermediate	Large
Applicable to highly unstable systems	No	Yes	No
Model equations fulfilled in each iteration step	Yes	Partially	No

On the other hand, model simulation and optimization are not performed entirely sequentially or simultaneously in the multiple shooting strategy.[22,23] In this optimization strategy, the state trajectories and the control vector are parametrized over a predetermined number of intervals. The initial value problems are solved separately on each multiple shooting interval with a prespecified numerical accuracy. This makes the technique well suited for parallel computations. The relatively large number of variables often necessitates the use of tailored NLP algorithms, which exploit the special structure of the problem, e.g., sparsity, to yield faster convergence than for the single shooting strategy. However, the continuity of the state trajectories is only fulfilled after successful termination of the optimization procedure as with the simultaneous technique. Table 9.1 summarizes the direct optimization strategies.

9.3 MPC for Batch and Continuous Chemical Processes

This section demonstrates the NMPC for a batch crystallization process and a continuous fermentation process. The case studies include excerpts of code from MATLAB (MathWorks) to illustrate how a closed-loop control system, an optimization objective function, and process constraints are set up.

9.3.1 NMPC of a Batch Crystallization Process

Batch crystallization is prevalent in the specialty chemical, food, and pharmaceutical industries for manufacturing and purification of high value-added chemical substances. Crystallization processes are governed by several physicochemical phenomena such as nucleation, crystal growth, and agglomeration, which arise from the copresence of a continuous phase (i.e., solution) and a dispersed phase (i.e., particles).[25,26] Optimal operation of batch crystallization processes is particularly challenging due to the complexity of their models, uncertainty of the crystallization kinetics, and sensor limitations in reliably measuring the process variables.[27]

This case study demonstrates the NMPC for seeded batch crystallization of an ammonium sulfate-water system. The dynamics of crystal size distribution over the batch time can be described by the population balance equation (PBE), which is a nonlinear partial differential

equation[28] (e.g., see Ref. 29 for an overview of various numerical solution methods for the PBE). In order to obtain a computationally efficient description of the process dynamics for the real-time control application at hand, the method of moments is applied to convert the PBE to a set of closed-form nonlinear ordinary differential equations of form (9.3) with the state vector

$$x = [m_0\, m_1\ m_2\, m_3\, m_4\, C]^\top,$$

where m_i is the ith moment of the crystal size distribution and C is the concentration of ammonium sulfate in solution. The input u consists of the heat input Q to the process, i.e., $u = [Q]$. A key process variable, which is closely related to the product quality and batch productivity, is the crystal growth rate that is defined in terms of the solute concentration

$$G = k_g (C - C^*)^g, \tag{9.11}$$

where k_g is the crystal growth rate constant; C^* is the saturation concentration; and g is the growth rate exponent. A detailed description of the batch crystallization process under study and the moment model can be found in Refs. 30,31. The objective of the NMPC controller is to maintain the crystal growth rate at a desired set point $G_{\max}$, which provides a trade-off between the batch productivity and product quality. To this end, the OCP (Eq. 9.10) is defined as

$$\min_{\boldsymbol{u}} \sum_{i=0}^{N_p} \left(\frac{G(i) - G_{\max}}{G_{\max}} \right)^2 \tag{9.12}$$

subject to:

$$\begin{aligned} &\overline{x}(i+1) = f\_k(\overline{x}(i), u(i), \theta), \quad i = 0 \ldots N_p - 1 \\ &9\text{kW} \le u(i) \le 13\text{kW}, \qquad\qquad i = 0 \ldots N_c - 1 \\ &\overline{x}(0) = x(k), \end{aligned}$$

where $G_{\max} = 2.5 \times 10^{-8}\text{m/s}$.

To solve the OCP in Eq. (9.12), the MATLAB constrained optimization function `fmincon`,[32] which is a sequential quadratic programming solver, is used. For the case of full-state feedback, the excerpt of the code for shrinking-horizon implementation of the OCP is shown in Box 9.1. The code resembles the closed-loop control system depicted in Fig. 9.5. The optimizer `fmincon` relies on the objective function `objbatch`, in which the objective function of Eq. (9.12) is defined, and the constraint function `conbatch`. At each sampling time, the optimizer `fmincon` takes the initial guess of the decision variables `uopt_past`, their lower bound `lb` and upper bound `ub`, and the most recent observed states of the process `x_plant(k,:)` to compute the optimal process inputs `uopt`. The decision variables of the dynamic optimization problem are parameterized using a prespecified number, `ninter`, of piecewise-constant intervals over the prediction horizon `tp`. Since the NMPC controller is implemented in a shrinking-horizon

BOX 9.1

```
for k = 1:length(ts)-1,
    % The OCP
    [uopt] = fmincon(@objbatch,uopt_past,[], [], [], [],...
        lb,ub,@conbatch,[],x_plant(k,:),tp,ninter);

    % Adjusting the prediction horizon for shrinking-horizon implementation of
      the OCP
    tp = tb - ts(k);

    % Plant
    [t,x] = ode15s(@crystallizer,[ts(k) ts(k+1)],x_plant(k,:),[],uopt(1));
    x_plant(k+1,:) = x(end,:);
    uopt_past = uopt;
end
```

mode, the prediction horizon `tp` is shortened at every measurement sampling time, that is, `tp = tb - ts(k)` with `tb` being the fixed batch time. The optimizer computes the optimal inputs `uopt` over the control horizon, the first element of which is applied to the plant. The plant outputs `x_plant(k+1,:)` are then used to initialize the optimizer at the next sampling time instant `k+1`. See the Appendix for the code of the process model.

The first argument in `fmincon` is the objective function `objbatch`, shown in Box 9.2. The arguments of `objbatch` are the initial guess of the decision variables `u=uopt_past`, the initial conditions `x0=x_plant(k,:)`, the prediction horizon `tp`, and the number of intervals `ninter` over which the decision variables are parameterized. Note that the process model `crystallizer` is solved numerically using the MATLAB solver `ode15s` over the `ninter` intervals of the prediction horizon, in each of which the decision variable `u` takes a constant value. As defined in the OCP (Eq. 9.12), the objective function `objbatch` aims to drive the crystal growth rate to its set point G_{max} over the course of the batch run time. Since the OCP does not include any state constraints, the constraint function `conbatch` in the optimizer need not be used.

The simulation results are shown in Fig. 9.6. To illustrate the importance of the receding-horizon implementation of the NMPC controller, three scenarios have been considered: open-loop control under the assumption of perfect model as well as open-loop and closed-loop control in the presence of plant-model mismatch. In both open-loop control scenarios, the optimal heat input to the crystallizer is computed off-line by solving the OCP (Eq. 9.12) without reinitializing the model at every sampling time (i.e., the online process measurements are disregarded). The offline-computed optimal control inputs are then implemented on the crystallizer. In the hypothetical scenario of having a perfect process model, the open-loop optimal heat input and crystal growth rate profiles are shown in Figs. 9.6A and B,

BOX 9.2

```
function obj = objbatch(u,x0,tp,ninter)
IC(1,:) = x0;  % Initial state conditions for each interval of control input
                parameterization
x1 = [];

% Model prediction over the prediction horizon using the parameterized control input
for i=1:ninter
    [t,x] = ode15s(@crystallizer,0:1:tp/ninter,IC(i,:),[],u(i));
    IC(i+1,:)=x(end,:);
    x1 = [x1;x];
end

% Calculate growth rate G using the predicted concentration
kg = 7.49567e-05;  % growth constant
g = 1;             % growth exponent
Cs = 0.456513;     % saturation concentration C*
Gmax = 2.5e-8;     % [m/s]
G = kg * (x1(:,6) - Cs).^g;

% Evaluate objective function
obj = sum(((G-Gmax)./Gmax).^2);
```

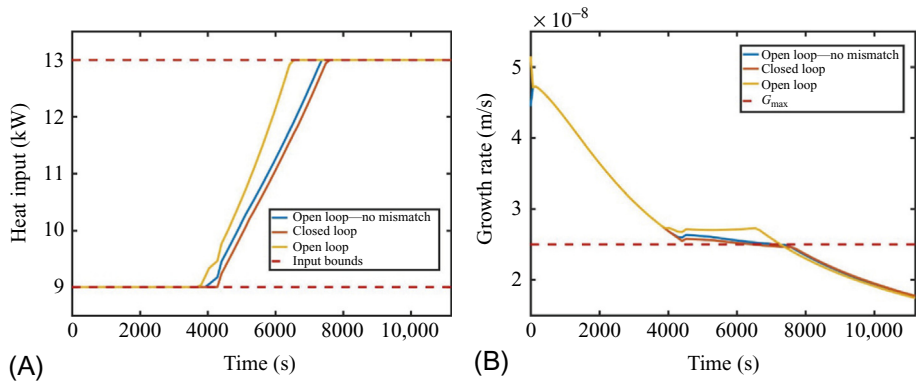

Fig. 9.6

The NMPC for the batch crystallization process. Comparison of three implementations of the OCP (Eq. 9.12): open-loop implementation under the assumption of perfect system model, open-loop and closed-loop implementations in the presence of plant-model mismatch. The open-loop implementation without plant-model mismatch is nearly identical to the closed-loop control in the presence of plant-model mismatch. This indicates the importance of the measurement feedback (i.e., receding-horizon implementation) in the NMPC controller to account for plant-model mismatch and process uncertainties. (A) Heat input to the crystallizer. (B) Crystal growth rate profile.

respectively. It is evident that the crystal growth rate can follow its set point only in a limited time period of 4200 s to 7800 s. This is due to the upper and lower bounds enforced on the process input. Initially, the controller must decrease the heat input below 9 kW to bring the crystal growth rate close to its set point. However, the lower heat input bound of 9 kW, imposed to ensure reproducible batch startups, prevents the controller from achieving its control objective. When the crystal growth rate naturally decays to its set point at approximately 4200 s, the controller starts increasing the heat input to maintain the growth rate at its desired set point. However, as soon as the heat input hits its upper bound of 13 kW due to the actuation limitations of the crystallizers, the controller can no longer effectively achieve its objective of $G_{\max}$ set point tracking. Fig. 9.6 clearly demonstrates the effect that the lack of process actuation (in this case due to the physical bounds on the heat input) can play in hampering effective process control.

The earlier discussed open-loop optimal control policy relied on the unrealistic assumption that a perfect process model was available. Due to process uncertainties and various modeling assumptions, there is always a certain degree of mismatch between the model predictions and the actual process behavior. Fig. 9.6 shows an open-loop optimal control policy that has been designed off-line with a process model that has parametric uncertainty in the crystallization kinetics (i.e., plant-model mismatch). Fig. 9.6B shows that the open-loop optimal control policy cannot closely follow the $G_{\max}$ set point even in the time period of 4200–7800 s where the controller can modulate the heat input to track the crystal growth set point. This is due to the fact that the OCP is solved off-line and, therefore, it cannot take corrective action using the online plant measurements to counteract the plant-model mismatch. Fig. 9.6 indicates that when the NMPC controller is implemented in closed loop by solving the OCP (Eq. 9.12) online, despite the plant-model mismatch, the optimal control inputs can fulfill the control objectives in the operating region that the process input can be adequately modulated (i.e., between 4200 s and 7800 s). As can be seen, the crystal growth rate profile in the case of the closed-loop control exhibits almost the same behavior as in the case of the open-loop control with no plant-model mismatch.

Notice that process constraints can be classified as either hard or soft constraints.[33] Hard constraints are those that are defined explicitly in the OCP and must be satisfied at all times. If a hard constraint is not satisfied, then the optimization problem will become infeasible. On the other hand, soft constraints are defined in the objective function. For example, in the OCP (Eq. 9.12), the crystal growth rate set point tracking can be interpreted as a soft constraint of the form $G \leq G_{\max}$. Soft constraints can be violated to ensure feasibility of the optimization problem in the interest of satisfaction of the hard state constraints and/or input bounds. In the batch crystallization case study, the actuation constraints on the process input led to violation of the soft constraint on the crystal growth rate during the initial and final phase of the batch.

9.3.2 NMPC of a Continuous ABE Fermentation Process

Biobutanol derived from sustainable renewable resources such as lignocellulosic biomass has shown promise as a renewable drop-in fuel.[34,35] Biobutanol can be produced by bacteria of genus *Clostridium* in the so-called acetone-butanol-ethanol (ABE) fermentation process, which is a biphasic fermentation that converts sugars into acids (acetate, butyrate) and solvents (acetone, butanol, ethanol). During the first phase, known as acidogenesis, the primary products are the acidic metabolites. As the metabolism shifts to the second phase, solventogenesis, the acids are assimilated into the ABE solvents that comprise the main fermentation products.[36]

In this case study, an NMPC controller is designed for a continuous ABE fermentation process. The nonlinear process model used for the NMPC design consists of 12 states, which are the concentrations of various chemicals and enzymes in the fermentation culture.[37,38] The dynamics of the continuous fermentation process are described by the nonlinear state-space model (9.3) with the state vector defined by

$$x = [C_{\mathrm{AC}}\ C_{\mathrm{A}}\ C_{\mathrm{En}}\ C_{\mathrm{AaC}}\ C_{\mathrm{Aa}}\ C_{\mathrm{BC}}\ C_{\mathrm{B}}\ C_{\mathrm{An}}\ C_{\mathrm{Bn}}\ C_{\mathrm{Ad}}\ C_{\mathrm{Cf}}\ C_{\mathrm{Ah}}]^{\top},$$

where the abbreviations correspond to the species in the fermentation culture: AC is acetyl-CoA, A is acetate, En is ethanol, AaC is acetoacetate-CoA, Aa is acetoacetate, BC is butyryl-CoA, B is butyrate, An is acetone, Bn is butanol, Ad is adc, Cf is ctfA/B, and Ah is adhE (see Ref. 38). The inputs to the process are

$$u = [D\ G_0]^{\top},$$

where D (h^{-1}) is the dilution rate and G_0 is the inlet glucose concentration (mM). A detailed description of the metabolic pathway of *Clostridium acetobutylicum* can be found in Ref. 36. In the following, we denote each state variable as x_i, where the subscript i denotes the ith entry of the state vector.

The control objective is to track a set point for butanol concentration, while enforcing state constraints on the concentration of acids in order to retain the pH of the fermentation culture within a desired range. For this case study, the OCP (Eq. 9.10) takes the form

$$\min_{\boldsymbol{u}} \sum_{i=0}^{N_{\mathrm{p}}} \left(x_9(i) - C_{\mathrm{Bn}}^{\mathrm{sp}}\right)^2 \tag{9.13}$$

subject to:

$$\begin{aligned}
&\hat{x}(i+1) = f_k(x(i), u(i), \theta), && i = 0 \ldots N_{\mathrm{p}} - 1\\
&0.005\,\mathrm{h}^{-1} \le u_1(i) \le 0.145\,\mathrm{h}^{-1}, && i = 0 \ldots N_{\mathrm{c}} - 1\\
&0\ \mathrm{mM} \le u_2(i) \le 80\,\mathrm{mM}, && i = 0 \ldots N_{\mathrm{c}} - 1\\
&13.83\,\mathrm{mM} \le x_2(i) \le 15.68\,\mathrm{mM}, && i = 1 \ldots N_{\mathrm{p}}\\
&10.55\ \mathrm{mM} \le x_7(i) \le 12.30\,\mathrm{mM}, && i = 1 \ldots N_{\mathrm{p}}\\
&\overline{x}(0) = x(k),
\end{aligned}$$

where C_{Bn}^{sp} denotes the butanol set point; u_1 denotes the dilution rate; u_2 denotes the inlet glucose concentration; x_2, x_7, and x_9 denote the acetate, butyrate, and butanol concentrations, respectively. The prediction and control horizons are chosen as $N_p = N_c = 150$ h. It is assumed that all the state variables are measured at every measurement sampling time. The sampling intervals are 0.33 h.

Like the previous case study, the MATLAB constrained optimization function `fmincon` is used to solve the OCP (Eq. 9.13). The code for the receding-horizon implementation of the OCP is given in Box 9.3. The optimizer `fmincon` calls the objective function `objfun` and the constraint function `constraints`. See the Appendix for the code of the process model.

In the objective function `objfun`, the process model is used to evaluate the cost function, which is defined as the sum of least squares between the predicted butanol concentration over the prediction horizon and the set point for butanol concentration. In the objective function `objfun`, `SP` $= C_{Bn}^{sp}$ is the set point for the butanol concentration and `Np` is the prediction horizon in hours (see Box 9.4). Here, `ode15s`, a stiff differential equation solver, is used to solve the system

BOX 9.3

```
for k = 1:N
    % The OCP
    u_opt(k+1,:)= fmincon(@objfun,u_opt(k,:),[], [], [],...
        [],lb,ub,@constraints,[],y(k,:),SP,Np);

    % Plant model
    [t,x]=ode15s(@ABE_model,meas_samp_time,y(k,:),[],...
        u_opt (k,:));

    % New measurements to reinitialize the OCP at the
      next sampling time
    y(k+1,:)=x(end,:);
end
```

BOX 9.4

```
function obj = objfun(u,x0,SP,Np)
    global A
    global B
    % model prediction
    [t, x]=ode15s(@ABE_model,0:1/3:Np,x0,[],u);
    A = x(:,2);  % acetate concentration
    B = x(:,7);  % butyrate concentration

    % objective function
    obj = sum((x(:,9)-SP).^2);
end
```

model `ABE_model`. In order to incorporate the state constraints into the optimization problem, the global variables `A` and `B` are declared and assigned as the predicted values of the acetate and butyrate concentrations over the prediction horizon. Through defining `A` and `B` as global variables, the predicted concentrations of acetate and butyrate are passed to the constraint function `constraints`, in which linear inequality constraints are defined in terms of the lower and upper bounds of each state (see Box 9.5). Notice that the arguments for the objective function `objfun` and constraint function `constraints` must be the same, regardless of whether or not each argument is used in both functions. The global variables `A` and `B` must be declared again inside of the `constraints` function. Once the predicted values of the states are evaluated in the objective function `objfun`, they are sent to the `constraints` function in order to impose the upper and lower bounds on the acetate and butyrate concentrations, as shown in Box 9.5.

The closed-loop simulation results of the NMPC controller are shown in Fig. 9.7. The process starts at steady state at time 0 h and a set point change is introduced at this time to increase the concentration of butanol by 10%. Once the fermentation process has reached the set point, another set point change is introduced at time 60 h to decrease the butanol concentration by 20%. The NMPC controller is tasked with accommodating the butanol set point changes while ensuring that the acetate and butyric concentrations remain within the process constraints. The closed-loop simulation results in Fig. 9.7 indicate that the NMPC controller can effectively retain the concentrations of acetate and butyrate within the specified limits, which is necessary to maintain the pH of the fernentation culture at a desired level and to remain within the validity range of the process model. At time 60 h, the acid concentrations decrease in response to the set point change in butanol concentration. The simulation results suggest that the NMPC controller enables effective transition of the process between the two operating conditions (i.e., butanol set points).

BOX 9.5

```
function [c,ceq] = constraints(u,x0,SP,time)
    global A
    global B
    Aub = 15.68; % upper bound on acetate concentration (mM)
    Alb = 13.83; % lower bound on acetate concentration (mM)
    Bub = 12.30; % upper bound on butyrate concentration(mM)
    Blb = 10.55; % lower bound on butyrate concentration(mM)

    % Inequality constraints
    c = [A - Aub; Alb - A; B - Bub; Blb - B];

    % Equality constraints
    ceq = [];
end
```

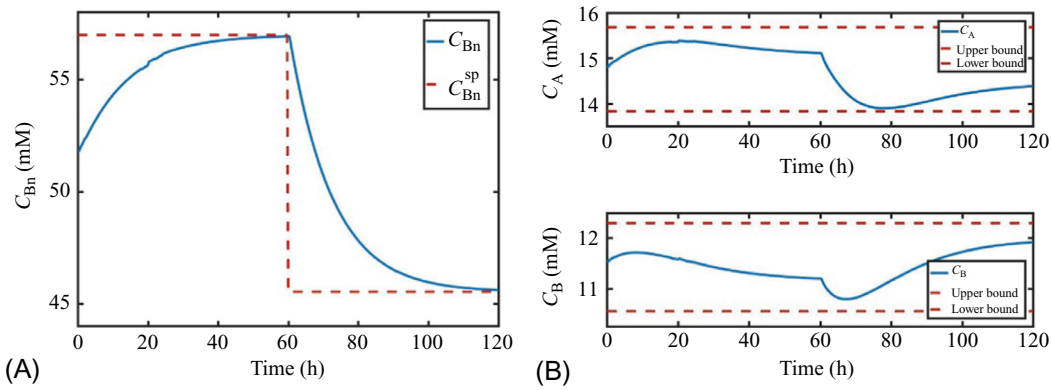

Fig. 9.7

The NMPC of the continuous ABE fermentation process. The NMPC controller enables transition from one operating condition to another (defined in terms of the butanol set point) without violating the state constraints on acetate and butyrate. (A) Set point tracking for butanol. (B) State constraints on acetate and butyrate.

The flexibility of using the NMPC controller over a wide operating range (i.e., changing the butanol set points) is due to the use of a nonlinear model in the NMPC controller. If the ABE fermentation process was to be controlled around a specific operating condition, a MPC controller could be designed using a linear model obtained via linearization of the nonlinear process model around the desired operating point. MPC based on linear models typically offers computational advantages over NMPC for real-time control applications.

9.4 Output-Feedback MPC

Up until this point, we have assumed that all of the process state variables can be measured and directly used for receding-horizon implementation of MPC in a full-state feedback fashion. In practice, however, it may not be feasible to measure all of the state variables due to various technical or economic limitations. This gives rise to the problem of state estimation. A state estimator uses a process model in combination with measurements from the process in order to reconstruct the state vector.[39,40] As shown in Fig. 9.8, the estimated states can then be used to reinitialize the process model in the MPC controller. This is known as the output-feedback implementation of MPC.

A wide range of methods have been developed for state estimation of nonlinear (bio)chemical processes.[41,42] A classification of nonlinear state estimation methods with exponential convergence properties is shown in Fig. 9.9. Broadly speaking, these methods can be categorized as methods developed under a deterministic framework and methods developed under the Bayesian estimation framework. The deterministic state estimation methods such as the extended Luenberger observer (ELO) neglect the stochastic process disturbances.[43,44]

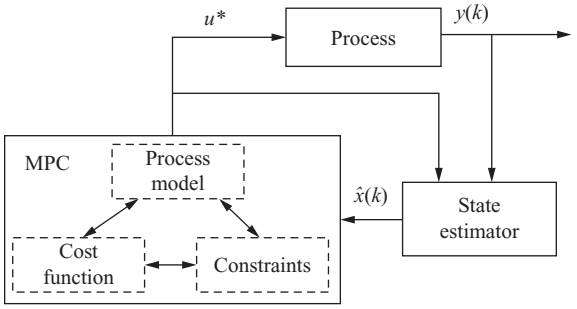

Fig. 9.8
Output-feedback implementation of MPC using a state estimator.

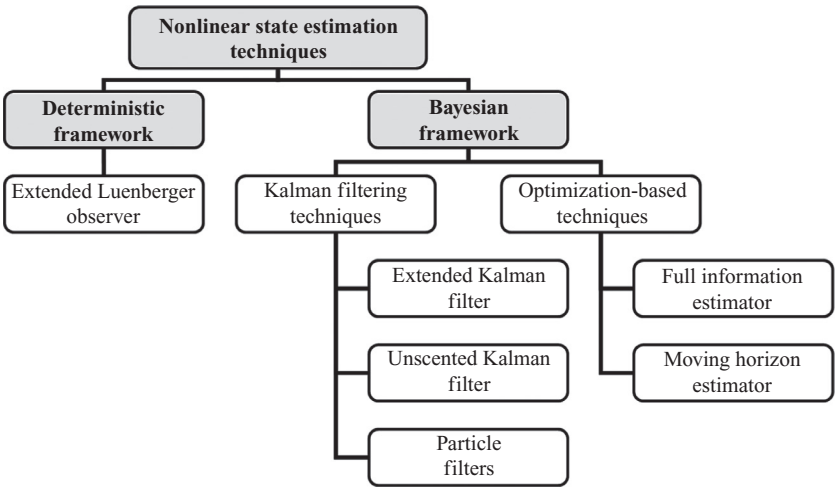

Fig. 9.9
A broad classification of nonlinear state estimation methods.

On the other hand, the Bayesian state estimation methods determine the probability density function (pdf) of the states under the effects of stochastic process and measurement noise. The most widely used nonlinear state estimation method is the extended Kalman filter (EKF), which is an adaptation of the Kalman filter for nonlinear systems.[45,46] EKF relies on linearization of the nonlinear process model around the current state estimates, and the assumption that the state variables and the process and measurement noise are described by Gaussian distributions. However, for highly nonlinear processes, the assumption that the state variables will retain their Gaussian distribution after propagation through the nonlinear process dynamics may not be valid. In addition, EKF may not be applicable to nondifferentiable systems since it requires linearization of the process model. These considerations have motivated the development of sample-based (derivative-free) methods for nonlinear state

estimation. In the unscented Kalman filter (UKF), the linearization of the process model is avoided by an unscented transformation of a set of heuristically chosen samples through the nonlinear process model, thus eliminating the need for linearization.[47] However, UKF stills relies on the assumption that the state variables are described by a Gaussian distribution. On the other hand, particle filters do not make any assumptions about the process dynamics or the type of state distributions.[48] In particle filters, the pdfs of states are constructed using a series of random samples, called particles. Monte Carlo techniques are typically used to generate samples and then the pdfs of states are estimated based on Bayes rule. As the number of samples grows, the estimated pdfs of states must converge to their true pdfs.[49]

A different class of state estimation methods includes optimization-based estimation techniques.[50] Since (bio)chemical processes are often subject to state constraints due to various operational and economic considerations, optimization-based state estimation techniques can be used to explicitly include the state constraints into the nonlinear estimation problem. In contrast to the earlier discussed estimation methods that use the most recent process measurements only, optimization-based state estimators utilize the measurements obtained over a prespecified estimation horizon to minimize the difference between the model predictions and the measurements in a weighted least-squares sense. In general, nonlinear optimization-based estimation techniques can be classified as full information or moving horizon estimators.[51] The estimation horizon of the full information estimators grows as new process measurements become available, whereas in the moving horizon estimators the optimization is performed over a finite estimation horizon to avoid excessively large computational burdens.

Since we have adopted a deterministic setting for process modeling in this chapter (i.e., the process model (9.3) includes no stochastic noise), only the Luenberger observer and its extension for nonlinear processes are introduced in the remainder of this section.

9.4.1 Luenberger Observer

The linear state-space model (9.2) can be written in the discrete form

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \quad x(k=0) = x_0 \\ y(k) &= Cx(k), \end{aligned} \tag{9.14}$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times n_u}$, and $C \in \mathbb{R}^{n_y \times n}$. A Luenberger observer is designed as[43]

$$\hat{x}(k+1) = Ax(k) + Bu(k) + K(k)(y(k) - C\hat{x}(k)), \tag{9.15}$$

where $\hat{x}(k)$ is an estimate of the state vector and $K(k)$ is the observer gain, which governs the accuracy and convergence properties of the state estimator. The right-hand side of Eq. (9.15) consists of a copy of the process model and a correction term that is defined as the difference between the estimated outputs and the measured process outputs multiplied by the observer gain.

The goal of the Luenberger observer is to provide an estimate of the state vector such that the estimation error

$$\begin{aligned} e(k+1) &= x(k+1) - \hat{x}(k+1) \\ &= A(\hat{x}(k) + e(k)) - A\hat{x}(k) - K(k)(C(\hat{x}(k) + e(k)) - C\hat{x}(k)), \end{aligned} \tag{9.16}$$

is zero. The estimation error is, by definition, the difference between the true states and the estimated states. The structure of the gain matrix $K(k)$ is defined on the basis of physical insights into the process, while its tuning parameters are typically obtained by running open-loop simulations and evaluating the evolution of the states.[52,53]

9.4.2 Extended Luenberger Observer

For nonlinear processes, the ELO is derived through linearization of the nonlinear process model. Using the nonlinear state-space model in Eq. (9.3), the ELO is written as

$$\hat{x}(k+1) = f_k(\hat{x}(k), u(k)) + K(k)(y(k) - h_k(\hat{x}(k))), \tag{9.17}$$

where the estimation error is of the form[44]

$$\begin{aligned} e(k+1) &= x(k+1) - \hat{x}(k+1) \\ &= f_k(\hat{x}(k) + e(k), u(k)) - f_k(\hat{x}(k), u(k)) - K(k)(h_k(\hat{x}(k) + e(k)) - h_k(\hat{x}(k))). \end{aligned} \tag{9.18}$$

Due to the nonlinear dynamics, the condition under which the error converges to zero cannot be readily determined from the error dynamics. Therefore, the observer must be designed based on a linearized version of the nonlinear process model. Linearization of the process model around the current state estimates $\hat{x}(k)$ yields

$$e(k+1) = (A(k) - K(k)C(k))e(k), \tag{9.19}$$

where $A(k) = \left[\frac{\partial f_k(x(k),u(k))}{\partial x}\right]\Big|_{x(k)=\hat{x}(k)}$ and $C(k) = \left[\frac{\partial h_k(x(k),u(k))}{\partial x}\right]\Big|_{x(k)=\hat{x}(k)}$. Generally, the estimation accuracy of the ELO depends on how well the linearized model represents the true process dynamics and how close the initial state estimates $\hat{x}(0)$ are to the true initial states (e.g., see Refs. 52,54).

9.4.3 NMPC of the Batch Crystallization Process Under Incomplete State Information

In the batch crystallization case study presented in Section 9.3, it was assumed that the full state vector could be measured. In many crystallization processes, however, obtaining reliable measurements for the solute concentration may be impractical. Revisiting this case study, we assume that the solute concentration measurement is not available. Thus, an ELO is designed to facilitate the output-feedback implementation of the NMPC controller. The ELO estimates the state variables, which are used for reinitializing the OCP (Eq. 9.12) at every measurement

BOX 9.6

```
for k = 1:length(ts)-1,
    % The OCP
    [uopt] = fmincon(@objbatch,uopt_past,[], [], [], [],...
        lb,ub,@conbatch,[],xhat(k,:),tp,ninter);

    % Adjusting the prediction horizon for shrinking-horizon implementation of the OCP
    tp = tb - ts(k);

    % Plant
    [t,x] = ode15s(@crystallizer,[ts(k) ts(k+1)],x_plant(k,:),[],uopt(1));
    x_plant(k+1,:) = x(end,:);
    uopt_past = uopt;
    % ELO (state estimation)

    [t,xobs] = ode15s(@ELO,[ts(k) ts(k+1)], xhat(k,:), [],uopt(1),
              x_plant(k,1:5), x_plant(k+1,1:5),ts(k),ts(k+1));
    xhat(k+1,:) = xobs(end,:);
end
```

sampling time. As demonstrated by the simulation results (see Fig. 9.6), the closed-loop implementation of the NMPC controller is essential for mitigating the performance degradation of the optimal control inputs due to plant-model mismatch and process uncertainties.

Box 9.6 gives the MATLAB excerpt for the output-feedback implementation of the NMPC, as depicted in Fig. 9.8. To this end, the feeback control loop described in Box 9.1 (for the case of full-state feedback) is modified by adding the ELO state estimator. At every sampling time `k`, the state estimator takes the plant outputs `x_plant` and constructs the vector of state estimates `xhat`. The state estimates `xhat` are then used for reinitializing the optimizer `fmincon`. Notice that the same process input `uopt(1)` applied to the process is also applied to the ELO. The state estimator `ELO` is given in Box 9.7. `ELO` involves solving the process model over the sampling interval. The model predictions are modified by the error terms `E` to account for the discrepancy between the model predictions and the process measurements. The observer gain `K` is computed off-line as discussed in Ref. 31. See Appendix for the MATLAB codes.

The closed-loop simulation results for the output-feedback NMPC controller are shown in Fig. 9.10. The optimal heat input and crystal growth rate profiles are similar to the case of the full-state feedback NMPC controller presented in Fig. 9.6 in spite of reconstructing the solute concentration using the available measurements. Fig. 9.10C shows the estimated solute concentration profile versus the true solute concentration profile, which is considered to be unmeasurable. The slight difference between the true concentration and the estimated concentration results from the plant-model mismatch that is due to the kinetic

BOX 9.7

```
function dxdt = ELO(t,x0,u,x1,x2,t1,t2)

% Initial state variables
m0 = x0(1); m1 = x0(2); m2 = x0(3); m3 = x0(4); m4 = x0(5); w = x0(6);

% Observer gain (computed offline)
K = [-0.00272458 84.5034 0.117009 0.441494 0.463708];

% Error signals
E(1) = x1(1) + (t - t1)*(x2(1) - x1(1))/(t2 - t1) - m0;
E(2) = x1(2) + (t - t1)*(x2(2) - x1(2))/(t2 - t1) - m1;
E(3) = x1(3) + (t - t1)*(x2(3) - x1(3))/(t2 - t1) - m2;
E(4) = x1(4) + (t - t1)*(x2(4) - x1(4))/(t2 - t1) - m3;
E(5) = x1(5) + (t - t1)*(x2(5) - x1(5))/(t2 - t1) - m4;

% Moment equations
G  = kg * (w - ws)^g;
B0 = kb * m3^km3 * G^kcg;

dm0dt =     B0 - m0*Qp/V + K(1)*E(4);
dm1dt =   G*m0 - m1*Qp/V + K(2)*E(5);
dm2dt = 2*G*m1 - m2*Qp/V + K(3)*E(3);
dm3dt = 3*G*m2 - m3*Qp/V + K(4)*E(4);
dm4dt = 4*G*m3 - m4*Qp/V + K(5)*E(5);

% Solute concentration balance
k1 = Hv*ws/(Hv-Hl)*(Rc/Rl-1+(Rl*Hl-Rc*Hc)/(Rl*Hv))-Rc/Rl;
k2 = ws/(V*Rl*(Hv-Hl));

dwdt = (Qp/V*(ws-w)+3*Kv*G*m2*(k1+w) + k2*Hin)/(1-Kv*m3);

dxdt=[dm0dt;dm1dt;dm2dt;dm3dt;dm4dt;dwdt];

end
```

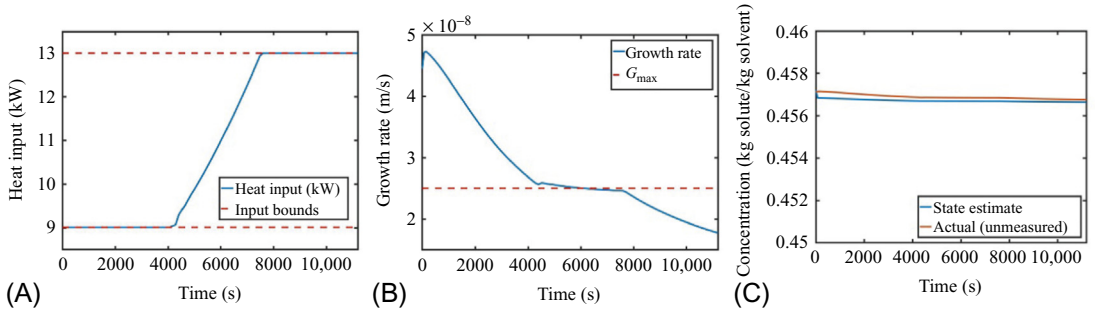

Fig. 9.10

The output-feedback NMPC of the batch crystallization process. The unmeasured solute concentration is estimated using the ELO. The ELO is able to effectively estimate the unmeasured solute concentration using the available process measurements, with a slight offset due to the plant-model mismatch in the underlying model of the ELO. (A) Heat input profile, (B) growth rate profile, and (C) a comparison between the (unmeasured) true and estimated profiles of the solute concentration.

parametric uncertainties and uncertainties in the initial process conditions. Despite the plant-model mismatch, updating the model predictions in the ELO based on the plant measurements enables obtaining accurate estimates of the state variables for the output-feedback implementation of the NMPC controller.

9.5 Advanced Process Control

In a chemical plant, MPC is typically incorporated into a hierarchical control scheme known as the advanced process control (APC)[55] scheme, which is illustrated in Fig. 9.11. The regulatory controllers, at the base of the APC hierarchy, are commonly made up of multi-loop PID controllers that provide improved stability and disturbance rejection. Constrained, multivariable optimization of the individual process units is performed in the MPC layer. As described earlier, MPC controllers predict the future outputs of a unit operation based on the current inputs and measurements and then determine the optimal control sequence to the unit. The first element of the optimal control sequence is applied to the process through the regulatory control layer. This implies that the set points of the PID controllers in the regulatory level are determined by the MPC controllers. In fact, the MPC layer above the regulatory layer enables optimal operation of the process close to the process constraints, typically where the process is the most profitable.

The production control layer (typically consisting of plant-wide optimization[56] and production planning and scheduling[57]) coordinates the operation of all of the process units across the plant. Running each unit at its local optimum may not lead to the most economical performance of the plant as a whole. The production control layer ensures that the plant operation meets the production demand, quality standards, plant utility constraints, and shipment requirements. The APC scheme shown in Fig. 9.11 offers a great advantage over traditional planning and scheduling approaches by streamlining of the entire control hierarchy into a single system.

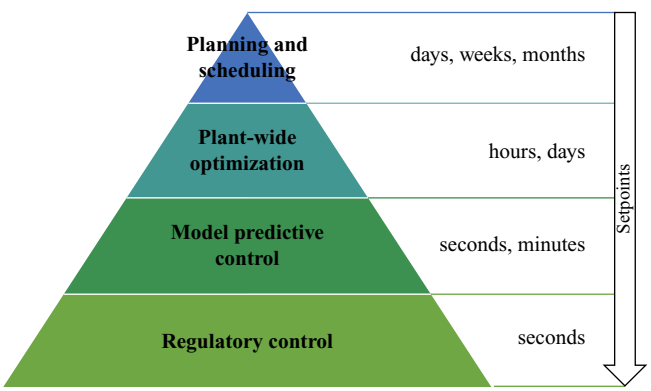

Fig. 9.11
Advanced process control scheme.[55]

9.6 Advanced Topics in MPC

9.6.1 Stability and Feasibility

Important theoretical properties of MPC consist of guaranteeing the feasibility and stability of the closed-loop control system. The feasibility requirement is to ensure that the OCP (Eq. 9.10) yields a feasible solution in light of the state constraints, whereas the stability requirement is to ensure that the optimal control inputs would not destabilize the control system. Generally speaking, conditions for establishing stability and feasibility are known for linear systems but often difficult to establish for nonlinear systems. Common approaches to enforcing closed-loop stability involve incorporating terminal equality constraints, terminal cost functions, and terminal constraint sets into the OCP. The addition of a terminal equality constraint forces the solution to the objective function to be a certain value at the end of the prediction horizon. When a terminal cost is used, an additional term (typically a quadratic function of the states) is added to the objective function. Although originally developed for linear systems, this method for establishing stability can be extended to nonlinear systems. A terminal constraint set is intended to drive the states to a constrained set within a finite length of time.

For further reading see Refs. 15,58–60.

9.6.2 MPC of Uncertain Systems

So far our discussion has focused on deterministic processes; however, all processes are subject to some form of uncertainty. These uncertainties may lead to degradation of the process model or cause constraint violation. Although the receding-horizon implementation of MPC improves the performance of the controller against uncertainties, it cannot systematically account for the process uncertainties. Robust MPC addresses uncertainties by utilizing a deterministic model with bounded, deterministic uncertainty descriptions. Robust MPC aims to ensure that certain process performance requirements and process constraints are met with respect to all possible process uncertainties. However, this may lead to conservative control. When the need for a process to remain within constraints is greater than that of achieving the performance objective, robust MPC offers an approach to guarantee that the process remains within the constraints. When probabilistic descriptions of uncertainties are available, a more natural approach to uncertainty handling will involve explicitly incorporating these probabilistic uncertainty descriptions into the OCP. Accounting for the probabilistic nature of uncertainties in the so-called stochastic MPC may allow for obtaining less conservative robust controllers.

For further reading see Refs. 61,62.

9.6.3 Distributed MPC

Complex, large-scale systems typically consist of a network of integrated subsystems that are connected to each other through mass, energy, and information streams. Examples of such large-scale processes include integrated chemical plants, power systems, and water distribution systems. Designing a centralized MPC for the system as a whole is typically impractical. However, complex systems can be divided into several subsystems, where each subsystem may only be coupled to few subsystems that share a few process variables in common. Distributed MPC involves designing MPC controllers for each individual subsystem such that some form of coordination takes place between the subsystems. Distributed control can address various practical and computational challenges associated with the MPC of large-scale processes.

For further reading see Refs. 63–65.

9.6.4 MPC With Integrated Model Adaptation

The performance of MPC controllers critically hinge on the quality of their underlying models. In most industrial process control applications, the inherent (gradual) time-varying nature of plant dynamics adversely impact the lifetime performance of MPC controllers. Changes in the plant dynamics over time increase the plant-model mismatch, which may eventually invalidate the model developed at the commissioning stage of a MPC controller. This necessitates regular maintenance of the process models and, consequently, the controller itself. An emerging area in the MPC of complex systems is MPC with integrated learning capability, where the control inputs must not only control the system dynamics in view of the control objectives but also gather informative process information for model adaptation. In MPC with integrated model adaptation, the OCP is modified to add some form of system perturbation capability to the control inputs. The addition of a process perturbation effect to the control inputs inevitably results in control performance loss due to the undesired system perturbations in the short run. However, the improved system learning facilitated by perturbations is expected to lead to better control performance over the future control stages and, as a result, decrease the overall performance loss that otherwise would be incurred due to large model uncertainty in the absence of learning.

For further reading see Refs. 66–69,69a,69b,69c.

9.6.5 Economic MPC

In chemical processes, achieving a desirable economic performance and control performance (e.g., in terms of set point tracking of a key process variable) may involve trade-offs. For an increasing number of processes, the hierarchical separation of MPC and (economic) plant-wide optimization is no longer optimal or desirable (see Fig. 9.11). An alternative to this hierarchical decomposition is to embed the economic objective of the process directly in the objective function of the MPC controller. In this approach, known as economic MPC, the controller optimizes directly the economic performance of the process in real time.

For further reading see Refs. 70–72.

Appendix

Batch Crystallization Case Study

Process model

```
function dxdt = crystallizer(t,x0,HI)

% State variables
m0 = x0(1);
m1 = x0(2);
m2 = x0(3);
m3 = x0(4);
m4 = x0(5);
w  = x0(6);

kg   = 7.49567e-05;
kb   = 1.02329e+14;

Hin = 0.001*HI; % [kW]
Qp  = 1.73120E-06; % Product flow (m3/s)

g   = 1;
kcg = 1;
km3 = 1;

Kv = 0.43;
V  = 75e-3; % [m^3]
Rc = 1767.3485; % [kg / m^3]
Rl = 1248.925; % [kg / m^3]
Hv = 2.590793151044623E+03; % [kJ / kg]
ws = 0.456513; % Solubility of AS at 50 degr. C. by Westhoff (2002) p. 159
Hl = 2.7945 * (50 - 25); % [kJ / kg]
Hc = 2.43   * (50 - 25); % [kJ / kg]
G  = kg * (w - ws)^g;
B0 = kb * m3^km3 * G^kcg;

% Moment equations
dm0dt =   B0   - m0*Qp/V;% + K1E(4);
dm1dt =   G*m0 - m1*Qp/V;% + K2*E(5);
dm2dt = 2*G*m1 - m2*Qp/V;% + K3*E(3);
dm3dt = 3*G*m2 - m3*Qp/V;% + K4*E(4);
dm4dt = 4*G*m3 - m4*Qp/V;% + K5*E(5);

% Solute concentration balance
k1 = Hv*ws/(Hv-Hl)*(Rc/Rl-1+(Rl*Hl-Rc*Hc)/(Rl*Hv))-Rc/Rl;
k2 = ws/(V*Rl*(Hv-Hl));

dwdt = (Qp/V*(ws-w)+3*Kv*G*m2*(k1+w) + k2*Hin)/(1-Kv*m3);

dxdt=[dm0dt;dm1dt;dm2dt;dm3dt;dm4dt;dwdt];

end
```

For full-state feedback, `crystallizer` and `crystallizer_model` have the same code. For the ELO case study:

```
kg  = 2*7.49567e-05;
kb  = 5*1.02329e+14;
```

in `crystallizer_model`, `objbatch`, and `ELO`.

Closed-loop MPC implementation with state estimator

```
ts = [0 70 144 214 288 352 426 494 584 656 728 838 960 1082 1194 1316...
       1438 1548 1666 1788 1908 2022 2142 2264 2382 2500 2616 2734 2852...
       2970 3092 3208 3328 3444 3562 3684 3800 3922 4036 4286 4406 4524...
       4642 4760 4878 4996 5116 5234 5354 5472 5590 5708 5826 5946 6064...
       6182 6300 6418 6538 6656 6774 6892 7010 7130 7248 7366 7486 7604...
       7722 7840 7960 8222 8340 8458 8696 8932 9050 9168 9288 9406 9524...
       9642 9762 9882 9998 10118 10236 10356 10474 10592 10712 10832...
       10948 11072 11186];

tb  = ts(end); % initial time horizon (s)
tp = tf;

x0(1,:) = [1.08457e10 342338 368.8758 0.0830 2.4860e-5 0.457106];
% initial values [m0 m1 m2 m3 m4 Cw]
xhat(1,:) = [1.08457e10 342338 368.8758 0.0830 2.4860e-5 0.4572];
% state estimates

x_plant(1,:) = x0; % plant measurements

ninter  = 3;
lb  = 9000*ones(1,ninter);
ub  = 13000*ones(1,ninter);

for k = 1:length(ts)-1,
    %Dynamic optimizer
    [uopt] = fmincon(@objbatch,uopt_past,[],[],[],[],...
        lb,ub,@conbatch, [], xhat(k,:),tp,ninter);
    % Adjusting the prediction horizon for shrinking-horizon implementation of
      the OCP
    tp = tb - ts(k+1);
    %Plant
    [t,x] = ode15s(@crystallizer,[ts(k) ts(k+1)],x_plant(k,:),[],uopt(1));
    uopt_past = uopt;
    x_plant(k+1,:) = x(end,:);
    % State  estimator
    [t,xobs]  = ode15s(@ELO,[ts(k) ts(k+1)],xhat(k,:),[],uopt(1),...
                x_plant(k,1:5), x_plant(k+1,1:5),ts(k),ts(k+1));
    xhat(k+1,:) = xobs(end,:);
end
```

Constraints function

```
function [conin,coneq] = conbatch(u,x0,tf,ninter)
conin = [];
coneq = [];
```

Extended Luenberger observer

```
function dxdt = ELO(t,x0,HI,x1,x2,t1,t2)

% State variables
m0 = x0(1);
m1 = x0(2);
m2 = x0(3);
m3 = x0(4);
m4 = x0(5);
w  = x0(6);

kg   = 2*7.49567e-05;
kb   = 5*1.02329e+14;

Hin = 0.001*HI; % [kW]
Qp  = 1.73120E-06; % Product flow (m3/s)

g   = 1;
kcg = 1;
km3 = 1;

Kv = 0.43;
V  = 75e-3; % [m^3]
Rc = 1767.3485; % [kg / m^3]
Rl = 1248.925; % [kg / m^3]
Hv = 2.590793151044623E+03; % [kJ / kg]
ws = 0.456513; % Solubility of AS at 50 degr. C. by Westhoff (2002) p. 159
Hl = 2.7945 * (50 - 25); % [kJ / kg]
Hc = 2.43   * (50 - 25); % [kJ / kg]
G  = kg * (w - ws)^g;
B0 = kb * m3^km3 * G^kcg;

K = [-0.00272458 84.5034 0.117009 0.441494 0.463708];

% Error signals
E(1) = x1(1) + (t - t1)*(x2(1) - x1(1))/(t2 - t1) - m0;
E(2) = x1(2) + (t - t1)*(x2(2) - x1(2))/(t2 - t1) - m1;
E(3) = x1(3) + (t - t1)*(x2(3) - x1(3))/(t2 - t1) - m2;
E(4) = x1(4) + (t - t1)*(x2(4) - x1(4))/(t2 - t1) - m3;
E(5) = x1(5) + (t - t1)*(x2(5) - x1(5))/(t2 - t1) - m4;

% Moment equations
dm0dt =   B0   - m0*Qp/V + K(1)*E(3);
dm1dt =   G*m0 - m1*Qp/V + K(2)*E(4);
dm2dt = 2*G*m1 - m2*Qp/V + K(3)*E(3);
dm3dt = 3*G*m2 - m3*Qp/V + K(4)*E(4);
dm4dt = 4*G*m3 - m4*Qp/V + K(5)*E(5);

% Solute concentration balance
k1 = Hv*ws/(Hv-Hl)*(Rc/Rl-1+(Rl*Hl-Rc*Hc)/(Rl*Hv))-Rc/Rl;
k2 = ws/(V*Rl*(Hv-Hl));

dwdt = (Qp/V*(ws-w)+3*Kv*G*m2*(k1+w) + k2*Hin)/(1-Kv*m3);

dxdt=[dm0dt;dm1dt;dm2dt;dm3dt;dm4dt;dwdt];

end
```

ABE Fermentation Case Study

Process model

```
function dxdt = ABE_model(t,x,u)

% Variables:
AC = x(1); %acetyl-CoA
A = x(2); %acetate
En = x(3); %ethanol
AaC = x(4); %acetoacetyl-CoA
Aa = x(5); %acetoacetate
BC = x(6); %butyryl-CoA
B = x(7); %butyrate
An = x(8); %acetone
Bn = x(9); %butanol
Ad = x(10); %adc
Cf = x(11); %ctfA/B
Ah = x(12); %adhE

%Kinetic parameters

V1 = 4.94;
V2 = 2.92;
V4 = 45.6;
V8 = 64.8;
V10 = 4.75;

K1 = 0.00158;
K2 = 0.00181;
K4 = 1.87;
K8 = 7.92E-6;
K10 = 1.40E-5;

a3 = 0.00517;
a5 = 0.0140;
a6 = 0.00537;
a7 = 4790;
a9 = 347000;

rad = 0.00547;
rcf = 0.000324;
rah = 0.289;

radp = 0.104;
rcfp = 1.06;
rahp = 2.56;

n = 485;
pstar = 4.50; %pH after switch

%inputs
D = u(1); %Dilution rate
G = u(2); %Glucose conc

p = 4.5; %keep pH constant

F = 1-tanh(n*(p-pstar)); %pH switch for enzymes
```

```
%Reactions
r1 = 2*V1*G/(K1+G);
r2 = V2*AC/(K2+AC);
r3 = a3*A*AaC*Cf;
r4 = V4*AC/(2*(K4+AC));
r5 = a5*AC*Ah;
r6 = a6*B*AaC*Cf;
r7 = a7*Aa*Ad;
r8 = V8*BC/(K8 + BC);
r9 = a9*BC*Ah;
r10 = V10*AaC/(K10+AaC);

%Species mass balances
dxdt(1) = r1 - r2 + r3 -r4 -r5 - D*AC;
dxdt(2) = r2 - r3 - D*A;
dxdt(3) = r5 - D*En; %ethanol
dxdt(4) = r4 - r3 - r6 - r10 - D*AaC;
dxdt(5) = r3 + r6 - r7 - D*Aa;
dxdt(6) = r10 - r8 + r6 - r9 - D*BC;
dxdt(7) = r8 - r6 - D*B;
dxdt(8) = r7 - D*An; %acetone
dxdt(9) = r9 - D*Bn; %butanol
dxdt(10) = rad + radp*F-D*Ad;
dxdt(11) = rcf + rcfp*F - D*Cf;
dxdt(12) = rah + rahp*F - D*Ah;

dxdt = dxdt';

end
```

Closed-loop MPC implementation with state constraints

```
Np = 60; %prediction horizon in hrs
meas_samp_time = 0:1/60:1/3; %measurement time in hrs (1/3 hr = 20 min)

%input bounds
lb =[0.005, 0]; % lower bound (hr^-1, mM)
ub = [0.145, 80]; % upper bound (hr^-1, mM)

%inputs, u =[D, G0]
u_opt(1,:) = [0.075 40];

%outputs, y = [AC, A, En, AaC, Aa, BC, B, An, Bn, Ad, Cf, Ah]
y0 = [0 13.73 3.02 0 0 0 8.81 1.82 0 0 0 0]; %initial condition

[t,x0]=ode15s(@ABE_model,0:1:150,y0,[],u(1,:));

y(1,:) = x0(end,:);

SP = 1.1*y(1,9);

global A; global B;
A=y(1,2); B=y(1,7);
```

```
for k = 1:(3*Np)
    % Optimizer
    u_opt(k+1,:)= fmincon(@objfun,u_opt(k,:),[], [], [], [],lb,ub,...
        @constraints,[],y(k,:),SP,Np);
    % Plant
    [t,x]=ode15s(@ABE_model,meas_samp_time,y(k,:),[],u_opt(k,:));
    % New measurements to reinitialize the optimizer at the next sampling time
    y(k+1,:)=x(end,:);
end

SP2 = 0.8*y(k,9);

for k = (3*Np)+1:(3*2*Np)
    % Optimizer
    u_opt(k+1,:)= fmincon(@objfun,u_opt(k,:),[],[],[],[],lb,ub,...
        @constraints,[],y(k,:),SP2,Np);
    % Plant
    [t,x]=ode15s(@ABE_model,meas_samp_time,y(k,:),[],u_opt(k,:));
    % New measurements to reinitialize the optimizer at the next sampling time
    y(k+1,:)=x(end,:);
end
```

Acknowledgments

Washington S. Reeder is acknowledged for his work on the ABE fermentation simulation case study.

References

1. Bennett S. A brief history of automatic control. *IEEE Control Syst* 1996;**16**(3):17–25.
2. Åström J, Hägglund T. *PID controllers: theory, design, and tuning research triangle park*. Raleigh, NC: Instrument Society of America; 1995.
3. Skogestad S, Postlethwaite I. *Multivariable feedback control: analysis and design*. 2nd ed. Southern Gate: John Wiley & Sons; 2005.
4. Ang KH, Chong G, Li Y. PID control system analysis, design and technology. *IEEE Trans Control Syst Technol* 2005;**13**(4):559–76.
5. Qin SJ, Badgwell TA. A survey of industrial model predictive control technology. *Control Eng Pract* 2003;**11**(7):733–64.
6. Cutler CR, Ramaker BL. Dynamic matrix control—a computer control algorithm. In: AIChE national meeting, Houston, TX; 1979.
7. Cutler C, Morshedi A, Haydel J. An industrial perspective on advanced control. In: AIChE national meeting, Washington, DC; 1983.
8. Basak K, Abhilash KS, Ganguly S, Saraf DN. On-line optimization of a crude distillation unit with constraints on product properties. *Ind Eng Chem Res* 2002;**41**(6):1557–68.
9. Richalet J. Industrial applications of model based predictive control. *Automatica* 1993;**29**(5):1251–74.
10. Garcia CE, Prett DM, Morari M. Model predictive control: theory and practice—a survey. *Automatica* 1989;**25**(3):335–48.
11. Richalet J, Rault A, Testud JL, Papon J. Algorithmic control of industrial processes. In: Proceedings of the 4th IFAC symposium on identification and system parameter estimation, Amsterdam; 1976. p. 1119–67.
12. Morari M, Lee JH. Model predictive control: past, present and future. *Comput Chem Eng* 1999;**23**(4):667–82.

13. Gidon D, Graves DB, Mesbah A. Model predictive control of thermal effects of an atmospheric plasma jet for biomedical applications. In: Proceedings of the American Control conference, Boston; 2016. p. 4889–94.
14. Schütze A, Jeong JY, Babyan SE, Park J, Selwyn GS, Hicks RF. The atmospheric-pressure plasma jet: a review and comparison to other plasma sources. *IEEE Trans Plasma Sci* 1998;**26**(6):1685–94.
15. Rawlings JB, Mayne DQ. *Model predictive control: theory and design*. Madison, WI: Nob Hill Publishing; 2015.
16. Allgöwer F, Findeisen R, Nagy ZK. Nonlinear model predictive control: from theory to application. *J Chin Inst Chem Eng* 2004;**35**(3):299–315.
17. Ljung L. *System identification: theory for the user*. Upper Saddle River, NJ: Prentice Hall PTR; 1999.
17a. Van Overschee P, De Moor BL. *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media; 2012.
18. Bequette BW. *Process control: modelling, design and simulation*. Upper Saddle River, NJ: Prentice Hall PTR; 2002.
19. Thomas MM, Kardos JL, Joseph B. Shrinking horizon model predictive control applied to autoclave curing of composite laminate materials. In: Proceedings of the American Control conference, Baltimore, MD; 1994. p. 505–9.
20. Biegler LT, Cuthrell JE. Improved infeasible path optimization for sequential modular simulators—II: the optimization algorithm. *Comput Chem Eng* 1985;**9**(3):257–67.
21. Cuthrell JE, Biegler LT. Simultaneous optimization and solution methods for batch reactor control profiles. *Comput Chem Eng* 1989;**13**(1):49–62.
22. Bock HG, Plitt KJ. A multiple shooting algorithm for direct solution of optimal control problems. In: Proceedings of the 9th IFAC World Congress, Budapest, Hungary; 1984. p. 242–7.
23. Diehl M, Bock HG, Schlöder JP, Findeisen R, Nagy Z, Allgöwer F. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *J Process Control* 2002;**12**(4):577–85.
24. Binder T, Blank L, Bock HG, Bulirsch R, Dahmen W, Diehl M, et al. Introduction to model based optimization of chemical processes on moving horizons. In: *Online optimization of large scale systems*. Berlin, Heidelberg: Springer; 2001. p. 295–339.
25. Randolph A, Larson M. *Theory of particulate processes*. New York, NY: Academic Press; 1971.
26. Myerson AS. *Handbook of industrial crystallization*. Woburn, MA: Butterworth-Heinemann; 2002.
27. Nagy ZK, Braatz RD. Advances and new directions in crystallization control. *Annu Rev Chem Biomol Eng* 2012;**3**:55–75.
28. Hulbert HM, Katz S. Some problems in particle technology: a statistical mechanical formulation. *Chem Eng Sci* 1964;**19**(8):555–74.
29. Mesbah A, Kramer HJ, Huesman AE, Van den Hof PM. A control oriented study on the numerical solution of the population balance equation for crystallization processes. *Chem Eng Sci* 2009;**64**(20):4262–77.
30. Mesbah A, Nagy ZK, Huesman AE, Kramer HJ, Van den Hof PM. Nonlienar model-based control of a semi-industrial batch crystallizer using a population balance modeling framework. *IEEE Trans Control Syst Technol* 2012;**20**(5):1188–201.
31. Mesbah A, Landlust J, Huesman AE, Kramer HJ, Jansens PJ, Van den Hof PM. A model-based control framework for industrial batch crystallization processes. *Chem Eng Res Des* 2010;**88**(9):1223–33.
32. MathWorks. *fmincon*. Natick, MA: The Mathworks; 2016.
33. Kendall JW. Hard and soft constraints in linear programming. *Omega* 1975;**3**(6):709–15.
34. Dürre P. Biobutanol: an attractive biofuel. *Biotechnol J* 2007;**2**(12):1525–34.
35. Köpke M, Dürre P. The past, present, and future of biolfuels—biobutanol as promising alternative. In: Marco Aurelio Dos Santos Bernardes, editor. *Biofuel Production-Recent Developments and Prospects*. In Tech; 2011. pp. 451–86. http://dx.doi.org/10.5772/20113. Available from: https://www.intechopen.com/books/biofuel-production-recent-developments-and-prospects/the-past-present-and-future-of-biofuels-biobutanol-as-promising-alternative.
36. Jones DT, Woods DR. Acetone-butanol fermentation revisited. *Microbiol Rev* 1986;**50**(4):484–524.

37. Haus S, Jabbari S, Millat T, Janssen H, Fischer RJ, Bahl H, et al. A systems biology approach to investigate the effect of pH-induced gene regulation on solvent production by Clostridium acetobutylicum in continuous culture. *BMC Syst Biol* 2011;**5**(10).
38. Buehler EA, Mesbah A. Kinetic study of acetone-butanol-ethanol fermentation in continuous culture. *Public Libr Sci* 2016;**11**(8):1–21.
39. Soroush M. State and parameter estimations and their applications in process control. *Comput Chem Eng* 1998;**23**(2):229–45.
40. Dochain D. State and paramter estimation in chemical and biochemical processes: a tutorial. *J Process Control* 2003;**13**(8):801–18.
41. Soroush M. Nonlinear state-observer design with application to reactors. *Chem Eng Sci* 1997;**52**(3):387–404.
42. Patwardhan SC, Narasimhan S, Jagadeesan P, Gopaluni B, Shah SL. Nonlinear Bayesian state estimation: a review of recent developments. *Control Eng Pract* 2012;**20**(10):933–53.
43. Luenberger D. Observing the state of a linear system. *IEEE Trans Mil Electron* 1964;**8**(2):74–80.
44. Zeitz M. Extended Luenberger observer for nonlinear systems. *Syst Control Lett* 1987;**9**(2):149–56.
45. Kalman RE. A new approach to linear filtering and prediction problems. *J Basic Eng* 1960;**82**(1):35–45.
46. Kalman RE, Bucy RS. New results in linear filtering and prediction theory. *J Basic Eng* 1961;**83**(1):95–108.
47. Wan EA, Van Der Merwe R. The unscented Kalman filter for nonlinear estimation. In: Adaptive systems for signal processing, communications, and control symposium, Alberta, Canada; 2000. p. 153–8.
48. Arulampalam MS, Maskell S, Cordon N, Clapp T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans Signal Process* 2002;**50**(2):174–88.
49. Crisan D, Doucet A. Survey of convergence results on particle filtering methods for practitioners. *IEEE Trans Signal Process* 2002;**50**(3):736–46.
50. Rawlings JB, Bakshi BR. Particle filtering and moving horizon estimation. *Comput Chem Eng* 2006;**30**(10):1529–41.
51. Rao CV, Rawlings JB, Mayne DQ. Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations. *IEEE Trans Autom Control* 2003;**48**(2):246–58.
52. Quintero-Marmol E, Luyben WL, Georgakis C. Application of an extended Luenberger observer to the control of multicomponent batch distillation. *Ind Eng Chem Res* 1991;**30**(8):1870–80.
53. Hulhoven X, Wouwever AV, Bogaerts P. Hybrid extended Luenberger-asymptotic observer for bioprocess state estimation. *Chem Eng Sci* 2006;**61**(21):7151–60.
54. Kazantzis N, Kravaris C, Wright RA. Nonlinear observer design for process monitoring. *Ind Eng Chem Res* 2000;**39**(2):408–19.
55. Robinson PR, Cima D. Advanced process control. In: *Practical advances in petroleum processing*. New York, NY: Springer; 2006. p. 695–703.
56. Tatjewski P. Advanced control and on-line optimization in multilayer structures. *Annu Rev Control* 2008;**32**(1):71–85.
57. Backx T, Bosgra O, Marquardt W. Integration of model predictive control and optimization of processes. In: IFAC symposium on advanced control of chemical processes, Pisa, Italy; 2000. p. 249–60.
58. Mayne DQ, Rawlings JB, Rao CV, Scokaert PO. Constrained model predictive control: stability and optimality. *Automatica* 2000;**36**(6):789–814.
59. Camacho EF, Bordons C. *Model predictive control*. London: Springer-Verlag; 1999.
60. Findeisen R, Imsland L, Allgöwer F, Foss BA. State and output feedback nonlinear model predictive control: an overview. *Eur J Control* 2003;**9**(2–3):190–206.
61. Bemporad A, Morari M. *Robust model predictive control: a survey. Robustness in identifiction and control*. London: Springer; 1999207–26.
62. Mesbah A. Stochastic model predictive control: an overview and perspectives for future research. *IEEE Control Syst Mag* 2016;**36**:30–44.
63. Stewart BT, Venkat AN, Rawlings JB, Wright SJ, Pannocchia G. Cooperative distributed model predictive control. *Syst Control Lett* 2010;**59**(8):460–9.

64. Venkat AN, Rawlings JB, Wright SJ. Stability and optimality of distributed model predictive control. In: Proceedings of the IEEE conference on decision and control, Seville, Spain; 2005. p. 6680–5.
65. Venkat AN, Rawlings JB, Wright SJ. Distributed model predictive control of large-scale systems. In: *Assessment and future directions of nonlinear model predictive control*. Berlin, Heidelberg: Springer; 2007. p. 591–605.
66. Larsson CA, Annergren M, Hjalmarsson H, Rojas CR, Bombois X, Mesbah A, et al. Model predictive control with integrated experiment design for output error systems. In: Proceedings of the European Control conference, Zurich, Switzerland; 2013. p. 3790–5.
67. Larsson CA, Rojas CR, Bombois X, Hjalmarsson H. Experimental evaluation of model predictive control with excitation (MPC-X) on an industrial depropanizer. *J Process Control* 2015;**31**:1–16.
68. Heirung TA, Foss B, Ydstie BE. MPC-based dual control with online experiment design. *J Process Control* 2015;**32**:64–76.
69. Marafioti G, Bitmead RR, Hovd M. Persistently exciting model predictive control. *Int J Adapt Control Signal Process* 2013;**28**:536–52.
69a. Bavdekar VA, Ehlinger V, Gidon D, Mesbah A. Stochastic predictive control with adaptive model maintenance. In: Proceedings of the 55th IEEE Conference on Decision and Control, Las Vegas; 2016. pp. 2745–50.
69b. Bavdekar V, Mesbah A. Stochastic model predictive control with integrated experiment design for nonlinear systems. In: Proceedings of the 11th IFAC Symposium on Dynamics and Control of Process Systems (DYCOPS), Trondheim; 2016. pp. 49–54.
69c. Heirung TAN, Mesbah A. Stochastic nonlinear model predictive control with active model discrimination for online fault detection. In: *Proceedings of the IFAC World Congress*, Toulouse; 2017. pp. 49–54, To Appear.
70. Ellis M, Durand H, Christofides PD. A tutorial review of economic model predictive control methods. *J Process Control* 2014;**24**(8):1156–78.
71. Ellis M, Liu J, Christofides PD. *Economic model predictive control: theory formulations and chemical process applications*. Switzerland: Springer International Publishing; 2017.
72. Rawlings JB, Angeli D, Bates CN. Fundamentals of economic model predictive control. In: IEEE conference on decision and control, Maui, HI; 2012. p. 3851–61.