Overall Objectives of Model Predictive Control

1. Prevent violations of input and output constraints.
2. Drive some output variables to their optimal set points, while maintaining other outputs within specified ranges.
3. Prevent excessive movement of the input variables.
4. If a sensor or actuator is not available, control as much of the process as possible.

Model Predictive Control: Basic Concepts

1. Future values of output variables are predicted using a dynamic model of the process and current measurements.
   - Unlike time delay compensation methods, the predictions are made for more than one time delay ahead.
2. The control calculations are based on both future predictions and current measurements.
3. The manipulated variables, $\boldsymbol{u}(k)$, at the k-th sampling instant are calculated so that they minimize an objective function, J.
   - **Example:** Minimize the sum of the squares of the deviations between predicted future outputs and specific reference trajectory.
   - The reference trajectory is based on set points calculated using RTO.
4. Inequality & equality constraints, and measured disturbances are included in the control calculations.
5. The calculated manipulated variables are implemented as set point for lower level control loops. (cf. cascade control).

Model Predictive Control: Calculations

1. At the *k*-th sampling instant, the values of the manipulated variables, ***u,*** at the next *M* sampling instants, {***u***(k), ***u***(k+1), …, ***u***(*k*+*M* -1)} are calculated.
   - This set of *M* "control moves" is calculated so as to minimize the predicted deviations from the reference trajectory over the next *P* sampling instants while satisfying the constraints.
   - Typically, an LP or QP problem is solved at each sampling instant.
   - Terminology: *M* = control horizon, *P* = prediction horizon
2. Then the first "control move", ***u***(*k*), is implemented.
3. At the next sampling instant, *k*+1, the *M*-step control policy is re-calculated for the next *M* sampling instants, *k*+1 to *k*+*M*, and implement the first control move, ***u***(*k*+1).
4. Then Steps 1 and 2 are repeated for subsequent sampling instants.

Note: This approach is an example of a *receding horizon approach.*

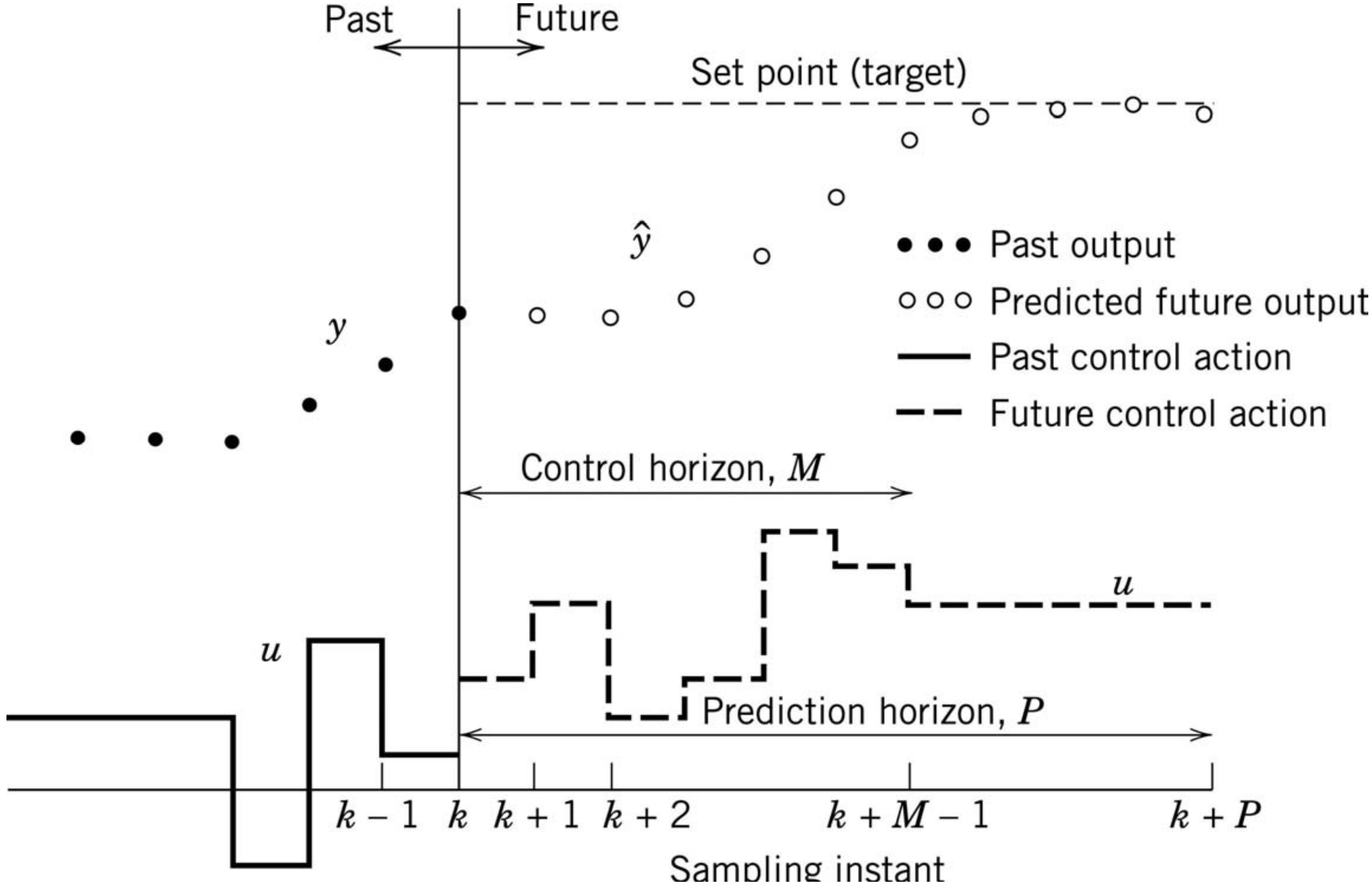


Figure 20.2 Basic concept for Model Predictive Control

When Should Predictive Control be Used?

1. Processes are difficult to control with standard PID algorithm (e.g., large time constants, substantial time delays, inverse response, etc.
2. There is significant process interactions between ***u*** and ***y***.
   - i.e., more than one manipulated variable has a significant effect on an important process variable.
3. Constraints (limits) on process variables and manipulated variables are important for normal control.

Terminology:

- ***y*** ↔ *CV*, ***u*** ↔ *MV*, ***d*** ↔ *DV*

Chapter 20

Model Predictive Control Originated in 1980s

- Techniques developed by industry:

1. ***Dynamic Matrix Control (DMC)***
    - Shell Development Co.: Cutler and Ramaker (1980),
    - Cutler later formed DMC, Inc.
    - DMC acquired by Aspentech in 1997.

2. ***Model Algorithmic Control (MAC)***
    - ADERSA/GERBIOS, Richalet et al. (1978) in France.

- Over 5000 applications of MPC since 1980

Reference: Qin and Badgwell, 1998 and 2003).

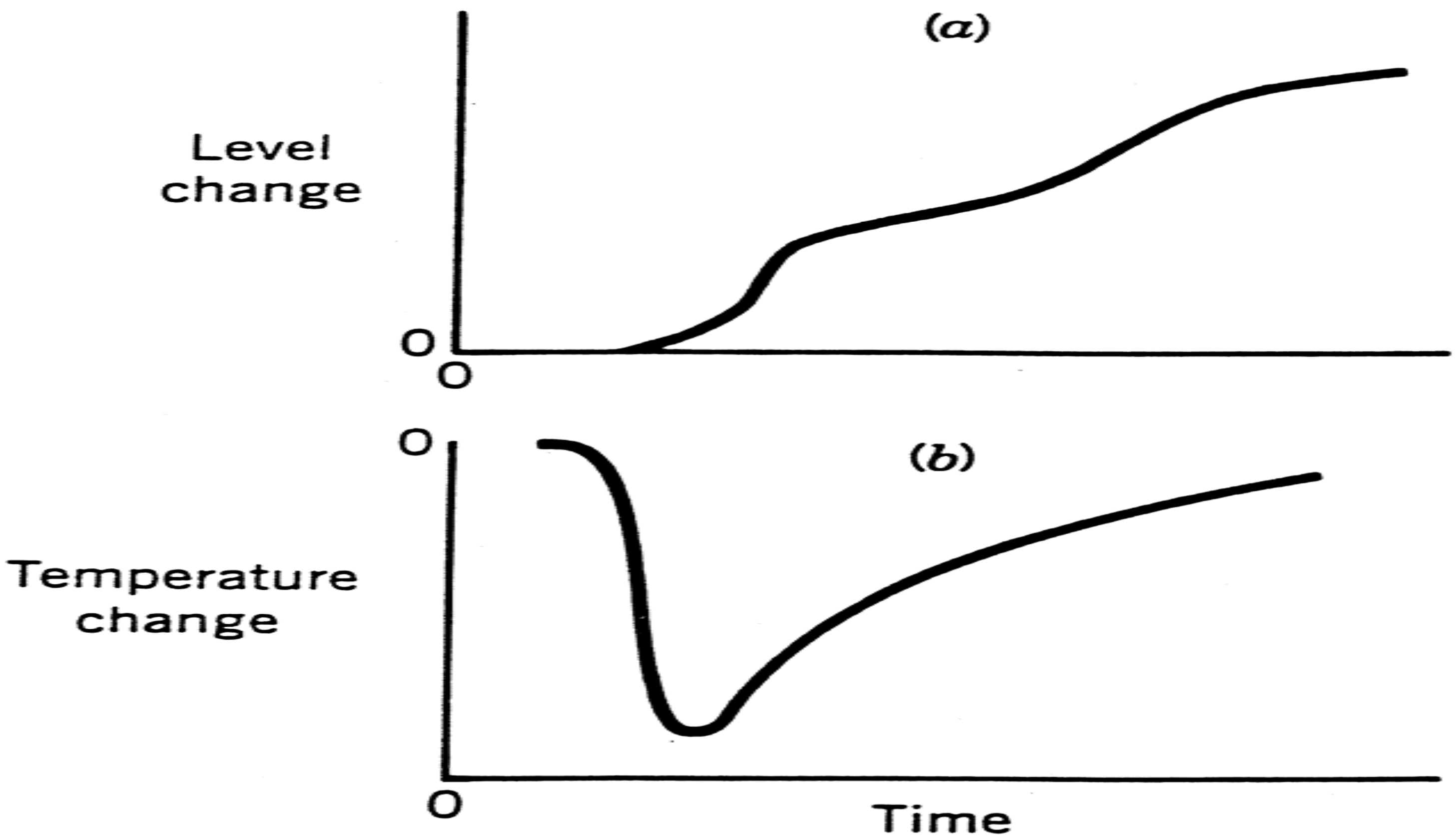


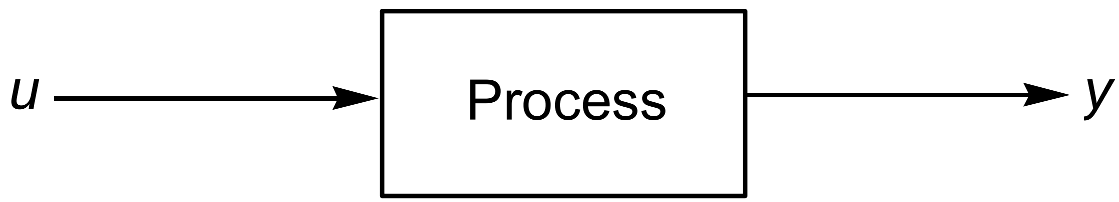
Figure A. Two processes exhibiting unusual dynamic behavior.
(a) change in base level due to a step change in feed rate to a distillation column.
(b) steam temperature change due to switching on soot blower in a boiler.

Dynamic Models for Model Predictive Control

- **Could be either:**
  1. Physical or empirical (but usually empirical)
  2. Linear or nonlinear (but usually linear)
- **Typical linear models used in MPC:**
  1. Step response models
  2. Transfer function models
  3. State-space models
- **Note: Can convert one type of linear model (above) to the other types.**

Discrete Step Response Models

Consider a single input, single output process:



where *u* and *y* are deviation variables (i.e., deviations from nominal steady-state values).

Chapter 20

Prediction for SISO Models:

Example: Step response model

$$y(k+1) = y_0 + \sum_{i=1}^{N-1} S_i \Delta u(k-i+1) + S_N u(k-N+1) \qquad (20\text{-}1)$$

S_i = the i-th step response coefficient

N = an integer (the *model horizon)*

y_0 = initial value at k=0

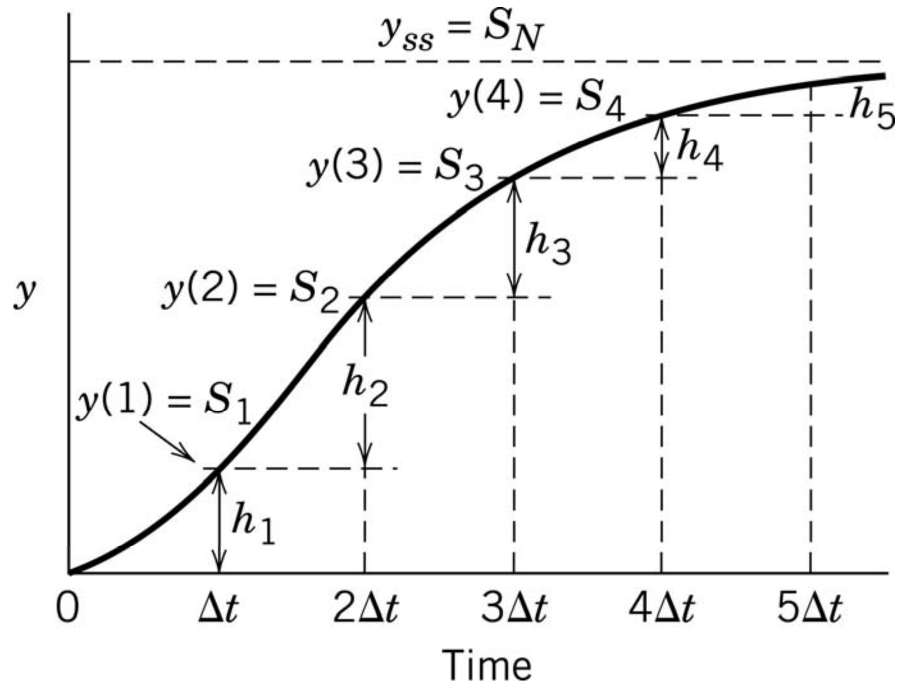


Figure 7.14. Unit Step Response

Chapter 20

Prediction for SISO Models:

Example: Step response model

$$y(k+1) = y_0 + \sum_{i=1}^{N-1} S_i \Delta u(k-i+1) + S_N\, u(k-N+1) \qquad \text{(20-1)}$$

- If y_0=0, this one-step-ahead prediction can be obtained from Eq. (20-1) by replacing $y(k+1)$ with $\hat{y}(k+1)$

$$\hat{y}(k+1) = \sum_{i=1}^{N-1} S_i \Delta u(k-i+1) + S_N\, u(k-N+1) \qquad (20-6)$$

- Equation (20-6) can be expanded as:

$$\hat{y}(k+1) = \underbrace{S_1 \Delta u(k)}_{\substack{\text{Effect of current}\\ \text{control action}}} + \underbrace{\sum_{i=2}^{N-1} S_i \Delta u(k-i+1) + S_N\, u(k-N+1)}_{\text{Effect of past control actions}}$$

Prediction for SISO Models: (continued)

Similarly, the j-th step ahead prediction is Eq. 20-10:

$$\hat{y}(k+j) = \underbrace{\sum_{i=1}^{j} S_i \Delta u(k+j-i)}_{\text{Effects of current and future control actions}} + \underbrace{\sum_{i=j+1}^{N-1} S_i \Delta u(k+j-i) + S_N u(k+j-N)}_{\text{Effects of past control actions}}$$

Define the predicted unforced response as:

$$\hat{y}^o(k+j) \triangleq \sum_{i=j+1}^{N-1} S_i \Delta u(k+j-i) + S_N u(k+j-N) \qquad (20-11)$$

and can write Eq. (20-10) as:

$$\hat{y}(k+j) = \sum_{i=1}^{j} S_i \Delta u(k+j-i) + \hat{y}^o(k+j) \qquad (20-12)$$

Vector Notation for Predictions

Define vectors:

$$\hat{\boldsymbol{Y}}(k+1) \triangleq col\,[\,\hat{y}(k+1), \hat{y}(k+2), \ldots, \hat{y}(k+P)] \qquad (20\text{-}16)$$

$$\hat{\boldsymbol{Y}}^{\boldsymbol{o}}(k+1) \triangleq col\,[\,\hat{y}^{o}(k+1), \hat{y}^{o}(k+2), \ldots, \hat{y}^{o}(k+P)] \qquad (20\text{-}17)$$

$$\Delta \boldsymbol{U}(k) \triangleq col\,[\,\Delta u(k), \Delta u(k+1), \cdots, \Delta u(k+M-1)] \qquad (20\text{-}18)$$

The model predictions in Eq. (20-12) can be written as:

$$\hat{\boldsymbol{Y}}(k+1) = \boldsymbol{S}\Delta \boldsymbol{U}(k) + \hat{\boldsymbol{Y}}^{\boldsymbol{o}}(k+1) \qquad (20\text{-}19)$$

Chapter 20

Dynamic Matrix Model

The model predictions in Eq. (20-12) can be written as:

$$\hat{\mathbf{Y}}(k+1) = \mathbf{S}\Delta\mathbf{U}(k) + \hat{\mathbf{Y}}^{o}(k+1) \qquad (20\text{-}19)$$

where $\mathbf{S}$ is the *P x M dynamic matrix:*

$$\mathbf{S} \triangleq \begin{bmatrix} S_1 & 0 & \cdots & 0 \\ S_2 & S_1 & 0 & \vdots \\ \vdots & \vdots & \ddots & 0 \\ S_M & S_{M-1} & \cdots & S_1 \\ S_{M+1} & S_M & \cdots & S_2 \\ \vdots & \vdots & \ddots & \vdots \\ S_P & S_{P-1} & \cdots & S_{P-M+1} \end{bmatrix} \qquad (20\text{-}20)$$

Chapter 20

Bias Correction

- The model predictions can be corrected by utilizing the latest measurement, $y(k)$.

- The *corrected prediction* is defined to be:

$$\tilde{y}(k+j) \triangleq \hat{y}(k+j) + [y(k) - \hat{y}(k)] \qquad (20\text{-}23)$$

- Similarly, adding this bias correction to each prediction in (20-19) gives:

$$\tilde{\mathbf{Y}}(k+1) = \mathbf{S}\,\Delta\mathbf{U}(k) + \hat{\mathbf{Y}}^{o}(k+1) + [y(k) - \hat{y}(k)]\,\mathbf{1} \qquad (20\text{-}24)$$

where $\tilde{\mathbf{Y}}(k+1)$ is defined as:

$$\tilde{\mathbf{Y}}(k+1) \triangleq col[\tilde{y}(k+1), \tilde{y}(k+2), \ldots, \tilde{y}(k+P)] \qquad (20\text{-}25)$$

EXAMPLE 20.4

The benefits of using corrected predictions will be illustrated by a simple example, the first-order plus-time-delay model of Example 20.1:

$$\frac{Y(s)}{U(s)} = \frac{5e^{-2s}}{15s+1} \qquad (20\text{-}26)$$

Assume that the disturbance transfer function is identical to the process transfer function, $G_d(s)=G_p(s)$. A unit step change in u occurs at time t=2 min and a step disturbance, d=0.15, occurs at t=8 min. The sampling period is Δt= 1 min.

(a) Compare the process response $y(k)$ with the predictions that were made 15 steps earlier based on a step response model with N=80. Consider both the corrected prediction

(b) Repeat part (**a**) for the situation where the step response coefficients are calculated using an incorrect model:

$$\frac{Y(s)}{U(s)} = \frac{4e^{-2s}}{20s+1} \qquad (20\text{-}27)$$

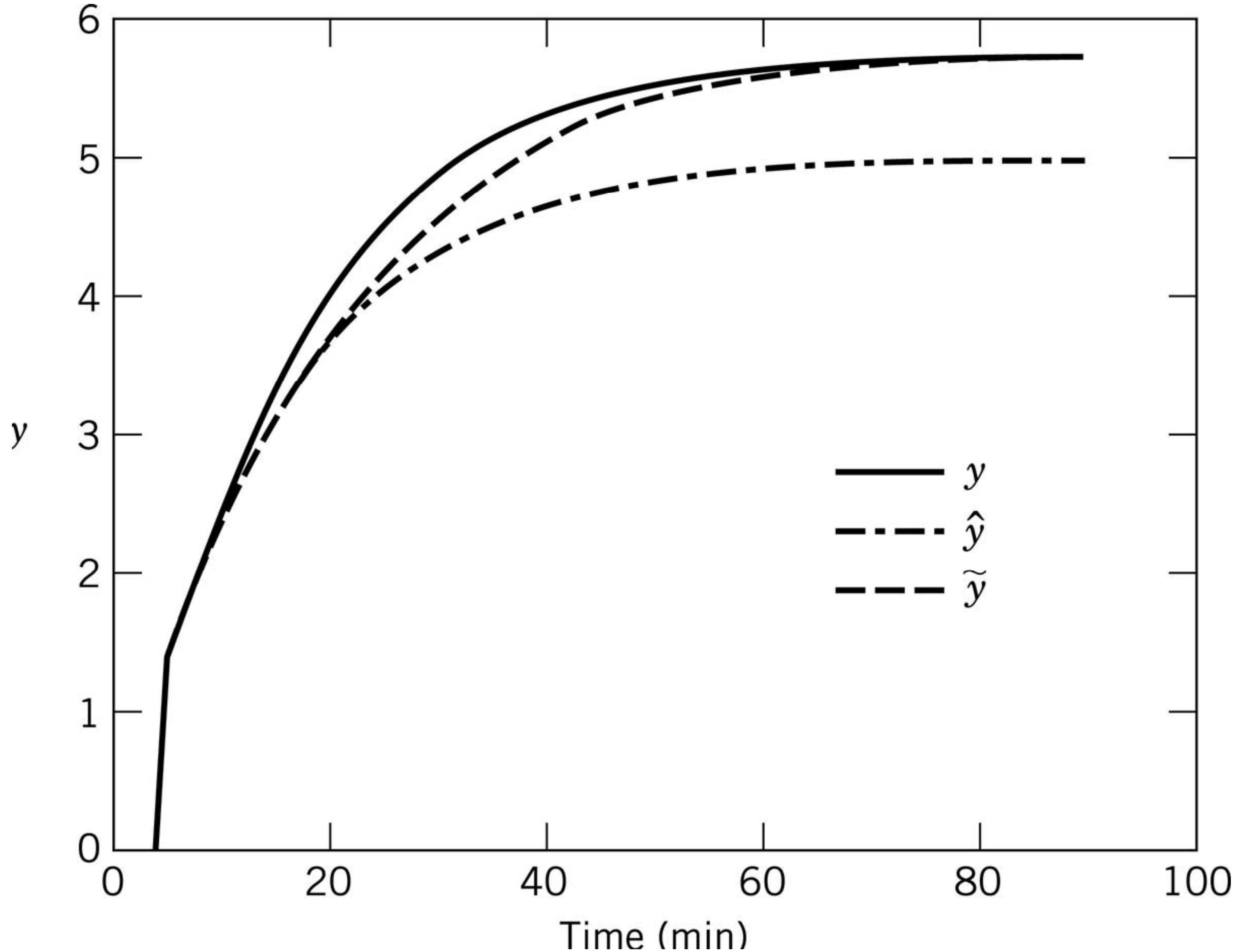


Figure 20.6 Without model error.

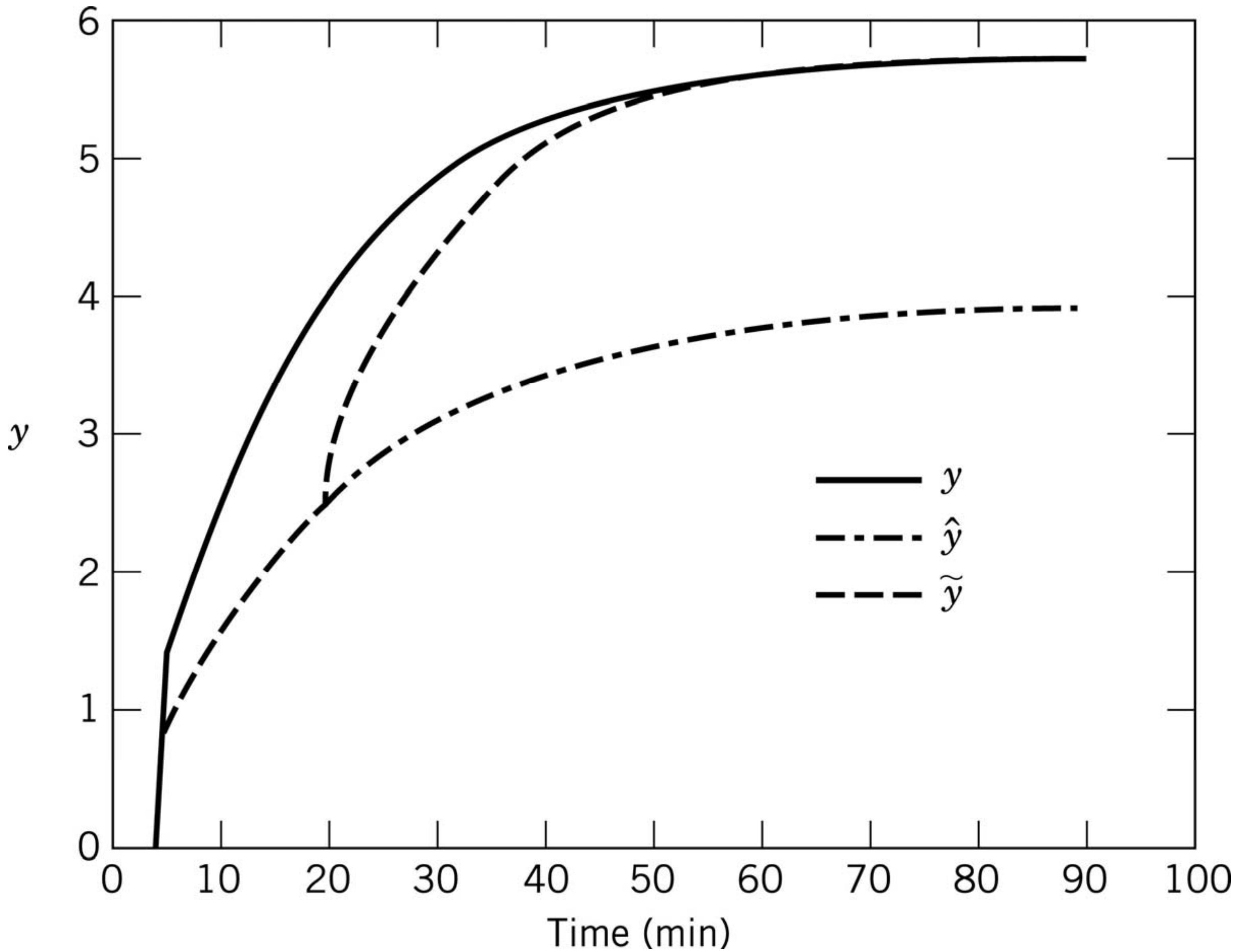


Figure 20.7 With model error.

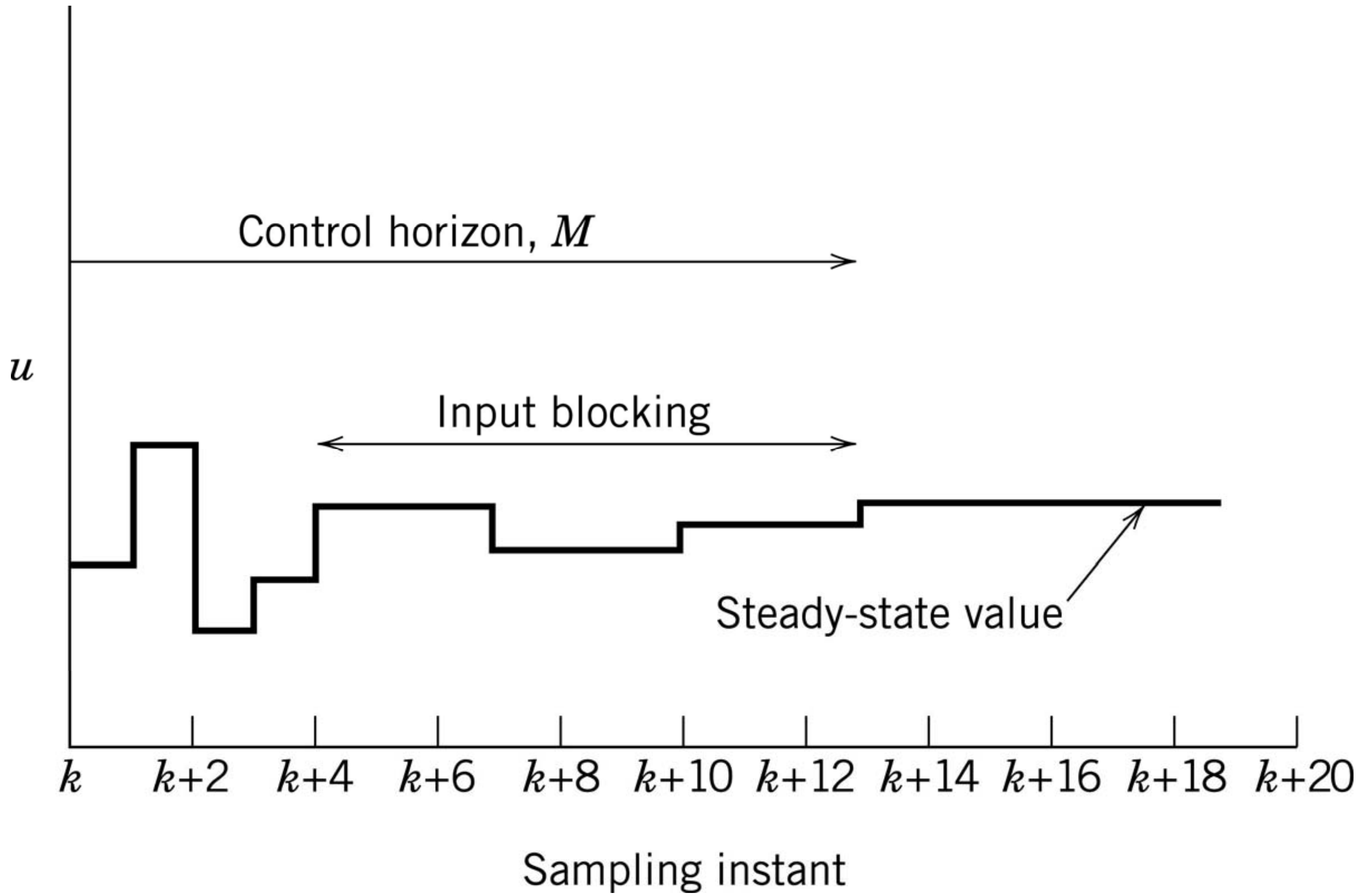


Figure 20.10 Input blocking.

Chapter 20

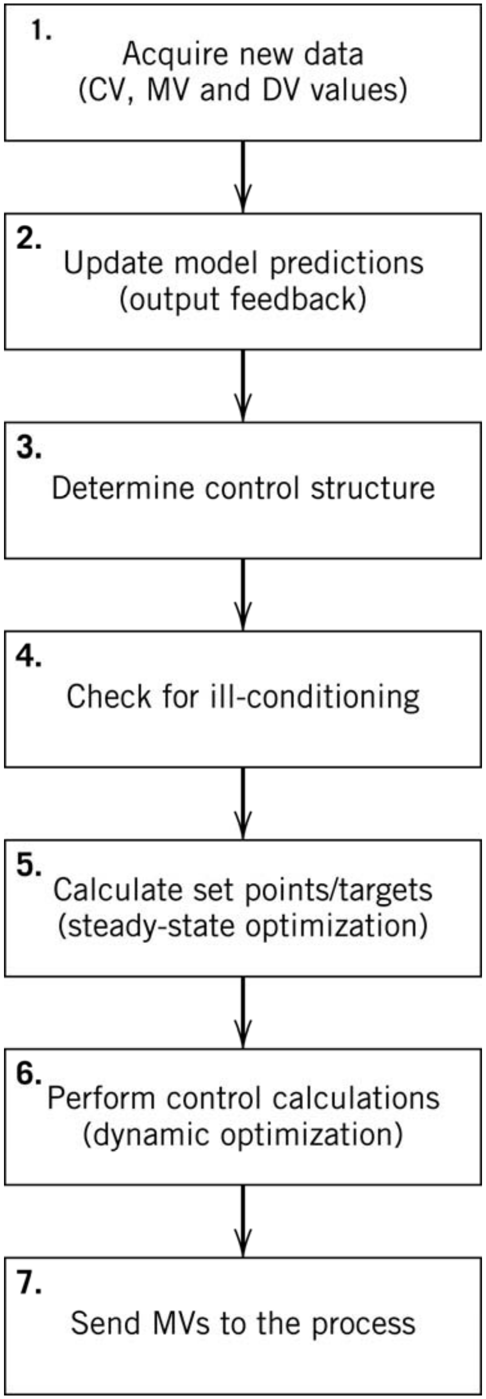


Figure 20.9 Flow chart for MPC calculations.

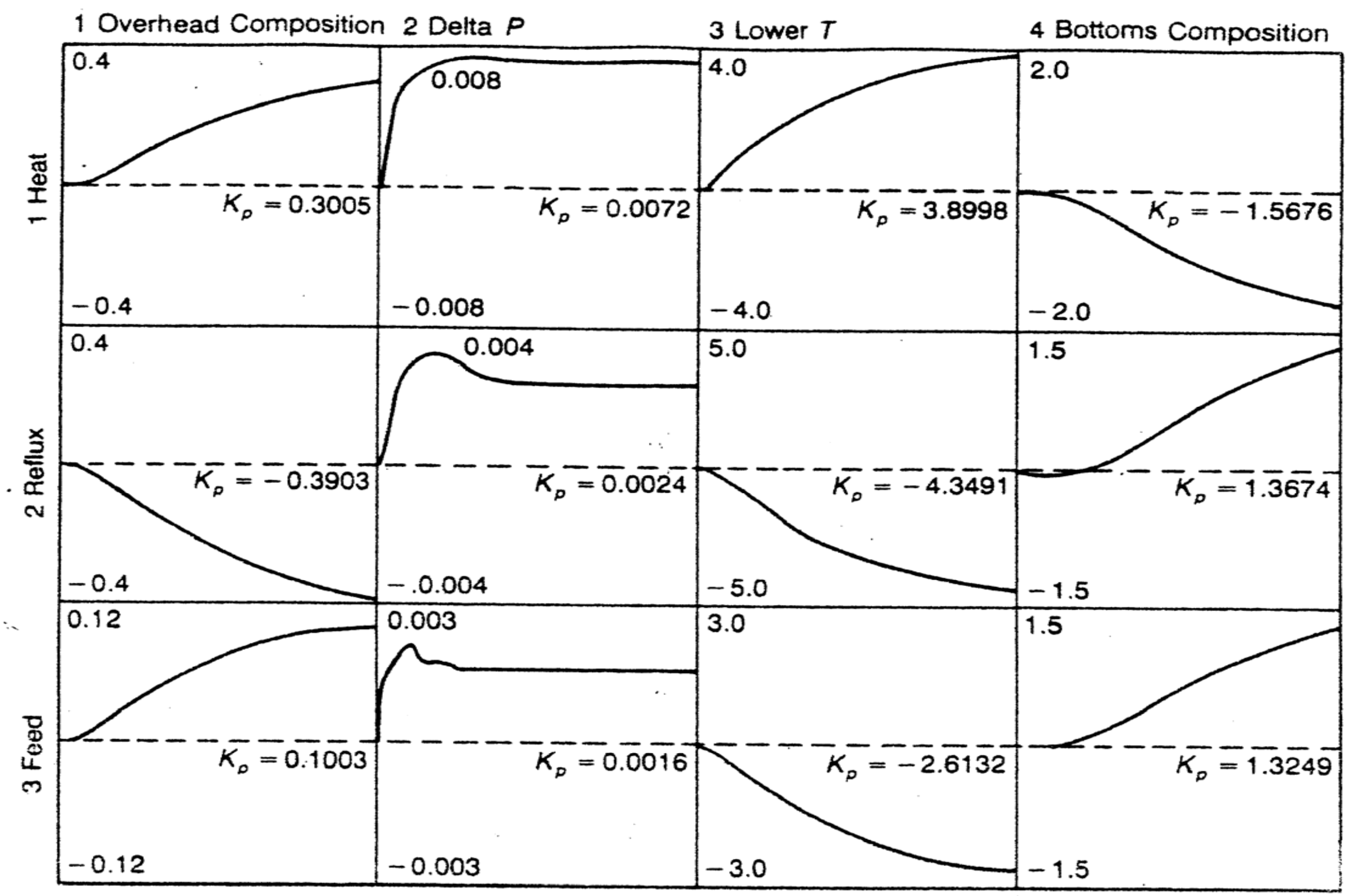


Figure 20.8. Individual step-response models for a distillation column with three inputs and four outputs. Each model represents the step response for 120 minutes. Reference: Hokanson and Gerstle (1992).

Reference Trajectory for MPC

Reference Trajectory

- A reference trajectory can be used to make a gradual transition to the desired set point.
- The *reference trajectory* $\boldsymbol{Y}_r$ can be specified in several different ways. Let the reference trajectory over the prediction horizon P be denoted as:

$$\boldsymbol{Y}_r(k+1) \triangleq col[\boldsymbol{y}_r(k+1), \boldsymbol{y}_r(k+2), \ldots, \boldsymbol{y}_r(k+P)] \qquad (20\text{-}47)$$

where $\boldsymbol{Y}_r$ is an mP vector where m is the number of outputs.

Exponential Trajectory from *y(k)* to $y_{sp}(k)$

A reasonable approach for the i-th output is to use:

$$y_{i,r}(k+j) = (a_i)^j y_i(k) + [1 - (a_i)^j]\, y_{i,sp}(k) \qquad (20\text{-}48)$$

for $i=1,2,\ldots, m$ and $j=1, 2, \ldots, P$.

MPC Control Calculations

- The control calculations are based on minimizing the predicted deviations between the reference trajectory.
- The *predicted error is defined as:*

$$\hat{\boldsymbol{E}}(k+1) \triangleq \boldsymbol{Y_r}(k+1) - \tilde{\boldsymbol{Y}}(k+1) \qquad (20\text{-}50)$$

where $\tilde{\boldsymbol{Y}}(k+1)$ is the corrected prediction defined in (20-37). Similarly, the *predicted unforced error*, $\hat{\boldsymbol{E}}^{o}(k+1)$, is defined as:

$$\hat{\boldsymbol{E}}^{o}(k+1) \triangleq \boldsymbol{Y_r}(k+1) - \tilde{\boldsymbol{Y}}^{o}(k+1) \qquad (20\text{-}51)$$

- Note that all of the above vectors are of dimension, *mP*.
- The objective of the control calculations is to calculate the control policy for the next *M* time intervals:

$$\Delta \boldsymbol{U}(k) \triangleq col[\Delta \mathbf{u}(k), \Delta \mathbf{u}(k+1), \cdots, \Delta \mathbf{u}(k+M-1)] \quad (20\text{-}18)$$

MPC Performance Index

- The *rM*-dimensional vector $\Delta \boldsymbol{U}(k)$ is calculated so as to minimize:

  a. The predicted errors over the prediction horizon, *P*.

  b. The size of the control move over the control horizon, *M*.

- **Example:** Consider a quadratic performance index:

$$\min_{\Delta \boldsymbol{U}(k)} \boldsymbol{J} = \hat{\boldsymbol{E}}(k+1)^T \boldsymbol{Q} \, \hat{\boldsymbol{E}}(k+1) + \Delta \boldsymbol{U}(k)^T \boldsymbol{R} \, \Delta \boldsymbol{U}(k) \qquad (20\text{-}54)$$

where $\boldsymbol{Q}$ is a positive-definite weighting matrix and $\boldsymbol{R}$ is a positive semi-definite matrix.

Both $\boldsymbol{Q}$ and $\boldsymbol{R}$ are usually diagonal matrices with positive diagonal elements.

The weighting matrices are used to weight the most important outputs and inputs (cf. Section 20.6).

MPC Control Law: Unconstrained Case

- The MPC control law that minimizes the objective function in Eq. (20-54) can be calculated analytically,

$$\Delta \boldsymbol{U}(k) = (\boldsymbol{S}^T \boldsymbol{Q}\,\boldsymbol{S} + \boldsymbol{R})^{-1} \boldsymbol{S}^T \boldsymbol{Q}\,\hat{\boldsymbol{E}}^{o}(k+1) \qquad (20\text{-}55)$$

where $\boldsymbol{S}$ is the dynamic matrix defined in (20-41).

- This control law can be written in a more compact form,

$$\Delta \boldsymbol{U}(k) = \boldsymbol{K}_c \hat{\boldsymbol{E}}^{o}(k+1) \qquad (20\text{-}56)$$

where controller gain matrix $\boldsymbol{K}_c$ is defined to be:

$$\boldsymbol{K}_c \triangleq (\boldsymbol{S}^T \boldsymbol{Q}\,\boldsymbol{S} + \boldsymbol{R})^{-1} \boldsymbol{S}^T \boldsymbol{Q} \qquad (20\text{-}57)$$

- Note that $\boldsymbol{K}_c$ can be evaluated off-line, rather than on-line, provided that the dynamic matrix $\boldsymbol{S}$ and weighting matrices, $\boldsymbol{Q}$ and $\boldsymbol{R}$, are constant.

- The calculation of $\boldsymbol{K}_c$ requires the inversion of an rM x rM matrix where r is the number of input variables and M is the control horizon.

MPC Control Law: Receding Horizon Approach

$r \times mP$

- MPC control law:

$$\Delta \boldsymbol{U}(k) = \boldsymbol{K}_c \hat{\boldsymbol{E}}^o(k+1) \qquad (20\text{-}56)$$

where:

$$\Delta \boldsymbol{U}(k) \triangleq col[\Delta \mathbf{u}(k), \Delta \mathbf{u}(k+1), \cdots, \Delta \mathbf{u}(k+M-1)] \qquad (20\text{-}18)$$

- Note that the controller gain matrix, $\boldsymbol{K}_c$, is an rM x mP matrix.

- In the *receding horizon control* approach, only the first step of the M-step control policy, $\Delta \boldsymbol{u}(k)$, in (20-18) is implemented.

$$\Delta \boldsymbol{u}(k) = \boldsymbol{K}_{c1} \hat{\boldsymbol{E}}^{\boldsymbol{o}}(k+1) \qquad (20\text{-}58)$$

where matrix $\boldsymbol{K}_{c1}$ is defined to be the first r rows of $\boldsymbol{K}_c$. Thus, $\boldsymbol{K}_{c1}$ has dimensions of r x mP.

Selection of Design Parameters

Model predictive control techniques include a number of design parameters:

- N: model horizon
- Δt: sampling period
- P: prediction horizon (number of predictions)
- M: control horizon (number of control moves)
- $\boldsymbol{Q}$: weighting matrix for predicted errors ($\boldsymbol{Q} > 0$)
- $\boldsymbol{R}$: weighting matrix for control moves ($\boldsymbol{R} \geq 0$)

Selection of Design Parameters (continued)

1. ***N* and Δ*t***

   These parameters should be selected so that $N\,\Delta t \geq$ open-loop settling time. Typical values of *N*:

   $$30 \leq N \leq 120$$

2. **Prediction Horizon, *P***

   Increasing *P* results in less aggressive control action

   Set $P = N + M$

3. **Control Horizon, *M***

   Increasing *M* makes the controller more aggressive and increases computational effort, typically:

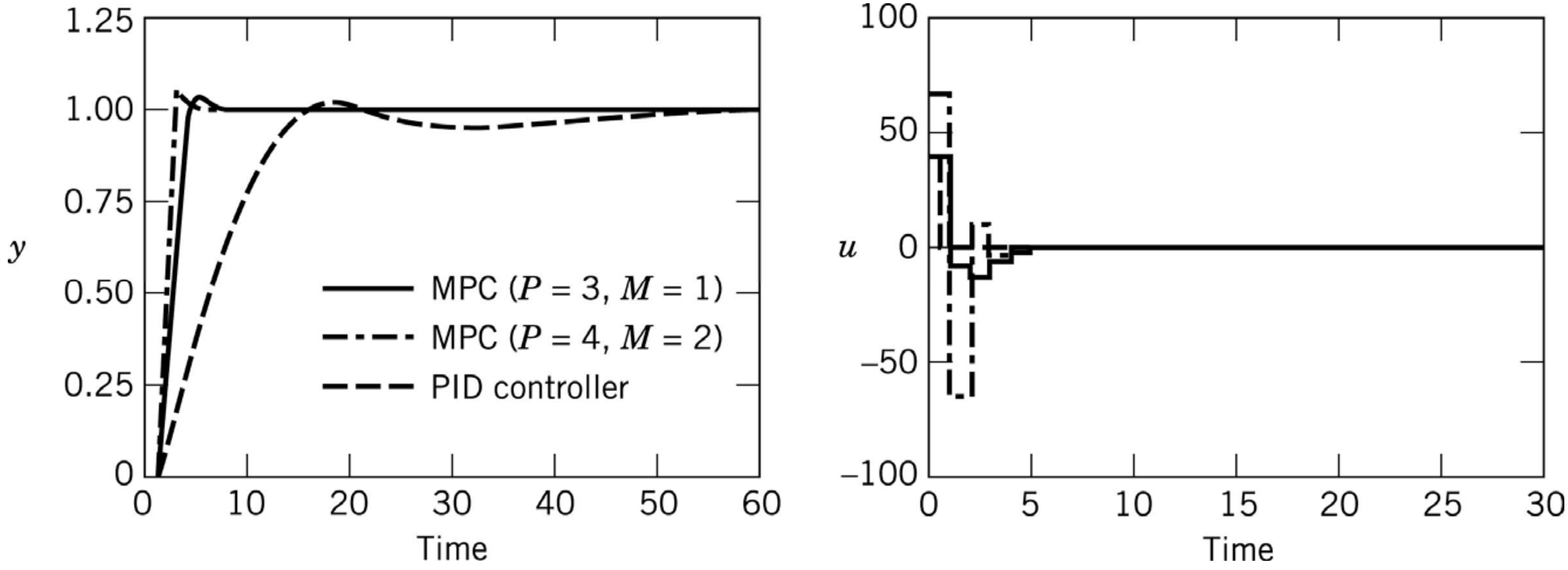
   $$5 \leq M \leq 20$$

4. **Weighting matrices *Q* and *R***

   Diagonal matrices with largest elements corresponding to most important variables

Example 20.5: set-point responses

$$G(s) = \frac{e^{-s}}{(10s+1)(5s+1)}$$



Example 20.5: disturbance responses