CHBE507 LECTURE II
MPC Revisited

Professor Dae Ryook Yang

Fall 2013
Dept. of Chemical and Biological Engineering
Korea University

Process Models

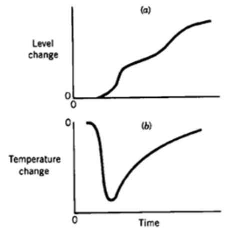
- **Transfer function models**
  - Fixed order and structure
  - Parametric: few parameters to identify
  - Need very high order model for unusual behavior
- **Convolution models**
  - Continuous form

$$y(t) = \int_0^t h(\tau)u(t-\tau)d\tau$$

  - Discrete form

$$y(k) = \sum_{i=0}^{k} h(i)u(k-i)$$

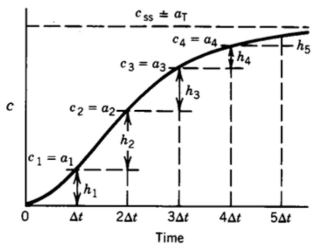
Impulse response



  - Many parameters, but easily obtained from the step or impulse response

Step Response Model

- **From open-loop step test**
  - Sampling time: Δt
  - Step response coefficients: a_i
  - Read the values of the unit step response



- **FSR model**
  - Finite step response (FSR)

$$y_k = a_k \quad (u_k = 1, \forall k \geq 0)$$

  - Using superposition principle for arbitrary input changes

$$u_k = \Delta u_0 + \Delta u_1 + \cdots + \Delta u_k \text{ where } \Delta u_i = u_i - u_{i-1}$$

$$y_k = y_0 + y_k\big|_{\Delta u_0} + y_k\big|_{\Delta u_1} + \cdots + y_k\big|_{\Delta u_{k-1}}$$

$$= y_0 + a_k\Delta u_0 + a_{k-1}\Delta u_1 + \cdots + a_1\Delta u_{k-1}$$

- **After $t = T\Delta t$, the step response reaches steady state at least 99%**

$$y_1 = y_0 + a_1\Delta u_0$$
$$y_2 = y_0 + a_2\Delta u_0 + a_1\Delta u_1$$
$$y_3 = y_0 + a_3\Delta u_0 + a_2\Delta u_1 + a_1\Delta u_2$$
$$\vdots$$
$$y_T = y_0 + a_T\Delta u_0 + a_{T-1}\Delta u_1 + \cdots + a_2\Delta u_{T-2} + a_1\Delta u_{T-1}$$
$$y_{T+1} = y_0 + a_T\Delta u_0 + a_T\Delta u_1 + a_{T-1}\Delta u_2 + \cdots + a_2\Delta u_{T-1} + a_1\Delta u_T$$
$$y_{T+2} = y_0 + a_T\Delta u_0 + a_T\Delta u_1 + a_T\Delta u_2 + a_{T-1}\Delta u_3 + \cdots + a_2\Delta u_T + a_1\Delta u_{T+1}$$
$$\vdots$$

$$\Rightarrow y_n = y_0 + \sum_{i=1}^{n} a_i\Delta u_{n-i} \quad (a_i = a_T, \forall i \geq T)$$ **(FSR Model)**

  - If there is a delay, the FSR coefficients during the delay will be zero.

Impulse Response Model

- **Impulse response coefficients**

$$h_i = a_i - a_{i-1} \quad (i = 1, 2, \cdots, T)$$
$$h_0 = 0$$

$$y_n = y_0 + \sum_{i=1}^{T} a_i \Delta u_{n-i} = y_0 + \sum_{i=1}^{T} a_i (u_{n-i} - u_{n-i-1})$$
$$= y_0 + (a_1 u_{n-1} - a_1 u_{n-2}) + (a_2 u_{n-2} - a_2 u_{n-3}) + \cdots + (a_n u_1 - a_n u_0) + (a_n u_0 - a_n u_{-1}) + \cdots$$
$$= y_0 + a_1 u_{n-1} + (a_2 - a_1) u_{n-2} + \cdots + (a_n - a_{n-1}) u_1 + (\overset{0}{a_n - a_n}) u_0 + \cdots$$
$$= y_0 + (a_1 - \overset{0}{a_0}) u_{n-1} + (a_2 - a_1) u_{n-2} + \cdots + (a_n - a_{n-1}) u_1$$

$$\Rightarrow y_n = y_0 + \sum_{i=1}^{T} h_i u_{n-i} \quad (h_i = 0, \forall i \geq T)$$ **(FIR Model)**

Matrix Form of the Predictive Model

- **Horizons**
  - **Model horizon: T (number of model coefficients)**
  - **Control horizon: U (number of control moves)**
  - **Prediction horizon: V (number of predictions in the future)**

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_V \end{bmatrix} = \begin{bmatrix} a_1 & 0 & 0 & \cdots & 0 \\ a_2 & a_1 & 0 & \cdots & 0 \\ \vdots & a_2 & a_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \\ a_V & a_{V-1} & a_{V-2} & & a_{V-U+1} \end{bmatrix} \begin{bmatrix} \Delta u_0 \\ \Delta u_1 \\ \Delta u_2 \\ \vdots \\ \Delta u_{U-1} \end{bmatrix}$$

$$\mathbf{y} = \mathbf{A} \Delta \mathbf{u}$$

  - **A: Dynamic matrix**

Single-Step Prediction

- **From the FIR model**

$$\hat{y}_n = y_0 + \sum_{i=1}^{T} h_i u_{n-i} \qquad \hat{y}_{n+1} = y_0 + \sum_{i=1}^{T} h_i u_{n+1-i}$$

$$\Rightarrow \hat{y}_{n+1} = \hat{y}_n + \sum_{i=1}^{T} h_i \Delta u_{n+1-i}$$ **(Recursive prediction)**

- **Corrected prediction based on the measurement**
  - **Assume the error between the model prediction and the measurement will present in the future with same magnitude**

$$y^*_{n+1} - \hat{y}_{n+1} = y_n - \hat{y}_n \quad (y_n \text{ is the current measurement})$$

$$\Rightarrow y^*_{n+1} = \hat{y}_{n+1} + (y_n - \hat{y}_n) = y_n + \sum_{i=1}^{T} h_i \Delta u_{n+1-i}$$

Multi-Step Prediction

- **From the single-step prediction (j-step prediction)**

$$\hat{y}_{n+j} = \hat{y}_{n+j-1} + \sum_{i=1}^{T} h_i \Delta u_{n+j-i} \quad (j = 1, 2, \cdots, V)$$
$$y^*_{n+j} - \hat{y}_{n+j} = y^*_{n+j-1} - \hat{y}_{n+j-1} \quad (y_{n+j-1} \text{ is not available if } j > 1)$$

$$\Rightarrow y^*_{n+j} = y^*_{n+j-1} + \sum_{i=1}^{T} h_i \Delta u_{n+j-i} \quad (j = 1, 2, \cdots, V)$$

- **Matrix form when V≥U**

Dynamic Matrix, A

$$\begin{bmatrix} y^*_{n+1} \\ y^*_{n+2} \\ y^*_{n+3} \\ \vdots \\ y^*_{n+V} \end{bmatrix} = \begin{bmatrix} a_1 & 0 & 0 & \cdots & 0 \\ a_2 & a_1 & 0 & \cdots & 0 \\ \vdots & a_2 & a_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \\ a_V & a_{V-1} & a_{V-2} & & a_{V-U+1} \end{bmatrix} \begin{bmatrix} \Delta u_n \\ \Delta u_{n+1} \\ \Delta u_{n+2} \\ \vdots \\ \Delta u_{n+U-1} \end{bmatrix} + \begin{bmatrix} y_n + P_1 \\ y_n + P_2 \\ y_n + P_3 \\ \vdots \\ y_n + P_V \end{bmatrix}$$

where

$$P_i = \sum_{j=1}^{i} S_j \quad (i = 1, 2, \cdots, V)$$

$$S_j = \sum_{i=1}^{T} h_i \Delta u_{n+j-i} \quad (i = 1, 2, \cdots, V)$$

- S_j: **the incremental effect of the past (previously implemented) movements of input on the ($n+j$)-th future output prediction (where n is current time)**
- P_i: **the projection which includes future prediction of y based on all previously implemented input changes.**
- P_i **and** S_j **depend only on past input changes.**

- **If the past information is known, then the future input changes will affect the future outputs and the future outputs can be adjusted by carefully selecting the future inputs.**

- **Currently, n is current time and y_n is measured.**

$$y^*_{n+1} = y_n + \sum_{i=1}^{T} h_i \Delta u_{n+1-i} = h_1 \Delta u_n + y_n + \sum_{i=2}^{T} h_i \Delta u_{n+1-i} = a_1 \Delta u_n + y_n + \sum_{i=2}^{T} h_i \Delta u_{n+1-i}$$

$$y^*_{n+2} = y^*_{n+1} + \sum_{i=1}^{T} h_i \Delta u_{n+2-i} = (h_2 + h_1)\Delta u_n + h_1 \Delta u_{n+1} + \sum_{i=3}^{T} h_i \Delta u_{n+2-i} + y_n + \sum_{i=2}^{T} h_i \Delta u_{n+1-i}$$

$$y^*_{n+3} = y^*_{n+3} + \sum_{i=1}^{T} h_i \Delta u_{n+3-i}$$

$$= (h_2 + h_2 + h_1)\Delta u_n + (h_2 + h_1)\Delta u_{n+1} + h_1 \Delta u_{n+2} + \sum_{i=4}^{T} h_i \Delta u_{n+2-i} + y_n + \sum_{i=2}^{T} h_i \Delta u_{n+1-i} + \sum_{i=3}^{T} h_i \Delta u_{n+2-i}$$

$$\vdots$$

$$y^*_{n+V} = y^*_{n+V-1} + \sum_{i=1}^{T} h_i \Delta u_{n+V-1-i} = a_V \Delta u_n + a_{V-1} \Delta u_{n+1} + \cdots + a_{V-U+1} \Delta u_{n+U-1}$$

$$+ y_n + \sum_{i=V+1}^{T} h_i \Delta u_{n+V-i} + \cdots + \sum_{i=3}^{T} h_i \Delta u_{n+2-i} + \sum_{i=2}^{T} h_i \Delta u_{n+1-i}$$

$$= a_V \Delta u_n + a_{V-1} \Delta u_{n+1} + \cdots + a_{V-U+1} \Delta u_{n+U-1} + y_n + \sum_{j=1}^{V} \sum_{i=j+1}^{T} h_i \Delta u_{n+j-i}$$

↑ **Depend on only future** ↑ **Depend on only past**

Controller Design Method (DMC)

- **Objective**
  - **Minimize errors between future set points and predictions**

$$\hat{\mathbf{E}} = \begin{bmatrix} r_{n+1} - y^*_{n+1} \\ r_{n+2} - y^*_{n+2} \\ \vdots \\ r_{n+V} - y^*_{n+V} \end{bmatrix} = \mathbf{r} - (\mathbf{A}\Delta\mathbf{u} + y_n \mathbf{e} + \mathbf{P}) = -\mathbf{A}\Delta\mathbf{u} + \hat{\mathbf{E}}'$$

← **Closed-loop prediction error based only on current and future control action**

where

$$\hat{\mathbf{E}}' = \begin{bmatrix} r_{n+1} - y_n - P_1 \\ r_{n+2} - y_n - P_2 \\ \vdots \\ r_{n+V} - y_n - P_V \end{bmatrix}$$

← **Open-loop prediction error based only on past control action**

- **Solution**

$$-\mathbf{A}\Delta\mathbf{u} + \hat{\mathbf{E}}' = \mathbf{0} \quad \Rightarrow \quad \Delta\mathbf{u} = (\mathbf{A}^*)^{-1}\hat{\mathbf{E}}'$$

↖ **Some inverse of A**

- **If $U=V$ and A is invertible,**

$$\Delta\mathbf{u} = \mathbf{A}^{-1}\hat{\mathbf{E}}'$$

← **It gives no steady-state offset since it has integral action.**

- **If $U<V$ (A is not invertible),**

$$\Delta\mathbf{u} = \boxed{(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T}\hat{\mathbf{E}}' = \mathbf{K}_c\hat{\mathbf{E}}'$$

A^+: Left pseudoinverse of A

$A^+A=I$: identity matrix

AA^+: *idempotent* matrix (BB=B)

- **Optimization concept**

$$\min(J = \hat{\mathbf{E}}^T\hat{\mathbf{E}}) = \min(-\mathbf{A}\Delta\mathbf{u} + \hat{\mathbf{E}}')^T(-\mathbf{A}\Delta\mathbf{u} + \hat{\mathbf{E}}')$$

$$\frac{\partial J}{\partial \Delta\mathbf{u}} = -2\mathbf{A}^T(-\mathbf{A}\Delta\mathbf{u} + \hat{\mathbf{E}}') = 2(\mathbf{A}^T\mathbf{A}\Delta\mathbf{u} - \mathbf{A}^T\hat{\mathbf{E}}') = 0$$

$$\Rightarrow \Delta\mathbf{u} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\hat{\mathbf{E}}'$$

$$\min J = (\hat{\mathbf{E}}^T\mathbf{W}_1\hat{\mathbf{E}} + \Delta\mathbf{u}^T\mathbf{W}_2\Delta\mathbf{u})$$

$$\frac{\partial J}{\partial \Delta\mathbf{u}} = -2\mathbf{A}^T\mathbf{W}_1(-\mathbf{A}\Delta\mathbf{u} + \hat{\mathbf{E}}') + 2\mathbf{W}_2\Delta\mathbf{u} = 2((\mathbf{A}^T\mathbf{W}_1\mathbf{A} + \mathbf{W}_2)\Delta\mathbf{u} - \mathbf{A}^T\mathbf{W}_1\hat{\mathbf{E}}') = 0$$

$$\Rightarrow \Delta\mathbf{u} = (\mathbf{A}^T\mathbf{W}_1\mathbf{A} + \mathbf{W}_2)^{-1}\mathbf{A}^T\mathbf{W}_1\hat{\mathbf{E}}'$$

- **Adjustable parameters of MPC (Tuning parameters)**
  - **Weighting matrices**
    - **If W_1>>W_2, the most important objective is to minimize error of the process outputs and inputs will move quite freely.**
    - **If W_1<<W_2, the most important objective is to minimize the input movements and controller cares much less the errors. (almost no control)**
    - **Otherwise, it depends on the relative size of the weighting matrices.**
      - **If W_1>W_2, aggressive action will be taken to reduce the error.**
      - **If W_1<W_2, conservative action will be taken to reduce the input movements while reduce the error if the action is not too aggressive.**
    - **The W_2 is called *input penalty* or *input move suppression factor*.**
    - **Typically, use W_1=I and W_2=f^2I and adjust *f*.**
    - **If a different weighting for outputs or inputs is required, use diagonal matrix as the weighting matrix.**

- **Horizons**
  - **Model horizon (*T*)**
    - **Select *T* such that** $T\Delta t \geq$ (open-loop settling time)
    - ***T* is typically 20 to 70.**
  - **Prediction horizon (*V*)**
    - **Increasing *V* results in more conservative control action, a stabilizing effect, and more computational burden.**
    - **An important tuning parameter**
  - **Control horizon (*U*)**
    - **Suitable first guess is to choose *U* so that** $U\Delta t \cong t_{60}$
    - **The larger the value of *U* is, the more computation time is required.**
    - **Too large a value of *U* results in excessive control action**
    - **Smaller value of U leads to a robust controller that is relatively insensitive to model error.**

MIMO Extension

- **2x2 case**

$$\hat{\mathbf{E}} = -\mathbf{A}\Delta\mathbf{u} + \hat{\mathbf{E}}'$$

where

$$\hat{\mathbf{E}} = [\hat{\mathbf{E}}_1; \hat{\mathbf{E}}_2] \qquad \Delta\mathbf{u} = [\Delta\mathbf{u}_1; \Delta\mathbf{u}_2]$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$$

- **General case**
  - **Extend the vectors and matrices in the same manner.**
  - **If the MPC is formulated in a different form such as state-space model, different form of MIMO extension is more convenient.**

Constraints Handling

- **Formulate and solve the MPC in an optimization framework**

$$\min J = (\hat{\mathbf{E}}^T \mathbf{W}_1 \hat{\mathbf{E}} + \Delta\mathbf{u}^T \mathbf{W}_2 \Delta\mathbf{u})$$

$$\text{subject to } \mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U$$

$$\mathbf{y}^L \leq \mathbf{y} \leq \mathbf{y}^U$$

$$\text{and other constraints}$$

- **Solve this optimization problem in QP**
  - **DMC by DMCC used LP**

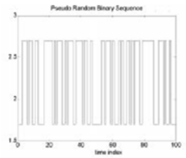
Identification of Models

- **FSR or FIR models: use step or pulse test**
  - **Assume operation at steady state**
  - **Make change in input Δu (or δu)**
    - **If Δu is too small, output change may not noticeable**
    - **If Δu is too large, linearity may not hold**
  - **Measure output at regular intervals Δt**
    - **The Δt should be chosen so that T is 20-70,typically 40.**
  - **Perform multiple experiments and average them and additional experiments for verification**
  - **High frequency information may not be accurate for step test.**
  - **Ideal pulse is hard to implement.**

- **Least Squares Identification**
  - **Get the output using PRBS (Pseudo Random Binary Signal)**

    $\mathbf{u} = [u_1 \; u_2 \; \cdots \; u_M] \qquad \mathbf{y} = [y_1 \; y_2 \; \cdots \; y_M]$
  - **Get the FIR model**

    $\tilde{y}_k = h_1 u_{k-1} + h_2 u_{k-2} + \cdots + h_N u_{k-N}$
  - **Minimize the error between measurements and output,** $d_k = y_k - \tilde{y}_k$



$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} u_0 & u_{-1} & \cdots & u_{1-N} \\ u_1 & u_0 & \cdots & u_{2-N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{M-1} & u_{M-2} & \cdots & u_{M-N} \end{bmatrix} \begin{bmatrix} h_1 \\ h_1 \\ \vdots \\ h_N \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_M \end{bmatrix} \qquad \mathbf{d} = \mathbf{y} - \mathbf{U}\mathbf{h}$$

$$\min_{\mathbf{h}} \mathbf{d}^T \mathbf{d} = \min_{\mathbf{h}} (\mathbf{y} - \mathbf{U}\mathbf{h})^T (\mathbf{y} - \mathbf{U}\mathbf{h}) \Rightarrow \mathbf{h} = (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \mathbf{y}$$

- **Discussions**
  - **Random input testing, if appropriately designed, gives better models than the step or pulse testing does since it can equally excite low to high frequency dynamics of the process.**
  - **If $\mathbf{U}^T\mathbf{U}$ is singular, the inverse doesn't exist and identification fails. (Need persistent excitation condition)**
  - **When the number of coefficients is large, $\mathbf{U}^T\mathbf{U}$ can be easily singular (or nearly singular). To avoid the numerical, a regularization term is added the the cost function. (ridge regression)**

$$\min_{\mathbf{h}} [(\mathbf{y} - \mathbf{U}\mathbf{h})^T (\mathbf{y} - \mathbf{U}\mathbf{h}) + \alpha \mathbf{h}^T \mathbf{h}] \Rightarrow \mathbf{h} = (\mathbf{U}^T \mathbf{U} + \alpha \mathbf{I})^{-1} \mathbf{U}^T \mathbf{y}$$

Data Treatments

- **The data need to be processed before they are used in identification.**

- **Spike/Outlier Removal**
  - **Check plots of data and remove obvious outliers (e.g., that are impossible with respect to surrounding data points). Fill in by interpolation.**
  - **After modeling, plot of actual vs. predicted output (using measured input and modeling equations) may suggest additional outliers. Remove and redo modeling, if necessary.**
  - **But don't remove data unless there is a clear justification.**

- **Bias Removal and Normalization**
  - Compute the data average and subtract it to create deviation variables, i.e.,

    $\hat{y}_k = (y_k - y_{ref}) / c_y$ where $y_{ref} = \sum_{i=1}^{M} y_i / M$

    $\hat{u}_k = (u_k - u_{ref}) / c_u$ where $u_{ref} = \sum_{i=1}^{M} u_i / M$
  - Use the given steady-state values of the variables instead to compute the deviation variables, i.e.,

    $\hat{y}_k = (y_k - y_{ss}) / c_y$ and $\hat{u}_k = (u_k - u_{ss}) / c_u$

    where y_{ss} and u_{ss} represent a priori given steady-state values of the process output and input respectively.
  - The input/output data can be biased by the nonzero steady state and also by load disturbance effects. To remove the (time-varying) bias, differencing can be performed for the input/output data.

    $\Delta y_k = (y_k - y_{k-1}) / c_v$ and $\Delta u_k = (u_k - u_{k-1}) / c_u$

    $\Rightarrow$ Identification for Δy_k and Δu_k
  - In all cases, the process data are conditioned by scaling before using in identification.

- **Prefiltering**
  - If the data contain too much frequency components over an undesired range and/or if we want to obtain a model that fits well the data over a certain frequency range, data prefiltering (via digital filters) can be done.