What is Model Predictive Control (MPC)?

- **Multivariable control**
  - Calculate all the MV’s at the same time based on all PV values
  - Not like multiloop control, no decoupling scheme is needed
  - More complicated

- **Constraints handling**
  - Process industry requires many constraints
    - Safety
    - Operational limitations
    - Product quality
  - No previous methodology handles constraints explicitly

- **Flexible formulation**
  - Many control objectives can be formulated as the objective function and constraints

What is Model Predictive Control (MPC)?

- **Features**
  - Computer control: sampled-data control
  - Model-based control: dynamic model is required
    - Fundamental model
    - Empirical model (usually step response based)
  - Predictive: adjust process based on the future prediction
    - Not just based on the current error
  - Optimization-based: no explicit control law
    - Formulated with objective function and constraints
    - Optimization is solved at each sampling time
  - Integrated: constraints and economic handling
    - Optimizing control
    - Servo or regulatory control
  - Receding horizon control: future window is moving forward
    - Repeat the prediction and optimization at each sample time
    - Update the input based on the new measurement

- **Similarity to human decision-making**
  - Sense: collect new information
  - Assess: update the memories
  - Predict: forecast the outcome for a variety of possible decisions
  - Optimize: determine the best decision for given objective and constraints
  - Implement: the action for this time is imposed
  - Repeat: information collection, update and optimization done every so often
Model Predictive Control Originated in 1980

- Techniques developed by industry:
  - Dynamic Matrix Control (DMC)
    - Shell Development Co., Cutler and Ramaker (1980)
    - Cutler later formed DMC, Inc.
    - DMC acquired by Aspentech in 1997
  - Model Algorithmic Control (MAC)
    - ADERSA/GERBIOS, Richalet et al (1978)

- Over 4500 applications of MPC by the end of 1999 since 1980 (Qin and Badgwell, 2003)
- Predominantly in the oil and petrochemical industries but the range of applications is expanding.
- Models used are predominantly empirical models developed through plant testing.
- Technology is used not only for multivariable control, but for most economic operation within constraint boundaries.

Reason for Popularity (1)

- MPC provides a systematic, consistent, and integrated solution to process control problems with complex features:
  - Delays, inverse responses and other complex dynamics.
  - Strong interactions (e.g., large RGA)
  - Constraints (e.g., actuator limits, output limits)

Example 1: Blending control system

- Control r, and r
- Control q if possible.
- Flowrates of additives are limited.

MPC: Solve at each time k

Classical Solution

\[
\min_{r, q} \sum_{i=1}^{k} \left( r_i(k+i) - r_i \right)^2 \\
\left( r_i(k+i) - q_i \right)^2 \\
\gamma \left( r_i(k+i) - q_i \right)^2 \\

\gamma < 1
\]
• **Example 2: Heavy Oil Fractionator**

![Diagram of Heavy Oil Fractionator](image)

- Keep $y_1 \geq T_{min}$
- Control the two compositions $y_1$ and $y_2$
- Minimize $u_2$ to maximize the heat recovery.

Solution using the classical tools will be very complicated and a satisfactory solution is not known.

It is fairly easy to translate the above objective (as well as the valve limits) as a minimization function and inequality constraints as required by MPC.

• **Advantages of MPC over Traditional APC**
  - Integrated solution
    - Automatic constraint handling
    - Feedforward/feedback
  - No need for decoupling or delay compensation
  - Efficient utilization of degrees of freedom
    - Can handle nonsquare systems (e.g., more MVs and CVs)
    - Assignable priorities, ideal settling values for MVs
  - Consistent, systematic methodology
  - Realized benefits
    - Higher on-line times
    - Cheaper implementation
    - Easier maintenance

• **Reason for Popularity (2)**

  • Emerging popularity of on-line optimization
  • Process optimization and control are often conflicting objectives
    - Optimization pushes the process to the boundary of constraints.
    - Quality of control determines how close one can push the process to the boundary.
  • Implications for process control
    - High performance control is needed to realize on-line optimization.
    - Constraint handling is a must.
    - The appropriate tradeoff between optimization and control is time-varying and is best handled within a single framework.

• **Synergy Between Optimization and Control**
• **Control Hierarchy**
  - Regulatory and basic control
    - PID control loops, cascade loops, independent actuators, etc.
    - The set point of each loop are given by advanced control
    - Fast sampling time and response (seconds or less)
  - Advanced control
    - Such as MPC
    - Manipulates the set points of the regulatory and basic controls
    - Higher-level set points are given by the plant-wide optimizer
    - Sampling time (seconds to minutes)
  - Plant-wide optimization
    - Calculate the optimum steady-states operating conditions based on the strategy from CIM
    - Sampling time (hours)
  - CIM (Computer Integrated Manufacturing)
    - Reflect corporate strategy and market condition
    - Production schedule
    - Sampling time (months)

• **Return on Investment (ROI) for APC**

• **Importance of Modeling**
  - Almost all models used in MPC are typically empirical models “identified” through plant tests rather than first-principles models.
    - Step responses, pulse responses from plant tests.
    - Transfer function models fitted to plant test data.
  - Up to 80% of time and expense involved in designing and installing a MPC is attributed to modeling/system identification. → should be improved.
  - Keep in mind that obtained models are imperfect (both in terms of structure and parameters).
    - Importance of feedback update of the model.
    - Penalize excessive input movements.

• **Design effort**

![Diagram](image-url)
Challenges

• Efficient identification of control-relevant model
• Managing the sometimes exorbitant on-line computational load
  – Nonlinear models → Nonlinear Programs (NLP)
  – Hybrid system models (continuous dynamics + discrete events or switches, e.g., pressure swing adsorption) → Mixed Integer Programs (MILP)
  – Difficult to solve these reliably on-line for large-scale problems.
• How do we design model, estimator (of model parameters and state), and optimization algorithm as an integrated system - that are simultaneously optimized - rather than disparate components?
• Long-term maintenance of control system.

Current Status on MPC

• MPC is the established advanced multivariable control technique for the process industry. It is already an indispensable tool and its importance is continuing to grow.
• It can be formulated to perform some economic optimization and can also be interfaced with a larger-scale (e.g., plant-wide) optimization scheme.
• Obtaining an accurate model and having reliable sensors for key parameters are key bottlenecks.
• A number of challenges remain to improve its use and performance.

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
    \[ y(k) = \sum_{i=0}^{\infty} h(i)u(k-i) \]
  – Many parameters, but easily obtained from the step or impulse response

Step Response Model

• From open-loop step test
  – Sampling time: \( \Delta t \)
  – Step response coefficients: \( a_i \)
  – Read the values of the unit step response
• FSR model
  – Finite step response (FSR)
  – Using superposition principle for arbitrary input changes
    \[
    u_k = \Delta u_0 + \Delta u_1 + \cdots + \Delta u_k \quad \text{where} \quad \Delta u_i = u_i - u_{i-1}
    \]
    \[
    y_k = y_0 + y_1 \Delta t_0 + y_2 \Delta t_0 \Delta t_0 + \cdots + y_k \Delta t_0 \Delta t_0 \Delta t_0 = y_0 + a_k \Delta u_0 + a_{k-1} \Delta u_1 + \cdots + a_1 \Delta u_k
    \]
• After $t = T \Delta t$, the step response reaches steady state at least 99%

$$
\begin{align*}
   y_1 &= y_3 + a_3 \Delta u_0 \\
   y_2 &= y_3 + a_3 \Delta u_0 + a_2 \Delta u_1 \\
   y_3 &= y_3 + a_3 \Delta u_0 + a_2 \Delta u_1 + a_1 \Delta u_2 \\
   &\vdots \\
   y_T &= y_3 + a_3 \Delta u_0 + a_2 \Delta u_1 + \cdots + a_2 \Delta u_{T-2} + a_1 \Delta u_{T-1} + a_0 \Delta u_T \\
   y_{T+1} &= y_3 + a_3 \Delta u_0 + a_2 \Delta u_1 + \cdots + a_2 \Delta u_{T-2} + a_1 \Delta u_{T-1} + a_0 \Delta u_T \\
   &\vdots \\
\end{align*}
$$

$$
\Rightarrow y_n = y_3 + \sum_{i=1}^{T} a_i \Delta u_{n-i}, \quad (a_i = a_i, \forall i \geq T) \quad \text{(FSR Model)}
$$

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

**Impulse Response Model**

• Impulse response coefficients

$$
\begin{align*}
   \hat{h}_i &= a_i - a_{i-1} \quad (i = 1, 2, \ldots, T) \\
   \hat{h}_0 &= 0
\end{align*}
$$

$$
\begin{align*}
   y_n &= y_3 + \sum_{i=1}^{T} a_i \Delta u_{n-i} = y_3 + \sum_{i=1}^{T} a_i \Delta u_{n-i} \\
   &= y_3 + (a_3 \Delta u_{n-1} - a_3 \Delta u_{n-2}) + (a_2 \Delta u_{n-2} - a_2 \Delta u_{n-3}) + \cdots + (a_1 \Delta u_{n-T} - a_1 \Delta u_{n-1}) + (a_0 \Delta u_{n-T} - a_0 \Delta u_{n-1}) + \cdots \\
   &= y_3 + a_3 \Delta u_{n-1} + (a_2 - a_2) \Delta u_{n-2} + \cdots + (a_1 - a_1) \Delta u_{n-T} + (a_0 - a_0) \Delta u_{n-T} \\
   &= y_3 + (a_3 - a_3) \Delta u_{n-1} + (a_2 - a_2) \Delta u_{n-2} + \cdots + (a_1 - a_1) \Delta u_{n-T} + (a_0 - a_0) \Delta u_{n-T} \\
   &= y_3 + \sum_{i=1}^{T} \hat{h}_i \Delta u_{n-i}, \quad (\hat{h}_i = 0, \forall i \geq T) \quad \text{(FIR Model)}
\end{align*}
$$

**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)

$$
\begin{bmatrix}
   y_1 \\
   y_2 \\
   \vdots \\
   y_T \\
   y_{T+1} \\
   \vdots \\
   y_{T+V-1}
\end{bmatrix} = \begin{bmatrix}
   a_1 & 0 & 0 & \cdots & 0 \\
   a_2 & a_1 & 0 & \cdots & 0 \\
   \vdots & \vdots & \vdots & \ddots & \vdots \\
   a_{T-1} & a_{T-2} & a_{T-3} & \cdots & 0 \\
   a_T & a_{T-1} & a_{T-2} & \cdots & a_{T-V+1}
\end{bmatrix} \begin{bmatrix}
   \Delta u_0 \\
   \Delta u_1 \\
   \vdots \\
   \Delta u_T \\
   \Delta u_{T+1} \\
   \vdots \\
   \Delta u_{T+V-1}
\end{bmatrix}
$$

- $y = A \Delta u$
  - $A$: Dynamic matrix

**Single-Step Prediction**

• From the FIR model

$$
\begin{align*}
   \hat{y}_{n+1} &= y_3 + \sum_{i=1}^{T} \hat{h}_i \Delta u_{n-i} \\
   \hat{y}_{n+1} &= y_3 + \sum_{i=1}^{T} \hat{h}_i \Delta u_{n-i} \\
\end{align*}
$$

$$
\Rightarrow \hat{y}_{n+1} = \hat{y}_{n+1} + \sum_{i=1}^{T} \hat{h}_i \Delta u_{n-i} \quad \text{(Recursive prediction)}
$$

• Corrected prediction based on the measurement

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

$$
\begin{align*}
   y_{n+1} &= y_{n+1} - \hat{y}_{n+1} \quad (y_n \text{ is the current measurement}) \\
   \Rightarrow y_{n+1} &= y_{n+1} + \sum_{i=1}^{T} \hat{h}_i \Delta u_{n-i} \quad \text{(Recursive prediction)}
\end{align*}
$$
Multi-Step Prediction

- From the single-step prediction ($j$-step prediction)
  \[ \hat{y}_{m+j} = \hat{y}_{m+j-1} + \sum_{i=1}^{j} h_i \Delta u_{m+i-1} \quad (j = 1, 2, \ldots, J) \]
  \[ y_{m+j} - \hat{y}_{m+j} = y_{m+j-1} - \hat{y}_{m+j-1} \quad (y_{m+j} \text{ is not available if } j > 1) \]
  \[ \Rightarrow \hat{y}_{m+j} = \hat{y}_{m+j-1} + \sum_{i=1}^{J} h_i \Delta u_{m+i-1} \quad (j = 1, 2, \ldots, J) \]

- Matrix form when $V \geq U$

\[
\begin{bmatrix}
\hat{y}_{m+1} \\
\hat{y}_{m+2} \\
\vdots \\
\hat{y}_{m+j}
\end{bmatrix} =
\begin{bmatrix}
a_0 & 0 & \cdots & 0 \\
a_1 & a_0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
a_j & a_{j-1} & \cdots & a_0
\end{bmatrix}
\begin{bmatrix}
\Delta u_1 \\
\Delta u_2 \\
\vdots \\
\Delta u_{m+j-1}
\end{bmatrix}
+ \begin{bmatrix}
y_{m+1} \\
y_{m+2} \\
\vdots \\
y_{m+j-1}
\end{bmatrix}
+ \begin{bmatrix}
\hat{y}_{m+1} \\
\hat{y}_{m+2} \\
\vdots \\
\hat{y}_{m+j-1}
\end{bmatrix} + P_j
\]

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

\[
y'_m = y'_n = \sum_{i=1}^{n} h_i \Delta u_{m+i-1} = a_0 \Delta u_1 + a_1 \Delta u_2 + \cdots + a_n \Delta u_n
\]

\[
y''_m = y''_n = \sum_{i=1}^{n} h_i \Delta u_{m+i-1} = a_0 \Delta u_1 + a_1 \Delta u_2 + \cdots + a_n \Delta u_n
\]

\[
y'''_m = y'''_n = \sum_{i=1}^{n} h_i \Delta u_{m+i-1} = a_0 \Delta u_1 + a_1 \Delta u_2 + \cdots + a_n \Delta u_n
\]

\[
\vdots
\]

\[
y''''_m = y''''_n = \sum_{i=1}^{n} h_i \Delta u_{m+i-1} = a_0 \Delta u_1 + a_1 \Delta u_2 + \cdots + a_n \Delta u_n
\]

Controller Design Method (DMC)

- Objective
  - Minimize errors between future set points and predictions

\[
\min_{\Delta u} (r_{n+T} - y_{n+T})^2 = \min_{\Delta u} \begin{bmatrix} r_{n+T} - y_{n+T} \\
\vdots \\
\end{bmatrix}^2 = \min_{\Delta u} (A \Delta u + E)^T (A \Delta u + E)
\]

where

\[
P_i = \sum_{j=1}^{J} S_j \quad (i = 1, 2, \ldots, P)
\]

\[
S_j = \sum_{i=1}^{J} h_i \Delta u_{m+i-1} \quad (i = 1, 2, \ldots, P)
\]

- $S_j$: the incremental effect of the past (previously implemented) movements of input on the ($m+j$)-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.

\[
y'_m = y'_n = \sum_{i=1}^{n} h_i \Delta u_{m+i-1} = a_0 \Delta u_1 + a_1 \Delta u_2 + \cdots + a_n \Delta u_n
\]

\[
y''_m = y''_n = \sum_{i=1}^{n} h_i \Delta u_{m+i-1} = a_0 \Delta u_1 + a_1 \Delta u_2 + \cdots + a_n \Delta u_n
\]

\[
y'''_m = y'''_n = \sum_{i=1}^{n} h_i \Delta u_{m+i-1} = a_0 \Delta u_1 + a_1 \Delta u_2 + \cdots + a_n \Delta u_n
\]

\[
\vdots
\]

\[
y''''_m = y''''_n = \sum_{i=1}^{n} h_i \Delta u_{m+i-1} = a_0 \Delta u_1 + a_1 \Delta u_2 + \cdots + a_n \Delta u_n
\]

\[
\uparrow \text{ Depend on only future} \quad \downarrow \text{ Depend on only past}
\]

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

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

- Solution

\[
-A \Delta u + E = 0 \quad \Rightarrow \quad \Delta u = (A^T A)^{-1} E
\]

Some inverse of $A$
• If $U=V$ and $A$ is invertible,
  \[ \Delta u = A^{-1} \bar{E}' \]
  It gives no steady-state offset since it has integral action.

• If $U<V$ (A is not invertible),
  \[ \Delta u = (A^T A)^{-1} A^T \bar{E}' = K_c \bar{E}' \]
  $A^+$: Left pseudoinverse of $A$

Optimization concept
\[
\min J = \bar{E}' \bar{E} = \min (-A \Delta u + \bar{E}' (-A \Delta u + \bar{E}'))
\]
\[ \frac{\partial J}{\partial \Delta u} = -2A' (-A \Delta u + \bar{E}') + 2(A' A) \Delta u = 0 \]
\[ \Delta u = (A' A)^{-1} A' \bar{E}' \]

• Adjustable parameters of MPC (Tuning parameters)
  - Weighting matrices
    - If $W_1 >> W_2$, the most important objective is to minimize the error of the process outputs.
    - If $W_1 << W_2$, the most important objective is to minimize the input movements.
    - 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 reducing the error if the action is not too aggressive.

• The $W_c$ is called input penalty or input move suppression factor.
  Typically, use $W_r = I$ and $W_c = \theta I$ and adjust $\theta$.

• 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 $T_M > \text{open-loop settling time}$
    - $T$ 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 $U_M > U_{\text{min}}$
    - 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
  \[ \bar{E} = -A \Delta u + \bar{E}' \]
  where
  \[ \bar{E} = \begin{bmatrix} \bar{E}_1' \\ \bar{E}_2' \end{bmatrix} \]
  \[ \Delta u = \begin{bmatrix} \Delta u_1' \\ \Delta u_2' \end{bmatrix} \]
  \[ A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \]

• 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**
  
  \[
  \min J = (\hat{E}^T \hat{W}_i \hat{E} + \Delta u^T \hat{W}_d \Delta u) \\
  \text{subject to} \quad u^T \leq u \leq u^T \\
  y^T \leq y \leq y^T \\
  \text{and other constraints}
  \]
  
- **Solve this optimization problem in QP**
  - DMC by DMCC used LP

**Model Algorithmic Control (MAC)**

- **Process model:**
  - based on $u$ not $\Delta u$

- **Set point:**
  - First-order approach to set point
  - $r_k = \alpha y_{k-1} + (1-\alpha) y_{k-1}$
  - Speed of response is determined by $\alpha$ (tuning parameter)

- **Tuning parameters**
  - Speed of desired response
  - $U = \nu$ (fixed, not used as tuning parameters)
  - $\nu$ is chosen so that $\nu \Delta t$ (open-loop settling time)
  - Time varying weight: $J = \sum \omega(k) (e(k))^2$

- **Solution is obtained using QP**

**Comments on MPC**

- **Implementation**
  - Update the prediction model based on the current measurement.
  - Calculate $U$ moves from the optimization and implement the first input moves and throw out the rest.

- **The MPC is minimizing the error between the set point and predicted output.**
  - In the prediction, the measurement is incorporated and it works as a feedback.
  - No steady-state offset: integrator in the control law

- **Disturbance Model can be added**
  - Known measured disturbance can be incorporated by adding disturbance model in the same manner.

**Example 1: Blending control system**

- **Objectives:**
  - Control the composition of A and B
  - Control total flow if possible

- **Constraints:**
  - Flow rates are limited

**Classical solution**

**MPC solution**

\[
\min \sum (r_{ij}(k+i) - r_{ij}^*)^2 + (r_{ij}(k+i) - r_{ij}^*)^2 \\
\quad + \nu (k+i) - q_i^*)^2
\]

subject to $u_{ij}^T \leq \nu_j \leq u_{ij}^T$ $(i = 1, 2, 3)$

\[
(j = k, \ldots, k+U-1) \\
\nu \ll 1
\]
**Example 2: Heavy Oil Fractionator**

- Keep $y_i \geq T_{min}$
- Control the two compositions $y_1$ and $y_2$
- Minimize $u_t$ to maximize the heat recovery.

$$
\min \sum_{t=0}^{T} (y_t(k+i(k)) - y_t)^2 + n_i(h_t)^2
$$

subject to $u_t^2 \leq u_t \ (t=1,2,3) \\
(\Delta k \cdot k \cdot U + 1) \\
y_t \geq T_{min}$

**Identification of Models**

- **FSR or FIR models: use step or pulse test**
  - Assume operation at steady state
  - Make change in input $\Delta u$ (or $\Delta U$)
    - If $\Delta u$ is too small, output change may not noticeable
    - If $\Delta u$ is too large, linearity may not hold
  - Measure output at regular intervals $\Delta t$
    - The $\Delta t$ 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)
  - $u = [u_t \ u_{t-1} \ ... \ u_{t-n}]$
  - $y = [y_t \ y_{t-1} \ ... \ y_{t-n}]$

- Get the FIR model
  $$
j = \hat{h}_0 u_{t-1} + \hat{h}_1 u_{t-2} + ... + \hat{h}_r u_{t-n} \ 
\rightarrow y_t = \sum_{i=0}^{r} \hat{h}_i u_{t-i}$$

- Minimize the error between measurements and output, $d_t = y_t - \hat{y}_t$

$$
\begin{bmatrix}
y_{t-1} \\
y_{t-2} \\
y_{t-3}\end{bmatrix} = 
\begin{bmatrix}
\hat{h}_0 & \hat{h}_1 & \hat{h}_2 & \hat{h}_3 \\
\hat{h}_1 & \hat{h}_2 & \hat{h}_3 & \hat{h}_4 \\
\hat{h}_2 & \hat{h}_3 & \hat{h}_4 & \hat{h}_5 \\
\hat{h}_3 & \hat{h}_4 & \hat{h}_5 & \hat{h}_6 \\
\hat{h}_4 & \hat{h}_5 & \hat{h}_6 & \hat{h}_7 \\
\hat{h}_5 & \hat{h}_6 & \hat{h}_7 & \hat{h}_8 \\
\hat{h}_6 & \hat{h}_7 & \hat{h}_8 & \hat{h}_9 \\
\hat{h}_7 & \hat{h}_8 & \hat{h}_9 & \hat{h}_{10} \\
\end{bmatrix} \begin{bmatrix}
d_1 \\
d_2 \\
d_3 \\
d_4 \\
d_5 \\
d_6 \\
d_7 \\
d_8 \\
d_9 \\
\end{bmatrix}$$

$$
d = y - Uh
$$

$$
\min_{h} d^T d = \min_{h} (y - Uh)^T (y - Uh) \Rightarrow h = (U^T U + \alpha I)^{-1} U^T y
$$

**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 $U^T U$ is singular, the inverse doesn't exist and identification fails. (Need persistent excitation condition)

- When the number of coefficients is large, $U^T U$ can be easily singular (or nearly singular). To avoid the numerical, a regularization term is added the the cost function. (ridge regression)

$$
\min_{h} (y - Uh)^T (y - Uh) + \alpha h^T h \Rightarrow h = (U^T U + \alpha I)^{-1} U^T y
$$
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.,
    \[ \hat{y}_k = \frac{y_k - \bar{y}_c}{c_y} \text{ where } \bar{y}_c = \frac{\sum_{i=1}^{N} y_i}{N} \]
    \[ \hat{u}_k = \frac{u_k - \bar{u}_c}{c_u} \text{ where } \bar{u}_c = \frac{\sum_{i=1}^{N} u_i}{N} \]
  - Use the given steady-state values of the variables instead to compute the deviation variables, i.e.,
    \[ \hat{y}_k = \frac{y_k - y_{ss}}{c_y} \text{ and } \hat{u}_k = \frac{u_k - u_{ss}}{c_u} \]
  - 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.
    \[ \Delta y_k = \frac{y_k - y_{ss}}{c_y} \text{ and } \Delta u_k = \frac{u_k - u_{ss}}{c_u} \]
    \[ \Rightarrow \text{Identification for } \Delta y_k \text{ and } \Delta u_k \]
  - 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.