MODEL PREDICTIVE CONTROL

Professor Dae Ryook Yang

Spring 2020
Dept. of Chemical and Biological Engineering
Korea University

* Some parts are from Jay H. Lee’s lecture notes
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
• 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
**Exemplary Algorithm**

\[
\begin{align*}
\min_{u_t()} & \int_t^{t+p} l_1[\text{Error}(\tau)] + l_2[\text{Input}(\tau)]d\tau \\
U(\cdot) & \in U, \quad Y_t(\cdot) \in Y
\end{align*}
\]

- Objective function
- Constraints
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

![Diagram of Blending System Model]

- Control $r_A$ and $r_B$.
- Control $q$ if possible.
- Flowrates of additives are limited.

MPC:
Solve at each time $k$

$p$ = Size of prediction window

$$\begin{align*}
\min_{u_1(j), u_2(j), u_3(j)} & \sum_{i=1}^{p} (r_A(k+i | k) - r_A^*)^2 \\
& + (r_B(k+i | k) - r_B^*)^2 \\
& + \gamma(q(k+i | k) - q^*)^2
\end{align*}$$

$$(u_i)_{\min} \leq u_i(j) \leq (u_i)_{\max}, i = 1, \ldots, 3,$$
$$\gamma \ll 1$$
• Example 2: Heavy Oil Fractionator

- Keep $y_7 \geq T_{\text{min}}$
- Control the two compositions $y_1$ and $y_2$
- Minimize $u_3$ 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

Traditional Control:
- Design and Tuning of Controller
  - Process Analysis

MPC:
- Modeling and Identification
  - Control Specification
Challenges

• Efficient identification of control-relevant model

• Managing the sometimes exorbitant on-line computational load
  – Nonlinear models $\rightarrow$ Nonlinear Programs (NLP)
  – Hybrid system models (continuous dynamics + discrete events or switches, e.g., pressure swing adsorption) $\rightarrow$ Mixed Integer Programs (MINLP)
  – 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
    \[ y(k) = \sum_{i=0}^{k} h(i)u(k - i) \]
  - 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)
    \[ y_k = a_k \left( u_k = 1, \forall k \geq 0 \right) \]
  - Using superposition principle for arbitrary input changes
    \[ u_k = \Delta u_0 + \Delta u_1 + \cdots + \Delta u_k \text{ where } \Delta u_i = u_i - u_{i-1} \]
    \[ y_k = y_0 + y_k \bigg|_{\Delta u_0} + y_k \bigg|_{\Delta u_1} + \cdots + y_k \bigg|_{\Delta u_{k-1}} = y_0 + a_k \Delta u_0 + a_{k-1} \Delta u_1 + \cdots + a_1 \Delta u_{k-1} \]
• **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

\[ h_i = a_i - a_{i-1} \quad (i = 1, 2, \ldots, T) \]

\[ h_0 = 0 \]

\[ y_n = y_0 + \sum_{i=1}^{T} a_i \Delta u_{n-i} = y_0 + \sum_{i=1}^{T} a_i (u_{n-i} - u_{n-i-1}) \]

\[ = y_0 + (a_1 u_{n-1} - a_1 u_{n-2}) + (a_2 u_{n-2} - a_2 u_{n-3}) + \cdots + (a_n u_1 - a_n u_0) + (a_n u_0 - a_n u_{-1}) + \cdots \]

\[ = y_0 + a_1 u_{n-1} + (a_2 - a_1) u_{n-2} + \cdots + (a_n - a_{n-1}) u_1 + (a_n - a_{n-1}) u_0 + \cdots \]

\[ = y_0 + (a_1 - a_0^0) u_{n-1} + (a_2 - a_1) u_{n-2} + \cdots + (a_n - a_{n-1}) u_1 \]

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

(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)

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

\[y = A \Delta u\]

- **A**: Dynamic matrix
Single-Step Prediction

• From the FIR model

\[ \hat{y}_n = y_0 + \sum_{i=1}^{T} h_i u_{n-i} \quad \hat{y}_{n+1} = y_0 + \sum_{i=1}^{T} h_i u_{n+1-i} \]

\[ \Rightarrow \hat{y}_{n+1} = \hat{y}_n + \sum_{i=1}^{T} h_i \Delta u_{n+1-i} \] (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 $V \geq U$

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

Dynamic Matrix, $A$
where

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

\[ S_j = \sum_{l=1}^{T} h_{l} \Delta u_{n+j-l} \ (i = 1, 2, \ldots, V) \]

- \( S_j \): 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.

\[
y_{n+1}^* = y_n + \sum_{i=1}^{T} h_i \Delta u_{n+1-i} = h_2 \Delta u_n + y_n + \sum_{i=2}^{T} h_i \Delta u_{n+1-i} = a_1 \Delta u_n + y_n + \sum_{i=2}^{T} h_i \Delta u_{n+1-i}
\]

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

\[
y_{n+3}^* = y_{n+2}^* + \sum_{i=1}^{T} h_i \Delta u_{n+3-i}
\]

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

\[
y_{n+V}^* = y_{n+V-1}^* + \sum_{i=1}^{T} h_i \Delta u_{n+V-i} = a_V \Delta u_n + a_{V-1} \Delta u_{n+1} + \cdots + a_{V-U+1} \Delta u_{n+U-1} + y_n + \sum_{i=V+1}^{T} h_i \Delta u_{n+V-i} + \sum_{i=3}^{T} h_i \Delta u_{n+2-i} + \sum_{i=2}^{T} h_i \Delta u_{n+1-i}
\]

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

\[\text{ Depend on only future } \quad \text{ Depend on only past}\]
Controller Design Method (DMC)

• Objective
  - Minimize errors between future set points and predictions

\[
\hat{E} = \begin{bmatrix}
  r_{n+1} - y_{n+1}^*
  \\
  r_{n+2} - y_{n+2}^*
  \\
  \vdots
  \\
  r_{n+N} - y_{n+N}^*
\end{bmatrix} = r - (A\Delta u + y_n e + P) = -A\Delta u + \hat{E}'
\]

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

where

\[
\hat{E}' = \begin{bmatrix}
  r_{n+1} - y_n - P_1 \\
  r_{n+2} - y_n - P_2 \\
  \vdots \\
  r_{n+N} - y_n - P_N
\end{bmatrix}
\]

Open-loop prediction error based only on past control action

• Solution

\[-A\Delta u + \hat{E}' = 0 \Rightarrow \Delta u = (A^*)^{-1}\hat{E}'\]

Some inverse of A
• If $U=V$ and $A$ is invertible,

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

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

• If $U<V$ (A is not invertible),

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

$A^+$: Left pseudoinverse of $A$

$A^+A=I$: identity matrix

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

• Optimization concept

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

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

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

$$\min J = (\hat{E}^T W_1 \hat{E} + \Delta u^T W_2 \Delta u)$$

$$\frac{\partial J}{\partial \Delta u} = -2A^T W_1 (-A\Delta u + \hat{E}') + 2W_2 \Delta u = 2((A^T W_1 A + W_2) \Delta u - A^T W_1 \hat{E}') = 0$$

$$\Rightarrow \Delta u = (A^T W_1 A + W_2)^{-1} A^T W_1 \hat{E}'$$
• **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 $W_2 = f^2 I$ 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 $U \Delta t \equiv t_{60}$
    - 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**

\[
\hat{E} = -A\Delta u + \hat{E}'
\]

where

\[
\hat{E} = [\hat{E}_1; \hat{E}_2] \quad \Delta u = [\Delta u_1; \Delta u_2]
\]

\[
A = \begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix}
\]

• **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

\[ \min J = (\hat{E}^T W_1 \hat{E} + \Delta u^T W_2 \Delta u) \]

subject to \( u^L \leq u \leq u^U \)
\[ y^L \leq y \leq y^U \]
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
    \[
    r_{k+i}^* = \alpha y_{k+i-1} + (1 - \alpha)r_{k+i}
    \]
  - Speed of response is determined by \( \alpha \) (tuning parameter)

- **Tuning parameters**
  - Speed of desired response
  - \( U=V \) (fixed, not used as tuning parameters)
  - \( V \) is chosen so that \( V \Delta t \approx \) (open-loop settling time)
  - Time varying weight: \( J = \sum_{i=1}^{V} w(i)e(k + i)^2 \)

- **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

\[
\min \sum_{i=1}^{V} (r_A(k + i | k) - r_A^*)^2 + (r_B(k + i | k) - r_B^*)^2 \\
+ w(q(k + i | k) - q^*)^2
\]

subject to \(u_i^L \leq u_i(j) \leq u_i^U\) \((i = 1, 2, 3)\)
\((j = k, \ldots, k + U - 1)\)
\(w \leq 1\)
• Example 2: Heavy Oil Fractionator

- Keep $y_7 \geq T_{\text{min}}$
- Control the two compositions $y_1$ and $y_2$
- Minimize $u_3$ to maximize the heat recovery.

$$
\min \sum_{i=1}^{V} (y_1(k+i|k)-y_1^*)^2 + (y_2(k+i|k)-y_2^*)^2 \\
+ w_1(u_3)^2
$$

subject to $u_i^L \leq u_i(j) \leq u_i^U \ (i = 1, 2, 3)$

$(j = k, \ldots, k + U - 1)$

$y_7 \geq T_{\text{min}}$
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)
    \[ u = [u_1 \ u_2 \ \cdots \ u_M] \quad y = [y_1 \ y_2 \ \cdots \ y_M] \]
  
  - 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, \( d_k = y_k - \tilde{y}_k \)
    
    \[
    \begin{bmatrix}
    y_1 \\
    y_2 \\
    \vdots \\
    y_M \\
    \end{bmatrix} = \begin{bmatrix}
    u_0 & u_{-1} & \cdots & u_{1-N} \\
    u_1 & u_0 & \cdots & u_{2-N} \\
    \vdots & \vdots & \ddots & \vdots \\
    u_{M-1} & u_{M-2} & \cdots & u_{M-N} \\
    \end{bmatrix} \begin{bmatrix}
    h_1 \\
    h_1 \\
    \vdots \\
    h_N \\
    \end{bmatrix} + \begin{bmatrix}
    d_1 \\
    d_2 \\
    \vdots \\
    d_M \\
    \end{bmatrix} \quad d = y - Uh \\
    \]

  \[
  \min_h d^T d = \min_h (y - Uh)^T (y - Uh) \Rightarrow h = (U^T U)^{-1} U^T y \\
  \]
• **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.,
    \[ \tilde{y}_k = \frac{y_k - y_{\text{ref}}}{c_y} \quad \text{where} \quad y_{\text{ref}} = \frac{\sum_{i=1}^{M} y_i}{M} \]
    \[ \tilde{u}_k = \frac{u_k - u_{\text{ref}}}{c_u} \quad \text{where} \quad u_{\text{ref}} = \frac{\sum_{i=1}^{M} u_i}{M} \]
  
  - Use the given steady-state values of the variables instead to compute the deviation variables, i.e.,
    \[ \tilde{y}_k = \frac{y_k - y_{ss}}{c_y} \quad \text{and} \quad \tilde{u}_k = \frac{u_k - u_{ss}}{c_u} \]
    
    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.
    \[ \Delta y_k = \frac{y_k - y_{k-1}}{c_y} \quad \text{and} \quad \Delta u_k = \frac{u_k - u_{k-1}}{c_u} \]
    
    \[ \Rightarrow \text{Identification for} \, \Delta y_k \, \text{and} \, \Delta u_k \]

  - 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.

  The two filters should be same.