Model Predictive Control

WebCAST Seminar
May 7, 2007

Professor Jay H. Lee
School of Chemical and Biomolecular Engineering
Georgia Institute of Technology
©Jay H. Lee, 2007
Overview

- Popularized in the late 70s and early 80s in refineries
- Standard APC method for refineries and petrochemical plants
- 4500+ reported industrial applications (Yr. 2000)
- Many vendors marketing software and engineering service
  – Aspen Tech, Honeywell, Invensys, etc.
- Strong theoretical basis and systematic design for stability and performance
Some Key Features

- **Computer based**: Sampled-data control
- **Model based**: Requires a dynamic process model (fundamental or empirical)
- **Feedback Update**: Model updated using on-line measurements.
- **Predictive**: Makes explicit prediction of the future time behavior of CVs within a chosen window.
Some Key Features (2)

- **Optimization Based**: Performs optimization (numerical search) on-line for optimal control adjustments.

  \[
  \min_{u_i} \sum_{i=0}^{p} \phi_i (x_i, u_i) \quad ? \quad u_0 = \mu(x_0) \\
  g_i (x_i, u_i) \geq 0 \quad \rightarrow \quad \text{HJB Eqn.}
  \]

  No *explicit* form of control law – just model, objective function, and constraints are specified.

- **Integrated** constraint handling and economic optimization with regulatory and servo control.

- **Receding Horizon Control**: Repeats the prediction and optimization at each sample time step to update the optimal input trajectory after a feedback update.
Analogy to Chess Playing

Opponent (The Disturbance)

I (The Controller)

The Opponent’s Move

New State
Industrial Use of MPC
Industrial Use of MPC

- Some trial of computer based control during 50s-60s (e.g., Standard Oil / IBM).
- Reappeared at Shell Oil and other refineries during late 70s and early 80s. – easier, cheaper implementation enabled by advances in microprocessors.
- Various commercial software
- Tens of thousands of worldwide installations
- Predominantly in the oil and petrochemical industries but the range of applications is expanding.
- Models used are predominantly empirical models developed through plant testing.
- The technology is not only for multivariable control, but for most economic operation within constraint boundaries.
### Result of a Survey in 1999 (Qin and Badgwell)

<table>
<thead>
<tr>
<th>Area</th>
<th>Aspen Technology</th>
<th>Honeywell Hi-Spec</th>
<th>Adersa¹</th>
<th>Invensys</th>
<th>SGS²</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refining</td>
<td>1200</td>
<td>480</td>
<td>280</td>
<td>25</td>
<td></td>
<td>1985</td>
</tr>
<tr>
<td>Petrochemicals</td>
<td>450</td>
<td>80</td>
<td>-</td>
<td>20</td>
<td></td>
<td>550</td>
</tr>
<tr>
<td>Chemicals</td>
<td>100</td>
<td>20</td>
<td>3</td>
<td>21</td>
<td></td>
<td>144</td>
</tr>
<tr>
<td>Pulp and Paper</td>
<td>18</td>
<td>50</td>
<td>-</td>
<td>-</td>
<td></td>
<td>68</td>
</tr>
<tr>
<td>Air &amp; Gas</td>
<td>-</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Utility</td>
<td>-</td>
<td>10</td>
<td>-</td>
<td>4</td>
<td></td>
<td>14</td>
</tr>
<tr>
<td>Mining/Metallurgy</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>16</td>
<td></td>
<td>37</td>
</tr>
<tr>
<td>Food Processing</td>
<td>-</td>
<td>-</td>
<td>41</td>
<td>10</td>
<td></td>
<td>51</td>
</tr>
<tr>
<td>Polymer</td>
<td>17</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>Furnaces</td>
<td>-</td>
<td>-</td>
<td>42</td>
<td>3</td>
<td></td>
<td>45</td>
</tr>
<tr>
<td>Aerospace/Defense</td>
<td>-</td>
<td>-</td>
<td>13</td>
<td>-</td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>Automotive</td>
<td>-</td>
<td>-</td>
<td>7</td>
<td>-</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>Unclassified</td>
<td>40</td>
<td>40</td>
<td>1045</td>
<td>26</td>
<td></td>
<td>450</td>
</tr>
<tr>
<td>Total</td>
<td>1833</td>
<td>696</td>
<td>1438</td>
<td>125</td>
<td></td>
<td>4542</td>
</tr>
</tbody>
</table>

First App.:
- DMC:1985
- IDCOM-M:1987
- OPC:1987
- PCT:1984
- RMPCT:1991
- IDCOM:1973
- HIECON:1986

Largest App:
- 603x283
- 225x85
- 31x12
Linear MPC Vendors and Packages

- **Aspentech**
  - DMCplus
  - DMCplus-Model

- **Honeywell**
  - Robust MPC Technology (RMPCT)

- **Adersa**
  - Predictive Functional Control (PFC)
  - Hierarchical Constraint Control (HIECON)
  - GLIDE (Identification package)

- **MDC Technology (Emerson)**
  - SMOC (licensed from Shell)
  - Delta V Predict

- **Predictive Control Limited (Invensys)**
  - Connoisseur

- **ABB**
  - 3d MPC
Result of A Survey for Nonlinear MPC (Qin and Badgwell)

<table>
<thead>
<tr>
<th>Area</th>
<th>Adersa Technology</th>
<th>Aspen Controls</th>
<th>Continental Controls</th>
<th>DOT Products</th>
<th>Pavilion Technologies</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air and Gas</td>
<td></td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>Chemicals</td>
<td>2</td>
<td></td>
<td></td>
<td>5</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>Food Processing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Polymers</td>
<td>1</td>
<td></td>
<td></td>
<td>5</td>
<td>15</td>
<td>21</td>
</tr>
<tr>
<td>Pulp &amp; Paper</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Refining</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>Utilities</td>
<td></td>
<td>5</td>
<td></td>
<td>2</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>Unclassified</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>3</td>
<td>6</td>
<td>36</td>
<td>5</td>
<td>43</td>
<td>93</td>
</tr>
</tbody>
</table>
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)

![Diagram showing the integration of Supervisory Control, Process Optimization, and Advanced MV Control with Low-level PID Loops and selectors, switches, delay compensations, anti-windups, decouplers, etc.]

More and more optimization is done at the MPC level.
Example 1: Blending System Control

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

**MPC:** Solve at each time $k$.

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{p} \left( (r_A(k+i | k) - r_A^*)^2 + (r_B(k+i | k) - r_B^*)^2 + \gamma (q(k+i | k) - q^*)^2 \right) \\
(u_i)_{\text{min}} & \leq u_i(j) \leq (u_i)_{\text{max}}, \quad i = 1, \ldots, 3, \\
\gamma & \ll 1
\end{align*}
\]

$p =$ Size of prediction window
Advantages of MPC over Traditional APC

• Integrated solution
  – Automatic constraint handling
  – Feedforward / feedback control
  – No need for decoupler 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

↓

Model Predictive Control
Conflict / Synergy Between Optimization and Control

Acknowledgment:
Aspen Technology
Bi-Level Optimization Used in MPC

Steady-State Optimization (LP)

Optimal setting values for the inputs and outputs (setpoints)

Economics Based Objective
(Maximum profit or throughput, minimum utility)
Control Based Constraints

Minimization of Error
(=Setpoint – Output and Input)
Constraints on actuator limits and safety-sensitive variables.

Dynamic Optimization (QP)

Steady-state Prediction Model

New Measurements (Feedback Update)

Adjustments to setpoints of low level loops or control valves
New Operational Hierarchy and Role of MPC

Large-scale (e.g., plantwide) optimization involving rigorous nonlinear models (AspenPlus)

Move the plant to the current optimal condition fast and smoothly w/o violating constraints

Local optimization + control
An Exemplary Application(1)

Acknowledgment:
Mitsubishi Chemicals
An Exemplary Application (2)

Acknowledgment: Mitsubishi Chemicals
Linear MPC
Popular Linear Model Structures

• Finite Impulse Response Model

\[ y(k) = h_1 u(k-1) + \cdots + h_N u(k-m) \]

• Truncated Step Response Model

\[ x(k+1) = M_1 x(k) + S_{\text{shift}} \Delta u(k) + S_{\text{step response}} \Delta u(k) \]

• Transfer Function Model

\[ y(k) = a_1 y(k-1) + \cdots + a_n y(k-n) + b_1 u(k-1) + \cdots + b_m u(k-m) \Rightarrow G(q) = \frac{b_1 q^{-1} + \cdots + b_m q^{-m}}{1 - a_1 q^{-1} - \cdots - a_n q^{-n}} \]

• State Space Model

\[
\begin{align*}
    x(k+1) &= A x(k) + B u(k) \\
    y(k) &= C x(k)
\end{align*}
\]
Key: Prediction Equation

- **Step Response Model**

Predicted future output samples

\[
\begin{bmatrix}
  y_{k+1|k} \\
  \vdots \\
  y_{k+p|k}
\end{bmatrix}
\]

The “state” stored in dynamic memory

\[
\begin{bmatrix}
  y_1(k) \\
  \vdots \\
  y_p(k)
\end{bmatrix}
\]

Future input moves (to be decided)

\[
\begin{bmatrix}
  \Delta u(k) \\
  \vdots \\
  \Delta u(k+p-1)
\end{bmatrix}
\]

Dynamic Matrices (made of step response coefficients)

\[
\Omega^d \begin{bmatrix}
  \Delta d(k) \\
  \vdots \\
  \Delta d(k+p-1)
\end{bmatrix}
\]

Feedback Error Correction

\[
e(k) = y_m(k) - y(k)
\]

Feedforward term: new measurement

(Assume \(\Delta d(k+1) = \ldots = \Delta d(k+p-1) = 0\)
Prediction Equation: General

- Regardless of model form, one gets the prediction equation in the form of

\[
\begin{bmatrix}
  y_{k+1|k} \\
  \vdots \\
  y_{k+p|k}
\end{bmatrix} = L^x x(k) + L^d \Delta d(k) + L^e e(k) + L^u u(k) \\
\begin{bmatrix}
  \Delta u(k) \\
  \Delta u(k + p - 1)
\end{bmatrix}
\]

\(\equiv \begin{bmatrix}
  y(k) \\
  \vdots \\
  y(k)
\end{bmatrix} + b(k)\]

- Assumptions
  - Measured DV (d) remains constant at the current value of d(k)
  - Model prediction error (e) remains constant at the current value of e(k)
Measurement Correction of State

State: Compact representation Of the past input and measurement record

State Update

New Input Move (JustImplemented)

Current State

Prediction

Prediction Model for Future Outputs

Future Input Moves (To Be Determined)

Measurement Correction

Feedback / Feedforward Measurements

To Optimization
State Update Equation

\[ \hat{x}(k + 1) = \Phi \hat{x}(k) + \Gamma^u \Delta u(k) + \Gamma^d \Delta d(k) + K \left( y_m(k) - \Xi \hat{x}(k) \right) \]

- \( K \) is the update gain matrix that can be found in various ways
  - *Pole placement*: Not so effective with systems with many states (most chemical processes)
  - *Kalman filtering*: Requires a **stochastic** model of form

\[ x(k + 1) = \Phi x(k) + \Gamma^u \Delta u(k) + \Gamma^d \Delta d(k) + w(k) \]
\[ y(k) = \Xi x(k) + v(k) \]

White noises of known covariances
Effect of unmeasured disturbances and noise
Prediction Equation

\[
\begin{bmatrix}
\Delta d(k) \\
\vdots \\
\Delta d(k + p - 1)
\end{bmatrix}
+ \begin{bmatrix}
e(k) \\
\vdots \\
e(k)
\end{bmatrix}
\]

\[
\Omega^d
\begin{bmatrix}
\Delta d(k) \\
\vdots \\
\Delta d(k + p - 1)
\end{bmatrix}
+ \begin{bmatrix}
e(k) \\
\vdots \\
e(k)
\end{bmatrix}
\]

Contains past feedback measurement corrections

Additional measurement correction NOT needed here!
What Are the Advantages of Using a State Estimator (Observer)?

• Can handle **unstable** processes
  – Integrating processes, run-away processes

• **Cross-channel** update
  – More effective update of output channels with delays or measurement problems based on other channels.

• Systematic handling of **multi-rate** measurements

• Optimal **extrapolation of output error** and **filtering of noise** (based on the given stochastic system model)
Objective Function

• Minimization Function: Quadratic cost (as in DMC)

\[
V(k) = \sum_{i=1}^{p} (y_{k+i|k} - y^*)^T \Lambda^v (y_{k+i|k} - y^*) + \sum_{i=0}^{m-1} \Delta u^T (k+i) \Lambda^u \Delta u(k+i)
\]

  - Consider only m input moves by assuming \( \Delta u(k+j)=0 \) for \( j \geq m \)
  - Penalize the tracking error as well as the magnitudes of adjustments

• \( V(k) \) is a quadratic function of \( \Delta u(k+j), j=0,\ldots,m-1 \)
Objective Function

\[ V(k) = \sum_{i=1}^{p} (y_{k+i|k} - y^*)^T \Lambda^y (y_{k+i|k} - y^*) + \sum_{i=0}^{m-1} \Delta u^T (k+i) \Lambda^u \Delta u(k+i) \]

\[ V(k) = (Y(k) - Y^*)^T \text{diag}(\Lambda^y)(Y(k) - Y^*) + \Delta U_m^T(k) \text{diag}(\Lambda^u) \Delta U_m(k) \]

Substitute

\[
\begin{bmatrix}
  y_{k+1|k} \\
  \vdots \\
  y_{k+p|k} \\
  Y(k)
\end{bmatrix} = \underbrace{L^x x(k) + L^d \Delta d(k) + L^e e(k) + L^u}_{\text{known } \equiv b(k)} + \Delta u(k) \\
\begin{bmatrix}
  \Delta u(k) \\
  \Delta u(k + p - 1) \\
  \Delta U(k)
\end{bmatrix}
\]

\[ V(k) = \Delta U_m^T(k) H \Delta U_m(k) + g^T(k) \Delta U_m(k) + c(k) \]
Constraints

\[ u_{\text{min}} \leq u(k + \ell | k) \leq u_{\text{max}} \]
\[ |\Delta u(k + \ell | k)| \leq \Delta u_{\text{max}}, \quad \ell = 0, \ldots, m - 1 \]
\[ y_{\text{min}} \leq y(k + j | k) \leq y_{\text{max}}, \quad j = 1, \ldots, p \]

Substitute the prediction equation and rearrange to

\[ C\Delta U_m(k) \geq h(k) \]
Optimization Problem

• Quadratic Program

\[
\min_{\Delta U_m(k)} \Delta U_m^T(k) H \Delta U_m(k) + g^T(k) \Delta U_m(k)
\]

such that \( C \Delta U_m(k) \geq h(k) \)

• Unconstrained Solution

\[
\Delta U_m(k) = -\frac{1}{2} H^{-1} g(k)
\]

• Constrained Solution
  – Must be solved numerically.
Quadratic Program

- Minimization of a quadratic function subject to linear constraints.
- Convex and therefore fundamentally tractable.
- Solution methods
  - Active set method: Determination of the active set of constraints on the basis of the KKT condition.
  - Interior point method: Use of barrier function to “trap” the solution inside the feasible region, Newton iteration
- Solvers
  - Off-the-shelf software, e.g., QPSOL
  - Customization is desirable for large-scale problems.
Bi-Level Optimization

Steady-State Optimization (Linear Program)

\[
\min_{u_s(k)} L(y_{\infty|k}, u_s(k))
\]

\[
C_s \begin{bmatrix} y_{\infty|k} \\ u_s(k) \end{bmatrix} \geq c_s(k)
\]

\[
u_s(k) = u(k-1) + \Delta u(k) + \cdots + \Delta u(k+m-1)
\]

\[
y_{\infty|k} = b_s(k) + L_s \Delta u_s(k)
\]

Optimal Setting Values (setpoints)

\[
y_{\infty|k}^*, u_s^*(k)
\]

To Dynamic Optimization (Quadratic Program)

Steady-State Prediction Eqn.

State Feedback Error

Feedforward Measurement

Dynamic Prediction Eqn.
Stability
Classical Optimal Control - LQR

- Quadratic objective

\[ \sum_{i=0}^{p} x_i^T Q x_i + \sum_{i=0}^{m-1} u_i^T R u_i \]

- Fairly general formulation:
  - State regulation, Output regulation, Setpoint tracking

- Unconstrained $\infty$ horizon problem has an analytical solution.
  $\rightarrow$ Linear state feedback law (Kalman’s LQR)

- Stability guaranteed for stabilizable system

- Solution is smooth with respect to the parameters

- BUT, presence of inequality constraints $\rightarrow$ no analytical solution via Riccati equation.

Linear State Space System Model

\[ x_{k+1} = A x_k + B u_k \]
\[ y_k = C x_k \]
Why Has Stability Analysis of MPC Been Difficult?

- MPC=Nonlinear state feedback control law
- Implicitly defined by an optimization
  - No explicit expression for the MPC control law
- Use of an observer
  - Lack of separation principle
Use of $\infty$ Prediction Horizon – Why?

• Stability guarantee
  – The optimal cost function can be shown to be the control Lyapunov function.

• Less parameters to tune

• More consistent, intuitive effect of weight parameters

• Close connection with the classical optimal control methods, e.g., LQG control
Step Response Model Case

\[ V(k) = \sum_{i=1}^{\infty} (y_{k+i|k} - y^*)^T \Lambda^y (y_{k+i|k} - y^*) + \sum_{i=0}^{m-1} \Delta u^T (k+i) \Lambda^u \Delta u(k+i) \]

\[ V(k) = \sum_{i=1}^{m+n-1} (y_{k+i|k} - y^*)^T \Lambda^y (y_{k+i|k} - y^*) + \sum_{i=0}^{m-1} \Delta u^T (k+i) \Lambda^u \Delta u(k+i) \]

with extra constraint \( y_{k+m+n-1|k} = y^* \)

Must be at \( y^* \) for the cost to be bounded.
Additional Comments

• Previously, we assumed \textbf{finite} settling time.

• Can be generalized to general state-space models
  – More complicated procedure to turn the $\infty$-horizon problem into a finite horizon problem
  – Requires solving a Lyapunov equation to get the terminal cost matrix
  – Also, must make sure that output constraints will be satisfied beyond the finite horizon $\rightarrow$ construction of an output admissible set.

• Use of a \textit{sufficiently large} horizon ($p \approx m + \text{the settling time}$) should have a similar effect.

• Can we always satisfy the settling constraint?
  – $y = y^*$ may not be feasible due to input constraints or insufficient $m$. $\rightarrow$ use two-level approach.
Two-Level Optimization

Steady-State Optimization
(Linear Program or Quadratic Program)

Optimal Setting Values (setpoints)
\[ y_{\infty|k}, u^*_s(k) \]

Dynamic Optimization (\(\infty\)-horizon MPC)

Constraint \( y_{k+m+n-1|k} = y^*_{\infty|k} \) is guaranteed to be feasible.
Constraint \( \Delta u(k) + \cdots + \Delta u(k + m-1) = \Delta u^*_s \rightarrow y_{k+m+n-1|k} = y^*_{\infty|k} \).
Process Identification
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:
- Process Analysis
- Design and Tuning of Controller

MPC:
- Modeling and Identification
- Control Specification
Model Structure (1)

- **I/O Model**

\[ y(k) = G(q)u(k) + H(q)e(k) \]

Models auto- and cross-correlations of the residual (not physical cause-effect)

Assume w.l.g. that \( H(0) = 1 \)
SISO I/O Model Structure(1)

- **FIR** (Past inputs only)
  
  \[ y(k) = h_1 u(k-1) + \cdots + h_N u(k-m) + e(k) \]

- **ARX** (Past inputs and outputs: “Equation Error”)
  
  \[ y(k) = a_1 y(k-1) + \cdots + a_n y(k-n) + b_1 u(k-1) + \cdots + b_m u(k-m) + e(k) \]

- **ARMAX** (Moving average of the noise term)
  
  \[ y(k) = a_1 y(k-1) + \cdots + a_n y(k-n) + b_1 u(k-1) + \cdots + b_m u(k-m) + e(k) + c_1 e(k-1) + \cdots + c_n e(k-n) \]

- **Output Error** (OE), **Box-Jenkins** (BJ), etc.
Overview

Model Structure Selection → Model Structure → Parameter (Model) Estimation

- Prediction Error Method
- Subspace ID Method
- IV Method
- Statistical Method
  - MLE
  - Bayesian
- ETFE
  - Frequency Domain

Model For Validation

Data
Prediction Error Method

• Predominant method at current time
• Developed by Ljung and coworkers
• Flexible
  – Can be applied to any model structure
  – Can be used in recursive form
• Well developed theories and software tools
  – Book by Ljung, System ID Toolbox for MATLAB
• Computational complexity depends on the model structure
  – ARX, FIR → Linear least squares
  – ARMAX, OE, BJ → Nonlinear optimization
Prediction Error Method

- Put the model in the predictor form
  \[ y(k) = G(q, \theta)u(k) + H(q, \theta)e(k) \rightarrow \]
  \[ y_{k|k-1} = G(q, \theta)u(k) + \left( I - H^{-1}(q, \theta) \right)(y(k) - G(q, \theta)u(k)) \]
  \text{Contains at least 1 delay}
  \[ e(k) = y(k) - y_{k|k-1} = H^{-1}(q, \theta)(y(k) - G(q, \theta)u(k)) \]

- Choose the parameter values to minimize the sum of the prediction error for the given N data points.
  \[
  \min_{\theta} \left\{ \frac{1}{N} \sum_{k=1}^{N} \| e(k) \|_2^2 \right\} \quad \text{e(k) = } H^{-1}(q, \theta)(y(k) - G(q, \theta)u(k))
  \]

- ARX, FIR \rightarrow \text{Linear least squares,}
- ARMAX, OE, BJ \rightarrow \text{Nonlinear least squares}

- Not easy to use for identifying \textit{multivariable} models.
MIMO I/O Model Structure

• Inputs and outputs are vectors. Coefficients are matrices.
• For example, ARX model becomes

\[
y(k) = A_1 y(k - 1) + \cdots + A_n y(k - n) \\
+ B_1 u(k - 1) + \cdots + B_m u(k - m) + e(k)
\]

\[A_i\] is an \(n_y \times n_y\) matrix. \([B_i]\) is an \(n_y \times n_u\) matrix.

• Identification is very difficult.
  – Different sets of coefficient matrices giving exactly same \(G(q)\) and \(H(q)\) through pole/zero cancellations. \(\rightarrow\) Problems in parameter estimation \(\rightarrow\) Requires special parameterization to avoid problem.
State Space Model

- **Deterministic**
  \[
  x(k+1) = Ax(k) + Bu(k) \\
  y(k) = Cx(k) + e(k)
  \]

- **Combined Deterministic / Stochastic**
  \[
  x(k+1) = Ax(k) + Bu(k) + Ke(k) \\
  y(k) = Cx(k) + e(k)
  \]

- **Identifiability** can be an issue here too
  - State coordinate transformation does not change the I/O relationship.
Subspace Method

- More recent development
- Dates back to the classical realization theories but rediscovered and extended by several people
- Identifies a state-space model
- Some theories and software tools
- Computationally simple
  - Non-iterative, linear algebra
- Good for identifying *multivariable* models.
  - No special parameterization is needed.
- Not optimal in any sense
- May need a lot of data for good results
- May be combined with PEM
  - Use SS method to obtain an initial guess for PEM.
SIM Procedures

SIM algorithms have two categories and contain two steps:

Acknowledgment:
Prof. Joe Qin
Use of Nonlinear Model
\[
\dot{x} = f(x, u, d) \\
y = g(x)
\]

Discretization?

Orthogonal Collocation

\[
x(k+1) = F(x(k), u(k), d(k)) \\
y(k) = g(x(k))
\]

\[
y_{k+1|k} = g \circ F(x(k), u(k), d(k)) + e(k) \\
y_{k+2|k} = g \circ F(F(x(k), u(k), d(k)), u(k+1), d(k)) + e(k) \\
\vdots \\
y_{k+p|k} = g \circ F^p(x(k), u(k), \cdots u(k+p-1), d(k)) + e(k)
\]

The prediction equation is nonlinear w.r.t. \(u(k), \ldots, u(k+p-1)\)

Nonlinear Program (Not so nice!)
Difficult (2)

State Estimation

\[ \dot{x} = f(x, u, d) + w \]
\[ y = g(x) + v \]

Extended Kalman Filtering

\[ x(k + 1) = \int_{k}^{k+1} f(x, u, d) + K(k)(y_m(k) - g(x(k))) \]

• Computationally more demanding steps, e.g., calculation of K at each time step.
• Based on linearization at each time step – not optimal, may not be stable.
• Best practical solution at the current time
• Promising alternative: Moving Horizon Estimation (requires solving NLP).
• Difficult to obtain with an appropriate stochastic system model (no ID technique)
Practical Algorithm

Model integration with constant input $u=u(k-1)$ and $d=d(k)$

Linear prediction equation

Dynamic Matrix based on the linearized model at the current state and input values.

Linear or Quadratic Program
Additional Comments / Summary

• Some refinements to the “Practical Algorithm” are possible.
  – Use the previously calculated input trajectory (instead of the constant input) in the integration and linearization step.
  – Iterate between integration/linearization and control input calculation.

• “Full-blown” nonlinear MPC is still computationally prohibitive in most applications.
  – A lot of recent developments in SQP solver.

• Some promising directions
  – Tabulation
  – Simulation based approach (Approximate dynamic programming)
Remaining Challenges

- **Efficient** identification of *control-relevant* models
- 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 (NLP)
  - 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 **performance** of MPC.
Conclusion

• 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., plantwide) 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.
Some References to Start With.


Extra Slides
State-Space Model (1)

\[ z(k+1) = Az(k) + B^u u(k) + B^d d(k) \]
\[ y(k) = Cz(k) \]

Discretization

Linearization

Fundamental Model

\[ \dot{z} = \tilde{A}z + \tilde{B}^u u + \tilde{B}^d d \]
\[ y = \tilde{C}z \]

State-Space Realization

State-Space Identification

\[ y(z) = G^u(z) u(z) + G^d(z) d(z) \]
\[ y(s) = G(s) u(s) + G^d(s) d(s) \]

or

I/O model Identification

Test Data

\[ \{y(i), u(i), i = 1, \cdots, N\} \]
State-Space Model (2)

\[
\begin{align*}
    z(k+1) &= Az(k) + B^u u(k) + B^d d(k) \\
    y(k+1) &= Cz(k+1) \\
    z(k) &= Az(k-1) + B^u u(k-1) + B^d d(k-1) \\
    y(k) &= Cz(k)
\end{align*}
\]

\[
\Delta z(k+1) = A\Delta z(k) + B^u \Delta u(k) + B^d \Delta d(k)
\]

\[
\Delta y(k+1) = C\Delta z(k+1) \quad \rightarrow \\

y(k+1) = y(k) + C\left(A\Delta z(k) + B^u \Delta u(k) + B^d \Delta d(k)\right)
\]

\[
\begin{bmatrix}
    \Delta z(k+1) \\
    y(k+1)
\end{bmatrix} =
\begin{bmatrix}
    A & 0 \\
    CA & I
\end{bmatrix}
\begin{bmatrix}
    \Delta z(k) \\
    y(k)
\end{bmatrix} +
\begin{bmatrix}
    B^u \\
    CB^u
\end{bmatrix}
\Delta u(k) +
\begin{bmatrix}
    B^d \\
    CB^d
\end{bmatrix}
\Delta d(k)
\]

\[
y(k) =
\begin{bmatrix}
    0 & I
\end{bmatrix}
\begin{bmatrix}
    \Delta z(k) \\
    y(k)
\end{bmatrix}
\]

State Update

\[
x(k+1) = \Phi x(k) + \Gamma^u \Delta u(k) + \Gamma^d \Delta d(k)
\]

\[
y(k) = \Xi x(k)
\]
State-Space Model (3)

- **Prediction**
  - Model prediction of $y(k)$
  - Model prediction error $e(k) = y_m(k) - y(k)$

\[
\begin{bmatrix}
  y_{k+1|k} \\
  \vdots \\
  y_{k+p|k}
\end{bmatrix}
= \begin{bmatrix}
  \Xi \Phi \\
  \vdots \\
  \Xi \Phi^p
\end{bmatrix}
\begin{bmatrix}
  x(k) + \Omega^u \\
  \vdots \\
  \Delta u(k + p - 1)
\end{bmatrix}
\]

- **The “state” stored in “memory”**
- **Future input moves (to be decided)**
- **Dynamic Matrix (made of step response coefficients)**
- **Feedback Error Correction**

Feedforward term: new measurement

(Assume $\Delta d(k+1) = \ldots = \Delta d(k+p-1) = 0$)