MPC Introduction

• Overview
  • Basic Concept of MPC
  • History

• Optimization Formulation
  • Models
  • Analytical Solution to Unconstrained Problem

• Summary
  • Limitations & a Look Ahead

B. Wayne Bequette
Motivation: Complex Processes
Important Issues in Petroleum Refining

- **Multivariable, Large Scale**
  - Challenge to tune individual SISO controllers

- **Operation at Constraints**
  - Anti-reset windup and other strategies for PID

- **Economic Payout for Advanced Control**
  - Economic return justifies capital and on-going maintenance costs

- **Model Predictive Control**
  - Evolved independently in the US and France refining industry
How is MPC used?

Unit 1 - PID Structure

Plant-Wide Optimization

Unit 1 Local Optimization

High/Low Select Logic

PID

SUM

Unit 1 Distributed Control System (PID)

FC  PC  TC  LC

Unit 2 - MPC Structure

Model Predictive Control (MPC)

Unit 2 Local Optimization

Global Steady-State Optimization (every day)

Local Steady-State Optimization (every hour)

Dynamic Constraint Control (every minute)

Supervisory Dynamic Control (every minute)

Basic Dynamic Control (every second)

From Tom Badgwell, 2003 Spring AIChE Meeting, New Orleans
Model Predictive Control (MPC)

Find current and future manipulated inputs that best meet a desired future output trajectory. Implement first “control move”.

- Type of model for predictions?
- Information needed at step $k$ for predictions?
- Objective function and optimization technique?
- Correction for model error?
Model Predictive Control (MPC)

Find current and future manipulated inputs that best meet a desired future output trajectory. Implement first “control move”.

At next sample time:

Correct for model mismatch, then perform new optimization.

This is a major issue – “disturbances” vs. model uncertainty

B. Wayne Bequette
MPC History

- **Intuitive**
  - Basically arose in two different “camps”

- **Dynamic Matrix Control (DMC)**
  - 1960’s and 1970’s – Shell Oil - US
    - Related to techniques developed in France (IDCOM)
    - Large-scale MIMO
    - Formulation for constraints important

- **Generalized Predictive Control (GPC)**
  - Evolved from adaptive control
  - Focus on SISO, awkward for MIMO

---


Model Predictive Heuristic Control: Applications to Industrial Processes
J. Richalet,† A. Rault,‡ J. L. Testud‡ and J. Papon‡


B. Wayne Bequette
Objective Functions

Quadratic Objective Function, Prediction Horizon \((P) = 3\), Control Horizon \((M) = 2\)

\[
J = \left((r_{k+1} - \hat{y}_{k+1})^2 + (r_{k+2} - \hat{y}_{k+2})^2 + (r_{k+3} - \hat{y}_{k+3})^2\right) + \Delta u_k^2 + \Delta u_{k+1}^2
\]

3 steps into future

Weight

2 control moves

General Representation of a Quadratic Objective Function

\[
J = \sum_{i=1}^{P} (r_{k+i} - \hat{y}_{k+i})^2 + \Delta u_{k+i}^2
\]

With linear models, results in analytical solution (w/o constraints)

B. Wayne Bequette
Alternative Objective Functions

Penalize $u$ rather than $\Delta u$

$$J = \sum_{i=1}^{P} (r_{k+i} - \hat{y}_{k+i})^2 + w \sum_{i=0}^{M-1} u_{k+i}^2$$

Will usually result in “offset”

Sum of absolute values (results in LP)

$$J = \sum_{i=1}^{P} |r_{k+i} - \hat{y}_{k+i}| + w \sum_{i=0}^{M-1} |\Delta u_{k+i}|$$

Existing LP methods are efficient, but solutions hop from one constraint to another

B. Wayne Bequette
Models

- State Space
- ARX (auto-regressive, exogenous input)
- Step Response
- Impulse (Pulse) Response

- Nonlinear, Fundamental (First-Principles)
- ANN (Artificial Neural Networks)
- Hammerstein (static NL with linear dynamics)
- Volterra
- Multiple Model
Discrete Linear Models used in MPC

\[ x_{k+1} = \Phi x_k + \Gamma u_k \]
\[ y_k = C x_k \]

State Space

Some texts/papers have different sign conventions

\[ y_k = -a_1 y_{k-1} - a_2 y_{k-2} - \cdots - a_n y_{k-n} + b_0 u_k + b_1 u_{k-1} + b_2 u_{k-2} + \cdots + b_m u_{k-m} \]

Input-Output (ARX)

usually \( b_0 = 0 \)

\[ y_k = \sum_{i=1}^{\infty} s_i \Delta u_{k-i} \]

Step Response

\[ = s_1 \Delta u_{k-1} + \cdots + s_N \Delta u_{k-N} + s_{N+1} \Delta u_{k-N-1} + \cdots + s_{N+\infty} \Delta u_{k-\infty} \]

Impulse Response

\[ y_k = \sum_{i=1}^{\infty} h_i u_{k-i} \]

\[ = h_1 u_{k-1} + \cdots + h_N u_{k-N} + h_{N+1} u_{k-N-1} + \cdots + h_{N+\infty} u_{k-\infty} \]

B. Wayne Bequette
Example Step Response Model

\[ S = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & \cdots & s_N \end{bmatrix}^T \]
Example Impulse Response Model

Impulse and step response coefficients are related:

\[ h_i = s_i - s_{i-1} \]

\[ s_i = \sum_{j=1}^{i} h_j \]
Step & Impulse Models from State Space Models

\[ x_{k+1} = \Phi x_k + \Gamma u_k \]

\[ y_k = C x_k \]

\[ H_i = C \Phi^{i-1} \Gamma \]

\[ S_k = \sum_{i=1}^{k} C \Phi^{i-1} \Gamma = \sum_{i=1}^{k} H_i \]
MPC based on State Space Models

\[ x_{k+1} = \Phi x_k + \Gamma u_k \]
\[ y_k = C x_k \]

with known current state, easy to propagate estimates

\[ x_{k+1} = \Phi x_k + \Gamma u_k \]
\[ y_{k+1} = C x_{k+1} = C\Phi x_k + C\Gamma u_k \]

and, using control changes

\[ u_k = u_{k-1} + \Delta u_k \]
\[ y_{k+1} = C\Phi x_k + C\Gamma u_{k-1} + C\Gamma \Delta u_k \]
Use ^ notation for model states

\[ \hat{y}_{k+1|k} = C\Phi\hat{x}_{k|k} + C\Gamma u_{k-1} + C\Gamma \Delta u_k \]

Now, propagate the prediction for P steps into the future

B. Wayne Bequette
Output Predictions

\[
\begin{bmatrix}
\hat{y}_{k+1|k} \\
\hat{y}_{k+2|k} \\
\vdots \\
\hat{y}_{k+P|k}
\end{bmatrix}
= \begin{bmatrix}
C\Phi \\
C\Phi^2 \\
\vdots \\
C\Phi^P
\end{bmatrix}
\begin{bmatrix}
\hat{x}_{k|k}
\end{bmatrix}
+ \begin{bmatrix}
C\Gamma \\
C\Phi\Gamma + C\Gamma \\
\vdots \\
\sum_{i=1}^{P} C\Phi^{i-1}\Gamma
\end{bmatrix}
\Delta u_{k-1}
\]

"free" or "unforced response" (if no more control moves are made)

\[
\begin{bmatrix}
C\Gamma & 0 & \cdots & 0 \\
C\Phi\Gamma + C\Gamma & C\Gamma & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots \\
\sum_{i=1}^{P} C\Phi^{i-1}\Gamma & \sum_{i=1}^{P-1} C\Phi^{i-1}\Gamma & \cdots & \Delta u_{k} \\
& & & \Delta u_{k+1} \\
& & & \Delta u_{k+M-1}
\end{bmatrix}
= \Delta u_{f}
\]

"forced" response

B. Wayne Bequette
Output Predictions

\[
\begin{bmatrix}
\hat{y}_{k+1|k} \\
\hat{y}_{k+2|k} \\
\vdots \\
\hat{y}_{k+P|k}
\end{bmatrix}
= \begin{bmatrix}
C\Phi \\
C\Phi^2 \\
\vdots \\
C\Phi^P
\end{bmatrix}
\hat{x}_{k|k} + \begin{bmatrix}
S_1 \\
S_2 \\
\vdots \\
S_P
\end{bmatrix} u_{k-1}
\]

"free" or "unforced response" (if no more control moves are made)

\[
\begin{bmatrix}
S_1 & 0 & \cdots & 0 \\
S_2 & S_1 & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots \\
S_P & S_{P-1} & \cdots & S_{P-M+1}
\end{bmatrix}
\begin{bmatrix}
\Delta u_k \\
\Delta u_{k+1} \\
\vdots \\
\Delta u_{k+M-1}
\end{bmatrix}
\]

"forced" response

\[\Delta u_f\]

B. Wayne Bequette
Optimization Problem

\[
\min_{\Delta u_f} J = \sum_{i=1}^{P} (r_{k+i|k} - \hat{y}_{k+i|k})^T W^y (r_{k+i|k} - \hat{y}_{k+i|k}) + \sum_{i=0}^{M-1} \Delta u_{k+i}^T W^U \Delta u_{k+i}
\]

\[
\hat{E}^T W^Y \hat{E}
\]

\[
\Delta u_f^T W^U \Delta u_f
\]

Where

\[
W^y = \begin{bmatrix}
W^y & 0 & 0 \\
0 & \ddots & 0 \\
0 & 0 & W^y
\end{bmatrix}
\]

\[
W^U = \begin{bmatrix}
W^u & 0 & 0 \\
0 & \ddots & 0 \\
0 & 0 & W^u
\end{bmatrix}
\]

future setpoints

and

\[
\hat{E} = r - \hat{Y} = r - f - S_f \Delta u_f
\]

so

\[
\hat{E} = E - S_f \Delta u_f
\]

“unforced” (free response) error

B. Wayne Bequette
Optimization Problem

\[ \hat{E}^T W^Y \hat{E} = (E - S_f \Delta u_f)^T W^Y (E - S_f \Delta u_f) \]

\[ = E^T W^Y E - 2 \Delta u_f^T S_f^T W^Y E + \Delta u_f^T S_f^T W^Y S_f \Delta u_f \]

so

\[ \min_{\Delta u_f} J = \hat{E}^T W^Y \hat{E} + \Delta u_f^T W^U \Delta u_f \]

can be written

\[ \min_{\Delta u_f} J = \Delta u_f^T \left( S_f^T W^Y S_f + W^U \right) \Delta u_f - 2 \Delta u_f^T S_f^T W^Y E \]

and the unconstrained solution is found from

\[ \frac{\partial J}{\partial \Delta u_f} = 0 \]
Unconstrained Solution

Analytical Solution for Unconstrained System

\[
\Delta u_f = \left( S_f^T W^Y S_f + W^U \right)^{-1} S_f^T W^Y E
\]

In practice, do not actually invert a matrix. Solve as a set of simultaneous equations (or use \ in MATLAB)

\[
\Delta u_f = \left( S_f^T W^Y S_f + W^U \right) \backslash S_f^T W^Y E
\]

“unforced” error
Vector of Control Moves

\[ \Delta u_f = \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+M-1} \end{bmatrix} \]

current and future moves

Although a set of control moves is computed, only the first move \( \Delta u_k \) is implemented

The next output at \( k+1 \) is obtained, then a new optimization problem is solved
MPC Tuning Parameters

- Prediction Horizon, $P$
- Control Horizon, $M$
- Manipulated Input Weighting, $W_u$

Usually, $P \gg M$ for robustness (less aggressive action). Sometimes $M = 1$, with $P$ varied for desired performance.

Sometimes larger input weights for robustness.
Example Inverse Response Process: Van de Vusse

\[ \frac{dC_a}{dt} = -k_1 C_a - k_3 C_a^2 + (C_{ain} - C_a)u \]

\[ \frac{dC_b}{dt} = k_1 C_a - k_2 C_b - C_b u \]

where \( u = F/V \)

B. Wayne Bequette
Example: Inverse Response Process

- First four coefficients are negative.
- Sum of first eight coefficients is positive.
Closed-Loop:
Compare P=10, M=1 with P=25, M=1

Short prediction horizons & long control horizons lead to more aggressive action
Results

Control Horizon: $M = 1$, Weighting: $W = 0$

- $P = 8$:
  - Stable

- $P = 7$:
  - Unstable

B. Wayne Bequette
Stability of Inverse Response Systems with DMC

- For a control horizon, $M = 1$, closed-loop MPC will be stable for a prediction horizon where the sum of the impulse response coefficients has the same sign as the process gain.

\[ P_{\text{min}} \sum_{i=1}^{P_{\text{min}}} s_i > 0 \]


Paul R. Maurath, Duncan A. Mellichamp, and Dale E. Seborg*
Department of Chemical and Nuclear Engineering, University of California, Santa Barbara, Santa Barbara, California 93106


For the example, $P_{\text{min}} = 8$, so $P = 7$ = unstable
Summary

- Concise overview of MPC

- State space model, unconstrained solution

- Have not discussed
  - State estimation and “corrected outputs”
  - Disturbances
  - Constraints
  - Other model forms