Explicit MPC with output feedback using self-optimizing control

Henrik Manum, Sridharakumar Narasimhan, Sigurd Skogestad

Department of Chemical Engineering
Norwegian University of Science and Technology
N-7491 Trondheim

17th IFAC World Congress, Seoul, Korea, 2008

NTNU
Outline

1. Optimal operation paradigms
2. Self optimizing control
3. Explicit MPC
4. Link between the two
5. Output feedback
6. Extension to noisy measurements
7. Examples
Optimal operation paradigms
Self optimizing control
Explicit MPC
Link between the two
Output feedback
Extension to noisy measurements
Examples
Outline

1 Optimal operation paradigms
2 Self optimizing control
3 Explicit MPC
4 Link between the two
5 Output feedback
6 Extension to noisy measurements
7 Examples
Outline

1. Optimal operation paradigms
2. Self optimizing control
3. Explicit MPC
4. Link between the two
5. Output feedback
6. Extension to noisy measurements
7. Examples
1 Optimal operation paradigms
2 Self optimizing control
3 Explicit MPC
4 Link between the two
5 Output feedback
6 Extension to noisy measurements
7 Examples
1 Optimal operation paradigms
2 Self optimizing control
3 Explicit MPC
4 Link between the two
5 Output feedback
6 Extension to noisy measurements
7 Examples
Explicit MPC with output feedback using self-optimizing control
Outline

1. Optimal operation paradigms
2. Self optimizing control
3. Explicit MPC
4. Link between the two
5. Output feedback
6. Extension to noisy measurements
7. Examples
1. Optimal operation paradigms
2. Self optimizing control
3. Explicit MPC
4. Link between the two
5. Output feedback
6. Extension to noisy measurements
7. Examples
Implementation of optimal operation using off-line computations

Paradigm 1
On-line optimizing control where measurements are primarily used to update the model. With arrival of new measurements, the optimization problem is resolved for the inputs.

Paradigm 2
Pre-computed solutions based on off-line optimization. Typically, the measurements are used to (indirectly) update the inputs using feedback control schemes. **Focus of this work.**
Implementation of optimal operation using off-line computations

Paradigm 1
On-line optimizing control where measurements are primarily used to update the model. With arrival of new measurements, the optimization problem is resolved for the inputs.

Example: Classical (implicit) MPC.

Paradigm 2
Pre-computed solutions based on off-line optimization. Typically, the measurements are used to (indirectly) update the inputs using feedback control schemes. Focus of this work.

Examples: Explicit MPC and self-optimizing control.
What variables should we control?

Controller

Plant

Measurement combination

\[ c_m = c + n \]

\[ d \rightarrow c_s \rightarrow u \rightarrow y \]

Self-optimizing control

Choice of H such that acceptable operation is achieved with constant setpoints (\(c_s\) constant).
What variables should we control?

Controller

Plant

Measurement combination $H$

$c_s$ $\rightarrow$ Controller $\rightarrow$ $c_m = c + n$ $\rightarrow$ $n$

$d$ $\rightarrow$ Plant $\rightarrow$ $u$ $\rightarrow$ $y$ $\rightarrow$ $c$

Self-optimizing control

*Choice of $H$ such that acceptable operation is achieved with constant setpoints ($c_s$ constant).*

Optimal $c_s$ is **invariant** with respect to disturbances $d$
What variables should we control?

**Self-optimizing control**

*Choice of H such that acceptable operation is achieved with constant setpoints (c_s constant).*

- Optimal c_s is *invariant* with respect to disturbances d
- Insensitive to measurement errors n
What variables should we control?

\[ c_1 = c_1^{sp} \]

![Graph showing the relationship between loss and disturbance with a minimum at \( d_0 \).]
What variables should we control?

\[ c_1 = c_1^{sp} \]

Loss

Disturbance

\[ d_0 \rightarrow d \]
What variables should we control?

\[ c_1 = c_1^{sp} \]

Loss due to constant setpoint policy

Loss

Disturbance

\[ d_0 \rightarrow d \]
What variables should we control?

\[
\begin{align*}
  c_1 &= c_1^{sp} \\
  c_2 &= c_2^{sp}
\end{align*}
\]
What variables should we control?

\[ c_1 = c_1^{sp} \]

\[ c_2 = c_2^{sp} \]

Acceptable loss

\[ d_0 \]

\[ d \]
What variables should we control?

\[ c_1 = c_1^{sp} \]

\[ c_2 = c_2^{sp} \]

\[ c_3 = Hy = c_3^{sp} \]

Loss

Disturbance

Henrik Manum, Sridharakumar Narasimhan, Sigurd Skogestad
Explicit MPC with output feedback using self-optimizing control
The nullspace method is restated for QP’s

Theorem (Nullspace method for QP)

Consider the quadratic problem

$$\min_u J = \begin{bmatrix} u & d \end{bmatrix} \begin{bmatrix} J_{uu} & J_{ud} \\ J_{ud}^T & J_{dd} \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix}$$ (1)

If there exists $n_y \geq n_u + n_d$ independent measurements, then the optimal solution to (1) has the property that there exists variable combinations $c = Hy$ that are invariant to the disturbances $d$.

$H$ may be found from $HF = 0$, where $F = \frac{\partial y^{opt}}{\partial d^T}$.
The “classical” MPC problem can, by substitution, be written as a **quadratic** problem:

\[
\min_U J(U, x(t)) = \begin{bmatrix} U^T & x(t)^T \end{bmatrix} \begin{bmatrix} H & F \\ H & Y \end{bmatrix} \begin{bmatrix} U \\ x(t) \end{bmatrix}
\]

s.t. \( GU \leq W + Ex(t) \)

The initial state \(x(t)\) is considered to be a parameter and a parametric program is solved.

The solution of the parametric program gives regions in the state space.

Given an algorithm for deciding the current region \((i)\), one implements a continuous piece-wise affine control law

\[
u = F^i x + g^i.
\]
Let

\[ d = x_0 \quad \text{and} \quad y = \begin{bmatrix} u \\ x \end{bmatrix} \]

The optimal combination

\[ c = Hy \]

can be written as the feedback law

\[ c = u - (Kx + g) \]

and \( H \) (or \( K \)) can be obtained from nullspace method.
The invariants can be used to track region changes

- By monitoring neighboring regions we switch regions when $c_i - c_j$ changes sign
When to switch region?

- The invariants can be used to track region changes.
- By monitoring neighboring regions, we switch regions when $c_i - c_j$ changes sign.

\[ c_1 = 0 \]

Region 1

Region 2

State $x$
When to switch region?

- The invariants can be used to track region changes.
- By monitoring neighboring regions, we switch regions when $c_i - c_j$ changes sign.

The diagram shows two regions:

- Region 1 with $c_1 = 0$.
- Region 2 with $c_2 = 0$.

State $x$ is the vertical axis, and the horizontal axis represents the state space.
The invariants can be used to track region changes.

By monitoring neighboring regions we switch regions when \( c_i - c_j \) changes sign.

\[
\begin{align*}
\text{Region 1:} & \quad c_1 = 0 \\
\text{Region 2:} & \quad c_2 = 0
\end{align*}
\]
When to switch region?

- The invariants can be used to track region changes.
- By monitoring neighboring regions we switch regions when $c_i - c_j$ changes sign.

Diagram:
- Region 1: $c_1 = 0$
- Region 2: $c_2 = 0$
- The line $c_1 - c_2$ represents the boundary between the two regions.
When to switch region?

- The invariants can be used to track region changes.
- By monitoring neighboring regions we switch regions when $c_i - c_j$ changes sign.

![Diagram showing the invariants and regions](image)

- $c_1 = 0$ (Region 1)
- $c_2 = 0$ (Region 2)
- $c_1 - c_2$ axis
Example 1: Output feedback

**Process**

- \( y(t) = \frac{2}{s^2 + 3s + 2} \)
- Input constraint: \(|u(t)| \leq 2\)
- Sample the system and get two-state discrete model
- Quadratic objective function
**Example 1: Output feedback**

### Process
- \( y(t) = \frac{2}{s^2 + 3s + 2} \)
- Input constraint: \(|u(t)| \leq 2\)
- Sample the system and get two-state discrete model
- Quadratic objective function

### Control
- **Alternative 1** \( u_k = -Kx_k \) + observer
- **Alternative 2** \( u_k = -K_y [y_k \ y_{k-1}]^T \)

Henrik Manum, Sridharakumar Narasimhan, Sigurd Skogestad: Explicit MPC with output feedback using self-optimizing control
Example 1: Output feedback

Process

- \( y(t) = \frac{2}{s^2 + 3s + 2} \)
- Input constraint: \( |u(t)| \leq 2 \)
- Sample the system and get two-state discrete model
- Quadratic objective function

Control

Alternative 1  \( u_k = -Kx_k + \text{observer} \)

Alternative 2  \( u_k = -K_y [y_k \ y_{k-1}]^T \)
Example 1: Output feedback

**Process**
- \( y(t) = \frac{2}{s^2+3s+2} \)
- Input constraint: \(|u(t)| \leq 2\)
- Sample the system and get two-state discrete model
- Quadratic objective function

**Control**

**Alternative 1**
\[ u_k = -Kx_k + \text{observer} \]

**Alternative 2**
\[ u_k = -K_y[y_k \ y_{k-1}]^T \]

**Alternative 2**
- \( y = (y_k, y_{k+1}, u_k, u_{k+1}) \)
- Write
  \[ y = G_y \begin{bmatrix} u_k \\ u_{k+1} \end{bmatrix} + G_{d}^y x_k \]
- Sensitivity
  \[ F = -(G_y J_{uu}^{-1} J_{ud} - G_{d}^y) \]
- Find \( H \) such that \( HF = 0 \)
Example 1: Output feedback

Process
- $y(t) = \frac{2}{s^2+3s+2}$
- Input constraint: $|u(t)| \leq 2$
- Sample the system and get two-state discrete model
- Quadratic objective function

Control
**Alternative 1** $u_k = -Kx_k + \text{observer}$
**Alternative 2** $u_k = -K_y[y_k \ y_{k-1}]^T$

Alternative 2
- $y = (y_k, y_{k+1}, u_k, u_{k+1})$
- Write
  $y = G^y \begin{bmatrix} u_k \\ u_{k+1} \end{bmatrix} + G_d^y x_k$
- Sensitivity
  $F = -(G^y J_{uu}^{-1} J_{ud} - G_d^y)$
- Find $H$ such that $HF = 0$

$u_k = -(16.7y_k + 13.7y_{k-1})$
Phase plane

Phase plane

States

Inputs

State space partition and simulation from $x_0 = (1, 1)$
Example 1: Output feedback

Phase plane

\[ u = -2 \]

\[ u = 2 \]

State space partition and simulation from \( x_0 = (1, 1) \)

States

Inputs

\[ u_k - Kx_k \]

\[ u_k - K_y [y_k \ y_{k-1}]^T \]
Noisy measurements

Cost $J$

$J_{\text{opt}}$

$C_{\text{opt}}$

Controlled variable $c$
Noisy measurements

\[ \text{Cost } J \]

\[ J_{\text{opt}} \]

\[ c_{\text{opt}} \]

\[ n \]

\[ \text{Controlled variable } c \]

- Implementation error: \( c = c_{\text{opt}} + n \).
Noisy measurements

Implementation error: \( c = c_{\text{opt}} + n. \)
Noisy measurements

\[ J_{\text{opt}} \]

\[ \text{Cost } J \]

\[ c_{\text{opt}} \]

\[ n \]

\[ \text{Loss} \]

\[ c = c_{\text{opt}} + n. \]

Want to find invariants \( c \) to both disturbances and noise.
Explicit formula for optimal $H$ for $n \neq 0$

Loss = $J(u, d) - J_{opt}(d)$. Keep $c = Hy$ constant, where $y = G^y u + G^y_d d + n^y$

**Theorem (Explicit formula for optimal $H$ (Alstad et al, 2008))**

Define $\tilde{F} = [FW_d \ W_{ny}]$. Then

$$H^T_{opt} = (\tilde{F} \tilde{F}^T)^{-1} G^y \left( (G^y)^T(\tilde{F} \tilde{F}^T)^{-1} G^y \right)^{-1} J_{uu}^{1/2}$$

Here $F$ is the optimal sensitivity matrix $F = \frac{\partial y_{opt}}{\partial d}$
Example 2: Output feedback with noise

Process

\[
x_{k+1} = \begin{bmatrix} 0.73 & -0.09 \\ 0.17 & 0.99 \end{bmatrix} x_k + \begin{bmatrix} 0.060 \\ 0.006 \end{bmatrix} u_k + w_k
\]

\[
y_k = \begin{bmatrix} 0 & 1.41 \end{bmatrix} x_k + v_k
\]
Example 2: Output feedback with noise

Process

\[
\begin{align*}
  x_{k+1} &= \begin{bmatrix} 0.73 & -0.09 \\ 0.17 & 0.99 \end{bmatrix} x_k + \begin{bmatrix} 0.060 \\ 0.006 \end{bmatrix} u_k + w_k \\
y_k &= \begin{bmatrix} 0 & 1.41 \end{bmatrix} x_k + v_k
\end{align*}
\]

Control

Alternative 1 \( u_k = -Kx_k + \text{Kalman filter} \)
Alternative 2 \( u_k = -K_y(y_k, y_{k-1}, y_{k-N}) \) from “noisy nullspace method”
Example 2: Output feedback with noise

**Process**

\[
\begin{align*}
    x_{k+1} &= \begin{bmatrix} 0.73 & -0.09 \\ 0.17 & 0.99 \end{bmatrix} x_k + \begin{bmatrix} 0.060 \\ 0.006 \end{bmatrix} u_k + w_k \\
y_k &= \begin{bmatrix} 0 & 1.41 \end{bmatrix} x_k + v_k
\end{align*}
\]

**Control**

- **Alternative 1** \( u_k = -K x_k + \text{Kalman filter} \)
- **Alternative 2** \( u_k = -K_y(y_k, y_{k-1}, y_{k-N}) \) from “noisy nullspace method”
Example 2: Output feedback with noise

Process

\[ x_{k+1} = \begin{bmatrix} 0.73 & -0.09 \\ 0.17 & 0.99 \end{bmatrix} x_k + \begin{bmatrix} 0.060 \\ 0.006 \end{bmatrix} u_k + w_k \]

\[ y_k = \begin{bmatrix} 0 & 1.41 \end{bmatrix} x_k + v_k \]

Control

Alternative 1

\[ u_k = -Kx_k + \text{Kalman filter} \]

Alternative 2

\[ u_k = -K_y(y_k, y_{k-1}, y_{k-N}) \text{ from “noisy nullspace method”} \]

Alternative 2

\[ R, Q, W_{ny}, W_d \implies K_y \]
Example 2: Output feedback with noise

Simulated costs \( J = \frac{1}{N} \sum_{i=1}^{N} x_i^T C^T Q^y C x_i + u_i^T R u_i \):

<table>
<thead>
<tr>
<th>Control equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_k = -[6.08 \ 6.07] x_k ) (perfect measurement)</td>
</tr>
<tr>
<td>( u_k = -[6.08 \ 6.07] \hat{x}_k ) (+ Kalman filter)*</td>
</tr>
<tr>
<td>( u_k = -(3.25 y_k) )</td>
</tr>
<tr>
<td>( u_k = -(1.54 y_k + 0.5 y_{k-1}) )</td>
</tr>
<tr>
<td>( u_k = -(0.78 y_k + 0.44 y_{k-1} - 0.03 y_{k-2}) )</td>
</tr>
<tr>
<td>( u_k = -(0.39 y_k + 0.28 y_{k-1} + 0.12 y_{k-2} - 0.09 y_{k-3}) )</td>
</tr>
</tbody>
</table>

*: Optimal for white noise signals
Simulated costs ($J = \frac{1}{N} \sum_{i=1}^{N} x_i^T C^T Q^y C x_i + u_i^T R u_i$):

<table>
<thead>
<tr>
<th>Control equation</th>
<th>$J_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_k = -[6.08 \ 6.07]x_k$ (perfect measurement)</td>
<td>2.86</td>
</tr>
<tr>
<td>$u_k = -[6.08 \ 6.07]\hat{x}_k$ (+ Kalman filter)*</td>
<td>3.40</td>
</tr>
<tr>
<td>$u_k = -(3.25y_k)$</td>
<td>5.27</td>
</tr>
<tr>
<td>$u_k = -(1.54y_k + 0.5y_{k-1})$</td>
<td>3.88</td>
</tr>
<tr>
<td>$u_k = -(0.78y_k + 0.44y_{k-1} - 0.03y_{k-2})$</td>
<td>3.88</td>
</tr>
<tr>
<td>$u_k = -(0.39y_k + 0.28y_{k-1} + 0.12y_{k-2} - 0.09y_{k-3})$</td>
<td>4.11</td>
</tr>
</tbody>
</table>

$J_1$  Process noise at all time instants

*: Optimal for white noise signals
Example 2: Output feedback with noise

Simulated costs \( J = \frac{1}{N} \sum_{i=1}^{N} x_i^T C^T Q^y C x_i + u_i^T R u_i \):

<table>
<thead>
<tr>
<th>Control equation</th>
<th>( J_1 )</th>
<th>( J_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_k = -[6.08 \ 6.07] x_k ) (perfect measurement)</td>
<td>2.86</td>
<td>0.284</td>
</tr>
<tr>
<td>( u_k = -[6.08 \ 6.07] \hat{x}_k ) (+ Kalman filter)*</td>
<td>3.40</td>
<td>0.400</td>
</tr>
<tr>
<td>( u_k = -(3.25 y_k) )</td>
<td>5.27</td>
<td>0.569</td>
</tr>
<tr>
<td>( u_k = -(1.54 y_k + 0.5 y_{k-1}) )</td>
<td>3.88</td>
<td>0.401</td>
</tr>
<tr>
<td>( u_k = -(0.78 y_k + 0.44 y_{k-1} - 0.03 y_{k-2}) )</td>
<td>3.88</td>
<td>0.394</td>
</tr>
<tr>
<td>( u_k = -(0.39 y_k + 0.28 y_{k-1} + 0.12 y_{k-2} - 0.09 y_{k-3}) )</td>
<td>4.11</td>
<td>0.416</td>
</tr>
</tbody>
</table>

\( J_1 \) Process noise at all time instants

\( J_2 \) Process noise at every 10th instant

*: Optimal for white noise signals
Current and future work

- Include measurement error in explicit MPC (with region switching)
- Explicit expressions for fixed low-order controllers, e.g. MIMO-PID
**Conclusion**

- **MPC**: Quadratic optimization problem
- **Self-optimizing control**: Exact results for QP’s, both noise-free and with noisy measurements
- **Link**: \( c = u - Kx \)
- **New results**:
  - \( c \)'s for region switching
  - Output feedback \( c = u - K^y y \)
  - Optimal invariants for *noisy* measurements