Allocating automation functions within an IIoT architecture

Advances in Process Automation and Control 2017
12–14 June 2017, Birmingham, UK
Edge computing architecture

Traditional Purdue Reference Architecture

- Level 5: Enterprise workstations & servers
- Level 4: Process History, Site workstations & servers
- Level 2: Supervisory control, Human machine interface
- Level 1: Basic Control, PLC Control
- Level 0: Safety system

IloT Architecture

Cloud
- Cloud Applications: Analytics, Visualization, Historian, Dashboards

Edge
- Edge Gateway
- DCS Server → SCADA → Local Historian
- Edge Device
- DCS, PLC, RTU
- Process Data
Edge computing architecture

- Security
- Scalability
  - Flexible ‘plug and play’ design
- Open Standards
  - Interoperability
  - Portability
- Programmability
- Reliability, Availability, and Serviceability

**Programming Standards**
- IEC61131-3
- PLCopen

**Communication Standards**
- OPC-UA
- MQTT / AMQP

Client Server

Publish Subscribe
Using IIoT to control large scale systems

- Large scale systems consist of complex networks of **interconnected** and **interacting** subsystems

- Examples:
  - Integrated process plants
  - Utility networks (Electrical, Gas, Water)
  - Buildings

- There is often significant benefit to be gained through optimisation of the **complete** system rather than treating it as series of distributed subsystems

- IIoT architectures provide an opportunity to better control larger and more geographically dispersed systems
Centralised control

MPC
\[ \min V (u_1, u_2, ...) \]

- Single centralized MPC
- Global objective function
- Stable for any interacting systems
- Complex & Inflexible
- Pareto optimal
- Benchmark solution
Decentralised control

- Each subsystem is controlled by a separate independent MPC
- Each MPC solves a local objective function
- No inter-controller communication
- MPC’s should be large enough to account for strong system interactions
- Not stable for strongly interacting subsystems
- Not globally optimal
Decentralised control with centralised coordinator

- Each subsystem is controlled by a separate independent MPC
- Each MPC solves a **local** objective function
- Each MPC shares its move plan with other MPC’s
- A centralised coordinator may be used to coordinate steady state objectives
- MPC’s should be large enough to account for strong system interactions
- Not stable for strongly interacting subsystems
- Nash equilibrium
Cooperative control

- Each subsystem is controlled by a separate independent MPC
- All controllers share a common objective function
- Each MPC shares its move plan with other controllers
- Stable for strongly interacting subsystems
- Stability is independent of MPC size and design
- High flexibility
- MPC’s can have different execution frequencies
- Pareto optimal (with sufficient iterations)
Each subsystem is controlled by a separate independent MPC

All controllers share a common objective function

Each MPC shares its move plan with other controllers

Stable for strongly interacting subsystems

Stability is independent of MPC size and design

High flexibility

MPC’s can have different execution frequencies

Pareto optimal (with sufficient iterations)
Consider a simple plant with two subsystems:

\[ V(x_1(0), x_2(0), u_1, u_2) = \sum_{i=1}^{N-1} \left[ \frac{1}{2} [x_i, u_i]_i^T H_i [x_i, u_i]_i + f_i^T [x_i, u_i]_i \right] + [x_i, u_i]_N^T H_N [x_i, u_i]_N \]

\[ x_{i+1} = Ax_i + B_1 u_i + B_2 u_2 \quad \text{and} \quad P_i [x_i, u_i] \leq b_i \]

This can be solved by iterating over the following steps:

Step 1. Set \( u_2 = u_2^p \): \( \min_{u_1} V(x_1(0), x_2(0), u_1, u_2) \rightarrow (u_1^*, u_2^p) \)

Step 2. Set \( u_1 = u_1^p \): \( \min_{u_2} V(x_1(0), x_2(0), u_1, u_2) \rightarrow (u_2^*, u_1^p) \)

Step 3: \( u_i^{p+1} = \alpha (u_i^p) + (1 - \alpha) (u_i^*) \)
Cooperative Model Predictive Control

MPC1
Initilise move plan \( u_1^0 \)

MPC2
Initilise move plan \( u_2^0 \)
Cooperative Model Predictive Control

**MPC1**

Set $u_2 = u_2^0$

$\min_{u_1} V(x_1(0), x_2(0), u_1, u_2) \rightarrow (u_1^*, u_2^0)$

**MPC2**

$u_2^0$
Cooperative Model Predictive Control

1. Set $u_1 = u_1^0$
2. $\min_{u_2} V(x_1(0), x_2(0), u_1, u_2) \rightarrow (u_2^*, u_1^0)$
Cooperative Model Predictive Control

\[
MPC1 \\
u_1^1 = \alpha(u_1^0) + (1 - \alpha)(u_1^*)
\]

\[
MPC2 \\
u_2^1 = \alpha(u_2^0) + (1 - \alpha)(u_2^*)
\]
**Cooperative Model Predictive Control**

**MPC1**
Set \( u_2 = u_2^1 \)

\[
\min_{u_1} V(x_1(0), x_2(0), u_1, u_2) \rightarrow (u_1^*, u_2^1)
\]

**MPC2**

\[
(u_1^*, u_2^p)
\]

\[
(u_1^p, u_2^p)
\]
Cooperative Model Predictive Control

**MPC1**

\[ u_1^1 \]

**MPC2**

Set \( u_1 = u_1^1 \)

\[
\min_{u_2} V(x_1(0), x_2(0), u_1, u_2) \rightarrow (u_2^*, u_1^1)
\]
Cooperative Model Predictive Control

**MPC1**

\[ u_1^2 = \alpha(u_1^1) + (1 - \alpha)(u_1^*) \]

**MPC2**

\[ u_2^2 = \alpha(u_2^1) + (1 - \alpha)(u_2^*) \]
Cooperative Model Predictive Control

- Iterating to convergence gives optimal plant-wide control
- Early termination of the optimization gives stable, suboptimal plant-wide control
- Stability can be proved using sub-optimal MPC theory
LV control of distillation column
# LV control of distillation column

<table>
<thead>
<tr>
<th></th>
<th>Cost</th>
<th>Performance loss (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized MPC</td>
<td>75.8</td>
<td>0</td>
</tr>
<tr>
<td>Cooperative MPC (10 iterates)</td>
<td>76.1</td>
<td>0.388</td>
</tr>
<tr>
<td>Cooperative MPC (1 iterate)</td>
<td>87.5</td>
<td>15.4</td>
</tr>
<tr>
<td>Non-cooperative MPC</td>
<td>382</td>
<td>404</td>
</tr>
<tr>
<td>Decentralized MPC</td>
<td>364</td>
<td>380</td>
</tr>
</tbody>
</table>

Rawlings and Stewart 2010
Ethylene Plant Ethane Feed

Ethylene Plant C2 Feed = f(Gas Plant Demethaniser Duty)

\[ \min_{\text{demeth}} V(\text{Gas Plant, NGL Plant, Ethylene Plant}) \]

100 km
Advantages of Distributed Cooperative Control

• Eliminates the requirement for a central optimiser or coordinator
• Can incorporate mixed dynamics
• Expandable and flexible -> plug and play design
• Stability is independent of the strength of process interactions or of individual controller design
• Systems can be geographically dispersed
• Suited to distributed hardware with limited processing capacity (such as embedded systems and edge devices)
• Ideally suited to an IIoT architecture
References


Get in touch

+44-(0)1244-568-941

patrick.thorpe@spirocontrol.com

Linkedin/SprioControl

@SpiroControl