Implemented Software Architecture

TU Eindhoven: Mario Balenovic
TU Delft: Dennis van Hessem
IPCOS: Peter Geerlings
Outline

1. Basic Idea
   • Software Implementation
   • Experimental implementation
   • Data server setup
5. Conclusions
Basic Idea

Various INCOOP modules:

- Estimation
- Time Scale Separation
- Dynamic Real Time Optimisation
- Model Predictive Control
- Plant Simulator

Plant (including base control)

- state and disturbance estimates (slow)
- state and disturbance estimates (fast)
- optimal reference trajectories
- control setpoints
- measurements
- market and environmental conditions
Basic Idea

Have a good software development environment

Modular setup

- Various modules should easily be added/changed or removed

Use mathematical tool such as Matlab

- Freedom in algorithmic development

Can use/integrate software from other suppliers

- SLICOT, optimization tools, C/C++ etc.

Use professional process modeling tool including integration routines such as gPROMS.
Software Implementation

INCOOP Software Architecture
Signal definitions

• For each new model define inputs, outputs and disturbances
• Automatically generate model information structure containing:
  – state/input/output names and indices
  – still retain the physical meaning of signals just as in model
• Use these physical names as basis for signals
  – root_name_signal.class.subclass
  – classes: predicted, history, estimated etc.
  – ex: column.tray(10).temp.history.measured.value
• For any signal use time-stamp
  – Easy to inject new reference signals (dynamic optimization)
  – Full integration of optimization and control
Basic Idea

Have a good experimental setup

Replace process simulator with real plant data
  – Have low implementation cost

Work with professional process operation tools
  – IPCOS Integrated Solution Platform (IISP)
  – Central data server
  – Communication protocols
INCOOP Software Architecture
Conclusions

Advantages

- Distributed modular structure.
- Easy to adapt to new processes.
- Equivalent simulation and process environment.
- Easy to interchange and add modules.
- Suited for large processes.
- Networks of various OPC servers are easily applied.

Typical experimental setup

- Three-server-network:
  • process server (connection to DCS)
  • calculation module server (standard INCOOP environment)
  • user interface server (visualisation)
- Failure in one module does not influence other modules.