A Framework for the Simulation and Validation of Distributed Control Architectures for Technical Systems of Systems

S. Nazari*, S. Wenzel*, L.S. Maxeiner*, C. Sonntag*,**, S. Engell*

*Process Dynamics and Operations Group, Dept. of Biochemical and Chemical Engineering, TU Dortmund, Germany

** euTeXoo GmbH, Germany
Outline

- Cyber-Physical Systems of Systems (CPSoS)
- Hierarchical / Distributed Control of CPSoS
- Objective of this Work
- The Simulation and Validation Framework (SVF)
- Case Studies
  - An Integrated Chemical Production Site
  - A Network of three Multi-Product Semi-Batch Reactors
- Summary and Outlook
Large, complex, often spatially distributed Cyber-physical Systems (CPS) that exhibit the features of Systems of Systems (SoS)

Cyber-Physical Systems of Systems (CPS)

- **Tight interaction** of many distributed, real-time computing systems and physical systems
  - Examples:
    - Airplanes
    - Cars
    - Ships
    - Buildings with advanced HVAC controls
    - Manufacturing plants
    - Power plants
    - ...

- **Many interacting components**
  - Examples:
    - Large industrial sites with many production units
    - Large networks of systems (electric grid, traffic systems, water distribution)

- **Physical connections**
  - Examples:
    - Material/energy streams
    - Shared resources (e.g., roads, airspace, rails, steam)
    - Communication networks

- **Examples of Cyber-physical Systems of Systems**
  - Integrated large production complexes
    - Major source of employment and income in Europe
    - Major consumer of energy and raw materials
    - Many interconnected production plants that are operated mostly autonomously with distributed management structures

  - Transportation networks (road, rail, air, maritime, ...)
    - Vital to the mobility of EU citizens and the movement of goods
    - Large integrated infrastructures with complex interactions, also across national borders
    - Involve multiple organizational and political structures

  - Many more examples, e.g., smart (energy, water, gas, ...) networks, supply chains, or manufacturing

Systems of Systems (SoS)

- **Dynamic reconfiguration**
  - Components may...
    - be switched on and off (as in living cells)
    - enter or leave (as in air traffic control)

- **Continuous evolution**
  - Continuous addition, removal, and modification of hardware and software over the complete life cycle (often many years)

- **Emerging behavior**
  - The overall SoS shows behaviours that do not result from simple interactions of subsystems
  - Usually not desired in technical systems, may lead to reduced performance or shut-downs

- **Partial autonomy**
  - Local actors with local authority and priorities
  - Autonomous systems ...
  - Examples:
    - cannot be fully controlled on the SoS level
    - need incentives towards global SoS goals

- **Examples**:
  - Local energy generation companies
  - Process units of a large chemical site

www.cpsos.eu

IFAC WC, Toulouse, France
July 9th-14th, 2017
Hierarchical / Distributed Control of CPSoS (1)

- Partial autonomy and distributed decisions in an integrated chemical production site
Centralized control of CPSoS is preferable, but not always feasible or desired

- Complexity of the management problem
- Privacy concerns may prohibit the sharing of operational details of the subsystems

Distributed management, coordination, and optimization approaches
Objective of this Work

- How can state-of-the-art distributed control algorithms be systematically validated on simulation models while...
  - ...re-using (pre-existing) simulation models
    - Heterogeneous, possibly from different simulation environments
  - ... not having to implement the communication and automation architectures manually? (which is time-consuming and error-prone)
  - ... being able to connect management methods to different CPSoS models effortlessly?
    - Avoiding proprietary implementations
The Simulation and Validation Framework (SVF) (1)

- A plug-and-play based approach
The Simulation and Validation Framework (SVF) (2)

- **Modelica**-based framework for the systematic interconnection of:
  - Validation models
  - Local and high-level optimization algorithms
    - Event-driven communication

- Standard interfaces for:
  - The interconnection of physical models
  - The interconnection of physical models and controllers
  - The interconnection of controllers
The Model Management Engine (MME)

- An intermediary component responsible for the coordination of the model components during simulation, e.g.:
  - Data communication between the components
  - Propagation of discrete events
SVF- Supported Languages and Features

- Support for:
  - White-box Modelica models and black-box models via co-simulation (FMI)
  - Modelica-based controllers, white-box and black-box external controllers
    - External support is done via the SVF External Function Interface
    - Python, Matlab and C are supported

- Generation of the communication structure
  - Via a generic XML-based configuration file
  - Reduces risk of errors
Case Studies

- Integrated chemical production complex
  - 9 processing plants whose models are derived from planning data
  - **Goal**: Balancing of the two steam networks

- Network of three semi-batch reactors that are operated autonomously
  - The reactors are coupled via discrete and continuous resources
  - Exothermic reaction \( A + B \rightarrow C \)
  - Goal: produce as much product C as possible for a given final time of 30 hours using a moving horizon optimization

Case Studies – Problem Formulation (1)

- For the chemical complex, the Alternating Direction Method of Multipliers (ADMM) is used [1]

  centralized problem for \( n \) subsystems:

  \[
  \min_{u_i \in U_i, \forall i} \sum_{i=1}^{n} j_i(u_i)
  \]

  s.t. \( \sum_{i=1}^{n} r_i(u_i) = 0 \)

  Balance of the shared resource networks

\[
\mathcal{L}_{\rho,i} = j_i(u_i) + (\lambda^k)^T \sum_{i=1}^{n} r_i(u_i) + \frac{\rho}{2} \sum_{i=1}^{n} \|r_i(u_i) - z_i^k\|^2_2
\]

Case Studies – Problem Formulation (2)

For the chemical complex, the Alternating Direction Method of Multipliers (ADMM) is used [1] centralized problem for \( n \) subsystems:

\[
\min_{u_i \in U_i, \forall i} \sum_{i=1}^{n} J_i(u_i) \\
\text{s.t. } r_i(u_i) = z_i \\
\sum_{i=1}^{n} z_i = 0.
\]

- Solved by the subsystems
- Solved by a high-level controller, i.e. the coordinator

\[
L_{\rho,i} = J_i(u_i) + (\lambda^k)^T \sum_{i=1}^{n} r_i(u_i) + \frac{\rho}{2} \sum_{i=1}^{n} \|r_i(u_i) - z_i^k\|_2^2
\]

- Relaxing of the coupling constraint
- Local systems \( \rightarrow u_i \)
- The coordinator manipulates the local decisions by setting the internal shared resource prices \( \lambda \) and values of \( z_i \)

Case Studies – Problem Formulation (3)

- For the reactor network, price-based coordination is used
  - The local problem for subsystem \( i \)
    \[
    \min_{u_i \in U_i} J_i(u_i) + \lambda^T r_i(u_i)
    \]
  \( \rightarrow \) Solved for \( u_i \)
  
  - The coordinator manipulates the local decisions by setting the internal shared resource prices \( \lambda \)
    \[
    d(\lambda) = \sum_{i=1}^{n} r_i(u_i(\lambda)) = 0
    \]
Chemical Production Complex– SVF Implementation

- **Matlab**-based implementations of local optimization algorithms and the coordinator
  - C-based DLL files (black box) are created using the **Matlab** compiler
- Iterative information exchange via event-driven communication architecture
Reactor Network– SVF Implementation

- Python-based implementations of local Model Predictive Controllers (NMPC) and the coordinator
  - Integrated using the C-Python API and the SVF external function interface
- Iterative information exchange via the event-driven communication architecture
Summary and Outlook

- A plug-and-play approach for simulation-based validation of distributed management and coordination architectures on simulation models of CPSoS
  - Reduces the currently large engineering effort by defining standard interfaces
    - Increased re-usability
    - Simplifies the deployment of new distributed architecture to real-world automation hardware
  - The Modelica-based framework provides an interface for the connection of external controller software components

- Under development: A software for the automatic generation of the interconnections and the communication structure
Thank you for your attention!

This project has received funding of European Commission under grant agreement number 611281. www.dymasos.eu