Real-Time HWIL Simulation of Liquid Food Process Lines

Magnus Gäfvert[a] Tomas Skoglund[b]
Hubertus Tummescheit[a] Johan Windahl[a]
Hans Wikander[c] Philip Reuterswärd[a]

[a]Modelon AB
Ideon Science Park, SE-223 70 Lund, Sweden
magnus.gafvert@modelon.se

[b]Tetra Pak Processing Systems
Ruben Rausings gata, SE-221 86 Lund, Sweden
tomas.skoglund@tetrapak.com

[c]Avensia Innovation AB
Gasverksgatan 1, SE-222 29 Lund, Sweden

Abstract

This paper describes a newly developed Modelica and Dymola based solution for hardware-in-the-loop (HWIL) simulation in the food processing industry. The solution has been evaluated for potential larger scale deployment into the operational processes of Tetra Pak Processing Systems. The solution consists of a real-time enabled model library for liquid food processing, which is compiled into a process simulator using Dymola, and custom developed software for communication between the process simulator and a production PLC control system using industry standard OPC protocols.

Keywords: physical modeling and simulation; hardware-in-the-loop; liquid food processing; process simulation; real-time simulation

1 Introduction

Dynamic simulation of liquid food process lines, e.g. pasteurization lines in dairies, see Figure 1, has already been practiced in a systematic way by means of the FoodProcessing library (FP), see Figure 2. This Modelica [1] and Dymola [2] based dynamic model library developed for in-house use has previously been reported in [3] (Skoglund, 2003), [4] (Skoglund and Dejmek, 2006) and [5] (Skoglund, 2007). Besides the fundamental laws of conservation, e.g. mass and energy, the model library addressed particular characteristics of liquid food process lines. For example dynamic propagation of fluid properties was considered due to the need of simulating start-up and shut-down with fluid changes, which are occurring frequently in the addressed applications.

Within the operations of Tetra Pak Processing Systems, the FoodProcessing library was used to simulate many processes with their control system as a tool for development or improvement. Simulation was also used as a means for trouble shooting.

Figure 1. A typical dairy process-line for pasteurization.

In the regular delivery process of Tetra Pak's order handling, food processing units are functionally tested by running them with water before they are shipped to the customers. This is carried out to secure high quality of the equipment. The test cannot be carried out before the machine is manufactured, which leads to the need of extra time before delivery. The test itself also requires costly test places with water, steam, electricity, compressed air and drain available. Also, water has different properties compared to the liquid food that will eventually be processed by the unit which means that the test result may deviate from real plant performance.

To enable shorter delivery time at a lower cost, alternatives to this functional test were investigated. One of the alternatives is to run real-time hardware-in-the-loop (HWIL) simulation where the real PLC (Programmable Logic Controller) control system is connected and run with the process model. Since the process model enables simulation with not just water, but real fluid models the HWIL simulation may also, in some cases, be more realistic. Furthermore, often the normal water test does not include special equipment (centrifugal separators or equipment upstream/downstream) due to practical problems. For simulation, this limitation does seldom exist. In simulation it is also possible to monitor virtually any dynamic variable in the system without the need for sensors, which may be of great help to quickly understand and resolve issues.

Furthermore HWIL simulation enables other possibilities, e.g. as a test, validation, and verification tool in PLC software development, and operator training [6] (De Prada et al., 2003) and [7] (Bäckman and Edwall, 2005).

This article describes:

- How the model library was adapted for real-time simulation
- How a communication program was developed as a link between the PLC and the simulator.

The work was carried out as a project with Tetra Pak Processing Systems and Modelon.

2 The "FoodProcessing" library

Since the start of the development of the "FoodProcessing" (FP) library [3] (Skoglund, 2003) much more work was spent to address characteristics of liquid food process lines. Thus Skoglund et al. (2006) [8] described a way to handle fluid transitions in heat exchangers that leads to thermal transients. A model for axial-dispersed plug flow (ADPF) was also described [9] (Skoglund and Dejmek, 2007) and extended to model first-order reaction kinetics [10] (Skoglund and Dejmek, 2007). Figure 2 shows the FoodProcessing library in the Modelica tool Dymola. The library has since been used to configure many process lines and to investigate various performance issues, e.g. product losses. Thus the development of a mixing zone was simulated for product filling and emptying in a commercial UHT line for milk sterilization [11] (Skoglund & Dejmek, 2007). The result was compared with measured data.

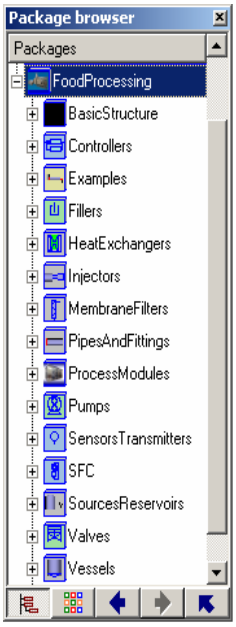


Figure 2. The FoodProcessing library

The library was also used for trouble shooting and testing new design ideas, both concerning process design and control algorithm.

3 Real-Time Aspects

The original FoodProcessing library was designed for high-fidelity desktop simulation with variable-step high-order solvers and many models were not suited for real-time simulation. It was decided to translate and adapt the library into a real-time FoodProcessing library (FPRT). The models in FPRT are made to simulate robustly with fixed-step solvers with a computational load that avoids computation over-runs when executed in real time on standard PC:s.

One major difficulty is the nonlinear equation systems that appear from the pressure dynamics for pipe networks with incompressible fluids. A related difficulty is the effective structural change that valve closing and opening implies on the equation systems. The overall system contains stiff modes and requires implicit methods for numerically stable integration. The inline integration feature of Dymola is used to take advantage of the symbolic reduction and transformation. To reduce the sizes of the resulting equation systems an explicit integration routine was introduced by inlining Modelica code in some components in the library. (The mixed implicit/explicit inline integration in Dymola does not handle the present type of models.)

Several numerical tweaks were introduced to increase robustness [12]. Several models have also been simplified and discretization grids have been made smaller. The number of dynamic states has been reduced.

4 HWIL Setup

HWIL simulation is often performed on dedicated computers with real-time operating systems and extensive I/O possibilities. In the present application the solution should be able to run on a standard PC with Microsoft Windows operating system with Ethernet or automation-bus communication with the PLC hardware. This means that hard real-time cannot be ensured. The sampling rate of feedback control-loops in the food processing applications are in the range of 100 ms or longer, and sufficient performance can be met with soft real-time.

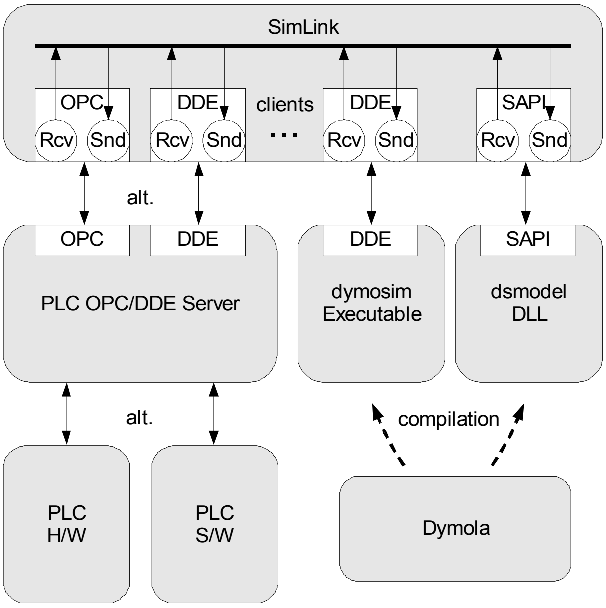


Figure 3. Overview of HWIL setup with SimLink signal routing application, PLC communication server, PLC hardware or emulated controller, and dymosim executable or dsmodel DLL process simulator.

The HWIL setup consists of a PLC system with control algorithms and programs, the process simulator, and a software to route signals between the PLC and the simulator. See Figure 3.

4.1 PLC System

PLC systems are digital computers for automation with extensive support for I/O arrangements and bus communication. The PLC computers host control programs for sequence control and sampled data feedback control typically expressed with IEC 61131-3 languages such as ladder diagrams or function block diagrams.

Tetra Pak work with several suppliers of PLC systems, such as Rockwell Automation, see Figure 4, and Siemens, and the HWIL solution must support them all. Most major systems support the DDE (Dynamic Data Exchange) and OPC (OLE for process control) technologies for interoperability [13,14].

Figure 4. Allen Bradley PLC Controller from Rockwell Automation.

DDE is an old technology for communication between multiple applications under Microsoft Windows. OPC is a standard protocol for open connectivity in industrial automation. DDE suffers from scalability and performance issues, and is more or less being superseded by newer technology. Therefore, the communication between the simulator and the PLC was decided to build mainly on OPC with DDE as fallback.

OPC is originally designed for communication with HMI (Human-Machine Interface) units, operator panels, and enterprise systems with moderate to low requirements on data bandwidth. The standard has then evolved to cover a wider class of communication tasks in industrial automation. OPC supports synchronous and asynchronous communication and is highly flexible and scalable. OPC is not primarily intended for feedback control or communication with high-bandwidth hard real-time requirements. With a soft real-time performance of about 400 items at a rate of about 20-30 ms, or 2000 items at about 100 ms it is still deemed sufficient for the present HWIL application.

In HWIL simulation the I/O signals in the PLC are re-routed from the physical I/O card to memory addresses associated with OPC items. This means that the PLC programs must be extended with a simulation mode to support simulated I/O.

Some vendors, such as Rockwell Automation, also offer emulators for their PLC computers. This makes it possible to also work with SWIL (software-in-the-loop) with the same setup.

4.2 Process Simulator

Dymola supports, via the dymosim executable, stand-alone real-time simulation with DDE communication. All variables in the model are then available as DDE items for subscription. The performance and scalability issues with DDE mean that alternative solutions have also been investigated.

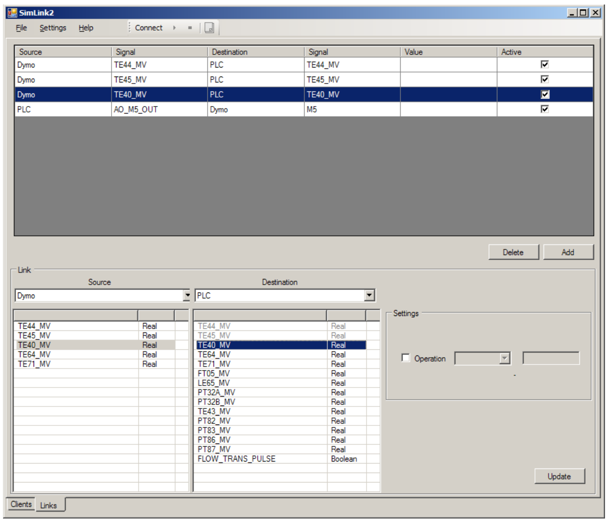
One attractive solution is to use a model DLL (Dynamically Linked Library) similar to that used in the DymolaBlock that enables Dymola models to be used in MATLAB/Simulink. This means that an external integration routine is used and the model derivatives and outputs are returned by direct function calls. An SAPI (Simulation Application Programming Interface) for calling the simulation model as a function was therefore developed together with build scripts to produce the model DLL. For real-time simulation the integration routine is a fixed-step explicit Euler with event management. The direct function calls means that virtually all communication overhead is eliminated.

4.3 Signal Routing with SimLink

The number of signals in a HWIL setup may be in the range of a few ten for small process modules, to several hundred for large processes. The signals represent all sensor and actuator values that logically connects the process with the PLC, but may also include values from "virtual sensors" that are not available on the real process. The signals may also represent alarms and warnings from the model components, for example to alert the user of operating points outside the range of validity.

A core component in the HWIL setup is the organization and synchronization of the signal routing. The SimLink software described in the following was developed for this purpose.

SimLink can be viewed as a coupling panel where input and output signals from different clients are connected or linked via a graphical user interface. SimLink is a Windows application and builds on the Microsoft .NET Framework and is based on OPC Core Components and OPC .NET API 2.0. SimLink is configured by specifying a set of clients, defining their signals, and then connect the signals by introducing links, see Figures 5 and 6. The configuration of clients, signals and links can be saved to a configuration file, that later can be loaded into the application. This makes it easy to maintain different configuration setups. The configuration file contains all information of the setup, and is stored in a human readable XML format.



Source	Signal	Destination	Signal	Value	Active
Dymo	TE44_MV	PLC	TE44_MV		☑
Dymo	TE45_MV	PLC	TE45_MV		☑
Dymo	TE40_MV	PLC	TE40_MV		☑
PLC	AO_M5_OUT	Dymo	M5		☑

Source: Dymo

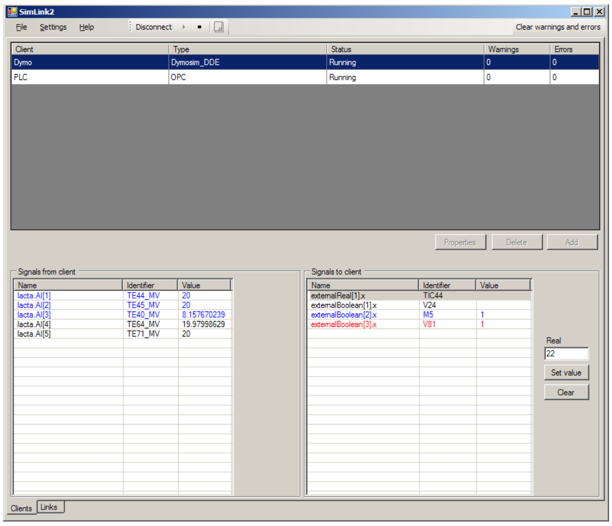
TE44_MV	Real
TE45_MV	Real
TE40_MV	Real
TE64_MV	Real
TE71_MV	Real

Destination: PLC

TE44_MV	Real
TE45_MV	Real
TE40_MV	Real
TE64_MV	Real
TE71_MV	Real
FT05_MV	Real
LE65_MV	Real
PT32A_MV	Real
PT32B_MV	Real
TE43_MV	Real
PT82_MV	Real
PT83_MV	Real
PT86_MV	Real
PT87_MV	Real
FLOW_TRANS_PULSE	Boolean

☐ Operation

Figure 5. The Links view displays details on signal links and offers a convenient user interface for connecting clients.



Client	Type	Status	Warnings	Errors
Dymo	Dymosim_DDE	Running	0	0
PLC	OPC	Running	0	0

Signals from client

Name	Identifier	Value
lacta.AI[1]	TE44_MV	20
lacta.AI[2]	TE45_MV	20
lacta.AI[3]	TE40_MV	8.157670239
lacta.AI[4]	TE64_MV	19.97998629
lacta.AI[5]	TE71_MV	20

Signals to client

Name	Identifier	Value
externalReal[1].x	TIC44	
externalBoolean[1].x	V24	
externalBoolean[2].x	M5	1
externalBoolean[3].x	V81	1

Real: 22

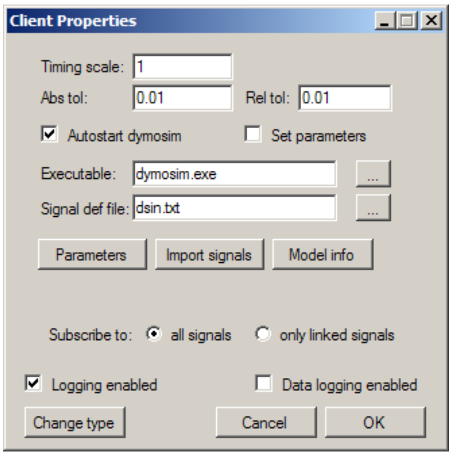
Figure 6. The Clients view gives an overview of configured clients and signals. Signals can be monitored and manipulated in run mode.

When the setup is finished, the next step is to connect to the clients. The application then makes sure that it has valid connections to all clients and that they are ready to send and receive signals. After connecting, the application is ready to go into run mode, and this is done by clicking the *play*-button. In run mode, SimLink is listening to all signals sent to the clients from external applications and internally routes them through the links to clients connected to the receiving applications. Run mode is ended by clicking the *stop*-button.

SimLink currently supports the client types listed in Table 1. Figures 7 – 9 shows the properties dialogs for the DDE, OPC, and SAPI client types.

Table 1. SimLink client types.

Client type	***Description***
DDE	Connects to programs that support Windows DDE.
OPC	Connects to programs with an OPC server
SAPI DLL	The simulation model resides in a DLL that is loaded into the client. The client contains the simulation algorithms and communicates with the model through a direct function API (SAPI).
Internal	Signal sink for testing purposes.
Trigger	Signal source that generates output signals at a specific rate.


Client Properties

Timing scale: 1

Abs tol: 0.01 Rel tol: 0.01

☑ Autostart dymosim ☐ Set parameters

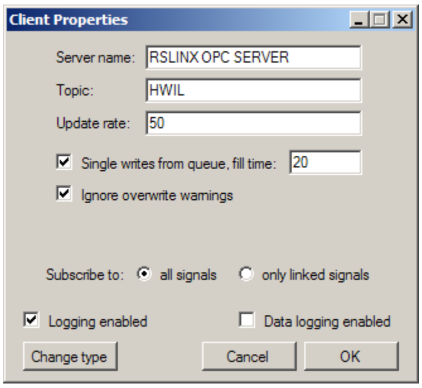
Executable: dymosim.exe

Signal def file: dsin.txt

Subscribe to: ● all signals ○ only linked signals

☑ Logging enabled ☐ Data logging enabled

Figure 7. Properties for DDE client.


Client Properties

Server name: RSLINX OPC SERVER

Topic: HWIL

Update rate: 50

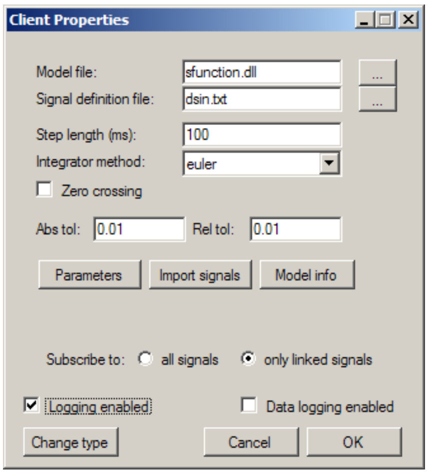
☑ Single writes from queue, fill time: 20

☑ Ignore overwrite warnings

Subscribe to: ● all signals ○ only linked signals

☑ Logging enabled ☐ Data logging enabled

Figure 8. Properties for OPC client.


Client Properties

Model file: sfunction.dll

Signal definition file: dsin.txt

Step length (ms): 100

Integrator method: euler

☐ Zero crossing

Abs tol: 0.01 Rel tol: 0.01

Subscribe to: ○ all signals ● only linked signals

☑ Logging enabled ☐ Data logging enabled

Figure 9. Properties for SAPI client.

The SimLink OPC client is to date verified to support PLC systems from Rockwell Automation and Siemens.

5 Process Examples

The process line that was chosen for the evaluation of the real-time HWIL simulation was a custom designed commercial processing module for dairy pasteurization[1], see also Figure 1. Figure 10 shows the top-level model diagram (flow chart) as configured by using the library FPRT.

The process consists of a balance tank, a plate heat exchanger for pre-heating and pasteurization, a deaerator, a homogenizer, holding cell, steam-powered hot-water unit, and pumps, valves, and sensors being monitored and controlled by the PLC system. The process supports a number of operating modes, e.g., start-up, production, cleaning, and hibernation, which have different flow configurations.

---

[1] Tetra Therm Lacta, designed and manufactured by Tetra Pak Processing Systems.

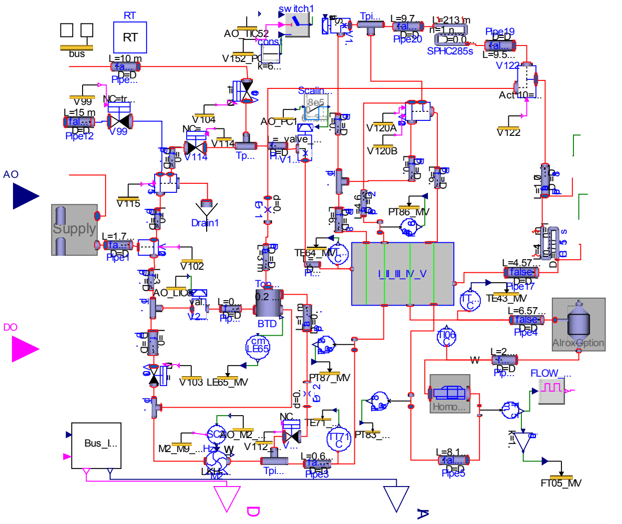


Figure 10. Flow diagram of the process line used to evaluate real-time HWIL simulation. The model is built in a hierarchy. The figure shows the top level view.

Figures 11 and 12 show the PLC operator panel that is used to control and monitor the process.

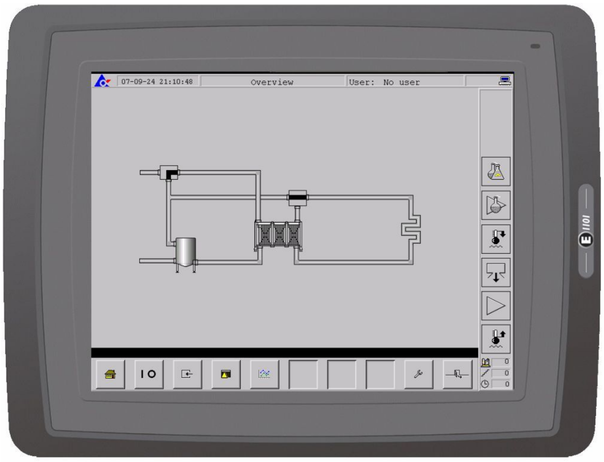


Figure 11. An overview picture of the PLC operator panel.

6 Results

The described HWIL solution is being evaluated both from a business perspective and a technical perspective. Technically, the presented solution seems to fulfil all given requirements. There have been a number of minor issues that have been resolved, but the core solution design and architecture has shown to be sound, scalable, and extensible.

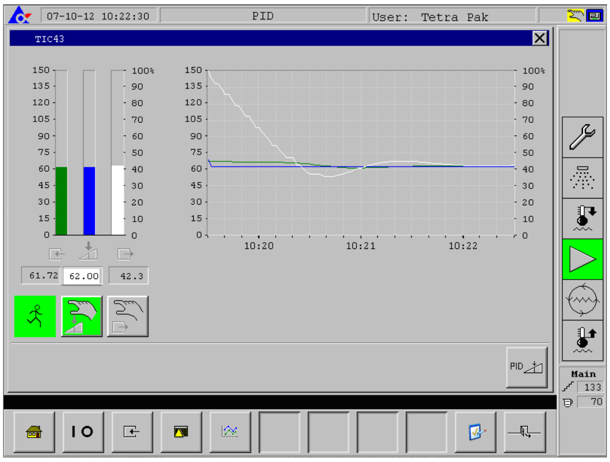


Figure 12. PLC operator panel showing the performance of a PID controller regulating a simulated process.

A HWIL simulation setup was prepared for the process module described in Section 5 with its production PLC system from Rockwell Automation (hardware and software). A number of evaluations were arranged where the HWIL testing procedure was compared with the traditional water testing described in Section 1. The tests were built on cases from water testing of actual units delivered to customers and now in production. The models and PLC programs were rigged with a set of faults from a database and testing personnel then performed an HWIL testing to see if these faults would be identified. The protocols from this HWIL testing reported all or most problems also found in water testing, and then some additional that was not found in water testing. The results so far indicate that HWIL simulation may indeed replace water testing and also result in better test coverage.

One comment from the de-briefing of the test personnel was that the simulated process was lacking the noises, sounds, vibrations, and other sensory information that is used by humans to monitor the process and detect deviations. Still, the overall impression was that the simulator had advantages over water testing. The SimLink software was extended with alarm/alert functionality to support emulated sensory information from the simulation model. A component can, for example, trigger a boolean signal to indicate that the fluid media is boiling. On the real process this might have been detected by noise. In SimLink this triggers an alarm that alerts the user of the abnormal situation.

The HWIL setup was also in parallel used for testing, validation, and verification in development of new PLC control software. The HWIL setup has proven to be very useful to find and identify issues and bugs at an early stage.

There have been some problems with robustness of the simulation at complex mode switches. Possibly, this will lead to some re-design of the fundamental models of the fluid dynamics. For example, the current models are designed for uni-directional flow, even though back-flow may occur in transients and mode switches.

7 Conclusions

Experiences from the presented solution indicate that the Modelica based HWIL technology may contribute significantly and in a wide range of operations in a business organization like Tetra Pak Processing Systems. Parts of a larger evaluation effort have been performed and indicate that expensive and time-consuming water testing may be replaced by simulation. Application of the HWIL solution for software debugging has also been done successfully.

Large parts of the FP library have been adapted for real-time simulation with the evaluated process module. Remaining parts will be adapted as HWIL simulation is introduced for other process modules.

The SimLink program was designed and developed for general HWIL simulation with PLC systems in the process industry. It has been continuously improved during the work described in this paper and has now become a stable and feature complete core component in the HWIL simulation environment.

References

[1] Modelica Association, http://www.modelica.org

[2] Dynasim AB, http://www.dynasim.se

[3] Skoglund, T. Simulation of Liquid Food Processes in Modelica. Proceedings of the 3rd International Modelica Conference 2003, 51-58. Linköping, Sweden, November 3-4, 2003, Organized by the Modelica Association and Linköping University, Sweden. Available at http://www.modelica.org.

[4] Skoglund, T. and Dejmek P. A model library for dynamic simulation of liquid food process lines. Proceedings of FOODSIM 2006, 5-12, Naples, Italy, June 15-17, 2006, Organized by EUROSIS.

[5] Skoglund, T. Dynamic Modelling and Simulation of Liquid Food Process Lines. Lund, Sweden: Doctoral thesis, Department of Food Technology, Engineering and Nutrition, Faculty of Engineering, LTH, Lund University, 2007.

[6] De Prada, C., Merino, A., Pelayo, F., Acebes, F., Alves, R. A simulator of Sugar Factories for Operator Training, AfoT 2003, II International Workshop on Information Technologies and Computing Techniques for the Agro-Food Sector, Barcelona, Spain, November 27-28, 2003, Monograph CIMNE No-86.

[7] Bäckman, J., Edvall, M. Using Modelica and Control Systems for Real-time Simulations in the Pulp. Proceedings of the 4th International Modelica Conference, March 7-8, 2005, pp. 579-583, Hamburg University of Technology (TUHH), Germany. Organized by the Modelica Association and TUHH. Available at www.modelica.org

[8] Skoglund, T., Årzén, K-E. and Dejmek, P. Dynamic object-oriented heat-exchanger models for simulation of fluid property transitions. International Journal of Heat and Mass Transfer, 2006, 49, pp. 2291-2303.

[9] Skoglund, T. and Dejmek, P. A dynamic object-oriented model for efficient simulation of fluid dispersion in turbulent flow with varying fluid properties. Chemical Engineering Science, 2007, 62, pp. 2168-2178

[10] Skoglund, T. and Dejmek, P. A dynamic object-oriented model for efficient simulation of microbial reduction in dispersed turbulent flow. Journal of Food Engineering, 2008, 86, pp. 358–369.

[11] Skoglund, T. and Dejmek, P. Fuzzy traceability – A process simulation derived extension of the traceability concept in continuous food processing. Trans IChemE Part C, Food and Bio products Processing, 2007, 85 (C4) pp. 354-359.

[12] Tummescheit, H. Design and Implementation of Object-Oriented Model Libraries using Modelica, Doctoral thesis, Lund University, 2002.

[13] Iwanitz, F., Lange, J., OPC Fundamentals, Implementation, and Application, 3rd Ed., Hüthig Verlag Heidelberg, 2006.

[14] The OPC Foundation. http://www.opcfoundation.org