HARDWARE-IN-THE-LOOP REAL TIME SIMULATION WITH OPENMODELICA FOR PROCESS AUTOMATION

V. Muralidharan[1], Prabir Purkayastha[2], Prof. Peter Fritzson[3], Dr. S. Rominus Valsalam[1], Prasad R[1], Abhir Raj Metkar[1], Roshna C P[1]

[1]Centre for Development of Advanced Computing (CDAC),
Control and Instrumentation Group, Trivandrum, Kerala, India
[2]Evonik Energy Systems pvt. Ltd. Delhi, India
[3]Linkoping University, Faculty of Computer Engineering,
Linköping, Sweden

ametkar@cdactvm.in (Abhir Raj Metkar)

Abstract

Control Engineers, Plant Operators and Research Scholars often need to work with equipment and systems for learning, designing and testing control systems. With the developments in Information Technology, there has been a totally new approach to the teaching, developing and experimenting with Advanced Control Algorithms for Process Control. Today, process models developed in Object Oriented Modelling languages are widely used to simulate real processes and rigorously design and test appropriate control systems. In todays modern Plant Automation and Control Systems, the Simulation platform is an integral part of the total system. This is provided to be able to learn the process behavior on-line during the normal operation of the plant and improve the performance of the controllers. An accurate simulation model allows operators to train under "live" conditions without exposing the plant to the consequences of their mistakes. The Hardware-in-the-loop set-up is formed using models in OpenModelica and a controller in a PC/104 module. One, two, and three tank systems were modeled in OpenModelica and the level control problems were simulated. The Real Time Synchronization module was a crucial development for interfacing the OpenModelica models and the PID controller in PC/104 system. The simulation results are useful for tuning the controllers and for operator training.

Keywords: Real-Time Simulation, OpenModelica, PC/104- Embedded Controller, Object Oriented Modelling, Connectivity matrix.

V. Muralidharan

Completed B.E in the year 1976 and M.E (Applied Electronics), PSG Tech, Coimbatore in the year 1978. Currently working as Additional Director and heading the Control and Instrumentation Group of Centre for Development of Advanced Computing, Thiruvananthapuram.

1 General

The increased use of real- time simulation as a tool for learning about complex systems and deriving the controller parameters on-line to take care of the changing environment is one of the most significant on-going developments in recent times. Industrial automation systems now move towards such smart operation. Advances in computer, communications, software, mathematical analysis, and process sensing and measurement devices have led to new methods and procedures for operating production facilities that offer the following:

- More Comprehensive and frequent measurements of the current state of the plant
- Increased use of models and their analytical techniques to compare what the plant is currently producing against what is expected and to understand the differences
- Early detection of anomalous conditions
- Tools to plan future operation with increased confidence

While we may be aware of these developments as individual advances, their cumulative and combinational aspects are perhaps less well recognized. The combination of these technologies has led to an evolutionary change in the way plants can operate. This change affects decisions and actions based primarily on the best available prediction of expected future conditions rather than ones principally triggered by reactions to what has just happened. This shift in focus is the defining characteristic of the "smart plant". Now we can improve the overall decision process by knowing better what the plant is doing. Now, better comparison of what the plant is doing against what it is expected to do, and understanding the differences, leads to model-based analysis and techniques to better comprehend the information. The result is improved by predicting the effect of alternate decisions in the future [5].

2 Modelling Specifications and Methodology

Advanced process control and supervision requires accurate process models. In order to develop these accurate process models, an object oriented model building tool is to be used. The process models are broadly divided into two categories: multivariable quantitative models and multivariable qualitative models. The quantitative models comprise two categories:

1. Analytical models developed from first principles such as the laws of conservation of energy, momentum etc.
2. Experimental models developed using system identification techniques.

First principles models are developed in terms of static, differential and partial differential equations and hence they are easily accommodated in the state-space structure for state estimation, filtering and prediction. This method is applied to systems that are mathematically well understood.

Solving the pressure flow matrix is the key to the formulation. The pressure flow network is built by defining the topology of the network and using component models of the plant. The component models should be such that they are reusable modules and can be expanded as a set of modules. The network consists of defining nodes and flow streams. Each equipment has ports which connect to other equipment or nodes through flow streams. Flow streams carry the process variables including the state variables of the process. Junctions or splitting of flow streams in the network are nodes in the system. Each flow stream carries variables such as pressure, flows, density, enthalpy, temperature leaving nodes that have to be computed from the pressure flow matrix and subsequent computations.

For each of the components, the specifications are developed including identifying the process equations and the variables for the flow streams. This generates a non-linear set of equations. The pressure flow matrix is built out of the above components and the topology defined so that any change of configurations is simply a matter of modifying, adding or deleting the nodes and components. These components and the topology can be embedded in a graphical tool kit using drag and drop methods. This specification will give the required properties to be embedded within the topology and the components so that the models can be generated through such graphical means.

2.1 Building a Model

The tank system is formed by connecting several basic components together. The following components will be built as basic models [3]. The basic components for building process model & network:

1. Valves (variable resistance)

2. Nodes
3. Source
4. Sink
5. Tanks
6. Resistance (long pipes or flow obstructions)
7. Pumps

For a model the variables are Pressure and Flow associated with each flow-stream. So the number of variables in a system is twice the number of flow-streams in that system. Also some more variables can be added as tank level, for tank system.

Model Equations

Models equations are represented in Pressure-Flow Matrix.

Linear and Incompressible flow systems— **AX=b**

Where,
A=Matrix of coefficients of variables
X=Matrix of variables
b=Matrix of constants

Non-Linear Systems— $\mathbf{AX+BX^2+CX^3+\ldots=b}$

Where,
A= Matrix of coefficients of variables of order 1
B=Matrix of coefficients of variables of order 2
C=Matrix of coefficients of variables of order 3
b=Matrix of constants

2.3 Pressure-flow Equations

Source component

Source pressure is constant say, b1

$P_{FL1}=b_1$……………………………..……… (1)

Control Valve 1 component

$M_{FL1}= K_1 \times (P_{FL1}-P_{FL2})$…………..……… (2)

K_1=conductance × density

Constant pressure across CV_1 say, b_2

$P_{FL2}=b_2$………………………………….... (3)

Conservation of mass

$M_{FL2}=M_{FL1}$…………………..………….. (4)

Tank component

$$\left[\begin{array}{l} V=\dfrac{m}{\rho}=L\times A \\ \therefore L=\dfrac{Vin-Vout}{A}=\dfrac{m/\rho(in)-m/\rho(out)}{A} \end{array}\right]$$

$L_{T1}=Lo+[(1/K_2)\times(M_{FL2}-M_{FL3})]$……….. (5)

K_2=density × area of tank

$P_{FL3}=P_{FL2}+(K_3\times L_{T1})+b_3$…………….... (6)

K_3=density × conversion factor

b_3=tank elevation × density × conversion factor

Control Valve 2 component

$M_{FL3}= K_4 \times (P_{FL3}-P_{FL4})$……..…………. (7)

K_4= conductance × density

Sink component

Constant pressure across CV_2 say, b_4

$P_{FL4}=b_4$ ………………………………….. (8)

Conservation of mass

$M_{FL4}=M_{FL3}$……………………………… (9)

Components used (5)

S_1=Source

CV_1=Control valve$_1$

T_1=Tank

CV_2=Control valve2

Si_1=Sink

Flow streams Total Variables (9)

1. FL1-between S_1 & CV_1
2. FL2-between CV_1& T_1
3. FL3-between T_1 & CV_2
4. FL4-between CV_2 & Si_1

The variables are: P_{FL1}, M_{FL1}, P_{FL2}, M_{FL2}, L_{T1}= the tank level, P_{FL3}, M_{FL3}, P_{FL4} and M_{FL4}

Connectivity matrix

$$\begin{array}{c} \\ S1 \\ CV1 \\ T1 \\ CV2 \\ Si1 \end{array} \begin{array}{c} \begin{array}{ccccc} S1 & CV1 & T1 & CV2 & Si1 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Equations in P-F matrix form

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & K1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1/K3 & 1 & 0 & 1/K3 & 0 & 0 \\
0 & 0 & -1 & 0 & -K3 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & K4 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
PFL1 \\ MFL1 \\ PFL2 \\ MFL2 \\ LT1 \\ PFL3 \\ MFL3 \\ PFL4 \\ MFL4
\end{bmatrix}
=
\begin{bmatrix}
b1 \\ 0 \\ b2 \\ 0 \\ 0 \\ b3 \\ 0 \\ b4 \\ 0
\end{bmatrix}
$$

Nomenclature of parameters:

Parameters	Description
P_{FL1}	Pressure Flow-stream 1
M_{FL1}	Mass Flow-stream 1
P_{FL2}	Pressure Flow-stream 2
M_{FL2}	Mass Flow-stream 2
P_{FL3}	Pressure Flow-stream 3
M_{FL3}	Mass Flow-stream 3
P_{FL4}	Pressure Flow-stream 4
M_{FL4}	Mass Flow-stream 4
L_{T1}	Level tank 1
Lo	Initial Level in tank
V	Volume
m	mass
ρ	density
L	level
A	Cross section area of tank

3. OpenModelica for Simulation

3.1 Introduction

Object oriented modeling is a fast-growing area of modeling and simulation that provides a structured, computer-supported way of doing mathematical and equation-based modeling. Modelica is today the most promising modeling and simulation language in that it effectively unifies and generalizes previous object-oriented modeling languages and provides a sound basis for the basic concepts.

The Modelica modeling language and technology is being warmly received by the world community in modeling and simulation with major applications in virtual prototyping. It is bringing about a revolution in this area, based on its ease of use, visual design of models with combination of lego-like predefined model building blocks, its ability to define model libraries with reusable components, its support for modeling and simulation of complex applications involving parts from several application domains, and many more useful facilities. To draw an analogy, Modelica is currently in a similar phase as Java early on, before the language became well known, but for virtual prototyping instead of Internet programming [1][2].

3.2 Component based tank models

The tank system model is formed by combining several individual components like, source, valve, tank and sink. Each of the components are connected to each other with connector ports, which are defined as interface components.

In particular for a one tank system we need five components as source, valve1, tank, valve2 and sink. Between each pair of components there is a flow-stream. The number of unknowns is 2 times the flow-streams in a system (as pressure-flow variables). Hence the total numbers of unknowns are 8 in this system. Therefore we require minimum 8 equations to model and simulate this system. In the tank model we have tank level as an extra variable, so total number of variables is 9 for the one tank system. The equations are as described below:

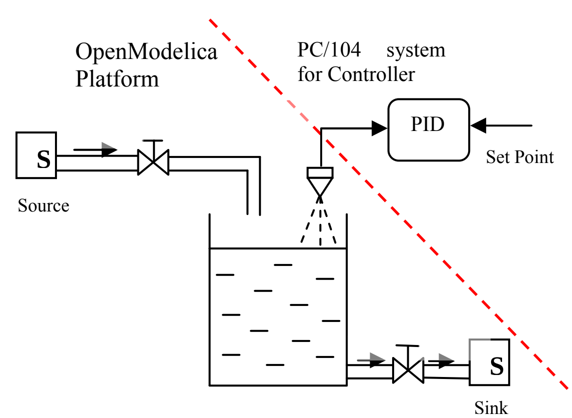

Fig 1: One Tank System

The two tank model can be formed by connecting the valve at the output of first tank as input of second tank. Similarly connecting the valve at the output of second tank as input of third tank forms the three tank system. Therefore two tank and three tank systems are extensions of one tank system with the same basic components.

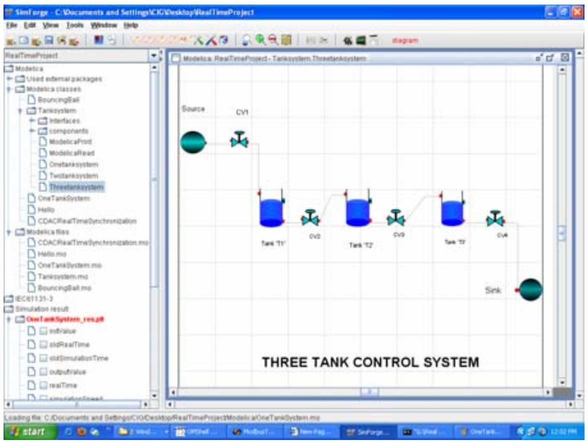

Fig.2: Three tank system model in OpenModelica GUI

4. Hardware-in-the-loop with PC/104

Hardware-in-the-loop (HIL) is the integration of real components and system models in a common simulation environment. This means that some parts of a system, which should be tested, are virtual and other parts are real. HIL simulations are an important method for the development of mechatronic systems. An important advantage of HIL is that it allows function tests of mechatronic systems or components of such systems under simulated real conditions. Therefore HIL helps to save cost and time compared to conventional test runs on a real prototype [4]. There are three important considerations for the implementation of a hardware-in-the-loop simulation:

• The simulation of the dynamic system, in other words the mathematical or physical models must be processed in real-time.

• There must be synchronization between the time in the real world (the so called real-time) and the digital simulation-time of the simulation tool.

• The simulation tool must be able to communicate e.g. with other applications or an I/O communication interface.

4.1 PC/104 system

The PC/104 system is an embedded controller module used for control applications. The PC/104 used in the setup is of industrial make. The CPU-1433 features the AMD™ Geode™ GX466 @333MHz Pentium MMX class processor and is Linux compatible.

IO Modules

The IO module is a versatile, PC/104-based analog input, analog output, and digital I/O board for high-accuracy and high-channel count analog and digital I/O. The board is based upon Linear Technologies' precision converters and voltage references which require no external calibration. The IO operates over the industrial temperature range of -40° to +85° Centigrade.

4.2 PID Controller in PC/104 system

A PID controller is a continuous controller and it is suitable for tank level control problem. The PID controller is configured in PC/104 system. It takes recent tank level as input and compares with the set point (configurable). The error is given as output of controller to control valve manipulation. The tank system model simulation in OpenModelica and the PID controller in PC/104 system have to be time synchronized to achieve the level control.

4.3 Real Time Synchronization Module

The CDACRealTimeSynchronization module has been developed for Openmodelica, under the Windows XP operating system. This module forces the simulation to run the simulation synchronized with real time and can be attached to any models to impart real time capabilities to them.

In order to synchronize the simulation with the real-time and in order to exchange the model inputs and outputs with the external world, some external C functions calls have been added in the CDACRealTimeSynchronization module. These functions are:

- initTime()
- getRealTime()
- ModelicaPrint()
- ModelicaRead()

If initial() is true in the Modelica code, initTime() performs the initialization of the hardware timer on main board or the initialization of the real-time process associated with modelica. During the execution of the program the function getRealTime() returns the current hardwire timer value of the CPU main board. At each sampling time the updated values of the time based variables are evaluated and compared. This program limits the simulation process by limiting the simulation time as well as simulation speed. As long as simulation time is behind realtime the simulation runs as fast as possible. When simulation time is ahead of real time the simulation runs slowly. So it is understood that simulation speed should be controlled. Simulation speed is calculated as quotient of elapsed simulation time and elapsed real time.

simulationSpeed=(time-oldSimulationTime)/ (realTime - oldRealTime)

As long as simulation time is ahead real time or simulation speed exceeds maximum simulation speed, simulation should be paused. We achieved this goal using a while clause.

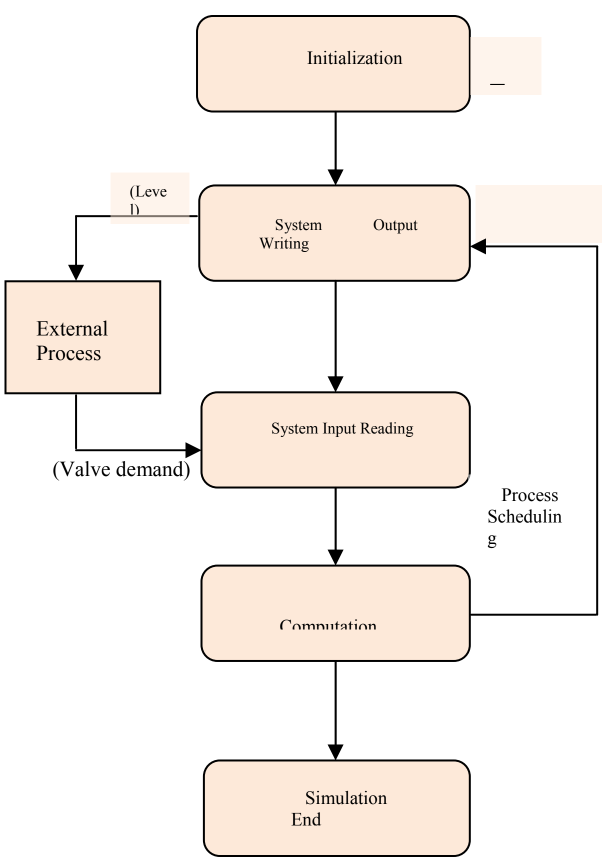

Fig.3: The simulation process execution flowchart

5. Possible Industrial Applications

Monitoring and control of filter-bed operation is very important for efficient operation of water treatment plants where accurate measurement as well as control has traditionally been a challenge. With a case study and subsequent implementation of an indigenous computerized automation system developed by CDAC, Thiruvananthapuram for the Kerala Water Authority, Water Treatment Plant No. III at Aluva, it is proved that considerable improvements in filter-bed life, water quality, and productivity, utilization of wash water and minimization of wastage of water can be achieved with a good automation system [6]. To guarantee a high drinking water quality and high productivity, CDAC is currently developing a modern automation system for water treatment plants to address these needs.

The filter-bed automation system regulates the filtered water flow of each filter-bed to assure the best quality of filtered water output. The controller for regulating the flow rate of filter-bed is to be tuned for optimum control. As an industrial application the Hardware-in-the-loop real-time simulations with OpenModelica is proposed to simulate the water filter process to enhance the controller performance.

The Combined Cycle Power Plant at Kayamkulam India, having 2 Gas turbines and a single Steam turbine with total generation of 350 MW. The gas and steam network flow is being modeled and simulated using OpenModelica to determine the optimum main steam electro-hydraulic valve opening characteristics to achieve the auto-coupling of HRSG (Heat Recovery Steam Generator).

Better comparison of actual and expected plant behavior and the understanding leads to model-based analysis and techniques which results better control. Also an interactive real-time simulation allows operators to train under “live” conditions without exposing the plant to the consequences of their mistakes.

6. Results and Conclusions

The paper briefs overview of the hardware-in-the-loop simulation of Tank process models using the object-oriented modeling language Modelica. The OpenModelica based simulation system uses inter-process communication method as ModbusTCP/IP. The important goal was to achieve the real time simulation of the Openmodelica models and to interface it with hardware platform to achieve the system control. The generic module named as CDACRealTimeSynchronization module developed in Openmodelica for real-time synchronization satisfies the constraints of periodic real-time execution.

The real time and simulation time is synchronized and exact overlapping is achieved as shown in Fig. 4. This result is proved further by controller output. As the controller is in PC/104 system the simulation time and real time must be synchronized to achieve control. The three tank system with interconnected tanks which is a generalized form of one tank and two tank system is considered for simulation. The three levels achieving respective desired set points as in Fig. 5 in the interconnected three tank systems simulated in OpenModelica and three PID controllers in PC/104 system demonstrates the hardware-in-loop simulation results successfully. This simulation set-up is useful as, the industrial controller is in the hardware; if results are not satisfactory then the loop can be tuned. This set-up allows tuning the loops off-line with results which would be very close to the real plant. The other advantage of this set-up is for testing advanced control algorithms on a real time simulation platform.

Sampling Time	500ms
Simulation Time	300s

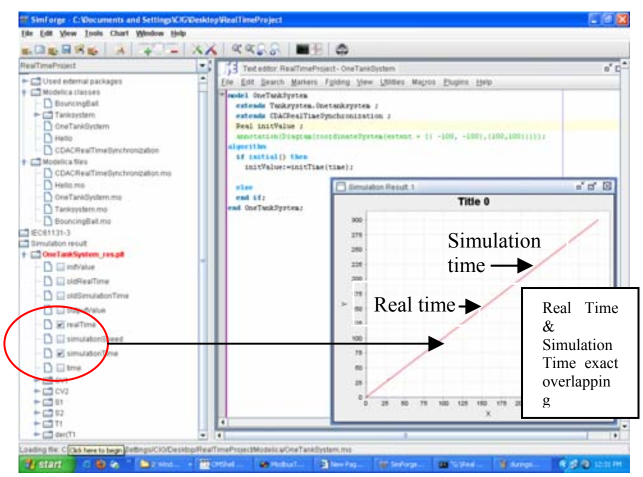

Fig. 4 Real Time & Simulation time Overlapping

Type	Set-point	Kp	Ti	Td
Tank1	2m	0.6	10	1
Tank2	3m	0.4	10	1
Tank3	2m	0.4	10	0

Sampling Time	500ms
Simulation Time	800s

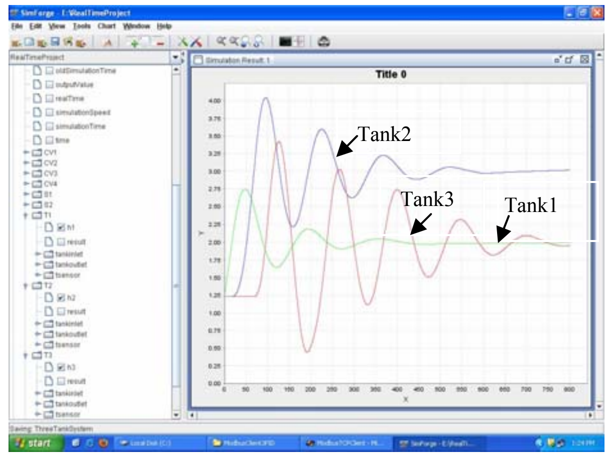

Fig. 5 Three tank level control Simulation output

7. References

[1] Peter Fritzson. Principle of Object Oriented Modeling and Simulation with Modelica 2.1. IEEE Press, Wiley InterScience, 2006.

[2] Michael M. Tiller. Introduction to Physical Modeling with Modelica. Kluwer Academic Publishers, 2004.

[3] http://www.openmodelica.org.

[4] Arno Ebner, Anton Haumer Dragan Simic, Franz Pirker. “Interacting Modelica using a Named Pipe for Hardware-in-the-loop”. The Modelica Association, *Modelica 2006*, Arsenal Research, Giefinggasse 2, 1210 Vienna, Austria.

[5] D. C. White. Creating the ‘smart plant’, Hydrocarbon processing, October 2003.

[6] Abhir Raj Metkar, Dr. S. Rominus Valsalam, Sujimon K T. “An Intelligent SCADA system for Efficiency Enhancement of Water Treatment Plants”. Trends In Industrial Measurements & Applications (TIMA-2009), Chennai, India.