Real-time Linux in chemical process control: some application results

Andrey Romanenko[a], **Lino O. Santos**[a], **Paulo A.F.N.A. Afonso**[b]
[a]GEPSI – PSE Group, Dep. de Engenharia Química, Universidade de Coimbra,
Pinhal de Marrocos, Pólo II, 3030 Coimbra, Portugal
{andrew,lino}@eq.uc.pt

[b]Escola Superior de Tecnologia e Gestão de Águeda,
Universidade de Aveiro,
Rua da Infantaria, 28, 3750 Águeda, Portugal
paulo@estga.ua.pt

Abstract

Many chemical processes require real-time control and supervision in order to operate them safely and profitably, while satisfying quality and environmental standards. As a means to comply with these requirements, it is common practice to use control software based on a proprietary operating system such as QNX, WxWorks, or MS Windows with real-time extensions. To our knowledge, the idea of using Real-Time Linux has not been embraced widely by research and industrial institutions in the area of Chemical Engineering. In this work we describe our experience at the Department of Chemical Engineering of the University of Coimbra, with two pilot plants that are under control of a system based on real-time Linux. One of them is a completed project and the other is under development. The experimental set-ups closely resemble industrial equipment and function in similar operating conditions. We hope that this successful application will encourage further deployment of real-time Linux in the Chemical Engineering research and industry.

1 Introduction

The significance of computers in contemporary science and technology is, undoubtedly, tremendous. The throughoutput of modern microprocessors has multiplied over the last years and so has the speed of communication equipment. Still, the hardware evolution alone would not warranty the positive impact that computers have made. Software, from low-level device specific routines to the most complex applications, is what make use of the equipment potential. Much of this software are proprietary products available on the market. In spite of the existing competition, their prices are considerably high, although often they cover not only the product *per se* but also technical support. Various incompatibilities in software of different vendors only aggravate the situation. Usually the proprietary source code is unavailable which hinders the tailoring of the program to the needs of a particular user and may even raise security concerns.

The above disadvantages of the proprietary software products have prompted the creation of a fascinating movement in the computer technology creating developer communities that work on open source software. This provoked a boost in the number and the range of available applications and their quality. Started as a small group of computer enthusiasts, the community has grown into a major software development force. Nowadays leading software enterprises consider it when they make their key business decisions (*e.g.* `http://www.ibm.com/linux`), which would be unheard of a decade ago. The scope of presently available open source software is big, ranging from system software (operating systems and the like) to user level programs. Unfortunately, it still suffers from a common perception that it is amateur, in the bad sense of the word.

The advantages of the open source model for academia and research are evident. The cost efficiency of free software solutions allows to alleviate the stringent budgeting of educational and research activities. Besides, it is in the very philosophy of science that it be open and that scientific results be distributed quickly [18].

Chemical process control is a very important subject in chemical engineering education. It provides the students with a extensive mathematical background of classical and modern control theories and their applications. If no sufficient hands-on experience is available, the course can become "another mathematical course" [5]. The use of simulators enable the students to get acquainted with a large number of processes, but it fails to provide the real engineering practice. This urged the development of new experimental rigs that vary from simple systems to integrated laboratories. For instance, in [11, 19] the distributed control system concept is employed, an approach widely used in industry. The range of data acquisition and control software, used in these labs, is represented by custom software [6] and proprietary packages, such as Foxboro PW-FB [7], LabVIEW [8], GENESIS [12], LABTECH NOTEBOOK software [20], MATLAB/SIMULINK [2], and others. To our knowledge, the idea of using Real-Time Linux has not been embraced widely by research and industrial institutions in the area of Chemical Engineering. In this work we describe our experience at the Department of Chemical Engineering of the University of Coimbra, with two pilot plants that are under control of a system based on real-time Linux.

The paper is organized as follows. Section 2 describes an experimental set-up that is used in a university course and in two research programs in the field of pulp and paper. This pilot plant has certain advantages as compared to other commercial equipment but they come at the price of a more complicated control scheme. Detailed chemical engineering aspects of its design and application is available elsewhere [14, 9]. Section 3 details a pilot plant that is used in research projects in advanced control topics, such as model predictive and fault tolerant control. Its reaction unit and the control system hardware underwent an upgrade lately. The development of the new control system is currently in progress. Finally, concluding remarks are given in Section 4.

2 Kraft pulping pilot plant

2.1 Description and operation

The reaction unit consists of a set of six equivalent flow through reactors in which wood chips are exposed to a circulating liquid flow containing the kraft pulping chemicals (Figure 1).

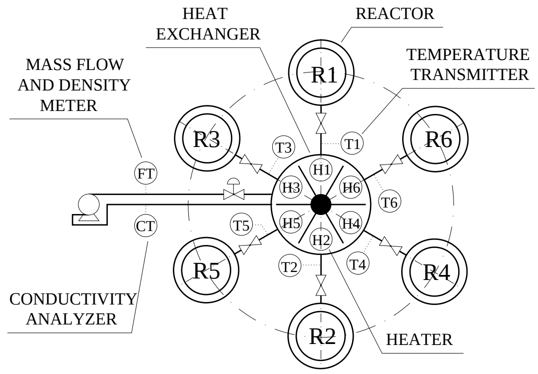


FIGURE 1: *Reaction unit. Top view*

The reaction is highly dependent both on the temperature and on the concentration of liquid reactants. During an experiment the reactors are sequentially shut off in order to stop the reaction and to sample the wood and liquid at different cooking times. The energy required for the chemical reaction is supplied by a 15 kW heat exchanger equipped with six electrical resistances that provide independent heating of the liquid streams circulating through each reactor. As shown in Figure 1, each heater is dedicated to the control of temperature in the corresponding reactor, although it is not fully isolated from the other heaters allowing for temperature compensation at high energy levels. At the bottom of the system (Figure 2) the flows leaving all the pressure vessels are mixed to ensure the same concentration history.

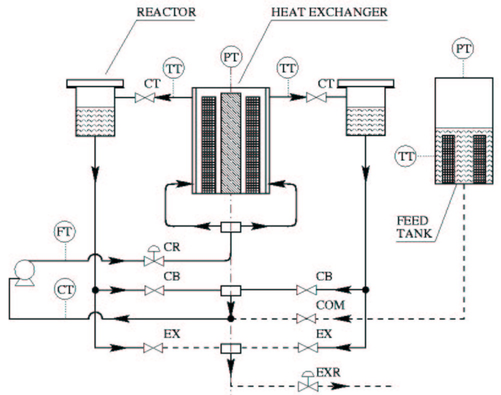


FIGURE 2: *Simplified scheme of the reaction unit*

Each reaction unit branch is equipped with a PT100 temperature sensor (Figure 1). At the top of the heat exchanger a pressure transmitter measures the total pressure in the system. The unit is also equipped with a mass flowmeter providing online flowrate and density measurements, and a conductivity analyzer. The pilot plant's actuators include nineteen on-off valves, a pneumatic control valve, and a circulating pump.

When the system is prepared for an experiment, the pump and the temperature control system are turned on to maintain continuous circulation of the liquid reactants and to provide a pre-specified time history of temperature similar to that often used in the pulp and paper industry. As the experiment proceeds, according to a given sampling strategy the reactors are sequentially shut off and depressurized. Samples of cooking liquor are collected during this depressurization step and the wood chips are taken for chemical analysis. It is critical for the system to follow the desired temperature profiles because chemical reactions that take place in the process of Kraft pulping are very sensitive to temperature. An adaptive control strategy was utilized to address this requirement.

2.2 Data acquisition and control system

In this application, the data acquisition and control system runs under NMT RT-Linux [3] version Beta10, Linux kernel 2.2.10, RedHat Linux 6.0 operating system on an APPRO 5U industrial PC. The rack mount chassis contains a 14 slot passive backplane, a CP-MA51 single board computer with a PENTIUM MMX 200 MHz and 32 MB RAM onboard. Six data acquisition and control boards from Computer Boards Inc. provide twenty four 4-20 mA analog input channels, sixteen 0-10 V analog output channels, two 4-20 mA analog output channels, as well as forty eight isolated digital input channels and forty eight digital output channels. The control code consists of two processes – a time-critical task, containing hardware drivers and necessary scheduling code, and a non-time critical task processing the data and running a graphical user interface. The first task is designed as a real-time module that can be dynamically loaded into the memory. It directly interacts with the hardware and does not need operating system resources, such as access to file systems and networking.

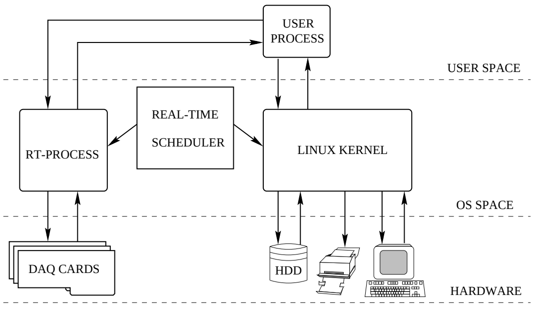
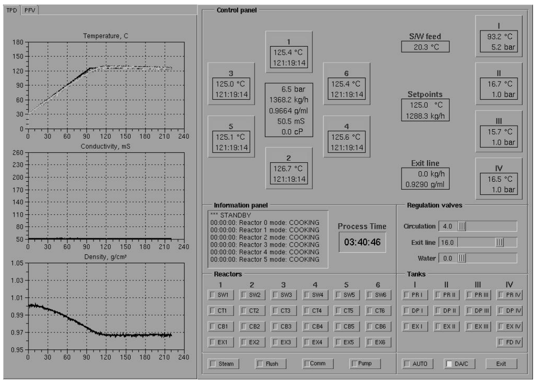


FIGURE 3: *Data flowchart*

The second task is a general user-space UNIX process. The two processes communicate over two FIFO (First-In-First-Out) channels (Figure 3). Both the Linux kernel and the RT-Process are supervised by the Real-Time Scheduler. The former is treated as a low priority real-time task and runs only if other RT-processes are idle. Such decoupling allows the installation of real-time safety "watchdogs" that can bring the pilot plant to a safe state even if there is a serious problem with the user space control module or the operating system itself. The resulting system has features both of a modern UNIX system with all appropriate services at hand and of a hard real-time system able to meet timing requirements typical for processes under control. Besides, its modular architecture simplifies further development, which is an important issue for an educational tool. The graphical user interface, based on the XForms library version 0.88, allows the user to monitor data from all the installed sensors and transmitters and to closely follow the current operating modes of each reactor (Figure 4). Using GUI controls, one can also operate the actuators. The interface has two graphic plot windows in which temperatures, conductivity, density, mass flowrate, and pressure are continuously displayed.



1: 125.4 °C 121:19:14
3: 125.0 °C 121:19:14
6: 125.4 °C 121:19:14
5: 125.1 °C 121:19:14
4: 125.6 °C 121:19:14
2: 126.7 °C 121:19:14
6.5 bar
1368.2 kg/h
0.9664 g/ml
50.5 mS
0.0 cP
S/W feed: 20.3 °C
Setpoints: 125.0 °C 1288.3 kg/h
Exit line: 0.0 kg/h 0.9290 g/ml
I: 93.2 °C 5.2 bar
II: 16.7 °C 1.0 bar
III: 15.7 °C 1.0 bar
IV: 16.5 °C 1.0 bar

```
*** STANDBY
00:00:00: Reactor 0 mode: COOKING
00:00:00: Reactor 1 mode: COOKING
00:00:00: Reactor 2 mode: COOKING
00:00:00: Reactor 3 mode: COOKING
00:00:00: Reactor 4 mode: COOKING
00:00:00: Reactor 5 mode: COOKING
```

Process Time: 03:40:46
Circulation: 4.0
Exit line: 16.0
Water: 0.0

FIGURE 4: *Graphical user interface*

The sampling schedule is defined in a datafile before beginning an experimental run. The policy parser is written using FLEX and BISON parser generators, providing for a simple eventual extension of the rules syntax.

3 Advanced control research pilot plant

3.1 Description and operation

Model Predictive Control (MPC) with nonlinear process models (or NMPC) is being applied increasingly often in the chemical industry and many of these applications are viewed as essential for a number of process control problems [17]. However, NMPC solutions may turn rather challenging because of the difficult nature of nonlinear dynamic optimization applied to first principle models, especially in the presence of a mismatch between the model and the actual plant. In fact, experimental validation of control algorithms is an important development step, which commonly carried out using pilot plants for cost and safety reasons. Once the results are deemed satisfactory, the control algorithms are applied on real plants, or the whole system is *scaled up*. In this respect, the experimental system described below has a number of advantages. First, the system is flexible, easily configured and safe. Second, it can handle exothermic reaction mechanisms of arbitrary kinetic complexity (in addition to the zero order reaction considered here), without the additional bother of handling hazardous chemicals. Moreover, as an experimental system (with steam valves, boilers, etc.) it incorporates unmeasured and unmodeled disturbances, model mismatches and many other real world limitations that often cannot be appreciated in a simulation study. Finally, the safety and relative simplicity of this system allows research to focus on the validation of advanced control and estimation algorithms under real world conditions, and to provide detailed case studies that are often not considered in more complex industrial systems. This pilot plant has been the subject of several studies and was used as a benchmark to test control algorithms and other tools [1, 16].

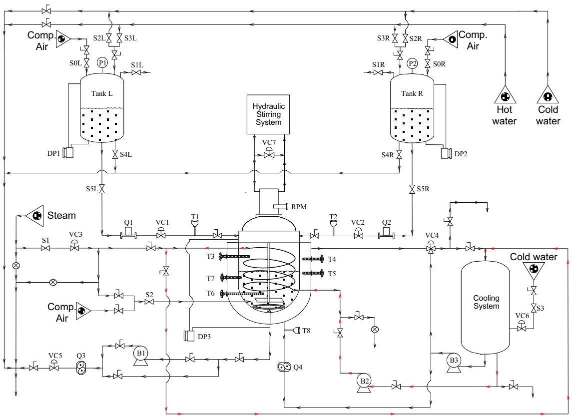


FIGURE 5: *Instalation flowsheet*

As seen in Figure 5, the reactor feed consists of two inlet water streams, with feed rates Q_1 and Q_2 and temperatures T_1 and T_2, while the outlet stream with flowrate Q_3 flows by gravity through valve Vc5. The reactor features a jacket equipped with a spiral baffle and a hydraulic stirring system. The flow rate of the cooling fluid, Q_4, with inlet temperature T_8 and outlet temperature T_4, is determined by the aperture of the valve VC4, λ_4. The speed of the agitator is controlled by manipulating valve VC7. A zero order exothermic chemical reaction is experimentally simulated by injecting steam directly into the liquid inside the reactor. Both the reactor vessel and pipes are insulated to minimize heat losses. The reactor is equipped with standard equipment to measure and manipulate the process variables, as well as with a computer interface for both acquisition and control, whose description follows.

3.2 Data acquisition and control system

The architecture of the old control system incorporated two nodes communicating over a local area network using Sun RPC mechanism: a low profile personal computer especially dedicated for data acquisition and low-level control operations running under MS-DOS and a Sun SparcStation executing supervisory control [1]. As the computational burden grew, the performance of this set-up became unsatisfactory and led to a hardware upgrade of both the control system hardware and of the plant itself.

The electronic hardware of the control system resides in a control cabinet and comprises an industrial computer, 24 VDC and 24 VAC power supplies, relay boards, process variable transmitters, inverters for the variable speed drive of the process pumps, a valve drive board, and safety equipment. The three way backlighted switches installed on the front panel allow to operate discrete devices manually or from the PC.

The rack mount chassis contains a Pentium 120 MHz computer with 32 MB RAM onboard. A CIO-DAS48-PGA board provides 48 analog input channels while an AOP-8 and a DOP-24 boards implement eight analog and twenty four digital outputs, respectively. The RS-232 interface is used for communication with the MicroMaster inverters. Besides, the hardware set-up is to be extended by an I2C interface to custom field devices.

Due to the facts that the distributed control architecture is common process industry practice and that the pilot plant closely resembles an industrial unit, its control system will be implemented as distributed, real-time, and able to work in a heterogeneous environment (Figure 6).

In this set-up, all control system tasks are divided into two major groups: real-time and nonreal-time. Time-critical tasks, such as low level control, fault detection, and watchdogs, are run either as real-time modules or real-time user-space processes thanks to (NEW)LXRT extension of RTAI [13]. Conventional processes, such as graphical user interfaces, loggers, higher level optimizers are run as Linux processes. The control system is concentrated around the state table, a concept adopted from MatPLC [22]. In the presence of high computational load the task set can be partitioned over several nodes with real-time communication among the latter, such as RETHER [21] or RTnet.

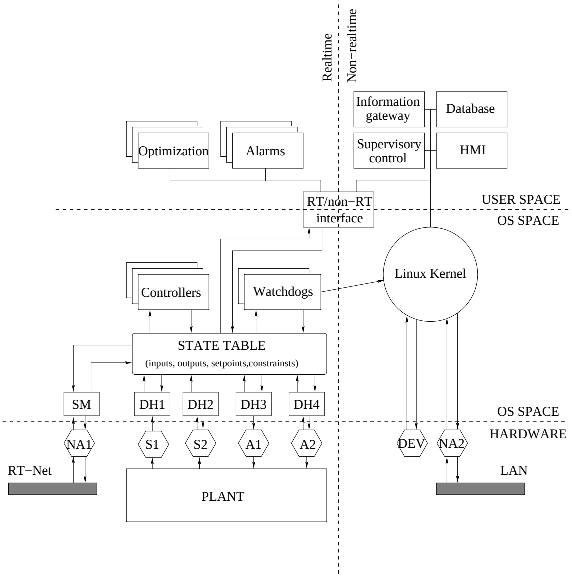


FIGURE 6: *System architecture [15]*

One of the research groups using the pilot plant has developed control software based on NI Labview under Windows. In order to unify the control systems running under various computing system environments the configuration and calibration data are stored in a file accessible from different operating systems.

4 Concluding remarks

Two applications of real-time Linux in the area of chemical process engineering research and education are detailed with the emphasis on the advantages of the real-time and free software paradigms. The authors are of the opinion that the area of chemical process engineering can greatly benefit from a wider use of open source software in the following ways: flexibility and lower costs of training, speed-up of the research pace and technology transfer, higher reliability and adaptability of various software components.

However, open source and free software is still met with considerable reluctance in the industry [10, 4]. In order to stay competetive, industrial units prefer to withhold their know-how, including software. Besides, for many companies it is important to have, albeit often very costly, around the clock software technical support. Warranty and insurance liabilities play an important role in their software related decisions.

Therefore, it is necessary, but not sufficient for real-time Linux, and applications based on it, to be robust and competetive concerning quality only. A significant "promotional" effort is required for its wider exposure to end-users and system integrators. In this respect, educational and research applications are an important contributing factor.

Acknowledgements

The authors are grateful to the Ministry of Science and Technology for financial support under Project PRAXIS 3/3.2/PAPEL/2327/95. Andrey Romanenko is thankful to Program PRAXIS for his scholarship PRAXIS XXI/BD/19609/99. The efforts of open source and free software developers are acknowledged. The authors are thankful to Mr. Miguel Ramos for Figure 5.

References

[1] P. A. F. N. A. Afonso, *Produção Assistida por Computador na Indústria dos Processos Químicos*, PhD thesis, University of Coimbra, 1998.

[2] O. Badmus, D. Grant Fisher, and S. L. Shah, *Real-time, sensor-based computing in the laboratory*, Chem. Eng. Ed., 30 (1996), pp. 280–285,289.

[3] M. Barabanov, *A Linux-based Real-Time Operating System*, Master's thesis, New Mexico Institute of Mining and Technology, 1997.

[4] T. E. Bihari and P. S. Gopinath, *Using real-time Linux in safety critical applications*, in Proceedings of the 2nd Real-Time Linux Workshop, Lake Buena Vista, FL, USA, 2000, The Real-Time Linux Foundation.

[5] W. R. Cluett, *Process control education: an academic perspective*, Pulp Pap. Can., 95 (1994), pp. 57–60.

[6] W. C. Conner Jr., *Incorporation of process control computers in the undergraduate laboratory*, Chem. Eng. Ed., 24 (1990), pp. 106–111,116.

[7] D. R. Coughanowr, *Microprocessor-based Controllers at Drexel University*, Chem. Eng. Ed., 27 (1993), pp. 188–192.

[8] R. A. Davis, *Create virtual unit operations with your data acquisition software*, Chem. Eng. Ed., 29 (1995), pp. 270–274.

[9] A. P. V. Egas, J. P. F. Simão, I. M. M. Costa, S. C. P. Francisco, and J. A. A. M. Castro, *Experimental methodology for heterogeneous studies in pulping of wood*, Industrial & Engineering Chemistry Research, 41 (2002), pp. 2529–2534.

[10] R. J. Jafrate, *The Linux PLC. requirements to gain acceptance in industry*, in Proceedings of the 3rd Real-Time Linux Workshop, Milano, Italy, 2001, The Real-Time Linux Foundation.

[11] B. Lennox and M. Brisk, *Network process control laboratory*, Chem. Eng. Ed., 32 (1998), pp. 314–317.

[12] C. T. Lira, *Computer control of a distillation experiment*, Chem. Eng. Ed., 26 (1992), pp. 38–43.

[13] P. Mantegazza, E. L. Dozio, and S. Papacharalambous, *RTAI:Real Time Application Interface*, Linux Journal, 2000 (2000), pp. 142–148.

[14] A. Romanenko and J. A. A. M. Castro, *An RT-Linux based control system of a pilot plant for reaction kinetics and process control studies*, Computers & Chemical Engineering, 24 (2000), pp. 1063–1068.

[15] A. Romanenko, N. M. C. Oliveira, L. O. Santos, and P. A. F. N. A. Afonso, *A system for chemical process control and supervision based on real-time Linux*, 2002. Submitted to PSE2003.

[16] L. O. Santos, *Multivariable Predictive Control of Chemical Processes*, PhD thesis, Universidade de Coimbra, Coimbra, Portugal, 2001.

[17] L. O. Santos, P. A. F. N. A. Afonso, J. A. A. M. Castro, N. M. C. Oliveira, and L. T. Biegler, *On-line implementation of nonlinear MPC: an experimental case study*, Control Engineering Practice, 9 (2001), pp. 847–857.

[18] M. Schwab, M. Karrenbach, and J. Claerbout, *Making scientific computations reproducible*, Computing in Science & Engineering, 2 (2000), pp. 61–67.

[19] M. Skliar, J. W. Price, and C. A. Tyler, *Experimental projects in teaching process control*, Chem. Eng. Ed., 32 (1998), pp. 254–259.

[20] P. T. Vasudevan, *A comprehensive process control laboratory course*, Chem. Eng. Ed., 27 (1993), pp. 184–187,193.

[21] C. Venkatramani, *The design, implementation and evaluation of RETHER: a real-time ethernet protocol*, PhD thesis, State university of New York, Stony Brook, USA, 1997.

[22] P. Wurmsdobler and M. de Sousa, *Open control systems*, in Proceedings of the 3rd Real-Time Linux Workshop, Milano, Italy, 2001, The Real-Time Linux Foundation.