

**Slovak University of Technology in Bratislava  
Institute of Information Engineering, Automation, and Mathematics**

**PROCEEDINGS**

**of the 18<sup>th</sup> International Conference on Process Control**

**Hotel Titris, Tatranská Lomnica, Slovakia, June 14 – 17, 2011**

**ISBN 978-80-227-3517-9**

<http://www.kirp.chtf.stuba.sk/pc11>

**Editors: M. Fikar and M. Kvasnica**

Kalúz, M., Čirka, L., Fikar, M.: Virtual Laboratory of Process Control, Editors: Fikar, M., Kvasnica, M., In *Proceedings of the 18th International Conference on Process Control*, Tatranská Lomnica, Slovakia, 348–351, 2011.

Full paper online: <http://www.kirp.chtf.stuba.sk/pc11/data/abstracts/056.html>

## Virtual laboratory of process control

M. Kalúz\*, Ľ. Čirka\*, M. Fikar\*

\* *Slovak University of Technology in Bratislava*  
*Faculty of Chemical and Food technology*  
*e-mail: {martin.kaluz, lubos.cirka, miroslav.fikar}@stuba.sk*  
*url: http://www.kirp.chof.stuba.sk*

---

**Abstract:** This paper describes new PID control features, which have been implemented in our virtual Flash laboratory that was originally designed for simulations of technological plants. We discuss methods of discrete PID implementation and show new functions of the virtual laboratory.

*Keywords:* Virtual Laboratory, Process Control, Simulation

---

### 1. INTRODUCTION

Nowadays, in the field of automation and process control it is essential to keep abreast of information technologies. Virtual laboratories became a very popular way how to improve education opportunities in fields of automation, process control, physics, mathematics, and others. These laboratories are usually publicly accessible through Internet and thus can be used by a wider group of students and other interested people. The most commonly used software technologies to create and run online computer simulations are Java, Flash, PHP, ASP, JavaScript, and development environments like Easy Java Simulations (EJS), Virtual Reality Modeling Language (VRML), LabVIEW, and MATLAB connected technologies.

At our department we try to give students practical experience with measurements using real technological devices. But it is not possible to provide them these opportunities anytime, because our capabilities depend on disponibility of real plants at our department. In past, we decided to develop an Internet virtual laboratory to handle this issue.

The virtual laboratories from the different fields of science can be found all over the web. The Virtual Laboratory of Evolutionary Computing, developed by students from FEI STU in Bratislava, provides simulations in field of technical computing, automation, and process control. Most of their simulations are based on MATLAB software with graphical user interfaces built on VRML (Virtual Reality Modeling Language). This virtual laboratory is freely accessible through Internet and can be found on web site <http://ural.elf.stuba.sk/vrlab/>.

Malki and Matarrita (2002) created the virtual laboratory for process control based on LabVIEW environment. They used virtual model software with PID control of second-order plants.

Another interesting project is Virtual engineering/science course, which is a virtual laboratory created by developers from the Department of Chemical Engineering at John Hopkins University, Baltimore, USA. Virtual laboratory

is based on Java technology (Java Applets) and includes models such as a robotic arm, heat conduction in materials, spatial distribution of sound, diffusion process, and some statistical and computing programs. Models are available at Web page <http://www.jhu.edu/~virtlab/virtlab.html>.

Our original virtual Flash laboratory, described in Kalúz (2010) and Čirka et al. (2010), was created using software programming platform Adobe Flash, and it provides virtual simulations of technological plants like liquid storage tank system, tube heat exchanger, and continuous stirred-tank reactor. We think that virtual laboratory including only simulations of system dynamics is not sufficient for actual education needs, so we have decided to improve it with features of process control. Our extensions for simulation applications in virtual laboratory contain feedback closed-loop control with PID controllers. Students and other users can access the virtual laboratory through e-learning system Moodle, located on web page of our department.

Recently, we have also created a virtual laboratory (Kalúz et al., 2010) based on Java Server Pages technology, including computing applications with MATLAB functions and simulations of storage tank system and heat exchangers.

### 2. OVERVIEW OF THE ORIGINAL APPLICATIONS

Our original applications in virtual laboratory course are simulations, based on computation of mathematical models of technological plants. The applications were created as Flash SWF files, published in regular Web page of Moodle course. We created three applications, presenting technological plants most commonly used in education at our department. The first application provides simulation of tank storage system dynamics (Fig. 1). It contains two general mathematical models of plant, the non-linear and linear (obtained by linearization). Each of the models consists optionally of 1-3 ordinary differential equations (ODE), depending on number of tanks in system. The second simulation contains model of a tube heat exchanger (Fig. 2), which is discretized to five segments. Mathematical model consists of five linear ODEs. The last application is simulation of continuous stirred-tank reactor dynamics

(Fig. 3), with mathematical model containing both types of ODEs (linear and non-linear).

So far, all applications are in Slovak language only, but we plan to provide interface with more languages as well.

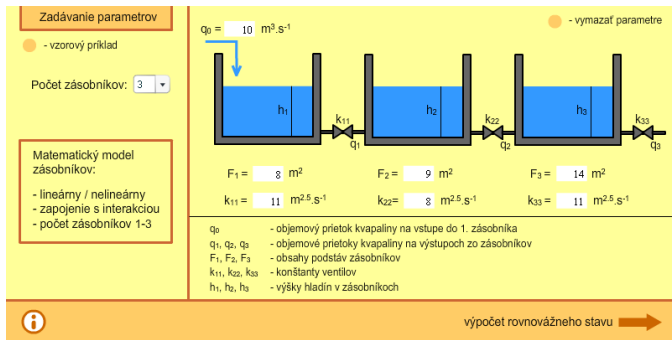


Fig. 1. The first screen of Flash application for liquid storage tank system, showing forms for input parameters, which are used for computation of mathematical model and steady state of system

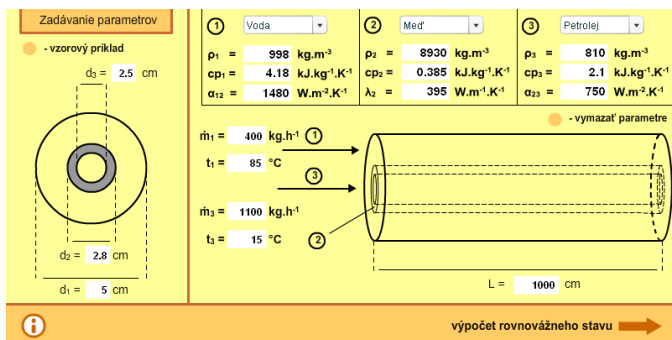


Fig. 2. The first screen of Flash application for tube heat exchanger

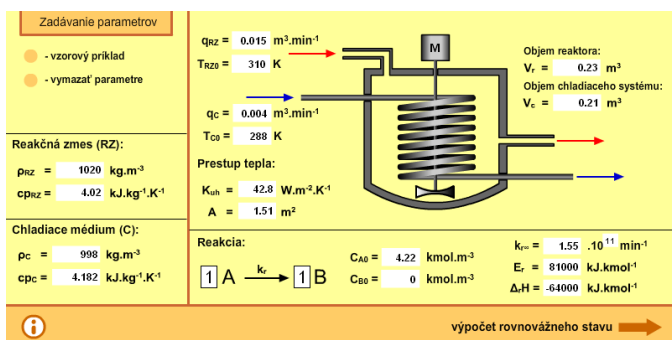


Fig. 3. The first screen of Flash application for continuous stirred-tank reactor

When user starts application, he has to fill input parameters for mathematical model computation. The application includes a script that checks the correct format for all input parameters. After this procedure, application will compute and display steady state of plant. Before dynamics simulation starts, user fills simulation parameters like duration and steps for input variables. In this phase, another script checks the correct format of parameters. Then user pushes the button for executing system simulation. If all input parameters are correctly filled and all necessary setting are done, the application switches to simulation screen. When

the simulation is running, the screen provides observation of plant states (numerically and graphically). After the simulation is finished, the application switches to another screen, where user can collect data in several structure types (text field, XML, MATLAB array).

### 3. MATHEMATICAL IMPLEMENTATION OF PID CONTROLLER

Mathematical model of simulated plant is in form of ordinary differential equations, therefore it must be solved by appropriate numerical method. During the development of original virtual laboratory for system dynamics simulation, we chose 4th order Runge-Kutta method to solve ODE problem. For the PID controller, we implement it in discrete form. To get appropriate discrete controller from a standard expression of the PID, one possibility is to transform derivative and integral components from continuous-time to discrete form (Bobál et al., 2005). The standard form is as follows

$$u(t_k) = K_p \left( e(t_k) + \frac{1}{T_i} \int_0^{t_k} e(\tau) d\tau + T_d \frac{de(t_k)}{dt} \right) \quad (1)$$

The first-order derivative is approximated by backward finite difference

$$\frac{de(t_k)}{dt} = \frac{e(t_k) - e(t_{k-1})}{\Delta t} \quad (2)$$

The integral term is discretized with a sampling time  $\Delta t$  to finite summation

$$\int_0^{t_k} e(\tau) d\tau = \sum_{i=1}^k e(t_i) \Delta t \quad (3)$$

Final expression for implementation of the PID controller (4) is obtained by differentiating  $u(t)$ , using first and second differences.

$$u(t_k) = u(t_{k-1}) + K_p \left[ \left( 1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t} \right) e(t_k) - \left( 1 + \frac{2T_d}{\Delta t} \right) e(t_{k-1}) + \frac{T_d}{\Delta t} e(t_{k-2}) \right] \quad (4)$$

### 4. OVERVIEW OF THE NEW FEATURES

The first screens of applications are the same as in the original ones (Fig. 1, 2, and 3). User has to setup the parameters for mathematical model computation. Mathematical model of steady state for each dynamical system used in applications is defined as a system of algebraic equations.

The steady state observation (Fig. 4) now features two new radio buttons (choice between simulation of dynamics and control). When simulation button is checked, application switches to its original part. To use new features of PID control, user has to check the process control button.

The control setup screen (Fig. 5) defines types of mathematical model (this feature is available only for storage tank system, where user can choose between linear and non-linear model), manipulated and output variable (MV and OV), controller structure (P, PI, or PID), controller parameters, and set points of MV.

When user pushes the button for simulation of process control, the application executed script that runs simple

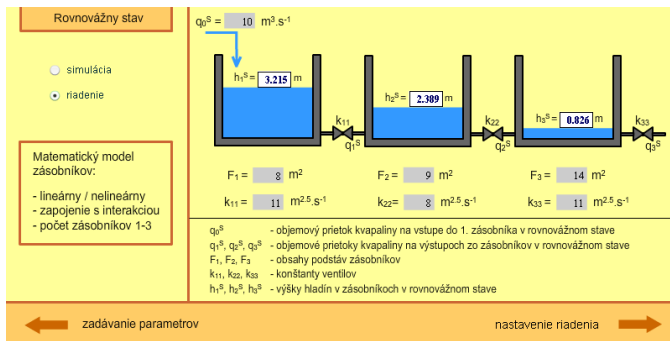


Fig. 4. Screen of Flash application showing steady state of system and choice between simulation and process control

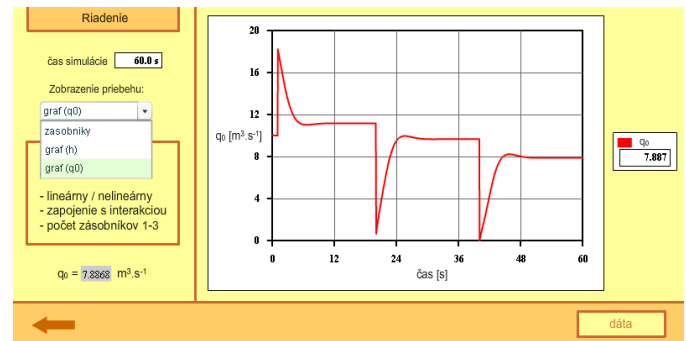


Fig. 7. Application interface, showing graph for MV during simulation of control

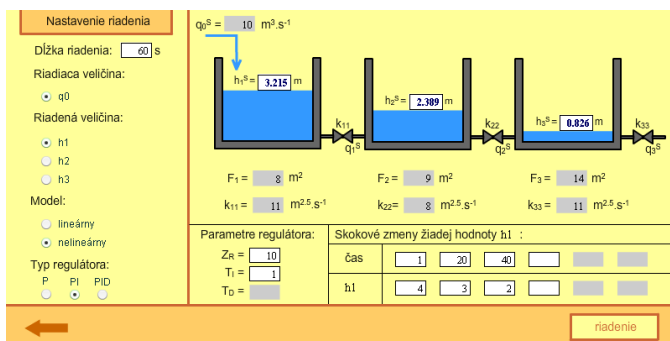


Fig. 5. Setup screen for OV/MV, controller type (P, PI, PID), controller parameters and setpoint step timing

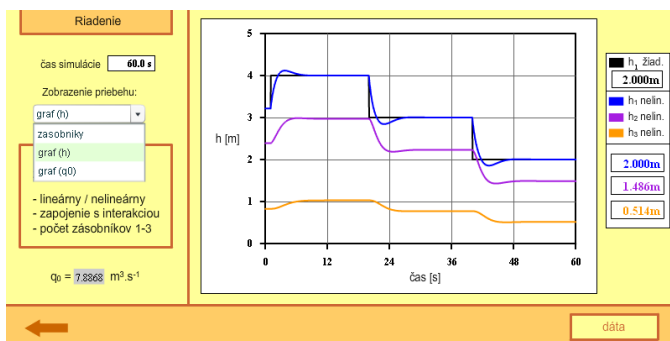


Fig. 6. Application interface, showing graph for levels of liquid in tanks and controlled variable

pre-simulation of control to get information about its behavior. This is important for setting up the graph scaling in simulation screen.

The simulation screen appears if all parameters are correctly filled. User can observe simulation as animated scheme of plant, where application shows only periodically updated values of plant states, or as graphs of MVs (Fig. 6) and OVs (Fig. 7).

After simulation of plant control is finished, the application switches to screen with results (Fig. 8). On this screen, user can collect data from simulation in several structures and formats. Application provides data as plain text field, XML structure, and MATLAB formatted array.

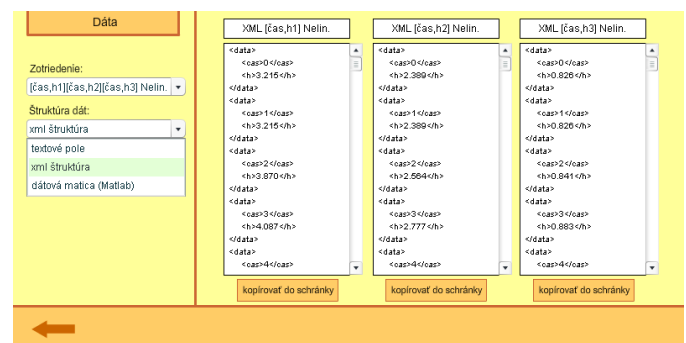


Fig. 8. Results from simulation can be exported in several structures (text field, XML, MATLAB array)

## 5. CONCLUSIONS

We have extended our virtual Flash laboratory with new features for PID control. Users can now either simulate or control process models. This new features have been added to all recently developed simulation applications, containing models of storage tank system, tube heat exchanger, and continuous stirred-tank reactor.

## ACKNOWLEDGMENTS

The authors are pleased to acknowledge the financial support of the Cultural and Educational Grant Agency KEGA of the Slovak Republic under grant No. 3/7245/09 and of the Scientific Grant Agency of the Slovak Republic under the grant 1/0071/09.

The paper is supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

## REFERENCES

Bobál, V., Böhm, J., Fessler, J., and Macháček, J. (2005). *Digital Self-tuning Controllers. Algorithms, Implementation and Applications*. Springer-Verlag, London, U.K.  
 Čirka, L., Kalúz, M., Kvasnica, M., and Fikar, M. (2010). Virtual laboratory. In *Proceedings of the 9th International Scientific - Technical Conference Process Control 2010, C029a – 1–C029a – 8*. University of Pardubice, Kouty nad Desnou, Czech Republic.

- Kalúz, M. (2010). *Virtuálne laboratórium*. Master's thesis, ÚIAM FCHPT STU v Bratislave, Radlinského 9, 812 37 Bratislava.
- Kalúz, M., Čírka, L., and Fikar, M. (2010). MATLAB Builder JA in control engineering education at FCFT STU. In *Technical Computing Bratislava 2010*, volume 18, 1–5.
- Malki, A.H. and Matarrita, A. (2002). Virtual labs for distance education classes. In *Proceedings of the 2002 ASEE Gulf-Southwest Annual Conference*. American Society for Engineering Education, The University of Louisiana, Lafayette.