Embedded Simulations in Real Remote Experiments for ISES e-Laboratory

Michal Gerža[1] František Schauer[1,2] Petr Dostál[1]

[1] Faculty of Applied Informatics, Tomas Bata University in Zlín,
Zlín, Czech Republic, gerza@fai.utb.cz
[2] Faculty of Education, Trnava University in Trnava,
Trnava, Slovak Republic, fschauer@volny.cz

Abstract

The paper focuses on the design of the module of embedded simulations for real remote Internet School Experimental System (ISES) experiments. The ISES experimental platform is intended for educational purpose laboratories at schools and universities providing computer oriented measuring environment for Engineering students and students of Natural sciences. At present, the ISES remote laboratories do not provide any provisions for concurrent interactive simulations in the form of virtual experiments. This drawback results in lesser attractiveness and understanding of real world phenomena. The designed solution uses the Easy JavaScript Simulations (EJS) environment to calculate the data, using equations provided by physics laws, and the ISES module for the simulated data transfer and visualization.

Keywords: RLMS, ISES, measureserver, phenomena simulation, remote laboratory, real experiment

1 Introduction

Traditional teaching methods for students at schools and universities are outdated and not effective enough. Students often expect faster and more understandable teaching methods in the field of physics, chemistry and electro-engineering, which can help them to better perceive real world phenomena. The problem is also an accessibility of the educational materials, especially for distant students who nowadays prefer studying scientific topics using their computers via the Internet. These problematic points are effectively solved by the remote laboratories (RLs) so called e-laboratories. The RLs built on the ISES platform (Lustig, 2009) have been developed for long time since 2002 by the RL Consortium of three universities (Charles University in Prague, Tomas Bata University in Zlín and Trnava University in Trnava) for educational purposes. The ISES is an advanced tool for real-time operation, data acquisition and processing.

The platform is an open system consisting of the ISES components for a hands-on experimentation called ISES WIN. However, there is also an alternative for using remote experiments (REs), the ISES WEB Control Kit. Both types of the experiments are built as the burst (fast) and normal (slow) to offer students a wider spectrum of knowledge. The initial version of ISES RLs has been developed by Charles University in Prague. When the ISES RL became the time-proven educational tool, it was significantly improved to a higher level tool by Tomas Bata University in Zlín in cooperation with Charles University in Prague. They implemented features for the new user environment, EASY REMOTE-ISES, to build REs by laymen (Krbeček *et al*, 2013).

Let us describe the ISES RL concept. It consists of five units as the HW components (signal convertor, control board, physical modules categorized as meters, sensors and devices), Measureserver, Imageserver, Webserver and Webclient. More details and deployed applications are available in (Zeman, 2011; Zeman, 2012; Krbeček *et al*, 2014; Hamid, Modammed, 2010; Drigas *et al*, 2006). All the ISES RLs were integrated into a new system platform called Remote Laboratory Management System (RLMS) REMLABNET accessed on www.remlabnet.eu. It provides units as services to the ISES RL administrators and clients. The schematic arrangement of all the involved autonomous units and defined communication is presented in Figure 1.

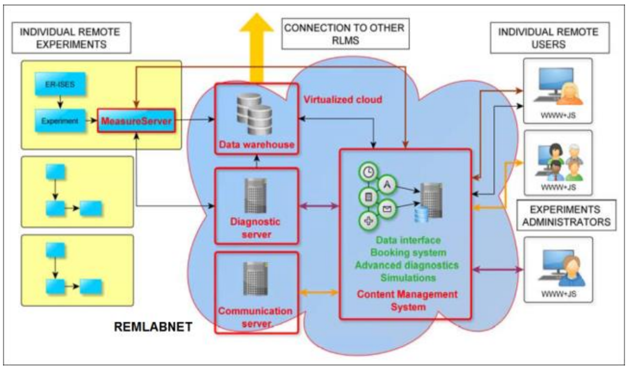


Figure 1. Scheme of the REMLABNET covering ISES remote laboratories with the experiments and clients.

2 State of the art

The ISES RL units internally exploit monitoring, controlling, and communication functions to cooperate with other units to dispatch measured data.

2.1 Physical hardware

There are two concepts in principle implemented for the ISES laboratories - local and remote, built on the same physical HW. The ISES is a modular platform based on three parts. As the first part used, it is the signal convertor installed as the PCI 1202 interface card inside a control computer. Further parts are the control board and the set of sensors for physics, chemistry and electro-engineering. The platform offers a possibility of the data measurement, data visualization and analysis. A complete set of the physical HW is illustrated in Figure 2.

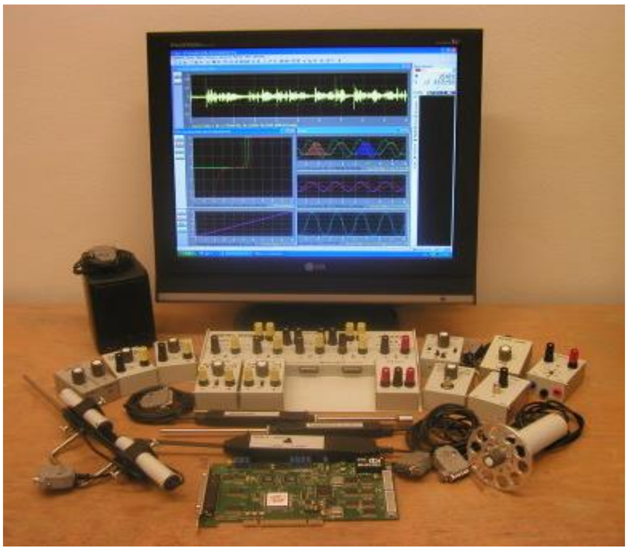


Figure 2. ISES physical hardware including the PCI 1202 interface card and a broad range of the involved meters, sensors and devices.

2.2 Measureserver unit

The Measureserver (MS) unit is a significant software part of the ISES RL concept. It is the processing and communicating server located between the physical HW and remote clients. The MS core is designed as an advanced finite-state machine to setup and process the logical instructions solving prescribed activities. Its functioning is drawn from the concise process script (PSC) file loaded to the MS before its startup.

With respect to the physical HW, the MS in reality communicates with the PCI 1202 interface card. This is the entirely digital process based on the direct reading of data (real values) from particular pins and writing of data to respective pins, which are translated by the signal convertor. These data pins are inputs and outputs located on the control board allowing an access to the particular physical modules like meters, and devices.

Instructions (specific commands), coming from a remote client, are processed by the listening MS. The communication is realized by standard protocols via the Internet. Certain commands go via the MS translator to the REMLABNET where clients can exploit additional services provided like the acquirement of measured data results from previously performed REs stored in the exposed database (warehouse) to analyze phenomena.

All the commands are processed in a deterministic way by two different underlying parsers. The first is called the LR(1) parser that processes commands from the configuration file for the purpose of the graphical user interface settings. This parser is based on static state transition tables (parsing tables), which are able to codify a given language grammar. These parsing tables are parameterized together with a lookahead terminal. This lookahead establishes the maximum tokens, the parser can use to decide, which rule it should use.

The second is the Recursive descent parser (RDP) that processes commands coming from the PSC source to create the defined data structures and logic schemes for the RE. It uses a general form of top-down parsing where backtracking may be involved. The parsing algorithm is based on the walking through a tree.

2.3 Webserver unit

The Webserver unit provides the Nginx services coming into the process when client enters a web page of the ISES RL via the REMLABNET platform. The Nginx is an open source reverse proxy server for HTTP, HTTPS, SMTP, POP3 and IMAP protocols.

2.4 Webclient unit

This unit is a graphical interface provided by web pages via the Internet allowing registered clients to simply control and observe a respective RE (either on the REMLABNET or EU RLMS Go-Lab platforms).

3 Example of real remote experiment and its mathematical model

Simulations play an increasingly important role in the way we teach or do science. This is especially true in education, where computers are being used more and more as a way to make lectures more attractive to students, and to effectively help them achieve a deeper understanding of the subject being taught. This section deals with the design and realization of the phenomena simulations module (PSM) that is able to run concurrently with a respective real RE on the client’s web page. The PSM is a part that does not need to be activated every time during the experimentation.

The PSM is designed as the optional autonomous module integrated into the MS structures in cooperation with the Easy Java Simulations (EJS) core to solve the evolution by a preset numerical method. Mathematical expressions, e.g. constants, variables, algebraic equations and ordinary differential equations (ODE), are defined in the PSC. It is notable to mention, the PSC is a textual source having own syntax of programming language similar to the C language. It is parsed by the RDP inside the MS core during the RE

DOI: 10.3384/ecp17142755

startup. The new objects and additional functions were implemented to identify, initialize and perform the simulation process. Simulated data, produced by the EJS solver, are being continuously transferred on the web page to either charts or tables.

For the purpose of the demonstration how the simulation may be embedded in a RE we use the measurements of the response of the passive parallel RLC circuit to a voltage perturbation in a time domain as shown in Figure 3 showing C capacitor, L inductor (with internal resistance RL) and variable resistors R1D and R2D; ISES voltmeter and amperemeter serve for measuring voltage and current response in the time domain to the perturbation by a unit step voltage, produced by the DC source U(0) and the switch S at the time t = 0. Variable parameters are used two resistive components, artificially introducing the damping. The desired results are all three parameters of the RLC circuit examined. The corresponding initial client's web page with all the preset widgets is illustrated in Figure 6. Let us next describe the problem in a more detailed way.

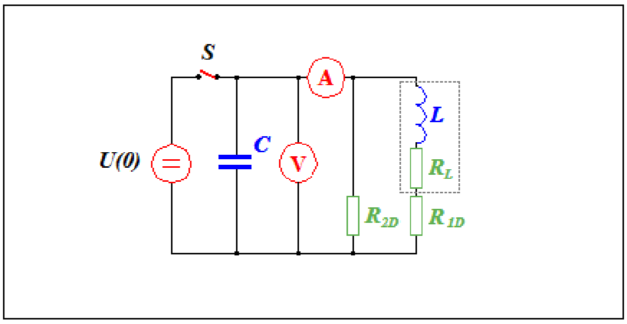


Figure 3. Schematic diagram of the parallel RLC circuit including three resistors.

3.1 Mathematical expressions definition

The mathematical description of the circuit shown in Figure 3 is a form of Kirchhoff's law for a parallel RLC circuit. For the numerical solution of the current *I=I(t)* and voltage *U=U(t)*, both parameters of the circuit - C, L, R_L and damping resistors R_{1D} and R_{2D}, together with the initial conditions should be adjusted and varied. The goal of the using along with the RE is to find the parameters of the circuit by varying both the damping resistors.

There are defined two substitutive resistors induced from Figure 3 to use for further operations.

$$R_1 = R_{1D} + R_L \tag{1}$$

$$R_2 = R_{2D} \tag{2}$$

From Kirchhoff's voltage law (KVL), the following second order differential equation can be constructed

$$\frac{d^2u}{dt^2} + 2b\frac{du}{dt} + \omega_0^2 u = 0 \tag{3}$$

where

$$2b = \left(\frac{1}{R_2 C} + \frac{R_1}{L}\right), [b] = s^{-1}, \tag{4}$$

$$\omega_0^2 = \left(\frac{1}{LC}\right)\left(1 + \frac{R_1}{R_2}\right), \tag{5}$$

$$\omega_1^2 = \omega_0^2 - b^2. \tag{6}$$

The solution, when the essential condition $\omega_0 > b$ is satisfied, follows in the form

$$u(t) = u(0)e^{-bt}\sin(\omega_1 t + \varphi), \tag{7}$$

$$\delta = bT = \ln\frac{u(t)}{u(t+T)} \tag{8}$$

where b is defined as the damping factor and δ is a logarithmic decrement.

Quantity e^{-b} defines how the amplitude relatively makes smaller per unit of time. Equation (7) presents damped oscillations observed in Figure 4, when all the involved elements are set correctly.

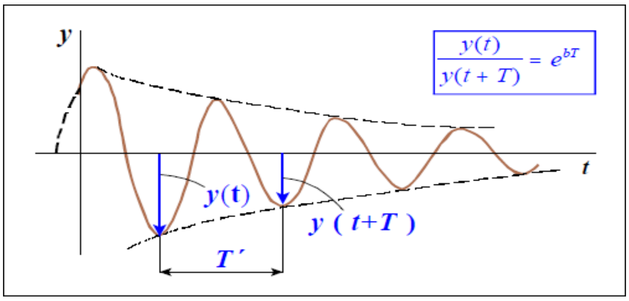
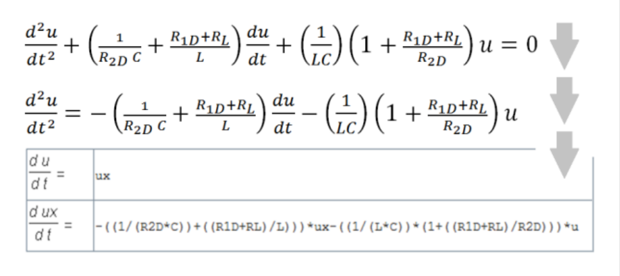


Figure 4. Demonstrative oscillations of the damped parallel RLC circuit.

The simulation process of the parallel RLC circuit with the variable damping is constructed by (3) with the exploitation of (4) and (5), and using original resistors R_{1D} and R_{2D} shown in (1) and (2) as follows.

$$\frac{d^2u}{dt^2} + \left(\frac{1}{R_{2D}C} + \frac{R_{1D}+R_L}{L}\right)\frac{du}{dt} + \left(\frac{1}{LC}\right)\left(1 + \frac{R_{1D}+R_L}{R_{2D}}\right)u = 0 \tag{9}$$

Second order differential equation (9) must be rewritten to a form as shown in Figure 5 to pass it with other parameters to the EJS solver integrated into the PSM. The goal is to obtain and visualize data of the voltage in the given circuit illustrated in Figure 3.

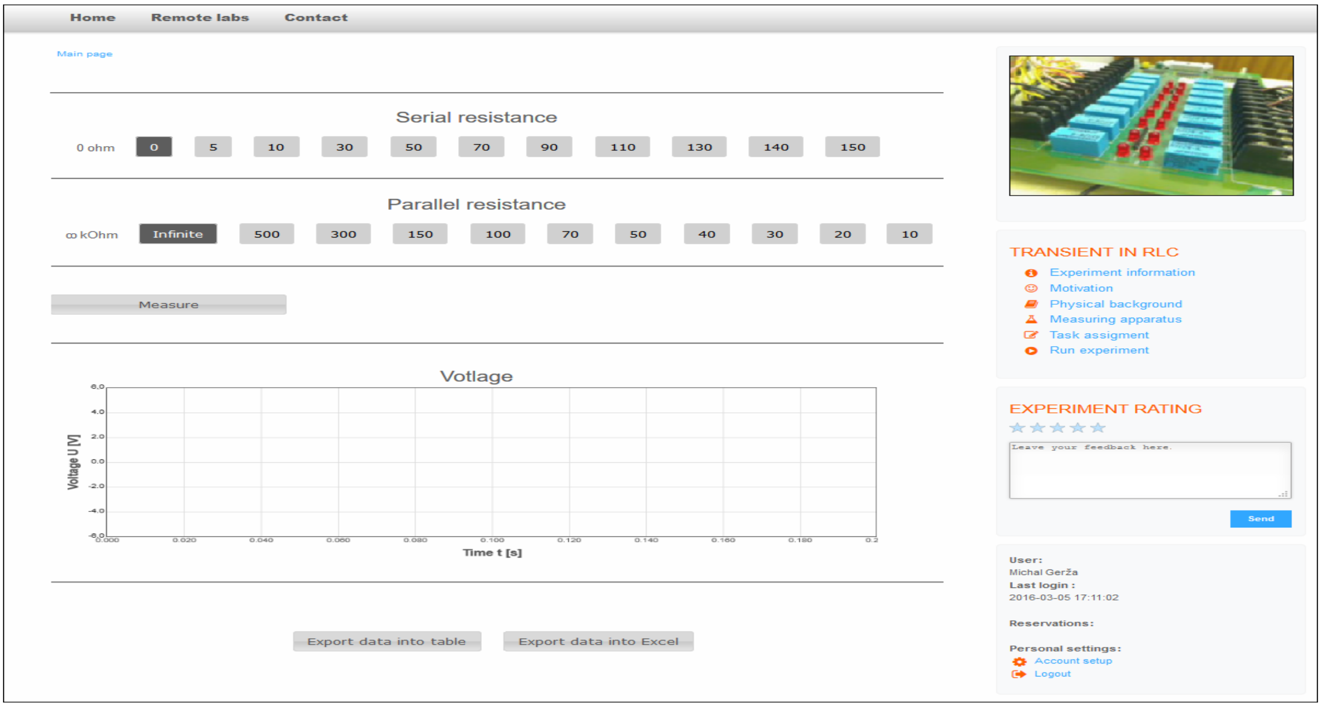


$$\frac{d^2u}{dt^2} + \left(\frac{1}{R_{2D}C} + \frac{R_{1D}+R_L}{L}\right)\frac{du}{dt} + \left(\frac{1}{LC}\right)\left(1 + \frac{R_{1D}+R_L}{R_{2D}}\right)u = 0$$

$$\frac{d^2u}{dt^2} = -\left(\frac{1}{R_{2D}C} + \frac{R_{1D}+R_L}{L}\right)\frac{du}{dt} - \left(\frac{1}{LC}\right)\left(1 + \frac{R_{1D}+R_L}{R_{2D}}\right)u$$

$\frac{du}{dt} =$	ux
$\frac{dux}{dt} =$	-((1/(R2D*C))+((R1D+RL)/L))*ux-((1/(L*C))*(1+((R1D+RL)/R2D)))*u

Figure 5. Rewriting (conversion) of the second order ordinary differential equation to a readable form used by the EJS solver.



Serial resistance: 0 ohm

Parallel resistance: ∞ kOhm

Leave your feedback here.

User:
Michal Gerža
Last login :
2016-03-05 17:11:02

Reservations:

Personal settings:

Figure 6. Web page of the ISES remote experiment “Transients in RLC”; the serial resistance R_{1D} and parallel resistance R_{2D} are allowed to set before running the voltage measurement. Resultant voltage progress is displayed in the chart and it can be also exported into two different data tables.

When all the mathematical expressions are properly defined, the EJS parser then generates a respective XML simulation file. In the next phase, the XML is referenced in the PSC control program file where the control process of the simulation is constructed by new commands to set inputs and outputs, including the solver startup when the initialization is complete.

Finally, the PSC is passed to the MS unit that performs all the commands to create and run the real RE together with the concurrent simulation process of the parallel RLC circuit with the variable damping.

3.2 Simulation process script definition

After the differential equation is rewritten into the readable form for EJS, the next step is saving of the transformed form of the simulation assignment into a respective XML file and its full path reference is then inserted into the PSC control file of the RE.

There are generally two alternatives, how to build in the simulation into PSC file. The first one and more complicated, is its direct inclusion into PSC control file, i.e. to code all the definitions manually. The RE designer should individually decide to which steps to connect the simulation. This is a more complicated alternative, necessitating a good knowledge of the programming language, intended just for programmers.

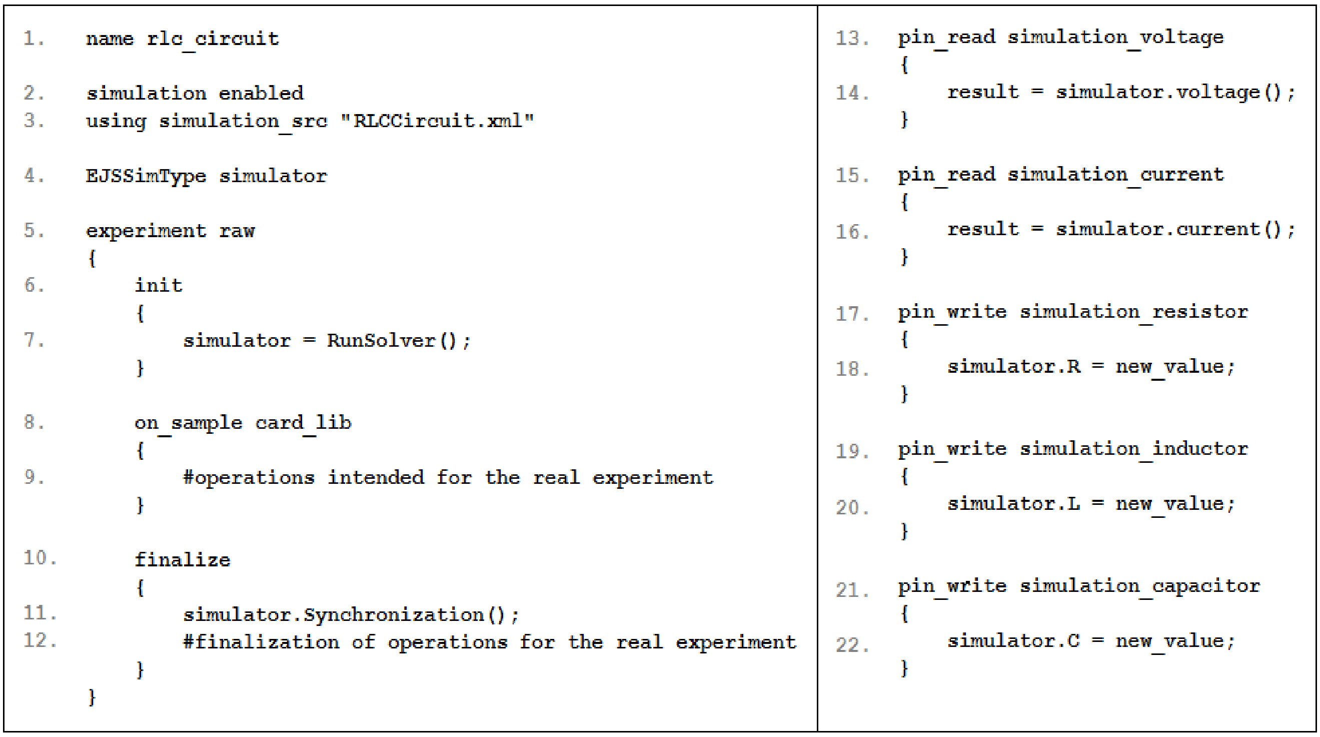
The second alternative, faster and more comfortable, suited to teachers, is the use of EASY REMOTE-ISES providing an intuitive graphical environment for the design and configuration of the RE and its simulation. The PSM providing the phenomena simulation is an optional feature and must be first declared and enabled in the header of the PSC file. An example, illustrating the coded sequence of functions configuring the concurrent simulation process with the respective variable coefficients (for the resistor, inductor and capacitor), is shown in Figure 7.

3.3 Measureserver core process

The MS unit is an important component in the process chain of completion of all calculations to reach the phenomenon simulation and its time synchronization with the real RE running concurrently.

There are five main activities the MS must pursue to provide all inevitable services as listed below:

1. *Parsing:* The RE control program with the simulation process is parsed by the RDP at its startup from the PSC file.
2. *Controlling:* The installed ISES devices are being controlled in specified time interval or scheme.
3. *Monitoring:* The ISES meters and sensors are being monitored in specified time interval.
4. *Archiving:* The measured data and metadata are being archived to the XML and LOG files and dispatched into the warehouse residing in the REMLABNET to fold and analyze.
5. *Simulating:* The PSM calculates an evolution of the respective phenomenon and generates resultant data for the visualization and analysis.



```
name rlc_circuit

simulation enabled
using simulation_src "RLCCircuit.xml"

EJSSimType simulator

experiment raw
{
    init
    {
        simulator = RunSolver();
    }

    on_sample card_lib
    {
        #operations intended for the real experiment
    }

    finalize
    {
        simulator.Synchronization();
        #finalization of operations for the real experiment
    }
}

pin_read simulation_voltage
{
    result = simulator.voltage();
}

pin_read simulation_current
{
    result = simulator.current();
}

pin_write simulation_resistor
{
    simulator.R = new_value;
}

pin_write simulation_inductor
{
    simulator.L = new_value;
}

pin_write simulation_capacitor
{
    simulator.C = new_value;
}
```

Figure 7. Coded sequence of implemented functions of PSM to enable, configure and run the simulation process stored in the PSC file.

The MS activities, mentioned in the five above points, are shown in Figure 8. There are depicted relationships among the PSC control program file, MS unit, ISES physical experiment and REMLABNET

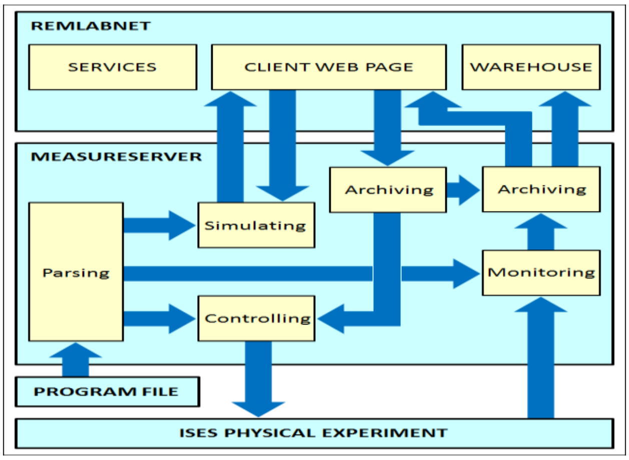


Figure 8. Communication relationships and the activities of the MS unit to realize the real ISES RE and concurrent phenomenon simulation.

3.4 Easy Java Simulations core process

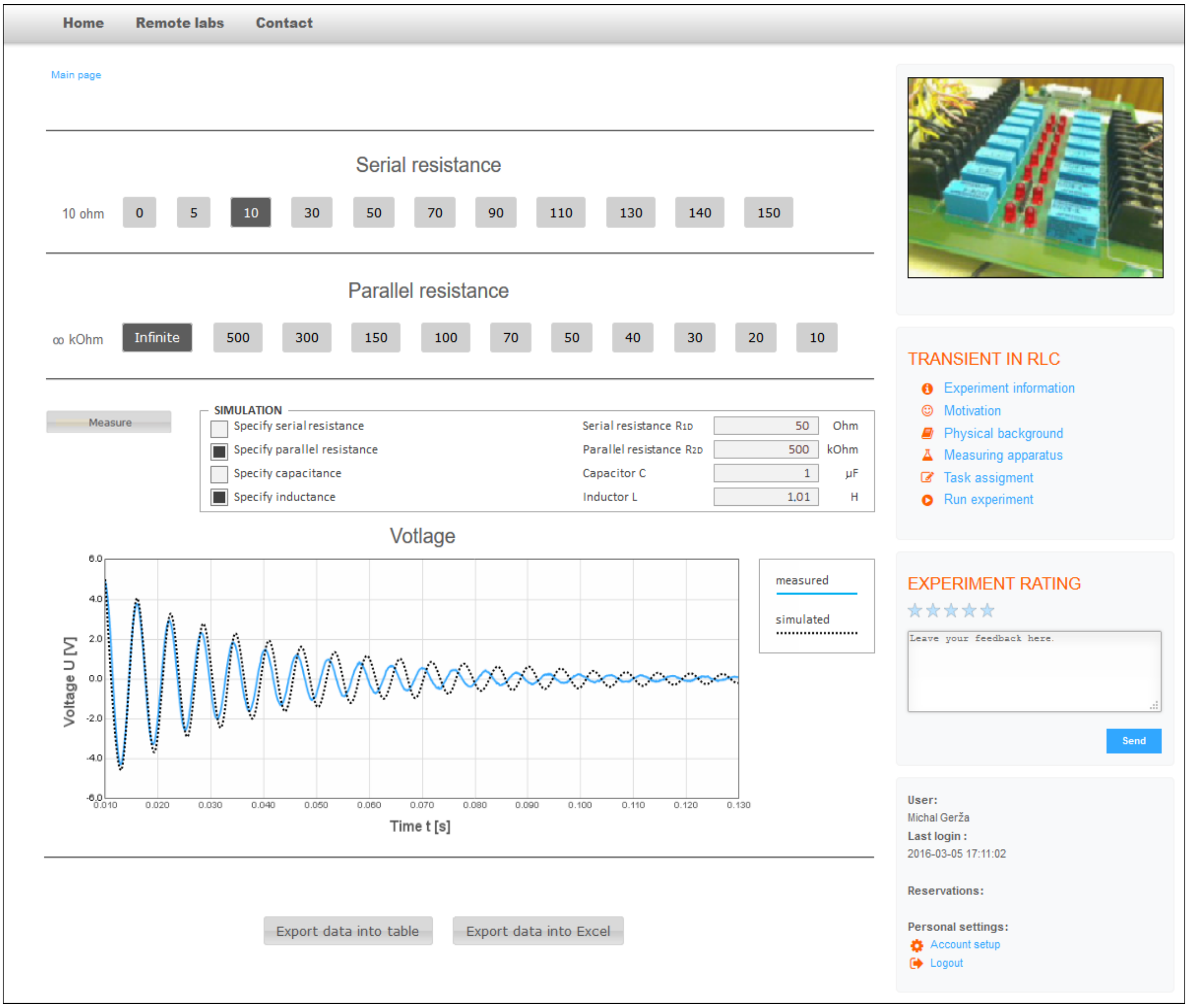
Easy Java Simulations is a software tool designed for the creation of simulations. The EJS is a modeling and authoring tool expressly devoted to science teachers and students. It has been designed to let its user work at a high conceptual level, using a set of simplified tools, and concentrating most of the time on the scientific aspects of the demanded simulation, asking the computer to perform all the other necessary but easily automated tasks.

Nevertheless, the final results, which are generated by the EJS from a user description, can be taken, in terms of efficiency and sophistication, as the implementation of programmer (Esquembre, 2015).

Since the EJS is Open Source under a GNU GPL license, we decided to exploit its optimized solver to perform phenomena simulations integrated into the PSM as a component part of the MS unit.

The EJS solver computes a numerical approximation to the solution of an ODE or, more precisely, of an initial value problem. That is, given the system of ODEs and the state of the system at a given time. The solver is able to approximate the solution of the ODE in a future time. Solver algorithms are all one-step methods, mostly based on Runge-Kutta (RK) schemes, can be explicit or implicit, fixed-step or adaptive, and all use an interpolation to provide a dense output. It means these algorithms produce solution points at any instant of time.

The PSM mostly uses two EJS solvers only. The first solver is the Runge-Kutta 4. It refers to the classical RK method which started it all. It is a fixed step, 4th-order algorithm that works well in most situations. This solver interpolates using one step of the bootstrapping algorithm applied to the Hermite interpolation (this gives order 4 interpolation). The second one is the Cash-Karp 5(4) that is an adaptive 4th order method. It is based on two embedded RK schemes of order 5 and 4. We use a local extrapolation



SIMULATION

□ Specify serial resistance	Serial resistance R1D	50	Ohm
■ Specify parallel resistance	Parallel resistance R2D	500	kOhm
□ Specify capacitance	Capacitor C	1	μF
■ Specify inductance	Inductor L	1,01	H

User:
Michal Gerža
Last login :
2016-03-05 17:11:02

Reservations:

Personal settings:

Figure 9. Web page of the ISES remote experiment “Transients in RLC” running together with the simulation process represented by the dotted black curve interlaced with a real measurement of the voltage in the parallel circuit.

and accept the 5th order approximation as the solution. This is the default solver for ODEs because it provides excellent performance in most situations. The solver interpolates using two steps of the bootstrapping algorithm applied to the Hermite interpolation (this gives order 5 interpolation). The interpolation scheme is very convenient because it significantly optimizes the number of evaluations of the ODE rate. The computational load is largely determined only by the tolerances and not by the reading step (Gonze, 2013). These solvers generate data, which are passed to the PSM output interface.

3.5 Concurrent data visualization

When the simulated data are received from the EJS solver to the PSM, the MS unit performs the synchronization with the concurrent real RE. Finally, the measured and simulated data are sent to the client’s web page to the chart. Differences should be obvious, that is when we compare interlaced curves of the real and calculated voltage in the circuit. The web page visualizing both voltage representations is shown in Figure 9. This example simulation process allows setting of the damping resistors R_{1D} and R_{2D} both separately and by the buttons too for serial and parallel resistances. The optional values of the capacitance and inductance can be entered to modify the process. The real curve indicates R_{1D} = 10Ω, R_{2D} = Infinity and L = 1H, whereas the simulated curve indicates R_{1D} = 50Ω, R_{2D} = 500KΩ and L = 1.01H to observe obvious differences between the physical experiment and its defined mathematical model.

4 Conclusions

The paper introduced a new module designed for the phenomena simulation. Its advantage is perceivable in a creation of the simulation running concurrently with a real experiment in the ISES remote laboratory.

This module was integrated into the Measureserver unit. It allows clients its activation when the simulation process is demanded to complement the ISES remote experiment for the purpose of providing an alternative to observe differences between the mathematical and physical model of a studied phenomenon.

The simulation module should notably help students to improve their learning procedure concerning a better understanding of a given taught subject.

We formulated the following conclusions.

- The experimenting provided by the ISES remote laboratory is a new method of teaching and learning in comparison with traditional forms.
- The Measureserver unit is a core part used for measurement, data processing and communicating among clients and the ISES physical modules.
- The phenomena simulations module is a feature providing means to realize simulation process running together with the real experiment.

Acknowledgement

This work was supported by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2. 1.00/03.0089 and also by the Internal Grant Agency of Tomas Bata University in Zlín under the project No. IGA/FAI/2016/017.

References

A. S. Drigas, J. Vrettaros, L. G. Koukianakis, and J. G. Glentzes. 2006. *A Virtual Lab and E-Learning System for Renewable Energy Sources*. WSEAS Transactions on Computers 5, pp. 337-41. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-33645137921&partnerID=tZOtx3y1

Francisco Esquembre. *Easy Java/Javascript Simulations* [online]. Universidad de Murcia, Murcia, Spain: Ejs Wiki, 2015 [cit. 2016-03-17]. Available: http://www.um.es/fem/EjsWiki/Main/HomePage

Didier Gonze. *Numerical methods for Ordinary Differential Equations*. In: Teaching activities [online]. Belgium: Université Libre de Bruxelles, 2013 [cit. 2016-06-11]. Available: http://homepages.ulb.ac.be/~dgonze/TEACHING/numerics.pdf

R. Hamid and S. A. Modammed. 2010. *Remote Access Laboratory System for Material Technology Laboratory Work*. In International Conference on Engineering Education, pp. 311-16. Available: http://www.scopus.com/inward/record.url?eid=2-s2.079958730131&partnerID=tZOtx3y1

M. Krbeček, F. Schauer, and F. Lustig, *Easy Remote ISES - Development Environment Remote Experiments*, Innovations 2013. USA, 2013, pages 81-100. Available: http://www.ises.info/ oldsite/clanky_pdf/Easy_Remote_ISES_2013.pdf

M. Krbeček, F. Schauer, and K. Vlček. *Communication Requirements of Laboratory Management System*. In: Latest Trends on Systems: Proceedings of the 18th International Conference on Systems, 2014, pp. 686-691. ISBN 978-1-61804-244-6.

F. Lustig. *Internet School Experimental System ISES* [online]. Prague, Czech Republic, 2009 [cit. 2014-06-05]. Available: http://www.ises.info/index.php/en/systemises

P. Zeman. *Software environment for control of remote experiments*. Ostrava: VŠB-Technical University of Ostrava, Czech Republic, 2011.

P. Zeman. *Software environment for integration of measured data from remote laboratory and simulation*. Ostrava. VŠB-Technical University of Ostrava, Czech Republic, 2012.

DOI: 10.3384/ecp17142755