

The Simulation Toolbox for Distance Learning

Antonio Magdić, student

antonio.magdic@fsb.hr

Tihomir Žilić, B. Eng.

tzilic@fsb.hr

Prof. dr. sc. Mario Essert

messert@fsb.hr

Doc. dr. sc. Željko Šitum

zsitum@fsb.hr

Mario Stanković, student

mario.stankovic@fsb.hr

Faculty of Mechanical Engineering and Naval Architecture, I. Lučića 1, HR-10000 Zagreb

Abstract. The distance educational tools have been developing with the fact that the topics for simulations can demand special approach. The only thing required to use the full power of it is a Web browser which supports frames and JavaScript. Interface is straight-forward and simple, which facilitates interaction between a professor and student. Using teaching experience the simulation toolbox will offer easy learning of simulations with the approach by example. Two approaches of simulation learning are available: the first one per which a student has to write for the partly set assignment one part of the program code and the second one per which student analyzes and learns by modification of parameters (input data) the obtained simulation results.

Keywords: distance learning, simulations, control toolbox

1. Introduction

“Educational Web System” is the system made by students of the Faculty of Mechanical Engineering and Naval Architecture [1], [3]. It consists of different program modules (Scriptrunner, WEB-exams, Sketcher etc.) for students’ education and distance learning. Common System requires freely available, open-source software (Apache, PHP, MySQL) and runs on both Windows and Linux operating systems.

Scriptrunner (see Fig. 1) supports writing and execution of programs written in standard programming languages: C/C++, Pascal, Fortran, Java, JavaScript and PHP which are all executed using open source compilers.

It also supports programs for a variety of professional mathematic tools, both open source (Octave and Scilab) and commercial (Matlab and WRI Mathematica). You can run a program written in source code in your browser by selecting an appropriate compiler and clicking on a 'Run' button.

Since Scriptrunner is a Web application built on client/server technology it is surprisingly easy to use - all you need to have is a Web browser. Every user has

learns in his/her own work environment, with all settings, and the files created during on-line session remain on the server for the next session. There is also a 'Public Files' folder where users can publish their files for others to see and execute.

The primary purpose of Scriptrunner is in the field of distance learning [5], especially for courses in which the supported programming languages are used. It can also be used for creating the databases of computer programs, collecting interactive on-line textbooks, tutorials etc. A very interesting possibility provided by Scriptrunner is writing and publishing of digital books. That possibility spreads the usage of Scriptrunner to a much wider range than program running and working in program languages.

The screenshot shows the Scriptrunner web interface. On the left is a file manager with a tree view containing folders like 'My Files', 'Public Files', 'HELP', 'Interactive books', 'Settings', and 'Logout'. The 'Interactive books' folder is expanded to show 'Modeling an Inverted Pendulum', which is further expanded to show '1. Problem setup and design requirements'. The main content area displays the text of this problem, which describes an inverted pendulum on a cart with an impulse force F . Below the text is a diagram of the cart and pendulum system, labeled 'Picture 1. - 1'. The diagram shows a red cart of mass M on wheels, with a blue pendulum of mass m and length l attached. The cart is on a horizontal surface with a coordinate x . The pendulum makes an angle θ with the vertical. Below the diagram, it says 'For this example, let's assume that M mass of the cart 0.5 kg'. At the bottom, there is a table of files and a file upload form.

File name	Size	Type	Date
test.tex	0.89 KB	.tex	02-11-2004 07:46:30
test.txt	2.67 KB	.txt	24-01-2005 19:07:08
latav nrdv.tex	1.58 KB	.tex	11-02-2005 14:51:12
2.nb	62.46 KB	.nb	21-11-2005 11:05:52

Figure 1: Scriptrunner environment

Moreover, each such book becomes 'live' because the programs with the preferred, regularly different, input data can be called from it.

Teaching simulations, especially by examples, is today much bigger problem than teaching classical topics and theories. So, we have developed such distance educational toolbox (Scriptrunners' plug-ins) which consists of two modules: programming (and running) simulations and interpreting (and comparing) results.

2. Programming simulations

The control system simulations are usually based on solving the system's differential equations and plotting the observed time behavior. Numerical methods for integrating ordinary differential equation are used from Matlab program.

Student calls an on-line assignment through Scriptrunner, he studies the problem and writes his Matlab code for the problem solution. A part of the code may be already written together with an assignment (as seen on the example [4] on Fig. 2). Student cannot change the teacher's code. It can be changed only by the teacher who wrote it.

The teacher has a blank sample by which he may create as many assignments of the same type as needed, but from the different field of curriculum. The common sample (Javascript program) includes the following items:

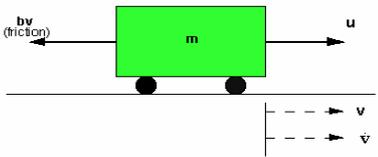
1. Assignment's title
2. Assignment's description with a picture (figure)
3. The set program code which shall not be changed
4. A detailed description of the required solutions
5. Text area for the entry of a student code
6. Result field.

One should bear in mind that the teacher may simply create (generate) from one assignment pattern several different assignments, so that every student obtains sufficiently different assignment, and consequently gets different results. In this way, the possibility of on-line results' copying is eliminated. Student shall send the written assignment to the professor over communication plug-ins.

Professor shall evaluate every assignment and the students on-line may check the obtained mark.

Tema: Regulacija brzine vozila

Zadan je sustav prema shemi. Inercija kotača vozila se zanemaruje, a zadane veličine su:
 m - masa vozila = 1000 kg, b - trenje = 50 N/s i u - sila motora = 500 N.



$m\dot{v} + bv = u$
 $y = v$

Korištenjem Newton's zakona, jednadžbe gibanja su

Za zadani Matlab kod:

```

m=1000;
b=50;
u=500;

```

Rezultati:

```

Transfer function:
1
-----
1000 s + 50

```

Napišite prijenosnu funkciju i jednadžbe prostora stanja, te pokrenite simulaciju otvorenog upravljačkog sustava. Programski kod upišite ispod. Usporedite rezultate:

```

% prijenosna funkcija
bro = [1];
naz = [-b/m];
vozilo_1 = tf(bro,naz)
step(u* vozilo_1)

% prostor stanja
A = [-b/m];
B = [1/m];
C = [1];
D = 0;
vozilo_2 = ss(A,B,C,D);
step(u* vozilo_2)

```

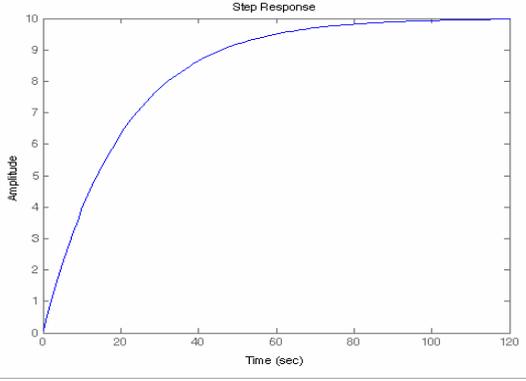


Figure 2: Example of programming simulations

3. Interpreting and comparing results

This way of learning does not demand programming. By changing parameters of simulation model, students will learn to interpret the behavior of the observed system. Fig. 3 shows the type of module that uses topic from automatic control theory [2]. That topic consists of the mathematical model of robot arm with two links and two rotational degrees of freedom. Behind the graphical environment lies very complicated nonlinear mathematical model of regulation which students do not need to know exactly. They need to know how to set controller parameters, choose which controllers to compare. They run simulation by clicking on the button "Pokreni". Given results

will be shown on eight graphs. Each of them shows the properties of specific controller in the moving of the robot's arm, i.e. how good will specific controller follow the desired trajectory of robot's arm. Fig. 3 also shows that the module has links named "help" in the corners of some groups of parameters. Clicking on it, student will get help with detailed explanation of usage (see Fig. 4).

The concept of learning will go in next way:

1. Setting parameters
2. Running simulation
3. Interpreting results
4. Comparing results
5. Changing specific parameters and repeating steps 1-4.

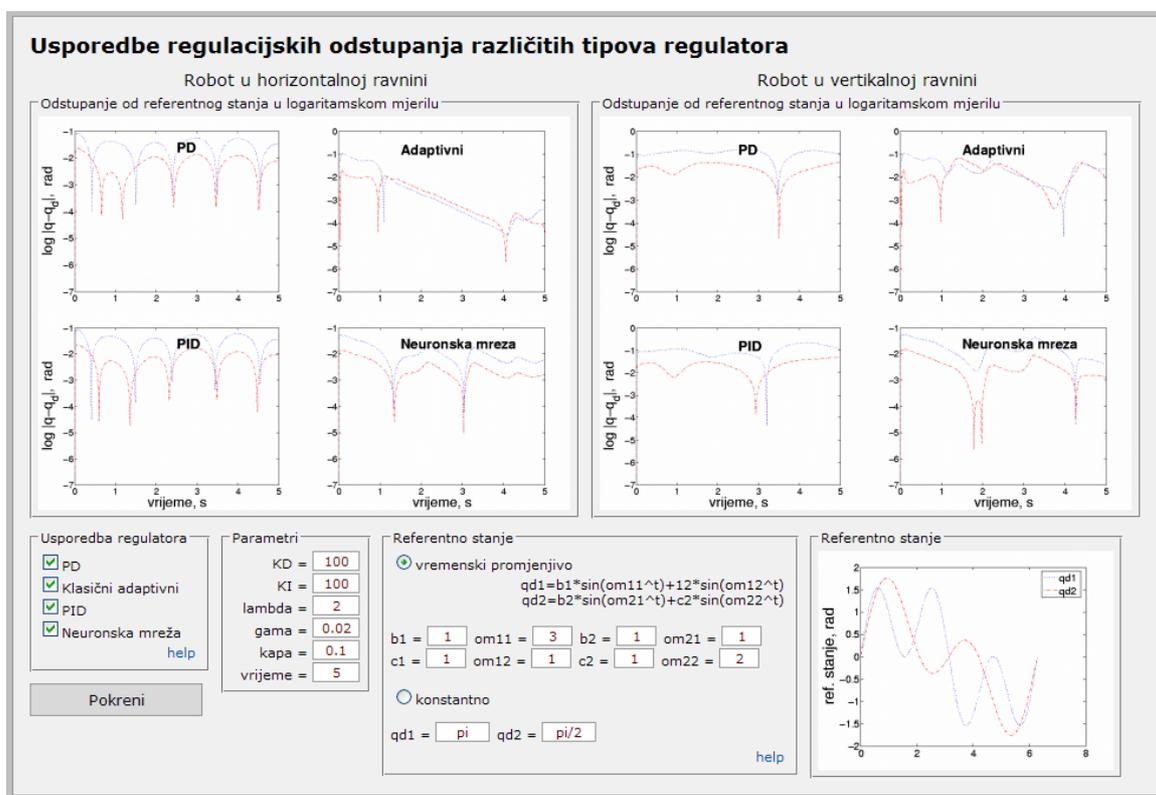


Figure 3: Graphical environment for interpreting simulations

Interesting and useful part of learning is comparing the results. To do this, students have to have simulation results one next to another.

This problem is solved in this module, because graphs are placed exactly one next to another. All of them are visible simultaneously. There are also possibilities to get them larger, if some details are needed. Double click on any one of the eight graphs will open separate window with enlarged graph, shown by Fig. 5.

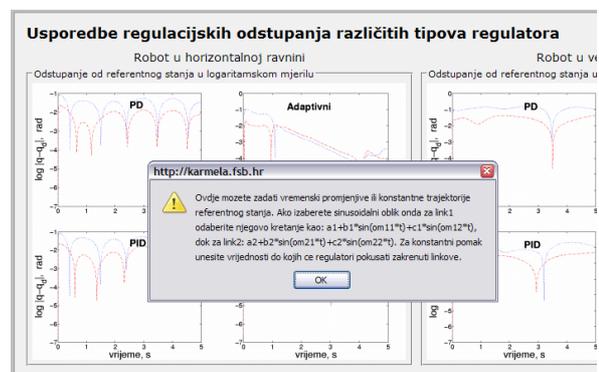


Figure 4: Getting help

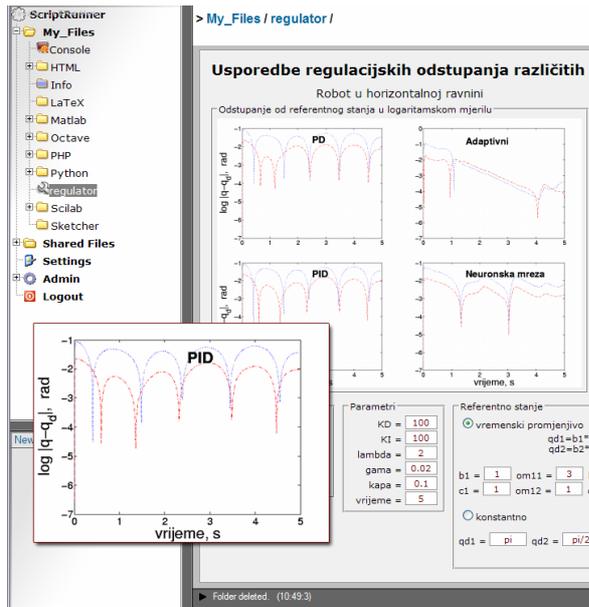


Figure 5: Increasing graph size

4. Conclusion

As Scriptrunner is a new program (although we have been developing it since 2000), only two books with it have been written by now (for PMF and FSB, Zagreb). However, the interest for it is ever growing, not only in, but also out of Croatia (the conference in Lugano [1], Switzerland and presentation at FernUniversität in Hagen support this standpoint) and there is an option for using Scriptrunner at several universities.

A possibility to include toolboxes in it gives us an excellent way of teaching. This work shows one toolbox for usage in automatic control theory, that is

still in the phase of developing. To teach students to simulate mathematical models is much easier on this way, because of independency of applications that are local, not distance. Another advantage of this approach is that we have made and will make such toolboxes that are used especially for education. Simulation, as specific topic for education needs to be treated on this way. We give the students one way of learning, currently as the best solution we think.

5. References

- [1] *Essert, Mario; Čengija, Davor*; Internet learning math, Barcelona, 3eem 2000;
- [2] *Kasać J., Novaković B., Majetić D., Brezak D.*; Global Positioning of Robot Manipulators with Mixed Revolute and Prismatic Joints. IEEE Trans. on Autom. Control, 2006.
- [3] *Mauser, Bojan; Essert, Mario*; Scriptrunner3, Proceedings of the 26th International Conference on Information Technology Interfaces / Cavtat, Dubrovnik 2003;
- [4] *Šitum, Željko; Petrić, Joško*; Modeling and Control of Servopneumatic Drive. Strojarstvo. 43 (2001) , 1-3; 29-39 (članak, znanstveni rad).
- [5] *Žilić, Tihomir; Magdić, Antonio; Essert, Mario*; Publishing HTML and LaTeX on-line documents, Proceedings of the 16th International Conference on Information and Intelligent Systems / Aurer, Boris; Bača, Miroslav (ur.). Varaždin, 2005;