

Online Design of Matlab/Simulink Block Schemes

[doi:10.3991/ijet.v6iS1.1614](https://doi.org/10.3991/ijet.v6iS1.1614)

Z. Janík and K. Žáková

Slovak University of Technology, Bratislava, Slovakia

Abstract—The paper presents a new online tool that enables to build a Matlab/Simulink block scheme in the Internet environment. The block scheme can be designed in similar manner as it is offered by local installation of Simulink. The application was created by widely used technologies as XHTML, CSS, JavaScript, PHP together with AJAX approach. The created application can be used as a supporting tool in virtual and remote laboratories.

Index Terms—computer aided engineering, control design, online services, student experiments

I. INTRODUCTION

The expansion of Internet brings new opportunities to almost all areas of everyday life. The education is not an exception. The importance of virtual and remote laboratories becomes more and more significant mainly in the area of technical education, like Engineering. They help students to understand basic problems because they illustrate and visualize the controlled dynamics, and the necessity to exercise all design steps starting with the plant identification and ending with the evaluation of the control results achieved with a particular model. The advantage is that such laboratories are available to all interested users 24 hours a day and from any place. Students and interested users can access them via commonly used Web browser on any machine that has an Internet connection.

In spite of the fact that virtual and remote laboratories can be set in a number of various areas, this paper is oriented to the area of automation and control where the existence of online laboratories is very welcome. This paper presents a new tool that supports building of simulations for online laboratories accomplished in Matlab/Simulink environment. For running such simulations it is necessary to build a block scheme corresponding to a control of virtual or remote device. To offer students a possibility to understand all aspects of the control problem, they have to be involved in its solution. Nowadays, it is not sufficient only to start and to follow the running experiment. Students need to interact with it. This means not only to run a pre-prepared simulation but also to influence a control process. It can be done by changing the control or simulation parameters or even by changing the whole control algorithm.

Most of Internet based solutions use the first alternative. The modification of parameters can be done very easily using a standard web form. The advantage is that if all inputs are restricted to numerical values from a predefined interval such solution doesn't bring any potential danger for developed application and the server installation. On the other hand student can use only control algorithms that

are available on the web page and therefore he or she is limited in implementation of own ideas.

The change of experiment parameters can also be done by uploading a file but it is not very often used because of the necessity to verify the format of the uploaded file. This task is very crucial because inside of the file it is possible to write not only numerical values of variables but also commands that can corrupt the whole consistency of the application. It is very advantageous if the uploaded file can be in native format of the used software environment that doesn't allow including any danger code. For example in the case of Matlab software the uploaded file can use the numerical file format with the extension .mat.

Internet based control applications that want to offer students better understanding of problem usually enable to modify not only parameters but also a control algorithm. The easiest way is to offer users several algorithms or control structures and student can choose one of them that is according to him or her the most suitable. Another possibility is to enable user to implement own solution. It can be again done by several ways.

Firstly, we can again use the standard web form where user can edit a text code with a control algorithm. The control structure can also be uploaded in a separate text file. It is also possible to enter control algorithm by parsing the text file that includes a control algorithm. In this case, the modification of the file relates only to those lines that are really changed. Headers and all other specifications in the file are not touched. However, all these possibilities bring potential danger for server installation. As we have already mentioned before, the editing of free text enables to include commands (e.g. format, ...) that can damage not only running application but also installation of the whole server.

Better solution is to upload the file that includes a control structure in the form of block scheme where only "safe" blocks are allowed. Such block schemes can be prepared e.g. in Simulink for Matlab or SciCos or Xcos for SciLab. They are saved in specific format of the given software. The safety of this solution can be solved by eliminating blocks that enables to include commands. Of course, there exist also other solutions but they are beyond the scope of this paper.

The presented tool offers an alternative way to these solutions. It enables to build a control scheme in a comfortable way via web interface by simple dragging and connecting graphical blocks. Such approach is typical, for example, in Matlab/Simulink or in Scilab/Scicos. The advantage consists in the fact that using this solution the change of control algorithm can be done very easily by

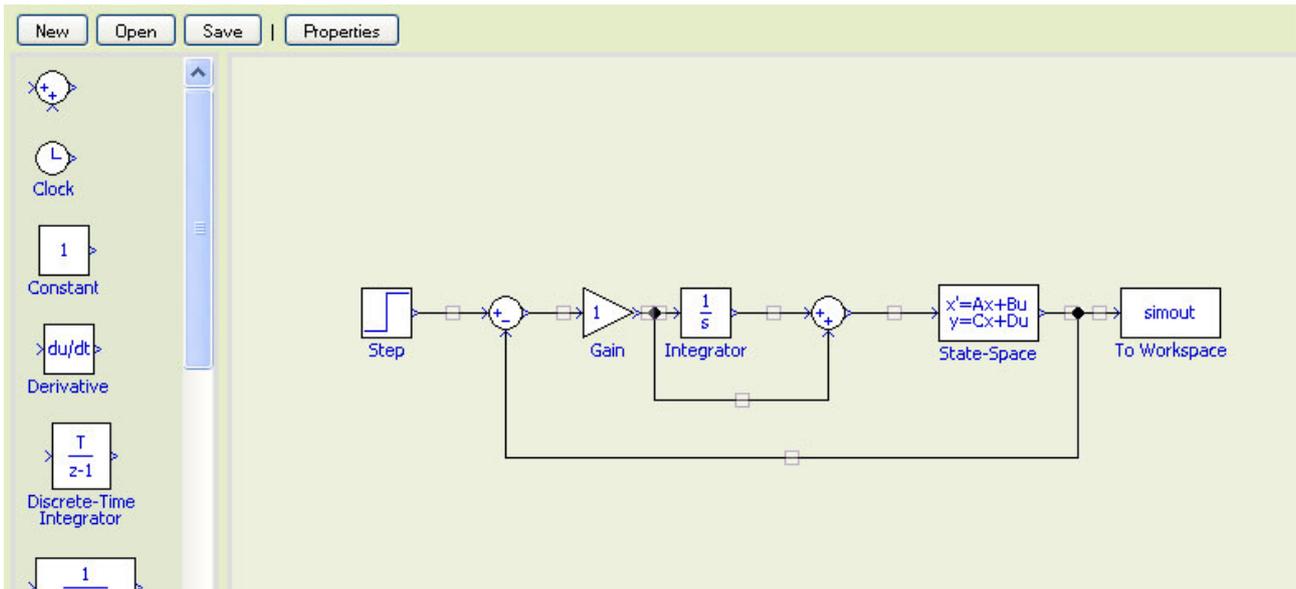


Figure 1. Block scheme building

simple modification of blocks scheme, i.e. by changing, deleting and adding blocks. In this paper the created application will be described in more details.

II. APPLICATION DESCRIPTION

Realization of the online tool requires creating a front end (client part) and a back end (server part) of the application. In our case both parts use the same configuration file that contains attributes and parameters of each block. Back end scripts perform export and import tasks triggered from the editor.

The client part of the whole application is shown in Figure 1. It basically consists of two parts: the toolbar with the list of all available blocks, and the canvas with current scheme. Block toolbar is automatically filled up with blocks that are present in the configuration file. When the editor starts up, the canvas is blank and it is ready for insertion of new blocks from the block toolbar.

In the application we considered 2 types of blocks:

- standard blocks are blocks from the toolbar panel that are used for modeling of dynamical behavior of systems;
- “pseudo” blocks are blocks that are generated automatically by the application for establishing an additional connection feature. In this way we recognize the *node block* that is used for multiple line connection and the *line block* that enables to modify the position of line segment placed between two standard blocks.

The block scheme with control structure is built by standard drag and drop method. The blocks attributes can be changed in the pop-up window that is displayed after clicking on any standard block in the canvas (see e.g. Figure 2). The content of the attributes window is dynamically changed and it is loaded via AJAX request from other script. The script builds the form (i.e. all window items) from attribute settings included in the configuration file whereby the form contains pre-filled values of current block attributes. The configuration file also defines the type of each attribute input that can be considered as text field, select box, and checkbox or radio button group.

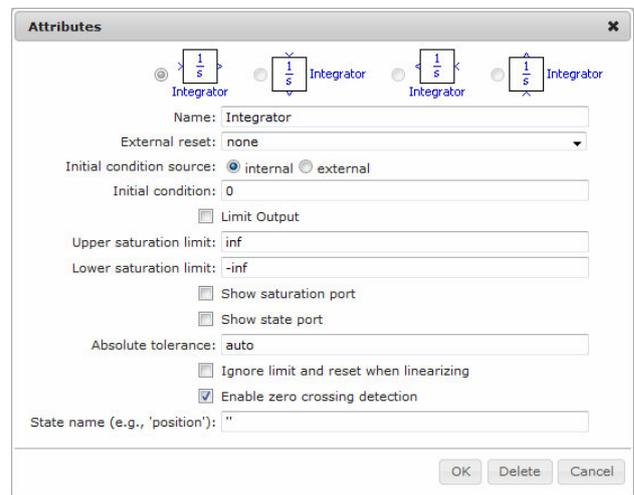


Figure 2. Attribute window for an integrator block

The configuration file is stored in XML format. This file contains all information about attributes that are used for displaying blocks, storing block parameters and converting scheme to Matlab/Simulink format.

In the present time the application obtains only basic Simulink blocks. However, the application is easily extendable by editing the XML file. It is not necessary to edit source codes of the application. New blocks can be easily added by modifying the XML configuration file. Provided DTD document enables to prevent errors in the modified XML structure.

III. TECHNOLOGY OVERVIEW

The created tool is based on XHTML, CSS¹ and JavaScript technology. The export and import is performed by PHP technology mainly through AJAX² requests. These technologies are selected in order to keep the widest availability of this tool. There are no special demands on the client's side. The only thing that is re-

¹ Cascading Style Sheets

² Asynchronous JavaScript and XML

quired is a compatible web browser (the application was tested mainly in FireFox 3.5 and Internet Explorer 8, but it should work with no limitations in IE 6.0+, FF 2+, Safari 3.0+, Opera 9.0+, Chrome).

JavaScript library jQuery is used to speed up the development and simplify the source code. The library is used for the object manipulation, changing CSS properties, visual functions and AJAX requests.

IV. DATA EXCHANGE WITH MATLAB

The main aim of the application is to offer the suitable data format for block schemes that can be simulated in Matlab/Simulink environment. It means that we have to export the created block scheme into the native Matlab format (mdl file). Since it is the readable ASCII text file it can be done by respecting the mdl file structure. The exported file can be later used in local or online simulations.

The only problem is that the structure of Simulink files is changing together with the Matlab software. In the present time the developed online tool is able to cooperate with Simulink version 7.0. For other Simulink versions, it is necessary to build separate configuration files to match structure of each specific version.

A. Export

The Export function requires preparing the block scheme and related data to the suitable format for transfer to the back-end server. We decided to use JSON³ as an exchange format. The main reason is that this format is easy to handle and since it is very compact, it takes lower demands on the network traffic.

The whole JSON object containing the scheme from the editor is sent to back-end server via AJAX request. The back-end script transforms the scheme to Simulink format. This transformation includes listing of blocks together with their location, rotation, parameter values and connections between blocks.

The transformed scheme is saved to a file that can be later used in the installed version of Simulink.

B. Import

The Import function enables to process a file containing the Simulink scheme. The uploaded file is parsed and transformed to the format suitable for the online graphical editor. After the successful transformation, the scheme is loaded into the web editor and it is available and ready for user to use it.

V. CONCLUSIONS

The introduced application can be used for online control of dynamical systems. It can be used both for virtual and remote laboratories. Its main advantage consists in the fact that the control structure and also the algorithm for the control of online experiment can be easily changed by drag and drop technology in the graphical window of the application. The accomplished change can be immediately considered in the realized experiment. Since the online tool is developed by wide spread technologies it can be combined with the most of solutions that enable to use Matlab via Internet. It brings a comfortable tool for students in all forms and areas of education.

REFERENCES

- [1] S. Holzner, *Mistrovství v AJAXu*, Computer Press. Brno, 2007 (in Slovak).
- [2] Z. Janík, "Internet based modification of Matlab/Simulink block schemes", *Diploma thesis*, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Bratislava, Slovakia, 2010 (in Slovak).
- [3] Z. Janík, K. Žáková, "Online Design of Matlab/Simulink Block Schemes", *10th International Conference Virtual University*, Bratislava, Slovak Republic, December 2009.
- [4] J. Resig and the jQuery Team. 2009. jQuery Documentation. <http://docs.jquery.com>
- [5] M.T. Restivo, J. Mendes, A.M. Lopes, C.M. Silva, F. Chouzal, A Remote Lab in Engineering Measurement, *IEEE Trans. on Industrial Electronics*, vol. 56, no.12, pp. 4436-4843, 2009.
- [6] F. Schauer, M. Ožvoldová, F. Lustig, „Real Remote Physics Experiments across Internet – Inherent Part of Integrated E-Learning”, *Int. Journal of Online Engineering (iJOE)*, 4, No 2, 2008.
- [7] Chr. Schmid, „Internet - basiertes Lernen“, *Automatisierungstechnik*, 51, No. 11, p. 485-493, 2003.
- [8] M. Šimuněk, P., Bisták, M. Huba, „Virtual Laboratory for Control of Real Systems”, *Conference Proceedings ICETA*, Košice, Slovakia, 2005.
- [9] A. J. Turgeon, "Implications of Web-Based Technology for Engaging Students in a Learning Society", *Journal of Public Service and Outreach* 2(2): 32-37.
- [10] K. Žáková, M. Sedlák, "Remote Control of Experiments via Matlab", *Int. Journal of Online Engineering (iJOE)*, 2, No. 3, 2006.
- [11] Refsnes Data. 1999-2009. JavaScript and HTML DOM Reference. <http://www.w3schools.com/jsref/default.asp>
- [12] The PHP Group. 2001-2009. PHP Documentation. <http://www.php.net/docs.php>

AUTHORS

Z. Janík and **K. Žáková** are with the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia (e-mail: zoltan.janik@stuba.sk, katarina.zakova@stuba.sk).

The work has been partially supported by Slovak Grant Agency, Grant KEGA No. 3/7245/09. It was also 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. This support is very gratefully acknowledged.

This article was modified from a presentation and paper at The 10th International Virtual University Conference, in 10-11 December 2009, Bratislava, Slovak Republic.

Received March 25th 2011. Published as resubmitted by the authors April 20th, 2011.

³ JavaScript Object Notation