Web Application for LTI Systems Analysis

Frantisek Gazdos, and Jiri Facuna

Tomas Bata University in Zlin, Faculty of Applied Informatics
Nam. T.G. Masaryka 5555, 760 01 Zlin, Czech Republic
gazdos@fai.utb.cz

Abstract. The number of web-applications which employ external computing tools to perform complex calculations on-line is rising. This contribution presents one such application – web-interface for analysis of linear time-invariant (LTI) systems. It is based on the interconnection of the MATLAB system with the latest web technologies and developer tools. It enables to analyze a single input – single output (SISO) system in a user-defined form and a user-friendly way on-line without the necessity to install the MATLAB environment or similar computing tools. The paper explains motivation for development of this site and gives also detailed description of the whole process including the Web – MATLAB interconnection. The results are presented using selected screenshots of the site on a simple example of a LTI system analysis.

Keywords: Web-application, MATLAB, LTI systems, analysis.

1 Introduction

Providing and sharing information using the Internet is a standard nowadays. Simple text information was soon supplemented with visual, audio and video data, forming user-friendly environment for the information acquisition. Last decade has brought interactivity into the web-pages, providing users with tools to better control the process of information transfer. This opened the door to the development of applications which enable to process user-entered input-data and display corresponding results. As the requirements for complex calculations on the Web grow, there are rising efforts to connect web-applications with effective computing tools such as The MathWorks's MATLAB [1-2], Wolfram Research's Mathematica [3-4], Maplesoft's Maple [5-6] and others which are more suitable for complex calculations and corresponding data processing. This paper presents one such web-application for LTI systems analysis which can be used widely by students, teachers or scientists interested in control engineering or similar areas where linear time-invariant systems are used for description of various processes. Users can choose different forms of data-input, e.g. transfer function or state-space description, either continuous or discrete-time with standard notation of the parameters as common in the MATLAB environment. They can do various transformations, such as continuous to discrete-time or state-space to transfer function conversions, and vice versa. Finally they obtain important information about the system, e.g. stability, gain, poles and zeros or selected characteristics, such as a

adfa, p. 1, 2011.

step-response or nyquist plot. The presented application was designed in a bilingual (English/Czech) version within the scope of the Master's thesis [7] and further tuned by the authors. A similar application can be found at [8], however, with limited possibilities and only in the Czech language.

This paper is structured as follows: after this introductory part, main goals and motivation are outlined in the next section, followed by the employed software, hardware and Internet tools. Next part is devoted to the process of connecting the MATLAB system with web-applications and a brief description of the implemented functions for the LTI systems analysis follows. Further, the web-application user interface is presented on a simple example and described in detail, followed by some concluding remarks and perspective at the end of the paper.

2 Motivation and Goals

Nowadays, the academic standard for systems analysis and synthesis is the MATLAB environment [9] with its powerful computing possibilities and a great number of toolboxes for systems analysis, synthesis or modeling and simulation. However this system is not free and consequently not every student, teacher or scientist can afford it. Mainly students have problems of this type and often seek other possibilities how to use their favorite functions for free. The Octave [10-11] or Scilab [12-13] software can be the alternatives, however they are not as capable as the MATLAB system and have different notation. Other problem is that generally not every computer in the university network is equipped with the MATLAB software, and, often happens that you need to test something outside of the university network – at home, halls of residence, etc. These problems have led to the idea of the development of a simple, user-friendly web-application for basic analysis of widely used LTI systems. The site is primarily intended for students' purposes, however teachers or scientist can also gain advantages of using it from every place connected to the Internet.

For the web-application development, the main goals were formulated as follows: to develop a web-application which enables to analyze a user-specified linear time-invariant SISO system with the following requirements:

- Bilingual (English/Czech) version;
- Similar notation as in the MATLAB environment (e.g. for polynomials);
- Possibility to work either with continuous or discrete-time models;
- Possibility to work with both transfer function and state-space description;
- Possibility to do various transformations (e.g. continuous ↔ discrete-time, state-space ↔ transfer function);
- Provide important information about the system, such as poles, zeros, gain and stability;
- Display selected characteristics, such as various responses (step/impulse), bode or nyquist graphs.

Besides this, the web-application has to be user-friendly enough and provide help information when needed.

For the systems analysis, standard functions of the MATLAB SW are employed, mainly from the popular Control System Toolbox [14]. More information about the used functions are provided further in this contribution.

The designed web-application is intended to support pedagogical and research activities and one of the main advantages can be seen in the open access to the application without the need to install any kind of software on a user computer or to have a particular kind of operating systems. Apart from this, it can be administered and updated easily. The application is accessible directly at the URL: http://matserver.utb.cz/LTI [15].

3 Methodology

This section further explains used SW and HW tools, the process of MATLAB ↔ Web interconnection and main implemented functions for the LTI systems analysis.

3.1 Software and Hardware Tools

Web-applications are generally operating on web-servers – computers with a suitable operating system and a special application, web service enabling operation of the web-applications. The developed application runs on the PC Fujitsu Siemens Esprimo P5625 with the AMD Athlon 64 X2 5600+ processor with 4GB of RAM and two 500 GB hard-drives in the RAID 1 configuration. It hosts the Microsoft Windows Web Server operating system with the web service IIS.

Web-applications have usually several tiers and most common is the 3-tiered architecture with a presentation, application and storage tier.

In the developed application, the first, presentation tier accessible by a web browser was created using the HTML, CSS and AJAX technologies.

The middle application tier (logic tier) processing requests and data from the presentation tier and generating user-friendly interface dynamically was designed using the ASP.NET technology, a part of the Microsoft .NET Framework. For algorithms implementation, C# programming language and the Microsoft Visual Web Developer Express Edition software was used. Complex computations needed for the LTI system analysis were realized with the help of the MATLAB system. The following MATLAB components were employed for the required functions, corresponding data-processing and deployment to the Web:

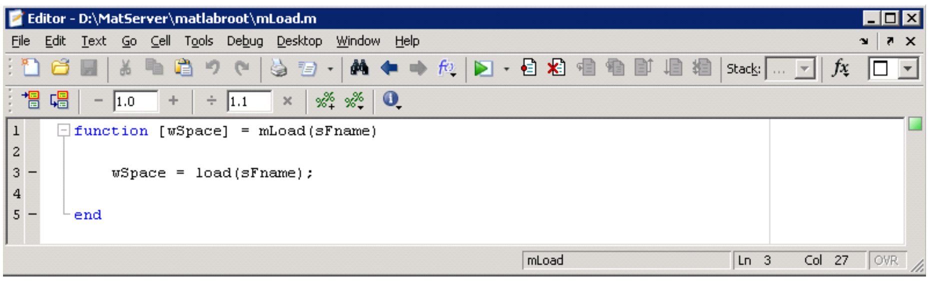
- MATLAB;
- Control System Toolbox;
- MATLAB Builder NE [for .NET];
- MATLAB Compiler.

The last, third web-application tier is the storage one, also known as the data tier, which is accessed through the middle application tier and enables data-retrieving, storage and update. In the developed application, it is represented by the file-system of the server operating system.

3.2 MATLAB ↔ Web Interconnection

Up to the MATLAB version R2006b, there was a possibility to connect HTML web-pages with the MATLAB functions simply using the MATLAB Web Server component, where the functions were implemented directly in the form of MATLAB source codes (m-files). Next versions of the MATLAB system do not include nor support the Web Server component and the connection has to be realized in a different way. MATLAB functions can be implemented (deployed) in the form of so-called components – dynamic-link libraries (DLL-files) in the Microsoft Windows operating system. Development of a web-application connected to the MATLAB system can then be divided into the two independent parts: preparation of the source m-files with required MATLAB functions and components generation in the first step, and development of the web-application and implementation (deployment) of the generated components (DLL's) in the second step. Web-applications connected to the MATLAB system can be developed using various technologies such as Microsoft .NET Framework or JAVA. In this work the former tool was used.

An example of a source MATLAB m-file used for a web-application component is presented in Fig. 1. This function simply loads all the variables from a MATLAB binary data file (*.mat).


Editor - D:\MatServer\matlabroot\mLoad.m
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack:
1.0
1.1
mLoad
Ln 3
Col 27
OVR

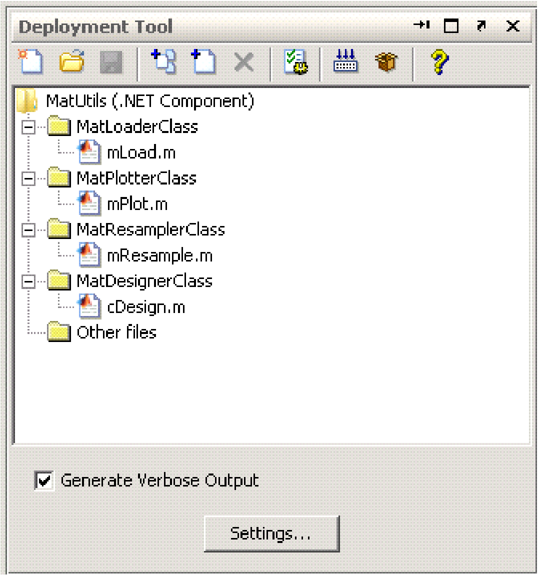
```
function [wSpace] = mLoad(sFname)

    wSpace = load(sFname);

end
```

Fig. 1. Deployed m-file example

The resultant components for the web-application are then created from the m-files (and relevant toolboxes) using the MATLAB Compiler toolbox. This toolbox uses other supporting toolboxes for the compilation, depending on the chosen technology of web-application development. For the applications developed in the Microsoft .NET environment, the MATLAB Builder NE [for .NET] is needed which creates .NET components containing Microsoft .NET Framework classes. An example of a .NET component with the source m-file above is displayed in Fig. 2.

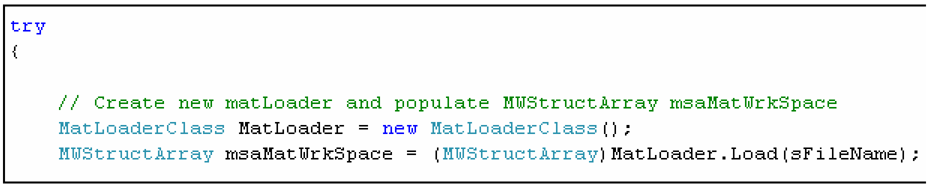


```
MatUtils (.NET Component)
    MatLoaderClass
        mLoad.m
    MatPlotterClass
        mPlot.m
    MatResamplerClass
        mResample.m
    MatDesignerClass
        cDesign.m
    Other files
```

☑ Generate Verbose Output

Fig. 2. Compiler toolbox

Components (DLL's files) created using the MATLAB Builder NE can be further used in the standard common way as other components of the Microsoft .NET Framework technology. It is only necessary to make a reference to the components in the web-application development environment. Then it is possible to use all the classes and functions from the source MATLAB m-files during the application development. An example of calling the function Load of the MatLoader object from the MatUtils component is given in Fig. 3. The component MatUtils.dll containing the MatLoaderClass was created from the m-file load.m, see Fig. 1, using the MATLAB Builder NE toolbox.



```
try
{

    // Create new matLoader and populate MWStructArray msaMatWrkSpace
    MatLoaderClass MatLoader = new MatLoaderClass();
    MWStructArray msaMatWrkSpace = (MWStructArray)MatLoader.Load(sFileName);
```

Fig. 3. Calling a function from an m-file

In web-application development environment, e.g. Microsoft Visual Studio, a reference is given to the MatUtils.dll file (see Fig. 4.) and then the class MatLoaderClass can be further used in the standard way as displayed in Fig. 3.

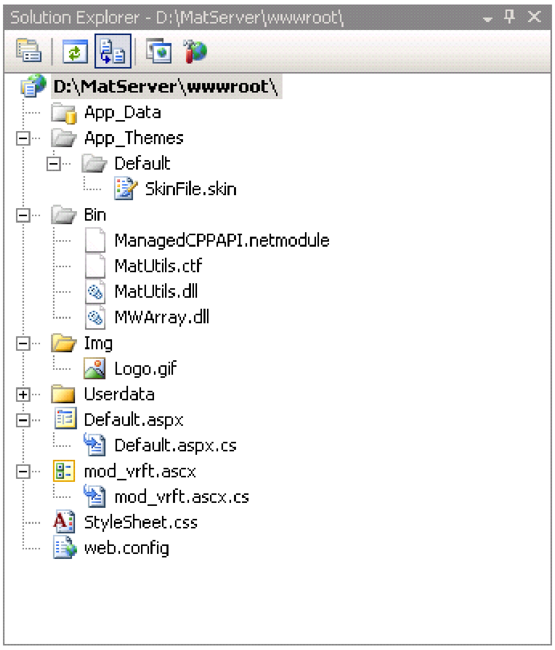


Fig. 4. Reference to a .NET component

3.3 Main Implemented Functions

All the used MATLAB functions (more than 40 programmed m-files) were divided into 4 main categories: *transfer* (for various system transformations), *test* (to test some system properties), *properties* (to determine certain system properties) and *graphs* (to generate chosen characteristics). In the *transfer* class, these main MATLAB functions were utilized:

- *ss*, *ssdata* for the state-space description and transformations;
- *tf*, *tfdata* for the transfer function description and conversions;
- *d2c*, *c2d*, *d2d* for the continuous ↔ discrete-time conversions and resampling.

The *test* class included implementation of these MATLAB functions:

- *size* to test dimensions (number of inputs/outputs) of the system;
- *isproper* to test if the system is proper.

The *properties* class was built on these standard MATLAB functions:

- *minreal* for the minimal realization of the system;
- *zero*, *pole* for zeros and poles of the system;
- *order* for the system order/degree;
- *isstable* to test stability;

- *ctrb*, *obsv* to test controllability and observability;
- *dcgain* for the system gain information.

The last class *graphs* contains implementation of these MATLAB functions:

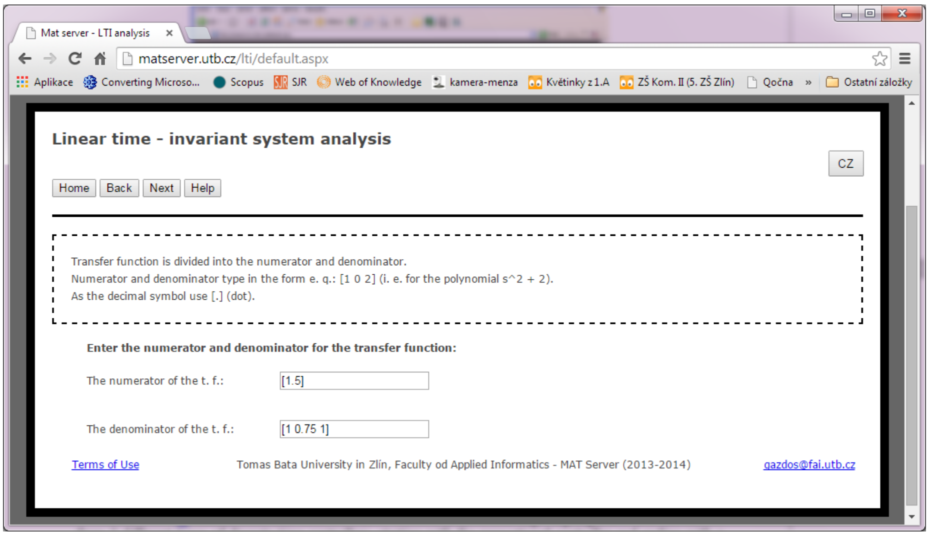
- *step* for the step-response;
- *impulse* for the impulse response;
- *nyquist* for the nyquist graph;
- *bode* for the bode graph;
- *pzmap* for the pole-zero map of the dynamic system.

Detailed documentation for the above given standard functions of the Control Sytem Toolbox, part of the MATLAB system can be found in the MathWorks documentation, e.g. [14].

4 Results

A testing version of the developed application is accessible via the Internet at the following URL: http://matserver.utb.cz/LTI [15]. The application user interface (presentation tier) is designed as a guide and allows entering the required information for a given LTI system in several interconnected steps. Common control buttons such as “Next” / “Back” and “Help” are used to navigate throughout the application and provide corresponding help information when needed. The interface consists of these main parts:

- Start-up screen which provides basic information about the web-application, its possible usage and contact information; a language control button (Czech / English) and also the help button are accessible throughout the whole process of work with this site;
- In the next step, a user can choose the form of the system input – either as a transfer function or in the form of state-space description. For discrete-time systems, it is possible to give a sampling period. For both continuous and discrete-time systems also a time-delay can be easily entered;
- Further, coefficients of the transfer function or matrices of the state-space description can be entered in the same form as in the MATLAB environment, as illustrated in Fig. 5;
- The fourth step can be used for various transformations (continuous ↔ discrete-time, transfer function ↔ state-space description) and to find the so-called minimal realization of the system; default setting is no transformation. The entered system is also tested here – if it is SISO and if it is proper (if no, the user is warned and he/she can enter a new system). In case of a discrete system, the application also offers to resample it with a new sampling period;
- In the next screen, it is possible to choose which system properties should be determined (zeros, poles, order, stability, gain, controllability and observability);



Linear time - invariant system analysis

Transfer function is divided into the numerator and denominator.
Numerator and denominator type in the form e. q.: [1 0 2] (i. e. for the polynomial s^2 + 2).
As the decimal symbol use [.] (dot).

Enter the numerator and denominator for the transfer function:

The numerator of the t. f.: [1.5]

The denominator of the t. f.: [1 0.75 1]

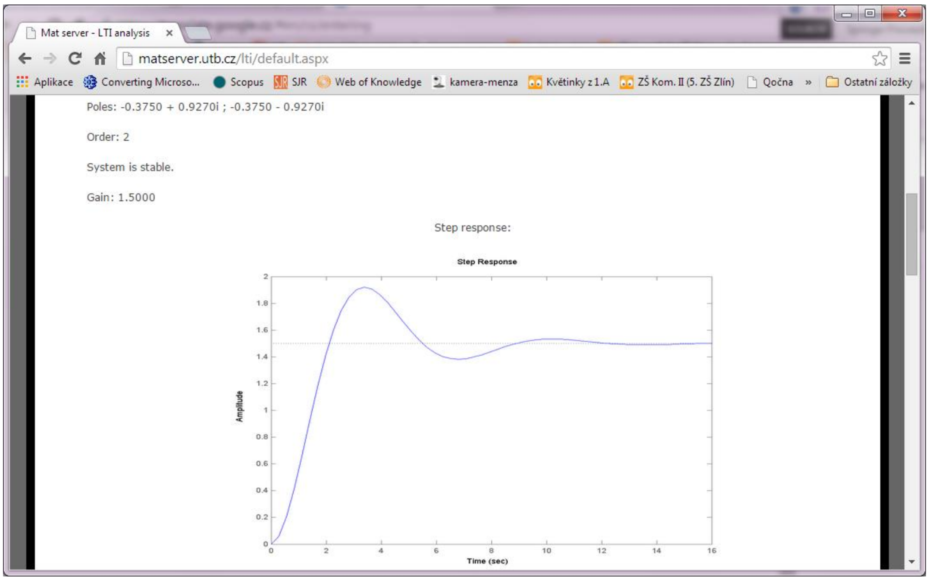
Terms of Use   Tomas Bata University in Zlín, Faculty od Applied Informatics - MAT Server (2013-2014)   gazdos@fai.utb.cz

Fig. 5. Entering the system parameters (transfer function description)

- The sixth step shows the selected properties form the previous step and enables to choose system characteristics to be displayed (step and impulse responses, bode and nyquist graphs, poles & zeros map in the complex plane);
- Finally, in the last step, all the entered information and performed transformation are summarized and displayed together with the required system properties and characteristics, as partly seen in Fig. 6. The graphs can be easily downloaded using the right mouse click in the *.png image format. In every part of the application it is possible to return to previous steps to correct the entered information or directly back home for a new session.

5 Conclusion and Perspective

This paper has presented how it is practically possible to connect web-applications with the computing system MATLAB. The result, in this case, is the web-application for linear time-invariant systems analysis suitable mainly for students (and their teachers) of control engineering and similar fields of interest with limited access to the MATLAB system (or similar computing tools). The whole process of the application development with motivation and goals, SW and HW tools, utilized functions for the analysis and the process of MATLAB-Web interconnection has been explained in detail. Description of the resultant web-application user interface is also given. The presented application is still in the development and it is currently being tested; some improvements are being prepared, e.g. the possibility to handle also multi input – multi output (MIMO) systems in a certain way. The important thing is that this application has a modular structure, it is open for improvements and accessible from anywhere with the Internet connection.



Poles: -0.3750 + 0.9270i ; -0.3750 - 0.9270i

Order: 2

System is stable.

Gain: 1.5000

Step response:

Fig. 6. Part of the final results screen

References

1. Attaway, S.: Matlab: A Practical Introduction to Programming and problem Solving. Butterworth-Heinemann, Boston (2013)
2. MATLAB – The Language of Technical Computing, http://www.mathworks.com/products/matlab
3. Wolfram, S.: The MATHEMATICA Book. Cambridge University Press, Cambridge (1999)
4. Wolfram Mathematica: Definitive System for Modern Technical Computing, http://www.wolfram.com/mathematica
5. Garvan, F.: The Maple Book. Chapman and Hall/CRC, London (2001)
6. Maple – Technical Computing Software for Engineers, Mathematicians, Scientists, Instructors and Students – Maplesoft, http://www.maplesoft.com/products/Maple
7. Facuna, J.: Web-application for the LTI Systems Analysis. Master's thesis, Tomas Bata University in Zlin, Zlin (2014)
8. System analysis, http://winf230e-1.vsb.cz/crl002
9. Chaparro, L.: Signals and Systems using MATLAB. Academic Press, London (2014)
10. Hansen, J.S.: GNU Octave Beginner's Guide. Packt Publishing, Birmingham (2011)
11. GNU Octave, https://gnu.org/software/octave
12. Gomez, C.: Engineering and Scientific Computing with Scilab. Birkhäuser, Boston (1999)
13. Scilab, http://www.scilab.org
14. The MathWorks, Inc.: Control System Toolbox: User's Guide. Natick, MA, USA (2013)
15. Mat Server – LTI analysis, http://matserver.utb.cz/LTI