WEB-APPLICATION FOR DIRECT CONTROLLER DESIGN AND TUNING FROM EXPERIMENTAL DATA

F. Gazdoš, D. Rakus

Department of Process Control, Faculty of Applied Informatics,
Tomas Bata University in Zlín, Czech Republic

Abstract

The number of web-applications employing external computing tools to perform complex calculations on-line is rising at the present time. This paper presents one such application enabling direct controller design and tuning from uploaded process data. It exploits interconnection of the MATLAB system with the latest web-technologies.

1 Introduction

Sharing and providing information using the World Wide Web 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, Wolfram Research's Mathematica, Maplesoft's Maple and others which are more suitable for complex calculations and corresponding data processing. This paper presents one such web-application for direct controller design and tuning from experimental data. Input data uploaded by a user are processed by functions from the MATLAB kernel and its toolboxes and the main result is optimal setting of controller's parameters according to a selected criterion. The developed application is a part of the project "Direct methods of controller design and tuning" [1] and it was designed within the scope of the Master's thesis [2].

The contribution is structured as follows: after this introductory part, main goals are outlined in the next section, followed by the employed hardware, software 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 algorithms for controller design and tuning follows. Further, the web-application user interface is presented and explained in detail and followed by some concluding remarks at the end of the paper.

2 Goals

When designing the interactive web-application, main goals consistent with the project [1] were formulated as follows: development of a web-application which allows designing or fine-tuning of a linear single input – single output discrete-time controller with the following input information:

- data-file with process input-output data (either from open or closed-loop),
- controller structure,
- criterion for optimization.

Output information provided by the application:

- optimal setting of controller's parameters according to the selected criterion,
- additional information concerning stability of the designed control system.

For the controller optimization, direct methods developed within the scope of the project [1] were implemented, as a part of the project results. More information about the project and its results will soon be accessible via the web-pages of the developed application [3].

The designed 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 system. Apart from this, it can be administered and updated easily.

3 Tools

Generally, web-applications are operating on web-servers which are computers with a suitable operating system and a special application (also called a web-server) – 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. The operating system is represented by the Microsoft Windows Web Server 2008 with the web service IIS 7.0 (Internet Information Services) acting as the web-server. An alternative way could be the usage of e.g. the Linux operating system with the Apache web-server.

Web-applications have usually several tiers and most common is the three-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 (Hyper text Markup Language), CSS (Cascading Style Sheets) and AJAX (asynchronous JavaScript and XML) tools. The last one enables development of interactive web-pages.

The middle application tier (also known as the “logic tier”) processing requests and data from the presentation tier and generating user-friendly interface dynamically was designed using the ASP.NET (successor to the Microsoft Active Server Pages) technology, a part of the Microsoft .NET Framework (ver. 3.0). For algorithms implementation, C# programming language and the Microsoft Visual Web Developer 2008 Express Edition software was used. Complex computations needed for the controller design and tuning are realized here with the help of the system MATLAB. The following MATLAB components were employed for the required optimization tasks, corresponding data-processing and deployment to the Web:

- MATLAB 7.7 (R2008b)
- Control System Toolbox 8.2
- MATLAB Builder NE [for .NET] 3.0
- MATLAB Compiler 4.9
- System Identification Toolbox 7.2.1

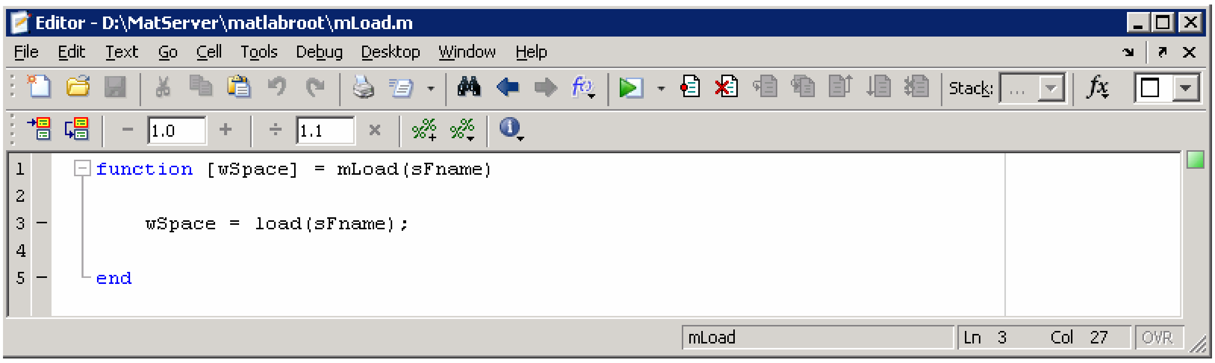
An alternative way to the usage of ASP.NET technology could be e.g. usage of the JSP (JavaServer Pages) technology.

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 and by the used version of the database server (Microsoft SQL Server 2008 Express).

4 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 as we have had experience with it.

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).



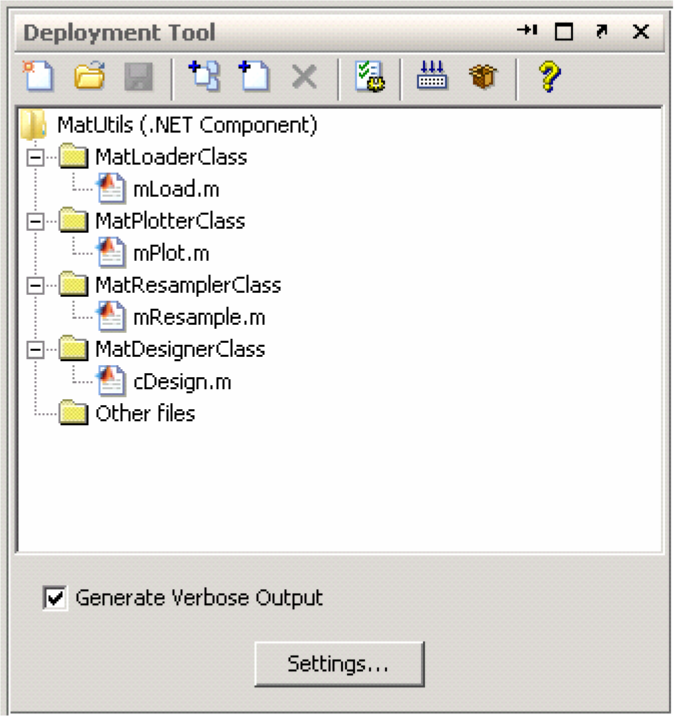
```
function [wSpace] = mLoad(sFname)

    wSpace = load(sFname);

end
```

Figure 1: Example of a deployed M-file

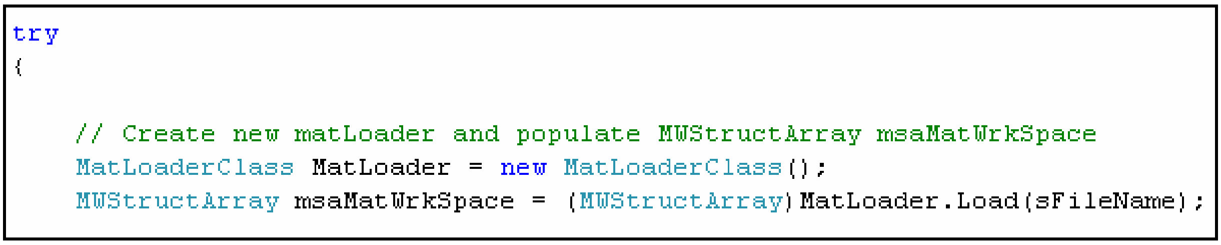
The resultant components usable in the web-application are then created from the m-files (and relevant toolboxes) using the MATLAB Compiler toolbox. This toolbox uses other supporting toolboxes for the compilation, depending on the chosen technology of web-application development. For the applications 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.



☑ Generate Verbose Output

Figure 2: MATLAB 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.

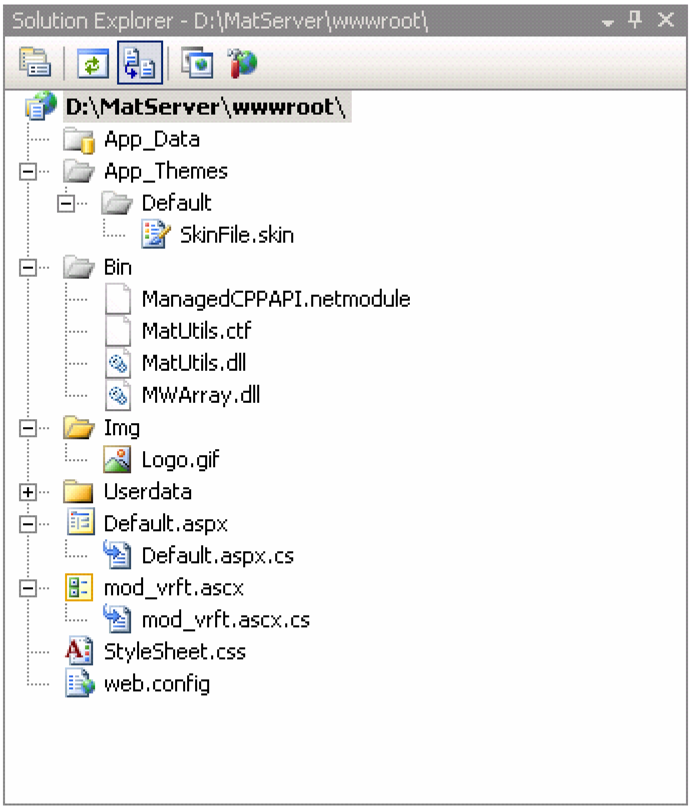


```
try
{

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

Figure 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.



```
Solution Explorer - D:\MatServer\wwwroot\
D:\MatServer\wwwroot\
  App_Data
⊟ App_Themes
  ⊟ Default
      SkinFile.skin
⊟ Bin
    ManagedCPPAPI.netmodule
    MatUtils.ctf
    MatUtils.dll
    MWArray.dll
⊟ Img
    Logo.gif
⊞ Userdata
⊟ Default.aspx
    Default.aspx.cs
⊟ mod_vrft.ascx
    mod_vrft.ascx.cs
  StyleSheet.css
  web.config
```

Figure 4: Reference to a .NET component

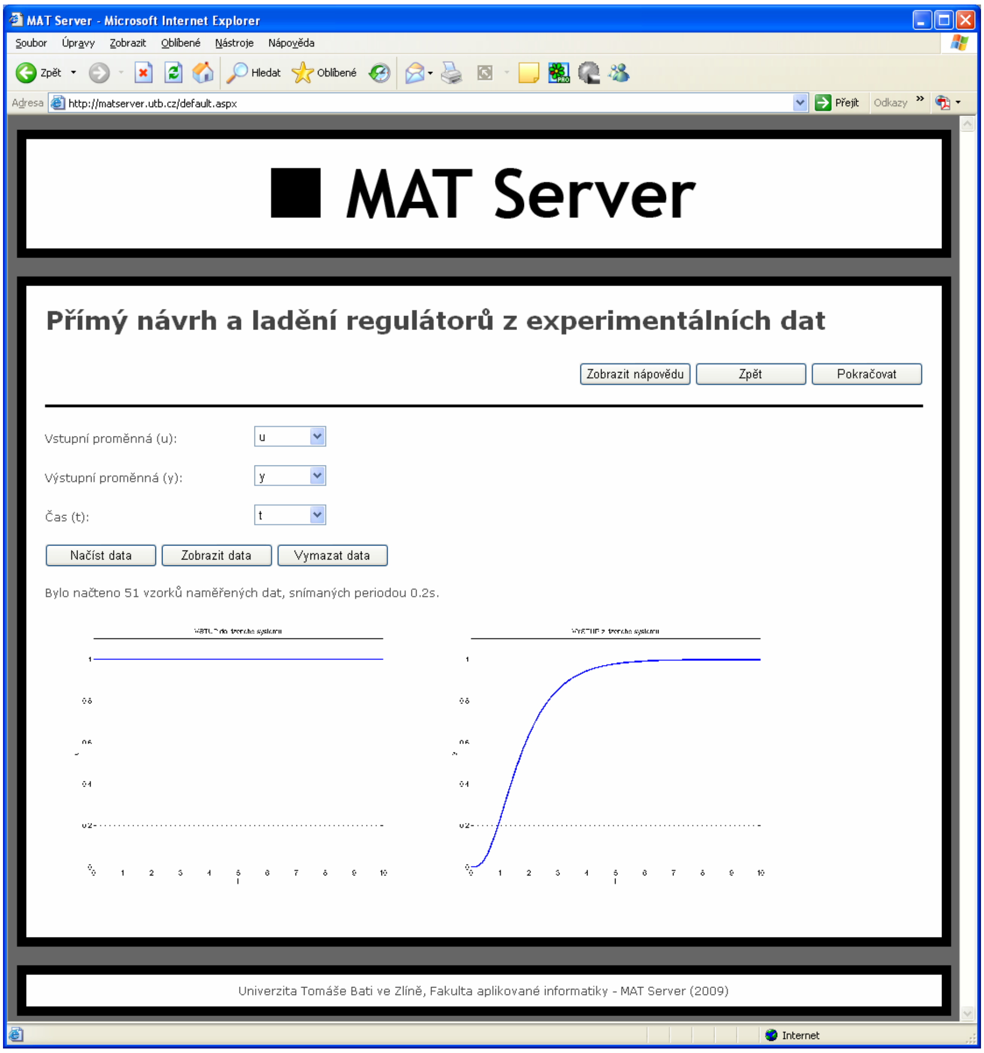
5 Controller Design and Tuning Method

For the controller optimization, direct methods developed within the scope of the project [1] were adopted. As they are direct they skip over the system identification step and use the measured process input – output data and suitable filtration to identify an optimal controller directly. The basic employed ideas are based on the Virtual Reference Feedback Tuning methodology, e.g. [4], [5] and the Unfalsified Control Concept, e.g. [6], [7]. The implemented algorithms use modifications of the above mentioned approaches to design or fine-tune a linear discrete-time single input –single output controller of a chosen structure. Here, criteria for the controller optimization can be chosen as the desired settling-time of the controlled process output, or it is possible to include also a desired maximum overshoot or, if needed, a general desired model of the closed-loop behaviour in the form of a continuous-time transfer function. Details about the algorithms can be found e.g. in the works [8], [9] and soon will be available through the web-application pages [3].

6 Web-Application User Interface

A testing version of the developed application (in the Czech language only for the moment) is accessible through the Internet at the following URL: http://matserver.utb.cz. The English version is being prepared. The application user interface (presentation tier) is designed as a guide and allows entering the required information for the controller optimization in several connected steps. Common control buttons such as "*Continue*" / "*Back*" and "*Show help*" / "*Hide help*" are used to navigate throughout the application and provide corresponding help information where needed. The interface consists of these main parts:

- Start-up screen which provides basic information about the web-application and its usage; a language control button (Czech / English), information about the project and its results will be available here soon.
- In the next step, process data can be uploaded. At present MATLAB binary data files (*.*mat* files) are supported and have to contain at least process input, output and time data stored in the vector variables of the "*array*" type. Processing of the standard text files is being prepared. It is also possible to download a test data file to test the application or to delete the already loaded data file and upload a new one.
- Further, process input, output and time data are selected from the uploaded file, loaded into the memory and can be displayed using graphs. A user is informed about the number of loaded samples, sampling period and can delete already loaded variables and select new ones to work with, see Fig. 5.



http://matserver.utb.cz/default.aspx

Přímý návrh a ladění regulátorů z experimentálních dat

Vstupní proměnná (u): u

Výstupní proměnná (y): y

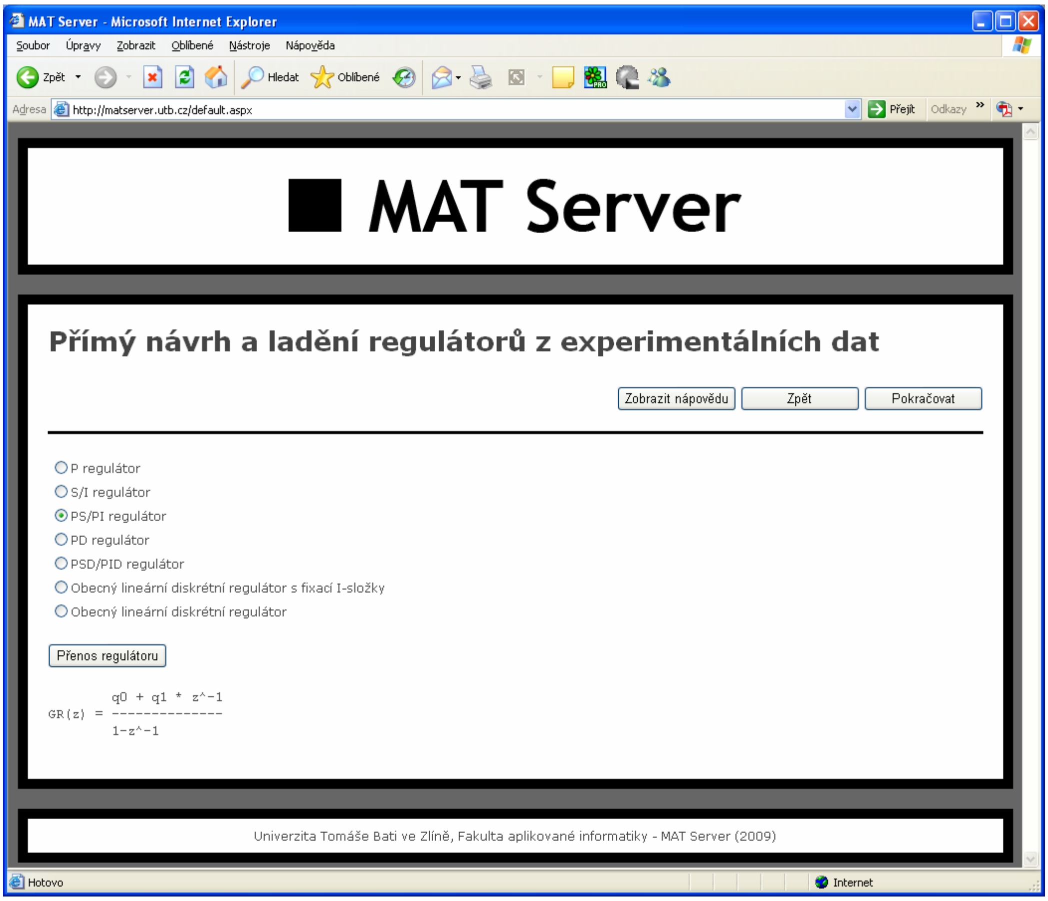
Čas (t): t

Bylo načteno 51 vzorků naměřených dat, snímaných periodou 0.2s.

Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky - MAT Server (2009)

Figure 5: Variables selection and display

- A suitable controller structure can be selected in the fourth step – a prospective user can choose from 7 different types of controllers, starting with the proportional controller and ending with a general linear discrete-time controller. The corresponding transfer function of the chosen controller can be displayed here, see Fig. 6 for details.


MAT Server - Microsoft Internet Explorer
Soubor Úpravy Zobrazit Oblíbené Nástroje Nápověda
Zpět
Hledat
Oblíbené
Adresa http://matserver.utb.cz/default.aspx
Přejít
Odkazy
Hotovo
Internet

MAT Server

Přímý návrh a ladění regulátorů z experimentálních dat

Zobrazit nápovědu | Zpět | Pokračovat

○ P regulátor
○ S/I regulátor
● PS/PI regulátor
○ PD regulátor
○ PSD/PID regulátor
○ Obecný lineární diskrétní regulátor s fixací I-složky
○ Obecný lineární diskrétní regulátor

Přenos regulátoru

```
        q0 + q1 * z^-1
GR(z) = --------------
        1-z^-1
```

Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky - MAT Server (2009)

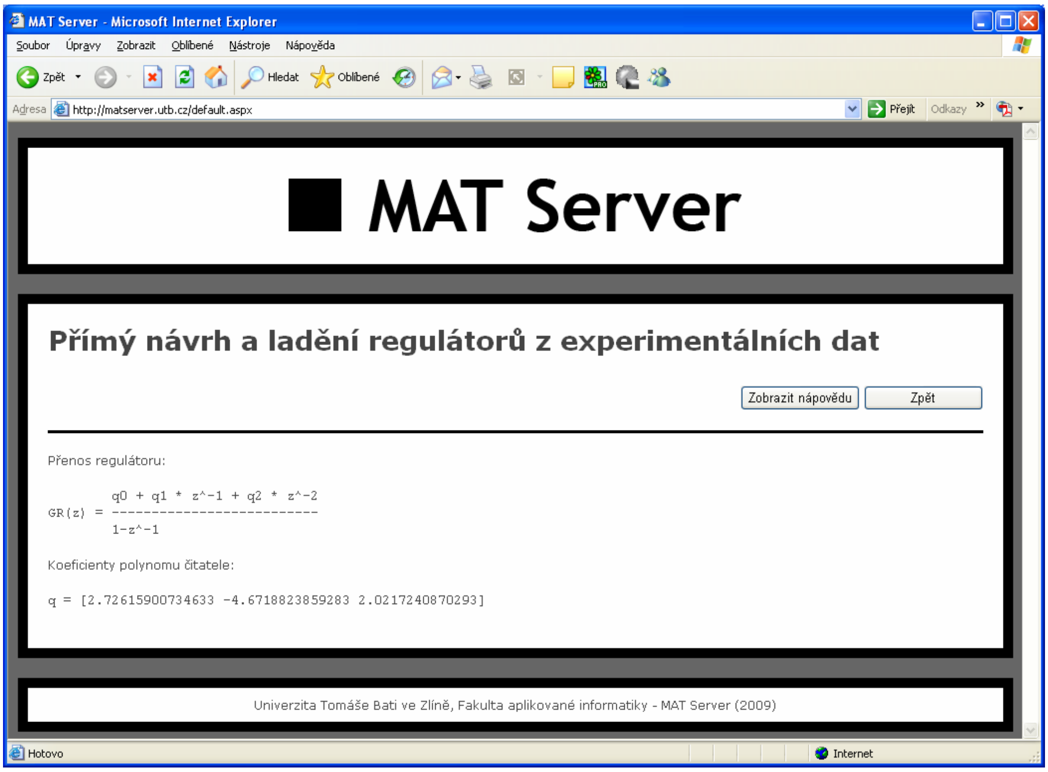
Figure 6: Controller choice

- In the next screen, additional information concerning the controlled process and data can be selected such as instability, non-minimum-phase behaviour or significant noise corruption of the data. This information will then be used to choose optimal algorithms for the controller design.

- The sixth step enables to choose the criterion for controller optimization and enter the corresponding information. At present, it is possible to select either a desired settling-time of the controlled variable, or to add also a desired maximum overshoot or to choose a general desired model of the closed-loop behaviour entered in the form of a continuous-time transfer function (vector of numerator and denominator coefficients in the MATLAB notation). A user will soon be able to display the desired model of behaviour graphically here.

- Next screen summarises all the entered information and options made in the previous steps and allows performing computations to design the optimal controller. In every part of the application it is possible to return to the previous steps and change / correct already entered data.

- Finally, in the last step, the user obtains optimized controller's coefficients according to the chosen criterion, selected controller structure and all the entered relevant data. Additional information concerning stability of the designed control loop together with export of the results is being prepared. Conversion of the resultant discrete-time controller to its continuous-time counterpart is also intended to enable tuning of already implemented and working continuous-time controllers such as commonly used PI's and PID's. Present appearance of this final screen is displayed in Fig. 7.


MAT Server - Microsoft Internet Explorer
Soubor Úpravy Zobrazit Oblíbené Nástroje Nápověda
Zpět
Hledat
Oblíbené
Adresa http://matserver.utb.cz/default.aspx
Přejít
Odkazy
MAT Server
Zobrazit nápovědu
Zpět
Hotovo
Internet

Přímý návrh a ladění regulátorů z experimentálních dat

Přenos regulátoru:

```
          q0 + q1 * z^-1 + q2 * z^-2
GR(z) = ---------------------------
          1-z^-1
```

Koeficienty polynomu čitatele:

```
q = [2.72615900734633 -4.6718823859283 2.0217240870293]
```

Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky - MAT Server (2009)

Figure 7: Results screen

7 Conclusions

This paper has presented how it is practically possible to connect developed web-applications with the computing system MATLAB. The particular result is the web-application for direct controller design and tuning from experimental data. The whole process of the application development with goals, used technologies, methodologies and the process of MATLAB-Web interconnection has been explained in detail together with description of the resultant web-application user interface. The resultant application, however, is not yet mature or finalised at the moment – it is still in development and many improvements are being prepared, some of them were mentioned within this paper. The important thing is that it is prepared not only for improvements in the presented application for controller design but the resultant web server solution also enables simple implementation of other computationally complex web-applications in a modular way.

Acknowledgements

This work, as a part of the project “Direct methods of controller design and tuning”, was supported by the Czech Science Foundation under the project no. 102/07/P148. This support is greatly acknowledged.

References

[1] F. Gazdoš. *Direct methods of controller design and tuning*. The Czech Science Foundation project no. GACR-102/07/P148, 2007-2009.

[2] D. Rakus. *Web-application for direct controller design and tuning from experimental data*. Master’s thesis. Tomas Bata University in Zlín, 2009.

[3] http://matserver.utb.cz

[4] M. C. Campi, A. Lecchini and M. Savaresi. Virtual reference feedback tuning: A direct method for the design of feedback controllers, *Automatica*, Vol. 38, pp. 1337-1346, 2002.

[5] A. Sala and A. Esparza. Extensions to ‘Virtual Reference Feedback Tuning: A Direct Method for the Design of Feedback Controllers’, *Automatica,* Vol. 41, pp. 1473-1476, 2005.

[6] M. G. Safonov and T. C. Tsao. The Unfalsified Control Concept: a Direct Path from Experiment to Controller. *Feedback Control, Nonlinear Systems, and Complexity*. B.A. Francis and A.R. Tannenbaum, eds, Springer-Verlag, New York, pp. 196-214, 1995.

[7] M. G. Safonov and F. B. Cabral. Fitting controllers to data. *Systems and Control Letters*, Vol. 43, No. 4, pp. 299-308, 2001.

[8] F. Gazdoš, P. Dostál and R. Pelikán. Optimizing settling-time and overshoot by a direct method. *International Review of Automatic Control (I.RE.A.CO.)*, Vol. 1, No. 3, pp. 287-293, 2008.

[9] F. Gazdoš and P. Dostál. Direct controller design and iterative tuning applied to the coupled drives apparatus. *Journal of Electrical Engineering*, Vol. 60, No. 2, pp. 57-68, 2009.

František Gazdoš
e-mail: gazdos@fai.utb.cz
tel.: +420 57 603 5199
fax: +420 57 603 2719
http://www.fai.utb.cz