SCRIPTRUNNER: Web Application for the Interactive Learning

Mario Essert, *Member, IEEE*, Bojan Mauser, and Ivica Nakić

***Abstract*—The purpose of the Internet is an exchange of the information which is obtained through the creation and the use of dynamical and interactive web pages. The web-application program Scriptrunner integrates in a simple and easy way such a creation and use in the form of the "living" documents - documents which can contain the text, pictures and multimedia and can also incorporate the execution of the programs written in various programming and scripting languages. Moreover, the reader of such a document has not only option to edit the input parameters of these programs, but can also write and execute user's own programs without being forced to install the necessary software on its computer, except browser.**

I. Introduction

Science learning is often considered to be a difficult pursuit. Today we have a possibility to increase the quality of science and technology education by the use of web technology. The World Wide Web has given educators unprecedented opportunities to provide information to students both within their own classes and around the globe. The need to provide students with both a strong theoretical base and engineering ability is our major challenge for education. Theoretical issues, typically related to mathematical techniques, can be well taught in the ordinary classroom style. Engineering ability, on the other hand, requires insight and intuition, which are not so easy to convey. To join these two requirements and make learning math and engineering techniques easy to students, we have developed the web application Scriptrunner, which can be accessed through the Intranet or Internet.

Through the use of Scriptrunner interactive capabilities not available in lectures or in texts students can have a heightened level of interaction with the material. Through the use of networking, traditionally isolated, distance education students are brought into a community of communicating peers. Courses can be developed once, by experts, and then replicated at low cost to any size audience at any location. This creates a potential for improved course quality, improved access, and reduced cost. Learning is learner-driven. Thus each delivery of the course is self-paced, and customized to accommodate either strong or weak students, as well as students with preexisting knowledge of some aspects of the course material.

The Scriptrunner is an application developed through the academic community and is used as a teaching aid in the various mathematical and computer science courses, but its use can be extended to other fields, as well as in the publishing and education of the widest range. It can also be used for creating the databases of computer programs, collecting the interactive on-line textbooks and tutorials, collecting benchmarks programs etc. It consists of a lot of modules (plug-ins) that provide education and distance learning at the easiest way. The simple and user-friendly module interfaces facilitate comfortable work.

Scriptrunner is a web application that enables editing and remote execution of the computer programs written in various programming languages. It also enables creation of the interactive e-books that can contain, besides mathematical equations and symbols, ready-to-run programming examples. These examples can be displayed either as an editable source code or as an executable file in such e-books. It is possible to change the input parameters of programs, thus creating the opportunity for the users to experiment. On-line writing of documents is possible through the LaTeX toolbar or internal HTML editor. Especially valuable feature of the Scriptrunner is the possibility of creating the "living" pdf-documents which enables the creation of the interactive documents which can also be used as a high-quality printed material. Of course, such documents can be used outside the Scriptrunner. Since this educational tool is intended for all courses that use mathematics and computer programs, the possibility of its usage is fairly wide.

Scriptrunner supports writing and execution of programs written in standard programming languages: C/C++, Pascal, Fortran, Java, JavaScript, Python and PHP which are all executed using open source compilers. It also supports programs for a variety of professional mathematic tools, both open source and commercial, such as Matlab, Octave, Maple and Scilab.

Scriptrunner is built using PHP/MySQL programming languages and is designed to run on Linux systems, and also uses JavaScript and Ajax technologies in the graphics interface. Since it uses open source technologies, it was possible to significantly lower the development costs and the costs of deployment. It is a Web application based on the client/server technology, so only browser is needed for work on the client side.

Even though several open source e-learning solutions are available, to the authors' best knowledge, no previous solution serves as a web platform to write and execute scripts written in various programming languages.

Today, Scriptrunner is used in a variety of courses on two Croatian universities, with students numbered in hundreds. It has proven to be a valuable teaching tool.

The rest of the paper is organized as follows. Section II presents related work and motivation. Section III presents the overall structure of Scriptrunner. Sections IV, V and VI

deal with Scriptrunner's capabilities in terms of the creation of interactive HTML documents, programming simulations and interactive PDF documents, respectively. Conclusions and directions for future work are in Section VII.

II. Related work and motivation

With the rapid development of the science and technology, the problem of the transfer of the knowledge becomes an increasingly harder problem. The transfer of knowledge through the examples, which is often the best approach, is getting more complicated because either of the complexity of the examples and/or the huge number of them. A part of the problem could be solved if we would have examples, in which we could change various parameters and/or the underlying algorithm, thus essentially obtaining a series of examples instead of just one. The reader could start with the simplest situation, and then increase the complexity of the example stepwise. In that way the example is adapting to the reader, not vice versa. The more experienced reader can immediately start with the more difficult examples, not wasting it's time with the simple ones.

That was the motivating idea of the number of web applications, both commercial (e.g. WebCT), and open source (e.g. Moodle). The common approach is nowadays the use of HTML and Flash technologies. The development of such applications goes back to early nineties (see, for example, [1], [2]), but is also of great interest today (see, for example [3], [4]). The shortcomings of the previous applications was that the content developer couldn't easily reuse the existing programs and needed to learn to use a sophisticated authoring tools. Our motivation was to create a web application which doesn't have these shortcomings. Namely, we wanted to implement a computer-aided learning system which, besides other properties common to most web applications, also

- Can be used by educators to create content without a long training period,
- Doesn't need expensive software to run,
- Enables the use of already written examples, especially programming examples,
- And is operating system agnostic.

III. Scriptrunner outside - an user friendly environment

Computer programs and interactive documents are stored on a remote computer (server), on a built, specially developed, remote file system. The user interface is similar to that of the MS Windows Explorer program.

Every user has his/her work environment (folder tree structures) and all settings and files created during on-line session remain on the server for the next session. This 'My Files' folder is unique for each user and contains the work environment that a user can organize. There is also a 'Public Files' folder where users can publish their files for others to see and execute.

There are also two or tree special folders in a folder tree:

- Settings – which contain user preferences that can be changed to meet own graphical user needs.

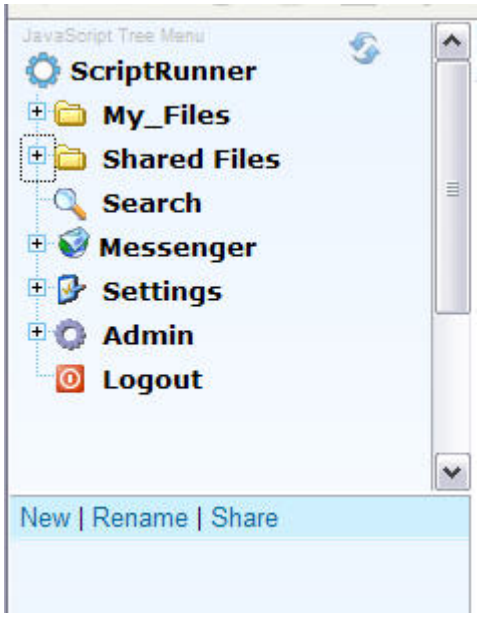


Figure 1. The remote file system

- Admin folder - which is visible and accessible only to user with administrator privileges
- Logout – to leave the program

Folder tree structure enables marking, renaming, deleting and moving (with all sub-branches) folders and/or files.

In Scriptrunner there are three types of uses which are different in authorizations they have:

- Ordinary user of Scriptrunner has his own 'My files' folder and is authorized to edit subfolders and files. This user can view and execute Public files, while he/she can edit his interface through the settings in the Settings folder.
- Administrator is authorized to add new users and groups, erase the existing users; assign the users' licenses to the program languages (plug-ins) which they can execute, etc. Administrators are also hierarchically organized.
- Public user is a user authorized to start and view the files in Public files folder.

Figure 2. The user interface

After successful login, following screen with four frames appears:

- Tree frame – contains already described folder maps
- Folder actions frame – contains actions like New, Rename, etc. Right-clicking the folder can also be used for the most common actions.
- Working frame – contains the most important part for editing sources/documents and running them.
- Status frame - displays messages relevant to the current actions.

IV. On-Line HTML documents

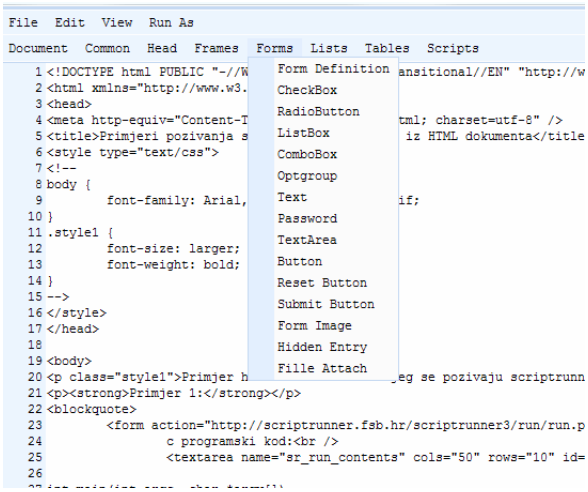


Figure 3. HTML editor

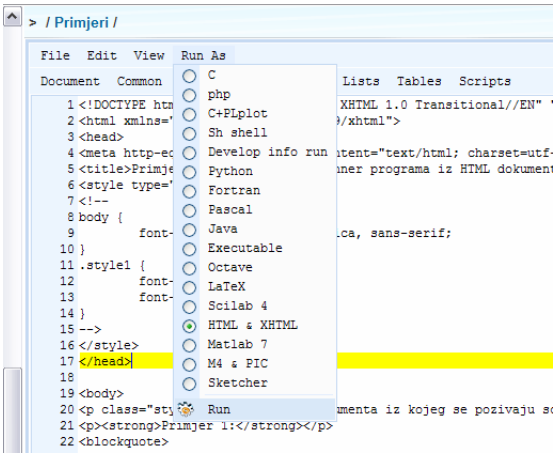


Figure 4. Run menu

A very interesting possibility provided by Scriptrunner3 is writing and publishing of the digital books. That possibility spreads usage of Scriptrunner to a much wider range than program running and working in program languages. Moreover, each such book becomes 'live' because the programs with the preferred, regularly different, input data can be called from it. The document (book, script, handbook) in Scriptrunner has numeration of tables, figures, equations, expressions or other objects together with their assigned references (references to objects). When the final document is created or published, renumeration of titles and quoted objects and their references is also possible. Writing of books, textbooks, scripts and similar is possible using two on-line ways:

- In LaTeX form which is converted into pdf document
- In HTML form, which can have interactive (live) programming examples.

For the LaTeX form there is a LaTeX toolbar with many falling menus with numerous symbols. For the HTML form, open-source FCK-editor was used, which has the toolbars similar to those in WORD. All the options for writing the text are available here: size and font type, alignment, counting, entering pictures, tables hyperlinks etc. If a Word document is available on a local computer, formatted text can be copied and pasted in this editor thus preserving previous formatting from the document. Also, the written Word text can be dragged and dropped into editor window of Scriptrunner.

V. 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 Octave program. Student calls an on-line assignment through Scriptrunner, he studies the problem and writes his Octave code for the problem solution. A part of the code may be already written together with an assignment. 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:

- Assignment's title
- Assignment's description with a picture (figure)
- The set program code which shall not be changed
- A detailed description of the required solutions
- Text area for the entry of a student code
- 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.

Another common use of Scriptrunner is a generation of complex examples which show a connection of various algorithms or engineering techniques. In the figure 5 we show an example of the use of Scriptrunner in teaching of control algorithms for robot manipulators.

VI. Interactive PDF documents

Scriptrunner has come a long way in facilitating the easy creation of interactive PDF documents, which enables one to create a document which has a high typographical quality and has interactive elements. The two main benefits which Scriptrunner brings into this field are:

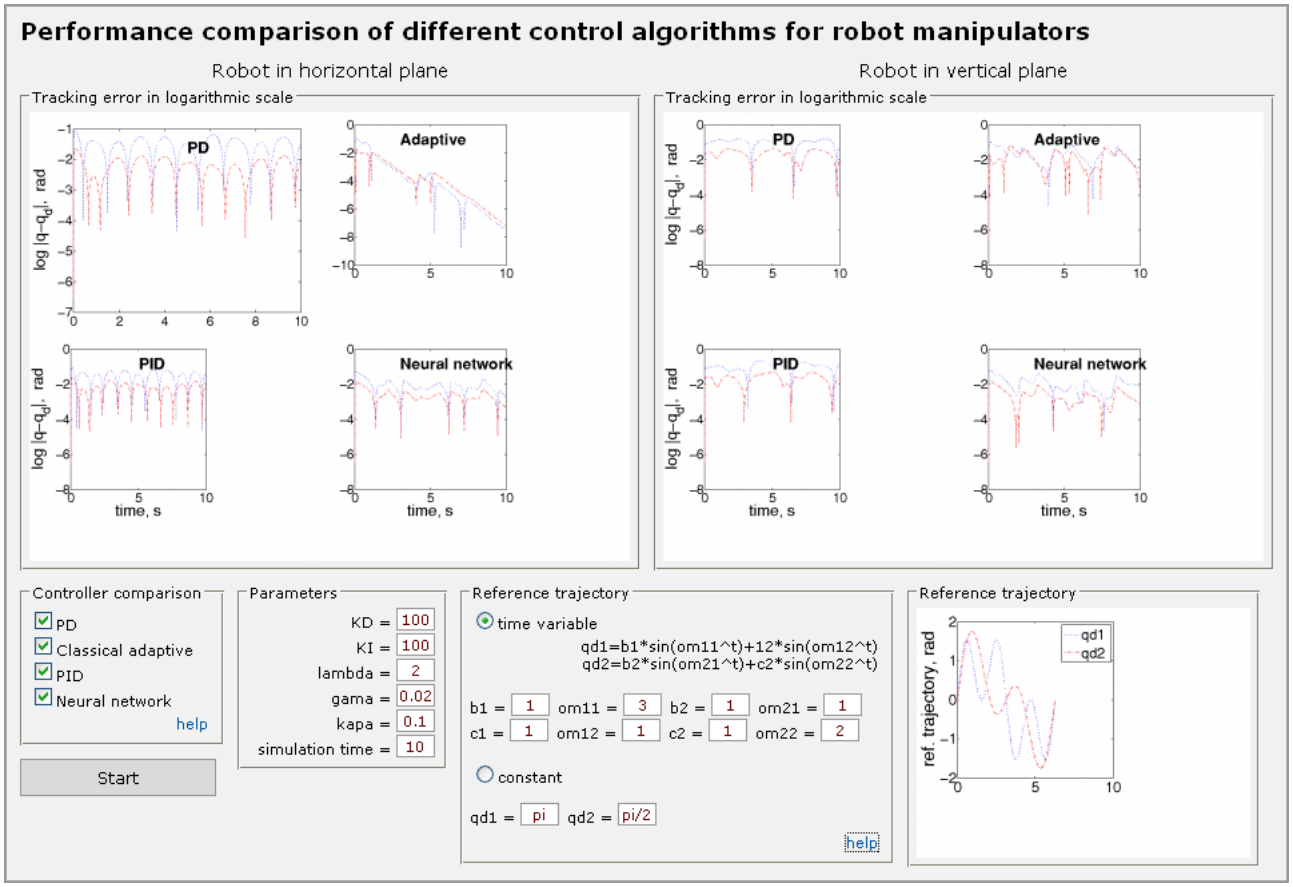


Figure 5. Regulator example

1) Easy creation of forms such as radio buttons, check boxes and text fields in PDF documents.
2) Ability to execute scripts or programs from Scriptrunner in PDF documents, not depending on the location of the documents, only on Internet access.

The engine which Scriptrunner uses for the generation of PDF files is LaTeX. This choice also enables users to typeset complicated technical documents with a lot of mathematical notation, obtaining professionally looking document. Although one can write the LaTeX code in her/his favorite editor, Scriptrunner has also a built-in LaTeXeditor:

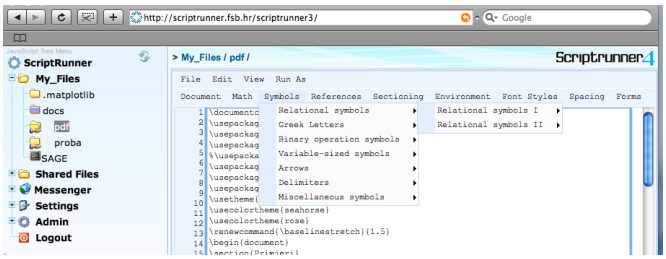


Figure 6. LaTeX editor

This editor can be used, for example, for some last-minute changes to the document.

The menu in editor contains various frequently used LaTeXcommands, for example the menu Environment has a number of frequently used LaTeX environments:

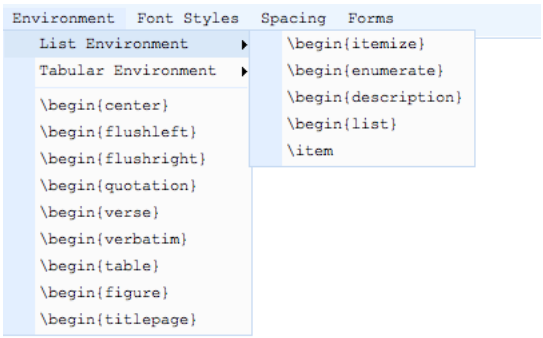


Figure 7. LaTeX editor menu

In the Scriptrunner settings the user can choose whether he/she wants to use latex or pdflatex compiler. If one chooses latex compiler, the result of the Run command in Scriptrunner will be Postscript document, and pdflatex compiler creates PDF output. The interactive elements can be used only with pdflatex compiler.

The menu Forms enables easy creation of forms in PDF document:

For example, the LaTeX command for the creation of the text field is *\inputTextField*, and has three parameters: name, length (in the numbers of letters) and default value. The extension of LaTeX which enables the use of these commands (in the form of a package) is based on the package AcroTeX [5], which in

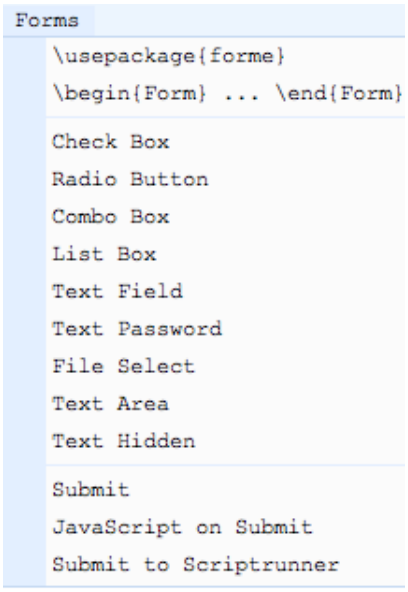


Figure 8. LaTeX forms menu

turn is based on the package hyperref [6].

An example of the possible use of Scriptrunner in creating interactive documents is the following scenario.

If one writes, for example, notes on control theory for students, he/she probably wants to give an example of the transformation of transfer function to the state system form. Instead of just giving one example with fixed parameters, with the use of Scriptrunner he/she can create an interactive example where the students can change parameters, hence enabling the students to obtain better understanding of the aforementioned transformation. To achieve this, first he/she will write a little script in, for example, Matlab which calls the corresponding function, in this case *tf2ss*. Suppose that the script is in the file *tf*.m. The next step is then to find the link needed to run this file. This is given in the file access info of created file.

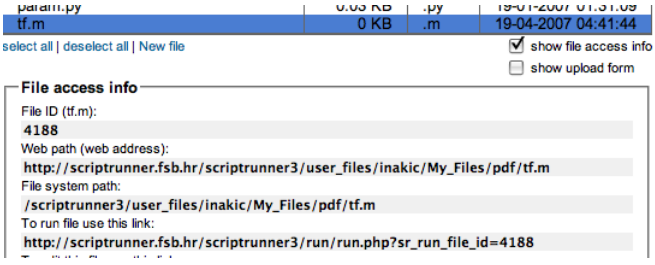


Figure 9. File access info

In this example, the link is *http://scriptrunner.fsb.hr/scriptrunner3/run/run.php?sr_run_file_id=4188*.

Next step is to put the interactive example in the PDF document. Hence one has to create forms which gather needed data (in this case for the transform function), and a submit button which will send this data to our script *tm.f*. We can collect data by the use of *\inputTextField*, in this case we need to collect the coefficients of the numerator and denumerator polynomials of the transfer function, and submit the data by the use of the command *\inputSubmitJavaScript*. For example, this is the needed code for the submition of data:

\inputTextHidden{sr_run_commline}{}

\inputSubmitJavaScript{Send}{ var SubmitField = new Array("num", "den");

SubmitFields(SubmitField,

"http://scriptrunner.fsb.hr/scriptrunner3/run/

run.php?sr_run_file_id=4188"); }

How does our script receive these data? By the use of the *sr_run_commline* variable, the data sent by our PDF file are treated like a command line input to our script. So, we need to write our script as if we would gather the needed data from the command line. For example, this simple script written in Python writes to the output all data which were sent by our PDF document:

from sys import argv

print argv

By default, Scriptrunner is treating the output as a plain text. But sometimes some other kind of output is desirable. This can be achieved with the use of *sr_run_content_type* variable. For example, if we would like to have output in html, we just need to send *sr_run_content_type variable* with the value text/html to the Scriptrunner. This can be achieved, for example, by appending *&sr_run_content_type=text/html* to the html address. In our example we would write: *http://scriptrunner.fsb.hr/scriptrunner3/run/*

run.php?sr_run_file_id=4188&sr_run_content_type=text/html. Our script (in Python) with the html output could be:

from sys import argv

print "<html><body>"

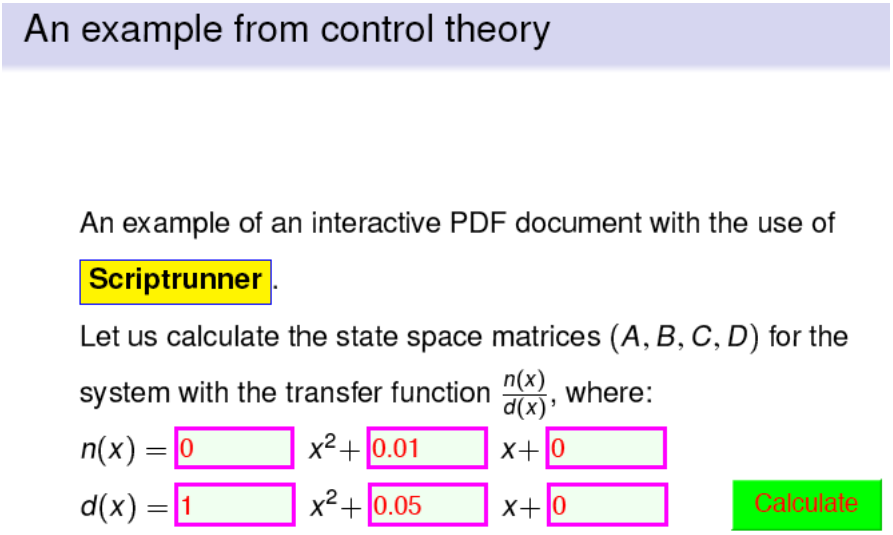
print argv

print "</body></html>"

The user can, of course, choose any mime type he/she needs.

As a concrete example, we wrote a little script in Octave which calculates the state space matrices (A, B, C, D) for the system with the given transfer function. The output of the script is formatted in HTML. We use this script in the interactive PDF document where the user chooses the coefficients of the transfer function and receives the state space matrices in a well-formatted HTML document. For the generation of mathematical notation in HTML we used Javascript library jsmath which enables the use of LaTeX notation in HTML documents. In that way we can achieve a consistent look of generated documents.

Figure 10. PDF document



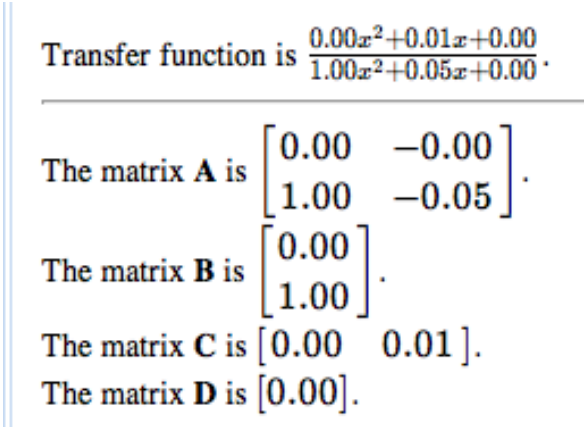


Figure 11. The output from the PDF form from Figure 10

We can also incorporate images in the output file. All we need to do is generated the image and save it in the filesystem, and then use HTML tag to include the image in the output file. For example, this code (in Python)

```
from pylab import *
t = arrange(0.0, 1.0+0.01, 0.01)
s = tan(2*2*pi*t)
plot(t,s)
savefig('plot.png')
print """\
<html> <body>
<img src="/path/to/plot.png"/>
</body> </html>"""
```

generates a HTML document with a generated image. Here the path to the generated image is the file system path which can be obtained from the file access info.

Of course, we can also include other multimedia objects in HTML, or any other media type which can be shown in web browser.

Scriptrunner also incorporates plugins for creation of figures by the use of various LaTeX dialects which are capable of exporting figures to common graphical formats such as JPEG, PNG, PDF and EPS.

VII. Conclusion and Future work

The described module is only one in a series of education modules which have been applied in a number of subjects within a school-system. These modules include home-works, laboratory exercises, and video demonstrations, reports, preliminary exams, finals and surveys. Since they may involve execution of any program, they truly have more advantages in comparison to other similar modules of the popular or commercial programs (Moodle, WEB CT). Surveys follow the complete teaching process – lectures, exercises, and exams, both, in terms of knowledge transfer (teaching material, contents, encumbrance etc.) and in terms of technical performance. Interactive work with a Scriptrunner was best accepted for home-works, reports and lab exercises, and less for exams (65 % of students choose to be tested on a paper, instead at computer). However, there was virtually no student who objected to the described opportunity of an interactive work of the HTML/PDF document. If at some time he was not able to establish an on-line connection he would read the document in a common way, and yet, if he managed to get on-line, then he could perform the given tasks or solve his own examples and check the obtained results on-line. Only one manual has been written in an interactive HTML format so far, but many lectures and exercises are available in interactive PDF formats and are offered to students for every lesson. Soon all these documents will be linked into the one e-manual with its hard copy as well (it will include only the crucial, invariable information) and its extension in a PDF format. Such manuals have another advantage – since they secure the copy right by means of a password for a program execution over a Scriptrunner, they also deliver valuable information to program's authors: i.e. how many times particular book sections have been processed, which example was performed and how many times, etc.

The anticipated further extension of such documents refers to involvement of streaming video (off or on-line). New programs have been developed and they enable such extension. At our laboratories, e.g. in electrical engineering or hydraulics

& pneumatics, this will allow a student to self-reliantly execute user's lab exercise from a distance (at the time allocated to him by the program, as was already done for preliminary exams or finals), and to receive in his document all video information from the laboratory (aided by two installed web cameras) on-line. For the object control, he shall enter his program code into the foreseen template; then he shall start the distance controlled program and shall watch within the same document whether it has been realized and how all happened at the real object. The initial experiences with similar laboratory automated distance control are already available. Other extensions refer to implementation of such documents in the field of liberal arts. Psychological tests have been conducted (perception measurements) and by connecting to video information the new quality of such measurements shall be established. Finally, the implementation of the system in paleographic issues of cultural heritage conducted at the Old Slavonic Institute has to be mentioned too. Glagolitic fonts for such documents have been developed, interactive browsing of ancient names, measurements, display of geographical sources and alike is available, too. Hierarchical structure of the saved documents in the Scriptrunner's tree and interactivity inside are a very strong incentive for further development of numerous applications, both from technical and from humanistic fields.

REFERENCES

[1] D. I. E. Bilota, M. Fiorito and P. Pantano, "An educational environment using www," *Computer Networks and ISDN Systems*, no. 27, pp. 905–909, 1995.

[2] S. F. B. Ibrahim, "Advanced educational uses of the world wide web," *Computer Networks and ISDN Systems*, no. 27, pp. 871–877, 1995.

[3] M. L. D. Kanellopoulos, E. Sakkopoulos and A. Tsakalidis, "Using web-based teaching interventions in computer science courses," *IEEE Transactions on Education*, vol. 50, no. 4, pp. 338–344, November 2007.

[4] J. G. A. G. G. S. E. S. J. T. C. Bouras, P. Destounis and T. Tsiatsos, "Efficient web-based open distance learning," *Journal Telematics and Informatics*, vol. 17, no. 3, pp. 213–237, 2000.

[5] Acrotex. [Online]. Available: http://www.math.uakron.edu/~dpstory/webeq.html

[6] Hyperref. [Online]. Available: http://www.tug.org/applications/hyperref/

PLACE PHOTO HERE

M ario Essert (1954) is a Full Professor in the Department of Control Engineering at the Faculty of Mechanical Eng. And Naval Arch., University of Zagreb. He received Ph.D. degress (1987) in the computer science from the Faculty of Electrical Eng. (ETF) at the University of Zagreb. As a member of the Group of Discrete mathematics at ETF (led by Prof. V. Ćepulić, dr.sc.) he participated in four international co-projects: Mainz (Germany), Kiev (Ukraine), Hangzou (China) and Heidelberg (Germany). His research interests include combinatorial algorithms, computer mathematics and educational improvements.

PLACE PHOTO HERE

B ojan Mauser received the M.Sc.degree in mechanical engineering from Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb in 2004. He worked as system engineer and IT specialist. Currently works at CARNet – Croatian Academic and Research Network as web programmer. His main interests are web programming and internet technologies.

PLACE PHOTO HERE

I vica Nakić received the M.Sc. degree in mathematics from University of Zagreb, Zagreb, Croatia, in 1998 and his Ph.D. degrees in mathematics from Fernuniversität Hagen, Hagen, Germany in 2003. He is currently Assistant Professor at the Department of Mathematics, University of Croatia. His research interests include control theory, optimization of vibrational systems and quantum control.