

Virtual Tools for Remote Analysis and Simulation

Carlos Andrés Ramos Paja and José Miguel Ramírez Scarpetta

Abstract—This document describes a general structure for the implementation of virtual tools for remote calculation through Internet. Two virtual tools for mathematical analysis (Analys) and simulation of dynamic systems (SimWeb), supported by the Matlab©, are described. These virtual tools, permit access to the Matlab from multiple work stations, without distance, power computer restrictions or user's knowledge in Matlab language; in a similar way, they allow the simulations of block diagrams through a friendly user interface. The remote calculation tools have low cost of implementation, allowing to obtain a system of high calculation power and accessible for a great number of users. The tools are oriented to automatic control systems analysis and simulation, but their use is expandable to other fields of engineering and basic sciences.

Index Terms—Analysis and simulation through Internet, remote calculation, remote execution of processes, engineering education.

I. INTRODUCTION

One of the main problems in education and research in engineering, is the access to powerful calculation tools for diverse reasons: The process capacity required for the workstations, the number of users who require access to the tools, the costs in licenses of specialized software, the geographic distribution of the working teams, etc. An emergent solution for this type of problems, consists of **virtual tools for remote calculation**, which allow the access to powerful calculation systems through a computer network, independent of workstations characteristics and of geographical distribution of the same ones. This document propose a general structure for the implementation of virtual tools for remote calculation; besides, two tools developed for analysis and simulation through Internet are introduced.

The next section presents a general structure for implementation of virtual tools for remote calculation, supported by the **Matlab©** calculation engine. After that, a virtual tool (called **Analys**) for mathematical analysis is presented. It allows the access to the Matlab command, in command line form. subsequently, a virtual tool (called **SimWeb**) for simulation of block diagrams is introduced, similar to the program **Simulink©** of Matlab; these virtual tools are accessible through a Web Browser. Finally, some conclusions and projects for further works are presented.

This work was supported by the Universidad del Valle.

C.A. Ramos is with Electrical and Electronic Engineering school, Universidad del Valle, Ciudad Universitaria - Melendez A.A 25360, Colombia caramos@univalle.edu.co

J.M. Ramírez is with Electrical and Electronic Engineering school, Universidad del Valle, Ciudad Universitaria - Melendez A.A 35360, Colombia, jomiram@univalle.edu.co

II. GENERAL STRUCTURE FOR REMOTE CALCULATION

There are multiple implementations of calculation tools guided to engineering education. These developments can be classified in three types:

- *Virtual laboratories*: The virtual laboratories allow the remote access to control and experimentation systems; some successful examples can be found in [1], [2] and [3]. These examples present computation systems handle through Web browsers, but the configuration level is minimum.
- *Learning systems*: The learning systems are guided to offer conceptual environments and mathematical tools to the students, in [4] and [5], some examples of Web-based implementations are observed.
- *Web-based simulations*: Some examples of Web-based simulators are in [6], [7], [8], [9], [10], [11] and [12]. These examples present client-server architectures, in which the clients access calculation systems with minimum configuration levels and with specific objectives.

With the objective of guaranteeing quality and precision of the calculation in the virtual tools, the software Matlab was chosen as a calculation engine, due to its extensive utilization in the scientist and educational fields. To carry out the interaction between the calculation engine and the Web, the *MATLAB Web Server ToolBox* is required; it addresses the information from a *HTML form* to the calculation engine and sends the results to the user Web Browser (additional information available in [13]).

In Figure 1, it is presented a general structure for the implementation of remote calculation tools with Matlab Web Server [13]. This consists of four main parts:

- *Web Pages*: The user interface is carried out in a Web environment developed with Web pages, figures, tables, etc. The results are also presented in Web pages; this characteristic allows the user to store or print the results locally. For the designers and programmers of calculations tools, is it possible to develop interfaces that are easy to use, without demanding the knowledge of Matlab language.
- *Web Server*: To access the user interface, it is necessary to have a Web Server, which carries out the communication between the user's Web Browser and Matlab Web Server ToolBox.
- *Matlab Web Server ToolBox*: As it was mentioned previously, Matlab Web Server sends requests from the user's Web Browser (through the Web Server) to Matlab, in the same way Matlab sends the results to the Web Browser. The limit of simultaneous requests

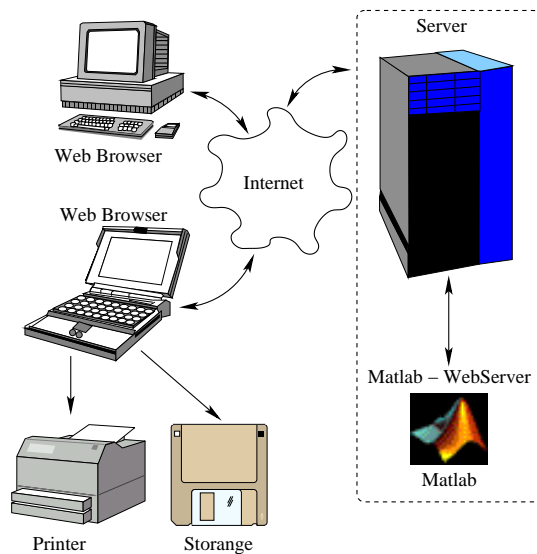


Fig. 1. General structure for the implementation of remote calculation tools with Matlab Web Server

depends on the capacity of the server; it has a system of sessions that facilitates the generation of data files for multiple users.

- **Matlab .m Programs:** Finally, the calculation engine of Matlab executes the programs *.m* to process the information sent by the user. For each available operation in the user interface, a segment of code in Matlab language should exist in a *.m* file, that produces the results, graphic and data files for the user.

Some successful implementations of calculation tools developed with Matlab are presented in [14], [15], [16], [17], [18]. These developments have two common characteristics: first, the user interfaces are built with static Web pages, limiting the possibilities to configure the calculation engine; second, the purpose of the tools is very specific, limiting their use to some few cases.

In Figure 2, is presented the general structure proposed for the implementation of virtual tools for remote calculation, oriented to develop general purpose tools.

The structure of the analysis and simulation tools is divided in two parts: the design of user interface and the design of the processing code. User interfaces are developed using HTML code and Javascript functions, allowing the interface to validate data before sending it to the server, to modify the user's options dynamically according to the operations that it selects, etc. A fundamental parameter was the minimization of the required bandwidth, with the objective of speeding up the transfer of information. This bandwidth minimization is guaranteed with three procedures:

- The information sent to the server is coded, implementing compression functions in Javascript language that are executed in the user's Web Browser; equally, decompression algorithms were implemented in Matlab language to recover the original data.
- The results are written directly as Web pages in the

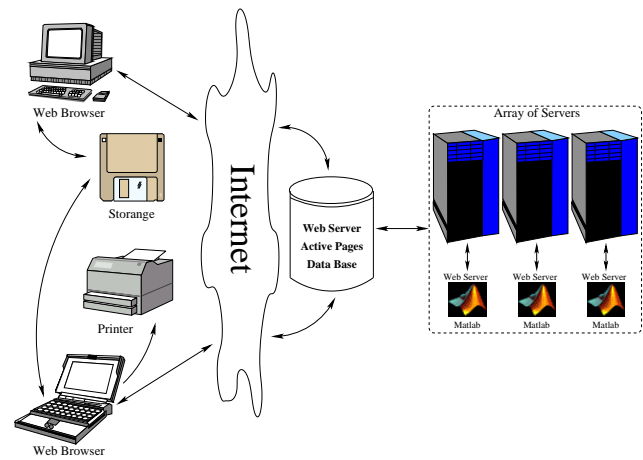


Fig. 2. General structure for the implementation of virtual tools for remote calculation

server (as HTML code), where the user can consult a standard page, with little traffic demand.

- The annexed files to the results, as graphic and data files, are available to be discharged by the user. The graphics are compressed in JPEG format and the data files are stored in 8 bit text format.

For the implementation presented in this document, the processing algorithms were developed in Matlab language, using the Control Systems ToolBox and the Symbolic Mathematics ToolBox. The users data are captured in data structures, which are decoded, processed according to the user's commands and finally the HTML, graphic and data files are generated.

The structure proposed in this document, allows to carry out complex user interfaces, using the Javascript programming language. This characteristic facilitates the development of virtual tools multi-purpose; also, the algorithms are developed in Matlab language, that let implement complex and efficient algorithms. This structure has the following implications:

- **Cost:** Matlab is a licensed package, with relatively high costs. When carrying out a implementation costs analysis of **Analys**, it is observed that it is much more economic to acquire licenses of Matlab for the server, instead of acquiring limited licenses for multiple computers; in the same way, **Analys** allows the use of Matlab from remote workstations, enlarging in this way the tool coverage, possibiliting use it in classrooms and laboratories.
- **Administration:** The centralization of the Matlab software make easy to upgrade or add new packages; moreover, add new computers for use **Analys** is trivial, since a Web Browser and a connection with the server is required.
- **Calculation power:** As a consequence of the centralization of the applications, it is convenient to use a robust and scalable server with high calculation power, in order to obtain low processing times.

Analys - Linea de Comandos

```

s=tf('s');
G=(s+1)/(s^2+3*s+5)
subplot(2,2,1)
step(G)
subplot(2,2,2)
impz(G)
subplot(2,2,3)
rlocus(G)
subplot(2,2,4)
bode(G)

```

Operacion =
Ejecutando ...

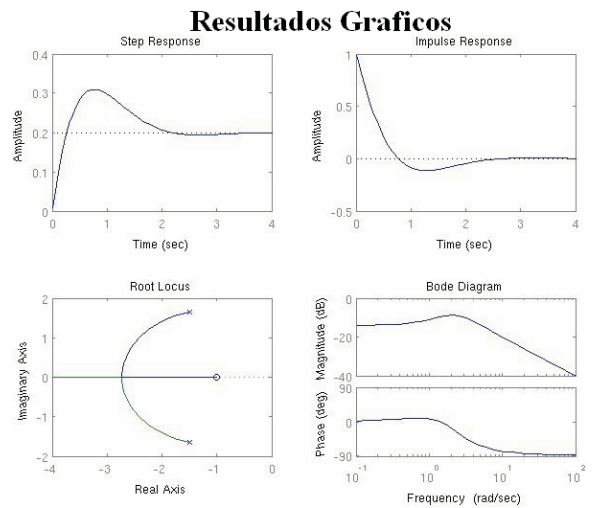
Transfer function:
s + 1

s^2 + 3 s + 5

Content-type: text/html

Ejecutar Volver

(a) **Analys** - Command line module



(b) **Analys** - Graphic results of Command line module

Fig. 3. **Analys** - Symbolic mathematics

III. VIRTUAL TOOL FOR MATHEMATICAL ANALYSIS - ANALYS

Analys is a general purpose virtual tool for mathematical analysis; it allows the remote use of great part of the available ToolBoxes of Matlab in the server. **Analys** is divided in two parts: first, **symbolic mathematics part**, conformed by the *command line module* and the *matrix algebra module*; the second part corresponds to the **automatic control systems**, conformed by the *analysis of dynamic systems module*.

A. Matrix algebra module

This module is designed to define the symbolic variables to use, the matrix and the operation to perform on them. It is not necessary to be an expert user on Matlab syntax to carry out symbolic calculations and operations with matrix. The results are presented in text format through an emergent window.

B. Command line module

With his module is possible to define a series of sentences in Matlab language to be executed sequentially, in a similar way to the standard command line of Matlab. This system allows the use of most of the functions of Matlab and its Toolbox, making it ideal for education and research purposes.

In Figure 3, an example is presented that shows several commands to create and analyze the dynamical system: $G(s) = \frac{s+1}{s^2+3s+5}$.

C. Control systems module

The control systems module is a graphic application to analyze remotely linear multivariable continuous or digital systems.

In Figure 4(a), it is presented the user interface, in which the mathematical model defined is visualized, as a transfer function (or matrix of transfer functions for multivariable

systems) and in state space representation. Here basic parameters of the systems are also presented: number of inputs/outputs, number of states, domain (continuous or discrete) and an analysis of the controlability and observability.

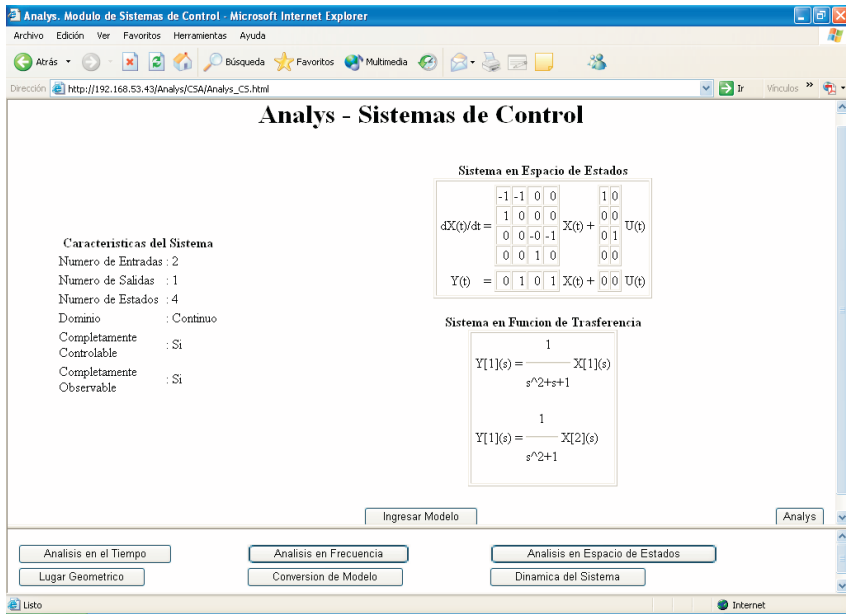
The **control systems module** allows different analysis in an intuitive way; users can define several class of models in transfer function or space of states in continuous or discrete time; the System Dynamics option, presents a summary of the most important model characteristics: bandwidth, auto-values, damping, natural frequencies, DC gain, poles, zeros and number of inputs/outputs. Models can be transformed in several canonical forms (modal, canonical controllable, canonical observable and canonical companion) by coordinate transformations [19] in the State Space Analysis option, or converted from continuous to discrete or discrete to continuous using the Model Conversion. For these models, time, frequency and root locus analysis are available; for example, the Figure 4(b) shows the response of the system described in Figure 4(a) to the user defined signal $2 \sin(3t)$.

The characteristics presented make of **Analys** an ideal tool for education and research, allowing the integration of the power and precision of Matlab with the easy access of Web applications.

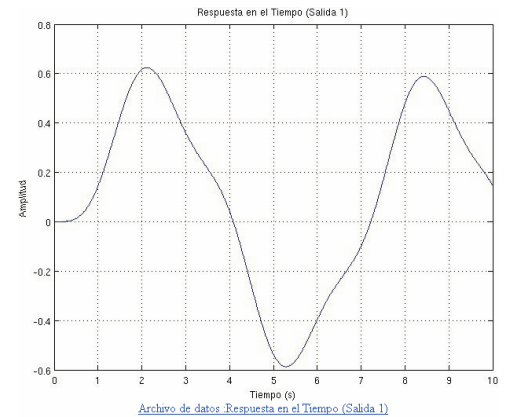
IV. VIRTUAL TOOL FOR SIMULATION OF DYNAMIC SYSTEMS - SIMWEB

SimWeb is a remote block diagram simulator that allows to design and simulate dynamical systems through Internet. **SimWeb** is similar to **Simulink**© of Matlab, but **SimWeb** is exclusively Matlab-based and doesn't need Simulink to work.

Some general purpose tools for remote simulation are based on block diagrams [20], [21], but they are developed in JAVA, being executed in the user's Web Browser, requiring high calculation capacities and a high bandwidth. **SimWeb** seeks to avoid these requirements of capacity and bandwidth.



(a) **Analys** - User interface of the control systems module



(b) **Analys** - User defined signal response

Fig. 4. **Analys** - Control systems module

SimWeb give the users the possibility to simulate linear and non-linear systems, continuous and discrete, monovariable or multivariable, represented in block diagrams, using a Web Browser and a connection with the server (local network or Internet). **SimWeb** it is divided in two parts: the graphical user interface and the execution system; the first was designed with active HTML pages and Javascript code, the second uses the calculation engine of Matlab.

A. Graphical user interface

In Figure 5(a), it is presented the graphical user interface of **SimWeb**, which is divided in three zones: blocks zone, work zone and configuration zone.

- **Blocks zone:** In this zone are the blocks to build the block diagrams. Due to the technology *Drag and Drop* implemented with the Javascript language, it is enough doing click over the blocks to select and take them to the work zone.
- **Work zone:** In this zone the block diagrams are designed. Due to the technique *Drag and Drop*, the blocks can be move freely on the screen and locate them as the user wants. In Figure 5(a), it is presented an example of block diagram, which consists on a *Continuous feedback process*, a *Zero-order hold* and two *Scopes*. The selected block is marked with a red border, indicating that the available parameters in the configuration zone belong to him. The names of the blocks are configurable for the user.
- **Configuration zone:** In this zone, is carried out the configuration of the blocks; all actions in this zone are associated to the block selected in the work zone; the available options in this zone are: name of the block, orientation of the block, number of inputs, parameters,

store the changes, user's manual, information of the block, connections tool, erase tool and simulation beginning.

- **Connections tool:** The inputs to a selected block are chosen from a menu with the names of the blocks that have outputs, see Figure 5(b).

B. Data structures

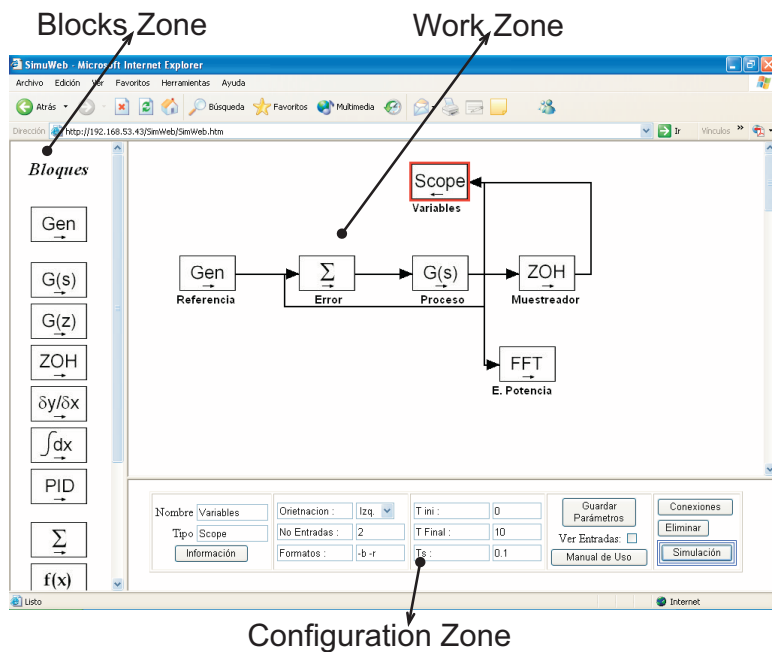
To support the *Drag and Drop* procedure, **SimWeb** uses dynamic data structures to store the block diagrams information (block types, blocks structures, connection structures and execution list) in the user's Web Browser; these structures must be created and destroyed dynamically.

To perform the requirement of *bandwidth minimization*, the information to be transmitted to the server is compressed in a HTML structure with the following parameters: first, the blocks with connections are simulated; second, only the parameters are sent, the algorithms are defined in the server; and third, the connections are described as references to the execution list, allowing a diagram of up to 256 blocks to use a single 8 bits word for address.

C. Execution environment

The block diagram simulation is carried out in the remote server where Matlab is installed; it is possible to use cluster of servers to increase the calculation power, distributing the load among more processors to reduce the processing time. It is important to highlight that due to the flexibility of the communication networks, it is possible to use **SimWeb** from portable computations systems (laptops, Pda, etc.), personal computers or Mac workstations.

The simulation procedure is implemented in Matlab language: The number of blocks to simulate is detected, the blocks connections are identified to create the execution list.

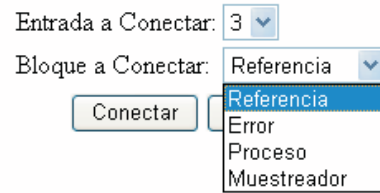


(a) SimWeb - User interface

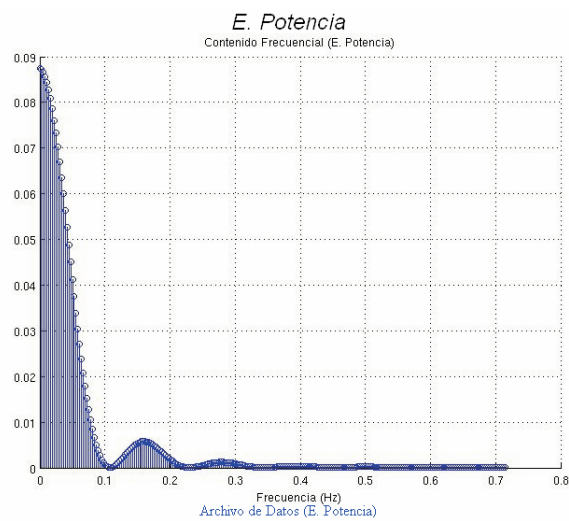
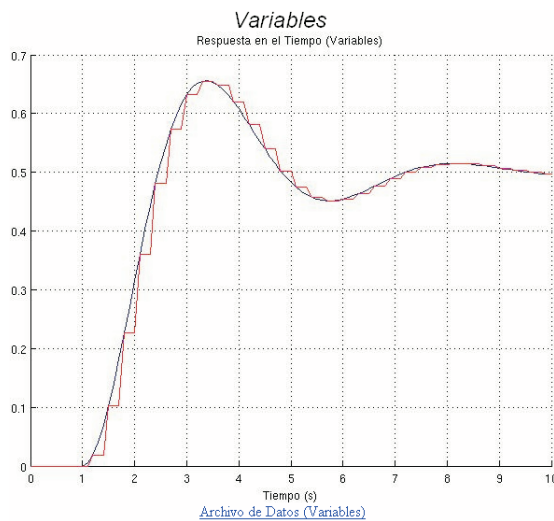
Parámetros del Bloque Variables

2 Conexión(es) Activa(s)

Entrada	Desde	Desconectar
1	Proceso	---> <---
2	Muestreador	---> <---



(b) SimWeb - Connections tool



(c) SimWeb - Simulations results

Fig. 5. SimWeb

Later on, the data structures for each block are created, creating dynamic objects in the work space of Matlab. Following the execution sequence, these objects are executed until completing the simulation. With the simulation data, data files, graphics and HTML pages for presentation of results are generated.

Finally, the system closes the Matlab session and all the dynamic objects are eliminated, recovering the memory space used. The system use dynamic memory management, allowing the attention of multiple sessions simultaneously. The graphic files and data files can be erased automatically or be stored for future analysis.

D. Implemented blocks

SimWeb has twelve defined blocks, constituted by one generating block, nine process blocks and two visualization blocks:

- *Input blocks*: Functions generator.
- *Process blocks*: Continuous dynamic system, discret dynamic system, sample and hold of zero order, derivative, integral, PID controller, summator, mathematical non-linear function, FiFo memory.
- *Scope blocks*: Time Scope, frequency scope (power spectrum).

E. Results Presentation:

SimWeb presents the simulations results in an emergent window (Figure 5(c)). The results consist (as Simulink of Matlab) in images and data files; the images are compressed in JPEG format and the files are generated in text format.

The Scope block allows the simultaneous visualization of different signals and comparing them; the block FFT plot the power spectrum of the input signal. The graphics are supported by data files links, which contain in adjacent columns the time (*Scope*) or frequency (*FFT*) and each one of the input signals; these files can be used for a later data analysis. As an example, in the block diagram of Figure 5(a), the *Scope* block is connected to the output of the *feedback process* and to the output of the *sample and hold*. Equally, the *FFT* block is connected to the output of the same *continuous process*. In Figure 5(c) the results of the simulation are presented, where it is observed that the output of the *Scope* block (left) has the continuous and sampled signals. The output of the *FFT* block presents a graphic of the output signal power spectrum.

The colors, point markers, line styles, etc., are configured by the user in vectors of formats for each *Scope* or *FFT* block; then, personalized graphics are obtained, adjusted to the user's necessities. These images can be copied directly to a text editor or stored in user's personal computer, to be used in works, reports, analysis, etc.

Finally, **SimWeb** is a versatile virtual tool for remote simulation of dynamic systems, with big possibilities of evolution and addition of blocks. It allows the generation of a powerful and robust simulation environment, where the process time depends on the capacity of calculation of the servers and not on the user's computers.

V. CONCLUSIONS AND FURTHER WORKS

The virtual tools of remote calculation presented, facilitate the access to the calculation power of Matlab from multiple work stations, without consider the distance or the power of the user's computer. These tools permits to carry out analysis of systems independently of the user's knowledge level of Matlab language; in a similar way, they allow the execution of simulations of block diagrams through a friendly user interface. These calculation tools present an additional advantage: the low cost of implementation, allowing to obtain a system of high calculation power and accessible for a great number of users, acquiring some few licenses.

The virtual tool of mathematical analysis **Analys**, allows the access to the mathematical capacities of Matlab; making it expandable to multiple areas of the knowledge; also, allows the development of graphic user interfaces to support different procedures; the authors seek to develop modules for *digital signal processing* and *neural networks*; also, is hoped to develop graphic interfaces for the basic courses of calculus and physics.

The virtual tool **SimWeb**, provides a simulation environment for dynamic systems, allowing the execution of complex and non-linear algorithms through Internet from a Web

Browser. **SimWeb** will be extended to vectorial systems, with manipulation of vectors of signals and the possibility to define and characterize the simulation algorithms.

REFERENCES

- [1] G. B. Michele Basso, "Artist: A real-time interactive simulink-based telelab," *Proceedings of the 2004 IEEE Conference on Computer Aided Control Systems Design*, Sept. 2004.
- [2] B. Wagner, "From computer-based teaching to virtual laboratories in automatic control," *29th ASEE/IEEE Frontiers in Education Conference*, Nov. 1999.
- [3] L. M. L. C. Bonivento, L. Gentili, "A web-based laboratory for control engineering education," *Dipartimento di Elettronica Informatica e Sistemistica University of Bologna*, 2002.
- [4] J. H. L. Dae Ryook Yang, "Java applet modules for undergraduate process control education," *Proceedings of the American Control Conference*, June 2003.
- [5] V. G. K. Vassilios Petridis, Spyros Kazarlis, "Aces: An interactive software platform for self-instruction and self-evaluation in automatic control systems," *IEEE Transactions on Education*, vol. 46, no. 1, Feb. 2003.
- [6] D. Z. Sheng-Jen (Tony) Hsieh, Patricia Yee Hsieh, "Web-based simulations and intelligent tutoring system for programmable logic controller," *33rd ASEE/IEEE Frontiers in Education Conference*, Nov. 2003.
- [7] L. N. Giancarlo Fortino, Domenico Grimaldi, "Distributed measurement patterns based on java and web tools," *Dipartimento di Elettronica Informatica e Sistemistica Universit della Calabria, Italy*, 2002.
- [8] P. Som Naidu, "Dynamic goal-based role-play simulation on the web: A case study," *Educational Technology & Society*, vol. 3, no. 3, 2000.
- [9] D. S. Myers, "An extensible component-based architecture for web-based simulation using standards-based web browsers," *Faculty of the Virginia Polytechnic Institute and State University*, jun 2004.
- [10] J. V. William Yurcik and L. Brumbaugh, "An interactive web-based simulation of a general computer architecture," *IEEE International Conference on Engineering and Computer Education (ICECE 2000)*, 2000.
- [11] J. S. P. G. Ernest H. Page, Robert L. Moose, "Web-based simulation in simjava using remote method invocation," *Proceedings of the 1997 Winter Simulation Conference*, 1997.
- [12] H. V. Manuel Alfonseca, Juan de Lara, "Web-based simulation of systems described by partial differential equations," *Proceedings of the 2001 Winter Simulation Conference*, 2001.
- [13] M. I. Matlab, "Mathworks home page," 2005, <http://www.mathworks.com>, consultado en 2005.
- [14] EASY-ROB, "Easy-rob 3d robot simulation tool," 2005, http://www.easy-rob.com/m2_1_2.htm.
- [15] A. I. Valeri Kroumov Keishi Shibayama, "Interactive learning tools for enhancing the education in control systems," *33rd ASEE/IEEE Frontiers in Education Conference*, Nov 2003.
- [16] O. R. C. F. R. N. Luis M. Jimenez, Rafael Puerto, "Recolab: Laboratorio remoto de control utilizando matlab y simulink," *Revista Iberoamericana de Automatica e Informatica Industrial*, vol. 2, no. 2, 2005.
- [17] M. V. Angel Valera, Jose Luis Diez and P. Albertos, "Virtual and remote control laboratory development," *IEEE Control Systems Magazine*, feb 2005.
- [18] C. A. J. V. R. G. J. D. V. Henry Jerez, Ramiro Jordan, "Web-based enabling of education and research in istec: The chipsnsalsa portal," *International Conference on Engineering Education*, aug 2002.
- [19] B. C. Kuo, *Chapters 5-9. Automatic Control Systems*, 1996.
- [20] A. S. Axel Clausen, "An internet-based computer laboratory for dsp courses," *Department of Electrical Engineering, Multidisciplinary Initiative on Distance Learning, Arizona State University*, 2000.
- [21] A. S. Thrassos Thrasyvoulou, Kostas Tsakalis, "J-dsp-c, a control systems simulation environment: Labs and assessment," *33rd ASEE/IEEE Frontiers in Education Conference*, 2003.