Modern laboratory concept for mechatronic education

Fetah Kolonić, Alen Poljugan, Alojz Slutej
Department of Electric Machines, Drives and Automation
Faculty of electrical engineering and computing, University of Zagreb
Address: Unska 3, 10 000 Zagreb, Croatia
Phone: (385) 1 6219 824 Fax: (385) 1 6129 705 E-mail:fetah.kolonic@fer.hr

Abstract - Paper describes development of mechatronic laboratory at University of Zagreb, Faculty of electrical engineering and computing. A key feature of our project is integration of a mechanical modelling, simulation and animation, control algorithm development and real-time control of mechatronic systems. Mechatronic models cover undergraduate (second course), project (seminars) graduate and research level with experimental work. Development system with one hardware/software platform and target system with real-time hardware/software support are connected in unique system. Proposed system use microprocessor of personal computer (PC) for simulation and software development as well as for real-time control. Control algorithm, designed and simulated in MATLAB/SIMULINK environment, use graphically oriented software interface for real-time code generation. This application oriented code is running under the same PC. Each mechatronic module can be used for different experiments from low to high level of difficulties. Some experiments, like crane anti-swaying control and optimal crane control algorithms are financially supported by Swedish company ABB.

I. INTRODUCTION

Tremendous growth in the field of mechatronics set a strong demand for educational institutions in order to integrate cut-edge knowledge from the different disciplines into a one. Electrical engineering, as a one of largest single academic discipline in Croatia, changed rapidly by itself and expands in its scope. Fifty years ago, an electrical engineering curriculum typically covered electromagnetics, theory of electric machines, power systems, electrotechnical materials, considerable mechanics courses, radio communications etc. Today, some of these issues disappeared or eventually form only a very small portion of an undergraduate's curriculum. Many areas such a computer engineering, control theory, artificial intelligence, coding theory, industrial communications, manufacturing automation, data acquisition and signal sensing, become integral parts of electrical engineering curriculum. At the same time, expansion of mechatronics puts a great deal of efforts on curriculum. While one of the primary objectives of this course is a laboratory experience, the most important task was how to improve existing laboratory resources in order to achieve the best possible efficiency in dynamic educational process. The previous laboratory solutions for controlling complex mechatronic systems required separate (embedded) microcontrollers based on the fast processors, mostly on DSPs, fig. 1. That concept required development system with software and hardware tools, determined by specific manufacturers (e.g. Motorola, Texas Instruments) of microprocessors, microcontrollers, DSPs etc. After application algorithms developing, testing, debugging and finally compiling, resulting code had to be downloaded in the target computers memory of the real system. If real world application program required some changes, new programs corrections had to be made on the development computer. Procedure for new load file generation is repeating again. As one can see, there are a lot of steps within a process from development to real time application running. It is time consuming and requires more software and hardware tools. From the education and research point of view, such concept is more complex then it should be. The request for simplicity, flexibility and modularity becomes stronger if laboratory concept is based on a few exercises running at the same time. This is mostly reality in the case of undergraduate curriculum.

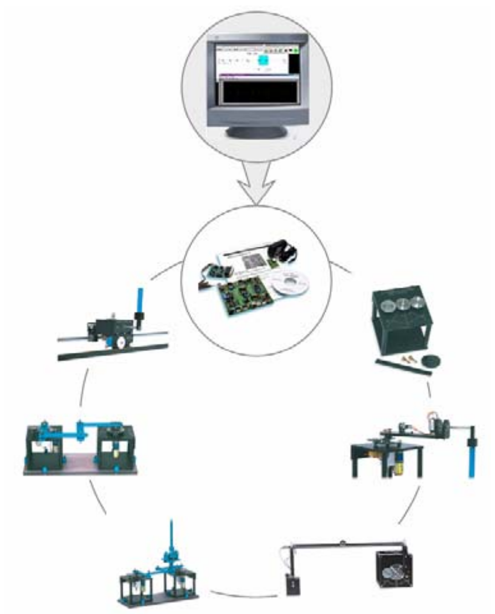
Fig.1. Laboratory model with embedded microcontroller system.

II. THE FEATURES OF PROPOSED CONCEPT

For the two mechatronics courses *Basics of Mechatronics* (VI semester) and *Mechatronic systems* (IX semester) new mechatronic laboratory is designed. The new concept is based on the four working places each one with a different electromechanical module, which can be controlled, analyzed and optimized from a personal computer. These four separate mechatronic systems are running at the same time, correlate to a four student groups in a laboratory. Instead of a separate target microcontroller for the each controlling tasks (each electromechanical module), proposed solution is based on the microprocessor of the personal computer and advanced software tools, fig.2.

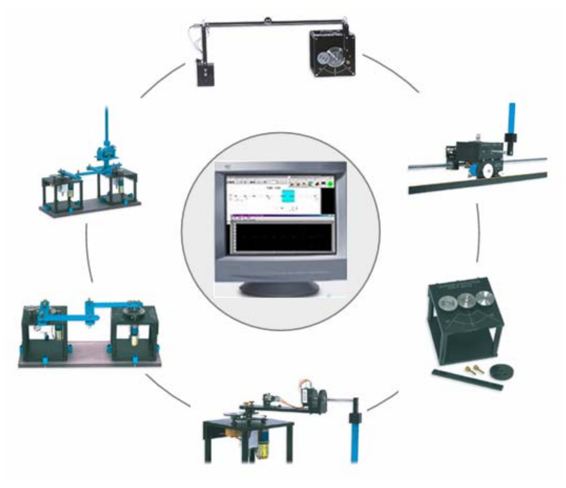

Fig.2. Laboratory model based on personal computer.

Each laboratory exercise consists of specific electromechanical plant (mechanical subsystem) controlled by PC. The user application is modelled, simulated, programmed and run on the PC. The communication with electromechanical plant is provided by a data acquisition card (DAC) mounted in PCI slot of a personal computer and terminal board, fig.3. The terminal board covers a broad range of input and output signals allowing interfacing to a variety of devices via analogue and digital signals as well as quadrature encoders. Communication between the computer and the electromechanical plant must be fast enough to ensure real time controlling of the system. This solution is based on the Windows operating system which is not real-time environment and because of that, specific optimized software tools have to be used.

III. SYSTEM COMPONENTS DESCRIPTION

Systems "hardware chain" consists of a personal computer (PC), data acquisition board (DAC), terminal board, power supply with amplifier unit (UPM) and different electromechanical plants, fig.3. There are no strong demands on PC, it should be Pentium class processor or better (the faster the better), 16 MB RAM minimum, with Windows 95/98/Me/NT/2000/XP. Terminal unit is connected to DAC board supplied with 16 differential 14 bit analogue inputs, 4 analogue 12 bit outputs, 6 optical encoder inputs, 48 programmable digital inputs. Universal power module (UPM) with +/-15V, 3A has amplifier for electromechanical plant's actuators (DC motors). Electromechanical plants are modular in construction, each one has module with rotational or translational output, [1] and [2]. Rotational module is equiped with DC motor with planetary gearbox, enkremental enkoder as a speed feedback, load antibacklash gearbox and additional mass for experiments with variable inertia load. In rotational experiments with pendulum, incremental encoder for pendulum angle measurement is added.

Translational module is a cart moving on the horizontal track. There is DC motor on the cart (the same as for rotational module) with planetary gearbox and two incremental encoders; for cart position feedback and pendulum angle measurement. These two modules are core of practically all mechatronic experiments. Other modules are coupled with basic rotational and translational accessories (pendulums, arms, gears, etc.) forming different type of experiments.

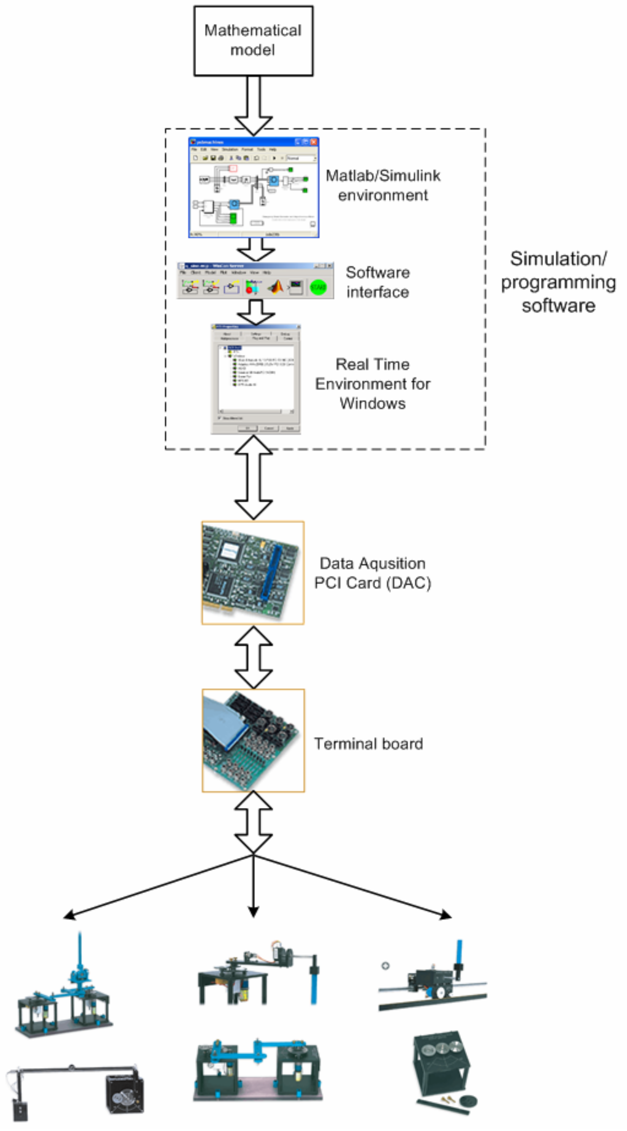
Mathematical model
Matlab/Simulink environment
Software interface
Real Time Environment for Windows
Simulation/ programming software
Data Aqusition PCI Card (DAC)
Terminal board

Fig. 3. Block diagram of laboratory real time control system

The system software core is WinCon, real-time Windows 2000/XP application, [3]. It allows running code generated from a Simulink diagram in real-time on the same PC (also known as local PC) or on a remote PC. There is no need to write code by hand. Before a Simulink model may be run in real-time, students have first to generate the real-time code in Real-Time Workshop (RTW). Changes are as easy as modifying the Simulink diagram. Data from the real-time running code may be plotted on-line in WinCon scopes and model parameters may be changed on the fly through WinCon control panels as well as Simulink. The automatically generated real-time code constitutes a stand-alone controller (i.e. independent from Simulink) and can be saved in WinCon projects together with its corresponding user-configured scopes and control panels.

IV. LOCAL AND REMOTE CONFIGURATIONS

Two possible lab configurations are supported: the local configuration (a single machine) and the remote configuration (two or more machines). In the local configuration, WinCon Client, executing the real-time code, runs on the same machine and at the same time as WinCon Server (the user-mode graphical interface), fig.4.

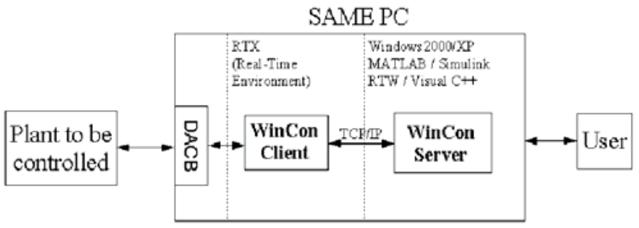

Fig.4. Local lab configuration.

The data acquisition and control board (DACB) is used to interface the real-time code to the plant to be controlled. The user interacts with the real-time code via either WinCon Server or the Simulink diagram. Data from the running controller may be plotted in real-time on the WinCon scopes and changing values on the Simulink diagram automatically change the corresponding parameters in the real-time code. It should be emphasize that running on the same PC with WinCon server, the real-time code has priority over everything else, so hard real-time performance is still achieve. Fig. 1 shows also the basic settings (local configurations) in our Mechatronics laboratory. In our laboratory 4 clients are running on the same machines with WinCon server.

In the remote configurations, WinCon Client, and hence the real-time code, runs on a separate platform than Simulink and WinCon Server (i.e. the user interface), fig.5. The minimal remote configuration is called remote configuration #1 and requires two different PC's, as for N=2 is depicted in fig.5. The two programs always communicate using the TCP/IP protocol. Each WinCon Server can communicate with several WinCon Clients, and reciprocally, each WinCon Client can communicate with several WinCon Servers.

A more general case is remote configuration #2, where N PC's are involved. Communication between the different machines is over a network (e.g. Local Area Network, Intranet, Internet). If the Internet is the intervening network, then the machines could be a world apart!

In our Mechatronics laboratory Local Area Network can be defined for connections between server and minimum 3 clients, (N=4) corresponding to number of laboratory experiment running at the same time. The advantage of these configurations is that PC's #2 to #(N-1) are not burdened by any other tasks. Thus, one can achieve the fastest possible performance in these configurations. It

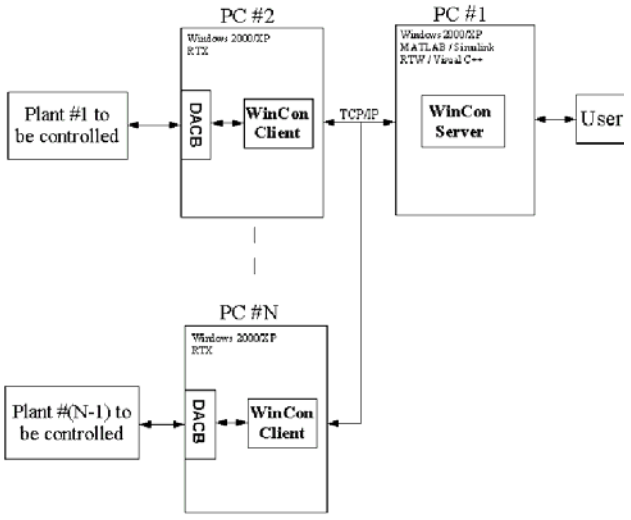

Fig.5. Remote lab configuration with N clients.

is possible, of course, to get similar performance in the local configuration because the real-time code takes priority over everything else; but for very fast sampling rates the controller would consume all the CPU time, leaving no time left for plotting or changing parameters! The remote configuration avoids this problem by moving the user interface portion to another machine. This configuration is also convenient for remote monitoring and tuning of the plants. For example, WinCon Server could be running in a remote office, while the WinCon Clients are located in a factory or laboratory setting. Finally, the most general case is remote configuration #3, which has multiple WinCon Servers and multiple WinCon Clients running as nodes on a TCP/IP network, [3].

WinCon enables students to create and control a real-time process entirely through Simulink and execute it entirely independent of Simulink. Amongst many possible configurations, students can plot real-time data and save it straight into the MATLAB workspace or to file. With Real-Time Workshop installed, student seamlessly builds C code that result in a Windows executable. After that it is run by WinCon, allowing moving beyond simulation and now control real hardware in real-time. WinCon's architecture ensures that the real-time process is afforded the highest CPU priority and is not pre-empted by any competing tasks other than the core OS functions.

V. WHAT ONE CAN RUN AND LEARN?

With basic rotational and translational plants and additional extension mechanical units (long and medium pendulum, 2D gantry module, 2D robot module, ball and beam module etc.) it is possible to run close to twenty different mechatronic experiments with different levels of difficulty. Some of them are:

- Position and speed control with rotational and translational electromechanical plants
- Ball and beam experiment with balancing the ball on the beam.
- Antipendulum control in rotational and translational moving (SISO and MIMO experiments)

- MIMO experiments with 2D gantry and 2D robot inverted pendulum
- Self erected inverted pendulum in rotational and translational moving (only SISO experiments).

There are also a variety of algorithms students can apply for each experiment:

- PID controller
- Lead and lag compensator
- Pole placement controller
- State feedback controller (generally)
- Optimal Linear Quadratic Regulator (LQR)
- Sliding mode robust control
- Fuzzy controller
- Neural Network based controller, etc.

Each experiment starts with prepared mathematical model of the specific electromechanical plant. After controller design, based on predetermined control criteria, students simulate specific electromechanical problem trying to find appropriate control solution. When the simulation gives satisfying results, the selected control algorithm can be tested on the real electromechanical system. The appropriate application code is generated and downloaded to the microprocessor of a personal computer by a specific builder (RTW) directly from Matlab/Simulink. There is no need for writing code in some programming language (C, Assembler, etc.) or opening any additional applications beside the Matlab/Simulink.

For basic course in Mechatronics, students can learn how to model electromechanical plants, i.e. to set the kinematical differential equations for specific motion using Newton's laws, D' Alambert principle or Lagrange (energy) method. Also, they become familiar with Matlab/Simulink tools, as a basic educational tool in power electronics and motion control in mechatronic systems. The choice of adequate controller and synthesis method for each specific motion is usually the main task for students. Simulation tools are unavoidable power software tools in mechatronic integration process as well as the use of LAN or/and Internet for control specific tasks remotely [4], [5]. Running in the real time and solving problems in the real world are practical experiences for students which can be used in their future professions.

VI. CONCLUSION

Mechatronic laboratory consists of specific electromechanical plants modelled, simulated, programmed and run on the PC. Using modular concept of design, close to twenty different mechatronic experiments are possible, four experiment running in the same time correlate to students group in the lab. Each experiment is designed with intention to integrate knowledge of basic mechatronics constituent fields; mechanics, electronics and information technology. Students pass through modelling, controller choice and synthesis, simulation, real-time code generation and experiment running in the real world. Also, they can investigate system identification, linear and nonlinear control, optimal control, robust and adaptive control, fuzzy and learning control as well as other control principles. Software support is realized by Matlab/Simulink and WinCon interface for real-time applications. There is no need for writing code in some programming language (C, Assembler, etc.) or opening any additional applications beside the Matlab/Simulink. With proposed lab concept, students can count on an easy-to-use, integrated environment that lets implement their designs rapidly, without lengthy coding and debugging. WinCon can communicate via the Internet allowing you to download controllers anywhere in the world and control them remotely. Besides the regular exercises, laboratory is a solid base for undergraduate, graduate and postgraduate seminars and any other additional research activities in the field of the mechatronics systems (like additional courses in automation, mechanics). Also, in collaboration with Swedish company ABB, in laboratory is continuous research in the field of crane anti-sway control and optimal crane control.

REFERENCES

[1] ….. SRV02, *Instruction Manual*, Quanser.
[2] ….. IP02, *Instruction Manual*, Quanser.
[3] ….. WinCon, *Instruction Manual*, Quanser.
[4] M.W.Spong, *The University of Illinois Control Laboratory Network*, NSF/CSS Workshop on directions in control engineering education, University of Illinois at Urbana-Champaign, 2-3 October, 1998.
[5] B. Oklay, *The Internet and engineering education, silicon and fiber replacing bricks and mortar*, NSF/CSS Workshop on directions in control engineering education, University of Illinois at Urbana-Champaign, 2-3, October, 1998.