LABORATORY-BASED AND INDUSTRIAL-ORIENTED COURSE IN MEHATRONICS

Fetah Kolonić, Alen Poljugan, Željko Jakopović
Faculty of Electrical Engineering and Computing, University of Zagreb
HR-10000 Zagreb, Unska 3, Croatia
fetah.kolonic@fer.hr, alen.poljugan@fer.hr, zeljko.jakopovic@fer.hr

Abstract: The broad educational issue in mechatronic course is how to achieve the optimum balance between narrowly defined problems with clearly defined solution and open ended problems that can have a multiple, even non-viable solutions. In the context of a mechatronic course, question has to be addressed on what technology to use, what support, instructors and teaching resources to provide, what tasks to set, how to evaluate a student performance and how engage students to create an exciting and active learning environment. In this paper mechatronic laboratory with industrial electromechanical (crane) models are used together with lectures and tutorials providing a broad and stimulating learning experience for the students. The laboratory concept is based on the different type of the PC based electromechanical models for simulation and software development as well as for real time control.

Keywords: Mechatronics education, laboratory-based, industrial-oriented, industrial crane model, real-time control, Matlab/Simulink, anti-swaying control, optimal control.

1. INTRODUCTION

How to motivate students in mechatronics course and stimulate them for more active learning and experimentation? How deep can we push them into problems and approach them this very interesting, emerging field? What technology, hardware equipment and software support, we need for that? And, at the end, if we have all answers on these questions, do we have financial and teaching resources for all of that?

The old mechatronics courses in undergraduate *Fundamentals of Mechatronics* and graduate *Mechatronic systems* courses on Faculty of Electrical Engineering and Computing in University of Zagreb, Croatia, were based on standard lectures, combined with laboratory exercises. For undergraduate study, experimental student active work in laboratory was minimal. With 70 students in course and limited technological and teaching resource, laboratory exercises are mainly based on instructor demonstrations on large control systems in crane applications (physical model), coordinate table, elevator drives. Active students work is activated only on PC simulation tools (Matlab/Simulink), which involved the identification and control of a "black box", failed to challenge and motivate students. In graduate course with maximum 20 students, situation was better then in fundamental course, there were smaller groups with project oriented tasks, but experimental test benches were limited. Definitely, large drives, even with relative open architecture, couldn't be just "multiplied" to enable maximum of two students per project task and active learning. Moreover, those laboratory solutions required separate development and target systems (i.e. embedded microcontrollers) based on the fast processors, mostly on DSPs. That concept required development system with software and hardware tools, determined by specific manufacturers (e.g. Motorola, Texas Instruments, Analog Devices) 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 compiled 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. 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 educational 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,[1,2,3]. This is mostly reality in the case of undergraduate curriculum.

2. CONCEPT OF NEW MECHATRONICS LABORATORY

The new concept of mechatronics laboratory 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.1. 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.2. 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 is 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 and optimized software tools have to be used.

Fig.1. Laboratory model based on PC and
different electromechanical models.

Systems "hardware chain" consists of a personal computer (PC), data acquisition board (DAC), terminal board, and power supply with amplifier unit (UPM) and different electromechanical plants, Fig.2. 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).

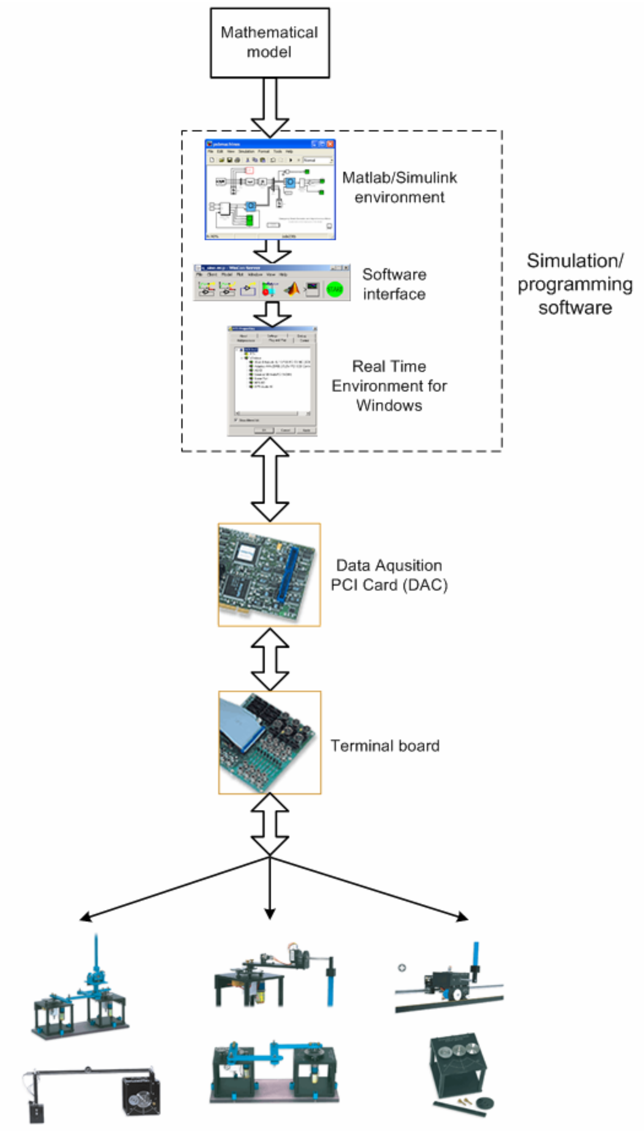


Fig. 2. Block diagram of laboratory real time control system

Electromechanical plants are modular in construction, each one has module with rotational or translational output, [4, 5]. Rotational module is equipped with DC motor with planetary gearbox, incremental encoder 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. 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)

The system software core is WinCon, real-time Windows 2000/XP application, [6]. 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.

3. STUDENTS PROJECT EXAMPLE

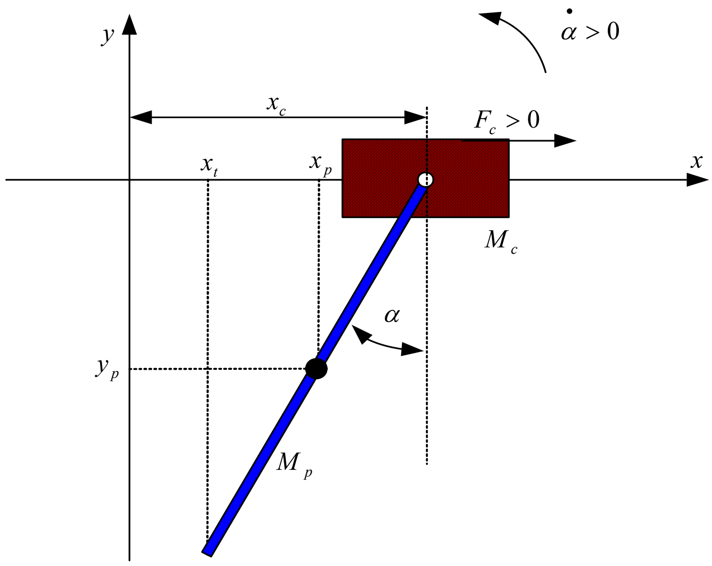
In this laboratory-based industrial-oriented course, student team has specific project for developing controller to meet defined performance specifications. It is based on physical translation crane model of industrial gantry crane, called Single Pendulum Gantry (SPG). Model is presented on Fig.3a, and schematic is indicated by the global Cartesian frame of coordinates in Fig.3b. A pendulum is suspended on the translational cart, with positive direction of displacement to the right when facing the cart.

PROJECT TASK:

For electromechanical model of the translational crane control system (Fig.3a), load position control (suspended pendulum tip) has to be controlled with state feedback controller according to the control requirements (1). The percent of the pendulum tip response overshot should be less then 5% and the settling time less then 2,2 s.

$$\begin{aligned} \sigma &= 5\% \\ t_s &= 2.2s \end{aligned} \qquad (1)$$

a)



b)

Fig.3. SPG as a physical gantry crane model a), schematic in Cartesian frame of coordinates b).

Step 1: Dynamic model derivation (NONlinear equations of motion)

In this approach, the single input to the system is considered to be F_c. The Lagrange's method is used to obtain the dynamic model of the system

$$\left(\frac{\partial}{\partial t \partial \dot{x}_c} L\right) - \left(\frac{\partial}{\partial x_c} L\right) = Q_{xc} \; ; \; \left(\frac{\partial}{\partial t \partial \dot{\alpha}} L\right) - \left(\frac{\partial}{\partial \alpha} L\right) = Q_{\alpha} \, , \qquad (2)$$

where L is Lagrangian defined through the calculation of the system's total potential and kinetic energies

$$L = T_T - V_T \qquad (3)$$

$$\frac{\partial^2}{\partial t^2} x_c = \left(\frac{\partial^2}{\partial t^2} x_c\right)(x_c, \alpha, F_c) \;\; \text{i} \;\; \frac{\partial^2}{\partial t^2} \alpha = \left(\frac{\partial^2}{\partial t^2} \alpha\right)(x_c, \alpha, F_c)$$

Solving the set of the two Lagrange's equations, gives the following two non-linear equations (4) and (5)

$$\ddot{x}_c = \frac{-(I_p + M_p l_p^2) B_{eq} \cdot \dot{x}_c + (M_p^2 l_p^3 + l_p M_p I_p) \sin(\alpha) \cdot \dot{\alpha}^2 + M_p l_p \cos(\alpha) B_p \cdot \dot{\alpha}}{(M_c + M_p) I_p + M_c M_p l_p^2 + M_p^2 l_p^2 \sin(\alpha(t))^2} + \\ + \frac{M_p^2 l_p^2 g \cos(\alpha) \sin(\alpha) + (I_p + M_p l_p^2) F_c}{(M_c + M_p) I_p + M_c M_p l_p^2 + M_p^2 l_p^2 \sin(\alpha)^2} \qquad (4)$$

$$\ddot{\alpha} = \frac{-(M_c + M_p)B_p \cdot \dot{\alpha} - M_p^2 l_p^2 \sin(\alpha)\cos(\alpha) \cdot \dot{\alpha}^2 + M_p l_p \cos(\alpha) B_{eq} \cdot \dot{x}_c}{(M_c + M_p)I_p + M_c M_p l_p^2 + M_p^2 l_p^2 \sin(\alpha)^2} + \\ + \frac{-(M_c + M_p)M_p g l_p \sin(\alpha) - F_c M_p l_p \cos(\alpha)}{(M_c + M_p)I_p + M_c M_p l_p^2 + M_p^2 l_p^2 \sin(\alpha)^2} \tag{5}$$

Step 2: Dynamic model derivation (Linear equations of motion)

In order to synthesize state feedback controller, nonlinear equations (4) i (5) are linearized around quiescent point of operation $\alpha = 0^\circ$. In the linear state space model

$$\dot{\mathbf{x}}(t) = \mathbf{A} \cdot \mathbf{x}(t) + \mathbf{b} \cdot u(t) \quad , \quad \mathbf{y}(t) = \mathbf{C} \cdot \mathbf{x}(t) + \mathbf{D} \cdot u(t), \tag{6}$$

state-space matrices with real parameters values represent a set of linear differential equations that describes the system's dynamics

$$\mathbf{x}(\mathrm{t}) = \begin{bmatrix} x_c(t) \\ \alpha(t) \\ \dot{x}_c(t) \\ \dot{\alpha}(t) \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1.5216 & -11.6513 & 0.0049 \\ 0 & -26.1093 & 26.8458 & -0.0841 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 1.5304 \\ -3.5261 \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{7}$$

Step 3: Synthesis of the PP (Pole Placement) controller, Example A

A1.) Dominant poles of the closed control loop are calculated according to relations

$$p_1 = -\zeta\omega_n + j\beta\omega_n \quad , \quad p_2 = -\zeta\omega_n - j\beta\omega_n . \tag{8}$$

Parameters ζ, β and ω_n are calculated as

$$\zeta = \frac{\ln\left|\frac{\sigma}{100}\right|}{\sqrt{\left(\ln\left(\frac{\sigma}{100}\right)\right)^2 + \pi^2}}, \quad \text{i} \quad \beta = \sqrt{1-\zeta^2}, \quad \omega_n = \frac{4}{\zeta \cdot t_s}, \tag{9}$$

and based on (8) and (9) dominant poles are $p_1 = -1.8182 + j1.9067$, $p_2 = -1.8182 - j1.9067$.

A2.) After setting two remaining closed-loop poles, p_3 and p_4, to arbitrary locations to the left of the dominating pair, p_1 and p_2, control function of state feedback controller is calculated as

$$u = -K_{pp} \tag{10}$$

where K_{pp} is vector components calculated using Matlab function "*place*"

Kpp = place(A,b,[p1 p2 p3 p4]), (11)

A3.) Simulation start for PP control design requirements

step(A-b*Kpp,Kpp(1)*b,c,d) (12)

Step 4: Synthesis of LQR (Linear Quadratic Regulator), Example B

B1.) Explanation of used control. In optimal control systems, students may set the rules for determining the control decisions. Performance index is function whose value indicates how well the actual performance of the system matches the desired performance. For linear system (6) optimal control function should be find to minimizes a quadratic performance index defined as

$$J = \frac{1}{2}\int_0^\infty (X^T Q X + u^T R u)dt\,, \tag{13}$$

where Q is symmetric nonnegative matrix and R is a symmetric positive definite matrix.

B2.) Students have to set requirements on the weighting matrices, so that the cost function makes sense. A frequent choice for matrices Q and R are $Q=C^TC$ and R= [1].

B3.) Solving ARE (Algebraic Riccati Equation)

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \tag{14}$$

and calculation control signal (gain vector) for state feedback controller

$$u = -K_{lqr}\,, \tag{15}$$

where $K = R^{-1}B^T P$. This is done with Matlab function

Klqr = lqr(A,b,Q,R). (16)

B4.) Simulation start for LQR control design requirements

step(A-b*Klqr,Klqr(1)*b,c,d). (17)

4. EXPERIMENTAL VERIFICATION

Real time (and simulation) model for experimental project evaluation is shown in Fig.4. Simulation model is performed replacing Real Crane Model box in Fig 4. with mathematical model (7). Both controllers are compared in Fig.5, with optimal parameters. During simulation process control variable is supervised to be less then limitation and if it is, iterative procedure is performed for additional controller parameters optimization.

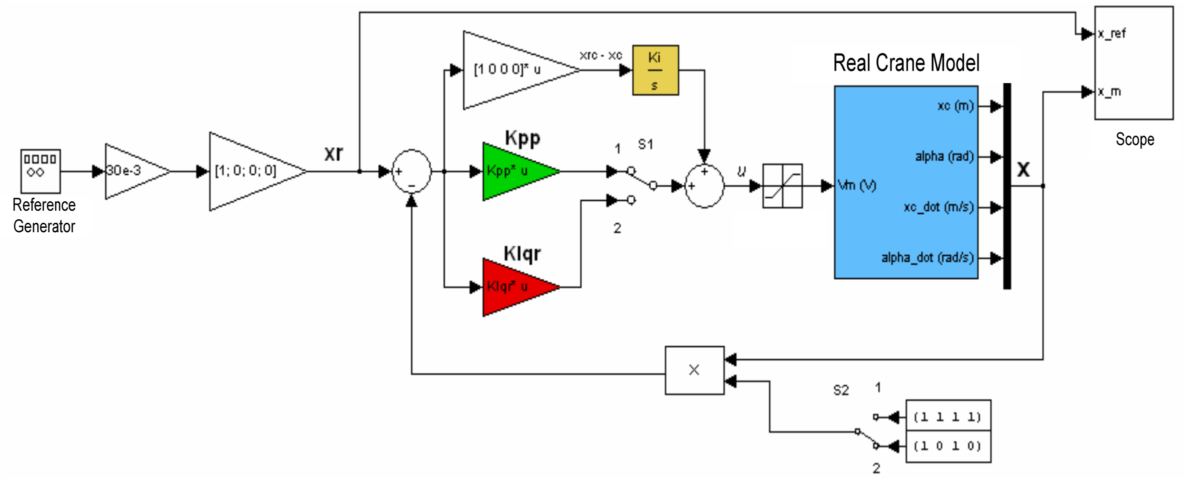


Fig.4. Model for experimental project evaluation.

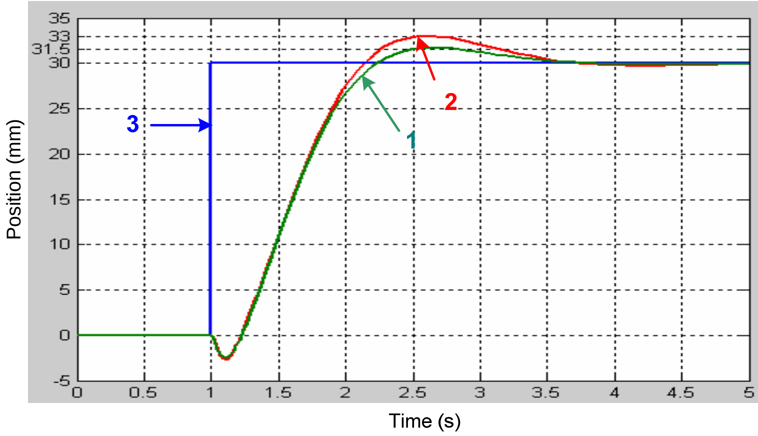
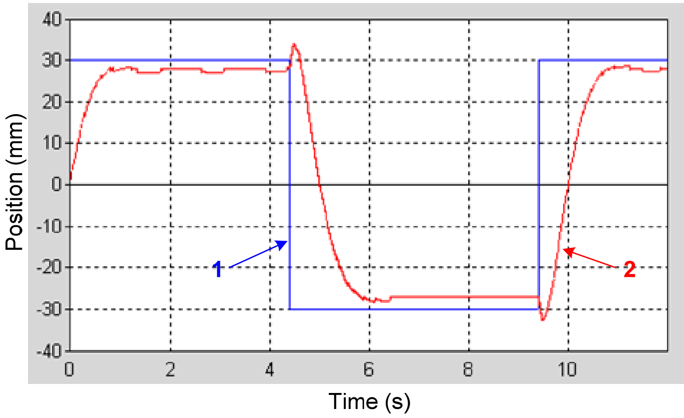
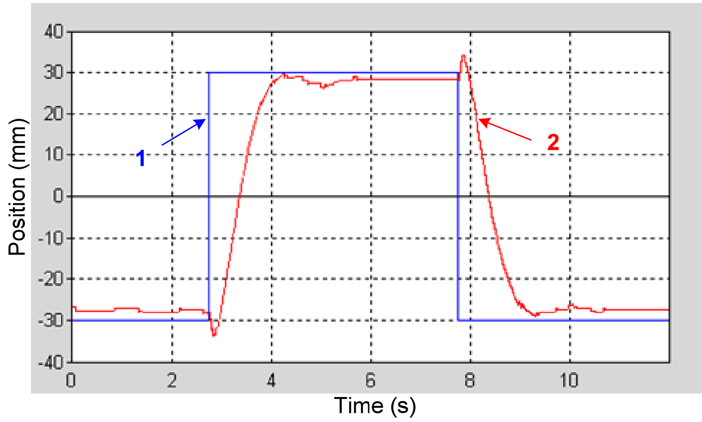


Fig.5. Reference for pendulum top position (3), position response with PP (1), and LQ controller (2); simulation results.

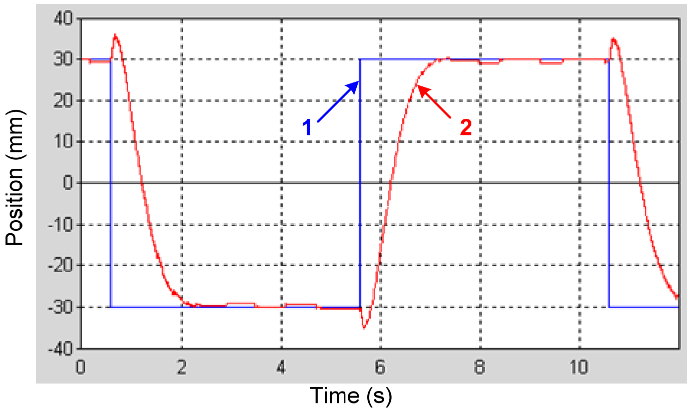


a) Reference (1) and actual pendulum top position response with PP controller (2); real time control.

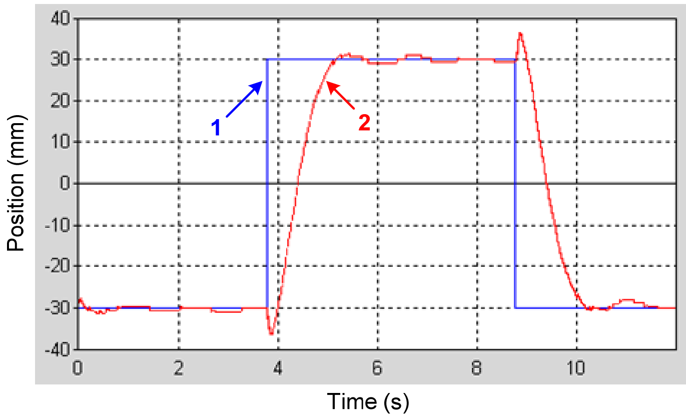


b) Reference (1) and actual pendulum top position response with LQ controller (2); real time control.

Fig.6. Pendulum (load) position control with PP and LQ controllers.



a) Reference (1) and actual pendulum top position response with PP controller and integrator (2) ; real time control.



b) Reference (1) and actual pendulum top position response with LQ controller and integrator (2); real time control.

Fig.7. Pendulum (load) position control with PP and LQ controllers and with integrator for static friction compensation.

The difference between simulation (Fig.5) and real time response (Fig.6,7) show the influence of the unmodelled friction. This step is very important from educational point of view, because it is asked from student to recognize problem and to find and apply the adequate solution(s). It is possible to make a few iterations until final satisfactory result is reached. They have to suggest another possible solution too.

1. Questions:

- Why state feedback control is effective control method? What are limitations of such control? Can you give explanation based on your experimental work?
- What is the cause of steady state pendulum top position error for step position reference? How did you eliminate this error? Can you propose some other solution than proposed one?
- Can you emphasize possible problems in LQR real world applications?
- What simplifications are used in crane physical model comparing with real gantry crane drive? What is the influence of each simplification? How does it affect on control performance? Can we get at all predefined control performance in the real crane drive using PP and LQR?

4. CONCLUSION

Mechatronic education in *Mechatronic system* course is based on laboratory with industrial project-oriented course. Students pass through modelling, controller choice and synthesis, simulation, real-time code generation and experiment running in the real world. 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. 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. Interactive learning is achieved through student-team formation and discussion of design-related issues, as well as through the encouragement of critical thinking throughout the course. Each group present their design and most students stay and ask questions for several hours. A competition serves to further motivate the students. Definitely, the challenge line must be an issue that underlies any laboratory experiment. It is obvious that the best way to engage student is to create an exciting active learning environment, where students can face different even opposite solutions for there tasks. At the end, we were impressed with the great effort the students gave and the quality of their designs. The important conclusion we can derive is: Students who are not engaged, generally do not succeed.

5. REFERENCES

[1] J. Wikander, M. Törngren, M. Hanson. *The science and education of Mechatronics Engineering* IEEE, Robotics and Automation Magazine, pp.20-26, June, 2001.

[2] R.Giegwart. *Grasping the interdisciplinarity in Mechatronics*, IEEE, Robotics and Automation Magazine, pp.27-34, June, 2001.

[3] M. Krishnan, S.Das, S.A.Yost. *Team oriented, project-based instruction in a new Mechatronics course,* 29th ASEE/IEEE Frontiers in Education Conference, November 10-13, San Juan, Puerto Rico, 1999.

[4] ….. SRV02, *Instruction Manual*, Quanser.

[5] ….. IP02, *Instruction Manual*, Quanser.

[6] ….. WinCon, *Instruction Manual*, Quanser.