LQG/LTR CONTROLLER DESIGN FOR ROTARY INVERTED PENDULUM QUANSER REAL-TIME EXPERIMENT

Cosmin Ionete
University of Craiova, Faculty of Automation, Computers and Electronics
Department of Automation, e-mail: cosmin@automation.ucv.ro

Abstract: This experiment consists of a rigid link (pendulum) rotating in a vertical plane. The rigid link is attached to a pivot arm, which is mounted on the load shaft of a DC-motor. The pivot arm can be rotated in the horizontal plane by the DC-motor. The DC-motor is instrumented with a potentiometer. In addition, a potentiometer is mounted on the pivot arm to measure the pendulum angle. The principal objective of this experiment is to balance the pendulum in the vertical-upright position and to position the pivot arm. Since the plant has two degrees of freedom but only one actuator, the system is under-actuated and exhibits significant nonlinear behavior for large pendulum excursion.
Our purpose is to design a robust controller in order to realize a real-time control of the pendulum position using a Quanser PC board and power module and the appropriate WinCon real-time software. For the controller design is used a well-known robust method, called LQG/LTR (Linear Quadratic Gauss Ian/Loop Transfer Recovery) which implements an optimal state-feedback.
The real-time experiment is realized in the Automatic Control laboratory.

Key words: MATLAB/Simulink, WinCon software, robust control, real-time experiment

SOFTWARE DESCRIPTION OF THE EXPERIMENT

WinCon™ is a real-time Windows 98/NT/2000/XP application. It allows you to run code generated from a Simulink diagram in real-time on the same PC (also known as local PC) or on a remote PC. 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.
WinCon software actually consists of two distinct parts: *WinCon Client* and *WinCon Server*. They communicate using the TCP/IP protocol. WinCon Client runs in hard real-time while WinCon Server is a separate graphical interface, running in user mode.
WinCon supports two possible configurations: the local configuration (i.e. a single machine) and the remote configuration (i.e. 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 (i.e. the user-mode graphical interface). In the remote configuration, WinCon Client runs on a separate machine from WinCon Server. 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.
The local configuration is shown below in Figure 5. The data acquisition card, in this case the MultiQ PCI, 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 changes the corresponding parameters in the real-time code. The real-time code, i.e. WinCon Client, runs on the same PC. However, the real-time code takes precedence over everything else, so hard real-time performance is still achieved.

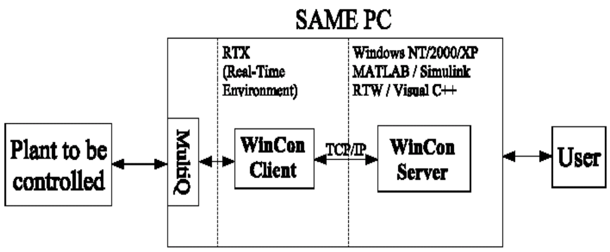

Figure 5: The local configuration

The PC running WinCon Server must have a compatible version of The MathWorks' MATLAB installed, in addition to Simulink, and the Real-Time Workshop toolbox. WinCon presently supports **MATLAB 5.3.x or 6.x** (i.e. Release 11.x or 12.x, respectively) with the corresponding Simulink (i.e. Version 3.0.x or 4.x, respectively) and Real-Time Workshop (i.e. Version 3.0.x or 4.x, respectively). Additionally, the Control System Toolbox can be useful for controller design.

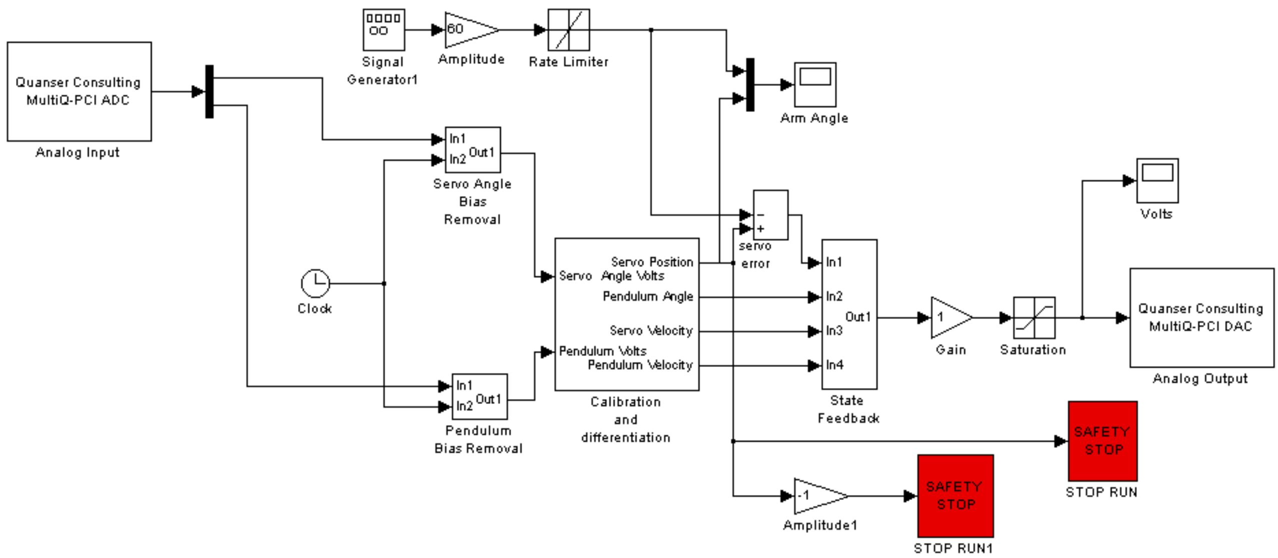

Figure 6: The Simulink model for the real-time experiment

Simulink Real-Time Application Model

In Figure 6 is presented the Simulink model for the real-time experiment. Using RTW (Real-Time Workshop) provided with MATLAB/Simulink, the WinCon software generate from this Simulink model the real-time code used for the pendulum control.
For real-time control purposes, the above model generates the reference for the pendulum position using a *Signal Generator* and a *Rate Limiter*. Thus the commands are ramped than stepped. This is because large step inputs would be too vigorous for the controller to stabilize.
Servo Angle Bias Removal and *Pendulum Bias Removal* are grouped Simulink blocks.

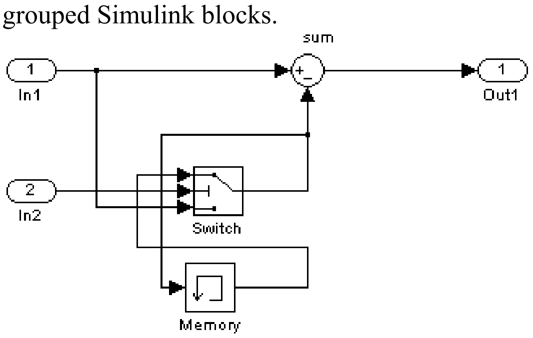

Figure 7: The bias removal block

As soon as the time is greater than a pre-defined amount of milliseconds (set in the dialog box of switch), the output of the switch is the value applied at the first line. When the switch trips over, the output of the switch is maintained to the last value before is switched over. The purpose of this block is to the user can hold the pendulum upright and the cart at x = 0. It then uses the measured bias to obtain the actual angle and cart positions.
Calibration and differentiation block converts the rotating arm position voltage and pendulum angle voltage to the appropriate units. Each signal is then differentiated to obtain the angular rates in degrees/sec.

The *State-feedback* block contains the four elements of the state feedback LQG/LTR controller that we design.

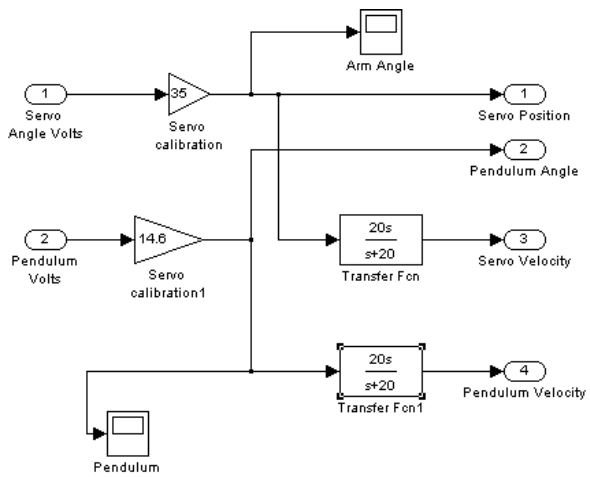

Figure 8: Calibration and differentiation block

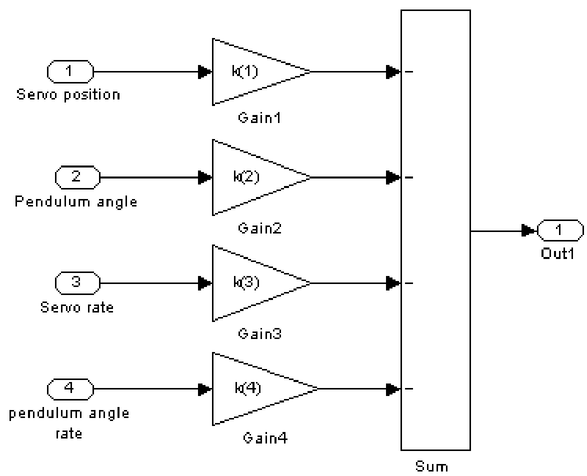

Figure 9: the State-feedback block

ROTARY INVERTED PENDULUM NONLINEAR AND LINEAR MODEL

The Rotary Inverted Pendulum module shown in Figure 10 consists of a flat arm at the end of which is a hinged potentiometer. The inverted pendulum is mounted to the hinge. Measurement of the pendulum angle is obtained using a potentiometer. The objective of the experiment is to design a feedback control system that positions the arm as well as maintains the inverted pendulum vertical. This problem is similar to the classical inverted pendulum (linear) except that the trajectory is circular.

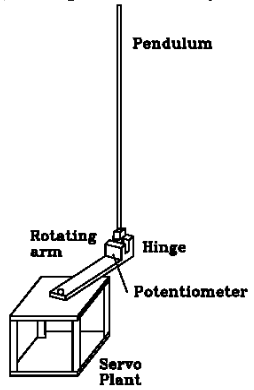

Figure 10: Schematic of Rotary Inverted Pendulum attached to a servo plant SRV02

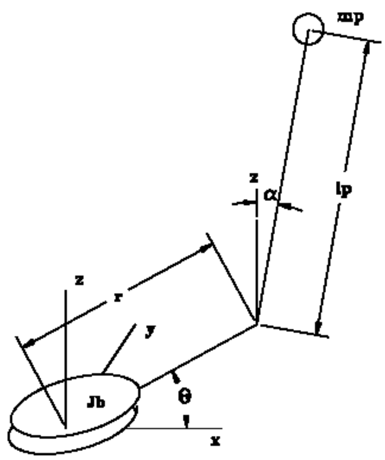

Figure 11: Simplified model for rotary inverted pendulum

Consider the simplified model in Figure 12. Note that l_p is half Lp, the actual length of the pendulum ($l_p = 0.5L_p$). The nonlinear differential equations are derived to be:

$$(m_p r^2 + J_b)\ddot{\theta} + m_p r \ddot{\alpha} l_p \cos(\alpha) - m_p r \dot{\alpha}^2 l_p \sin(\alpha) = T$$

$$m_p l_p \cos(\alpha) \ddot{\theta} r - m_p l_p \sin(\alpha) \dot{\alpha} \dot{\theta} r + m_p \ddot{\alpha} l_p^2 - m_p g l_p \sin(\alpha) = 0$$

where

T :	input torque from motor (Nm)
m_p:	mass of rod (Kg)
l_p:	centre of gravity of rod (m) (half of full length)
J_b :	Inertia of Arm and gears(Kgm)
θ:	Deflection of arm from zero position(Rad)
α :	Deflection of pendulum from vertical UP position (Rad)

The linear equations resulting from the above are:

$$\begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{m_p r g}{J_b} & -\frac{K_m^2 K_g^2}{J_b R} & 0 \\ 0 & g\frac{J_b + m_p r^2}{l_p J_b} & \frac{r K_m^2 K_g^2}{l_p J_b R} & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \alpha \\ \dot{\theta} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{J_b} \\ -\frac{r}{l_p J_b} \end{bmatrix} \tau$$

Note that the zero position for all the above equations is defined as the pendulum being vertical "up".
The motor equations are:

$$V = I_m R_m + K_m K_g \dot{\theta}$$

where

V (volts) :	Voltage applied to motor
I_m (Amp):	Current in motor
K_m (V/ (rad *sec):	Back EMF Constant
K_g :	Gear ratio in motor gearbox and external gears
θ (Rad):	Arm angular position

The torque generated by the motor is:

$$T = K_m K_g I_m = J_b \ddot{\theta}$$

$$\frac{\theta}{V} = \frac{1}{s\left(\frac{J_b R_m}{K_m K_g} s + K_m K_g\right)}$$

The linear model that was developed is based on a torque **T** applied to the arm. The actual system however is voltage driven. From the motor equations derived above we obtain:

$$T = V\frac{K_m K_g}{R} - \frac{K_m^2 K_g^2}{R}\dot{\theta}$$

Finally, we obtain the following linear model:

$$\begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{m_p r g}{J_b} & -\frac{K_m^2 K_g^2}{J_b R} & 0 \\ 0 & g\frac{J_b + m_p r^2}{l_p J_b} & \frac{r K_m^2 K_g^2}{l_p J_b R} & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \alpha \\ \dot{\theta} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{K_m K_g}{J_b R} \\ -\frac{r K_m K_g}{l_p J_b R} \end{bmatrix} V$$

LQR CONTROL FOR ROTARY INVERTED PENDULUM

Assuming the pendulum is almost upright, a state feedback controller can be implemented that would maintain it upright (and handle disturbances up to a certain point). The state feedback controller is designed using the linear quadratic regulator and the linear model of the system.

The LQG/LTR theory is a powerful method for the control of linear systems in the state-space domain. The LQR technique generates controllers with guaranteed closed loop stability robustness property even in the face of certain gain and phase variation at the plant input/output. In addition, the LQR-based controllers provide reliable closed-loop system performance despite of stochastic plant disturbance. The LQ control design framework is applicable to the class of stabilizable linear systems.

Briefly, the LQG/LTR theory says that, given a n^{th} order stabilizable system

$$\dot{x}(t) = Ax(t) + Bu(t),\ t \geq 0,\ \ x(0) = x_0$$

where $x(t) \in \Re^n$ is the state vector and $u(t) \in \Re^m$ is the input vector, determine the matrix gain $K \in \Re^{nxm}$ such that the static, full-state feedback control law

$$u(t) = -Kx(t)$$

satisfies the following criteria

a) the closed-loop system is asymptotically stable and

b) the quadratic performance functional

$$J(K) \overset{\Delta}{=} \int_0^\infty \left[x^T(t)Qx(t) + u^T(t)Ru(t)\right]dt$$

is minimized.

Q is a nonnegative-definite matrix that penalizes the departure of system states from the equilibrium and R is a positive-definite matrix that penalizes the control input. The solution of the LQR problem can be obtained via a Lagrange multiplier-based optimization technique and is given by

$$K = R^{-1}B^T P$$

where $P \in \Re^{nxn}$ is a nonnegative-definite matrix satisfying the matrix Riccati equation

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

Note that it follows that the LQR-based control design requires the availability of all state variables for feedback purpose. The state variables for the laboratory experiment are

$$x(t) = \begin{bmatrix} \theta(t) \\ \alpha(t) \\ \dot{\theta}(t) \\ \dot{\alpha}(t) \end{bmatrix}$$

For our laboratory model, the pivot arm angle θ and the pendulum angular position α are measured by two potentiometers. The pivot arm angular velocity $\dot{\theta}$ and pendulum angular velocity $\dot{\alpha}$ are not measured by any physical sensor, instead, we numerically compute $\dot{\theta}$ and $\dot{\alpha}$ by implementing a low-pass differentiator, e.g. $\frac{20s}{s+20}$, as a part of the overall control scheme.

In order to design an LQR controller, we need the plant dynamic parameters A and input matrix B.

EXPERIMENTAL RESULTS AGAINST SIMULATION RESULTS

The LQG/LTR design method uses a linear model for the real plant. Due to nonlinearities the real experimental behavior is not identical with the simulated one.

For example, in the first experiment we use the following design values:

q = diag([.5 100 .1 0]);

r = .05;

k=[-0.0552 -1.1864 -0.0602 -0.1346]

As we can see, the rotary arm error is not hardly penalized comparing with the pendulum angle.

We give e square waveform reference of 30 degrees and we obtain the behavior presented in Figure 12.

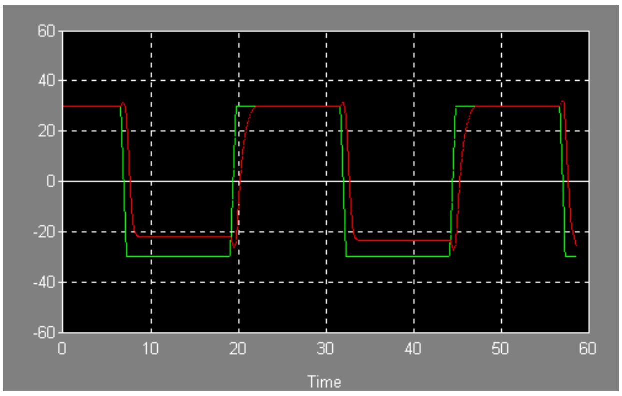

Figure 12.

We obtain a smooth command wich lead to a nonoscillant pendulum, but we have a great steady-state error.

Increasing the weight for the rotary arm error, we obtain the results from Fig. :

q = diag([20 200 .1 0]);

r = .4;

k =[-0.1234 -0.9322 -0.0681 -0.1277]

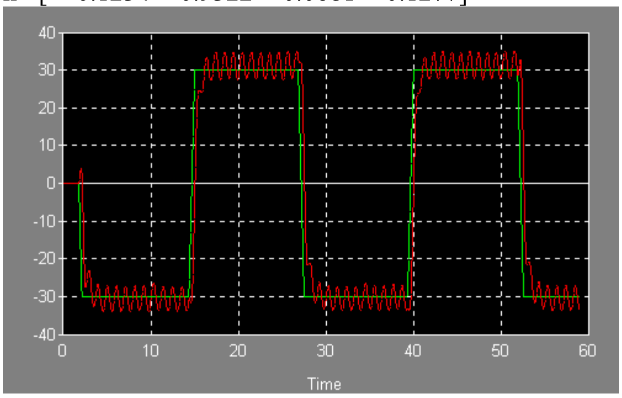

Figure 13. Experimental response

The steady-state error is eliminated but we obtain an oscilant behaviour.

It is very important to obtain an accurate model of the rotary inverted pendulum experiment. With this model we are able to simulate the evolution of the control architecture and to tune the feedback parameters in order to obtain better performances.

With the above model (saturation +/- 5V, Dead zone +/- 0.25V, backlash 0.3V) we obtain the following simulated response:

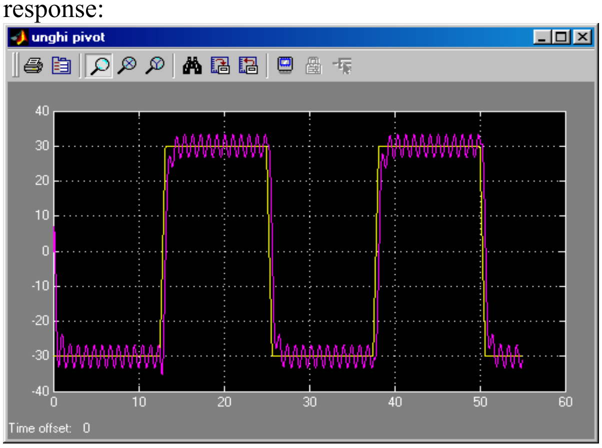

Figure 14. Simulated response

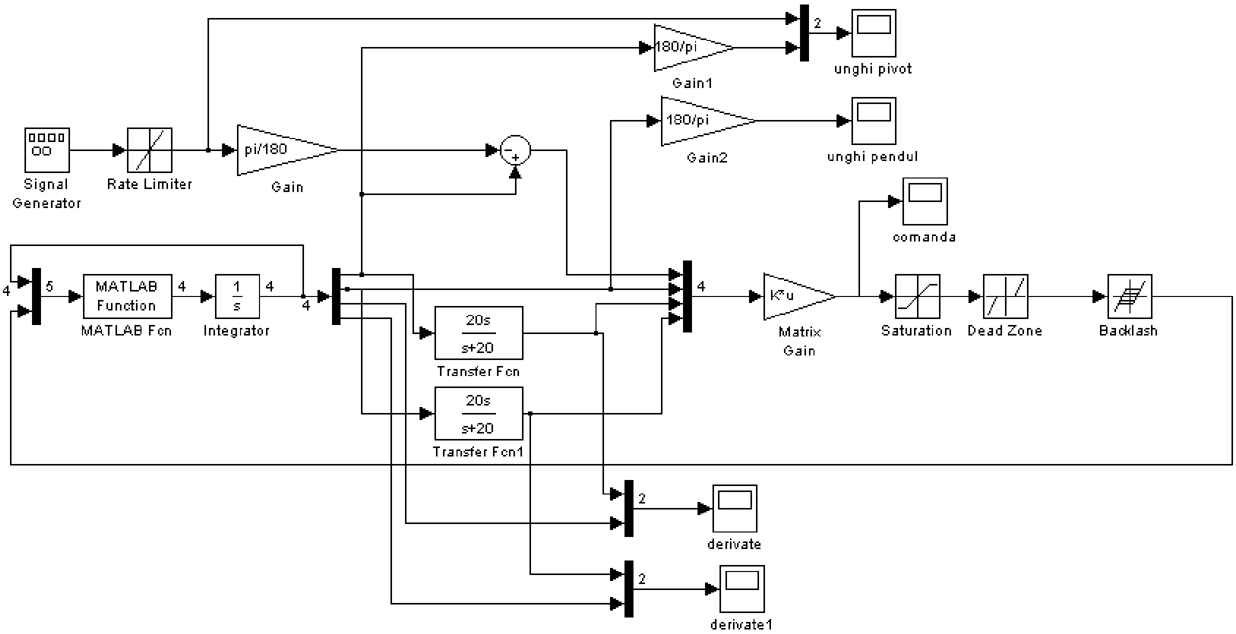

Figure 15. Simulink model of the inverted pendulum control architecture

As we can see, the experimental and simulated responses are quite similar. However, if we want to regulate precisely the pendulum position, we can introduce another state, the integral of the rotary arm error. The state vector becomes:

$$x(t) = \begin{bmatrix} \theta(t) \\ \alpha(t) \\ \dot{\theta}(t) \\ \dot{\alpha}(t) \\ \int \theta(t)dt \end{bmatrix}$$

The state feedback bloc in the control architecture is modified as follows:

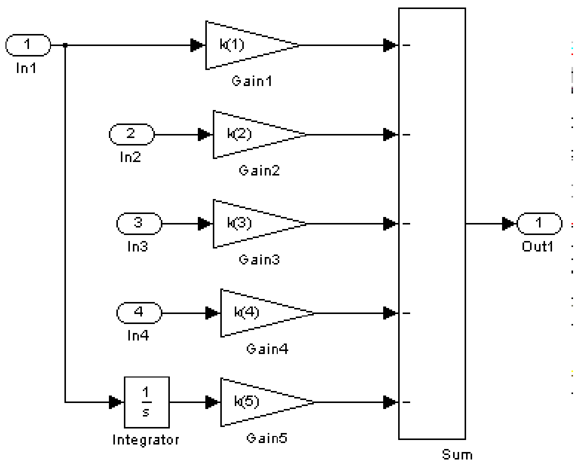

Figure 16. The state feedback block with integrator

For example, using a new state feedback, we obtain:
k =[-0.0700   -0.9000   -0.0700   -0.1000   -0.0200]

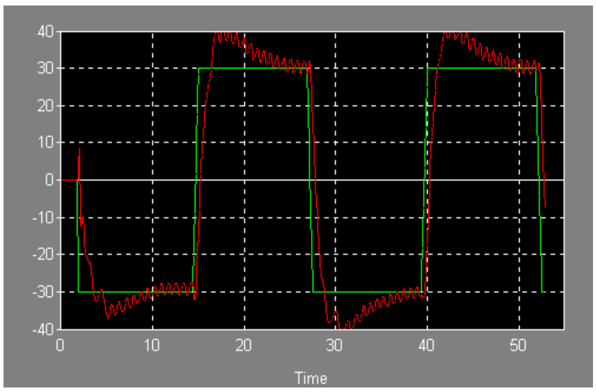

Figure 17. Experimental response with integrator

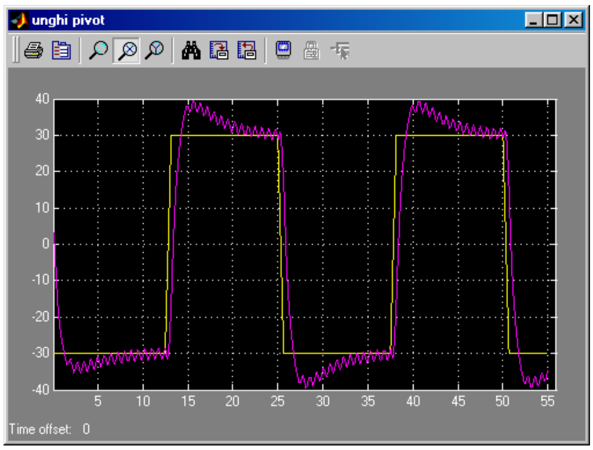

Figure 18. Simulated response with integrator

The new control can be tested against pendulum angular perturbations. The experimental results are presented in Figure 17. First graph represents the angular position and the second pendulum angle.

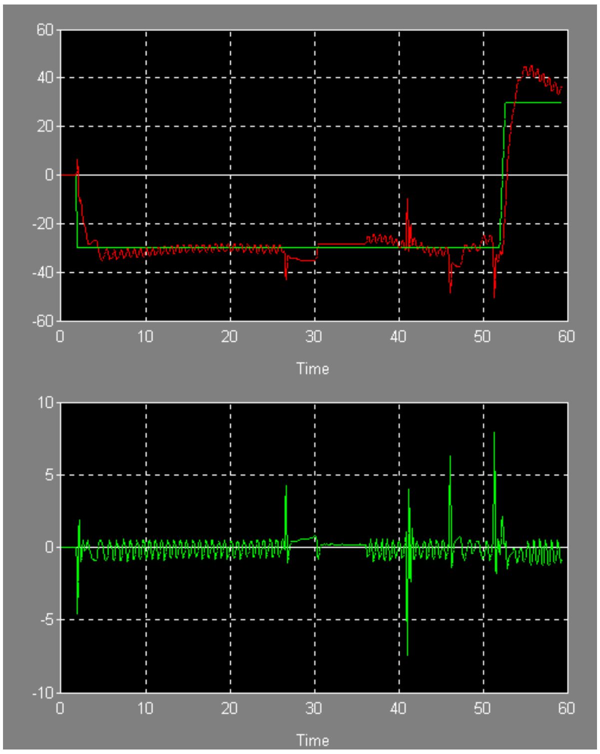

Figure 19. Perturbation rejection test

Perturbations about +/- 10 degrees can be rejected, fact that proves the robustness of the LQG/LTR design method.
Another testing set can be done with sinusoidal references. The following figures illustrate the experimental and simulated test to a sinusoidal input of 0.04 Hz with an amplitude of 30 degrees.

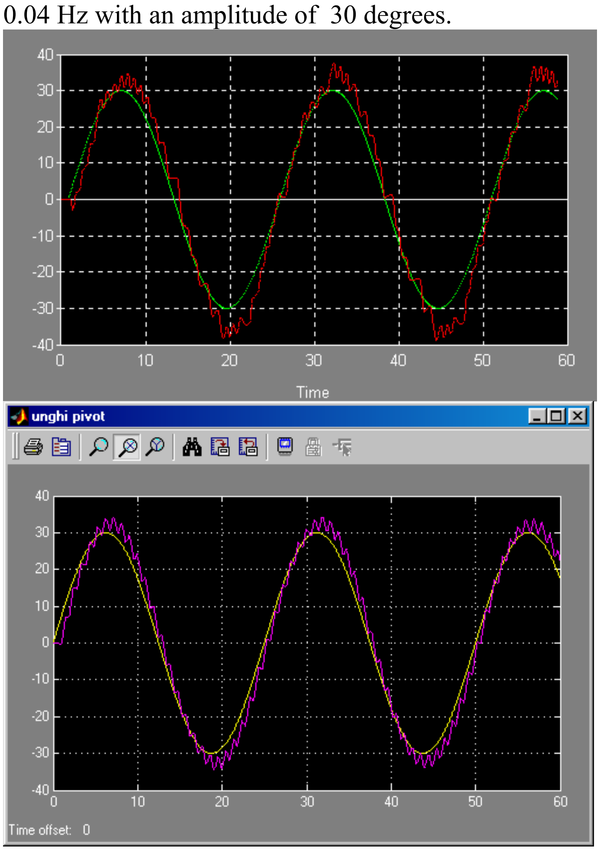

Figure 21. Experimental and simulated response to sinusoidal reference with integrator

CONCLUSIONS

In this article is presented a Quanser Consulting Inc. laboratory experiment: the Rotary Inverted Pendulum stabilization and control.
Using a single command, the DC motor voltage, we are able to control two outputs: the rotary arm angular position and pendulum verticality.
The controller design method is LQG/LTR, a robust state feedback.
Using a set of experiments is developed a nonlinear model of the rotary inverted pendulum. Using MATLAB/Simulink design environment we are able to obtain the state feedback vector in two cases: without and with an integral over the rotary arm angular error.
The experimental results are presented using Simulink and WinCon plots.

REFERENCES

[1] *** Quanser Consulting Inc., *MultiQ, Programming Manual*

[2] *** Quanser Consulting Inc., *WinCon 3.2, Real-Time Digital Signal Processing and Control under Windows 95/98 and Windows NT using Simulink and TCP/IP Technology*

[3] *** Quanser Consulting Inc., *Self Erecting Rotary Inverted Pendulum*, 1998

[3] CALIFORNIA INSTITUTE OF TECHNOLOGY, Department of Mechanical Engineering, *Analysis and Control of an Inverted Pendulum*, 1994

[4] Stein, G., M., Athans (1987) *The LQG/LTR procedure for multivariable feedback control design*, IEEE Trans. Automat. Control, AC-32, pp. 105-114

[5] R.Y. Chiang, M.G. Safonov: *"Robust Control Toolbox"*, The MathWorks, Inc., 1992.

[6] S. Skogestad, I. Postlethwaite: *"Multivariable Feedback Control, Analysis and design"*, John Willey & Sons, 1995

[7] Ivanescu M. (1994), *Industrial robots*, Ed. Universitaria, Craiova, pp. 149-186

[8] Ortega R., Spong M.W. (1989), *Adaptive motion control of rigid robots: a tutorial*, Automatica, 25, pp. 877-888.

[9] M. Gopal: *"Modern Control System Theory"*, Second Edition, John Wiley & Sons, 1993.

[10] J.C. Doyle, G. Stein: *"Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis"*, IEEE Trans. on Automat. Control, AC-26, pp. 4-16, 1981.