

Hardware Implementation of a an Optimal Pole Placement Controller for a Liquid Level System

Basil M. Al-Hadithi¹, Juan Suardiaz Muro¹, Susana Ortega Cisneros²,

Juan J. Raygoza Panduro², Juan A. López Riquelme¹

¹ División de Sistemas e Ingeniería Electrónica. Universidad Politécnica de Cartagena.
Campus Muralla del Mar. 30202 Cartagena (Murcia).
bmal@uax.es, {juan.suardiaz, jantonio.lopez}@upct.es
<http://www.dsie.upct.es>

² Departamento de Electrónica y Computación CUCEI, Universidad de Guadalajara. Blvd.
Marcelino García Barragán 1421 Guadalajara, México

{susana.ortega, juan.raygoza}@cucei.udg.mx

(Paper received on February 14, 2008, accepted on April 15, 2008)

Abstract. This paper presents a hardware implementation of an optimal pole placement controller for a liquid level system. It is designed to control the level set point by adjusting the flow rate of a liquid entering the tank through a feed pump [1][4]. The control algorithm is used to achieve optimal performance using the Integral of Time Multiplied Absolute Error (ITAE) criterion, which is characterized by little overshoot in the transient response and well-damped oscillations. The performance of the proposed controller is verified using the Xilinx System Generator modelling tool. The results show that the response of this proposed hardware implementation was good and robust implementation.

1 Introduction

This paper presents a hardware implementation of a design method which extends the basic pole placement method to discrete time systems. This method is intuitive and simple to use and can be applied equally well to either continuous or discrete time systems.

Several authors have applied the pole placement technique [9] [3] [5]. In [6][7] [8] studies were carried out to show how the pole placement controller affects the robust stability of the resulting closed-loop system.

In this paper a hardware implementation is applied using the Xilinx System Generator [11] tool to examine the robustness of the proposed controller and also to check the accuracy of the results obtained using hardware tools against those obtained by simulation. The Xilinx System Generator is a tool that extends Simulink with software and blocks for bit and cycle-accurate modelling of digital logic and DSP functions. It also has a translator that converts a Simulink model into hardware for Xilinx eld programmable gate arrays (FPGAs). Models constructed from Xilinx blocks behave exactly in the same way in Simulink as they do in hardware. As part of the

© E. V. Cuevas, M. A. Perez, D. Zaldivar, H. Sossa, R. Rojas (Eds.)
Special Issue in Electronics and Biomedical Informatics,
Computer Science and Informatics
Research in Computing Science 35, 2008, pp. 3-10



Simulink environment, these models can readily be combined with non Xilinx blocks to model parts of a system not bound to the FPGA. Xilinx blocks, like most Simulink blocks, can be customized by setting parameters from the MATLAB work space. System Generator translates into hardware only the portion of a model built from Xilinx blocks. The behaviour of the non Xilinx portion of a System Generator model can be captured on HDL test bench using simulation vectors computed in Simulink.

2 Pole Assignment Method

Let us consider a completely controllable and observable multiple inputs multiple outputs (MIMO) dynamic process with 'm' inputs, 'n' states and 'm' outputs, described by the following linear discrete-time nominal model.

$$\begin{aligned} x(k+1) &= G \cdot x(k) + H \cdot u(k) \\ y(k) &= C \cdot x(k) \end{aligned} \quad (1)$$

where

$x(k)$ = state vector.
 $u(k)$ = control signal.
 $y(k)$ = output signal.

The basic idea of the pole placement method is the design of state feedback such that all poles of the closed-loop system assume prescribed values. The control system is shown in Fig. 1a.

Applying the state feedback $u(k) = -Kx(k)$ with K being the solution of the pole placement technique according to the Ackermann's formula [3] obtained once the desired closed-loop poles are specified as shown in Fig. 1b.

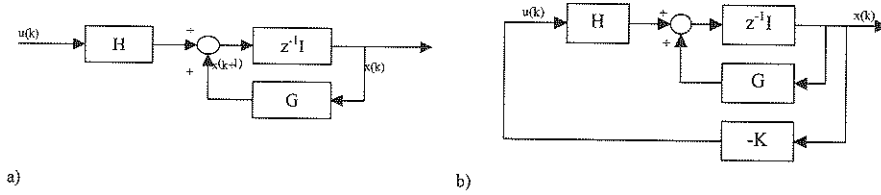


Fig. 1. (a) Control system scheme. (b) State Feedback Scheme.

Therefore the system becomes a closed loop one with the state equation as follows:

$$x(k+1) = (G - HK) \cdot x(k) \quad (2)$$

In this paper, the ITAE criterion specified by [2] is characterized by little overshoot and well-damped oscillations. This criterion possesses good selectivity.

It is however difficult to evaluate analytically, although it can readily be measured experimentally. 4

Let us consider a control system whose desired output and actual output are $x(t)$ and $y(t)$ respectively. The error is defined as:

$$e(t) = x(t) - y(t) \quad (3)$$

The optimum response one that minimizes the following Performance index is:

$$\int_0^{\infty} t|e(t)|dt \quad (4)$$

In applying the ITAE, a large initial error in a unit step is weighted lightly, while errors occurring late in the transient response are penalized heavily.

3 Control System Design and Experiments

The process consists of a pump, which is considered to be a 2nd linear system. It pumps the liquid into a tank with a uniform horizontal cross-section up to a predetermined level, and the liquid is discharged through a narrow outlet in the base as shown in Fig. 2

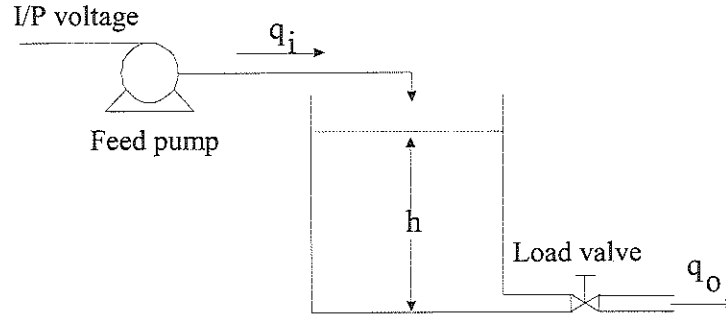


Fig. 2. Liquid Level System

The feed pump unit has been approximated to a second order linear system

$$\frac{Q_i(s)}{U(s)} = \frac{K_p}{(1 + T_1 s)(1 + T_2 s)} = \frac{1.3}{(1 + 0.36s)(1 + 0.71s)} \quad (5)$$

K_p represents the steady state gain and T_1 and T_2 represent the time constants of the pump. Their values are evaluated using the data obtained from the experimental tests. The tank is a non-linear system with the output flow rate proportional to the square root of the decreasing level. The differential equation is:

$$A \frac{dh}{dt} = q_i(t) - q_o(t) \quad (6)$$

where $A=10 \text{ cm}^2$ is the capacitance of the tank, which is defined as the change in the quantity of stored liquid necessary to cause a unit change in the potential (head). The capacitance of the tank is equal to its cross-sectional area. q_i and q_o represent the

inflow rate and outflow rate. The relationship between q_o and h can be represented as:

$$q_o(t) = k \cdot \sqrt{h(t)} \quad (7)$$

where k is related to the restriction of the valve at the tank output. Substituting (6) in (5) we obtain:

$$A \cdot \frac{dh}{dt} + k \cdot \sqrt{h(t)} = q_i(t) \quad (8)$$

Linearizing the resulting non-linear system (8) and substituting in (5):

$$\begin{aligned} \frac{d^3 h}{dt^3} + \left[\frac{A(T_1 + T_2) + \frac{k T T_2}{2\sqrt{h_0}}}{A T_1 T_2} \right] \frac{d^2 h}{dt^2} + \left[\frac{A + \frac{k}{2\sqrt{h_0}}(T_1 + T_2)}{A T_1 T_2} \right] \frac{dh}{dt} + \\ \left[\frac{k}{2\sqrt{h_0}} \right] h = \frac{[K_p u]}{A T_1 T_2} \end{aligned} \quad (9)$$

The tank was simulated on the basis of a number of experiments to find a suitable model that would coincide with the non-linear model given by (8). The area of the tank A is assumed to be 10 cm^2 . By applying (9), which represents the mathematical model of the controlled process, we get:

$$\frac{Y(s)}{U(s)} = \frac{0.2}{s^3 + 6.5s^2 + 13.5s + 8.7} \quad (10)$$

By choosing a sampling time of 0.1 seconds, the equivalent discrete process model is:

$$\frac{Y(z)}{U(z)} = \frac{0.0000205}{z^3 - 2.42z^2 + 1.949z - 0.522} \quad (11)$$

which can be described in state space form as follows:

$$G = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 2.42 & -1.949 & 0.522 \end{bmatrix} \quad H = \begin{bmatrix} 0 \\ 0 \\ 2.05 \cdot 10^{-5} \end{bmatrix} \quad (12)$$

Using the pole placement method, the desired closed loop poles are $(0.2856 \pm j0.52, 0.4938)$ and the feedback controller parameters are: $[16985 \quad -64112 \quad 66097]$.

4 Hardware Implementation

Firstly, a Simulink model of the system under control was developed as follows:

$$\begin{aligned} X1[n+1] &= X2[n] \\ X2[n+1] &= X3[n] \\ X3[n+1] &= SX1 \cdot X1[n] + SX2 \cdot X2[n] + SX3 \cdot X3[n] + K_{Reg} \cdot U[n] \end{aligned} \quad (13)$$

with the previously deduced values: $SX1 = 0.522$, $SX2 = -1.949$, $SX3 = 2.42$ and 6

$$K_{Reg} = 2.05 \cdot 10^{-5}.$$

The next design step is to compose a hardware description of the proposed controller using the Xilinx System Generator modelling tool. The response of the regulator has to obey the following rule:

$$U[n] = V_{ref} - C1 \cdot X1[n] - C2 \cdot X2[n] - C3 \cdot X3[n] \quad (14)$$

This hardware model is depicted in Fig. 3. Note that special input units are needed to represent the A/D conversion. As we will discuss later, this is the critical point of the design, because the accuracy of the real response is dependent on the selected data format.

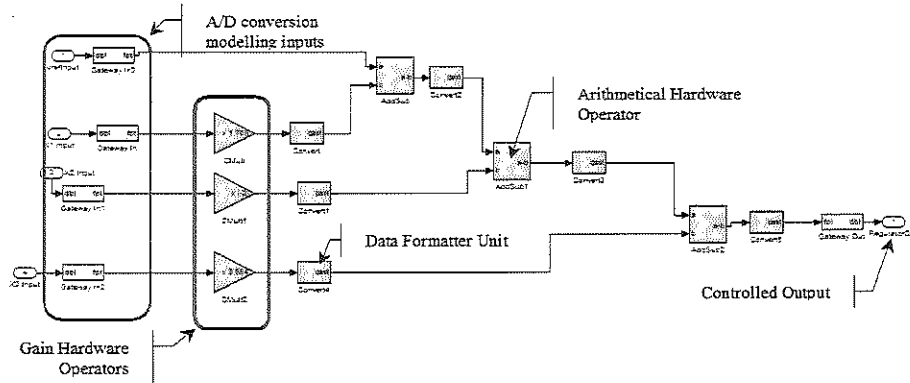


Fig. 3. Hardware model of the proposed controller.

In order to avoid data overflow when performing hardware computations, it has been assumed that there is a signal conditioner between the inputs and outputs of the hardware block and the real system, which changes the voltages levels to allow the minimization of the digital formats used by the hardware controller.

Once the hardware model is described, it can be connected to the previously defined system model and simulated in the Simulink environment.

Several simulations have been performed varying the data format to obtain the best ratio of FPGA occupancy versus response accuracy. Desired results and real results for the hardware controller diverge when a data format of 8 bits is used with 5 bits for the fractional part. If a 24-bit representation is used, with 12 bits allocated to the fractional part, both responses converge exactly. However the FPGA area consumption of inner resources is not optimized. Best results were obtained with a data format of 16 bits, using 8 for the fractional part. With this implementation, results are satisfactorily obtained and the inner resources of the FPGA are also optimized. The maximum error of this representation between the desired and real response is 0.0068, which is acceptable for the purposes of this controller.

5 Results

The performance of the proposed controller has been verified using the Xilinx System Generator modeling tool. The results show a good robust response for the proposed hardware implementation. It is shown that the precision of the results is highly influenced by the word-length. As expected, the greater the word-length, the higher was the precision. Also the resolution has been improved without the need of a large size by allocating the major length to the fractional part. The results suggest that the 16-bit fixed-point implementation, which allocates 8 bits to the fractional part achieves good performance (very similar to that offered by a 24/12 implementation) while optimizing the area consumption of the final design. Table 1 summarizes both, implementation results using a Virtex2 XC2V1000 device and the percentage of occupation for each implementation.

Table 1. Implementation results.

Resource	16_8	24_12
External IOBs	80 (48%)	120 (72%)
Slices	161 (1%)	319 (3%)
Max Error value	0.0068	0.0004

The first value, External IOBs, is a fixed value associated with the data format chosen. The controller has four inputs (X1, X2, X3 and Vref) and one output (Regulator Output). When a 16-bit format is selected, the resulting usage need is $5 \times 16 = 80$ pins. However, if the associated format is 24-bit, the new need is $5 \times 24 = 120$ pins. The second value, 'Slices' is normally used to measure the overall device usage in terms of processing elements [6]. It should be noted that the 16-bit solution takes up almost three times less area than the 24-bit solution with similar response accuracy. In any case, there is still enough area available to perform additional tasks inside the FPGA, which will be used at later to develop more dense controlling algorithms.

Finally the 'Max Error Value' measures the maxima deviation from the expected controller output. Although the 24-bit implementation presents a lower error, the 16-bit solution presents an acceptable associated error plus the advantage of optimized device occupation. Therefore, it still remains a lot of free space inside the FPGA for further improvements, or a smaller FPGA device could be used for such implementation.

The responses of the designed hardware controller are depicted in Fig.4 for a functional simulation and comparison between the continuous controller response and its associated hardware response. As it can be observed, both response superposes, as it was desired.

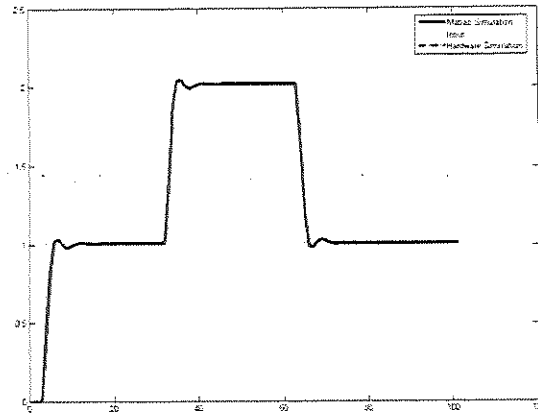


Fig. 4. Responses of the designed hardware controller to perturbations in the level input

Figure 5a shows the real response under a uncontrolled situation and Figure 5b shows the output when the FPGA based hardware controller is connected to the system. Note that the uncontrolled system presents a very slow dynamics, and when the FPGA based controlled is connected to the hardware system the closed loop dynamics is highly improved, according to the theoretical response described in section 4.

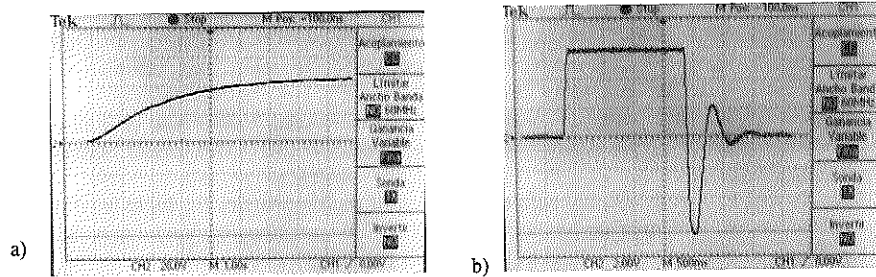


Fig. 5. Hardware responses to unitary step input. (a)No controlled (b) FPGA based control

6 Conclusions

In the hardware implementation of a pole placement control presented in this paper, the problem was to control level set point changes by adjusting the flow rate of liquid entering the tank through a feed pump. The results show a fast response with little overshoot in the transient response and well damped oscillations with zero steady state error.

The design process was optimized using the Xilinx System Generator toolbox, which allows an easy way to analyze and verify each proposed solution. And finally it was successfully implemented in a FPGA device using the ISE design environment. Optimal results were achieved using a 16 bits implementation for the proposed controller.

Acknowledgments. This work was partially supported by Seneca Program with reference ID-02998-PI-05.

References

1. Al-Hadithi, B. M., F. Matía and A. Jiménez, "Fuzzy Control for Liquid Level System," EUSFLAT, International Conference in Fuzzy Logic and Technology, Zittau, Germany, September, 2003, pp. 355-360.
2. Anderson, B. D. O. and J. B. Moore, "Optimal Control-Linear Quadratic Methods", Prentice Hall, Englewood Cliffs, NJ, (1990)
3. Garcia, G., J. Daafouz and J. Bernussou Output feedback disk pole placement with positive real uncertainty. IEEE Transactions on Automatic Control AC-41, 1385-1391 (1996).
4. Graham, B. P. and R. B. Newell, "Fuzzy identification and control of a liquid rig.," Fuzzy Sets and Systems, 1988, Vol. 26, pp. 255-273.
5. Jiandong Zhu; Yu-Ping Tian. "Pole placement and stabilization of discrete systems with unknown equilibrium points" Control Conference, 2006. CCC 2006. Aug. 2006 pp.867 - 872
6. Kyoung Kwan Ahn; Ho Pham Huy Anh. "System Identification and Self-Tuning Pole Placement Control of the Two-Axes Pneumatic Artificial Muscle Manipulator Optimized by Genetic Algorithm". International Conference on Mechatronics and Automation, 2007. ICMA 2007. Aug. 2007 pp.2604 - 2609
7. Michiels W., K. Engelborghs, P. Vansevenant and D. Roose, "Continuous pole placement for delay equations", Auto-matica, vol. 38, no. 5, pp. 747-761(15). Elsevier Science May (2002).
8. Min Xu and F. J. Kurdahi, "Accurate prediction of quality metrics for logic level designs targeted toward lookup-table-based FPGA's", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 7(4):411-418, December (1999).
9. Sen M.D. "An Algebraic Method for Pole Placement in Multi-variable Systems with Internal and External Point Delays by Using Single Rate or Multirate Sampling", Dynamics and Control vol. 10, no. 1, pp. 5-31(27), Kluwer Academic Publishers, January (2000).
10. Soylemez, M. T., "Pole Placement for Uncertain Systems. UMIST Control Systems Centre Series, Research Studies Press. Baldock, UK (1999).
11. Xilinx Inc. : <http://www.xilinx.com>