Modeling, Simulation, and Graphical Visualization of a Liquid Level Control System

Mark F. DeHerrera [*] Armando A. Rodriguez [†] Richard P. Metzger Jr [‡] Daniel Cartagena [§]

Department of Electrical Engineering
Center for System Science and Engineering
Arizona State University
Tempe, AZ 85287-7606

October 1, 1996

Keywords: Graphical visualization, control, educational software, liquid level.

Abstract

This paper describes a Windows/C_{++}-based PC environment for simulating and animating a liquid level tank under automatic control. The program consists of four modules: (i) a program user-interface module,(ii) a simulation module,(iii) a graphics/animation module, and (iv) an education module. The program-user interface module allows the user to interact with the program. More specifically, this module permits the user to select the desired liquid level model structure, model parameters, control law structure, control law parameters, reference commands, disturbances, initial conditions, integration routine, and integration routine parameters. The simulation module is responsible for generating a numerical solution for the closed loop dynamics. The graphics/animation module updates plots and animation on the screen using data generated by the simulation module. The education module provides the user with interactive lessons designed to teach fundamental systems and controls concepts. Designed to communicate with MATLAB, it is shown how this environment may be used to analyze many "what if" scenarios - observing system dynamics both graphically and through animation, making the environment a very useful tool for teaching control system design concepts.

1 Introduction

The recent revolution in personal computing has seen computing power soar while prices have dropped. As a result, powerful PC's are now commonly available. Given this, it is now possible to exploit the new technology to significantly enhance the way in which systems and controls education is delivered. Because of today's computing power [5], for example, complex simulations - not long ago considered formidable - can now be easily performed. Recent trends in computer speed have also made fast animation of dynamical systems a possibility [2]. In short, today's PC technology now permits the development of new interactive graphical visualization environments which could revolutionize the teaching of systems and controls. This opportunity, has permitted the authors to develop many such environments [3], [4], [9], [13], [15],[16], [17]. This paper describes the development of a Windows/C++ PC environment designed to simulate, visualize and teach classical control concepts related to a *Liquid Level Control System* (LLCS). Since this environment is tailored numerically and graphically to a specific system, lessons may be developed which exploit the system's specific character-

---

[*] Currently completing a Bachelor of Science in EE.

[†] This research has been supported, in part, by the National Science Foundation (NSF) through the Coalition to Increase Minority Degrees (CIMD), the ASU Center for Innovation in Engineering Education (CIEE), the Boeing A.D. Welliver Faculty Fellowship Program, and the Intel Corporation. For additional information please contact aar@asu.edu.

[‡] Currently completing a Master of Science in EE.

[§] Currently completing a Bachelor of Science in EE.

istics. Because of its simple and intuitive nature, the environment is useful for reinforcing a wide variety of systems and controls concepts. It is shown how many fundamental system and control concepts may be readily observed and understood with such an environment. The remainder of this paper is organized as follows. Section 2 describes the LLCS plant and control models available to the user. Mathematical models are discussed in this section. In Section 3, features within the liquid level environment are described. Section 4 then demonstrates the utility of the environment as an educational tool. Finally, Section 5 summarizes the paper and presents directions for future research.

2 System Description: Models

In this section the LLCS is described. The mathematical models and control systems which are available to the user are also described.

System Description. The liquid containment chamber is assumed to have a surface area A, measured in square meters. It is also assumed to have one control valve which is responsible for controlling the flow of liquid into and out of the tank. The net flow rate of liquid entering the tank will be denoted by the symbol w_{in} and is measured in cubic meters per second. Finally, it is assumed that a disturbance flow rate, denoted by the symbol w_{out} and measured in cubic meters per second, is present. This disturbance is responsible for draining liquid from and into the tank. The liquid level in the tank will be denoted by the symbol h and is measured in meters. In short, the central issue at hand is to maintain a commanded liquid level in the presence of the disturbance w_{out}. The commanded liquid level, or desired liquid level, will be denoted by the symbol r and is measured in meters.

Several mathematical models which describe the tank water-level dynamics are available to the user. Each model is now described.

First Order Model. The user may select a simple first order ordinary differential equation:

$$A\dot{h} = w_{in} - w_{out} \tag{1}$$

to describe the tank water-level dynamics.

Second Order Model. The user may also select a second order linear ordinary differential equation:

$$\frac{A\ddot{h} + A(C_2 + C_3)\dot{h} + AC_2C_3}{C_1} = w_{in} - w_{out} \tag{2}$$

to describe the tank water-level dynamics. In Equation 2, C_1,C_2, and C_3 denote user specified constants.

Actuator Model. As discussed above, the tank has a valve which is used to control the flow of liquid into and out of the tank. The valve can be described using a first order model, specified in the frequency domain as follows:

$$V(s) = \frac{a}{s + a} \tag{3}$$

where a is a user-specified parameter. A large a indicates a fast, more expensive, actuator while a small a indicates a slow, less expensive, actuator.

Liquid Level Controllers. There are three basic control laws structure which are currently available to the user:

- Proportional control (P),

$$w_{in} = k_1\ (r - h) \tag{4}$$

- Proportional plus integral control (PI), and

$$w_{in} = \frac{k_1 s + k_2}{s}\ (r - h) \tag{5}$$

- Proportional plus derivative control (PD).

$$w_{in} = [\ k_3 s + k_1\ ]\ (r - h) \tag{6}$$

where k_1 is the proportional constant, k_2 is the integral constant, and k_3 is the derivative constant [10]. Other control laws will also be made available.

3 Description of Environment

The LLCS environment is written in Windows/C_{++} [1],[6],[8],[12],[18]. It consists of four modules:

- a program user interface (PUI) module,
- a simulation module,
- a graphics/animation module, and
- an education module.

The environment was divided into modules to facilitate updates and enhancements created by different programmers.

Program User Interface Module. The PUI provides an interface between a user and the LLCS program. Written in Windows/C_{++}, the environment provides pull-down menus which permit the user to select which liquid level system model to use. Menus also permit the user to modify critical system parameters in real-time. These parameters include, for example, tank proportions, liquid density, control law constants, actuator settings, measuring device dynamics, etc. The user may also select amongst different control laws (e.g. proportional, PI and PD). Initial conditions, reference commands, disturbances, and integration routines(e.g. Euler, Runge-Kutta, etc.) may be selected by the user from a menu. Other menu options for data storage and plotting exist. Data storage routines automatically format saved simulation data for use within other environment modules as well as external programs (e.g. MathWork's MATLAB and Microsoft's Excel). User-selected variables saved from an earlier simulation may be plotted against current simulation data. Multimedia lessons, which use live audio and video will also be accessible through the PUI.

Simulation Module. The main purpose of the simulation module is to accurately solve the appropriate set of ordinary differential equations. The simulation module contains routines required by the different liquid level systems, control laws, integration methods, and data storage routines. All of the environment's liquid level models and control law routines are included in this module.

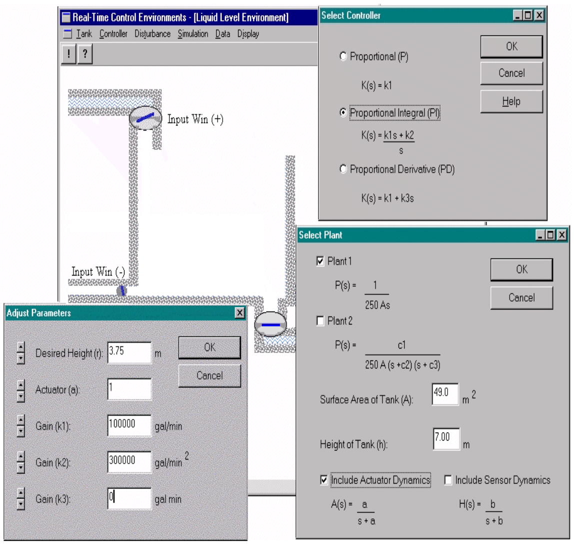
Graphics/Animation Module. The main purpose of the graphics module is to use data provided by the simulation module to update graphics (i.e. plots) and animations. Data and plots are displayed within *child windows* [7], [11]. Animation is created using high quality bitmaps generated with Corel's CorelDRAW and CorelPHOTO-PAINT [1].

Education Module. The education module contains routines which implement interactive multimedia lessons. The lessons use text, audio, video, and animation to convey ideas. Users answer questions via interactive menus. Correct answers are supported with a multimedia explanation - including "supportive audio." Incorrect answers are followed up by hints, partial explanations, and additional chances.

4 Educational Utility

The educational utility of the environment is now demonstrated by studying the effect of varying control system parameters on closed loop system performance. Through the PUI, users can select model, controller, actuator and sensor parameters. Figure 1 displays the *child windows* where the users enter LLCS parameters. In a similar fashion, controller gains and reference commands can be changed in real time via pull-down menus.

In this demonstration the affect of different controllers on such system performance characteristics as rise time, settling time, and command following is examined. The P and PI control structures were implemented using a first order model for tank liquid-level dynamics. The liquid surface area, A, was selected to be $49m^2$ and the desired liquid height was $r = 1m$. An actuator model was included, with an actuator constant $a = 0.5$.



○ Proportional (P) $K(s) = k1$
● Proportional Integral (PI) $K(s) = \frac{k1s + k2}{s}$
○ Proportional Derivative (PD) $K(s) = k1 + k3s$

☑ Plant 1 $P(s) = \frac{1}{250 As}$
☐ Plant 2 $P(s) = \frac{c1}{250 A (s + c2)(s + c3)}$
Surface Area of Tank (A): 49.0 m^2
Height of Tank (h): 7.00 m
☑ Include Actuator Dynamics $A(s) = \frac{a}{s + a}$
☐ Include Sensor Dynamics $H(s) = \frac{b}{s + b}$

Desired Height (r): 3.75 m
Actuator (a): 1
Gain (k1): 100000 gal/min
Gain (k2): 300000 gal/min ^{2}
Gain (k3): 0 gal min

Figure 1: Menus for Setting Plant and Controller Characteristics

---

[1] CorelDraw and CorelPHOTO-PAINT are trademarks of Corel Corp. CorelDraw is a general purpose drawing/graphics manipulation utility. CorelPHOTO-PAINT is a general purpose image processing utility.

Proportional Control. Figure 2 shows the closed loop liquid level response h to a step command under proportional control with different proportional control constants:

$$k_1 = 10^4, 10^5, 3 \times 10^5, 1.5 \times 10^6. \tag{7}$$

The figure shows that as the proportianl control constant k_1 increases, the rise time decreases. If k_1 gets too large, then overshoot and oscillations occur in liquid height response. This may be concluded readily be examining the closed loop transfer function from the reference command r to h:

$$T_{rh}(s) = \frac{\frac{k_1 a}{\rho A}}{s^2 + as + \frac{k_1 a}{\rho A}}. \tag{8}$$

Figure 3 shows the required control input, w_{in}. The figure shows that as k_1 increases, the required control action increases. This, however, is expected from the dependence of h on k_1 and the relationship between w_{in} and h:

$$w_{in} = \rho A \dot{h}. \tag{9}$$

Proportional Plus Integral Control. Figure 4 shows the closed loop liquid level response h to a step command under PI control for a fixed integral constant,

$$k_2 = 5 \times 10^6, \tag{10}$$

and several proportional constants:

$$k_1 = 10^4, 10^5, 3 \times 10^5, 1.5 \times 10^6. \tag{11}$$

As k_1 increases the liquid height response becomes less oscillatory, the rise time decreases, and the settling time decreases. For small k_1 values the closed loop system becomes unstable. This may be deduced via simple root locus or Routh analysis [10] of the closed loop characteristic equation:

$$\phi_{cl}(s) = s^3 + as^2 + \frac{k_2 a}{\rho A} s + \frac{k_1 a}{\rho A}. \tag{12}$$

The impact of k_1 on the control w_{in} is presented in Figure 5. Increasing k_1 causes the required control action to converge to zero more quickly.

Next, the effect of the PI integral constant on liquid level response is investigated. In figure 6 the liquid level, h, is plotted for a fixed proportional constant,

$$k_1 = 3 \times 10^5, \tag{13}$$

and several integral constants:

$$k_2 = 10^2, 10^6, 3 \times 10^6, 10^7. \tag{14}$$

As k_2 increases the closed loop response becomes more oscillatory, the rise time decreases, and the settling time increases. For large values of k_2, the closed loop system becomes unstable. Once again, this

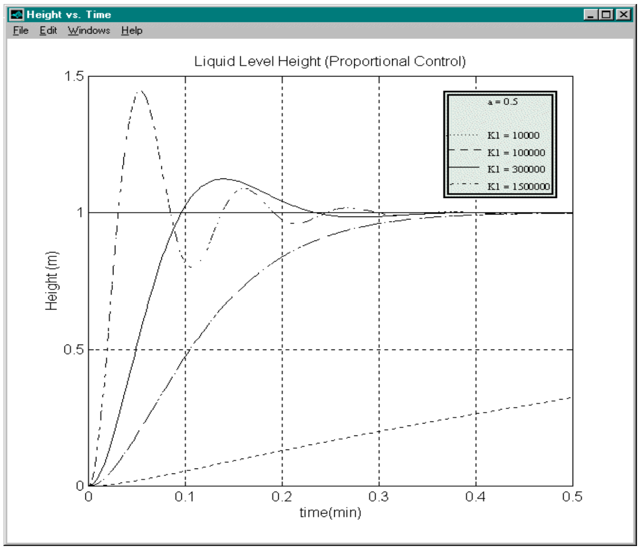


Figure 2: Output for Proportional Control - Varying Proportional Constant

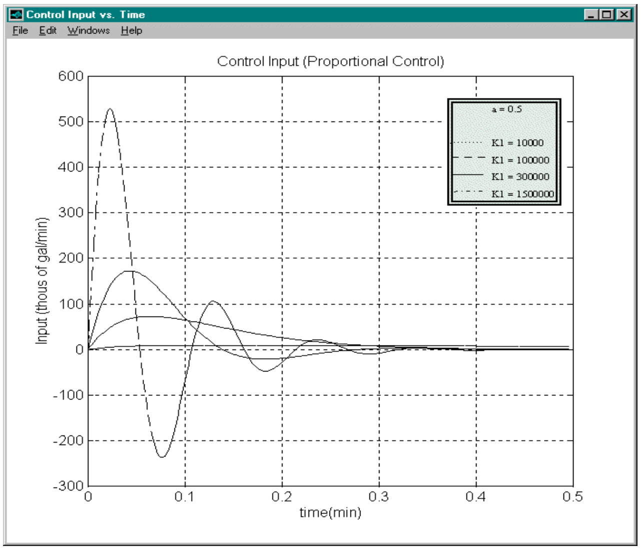


Figure 3: Input for Proportional Control - Varying Proportional Constant

follows from a root locus analysis of the closed loop characteristic polynomial $\phi_{cl}(s)$ in Equation 12. In Figure 7 the required control is plotted for the same k_1 and k_2. As k_2 increases, growing oscillations are visible in the control w_{in}.

The above simple demonstrations show how this environment may be used to visualize, understand, and develop insight into control system design concepts. The environment could be used to investigate the effect of different plant models, actuators, controllers, sensors, and associated parameters on closed loop system response.

5 Summary and Future Directions

This paper has described an environment for evaluating a liquid level tank under automatic control and teaching related controls concepts. The environment is designed so that students can experiment and develop insight into control systems in general. The simplicity of the system lends itself well to instruction on a large variety of controls related topics. Currently, help and instruct data bases are being developed to help users master the program features and provide them with instruction on relevant control concepts. Future work will include the development of interactive computer-aided-lessons (ICALs) to teach specific system and control concepts [14].

References

[1] L. Adams, *Supercharged C++ Graphics*, McGraw Hill, 1992, pp. 207-223. Teaching Simulation in Engineering Education," IEEE Transactions on Education, Vol. 35, No. 1, February 1992, pp. 50-56.

[2] R. Braham, "Math and visualization: New Tools, New Frontiers," *IEEE Spectrum,* November 1992, pp. 17-36.

[3] K.J. Elliott and A.A. Rodriguez, "A Graphical Dual Robot Simulation for the PC," *Proceedings of the 1995 International Conference on Simulation in Engineering Education*, Las Vegas, NV, January 15-18, 1995, pp. 77-81.

[4] M.F. DeHerrera and A.A. Rodriguez, "Trying to 'Shoot' an Evasive Monkey: A Tool for Designing and Evaluating Adaptive Learning Algorithms,"*Proceedings of the 1996 International Conference on Simulation in Engineering Education*, San Diego, CA, January 14-17, 1996, pp. 31-36.

[5] L. Geppert, "The New Contenders," *IEEE SPECTRUM: Fast and Powerful*, December 1993.

[6] L. Heing, *Advanced Graphics Programming Using C/C++*, John Wiley & Sons Inc., 1993, Chapter 1.

[7] L. Heiny, *Windows Graphics Programming with Borland C++*, John Wiley & Sons Inc., New York, NY, 1992.

[8] S. Holzner, *Borland C++ Windows Programming*, Brady Publishing, Indianapolis, IN, 3rd Edition, 1994.

[9] R.P. Metzger, K.J. Elliott, and A.A. Rodriguez, "Modelling, Analysis, and Graphical Visualization of a Dual Robot Arm System: A PC BasedEnvironment," *Proceedings of the 1996 International Conference on Simulation in Engineering Education*, San Diego, CA, January 14-17, 1996, pp. 175-180.

[10] K. Ogata, *Modern Control Engineering*, Prentice Hall, Englewod Cliffs, NJ, 1990.

[11] S. Oualline, *Windows Programming with Borland C++*, M&T Books, New York, NY, 1993. .

[12] S. Potts and T. S. Monk, *Borland C++ 4 By Example*, Que Corporation, Indianapolis, IN, 1994.

[13] A.A. Rodriguez and R. Aguilar, "Graphical Visualization of Missile-Target Air-to-Air Engagements: An Educational Tool for Designing and Evaluating Missile Guidance and Control Systems," *Journal of Computer Applications in Engineering Education*, Vol. 3, No. 1, 1995, pp. 5-20.

[14] A.A. Rodriguez, M.F. DeHerrera, and R.P Metzger, "An Interactive MatLab-Based Tool for Teaching Classical Systems and Controls," to appear in the Proceedings of the 1996 Conference on *Frontiers In Education,* Salt Lake City, Utah, Nov 6-9, 1996.

[15] A.A. Rodriguez and M.F. DeHerrera, "Modeling, Simulation, and Graphical Visualization of a Twin Lift Helicopter System Under AutomaticControl: An Educational Tool," to appear in the Proceedings of the 1996 Conference on *Frontiers In Education,* Salt Lake City, Utah, Nov 6-9, 1996.

[16] M. Sonne, A.A. Rodriguez, "A PC-based Graphics System for the Evaluation of Missile Guidance and Control Laws", Proceedings of the American Control Conference. Baltimore, MD. June 29-July 1, 1994, pp. 2726-2730.

[17] M. Sonne, A.A.Rodriguez, "PC's in the Design and Evaluation of Guidance and Control Systems for Missiles," Proceedings of the ICSEE. Tempe, AZ.Jan 21-23, 1994, pp. 329-334.

[18] P. Yao, *Borland C++ 4.0 Programming for Windows*, Random House Inc., New York, NY, 1994.

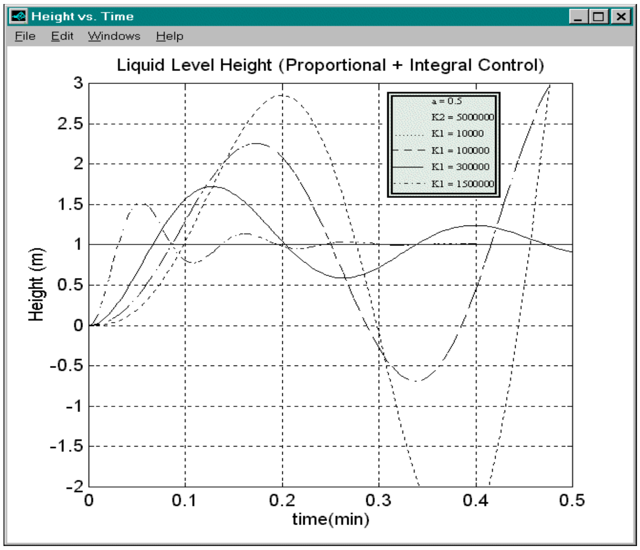


Figure 4: Output for PI Control - Varying Proportional Constant

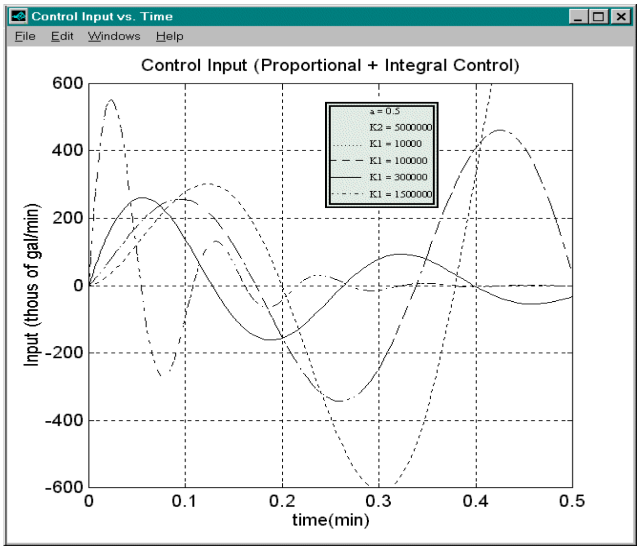


Figure 5: Input for PI Control - Varying Proportional Constant

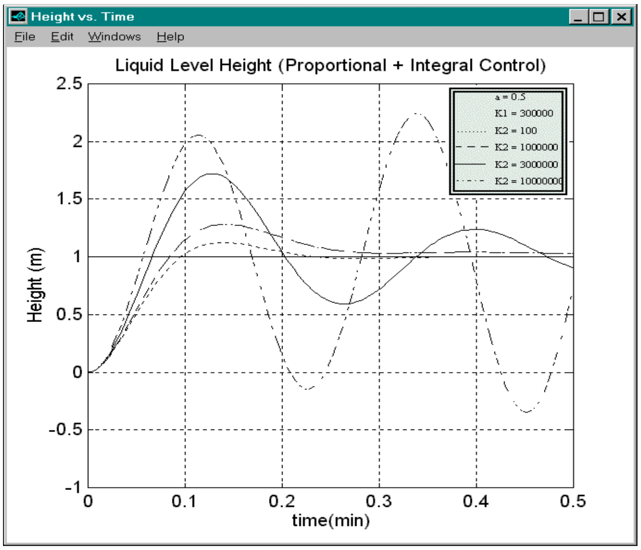


Figure 6: Output for PI Control - Varying Integral Constant

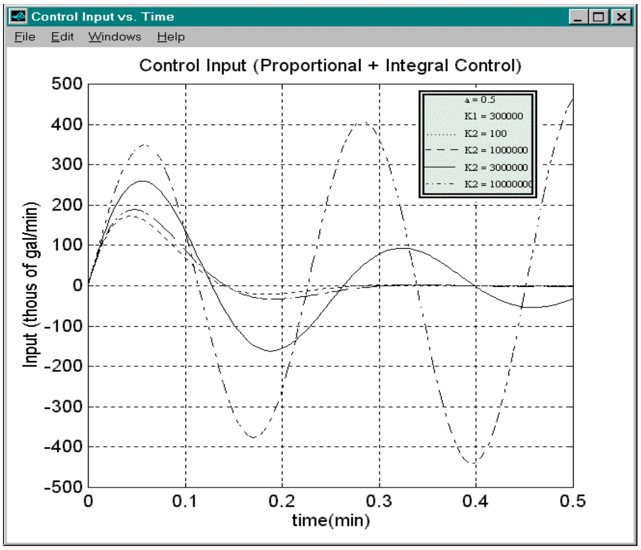


Figure 7: Input for PI Control - Varying Integral Constant