The programming of a microcontroller as an integral part of process control for undergraduate chemical engineers

Keith B. Lodge[*]
Associate Professor, Department of Chemical Engineering, the University of Minnesota Duluth,
1303 Ordean Court, Duluth, Minnesota 55812-3025

Abstract

New funding in our College of Science and Engineering has become available for the enhancement of courses with computer technology. I took this opportunity to try an experiment in the teaching of process control to chemical engineers. Inexpensive and reliable microcontrollers are now commonplace. I am using the Basic Stamp, Parallax Inc., with the intention of getting the students to build and tune their own liquid-level loops. This experiment was tried in the spring of 2002 with a class of 26 students divided into groups of 2; each group was assigned its own microcontroller and apparatus. There is no dedicated laboratory section associated with this class.

In essence the use of a microcontroller allows topics, hitherto taught purely in the classroom (sometimes with the aid of demonstrations), to become the object of direct practical experience for the students. The topics include A/D conversion, sensor calibration, self regulation, on-off control, proportional (P) control and offset, proportional and integral (PI) control, the position and velocity forms of the PID algorithm, selection of the control interval and signal aliasing, step tests and tuning. I will describe (1) the microcontroller and its programming language, a form of Basic; (2) the hardware with which the level-control loop is constructed including the design of a simple actuator; (3) the data generated and the tests performed by the students; (4) the logistics and problems encountered; and (5) how the original course was modified to include this experiment.

Introduction

Presently, Chemical Engineering at the University of Minnesota Duluth (UMD) is an undergraduate program only. The number of students graduating is typically 20-30 per annum. One or two students at the most directly enter a graduate school in chemical engineering; a few will do so after a number of years in the work place. UMD is situated about 150 miles to the NNE of the Twin Cities of Minneapolis & St. Paul at the western end of Lake Superior.

Despite its seemingly remote location, UMD has not been kept isolated from “the technology initiative”. New funding generated by the College of Science and Engineering has financed enhancements of courses with computer technology. Here I describe briefly an enhancement, or perhaps a new direction is a more apt description, of Process Control (ChE 4401); this is offered once a year in the spring semester. Typically students take this course in their last semester; by this time the novelty of life in the classroom is wearing a bit thin! For the most part, the course has roughly followed the first ten chapters of “Chemical Process Control “ (Riggs, 2001), although this text has only been used for the last three years.

It is now easy to obtain inexpensive and reliable microcontrollers. In the spring of 2002, students worked the Board of Education (BOE) from Parallax (Parallax, 2002a); this is a convenient system with which to build circuits and to program the Basic Stamp, Parallax’s microcontroller. Of course, Parallax is

---

[*] E-mail: klodge@d.umn.edu Paper submitted on July 15, 2002, to the 2002 North Midwest Section Annual Conference of the American Society for Engineering Education, October 10-13, 2002, Madison, Wisconsin.

not the sole source of microcontrollers; however, the depth and breadth of their educational material (Parallax 2002b) is particularly useful for those individuals who, like myself, are new to the field. My research interests are not in process control and I took over the course after the departure of a specialist in the field. My single advantage is being an experimentalist. My preparation started in the summer of 2001; this exercise would not have been impossible were it not for my single semester leave in the fall of 2001. During this time, I attended one of the regular two-day short courses organized by Parallax (Parallax, 2002b).

On the BOE, one assembles the desired circuit on a breadboard affixed to a printed circuit board; this contains the microcontroller, connections for power sources (wall transformer, 9V battery) and input-output (I/O) connections to the microcontroller. The BOE provides a 5 VDC power source from which transducers may be run. For the purposes here, it is not necessary to design circuits; all the exercises were done with existing circuits with some minor modifications; these include adjustments to potentiometers and adding a switch. Our students meet the basics of electricity in freshman physics, and then go further in the required electrical power course (Engr 3201), with which a lab is associated. Essentially, students have to be able to construct simple circuits from available diagrams involving the ability to locate pin 1, etc., on a chip, and recognize the polarity of an LED and an electrolytic capacitor. I encourage them to use a DVM to check their interpretations of the color code on resistors! A separate switchable power strip was used to power the BOE through its wall transformer; it is necessary to switch off the power to the BOE before making any changes to the circuit.

Communication from the BOE to the outside world is made through a serial connection to a personal computer (PC). The operating system should be Windows 95 or higher. Programs are written in a special form of Basic, PBASIC; they then complied and downloaded to the microcontroller. Data can be transmiited back to the PC and viewed in debug window (*debug* command), or plotted and saved in a file using Stamp Plot Lite (Selmaware 2002). The *debug* command is especially useful in developing programs. PBASIC contains special commands to handle data from an A-to-D converter (*shiftout*) and to run a servo motor (*pulseout*). Most of the work here is programming; our students will have taken a required programming course in the lower division. In the previous semester to Process Control, most will have taken Separations (ChE 4111); here MathCad and its programming facilities are used extensively.

Process Control is a 3-credit course meeting for three 50-min sessions per week in a 15-weeek semester. Just over a third of the class time was devoted to the work with the microcontroller; to fit this in some modifications of the curriculum were necessary. In addition to developing a more economical presentation of the subject matter, the following steps were taken:

1. Bode stability analysis was omitted.
2. The treatment of second-order systems was reduced to a bare minimum; enough was taught to explain stability, instability and quarter amplitude damping (QAD). (The discussion of first-order systems remained unchanged as did the discussion of the first-order-plus-dead-time, or FOPDT, model).
3. Various examples of Laplace transforms were omitted. The basics of Laplace transforms were kept. (Students need these for the Fundamentals-of-Engineering exam in order to become registered professional engineers in Minnesota.)
4. Modeling of dynamical systems was restricted to lumped systems.
5. The number of demonstrations that were held in the classroom was reduced.

Class Work

The class work fell naturally in two parts. The first part was devoted to formal training; the syllabus for this is shown in the Table that follows. The early classes were held in a University-run computer lab. About half way through the semester, we moved into a regular laboratory, because water was required. Because of its very nature, it is possible to make progress within a 50-min session in programming a microcontroller; this is a rare advantage for practical work and makes this an ideal exercise for “hands-on” learning. The aim of the first part was to give the students all the tools necessary to do the work on the liquid-level control loop, the second part; this was couched in terms of a project.

Table. **Syllabus for the programming of a microcontroller in Process Control**

Formal training Subject matter covered in class relating to the programming of the Basic Stamp		Project With the circuit constructed so far[**] it should be possible to run experiments with the liquid-level system to demonstrate:
Topics	Source	
1. Input and output, Use of a switch for single input	Experiments 1 & 2 in “What’s a Microcontroller”[*]	**1.** the calibration of the sensor **2.** self-regulation **3.** a step test to determine the parameters in a FOPDT model of the system **4.** on-off control **5.** P-control **6.** offset **7.** PI control **8.** signal aliasing (control interval selection) **9.** disturbance rejection with various modes of control **10.** tuning performance (e.g., QAD, if appropriate).
2. Use of switches to input binary numbers. Serial and synchronous data	Experiment 2 in “Basic Analog and Digital”[*]	
3. Basic analog to digital conversion (8-bit converter)	Experiment 3 in “Basic Analog and Digital”[*]	
4. Example of analog-to-digital conversion with the LM53 temperature probe	Part of Experiment 4 in “Industrial Control”[*]	
5. Plotting and capture of analog and digital data from the Basic Stamp	Appendix A: Stamp Plot Lite in “Industrial Control”[*]	
6. Driving a servo motor with the Basic Stamp	Experiment 3 in “What’s a Microcontroller”[*]	
[*] Student workbooks published by Parallax (Parallax 1999a-c).		[**] The circuit after topic 4 in the formal training.

All the circuitry with which to do the project, the second part, was constructed in the first part. In the project, the students had to replace the temperature sensor with the pressure sensor; adjustments of the span and offset (potentiometer adjustments) and the calibration code were then necessary. The essential tasks for the project are summarized in the Table above. Most of the project comprised the writing of programs to demonstrate the various phenomena. Students came to realize that combinations of various phenomena could be demonstrated with a single program. Most of the class time for the last three weeks of the semester was devoted to the students’ project work. The assignment of weekly homework was stopped and the lab was opened up for extra periods so students could work on their projects in lieu of homework. A project report (20% of their final grade) was required; this comprised fully documented programs and graphs showing the various phenomena. (Data collected from Stamp Plot Lite are easily brought in to Excel.)

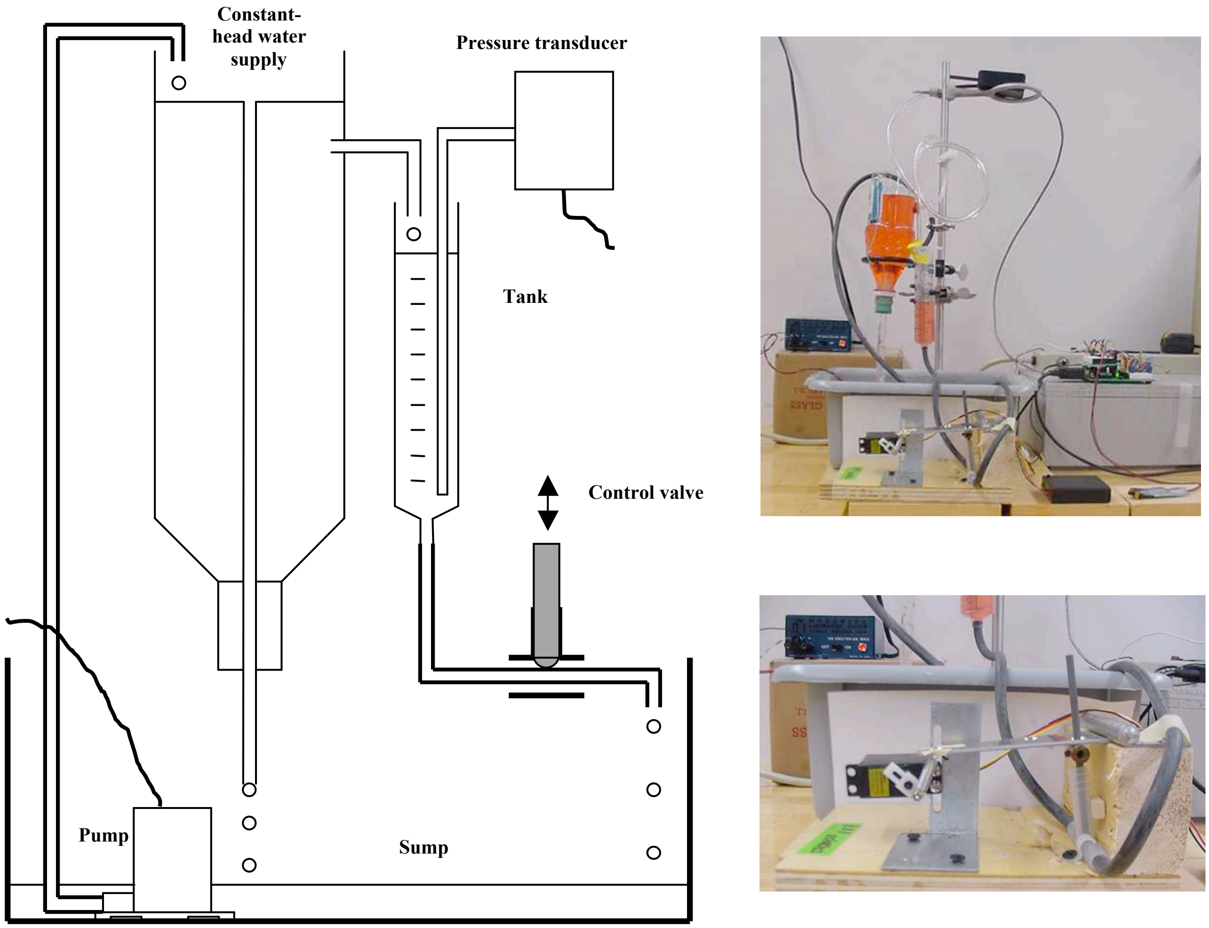


Figure 1. The liquid-level apparatus
The water contains a red food dye. The photograph in lower right-hand corner shows closer view of the control valve.

Experimental

In addition to the BOEs, kits obtained from Parallax relating to "Industrial Control " and "Earth Measurements" (Parallax, 1999c,d) were sufficient to provide the required electronic components and the pump. The liquid-level apparatus is shown in Fig. 1; it comprises

1. A sump - a plastic pan (Consolidated Plastics 67003LL) serves this function well.
2. A miniature water pump (Edmund Scientific No. 50,345, part of the "Earth Measurements" kit). This requires a 3VDC power source. It may be driven from the BOE by a transistor circuit and a power resistor; it can then be switched on and off in a PBASIC program. However, in more recent developments we are using a separate power supply.
3. A constant head reservoir constructed by cutting the bottom off a 12-oz plastic soda pop bottle.
4. The "tank" within which the level of water is to be controlled. This is a 60-mL disposable plastic syringe (Becton Dickinson 309663); its scale is useful. (One of the problems with teaching experiments and demonstrations in process control is ensuring that response times are not too long; the students' attention span is always shorter than a response time! So apparatus should be reasonably small.)

5. The level sensor. This comprises a glass tube fixed into the body of the syringe with aluminum wire. The top of the glass tube is connected with Tygon tubing to a pressure sensor (Schaevitz GA100-005WD, available from DigiKey). The sensor is mounted on a PCB board within a small plastic box; it should be kept dry so the sensor is best mounted well above the tank. It will measure pressures in the range 0-5 in. of water and provide a voltage output of 0.5-4.5 VDC. It requires a supply of 5VDC and so is conveniently powered from the BOE.
6. The control valve. Rubber tubing (5/16 in. OD, 1/16 in. wall) is connected to the outlet of the syringe and it passes through a tee-connector (for 5/16 in. ID tubing). A metal rod (5/16 in. OD) is set in the vertical arm of the tee; its position is altered by a beam, constructed from a 4-in hinge. The end of the beam is connected by a tension spring to the rotor arm of the servo motor, or servo (Parallax Standard or Hobbico CS-61 from FerretTronics). The servo is driven by a series of high-low voltage pulses; the duration of the high pulse fixes the position of the rotor arm. The PBASIC command, *pulseout*, creates these pulses. Under no circumstances should the power for the servo be obtained from the BOE, if the BOE's power is supplied by a wall transformer. A battery pack with an on-off switch (Radio Shack 270-409, 4 AA batteries) was found to be the best option here; for diagnostic purposes, it is convenient to be able to switch off the servo while a program is running.

Programming

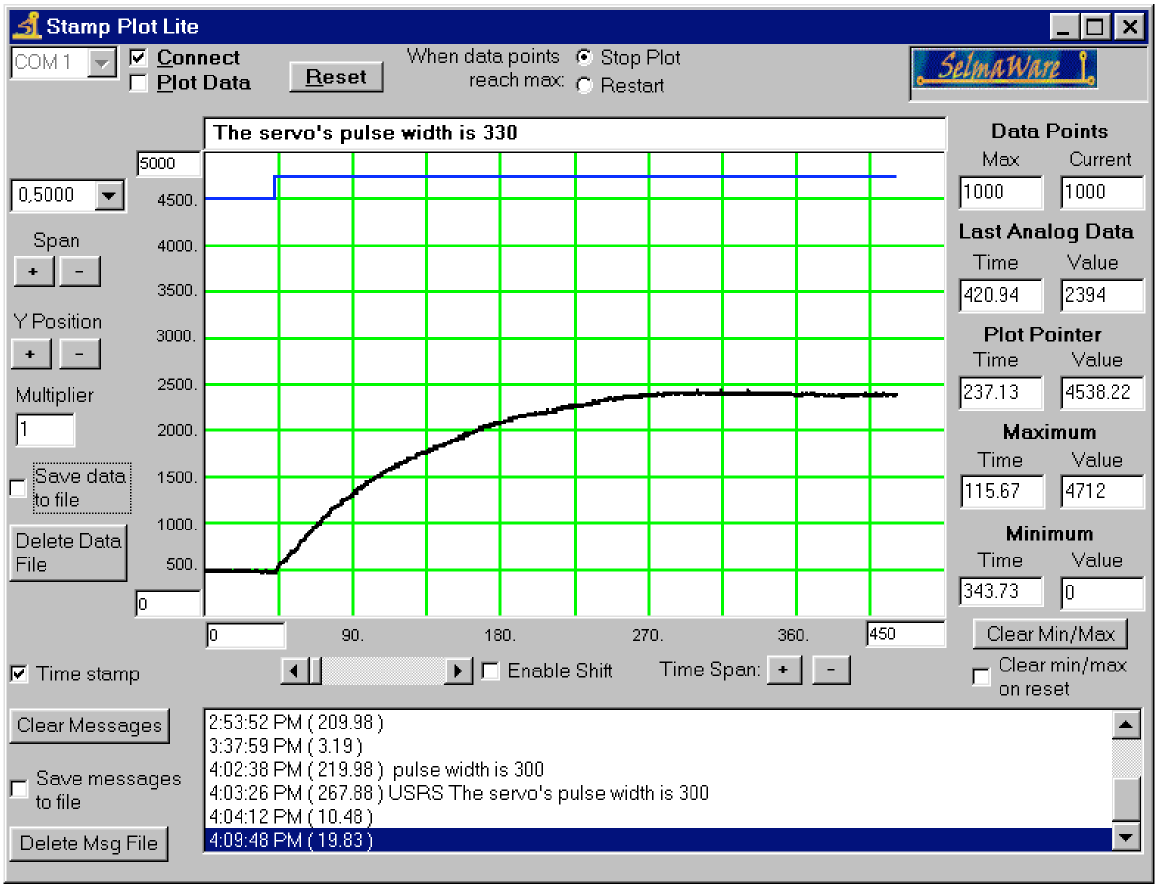
Rapidly, it became apparent to students that programs should have a loop structure in which you measure the water level and then take some action there from. However, a difficulty arises; the servo requires a continuous series of pulses to maintain a given position; this is achieved by the repeated execution of the *pulseout* command. The microcontroller only executes a single command at a time. Therefore, to measure the level, the operation of the control valve has to be temporarily suspended. Students then came to realize that the measurement of the water level is not necessary for every cycle through the program; a counter could be established to measure the level on every 100th pass, say. In this way, the microcontroller could be programmed to spend most its time operating the valve. This experience here leads naturally to the idea of a control interval.

Fortunately, the Basic Stamp attracts an accomplished entourage of enthusiasts with the result there are good sources of code, ideas and devices to be found, especially on the web. The majority of its applications are in robotics, and the application here may represent a new departure. The problem described in the previous paragraph is of pedagogical value; however, it may be circumvented by use of a chip called the PAK-VIII Pulse Output Coprocessor (AWC Electronics). Upon the receipt of a single batch of commands from the microcontroller, the PAK-VIII will generate a continuous stream of pulses to position the servo until another batch of commands is received. Examples of the code required are readily available (AWC Electronics, 2002). The microcontroller is then free to do other things, and is not constrained to spend most of its time running the servo. This will give the students greater flexibility in writing programs.

PBASIC is easy to use and the CD-ROM that comes with the BOE contains the reference manual of commands. It is possible to have the manual open (with Acrobat Reader) while you are programming; examples of code can be cut and pasted in the active programming window, and tried out.

Results and Discussion

Figure 2 comprises the window of Stamp Plot Lite after running a step test. A switch was used to move the valve from position to another. Stamp Plot Lite will plot the individual bits of a 4-bit, or lower bit, binary number and a single analog signal. The upper trace (binary signal) shows the state of the switch (0



COM 1
☑ Connect
☐ Plot Data
● Stop Plot
○ Restart

The servo's pulse width is 330

0,5000
Multiplier: 1
☐ Save data to file
☑ Time stamp
☐ Enable Shift
☐ Clear min/max on reset
☐ Save messages to file

Data Points — Max: 1000, Current: 1000
Last Analog Data — Time: 420.94, Value: 2394
Plot Pointer — Time: 237.13, Value: 4538.22
Maximum — Time: 115.67, Value: 4712
Minimum — Time: 343.73, Value: 0

```
2:53:52 PM ( 209.98 )
3:37:59 PM ( 3.19 )
4:02:38 PM ( 219.98 )  pulse width is 300
4:03:26 PM ( 267.88 ) USRS The servo's pulse width is 300
4:04:12 PM ( 10.48 )
4:09:48 PM ( 19.83 )
```

Figure 2. The results of a step test shown in Stamp Plot Lite

or 1); this is most convenient for establishing the time at which the change is made. The liquid level, the analog signal, is shown in the lower trace. The x-axis represents the time in seconds and the y-axis represents the liquid level in units of one thousandths of an inch (mil). The use of the latter units is a convenience arising from the fact that arithmetic operations are integer only. Students had difficulties in appreciating that computations are done with integer arithmetic in small devices such as the Basic Stamp. Although we spent time doing examples within PBASIC in class, questions involving integer arithmetic on the final exam were poorly done.

The response to the step test demonstrates the self-regulatory nature of the system; it moves from one steady state to another after a change. The clarity of the result shown here (almost text-book perfect!) belies the difficulties that were encountered by most students. Steady-state operation was difficult to achieve; this generated many complaints, which were met with the retort: "You now know a reason why process control is necessary!" From the step test, the time constant (here about 75 sec), the dead time (here about 2 sec) and the steady state process gain can be obtained; these data can then used to obtain tuning parameters using the Cohen and Coon settings, for instance. Stamp Plot Lite allows for data to be saved to a file and subsequent figures are generated from these data files.

Data from experiments with on-off control are shown in Figure 3. The set point for the water level is 2 in (2000 mil). The control loop's ability to reject disturbances is of paramount importance for chemical engineering. Here disturbances were mimicked by quickly pouring in 5 mL of water into the tank. The two traces represent different choices of control interval. Some students recognized that the valve is continually switching on and off and this might lead to early failure. This, in part, could be alleviated by

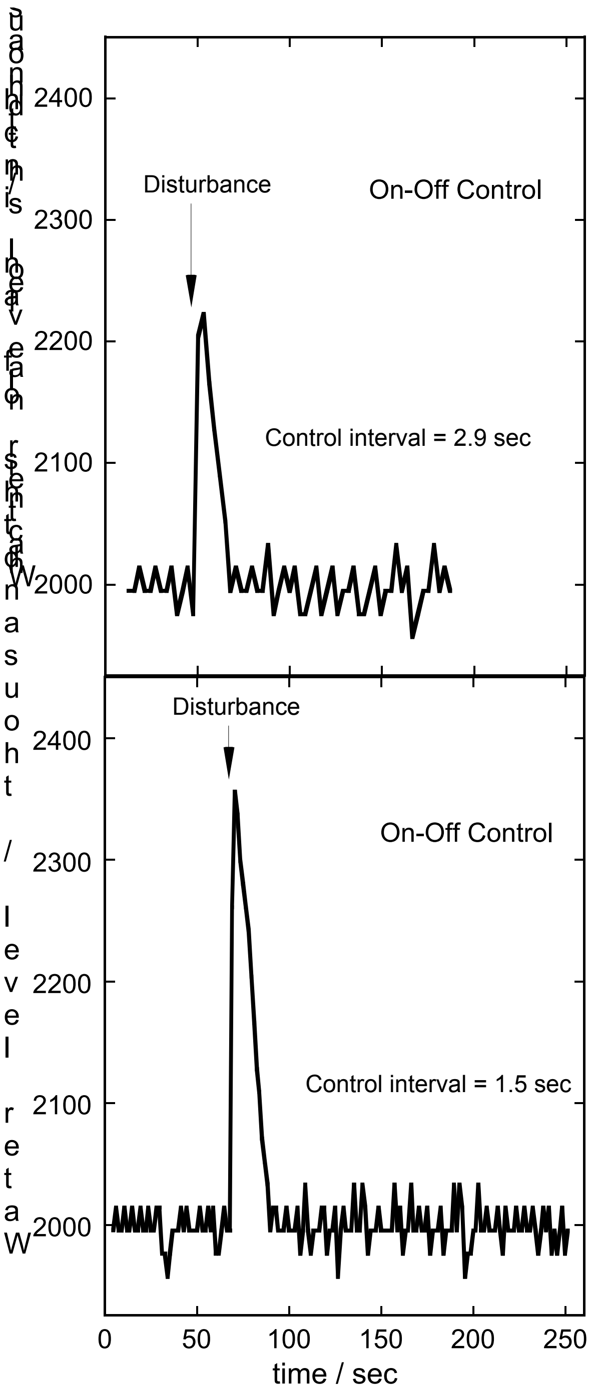


Figure 3. On-off control with different control intervals showing disturbance-rejection properties

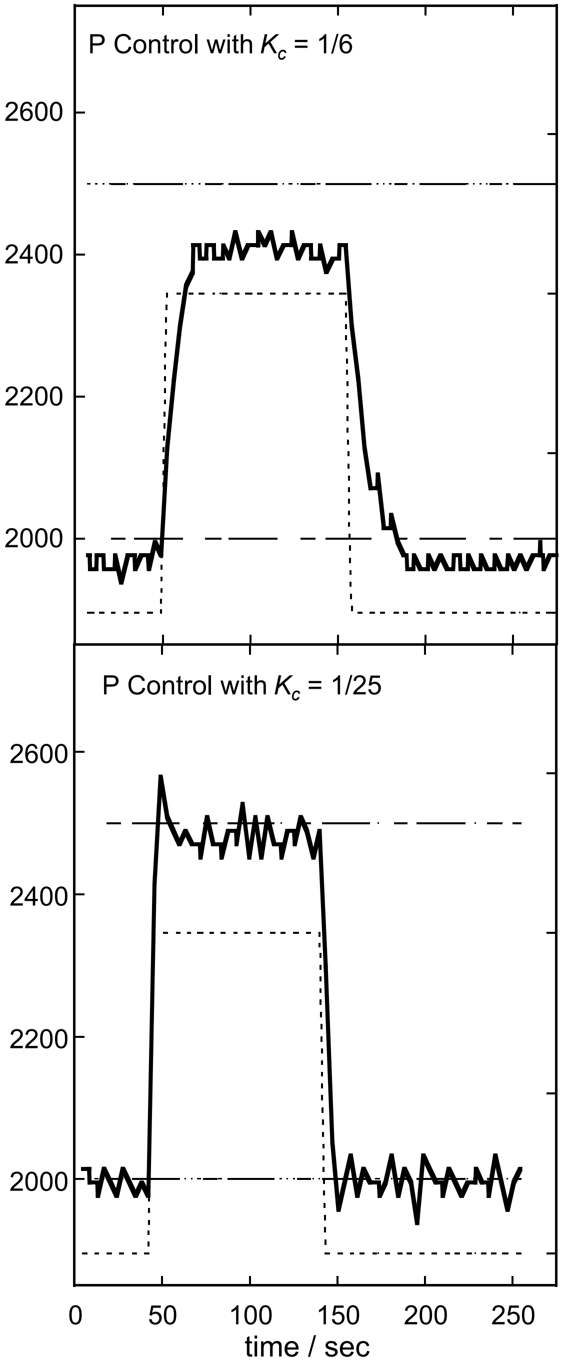


Figure 4. Change of set point under proportional control for different values of the proportional gain, K_c
The units of K_c are 5 μsec/mil.

increasing the control interval. Another solution (before using PID control) is to employ differential gap control (Parallax 1999c) and some students programmed this mode of control.

Figure 4 shows data from experiments with proportional control. A switch was used to choose between two values for the set point, 2000 and 2500 mil. The lower graph shows the response when the proportional gain is 1/25 with units of 5 µsec/mil. The trace with the narrowest dashes (block shape) represents the switch state (0 or 1). The upper graph shows the response for the higher value of the controller gain; here the ultimate response is offset by about 100 mil, or 0.1 in.

Students had difficulties with the units of the proportional gain. I found it helpful to remind them of the block diagrams and algebra discussed in class. The time units of 5 µsec (Pak-VIII time base) arise from time base used for the pulse width of the signal sent to the servo. (The time base for the *pulseout* command is 2 µsec.)

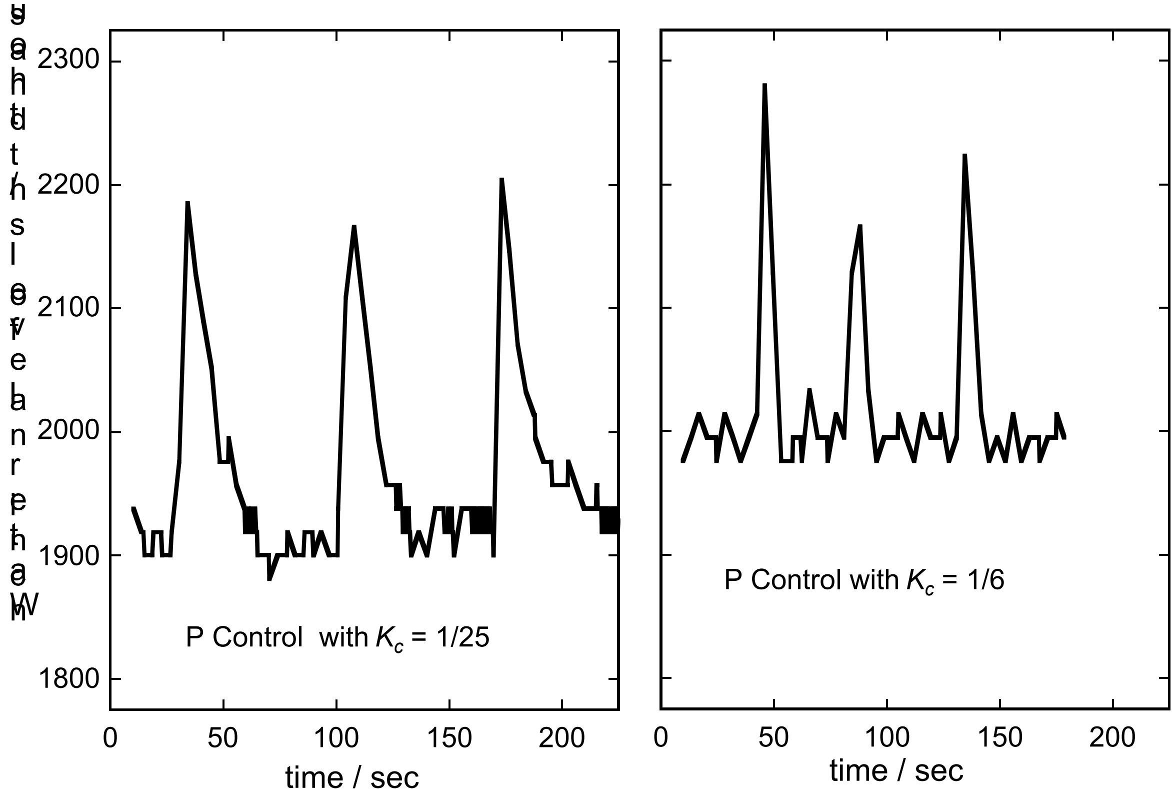


Figure 3. Results of disturbance-rejection experiments for the proportional control loop with different values of the proportional gain The units of K_c are 5µsec/mil.

The disturbance rejection properties of the proportional control loop are shown in Figure 5. Here 5 mL of water were dumped quickly into the tank three times. It can be seen that the loop with the higher proportional gain shows the quicker recovery.

Concluding Remarks

I have described briefly the first semester with this experiment. It was not possible to achieve all that I wished – the students barely got into P-control in their project work. The problems encountered were many; the most time was spent with personal computers and multifarious problems associated with different operating systems and special problems arising from networked configurations in the University computer lab. (None of the problems were attributable to the software from Parallax.) I intend to continue this experiment next spring by which time most of the infrastructure problems should be solved and these should not determine the rate of progress.

However, it is apparent to me that the students' rate of progress was significantly hampered by their programming skills. Although their exposure to "computer technology" is much greater than mine at the equivalent stage of life, their programming skills seem to be inferior. It seems that learning to program with punched cards embodies one with an undefined discipline that seems to be lacking now. The important intellectual discipline that programming brings is the ability to imagine how events, or processes, should occur with time.

On the whole, students welcomed the change of pace from usual classroom routine and the opportunity to do some practical work that did not require considerable time. It may be overly ambitious to do this exercise in 50-min periods; certainly longer periods or a separate lab section would make it easier. It should be emphasized that this required a considerable amount of extra effort on my part, and will do so for some time to come. However, I think the hiring of teaching assistant with a background in electronics, or majoring in electrical & computer engineering, would make life easier.

The costs involved are reasonable amounting to about $300 per system (the BOE, components and liquid-level apparatus) at this stage. The costs of items that may fail or get destroyed inadvertently are small; for example, the servo costs about $15. When you consider that an air-controlled actuator costs about $1000, then this is a cost effective way of exposing students to practical experience in process control. The software is free and can be downloaded from various websites (Parallax 2002a,b; Selmaware 2002). Lack of space prohibits us from letting students work alone on a single system.

In addition to developing their programming abilities, students learn about other facets of computing, viz. bits, bytes, words, serial communication, synchronous and asynchronous communication, and A/D conversion. Also, they get some practical experience with fluids; many opportunities arise to remind them of material covered in Fluid Mechanics (ChE 3111), our gateway course into the upper division. The liquid-level problem, and its linearization, is covered in formal class periods and with homework in the process control course. However, the most important benefit for students is practical experience in fault detection. Most faults were attributable to incorrect wiring; the natural instinct was to conclude that a component had failed. Only one component, a potentiometer, failed during the semester; the Basic Stamp itself is very robust.

From the instructor's view a particular benefit was more direct interaction with students. Soon it was possible to tell what stage they had reached from the questions they asked. Even, the shyer students recognized the need to ask questions if they were to make progress. The weaker students were able to feel a sense of accomplishment with their project work, although they did get as far as the more able ones.

Acknowledgements

I am grateful to UMD and the Department of Chemical Engineering for a single semester leave in the Fall of 2001. The short course, organized by Professor W. Schlick of Schoolcraft College, Livonia, Michigan and taught by A. Lindsay of Parallax in November 2001, was most useful. For unrelenting technical assistance, I am indebted to D. Long. The work of C. Patullo, an undergraduate assistant, was essential in running this experiment.

References

AWC Electronics, 2002, http://www.al-williams.com/awce/, accessed July 15th, 2002

Parallax, Inc., 2002a, http://www.parallaxinc.com/, accessed July 15th, 2002

Parallax, Inc., 2002b, http://www.StampsInClass.com/, accessed July 15th, 2002

Parallax, Inc., 1999a, “What’s a MICROCONTROLLER?”, Student Workbook Version 1.8

Parallax, Inc., 1999b, “Basic ANALOG and DIGITAL”, Student Workbook Version 1.0

Parallax, Inc., 1999c, “INDUSTRIAL Control”, Student Workbook Version 1.0

Parallax, Inc., 1999d, “Earth MEASUREMENTS”, Student Workbook Version 1.2

Riggs, J.R., 2001, “Chemical Process Control”, 2nd edition, Ferret Publishing, Lubbock, Texas.

Selmaware, Inc., 2002, http://www.selmaware.com/, accessed July 15th, 2002