

Process Control Teaching Tools

Roberto Werneck do Carmo

Pontifícia Universidade Católica do Rio de Janeiro (PUC-RIO)

R. Marques de S. Vicente, 225 Rio de Janeiro, RJ 22453-900 Brasil

e-mail: werneckr@venus.rdc.puc-rio.br

Abstract - This paper describes the initial results and the current development status of a Web-based environment that has proved invaluable as a tool for teaching Process Control to Chemical Engineering students. The homepage, which can be found at <http://venus.rdc.puc-rio.br/werneckr>, also covers numerical methods that are useful for Process Control simulations. A continuous evolution of this Web-based teaching environment, as a result of the class activities during the next semesters, is intended.

Introduction

The World Wide Web (WWW) has gained widespread popularity, and educators have been quick to take advantage of it. Process Control professionals were no exception. Homepages, such as Dr. Sommerfeld's Process Control page [1], were built in order to allow easy access to Process Control handouts. Prof. Price's homepage [2] was probably among the first pages to distribute Process Control course content through the Web. A number of Process Control tutorials can also be found in the Web, ranging from theoretical descriptions of control algorithms (such as those by StClair [3] and by Shaw [4]) to hands-on tutorials (Tilbury and Messner [5]).

The development of our Web-based teaching environment started in 1996 with a WWW homepage containing the course materials and some links to other resources found elsewhere on the Internet. A very convenient way to distribute handouts and notes, the homepage also provided an extension of the class environment. New ideas born in the classroom, for instance, could be included in updated pages posted overnight.

The homepage also proved a very convenient way to distribute different kinds of course material, not limited to text and graphics. Dynamic content such as data files and spreadsheets soon found its way into the homepage. This is particularly advantageous in Process Control, since in order to grasp the basics the student needs to be able to visualize dynamic phenomena - those in which properties of a given system change with time.

We have developed a number of simple simulations by using Microsoft Excel and Visual Basic for Applications, in which the Visual Basic modules handle the scientific calculations and the few

commands required in order to read and write data from the spreadsheet.

Computer simulation, a very useful visualization tool, is usually implemented with programming languages (FORTRAN, C++) or with mathematical software. Most students are familiar with the basics of programming, but have a hard time putting together all the routines for data input and graphics required for a useful simulation. By using a commercial spreadsheet, the learning curve for data input and visualization is virtually eliminated, since these are very simple to use and often well known to many students; after all, they are a popular tool to generate eye-catching reports.

Joining these two developments - the WWW homepage and spreadsheet-based simulations - helped motivate some students that might be otherwise frustrated should they have to write computer code for a graphical user interface. By reducing the learning curve, these tools allowed the students to focus on the results obtained, which helped convey the basics of Process Control with very simple, yet useful visualization tools.

Facing a challenge

One goal when putting together the support material for the Process Control course at the Pontifícia Universidade Católica do Rio de Janeiro (PUC-RIO) was to provide the students with a teaching environment that did not divert focus from the dynamic phenomena which are the subject of the course.

This challenge was compounded by the fact that part-time dedication to the University and frequent travel impose limits the face-to-face interaction with the students outside the classroom. Resources capable of bridging this gap and keeping the student interest up in between classes were thus critical to a successful teaching environment.

Process Control

Some of the skills an engineer learns relate to the calculation of a number, or set of numbers, that describe a state of a given system. Process data (such as pressures, temperatures and flow rates), equipment sizing results (reactor volume, heat transfer area, number of trays in a column), are examples of

meaningful engineering information that can be conveyed by numbers. Control engineering, however, involves dynamic phenomena which often require functions of time to be fully understood.

Process control is the branch of control engineering that deals with the systems found in the process industries, such as oil & gas, petrochemical and chemical plants. The objective of process control has been described by Marlin [6] as to maintain desired conditions in a physical system by adjusting selected variables in the system.

In order to fully understand the principles of Process Control, the student must be able to visualize the dynamic behavior of the system without control and with different control strategies; a clear understanding of the process goals, environmental impact, safety, and economics is required in order to select the “best” strategy, and to eventually select and “tune” the control system. Dynamic simulations provide a very convenient way of visualizing most of the dynamic phenomena the course deals with.

Building the homepage

The World Wide Web (WWW) was the natural place to look in order to extend the teaching environment beyond class hours. At first, a plain vanilla homepage containing an index and linking to some text pages containing course notes was assembled. Students willing to go through the trouble of looking up the homepage were thus rewarded with early releases of the course materials.

The initial task was designing the homepage to include more useful material and a meaningful navigation structure. This has been achieved as more notes, including graphics, links to other resources in the Internet, and exercises, have been gradually added. The homepage was also enriched by the inclusion of a related course (Numerical Methods for Chemical Engineering) that provides methods that are used extensively in Process Control.

Updates to the homepage have typically been frequent, providing rapid feedback when necessary. New content, including proposed exercises, can be readily recognized and easily retrieved.

Dynamic Simulations

Digital dynamic simulations are convenient ways to visualize the time-dependent behavior of physical systems. A mathematical model of the system (usually a set of differential equations) is solved by numerical methods, and the results are shown - preferably in a graphical format. Traditionally, dynamic simulation results have been printed as plots with time as the abscissa.

A simple dynamic simulation of a gravity-flow tank presented by Luyben [7] illustrates this. A fluid is pumped into a tank (figure 1) at a flow rate F_e ,

which can be a function of time. The outflow from the tank (F_s) is gravitational and depends on the level (h).

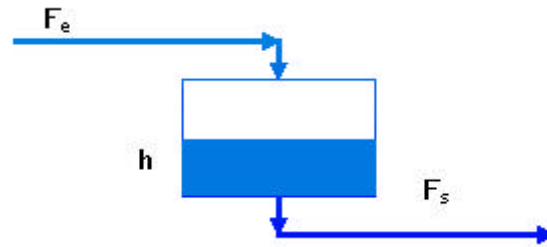


Figure 1: gravity-flow tank

This system can be modeled by two differential equations. Equation 1 shows that the outflow F_s can be affected by two terms: a positive contribution from gravity (proportional to the fluid level in the tank) and a negative contribution due to pipe friction losses (proportional to the square of the outflow). Equation 2 is a simple material balance stating that the fluid level depends on the accumulation of fluid in the tank due to the different flow rates. In these equations, ρ , μ , and A_t are the model parameters.

$$\frac{dF_s(t)}{dt} = h(t) - (F_s(t))^2 \quad (1)$$

$$A_t \frac{dh(t)}{dt} = F_e(t) - F_s(t) \quad (2)$$

Figure 2 shows the result of a simulation where the tank was initially at steady state (inflow = outflow, thus constant level). Soon after the simulation was started, the inflow was increased by 20%. The plot shows how the tank level rises in response to the higher inflow, and then tends to a new steady state.

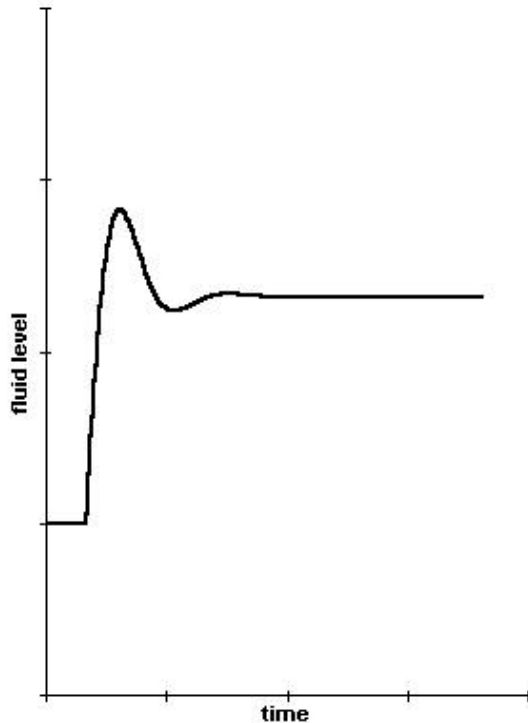


Figure 2: gravity-flow tank response to a change in inflow

A simple, yet efficient dynamic simulation that allows selecting three different controller modes, with adjustable parameters, can be found at Prof. Williams's homepage [8]. Exciting examples of Java-based simulations can be found at the Labore homepage [9].

All these simulations have a common feature: they represent a physical system that can be easily visualized, yet they define it with some abstraction, which allows the person running the simulation to tailor the system by changing a number of parameters. Given the ability to change the simulation parameters (and thus the properties of the system it represents), the student can get insight into the workings of the system that would otherwise require a lot of effort.

Simulations in Real-time

Simulations can also be motivating. The similarity to computer-based games is not to be overlooked - learning from a well-devised simulation can be fun, and being able to customize or to create a simulation can be even more fun. There is an evident catch, however: a boring simulation, or one that does not give the student the chance to act on the model as he or she desires, can inflict significant damage to student motivation.

The use of animation is a natural for dynamic simulations. Instead of using plots to convey the results of a dynamic simulation, one can easily use a graphical model of the system and use animation to

update the graphical model. By synchronizing the numerical solution of the mathematical model and the updates to the graphical model, a meaningful "real-time" animation can be obtained.

Using animation to display the results of a simulation opens new opportunities to make the model more engaging and realistic. For instance, alarms can be added to show if the model is close to an unsafe operating point or a shutdown message can be issued when unsafe conditions prevail. Although the animation presents essentially the same information as the plots, unsafe conditions are perceived differently and there is a stronger motivation to find ways to prevent them from occurring. Programming such extreme conditions in the model is usually simple (and often trivial), but can change the way the student relates to the solution of the problem: it is easier to figure out that what is required is not a good-looking plot, but safe and lucrative operating conditions.

The user interface

Special care was dedicated to the user interface of the simulation. By emphasizing that similar graphical screens are used in actual control of industrial plants, the student is motivated to understand the difference between a clean and a poorly designed user interface.

Härtel [10] cautions that an all-purpose user interface, full of menu choices that allow changing a large number of system parameters, can be too complex and distracting. Using a spreadsheet, the student does not need to drill down layers of menus; she or he is instead encouraged to rearrange the layout so that the parameters most likely to be changed are located conveniently. By getting the students involved in the development of a user interface, no matter how simple, the requirement of an efficient user interface is stressed.

Scientific programming inside a spreadsheet

The mathematical model equations are easy to code in the spreadsheets; programming in Visual Basic for Applications (VBA), for instance, requires only traditional BASIC language operators and commands that are well-known to students. However, the traditional Input/ Output routines (readln and writeln) are not up to the job of displaying a large quantity of numbers that change with time.

VBA provides a very convenient environment for entering data: each spreadsheet cell is an object with properties that can be read from or written to by a VBA program. The code fragment below shows how Equation (1) can be coded. The actual calculation of the time derivative of F_s is done in the third line; the first and second lines get the current values of two cells in a spreadsheet labeled "Data".

```

Function deriv2(h As Double,
                Fs As Double) As
Double

    alpha = Worksheets("Data").
           Cells(13, 3).Value
    beta  = Worksheets("Data").
           Cells(14, 3).Value
    deriv2 = alpha * h - beta * Fs ^
2

End Function

```

It is important to notice that this time derivative is calculated over and over during the simulation. One might be tempted to move the first two lines outside the function, reasoning that the parameters alpha and beta could be read just once, when the program starts. This would be correct only if these parameters did not change during the simulation. The code fragment above works even if alpha and beta vary with time; any change in the spreadsheet is readily incorporated in the following calculation without the need to stop the calculation, thus allowing the student to simulate time-dependent system properties and load disturbances.

Once the student understands how to read and write data from the spreadsheet, the user interface for the simulation can be built from standard spreadsheet features such as plots. Of course, more effort may be required for a more detailed graphical model of the physical system.

A more subtle requirement of the simulation relates to running it in a continuous form, repeating the calculations for each time interval. VBA provided

a function (Application.OnTimer) that allows scheduling a new calculation after a specified time interval. One solution is to place all the calculations inside a VBA procedure and schedule this procedure for repeated execution with Application.OnTimer.

After the two spreadsheet-specific functions shown above are explained to the students (which should take a few minutes), they are in a position to start modifying a template by including their own models and user interface elements; this is a surprisingly smooth learning curve for a dynamic simulation tool.

Viewing the spreadsheet online

Reading computer magazines, one gets the impression that Java programming (or similar Web-based active content) is a requirement in order to generate active content that can be viewed in the Internet. However, if platform independence is not the major requirement, there are many other ways to make active content available.

The spreadsheet files can be loaded from the Web in a number of different ways, the more conventional of which is to download the spreadsheet file and save it to disk. However, current browsers can open a spreadsheet (using OLE in-place editing) as long as the user has an appropriate viewer installed. Figure 3 is a screen shot showing an Excel spreadsheet inside Netscape Navigator. Note that the Navigator toolbar is shown just below the menu, and that the Excel toolbox is displayed below it.

Note that Figure 3 does not represent a static snapshot of the simulation: the Visual Basic code is there and can be executed - no Java required.

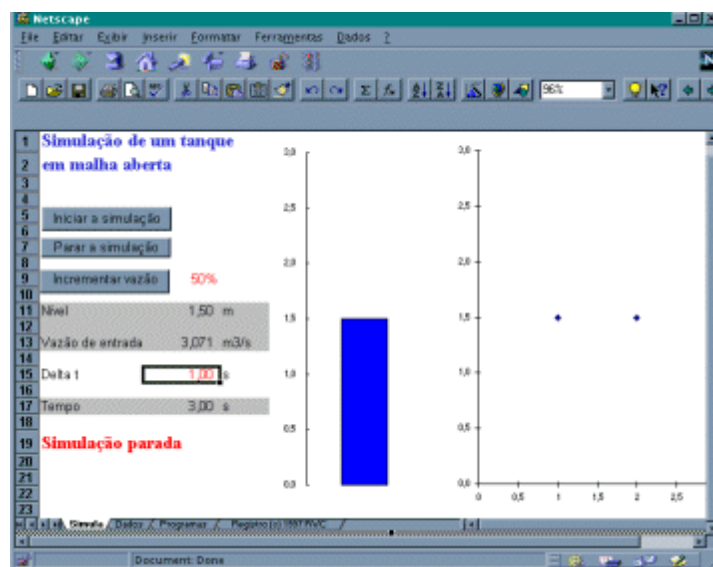


Figure 3: Using Netscape Navigator to execute a spreadsheet

Conclusions

The development of the Web-based teaching environment described herein was met with enthusiastic response from some students, and proved quite extensible - additional content and new features have been steadily included in the homepage. The dynamic simulations currently available include significant contribution from the students. The Web-based environment extends the classroom environment, enabling students to take advantage of their available time.

Future developments will not be restricted to spreadsheet-based simulations. Priority will be given to technology that will allow the students to interact more directly with the Web-based environment.

References

- [1] Sommerfeld, J., "Process Control I homepage",
<http://www.chemse.gatech.edu/~che4417/secC>
- [2] Price, R.M., "Chemical Process Control homepage",
<http://sunset.backbone.olemiss.edu/~cmprice/511>
- [3] St.Clair, D.W., "The PID Algorithm homepage",
http://members.aol.com/pidcontrol/pid_algorithm.html
- [4] Shaw, J., "The PID Control Algorithm homepage",
<http://members.aol.com/JohnShaw/pid/index.html>
- [5] Tilbury, D. and Messner, W., "Development and Integration of Web-based Software Tutorials for an Undergraduate Curriculum: Control Tutorials for MATLAB", *Proceedings, 1997 ASEE/IEEE Frontiers in Education Conference*, 1997
- [6] Marlin, T. E., *Process Control - Designing Processes and Control Systems for Dynamic Performance*, McGraw-Hill, 1995
- [7] Luyben, W. L., *Process Modeling, Simulation and Control for Chemical Engineers*, 2nd ed., McGraw-Hill, 1990
- [8] Williams, C.D.H, "Feedback and Temperature Control homepage",
<http://newton.ex.ac.uk/teaching/CDHW/Feedback>
- [9] LABORE/ENQ/UFSC, "Java Developments homepage",
<http://www.enq.ufsc.br/labore/java/development.html>, 1997

- [10] Härtel, H., "Efficiency of Computer Supported Learning Material", *Proceedings, 1996 Frontiers in Education Conference*, 1996