World Transactions on Engineering and Technology Education
Vol.6, No.2, 2007

Control systems demonstrations using spreadsheets

Nourdine Aliane

Universidad Europea de Madrid
Madrid, Spain

ABSTRACT: In this article, the author illustrates the use of *Excel* spreadsheets as a tool for constructing computer demonstrations in control systems simulation. To show some of these functionalities, two examples are included. The first one shows a simulation of a continuous-time system, whereas the second demonstrates the simulation of a digital control system. Both simulations are wrapped within simple interfaces to provide interactivity and make them valuable for teaching. The examples are not comprehensive since they do not fully exploit all of the features of *Excel*. They were chosen only because they are straightforward and self-contained illustrations of how an *Excel* spreadsheet can be used to create simple interactive simulations.

INTRODUCTION

In the last few years, spreadsheets have become a popular computational tool and a powerful platform for performing engineering calculations. The simplicity of spreadsheet programming, in addition to their rich library of built-in functions, plotting capabilities and other provided utilities, have made them attractive tools in many areas of engineering and science. Moreover, spreadsheets come with a macro language that allows the inclusion of standard computer code.

In the case of *Excel*, the introduction of the Visual BASIC for Application (VBA) language enables developers to greatly extend the capabilities of *Excel* spreadsheets by designing specific add-ins. In fact, VBA has the flexibility of a general programming language and, therefore, it is possible to program any of the standard numerical algorithms for solving systems of equations, ordinary differential equations and so on. The references [1-3] constitute a good core library for anyone who considers the use of spreadsheets for engineering computations.

Spreadsheets are also used as a teaching tool for constructing computer demonstrations and laboratory simulations in a number of engineering areas [4-7]. The immediacy of the spreadsheets and the convenience of their graphical representations make them a powerful didactic tool.

In the areas of control engineering, the use of spreadsheets has not received sufficient attention and the control engineering community uses preferentially professional software such as *MATLAB*/*Simulink* and *LabVIEW*. However, there are some interesting works in the context of this article, where spreadsheets are used as an alternative tool for simulating linear and nonlinear systems [8-11]. Thus, many people might be surprised to discover that *Excel* spreadsheets can be used for simulating control systems. In fact, simple and more complex simulations of common types of systems (discrete, continuous and hybrid) can be performed easily.

The purpose of this article is to present two illustrative examples of how control systems are handled within the *Excel* environment. The first example shows a simulation of a continuous-time system, whereas the second one demonstrates a simulation of computer-controlled system. Both simulations are wrapped within simple interfaces to provide them interactivity and make them valuable for teaching. The two examples are not comprehensive since they do not fully exploit all the features of *Excel*. They were chosen because they are straightforward and self-contained illustrations of how *Excel* spreadsheets can be used to create simple interactive simulations.

THE SIMULATION OF CONTINUOUS SYSTEMS

In this first example, it is intended to design a simple interactive tool that covers the fundamental ideas with respect to linear continuous-time systems. The system considered in this example is characterised by a general second-order transfer function given by:

$$G(s) = \frac{b_1 s + b_2}{s^2 + a_1 s + a_2}. \quad (1)$$

This model has four parameters. It has two poles that may be real or complex, one zero, and the right half-plane zero can be used as an approximation of a time delay. It grasps the dynamics of many systems that can be met in practice: pure integrator, double integrator, first order system, second-order system, oscillatory systems and even systems with a non-minimum phase.

Many analysis methods of linear systems are based on how dynamical systems react to typical inputs, such as impulse, step, ramp or sinusoidal signals.

A simulation of a continuous-time system on a digital computer is an estimation of the solution using a numerical integration algorithm (solver). Many solvers are available and the fourth-order Runge-Kutta algorithm (RK-4) is one of the most commonly-used algorithms in engineering calculations. The RK-4 solver can be implemented using standard spreadsheets. However, programming the solver using VBA is a more versatile approach and gives greater freedom to programmers. A number of traditional sources provide ready-made C/C++ code for standard engineering calculations. *Numerical Recipes in C* is an excellent source and conversion of RK-4 program from C into VBA takes only a few minutes [12].

Additionally, systems given in the state-space formulation are easier to simulate. So the system (1) is rewritten to its equivalent state-space description:

$$\begin{cases} \dot{x}_1 = -a_2 x_2 + b_2 u \\ \dot{x}_2 = x_1 - a_1 x_2 - b_1 x_1 \\ y = x_2 \end{cases} \qquad (2)$$

where $x(t)$ is the system state, u is the control input, and y is the output of the process. This set of differential equations can be easily dumped into the RK-4 solver.

User Interface

After the fundamental simulation model has been built, the core functionality is essentially done. The next step is to wrap the simulation with a user interface. The most important feature of spreadsheets is that a worksheet itself can be used as an interface for the simulation, so one does not need to design an input form, although it can be created within *Excel*.

The simulation layout is shown in Figure 1 and its design is extremely straight forward.

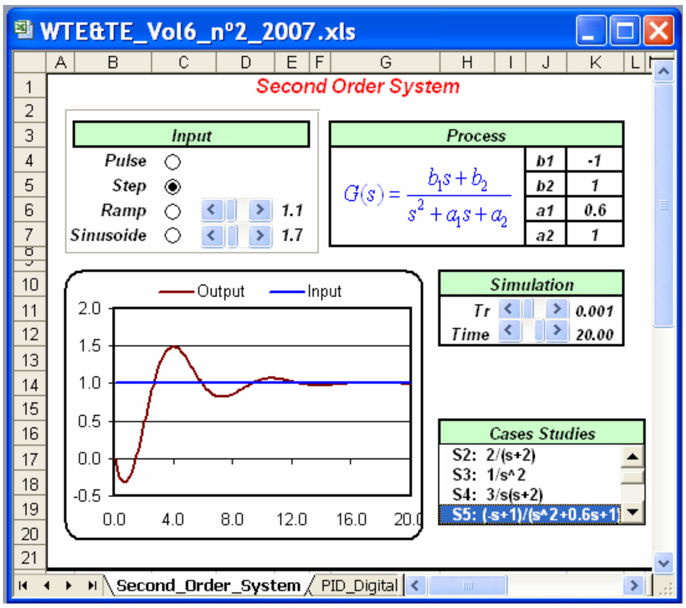

Figure 1: A simulation of continuous systems.

First, a table containing the parameters used by the simulation (the process parameters, the step time and the time simulation) is set up. All these data can be read from a VBA subroutine. Second, radio buttons are used to select the input references and scrollbars are set up to provide simple means for the user to vary the values (the rate of the ramp and the frequency of the sinusoid) in the input reference. In order to speed up and ensure accuracy when users are entering data into a spreadsheet, a number of predefined transfer functions (common model met in practice) are organised as *choose* options from a list box. A table is also reserved that will be filled in automatically and holds the solution (t, u, y) as it is computed. Finally, a chart is added to show the resulting curve that will automatically be updated when an input reference is selected.

This interactive tool is ready to use and does not need any configuration. The user defines a model by entering the parameters and immediately sees its time response plotted in a chart by clicking one button in the input group.

THE SIMULATION OF DIGITAL CONTROL SYSTEMS

Most control systems developed today are implemented using some sort of computers (embedded computer, PLC, etc). Knowledge of digital implementation of controllers is therefore essential. A schematic diagram of a digital control system is shown in Figure 2, where the plant to be controlled $G(s)$ is a continuous-time system, $D(z)$ is the digital controller and *ZOH* is the zero-order hold device.

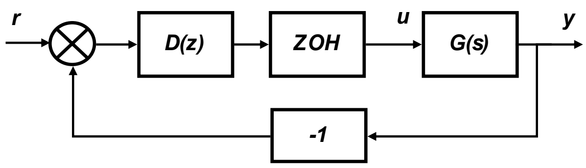

Figure 2: Digital control system diagram.

Simulating hybrid systems, combination of continuous and discrete systems, is also fairly straightforward. In this case, two time intervals (sample time) are used. The continuous system $G(s)$ is integrated using, for example, the RK-4 solver at each sample period Tr, whereas the control input is updated at each time kT. It is clear that $Tr << T$ and Tr should be selected as an integral divisor of T. To simulate a digital control on a computer, one can use the scheme shown in Figure 3.

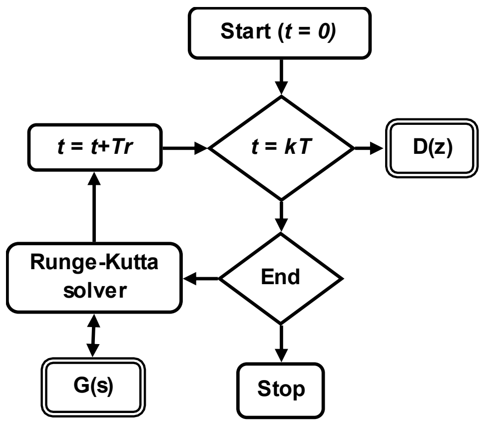

Figure 3: Digital control simulation scheme.

The subtlety of hybrid systems is that the updating routines are called at all sample times, but within these routines, a test is performed to determine if updates are necessary or not.

Digital PID Controller

Tuning a PID controller is one of the most frequent controller design problems. Therefore, it is an unavoidable chapter in basic control education. Thus, in this second example, it is intended to design an interactive tool that covers the fundamental ideas with respect to a digital PID controller. It is not intended to create a complete and sophisticated PID trainer, but only to show how a customised simulation can be designed using the *Excel*-VBA environment.

As an example, it is assumed that the process to be controlled is the model given in (1). Let the controller be a standard continuous PID with the transfer function given by:

$$K(s) = K\left(1+\frac{1}{T_i s}+\frac{T_d s}{1+Tds/N}\right) \qquad (3)$$

where K is the proportional gain, T_i is the integration time constant and T_d is the derivative time constant. N is used to place the pole of the filtered derivative. The discrete transfer function of the PID controller can be obtained using Euler or Tustin approximations [13].

Now suppose that the control signal is u, the error signal is e, and the parallel implementation using Tustin's approximation yields the following difference equations:

$$\begin{cases} u_i[n] = u_i[n-1] + A\left(e[n]+e[n-1]\right) \\ u_d[n] = \alpha u_d[n-1] + B\left(e[n]-e[n-1]\right) \\ u[n] = K\left(e[n]+u_i[n]+u_d[n]\right) \end{cases} \qquad (4)$$

with the constants:

$$A = \frac{T}{2T_i}, \quad B = \frac{2T_d N}{2T_d + NT} \quad \text{and} \quad \alpha = \frac{2T_d - NT}{2T_d + NT}$$

In this example, both the process and controller contain parameters that represent the relevant aspects of the controlled system, and it is convenient to see how they affect the system behaviour.

The interactive simulation layout is shown in Figure 4, where the process parameters can be changed by entering data manually, whereas the controller parameters values can be varied by using slider (scrollbars).

The tool is highly interactive in the sense that any click in the sliders or radio buttons performs an immediate recalculation and presentation of the process output and the control signal (or controller output). Thus, this mechanism makes it possible to observe the effect of varying the controller parameters without typing any command, and makes it easy to explore and understand how a given parameter affects the overall behaviour of the controlled system.

Graphical Plots

If one wants to obtain higher computational accuracy, a shorter simulation sample period will have to be used, but this implies increasing the total number of steps, and hence the number of rows in the spreadsheet. On the other hand, there is a practical limit on the length of a column that can be used (65,536 in *Excel 2003*), not only because spreadsheets have a finite number of rows, but also, more importantly, because calculations on long columns slow down the simulation and the data in long columns sometimes cannot even be plotted.

In order to overcome this problem, the numerical method with a compaction technique is adopted. It consists of compressing many steps into a single cell by storing one result for every n calculation steps [14]. For example, with n = 10 compaction, the simulation performs the computation n times before writing its result to the cell. This achieves the same computational result as it would have been obtained by using a 10-times longer column. Thus, with 200 points, it is sufficient to capture all the interesting parts of a transition response.

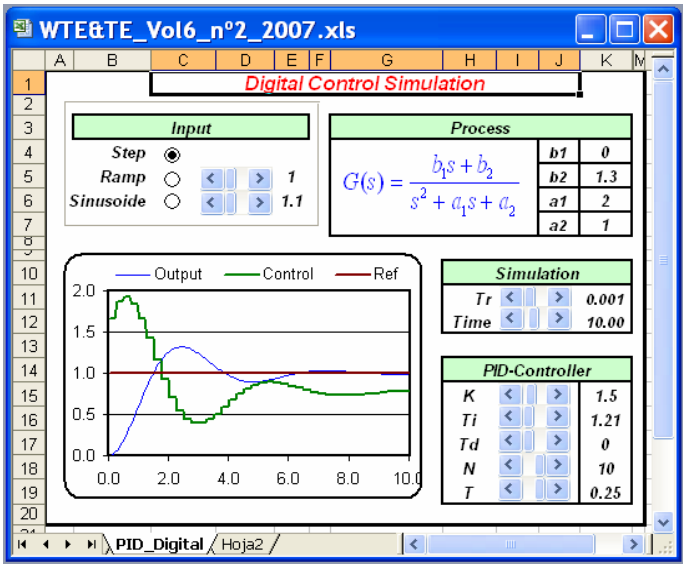

Figure 4: An interactive digital PID control.

PEDAGOGICAL ISSUES

This kind of tool is suitable for both students and teachers who are interested in the first control engineering course, since they offer ease of use and a shorter learning curve than professional software systems. They are intended to complement rather than replace existing approaches to learning, such as lectures and practical laboratory assignments.

The instructor can use them with some sort of projection facilities in lecture presentations for quick simulations and enhance the classroom environment. Because of its high interactivity, students can readily become familiar with the simulation panel quickly and so that their knowledge can be tested. Students can use them as a self-studying framework to help them to grasp quickly many abstract concepts. Furthermore, the high level of interactivity can give students enhanced intuition and a better understanding that can be difficult to achieve from developing exercises using a traditional approach.

These kind of tools are not only effective in presenting control process concepts in the classroom, but are also beneficial in extending students to perform assignments designing complex simulations using spreadsheets. Students often find that incorporating interactivity in their assignments presents them with appealing opportunities for designing attractive outputs, and as a teacher, the author found that this approach made them active and involved in their own learning process. Thus, this initiative can be most useful since it arouses students' curiosity to discover and improve their engagement.

CONCLUSION

In this article, the author shows how interactive tools for control systems education can be created using the *Excel*-VBA environment. The immediacy of spreadsheets and the convenience of its graphical representations can be combined with the wide availability in the literature of sophisticated higher-level programs to design powerful didactic tools.

Two simple interactive tools, a second-order linear system and a digital control systems simulation, are given as illustrations. All the parameters involved in both simulations can be entered manually or varied using sliders, and all the graphics are updated in a coherent manner, instantly reflecting the changes. In order to maximise clarity and maintain simplicity, each interactive simulation is focused on addressing a single core concept. This approach allows students to concentrate on a specific topic and to understand their characteristics better.

REFERENCES

1. Gottfried, B.S., *Spreadsheet Tools for Engineers Using Excel* (3rd edn). New York: McGraw-Hill (2005).
2. Bourg, D.M., *Excel Scientific and Engineering Cookbook*. Sebastopol: O´Reilly Media (2006).
3. Webb, J. and Saunders, S., *Programming Excel with VBA and .NET*. Sebastopol: O´Reilly Media (2006).
4. Chehab, A. and El-Hajj, A., Spreadsheet applications in electrical engineering: a review. *Inter. J. of Engng. Educ*., 20, **6**, 902-908 (2004).
5. Oke, S.A., Spreadsheet applications in engineering education: a review. *Inter. J. of Eng. Educ*., 20, **6**, 893-901 (2004).
6. Schumack, M., Use of a spreadsheet package to demonstrate fundamentals of computational fluid dynamics and heat transfer. *Inter. J. of Engng. Educ*., 20, **6**, 974-983 (2004).
7. Hanania, T.S. and Levin, I., Spreadsheet-based logic controller for teaching fundamentals of requirements engineering. *Inter. J. of Engng. Educ*., 20, **6**, 939-948 (2004).
8. El-Hajj, A., Kabalan, K.Y. and Khoury, S., A linear control system simulation toolbox using spreadsheets. *IEEE Control Systems Magazine*, 20, **6**, 8-14 (2000).
9. El-Hajj, A., Karaki, S. and Kabalan, K., Graphical simulation of an analogue computer using spreadsheets. *Inter. J. of Engng. Educ*., 18, **6**, 704-710 (2002).
10. El-Hajj, A., Karaki, S. and Kabalan, K.Y., Digital control systems simulation using spreadsheets. *J. of Computer Applications in Engng. Educ.*, 11, **1**, 6-12 (2003).
11. El-Hajj, A., Karaki, S. and Karim, Y.K., A nonlinear control system simulation toolbox using spreadsheets. *Proc. Inter. Conf. on Modelling and Simulation*, Pittsburgh, USA, 83-87 (2001).
12. Press, W.H., Flannery, B.P., Teukolsky S.A. and Vetterling, W.T., *Numerical Recipes in C: the Art of Scientific Computing* (2nd edn). Cambridge: Cambridge University Press (1992).
13. Åström, K.J. and Wittenmark, B., *Computer Controlled Systems* (3rd edn). New York: Prentice Hall (1997).
14. Levie, R., Spreadsheet simulation of chemical kinetics. *Critical Reviews in Analytical Chemistry*, 32, **1**, 97-107 (2002).