Int. J. Engng Ed. Vol. 25, No. 5, pp. 969–978, 2009
Printed in Great Britain.

0949-149X/91 $3.00+0.00
© 2009 TEMPUS Publications.

Use of Spreadsheets in Control Engineering Education*

NOURDINE ALIANE
Dpto de Arquitectura de Computadores y Automática, Universidad Europea de Madrid, Villaviciosa de Odón, Madrid, Spain. E-mail: nourdine.aliane@uem.es

Spreadsheets constitute a powerful modeling tool and an easy computer programming language. Spreadsheets' ubiquity, their low cost, their flexible programmable environment, as well as their plotting capabilities, make them attractive as an educational tool. Their capability is illustrated below through several examples ranging from classical control theory to more advanced topics such as optimal control or state-observer. Some of these examples are developed as classroom activities, whereas others are designed as interactive learning modules for illustrating specific control topics. Our approach focuses exclusively on the standard use of spreadsheets; that is, no macros nor macro language support are used. Pedagogical issues related to the use of spreadsheets in the classroom are also discussed.

Keywords: control engineering education; computers in education; spreadsheets

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, and plotting capabilities, have made them attractive tools in many areas of engineering and science [1, 2].

Spreadsheets are also used as a teaching tool to construct computer demonstrations and laboratory simulations in many areas of engineering education [3]. The combination of their flexible programmable environment and their plotting capabilities makes them powerful didactic tools. Spreadsheets' ubiquity and their low cost compared to professional software allow us to consider them as an alternative tool for education.

The idea of using spreadsheets for control is not new. Reference [4], which is one of the earliest papers on this topic, illustrates a spreadsheet implementation of a PID control of a DC-motor model. The use of spreadsheets for teaching process control was reported in [5], where the numerical approach was adopted to solve the differential equations of open and closed loop systems. It is also worth mentioning reference [6], which describes work where spreadsheets were used for designing a graphical tool to simulate linear systems. This tool was also extended to simulate discrete systems [7], and non lineal systems [8]. Thus, many people might be surprised to discover that spreadsheets can be used for control systems. However, all of the abovementioned works did not pay attention to techniques inherent in state space representation, or to how to exploit Excel-Solver utility to tackle optimization problems, and neither did they consider spreadsheets for designing interactive modules to illustrate specific control concepts.

Below I shall discuss how to use spreadsheets for control education through examples ranging from classical control theory to more advanced topics, such as optimal control or state-observer, as well as how to design simple interactive modules. Unlike the works [9, 10], where the author used Excel spreadsheets in conjunction with Visual Basic for Application (VBA) macro language to develop a command-oriented control toolbox for control system analysis and design, and to design interactive modules for control education for the Microsoft Excel environment, the here I focus exclusively on the standard use of spreadsheets; that is, entering data and formulae into worksheets without the need for macro programming language. This approach is effortless and particularly suitable for students or instructors with no programming background.

How do we use spreadsheets?

Our exploration of potential benefits in the use of spreadsheets for control education began in 2004, aiming their use at reducing mathematical manipulation during exercises in an introductory course in control. Later, in 2006, we exploited the spreadsheets' interactivity to develop some interactive learning modules to supplement lectures.

In the context of problem-solving activities, students are traditionally exposed to analytical techniques for plotting time and frequency responses of linear systems, analyzing their stability, or performing calculations for tuning controller parameters, to mention only a few examples.

* Accepted 2 April 2009

The technique based on hand-driven calculations is easy to carry out with simple systems, but requires more mathematical manipulations when applied to more complicated systems. Furthermore, this technique does not convey enough intuition to the students, since they focus more on the mathematical aspects rather than on control concepts. Systems analysis can be confusing for students because of its mathematical machinery rather than its complexity. In this context, we used spreadsheets to alleviate and simplify analytical development. This approach is not really different from analytical reasoning since students are still thinking in terms of mathematical objects, but with a substantial reduction in mathematical development. Indeed, spreadsheets have all the suitable tolls for performing control systems exercises. The use of spreadsheets is simple and involves entering appropriate formulae into worksheets, but requires students to use care and creativity in order to get the correct results.

On the other hand, an introductory course in control exposes students to theoretical concepts that are difficult to grasp quickly [11]. In such circumstances, we designed some spreadsheet-based interactive modules with animated graphs, which are aimed at providing insight into the fundamental concepts and present theoretical concepts in a way that would help students to develop good intuition for control fundamentals. An interactive module is a collection of dynamical graphics with active and clickable elements, which is aimed at explaining a few concepts [12]. In control education, many interactive tools have been developed using specialized software, such as Matlab in [13], Sysquake in [14], or JAVA [15]. In the references [13, 14], "the spreadsheet metaphor" is used to elucidate the interactivity feature, where users can dynamically change parameters and immediately see their effects. Paradoxically, spreadsheet-based tools for control education are practically non-existent, even though programs such as Microsoft Excel provide broad possibilities and an appropriate framework for creating interactive learning modules.

In addition to the descriptions of the techniques used to develop the examples, corresponding Excel files can be downloaded from the web [16] so that readers can study the details of the implementations. Even though the provided examples are developed using Microsoft Excel, the modules can be converted to any other spreadsheet program with minor adaptations.

CLASSICAL CONTROL CONCEPTS

Time and frequency responses, stability analysis, and feedback embody the fundamental ideas with respect to continuous-time linear systems and constitute the core of an introductory course in control. Below we describe using simple examples how spreadsheets can be used to plot frequency responses, such as Bode and Nyquist diagrams, plot time responses of linear systems to common input signals, such as impulse, step, ramp, or sinusoid, perform stability analysis using the Routh-Hurwitz criterion, and how to perform a simulation of a closed-loop control system.

Frequency response

Frequency response, such as Bode and Nyquist diagrams, basically requires calculations with logarithms and complex numbers. These types of calculation can be performed easily using functions provided by the Analysis ToolPack Add-in. This tool should be activated in (Tools->Add-Ins->Analysis ToolPak) menu. To illustrate the use of these functions, let us consider a second-order system given by the transfer function

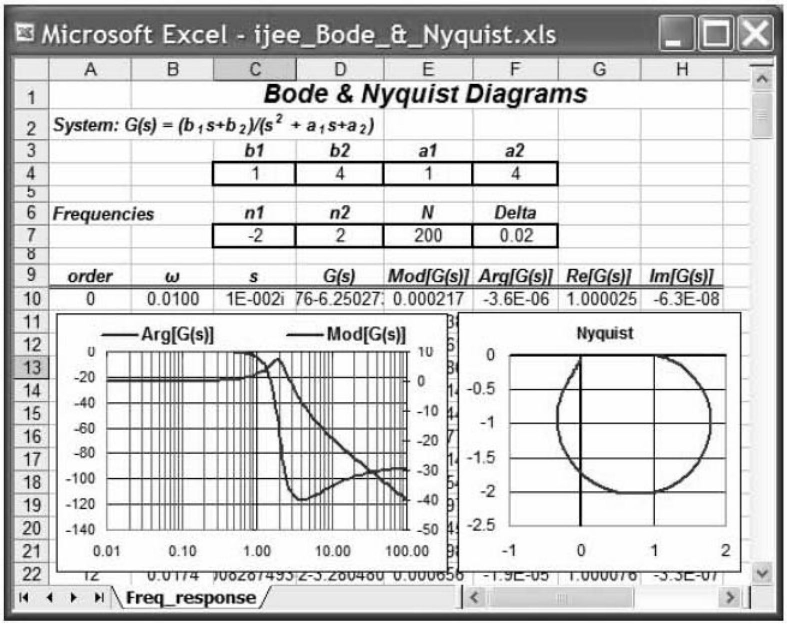
$$G(s) = \frac{b_1 s + b_2}{s^2 + a_1 s + a_2} \tag{1}$$

To obtain Bode and Nyquist diagrams, first, some cells are reserved for entering the parameters of the model, and two parameters (let us say n1 and n2) are used to specify the frequency range (10^{-n1} to1 10^{n2}). Next, a column with representative frequencies evenly spaced along a logarithmic scale between 10^{-d1} and 10^{d2} is created. A companion column containing complex numbers $s = j\omega$ is also created using the COMPLEX function. Afterwards, the transfer function is evaluated as complex number $G(j\omega)$ by entering its corresponding formula. Complex numbers are text entries with the rectangular form $(a + bi)$. The real and imaginary parts, the module, and the phase of complex numbers are obtained using IMREAL, IMAGINARY, IMABS, and IMARGUMENT functions, respectively. Finally, Bode and Nyquist plots can be generated using the standard Excel charting facility. Figure 1 shows a screenshot of the implementation of this example.

Many formulae of the examples provided in this paper use the naming cell, which is a mechanism that allows the assignment of meaningful names to cells. These names can be used as references, rather than cell addresses, in formulae. The naming mechanism is one of Excel's most valuable features, and helps to improve the readability of formulae and design well-organized worksheets. The Excel naming tool is accessed from the (Insert->Name->Create) menu.

Time response

Simulation of continuous-time systems is also straightforward. The approach consists in transforming continuous transfer functions to discrete ones by using one of the many approximation methods that map the s-domain into z-domain, such as Euler's or Tustin's methods [17]. This technique is systematic, and does not burden students with numerical issues for solving ordinary differential equations, since it does not require a lot of mathematical machinery.



Microsoft Excel - ijee_Bode_&_Nyquist.xls

	A	B	C	D	E	F	G	H
1			Bode & Nyquist Diagrams					
2	System: $G(s) = (b_1 s+b_2)/(s^2 + a_1 s+a_2)$							
3			b1	b2	a1	a2		
4			1	4	1	4		
5								
6	Frequencies		n1	n2	N	Delta		
7			-2	2	200	0.02		
8								
9	order	ω	s	G(s)	Mod[G(s)]	Arg[G(s)]	Re[G(s)]	Im[G(s)]
10	0	0.0100	1E-002i	76-6.25027:	0.000217	-3.6E-06	1.000025	-6.3E-08

Fig. 1. Bode and Nyquist diagrams of a second-order system.

For example, the discrete system that approximates the system given in (1), using Euler's forward approximation, is achieved through the substitution given by $s = (z-1)/T$, where T is the sampling period. The substitution leads to the following discrete transfer function,

$$\frac{Y(z)}{E(z)} = \frac{d_1 z^{-1} + d_2 z^{-2}}{1 + c_1 z^{-1} + c_2 z^{-2}} \tag{2}$$

where

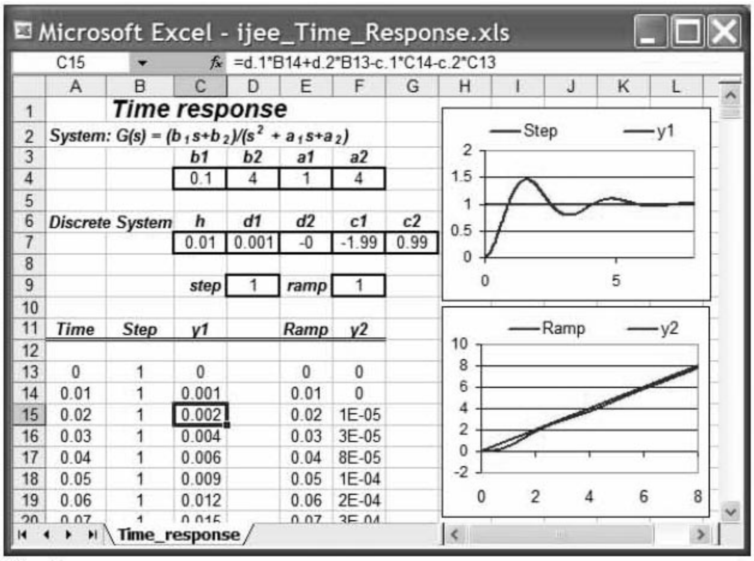
$$c_1 = a_1 T - 2; \quad c_2 = 1 - a_1 T + a_2 T^2$$
$$d_1 = b_1 T; \quad d_2 = b_2 T^2 - b_1 T$$

It is also worth mentioning that the variable z arises strictly in the context of systems simulation. The z-transform is introduced in the simplest manner possible, limiting the explanation to z^{-1} being a notation that represents one step time delay. Ultimately, the difference equation for implementation purpose is

$$y(k) = d_1 e(k-1) + d_2 e(k-2) - c_1 y(k-1) - c_2 y(k-2) \tag{3}$$

This recursive equation can be easily implemented using spreadsheets. The scheme works very well if the sampling interval is small enough. Figure 2 shows the step and ramp responses of a second-order system.

The method explained below can be easily

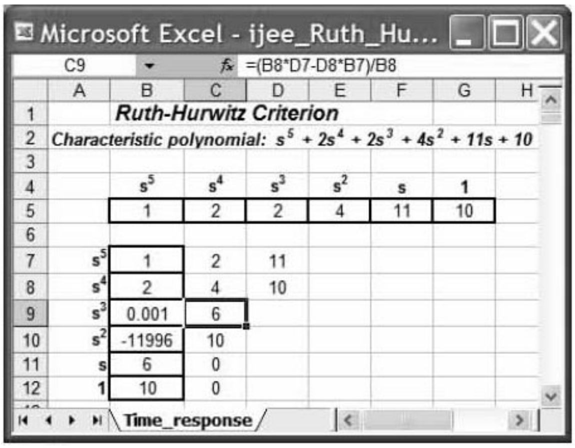


Microsoft Excel - ijee_Time_Response.xls

C15 =d.1*B14+d.2*B13-c.1*C14-c.2*C13

	A	B	C	D	E	F	G
1		Time response					
2	System: $G(s) = (b_1 s+b_2)/(s^2 + a_1 s+a_2)$						
3			b1	b2	a1	a2	
4			0.1	4	1	4	
5							
6	Discrete System		h	d1	d2	c1	c2
7			0.01	0.001	-0	-1.99	0.99
8							
9			step	1	ramp	1	
10							
11	Time	Step	y1		Ramp	y2	
12							
13	0	1	0		0	0	
14	0.01	1	0.001		0.01	0	
15	0.02	1	0.002		0.02	1E-05	
16	0.03	1	0.004		0.03	3E-05	
17	0.04	1	0.006		0.04	8E-05	
18	0.05	1	0.009		0.05	1E-04	
19	0.06	1	0.012		0.06	2E-04	
20	0.07	1	0.015		0.07	3E-04	

Fig. 2. Step and ramp responses of second-order system.



Microsoft Excel - ijee_Ruth_Hu...

C9 =(B8*D7-D8*B7)/B8

	A	B	C	D	E	F	G	H
1		Ruth-Hurwitz Criterion						
2	Characteristic polynomial: $s^5 + 2s^4 + 2s^3 + 4s^2 + 11s + 10$							
3								
4		s^5	s^4	s^3	s^2	s	1	
5		1	2	2	4	11	10	
6								
7	s^5	1	2	11				
8	s^4	2	4	10				
9	s^3	0.001	6					
10	s^2	-11996	10					
11	s	6	0					
12	1	10	0					

Time_response

Fig. 3. Implementation of Ruth-Hurwitz criteria.

extended to simulate parallel and series connections of linear systems. This possibility is particularly appealing since it helps students to simulate the effect of adding poles and zeros.

Routh-Hurwitz stability criterion

The Routh-Hurwitz stability criterion is a method for determining whether a linear system is stable or not by examining the locations of the roots of the characteristic equation of the system. The criterion establishes a systematic way for determining whether any of the poles of any size polynomial lie in the right half of the plane without having to explicitly work out the poles. The method is tabular and consists in building a table from the coefficients of the polynomial

$$P(s) = a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0$$

The table has $(n+1)$ rows with the following structure

$$\begin{array}{|llll} a_n & a_{n-2} & a_{n-4} & \ldots \\ a_{n-1} & a_{n-3} & a_{n-5} & \ldots \\ b_1 & b_2 & b_3 & \ldots \\ c_1 & c_2 & c_3 & \ldots \\ \ldots & \ldots & \ldots & \ldots \end{array}$$

where the elements b_i and c_i can be computed as follows:

$$b_i = \frac{a_{n-1}a_{n-2i} - a_n a_{n-2i-1}}{a_{n-1}},$$

$$c_i = \frac{b_1 a_{n-2i-1} - b_{i+1} a_{n-1}}{b_1}$$

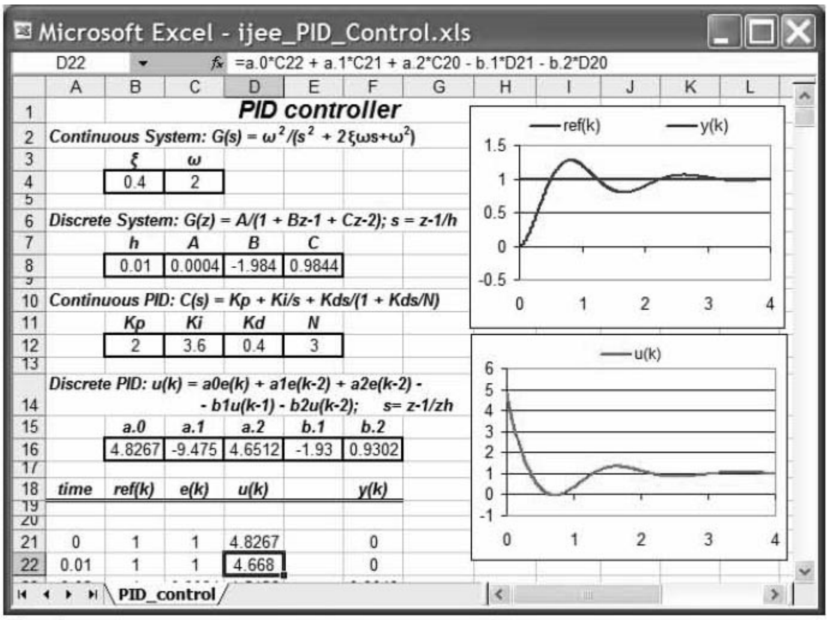
$$i = 1, 2, .., n$$

When completed, the number of sign changes in the first column will be the number of non-negative poles. This method is suitable for spreadsheet implementation (see Figure 3).

The special case when the first element of a row is zero can be handled also correctly. In this case, the zero element is replaced by a small value (let us say 0.0001), and then, the results are interpreted assuming that this small number has the same sign as the element above it. This method may also be used to establish the limiting values for a variable parameter, beyond which a system would become unstable. In this case, these variable parameters can be changed causing new results to take place immediately.

Feedback and PID control

PID controller is of special interest in the study of feedback control systems. In fact, it is an unavoidable chapter in a basic control course.



Microsoft Excel - ijee_PID_Control.xls

D22 =a.0*C22 + a.1*C21 + a.2*C20 - b.1*D21 - b.2*D20

	A	B	C	D	E	F	G
1				PID controller			
2	Continuous System: $G(s) = \omega^2/(s^2 + 2\xi\omega s+\omega^2)$						
3		ξ	ω				
4		0.4	2				
5							
6	Discrete System: G(z) = A/(1 + Bz-1 + Cz-2); s = z-1/h						
7		h	A	B	C		
8		0.01	0.0004	-1.984	0.9844		
9							
10	Continuous PID: C(s) = Kp + Ki/s + Kds/(1 + Kds/N)						
11		Kp	Ki	Kd	N		
12		2	3.6	0.4	3		
13							
14	Discrete PID: u(k) = a0e(k) + a1e(k-2) + a2e(k-2) - - b1u(k-1) - b2u(k-2); s= z-1/zh						
15		a.0	a.1	a.2	b.1	b.2	
16		4.8267	-9.475	4.6512	-1.93	0.9302	
17							
18	time	ref(k)	e(k)	u(k)		y(k)	
19							
20							
21	0	1	1	4.8267		0	
22	0.01	1	1	4.668		0	

PID_control

Fig. 4. PID control of second-order continuous system.

Here it is intended to show a spreadsheet implementation of feedback using such a controller. The transfer function of a standard PID controller is given by

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + \frac{K_d s}{1 + K_d s/N} \tag{4}$$

where $u(t)$ is the control signal, $e(t)$ is the input, and K_p, K_i, and K_d are the tuning parameters.

Again, applying Euler's approximation yields the following difference equation:

$$\begin{aligned} u[n] &= a_0 e[n] + a_1 e[n-1] + a_2 e[n-2] \\ &\quad - b_1 u[n-1] - b_2 u[n-2] \end{aligned} \tag{5}$$

with

$$\begin{aligned} a_0 &= (k_p NT + k_p k_d + k_i NT^2 + k_i k_d T + k_d N)/ \\ &\quad (NT + k_d) \\ a_1 &= -(2k_d N + k_d NT + 2k_d k_d + k_i k_d T)/ \\ &\quad (NT + k_d) \\ a_2 &= (k_p k_d + k_d N)/(NT + k_d) \\ b_1 &= -(NT + 2k_d)/(NT + k_d) \\ b_2 &= k_d/(NT + k_d) \end{aligned}$$

T is the sampling period.

Equations in (5) can be easily programmed in a spreadsheet. A simulation of PID control of a second-order continuous system is illustrated in Figure 4.

STATE-SPACE REPRESENTATION

State-space representation, also known as the "time-domain approach", provides a convenient and compact way to model and analyze SISO and MIMO systems. State-space representation is particularly suitable for numerical computations, and many calculations are carried out through pure matrix algebra.

Unbeknown to many users, Excel can do matrix operations very efficiently through its built-in matrix functions, allowing us to handle state-space calculations in a straightforward manner. In Excel, some matrix operations use standard operators (+, −, *), while others require the use of matrix functions such as MMULT, TRANSPOSE, SUMPRODUCT, or MINVERSE. Matrix formulae return results that can be a matrix, a vector, or a scalar, and an area of the correct size must be selected before entering the formula. After typing the formula, the results are displayed by pressing the three keys (Ctrl-Shift-Enter) at once.

Simulating systems given in state-space representation consists in, once again, transforming the continuous system to a discrete one. Applying Tustin's approximation to multi-variable linear systems given by

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + B_c u(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{6}$$

yields a discretized system of the form [18]

$$\begin{aligned} x_{k+1} &= Ax_k + B_1 u_{k+1} + B_0 u_k \\ y_k &= Cx_k + Du_k \end{aligned} \tag{7}$$

with

$$A = \left[I - A_c\frac{T}{2}\right]^{-1}\left[I + A_c\frac{T}{2}\right],$$

$$B_1 = B_0 = \left[I - A_c\frac{T}{2}\right]^{-1}\frac{T}{2}B_c,$$

I is the identity matrix, and T the sampling period.

The system given in (7) is simply a set of difference equations that can be easily implemented using spreadsheet method. Indeed, spreadsheets provide an appropriate framework to work with state-space representation. Many control concepts, such as state-variable feedback, testing controllability and observability, or simulation of multi-variable systems can be performed in a straightforward manner. We take this one stage further by presenting two examples developed as classroom activities: the first deals with the design pole-placement feedback technique using Ackermann's formula, the second shows how to use the Excel Solver utility to tackle optimal control strategy. Later, an interactive module illustrating the state estimation concept (Luenberger observer) is also provided.

Pole placement and Ackermann's formula

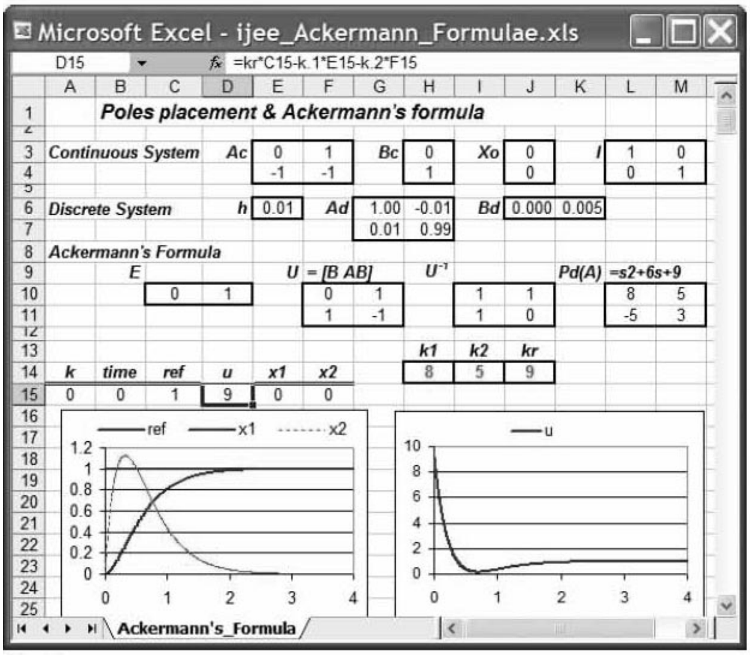
State-space design based on the pole-placement method entails finding a state-variable feedback gain so that the closed-loop poles are located at desired locations. There are many techniques to find a suitable state-variable feedback, as long as the system is reachable. One such technique is Ackermann's formula. If we consider the system given in (6) with the state feedback, $u = -Kx$, and the desired closed-loop poles locations yielding a characteristic polynomial P(s), then the state feedback is given by

$$K = EU^{-1}P(A_c) \tag{8}$$

where $E = [0 \cdots 0\ 1]$, $U = [B\ AB \cdots]$ is the reachability matrix, and $P(s)$ is the desired characteristic polynomial evaluated at the matrix A_c.

This design technique is a pure matrix calculation and can be implemented using spreadsheets. Figure 5 shows a state-variable feedback using Ackermann's method.

The interactive capacity of spreadsheets makes easy to test different aspects, such as changing the



Microsoft Excel - ijee_Ackermann_Formulae.xls

D15 =kr*C15-k.1*E15-k.2*F15

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		Poles placement & Ackermann's formula											
3	Continuous System			Ac	0	1	Bc	0	Xo	0	I	1	0
4					-1	-1		1		0		0	1
6	Discrete System			h	0.01	Ad	1.00	-0.01	Bd	0.000	0.005		
7							0.01	0.99					
8	Ackermann's Formula												
9		E				U = [B AB]		U⁻¹			Pd(A)	=s2+6s+9	
10			0	1		0	1		1	1		8	5
11						1	-1		1	0		-5	3
13								k1	k2	kr			
14	k	time	ref	u	x1	x2		8	5	9			
15	0	0	1	9	0	0							

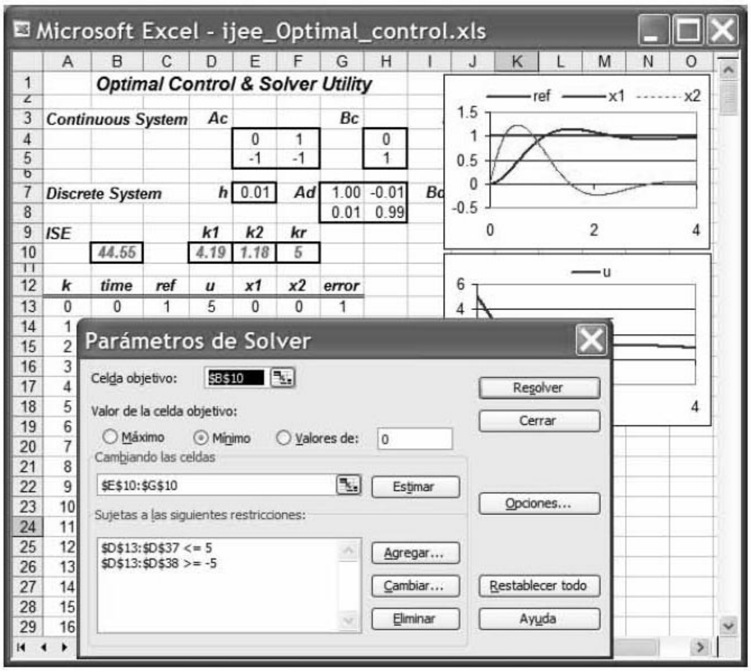
Fig. 5. Implementation of pole-placement feedback using Ackermann's formula.

poles' locations, or modifying the system parameters, and so forth.

Optimal control

Optimal control is a design technique that seeks to optimize a given criteria (a performance index), which is a quantitative measure of the quality of the applied control. For example, find the best possible tuning parameters of a controller that minimizes some specified performance index, so as to obtain minimum control energy expenditure, minimum rise time, or some similar criteria. However, obtaining this optimal adjustment is a non-trivial task, which is often a result of heavy mathematical analysis.

In the Excel environment, optimal control can be tackled using the Solver tool [19]. Solver is a free add-in to Excel that implements the generalized reduced gradient non-linear optimization method, and provides a convenient way to solve a variety of optimization problems numerically. It is intuitive and easy to use, since it provides an integrated



Microsoft Excel - ijee_Optimal_control.xls

	A	B	C	D	E	F	G	H	I
1		Optimal Control & Solver Utility							
3	Continuous System			Ac			Bc		
4					0	1		0	
5					-1	-1		1	
7	Discrete System			h	0.01	Ad	1.00	-0.01	Bc
8							0.01	0.99	
9	ISE			k1	k2	kr			
10		44.55		4.19	1.18	5			
12	k	time	ref	u	x1	x2	error		
13	0	0	1	5	0	0	1		
14	1								
15	2								
16	3								
17	4								
18	5								
19	6								
20	7								
21	8								
22	9								
23	10								
24	11								
25	12								
26	13								
27	14								
28	15								
29	16								

Parámetros de Solver

Celda objetivo: $B$10

Valor de la celda objetivo:

○ Máximo ● Mínimo ○ Valores de: 0

Cambiando las celdas

$E$10:$G$10

Sujetas a las siguientes restricciones:

$D$13:$D$37 <= 5
$D$13:$D$38 >= -5

Fig. 6. Optimal control using Solver utility.

framework for problem setting. Its use could represent a valuable tool to introduce students to optimal control without a full understanding of the underlying mathematics and without the need for mastering advanced programming techniques. Figure 6 shows a simulation using the solver to find the optimal state-variable feedback gains of a second-order system that minimizes the integral squared error (ISE) index,

$$J = \min \int_0^\infty e^2 dt \tag{9}$$

and that requires the control to satisfy the constraint $|u(t)| \leq 5$.

In the ensuing solver dialog box, we first enter the objective function (the ISE index) that is to be minimized in the target cell. This cell contains the SUM.SQUARE formula calculating the squared errors of the error column. Second, the adjustable cells (feedback gains) are also indicated, which could have initial values. Finally, it is also possible to impose constraints on variables, such as control signal. The constraints in this example are, \$B\$6:\$B\$30 <= 5 and \$B\$6:\$B\$30 >= −5, and define the control signal saturation limits. Upon completing the execution of the algorithms, Excel presents the user with the optimal tuning parameters.

Solver is suitable for tackling many other problems of optimization in an intuitive way. Its use allows students to gain a better understanding of the qualitative aspects of optimal control concepts, and sets up a background for extensions to other modern methods such as adaptive control, model identification, and so forth.

INTERACTIVE LEARNING MODULES

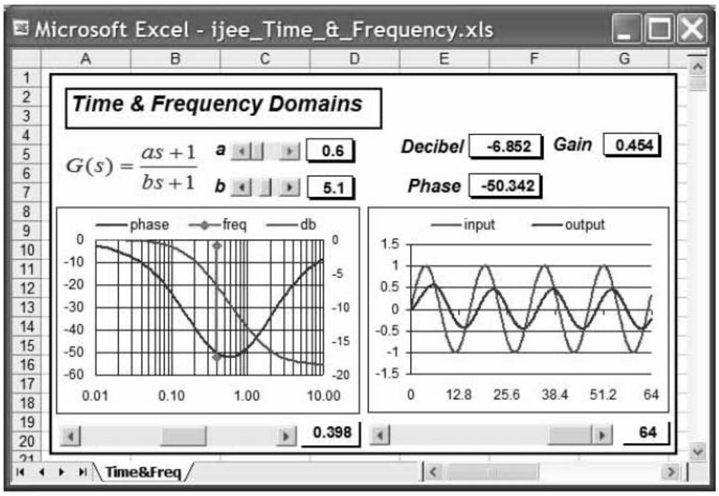
Modern spreadsheets, such as Microsoft Excel, provide utilities to create graphical user interfaces (GUI's). Wrapping simulations, such ones developed throughout this paper, within a GUI, will improve interactivity and would be advantageous for teaching and learning. This feature allows users to change parameters and view simulations without having to deal with the values of cells. Indeed, Excel includes a built-in tool with elements such as sliders, radio buttons, check boxes, etc, which allows developers to easily design GUIs for their Excel applications. The reader can go even further with Excel's help. As we point out below, Microsoft Excel provides broad possibilities for creating interactive learning modules.

Time versus frequency domain

One of the properties of linear systems is that if they are submitted to a sinusoidal input at a given frequency, their outputs oscillate at the same frequency, but the amplitude and the phase may change. This property is generally difficult for students to grasp quickly. This is a typical concept that can be explained better with the help of an interactive module. Figure 7 shows the screenshot of our Excel construction developed for classroom demonstrations.

This module implements a lag/lead filter submitted to a sinusoidal input with an adjustable frequency. The module plots Bode diagrams and the time response to the corresponding sinusoidal input. Users can vary the frequency from a slow to a fast oscillatory input by just dragging a slider, and immediately see the corresponding effect. The amplitude ratio and the phase-shift are evaluated, and can be checked immediately in the time response.

The spreadsheet equations are wrapped within a simple user interface, which is actually an Excel worksheet. When using spreadsheets, there is no need to design a specific input form, and the input parameters of the simulation can be given through reserved cells. A table is also reserved for holding the spreadsheet formulae and the solutions.



Microsoft Excel - ijee_Time_&_Frequency.xls

Time & Frequency Domains

$G(s) = \dfrac{as+1}{bs+1}$

a 0.6

b 5.1

Decibel -6.852

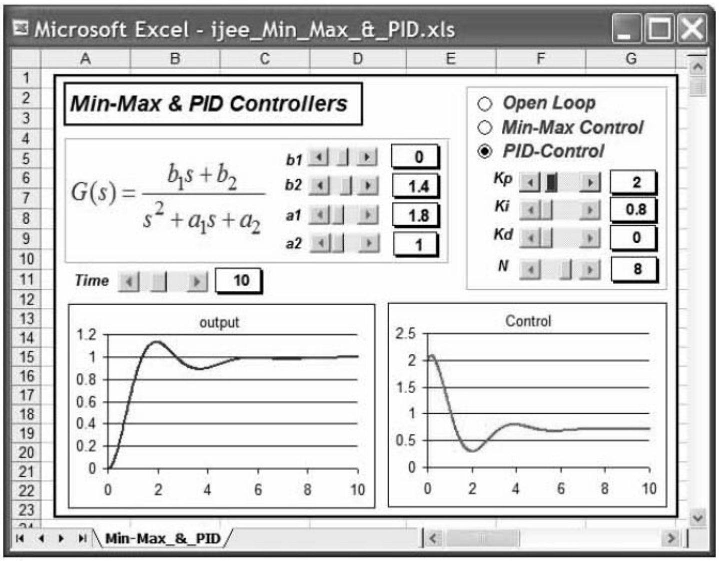
Gain 0.454

Phase -50.342

0.398

64

Fig. 7. Time v. frequency response.


Microsoft Excel - ijee_Min_Max_&_PID.xls

Min-Max & PID Controllers

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

b1 0

b2 1.4

a1 1.8

a2 1

○ Open Loop

○ Min-Max Control

● PID-Control

Kp 2

Ki 0.8

Kd 0

N 8

Time 10

Fig. 8. Min-Max and PID Control.

Finally, plots of the simulation results are generated using standard Excel utilities.

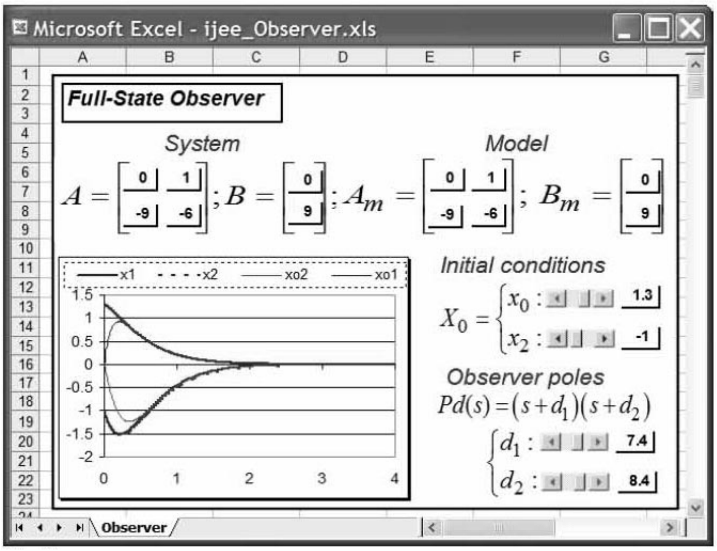
Interactivity is driven, basically, using sliders. Their use is straightforward and does not involve programming of any kind. The insertion of a slider into a worksheet is performed by selecting it from the (View->Toolbars->Control) menu. After dragging out an image, we have to set up the linked cell, enter the maximum and the minimum values, and the slider is ready for use. This feature makes it easy to set up interactive Excel-driven simulations.

Min-Max and PID controllers

The PID controller is the most common solution to practical control problems, and tuning it is one of the most frequent design problems. Therefore, it is appropriate to give it special attention. The screenshot shown in Figure 8 below is another interactive module that illustrates the PID and Min-Max control of a continuous second-order system. The module plots the system responses to step input, as well as the control signal. The interaction with the module is driven by sliders. Students can explore a full range of parameters by adjusting the values from low to high or vice versa. At a first stage, students should get a feel for the qualitative aspects of control, exploring what happens if a given tuning parameter gain is increased or decreased. The quantitative aspects do not matter as much. This routine enables students to quickly perceive the significance of a closed loop, and ultimately understand the meaning of the controller's tuning parameters better.

State observer

Simulations performed using state-space representation can also be wrapped within interactive user interfaces. As an example, Figure 9 below


Microsoft Excel - ijee_Observer.xls

Full-State Observer

System

Model

$$A = \begin{bmatrix} 0 & 1 \\ -9 & -6 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 9 \end{bmatrix}; A_m = \begin{bmatrix} 0 & 1 \\ -9 & -6 \end{bmatrix}; B_m = \begin{bmatrix} 0 \\ 9 \end{bmatrix}$$

Initial conditions

$$X_0 = \begin{cases} x_0 : 1.3 \\ x_2 : -1 \end{cases}$$

Observer poles

$$Pd(s) = (s+d_1)(s+d_2)$$

$$\begin{cases} d_1 : 7.4 \\ d_2 : 8.4 \end{cases}$$

Fig. 9. Interactive state observer module.

shows the screenshot of a module designed to introduce the state observer concept to students. The module simulates a second-order system, and a full-state observer is used to estimate the state variables of the systems. The system's initial condition as well as observer's poles can be changed interactively using sliders. After dragging sliders, all the graphics are updated in a coherent way, reflecting the changes instantaneously. As the observer is an approximate model of the system (real plant), it is also appealing to investigate the dynamic response of the observer when the model parameters present some deviation from the real plant.

DISCUSSION

The approach, based on Excel spreadsheet, was used in an introductory course in control for two consecutive academic years, from 2005–06 and 2006–07, at the Universidad Europea de Madrid. The use of spreadsheets benefits students and instructors in various respects.

The fact that the majority of students are familiar with Microsoft Excel makes it convenient for classroom activities. Some of these activities are described throughout this paper, and can be extended easily to different control topics. The use of spreadsheets helps students to master analytical techniques and proceed quickly. The reduction of the analytical developments allows students to focus more on control concepts rather than on programming, which ends up strengthening their instruction. The inherent interactivity of spreadsheets allows students to explore the effect of varying parameters and easily experiment with "what-if" scenarios.

Spreadsheets such as Microsoft Excel, provide many utilities for designing interactive simulations, and therefore can be used as an alternative platform for developing educational tools. Educators can become proficient in programming spreadsheets in a matter of hours, and thereby design their own classroom demonstrations. Interactive simulations and effective applications can be developed in a reasonable amount of time. These interactive modules are helpful in delivering lectures by providing quick simulations, generating multiple examples, and using them to reinforce the study of many theoretical concepts. In this sense, spreadsheets are not just a computational environment, but a valuable instrument for communicating.

Furthermore, it is worth extending the use of spreadsheets asking students to perform homework assignments designing specific modules similar to the ones presented in this paper. The incorporation of interactivity into the students' assignments improves their engagement, and designing attractive outputs helps them to become involved in their own learning process.

From the students' point of view, an informal focus group interview [20] conducted by the instructor, revealed that the majority of students consider that spreadsheet programs constitute a convenient framework for developing control exercises. They uniformly agreed that spreadsheets are easy to use, and they especially do not add a significant cognitive load. Furthermore, the interactivity and responsiveness of spreadsheets are considered valuable features for learning, since they allow students to observe the effect of varying parameters on the system behaviors. In summary, students believe that the use of spreadsheets is effective in supplementing the course material.

CONCLUSIONS

This paper demonstrates how spreadsheets can be used in control education. Examples of a broad spectrum of topics, ranging from classical to modern control theory, are provided. Interactive tools with a good level of interactivity can also be created with minimum effort. All of the examples provided above have been developed making standard use of spreadsheets. Spreadsheets' ubiquity, their low cost, the convenience of their graphical representations, and their high interactivity allows us to consider them as an alternative platform for control education.

REFERENCES

1. B. S. Gottfried, *Spreadsheet Tools for Engineers using Excel*, McGraw-Hill, 3rd edition, (2005).
2. M. B. David, *Excel Scientific and Engineering Cookbook*, O'Reilly, 1st edition, (2006).
3. Special issue: Spreadsheet applications in engineering education, *Int. J. Eng. Educ.,* **20**(6) (2004).
4. C. J. Fraser, Spreadsheet Applications for Control System Modelling and Analysis, *Int. J. Eng. Educ.*, **12**(1), (1996), Article 10.
5. C. Rives, and D. J. Lacks, Teaching process control with a numerical approach based on spreadsheets, *Chem. Eng. Educ.,* **36**(4), (2002) pp. 242–248.
6. El-Hajj, S. Karaki and K. Kabalan, Graphical Simulation of an Analog Computer Using Spreadsheets, *Int. J. Eng. Educ.*, **18**(6), (2002) pp. 704–710.
7. El-Hajj, S. Karaki and K. Y. Kabalan, Digital control systems simulation using spreadsheets, *Comput. Applic. Eng. Educ.,* **11**(1), (2003) pp. 6–12.
8. El-Hajj, S. Karaki, and Y. K. Karim, A nonlinear control system simulation toolbox using spreadsheets, in *Proc. Int. Conf. Modelling and Simulation*, Pittsburgh, U.S.A, (2001), pp. 83–87.
9. N. Aliane, Spreadsheet-based control system analysis and design: A command-oriented toolbox for education, *IEEE Contr. Syst. Mag.*, **28**(5), 2008 pp. 108–113.

10. N. Aliane, Spreadsheet-based interactive modules for control education, *Comput. Applic. Eng. Educ.,* (in press), 2009.
11. S. Dormido, Control learning: present and future, *Annual Review in Control,* **28**(1), 2004 pp. 115–136.
12. J. Sánchez, S. Dormido, and F. Esquembre, The learning of control concepts using interactive tools, *Comput. Applic. Eng. Educ.,***13**(1) 2005 pp. 84–98.
13. M. Johansson, M. Gäfvert, and K. J. Åström, Interactive tools for education in automatic control, *IEEE Contr. Syst. Mag.*, **18**(3) 1998 pp. 33–40.
14. J. L. Guzmán, K. J. Åström, S. Dormido, T. Hägglund, and Y. Piguet, Interactive learning modules for PID control, *IEEE Contr. Syst. Mag.*, **28**(5), 2008 pp. 118–134.
15. J. M. Albino, C. Lorenzo, L. Acosta, S. Torres, and E. González, A web-based tool for control engineering teaching, *Comput. Applic. Eng. Educ.,* **14**(3), 2006 pp. 178–187.
16. Examples [On-line]: http://www.esp.uem.es/aliane/ijee/ijee_examples.zip
17. K. J. Åström and B. Wittenmark, *Computer Controlled Systems*, 3rd edition. Prentice Hall, (1997).
18. H. Hanselmann, Implementation of digital Controllers—A survey, *Automatica*, **23**(1), 1987 pp. 7–32.
19. D. Fylstra, L. S. Lasdon, A. D. Waren and J. Watson, Design and use of the Microsoft Excel Solver, *Interfaces,* **28**(5), 1998 pp. 29–55.
20. M. Olds, B. M. Moskal and R. L. Miller, Assessment in Engineering Education: Evolution, Approaches, and Future Collaborations, *J. Eng. Educ.*, **94**(1), 2005 pp. 13–25.

Nourdine Aliane obtained the Electrical Engineering degree in 1990 from the École Nationale Polytechnique d'Alger, and the Ph.D. in physics in 1995 from the Universidad Complutense de Madrid. He is currently a professor of control engineering and robotics at the Universidad Europea de Madrid, Spain. His interests include control systems, robotics, and education.