Spreadsheet-Based Control System Analysis and Design

A Command-Oriented Toolbox for Education

NOURDINE ALIANE

Computer simulations are widely used in control education and constitute a natural complement to theoretical material. This statement is particularly true for an introductory course in control, which exposes students to abstract concepts that are difficult to grasp quickly [1]. In these circumstances, computer simulations can provide insight into the fundamental concepts and offer a valuable instrument to facilitate learning.

In control education, numerous professional software packages such as Matlab/Simulink and LabView have been adopted as teaching tools. However, to enhance learning, many educators have developed additional educational tools [2]–[7] that provide intuitive and interactive user interfaces. Interactivity, which means that users can dynamically change the system parameters and immediately see their effect on the system behavior, is a useful feature in control education [1]. Such interactivity can be found in educational tools such as *CCSdemo* [2], *Ictools* [3], *LSAD* [4], as well as in the more recent suite *PID-Basics*, *PID-Loop-Shaping*, and *PID-Windup* [5].

Developing simulation tools requires specialized software, such as Matlab or Sysquake [8], which is a Matlab dialect with interactive support. In addition to implementing the simulator engine (the solver), the development of such tools involves constructing a graphical user interface (GUI), which requires knowledge of advanced programming techniques such as event handling or multitasking. The creation of GUIs using only Matlab requires programming skills and effort. Furthermore, adding a basic level of interactivity to Matlab applications is a time-consuming activity.

In this article, we use Microsoft Excel as an alternative platform for developing interactive learning tools for control education. Modern spreadsheets include a macro language that permits the inclusion of standard computer code and allows users to incorporate specific programs to manage worksheets. In the case of Microsoft Excel, its built-in Visual Basic for Application (VBA) has the flexibility of a general-purpose programming language, thereby enabling developers to greatly extend its capabilities by designing specific add-ins [9].

Spreadsheets can also be used as teaching tools for constructing computer demonstrations and laboratory simulations in many areas of engineering education [10]–[14]. The combination of their flexible programmable environment and plotting capabilities makes them powerful didactic tools. Spreadsheets' ubiquity and low cost compared to professional software permit us to consider them as an alternative tool for education.

In control education, however, the use of spreadsheets has received little attention. Some work can be found where spreadsheets are used as an alternative tool for simulating linear systems [15], digital control systems [16], and nonlinear systems [17]. In fact, simple and complex simulations of continuous, discrete, and hybrid systems can be easily performed.

In this article we present a spreadsheet-based tool for control education. This tool is a command-oriented control toolbox (COCT) for working with linear systems in the Excel environment. Currently, COCT supports single-input/single-output linear systems described by transfer functions and provides many of the essential utilities necessary for an introductory course in control. The COCT workbook is freely available on the Web [18].

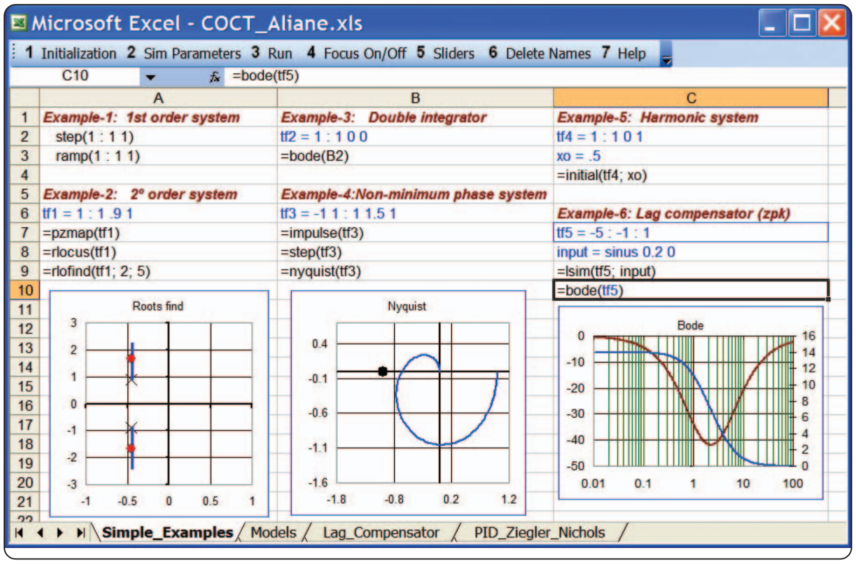
WHAT IS COCT?

COCT is an Excel workbook based on a set of macros and functions written in the VBA language. These functions are organized as a toolbox that provides utilities for handling control objects such as transfer functions, time and frequency responses, closed-loop simulations, and other minor objects. Once activated, these functions are integrated within Excel, behaving like any built-in function. The COCT workbook is embedded in a single Excel file, and its use does not require installation or configuration of any kind.

In the COCT environment, the cells of worksheets are used as containers for entering and storing control objects. Consequently, the active worksheet acts as a user interface for control system analysis and design, negating the need to design a specific input form. Control objects are entered into cells in the same way as data are entered into spreadsheets. Additionally, several models and simulations can be entered into the same worksheet, and the user can switch from one simulation to another by moving the focus from one cell to another and running the corresponding simulations. Figure 1 shows a screenshot of a COCT worksheet with multiple models and simulations. To manage the COCT environment and make it user friendly, several mechanisms, such as

Digital Object Identifier 10.1109/MCS.2008.927960

initializing, running, or focusing simulations, are programmed as VBA macros and arranged as a floating toolbar, which is automatically displayed when the COCT workbook is opened (see Figure 1).



=bode(tf5)

	A	B	C
1	*Example-1: 1st order system*	*Example-3: Double integrator*	*Example-5: Harmonic system*
2	step(1 : 1 1)	tf2 = 1 : 1 0 0	tf4 = 1 : 1 0 1
3	ramp(1 : 1 1)	=bode(B2)	xo = .5
4			=initial(tf4; xo)
5	*Example-2: 2º order system*	*Example-4:Non-minimum phase system*	
6	tf1 = 1 : 1 .9 1	tf3 = -1 1 : 1 1.5 1	*Example-6: Lag compensator (zpk)*
7	=pzmap(tf1)	=impulse(tf3)	tf5 = -5 : -1 : 1
8	=rlocus(tf1)	=step(tf3)	input = sinus 0.2 0
9	=rlofind(tf1; 2; 5)	=nyquist(tf3)	=lsim(tf5; input)
10			=bode(tf5)

FIGURE 1 The command-oriented control toolbox (COCT) environment. The worksheet is configured to display the formulas in Excel standard format. The worksheet contains six examples. In the first column, the step response of the first-order system $G(s) = 1/(s+1)$ is entered in cell A2. The step response is displayed by making the cell A2 active and then running the simulation by either clicking on the Run button or by pressing the Enter key. Management of the COCT environment is done through functions arranged in a floating toolbar.

Motivation

The development of COCT is motivated by the need to provide teachers and students with a learning tool that enables them to focus exclusively on control system concepts rather than on programming tasks. COCT is intended to be interactive in the sense that it allows users to see the effects of varying parameter values on the system behavior. Interactivity stimulates the students to learn and is therefore a key to enhancing their intuition about control fundamentals. Furthermore, COCT is designed to speed up learning by reducing the time needed to design and run simulations. Finally, this tool is intended to complement rather than replace existing approaches, such as lectures and practical laboratory assignments.

SCOPE OF COCT

COCT provides utilities that cover the fundamental control concepts for continuous-time linear systems. Users can enter transfer functions into a worksheet, plot the time and frequency responses to understand how the systems behave, explore how the locations of poles and zeros in the complex plane affect the time and frequency responses, or observe the effects of adding poles and zeros to systems, to mention only a few examples. COCT also provides utilities for simulating feedback control systems. The user can specify a model, tune the controller parameters interactively, and verify the performance in terms of rise time, steady-state error, or gain and phase margins. The principal capabilities of COCT are listed in Table 1. Most of these capabilities are handled through model parameterization and interactivity, as discussed below.

Model Parameterization

When using COCT, transfer functions can be defined in rational polynomial or zero-pole-gain forms and can be entered into a worksheet in multiple ways. For example, a simple way to enter a transfer function is by typing it

Table 1 Scope of command-oriented control toolbox (COCT). COCT supports many commonly used functions for control system analysis and design. The user can specify a model, plot its time or frequency response, simulate a feedback control system, and tune the controller parameters.

1) Model definition and conversion
   Polynomial and zero-pole-gain forms
   Model conversion
   Series connection

2) Time domain
   Step response
   Impulse response
   Ramp response
   Initial response
   Sinusoidal response

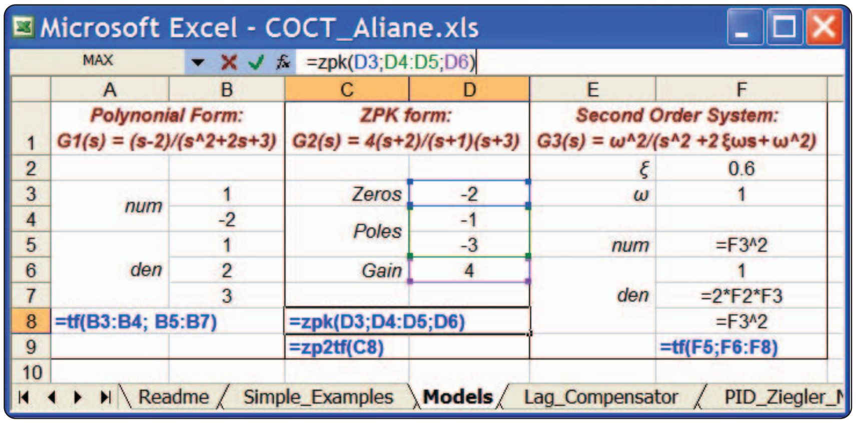
3) Root locus
   Pole-zero map

4) Frequency domain
   Bode plots
   Gain and phase margin calculation
   Nyquist diagrams

5) Closed loop
   Feedback using an arbitrary transfer function
   PID control
   Time response to input reference and load disturbances

6) Relevant features
   Maintaining multiple simulations
   Model parameterization
   Interactivity

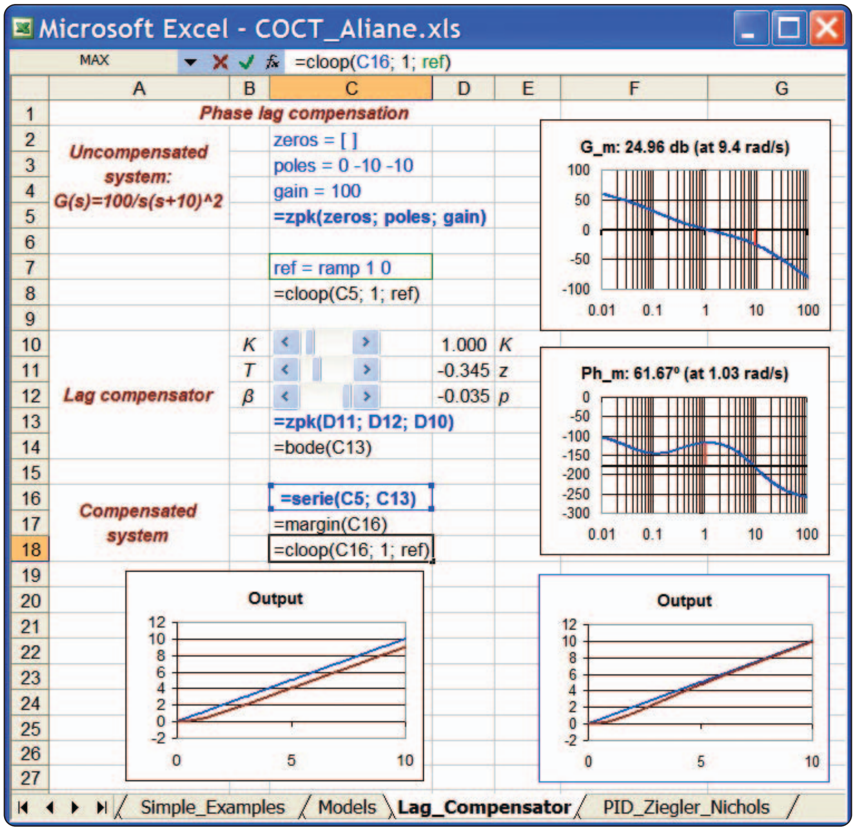
in a cell as a text entry. Alternatively, transfer functions can be assigned names, which can be used as references rather than using cell addresses in formulas. The convention adopted in COCT is that any cell with an entry in the form "alias = item" is assigned "alias" as a name. Additionally, the naming operation is programmed as a macro so that transfer functions are assigned names when they are entered, instead of the traditional way, that is, Insert->Name->Create. Named transfer functions are particularly useful for creating well-organized worksheets. Cells with transfer functions can be referenced anywhere in the workbook by functions that perform simulations and control calculations.



=zpk(D3;D4:D5;D6)

	A	B	C	D	E	F
1	Polynonial Form: G1(s) = (s-2)/(s^2+2s+3)		ZPK form: G2(s) = 4(s+2)/(s+1)(s+3)		Second Order System: G3(s) = ω^2/(s^2 +2ξωs+ω^2)	
2					ξ	0.6
3	num	1	Zeros	-2	ω	1
4		-2	Poles	-1		
5	den	1		-3	num	=F3^2
6		2	Gain	4	den	1
7		3				=2*F2*F3
8	=tf(B3:B4; B5:B7)		=zpk(D3;D4:D5;D6)			=F3^2
9			=zp2tf(C8)			=tf(F5;F6:F8)
10						

FIGURE 2 Model parameterization. Transfer functions $G1(s) = (s-2)/(s^2+2s+3)$, $G2(s) = 4(s+2)/(s+1)(s+3)$, and $G3(s) = \omega^2/(s^2+2\xi\omega s+\omega^2)$ are entered as parameterized models in cells A8, C8, and F9, respectively. In the definition of $G3(s)$, the damping ratio ζ and the natural frequency ω are entered in separate cells, which are used in Excel formulas to define the model.



=cloop(C16; 1; ref)

	A	B	C	D	E
1	Phase lag compensation				
2	Uncompensated system: G(s)=100/s(s+10)^2		zeros = [ ]		
3			poles = 0 -10 -10		
4			gain = 100		
5			=zpk(zeros; poles; gain)		
6					
7			ref = ramp 1 0		
8			=cloop(C5; 1; ref)		
9					
10	Lag compensator	K		1.000	K
11		T		-0.345	z
12		β		-0.035	p
13			=zpk(D11; D12; D10)		
14			=bode(C13)		
15					
16	Compensated system		=serie(C5; C13)		
17			=margin(C16)		
18			=cloop(C16; 1; ref)		

FIGURE 3 Phase-lag compensation. The transfer function of the system $G(s) = 100/s(s+10)^2$ is entered in cell C5. The lag compensator $C(s) = K(1+Ts)/(1+\beta Ts)$, with $\beta > 1$, is entered in C13. The series connection of the compensator and the system is entered in B16. The Bode plots with gain and phase margin calculation are entered in C17. The time responses to a ramp input of both uncompensated and compensated systems are entered in cells C8 and C18, respectively.

COCT also allows for entering models into a worksheet in a parameterized way so that their parameters can be changed in a straightforward manner. This versatile approach in creating models is achieved by two custom functions "=tf()" and "=zpk()," which can accept input parameters defined in individual cells or a range of cells. An open and parameterized model is constructed by entering its parameters in separate cells. Consequently, any changes in these parameters update the model automatically. Furthermore, the parameters of a model can be obtained using Excel formulas. Figure 2 shows an example of three parameterized models.

Essential Control System Functions

COCT provides several functions for analyzing linear systems in both the time and frequency domains. The major topics included are time responses to common input signals such as impulse, step, ramp, and sinusoid; frequency responses using the Bode and Nyquist plots; and additional utilities such as root locus or pole-zero maps. In the COCT environment, the time and frequency responses are identified as plot objects, which are obtained by using specific functions such as "=step()," "=bode()," "=nyquist()," or "=rlocus()." These functions are used much like any of the Excel built-in functions. A plot object is displayed by making its cell active and running the corresponding simulation.

COCT also provides utilities for simulating feedback control systems.

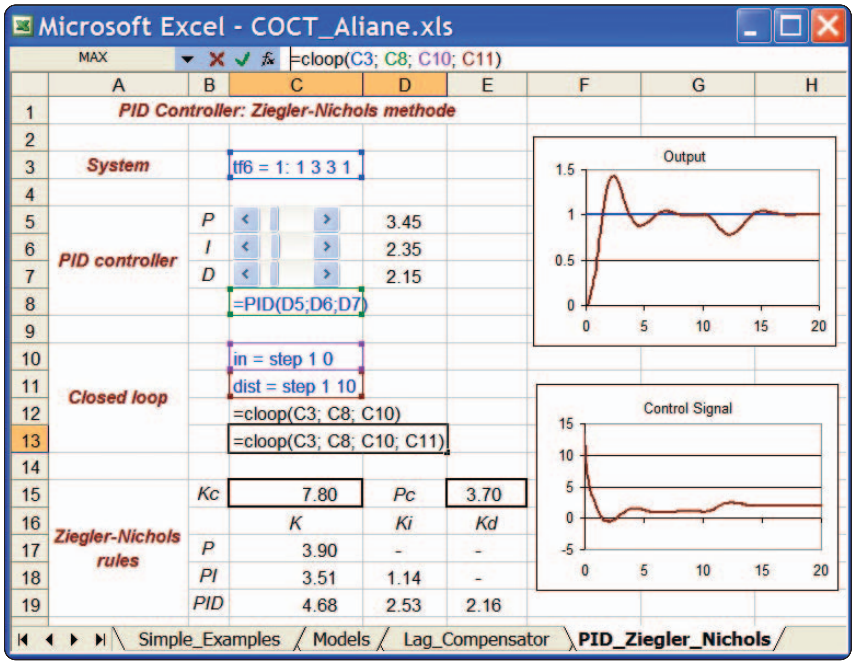
A closed-loop is defined by the "=cloop()" function, specifying the model, the controller, the input reference, and an optional external disturbance. PID controllers are of particular interest when studying control systems and constitute a required chapter in a basic control course. A PID controller is defined by the "=pid()" function specifying the three tuning parameters. The influence on the system behavior of each gain in the PID controller can be observed interactively by entering the tuning parameters in separate cells.

Furthermore, COCT provides a comprehensive tutorial on its use as well as a help section on how to use all of the supported functions.

Interactivity

The COCT environment supports interactivity in the sense that changes in any active element of a given simulation result in an immediate recalculation and presentation of the results. Thus, simulations with dynamic and animated graphs can be produced. This feature is achieved by focusing a cell containing a plot object, in which a programmed mechanism detects changes on the worksheet and subsequently runs the corresponding simulation.

To make the simulations even more interactive, elements such as sliders can be used to change the parameter values. This interface allows users to directly explore and understand the effect of varying parameters on the system behavior. For example, sliders can be used to investigate control system stability, analyze gain and phase margins, observe how the shape of both time and frequency responses are modified, adjust the controller's tuning parameters, and explore how the poles of a closed-loop system move along the root locus. In the COCT environment, the use of sliders does not require programming of any kind. The insertion of a slider into a worksheet is performed by selecting it from the (View->Toolbars->Control) menu and by dragging out an image. This feature makes the COCT environment highly interactive and enables users to perform "what-if" experiments in a straightforward manner.



Microsoft Excel - COCT_Aliane.xls

MAX | =cloop(C3; C8; C10; C11)

	A	B	C	D	E
1	PID Controller: Ziegler-Nichols methode				
2					
3	System		tf6 = 1: 1 3 3 1		
4					
5		P		3.45	
6	PID controller	I		2.35	
7		D		2.15	
8			=PID(D5;D6;D7)		
9					
10			in = step 1 0		
11			dist = step 1 10		
12	Closed loop		=cloop(C3; C8; C10)		
13			=cloop(C3; C8; C10; C11)		
14					
15		Kc	7.80	Pc	3.70
16			K	Ki	Kd
17	Ziegler-Nichols rules	P	3.90	-	-
18		PI	3.51	1.14	-
19		PID	4.68	2.53	2.16

FIGURE 4 PID control. The transfer function of the system $G(s) = 1/(s+1)^3$ is entered in cell C3. The PID controller is entered in cell C8, and its tuning parameters are entered in separate cells. The simulation of the closed loop is entered in C13. The controlled system is subject to an input reference (a unit step applied at t = 0 s), and to an external disturbance (a unit step applied at t = 10 s), which are entered in cells C10 and C11, respectively. Ziegler-Nichols rules are also entered as standard Excel formulas.

EXAMPLES

This section presents two examples that illustrate some of COCT's capabilities. Although these examples do not fully exploit all of the COCT features, they illustrate how the iterative process involved in the design of classical controllers is carried out interactively.

Example 1: Phase-Lag Compensation

The purpose of phase-lag compensation in the frequency domain is to satisfy specifications on steady-state accuracy and phase margin. The design process is graphical in nature and consists of two steps. The first step is to determine the gain of the compensator to satisfy the steady-state error specification, while the second step is to determine the compensator's pole and zero locations that satisfy the phase-margin requirement.

COCT offers utilities to simplify the phase-lag compensation design process and help students understand its characteristics. One such simulation is shown in Figure 3. To visualize the design process, the compensator parameters are entered in separate cells and linked to sliders. Additionally, Bode diagrams showing the gain and phase margins as well as the time response to a ramp input of both uncompensated and compensated systems are visualized. Dragging the sliders immediately updates the plots, and, consequently, students can obtain an immediate sense of how moving a compensator pole or zero location influences the gain and phase margins.

Example 2: PID Controller and Ziegler-Nichols Method

In this example, we consider tuning a PID controller using the Ziegler-Nichols rules. According to the margin stability method, a PID controller is first tuned by setting it to the proportional mode and adjusting the gain to make the system marginally stable. The final recommended values of the tuning parameters are obtained by finding the corresponding critical gain and the ultimate period of oscillation. This process is iterative and suitable for graphical treatment.

The Ziegler-Nichols method is illustrated graphically in the simulation shown in Figure 4. The PID parameters are entered in separate cells, and their values can be changed interactively. Ziegler-Nichols rules are also entered as Excel formulas. The design process is simplified by taking advantage of interactivity. The interaction is more straightforward due to the sliders, allowing students to clearly perceive changes in the performance. Furthermore, the user can improve the parameter tuning through the trial and error approach.

USAGE EXPERIENCE

In the 2006–2007 academic year, we projected COCT screenshots during lectures at Universidad Europea de Madrid. COCT capability of quickly generating multiple examples with plots of time and frequency responses proved to be helpful. Its interactive environment helps to draw comparisons between simulations, explore the effect of varying parameters, and give explanations through "what-if" scenarios. Many of the examples used in the lectures are collected in a single workbook. The fact that the COCT environment reduces the effort needed to specify and run simulations helps the instructor to proceed quickly. Finally, visibility of the entire worksheet highlights all of the parts of the simulations, and cells related to formulas are clearly indicated. This feature allows the instructor to quickly pose different cases in response to student questions.

The tool was also used by about 20 students as support material for exercises and as a basis for group discussions on more in-depth theoretical topics. Since the majority of students are familiar with Excel, they found COCT easy to learn and use. The students agreed that its most valuable feature is its simplicity in setting sliders to explore the effects of varying parameters on the system behavior. In this aspect, sliders are used for interactively exploring control concepts. This sort of interaction encourages students to pay more attention and consequently helps them to gain better understanding of the qualitative aspects of control concepts.

FUTURE DEVELOPMENTS

Although the COCT toolbox is in its infancy, the current version marks the first step in developing a more complete control systems software package for the Excel environment. For example, the COCT kernel can be used to design specific demonstrations and interactive modules. However, the most challenging prospect is to explore many of the Microsoft Excel facilities for extending COCT's capabilities, for which we have two projects underway. The first one consists of interfacing COCT with external libraries such as LAPAK or EISPAK for performing large calculations, and the second one is related to interfacing COCT with external hardware for performing real-time control.

CONCLUSIONS

A spreadsheet-based tool for control education has been presented. The tool is a COCT for working with linear systems in the Excel environment. The tool provides the essential utilities for linear system analysis, controller design, and simulation of closed-loop systems. Currently, COCT supports only single-input, single-output linear systems described by scalar transfer functions.

COCT supports multiple simulations and system analysis on a single worksheet and allows users to switch from one simulation to another by moving the focus from one cell to another. Thanks to its interactivity, users can define "what-if" scenarios, in which they can dynamically change the system parameters and observe their influence on the system behavior.

Since COCT is embedded in a single Excel file, it does not require installation or configuration of any kind. The COCT environment reduces the effort needed to specify and run simulations. Moreover, experience with Excel is not required to develop helpful simulations with a good degree of interactivity.

REFERENCES

[1] S. Dormido, "Control learning: present and future," *Annu. Rev. Contr.*, vol. 28, no. 1, pp. 115–136, 2004.

[2] B. Wittenmark, H. Hägglund, and M. Johansson, "Dynamic pictures and interactive learning," *IEEE Contr. Syst. Mag.*, vol. 18, no. 3, pp. 26–32, 1998.

[3] M. Johansson, M. Gäfvert, and K. J. Åström, "Interactive tools for education in automatic control," *IEEE Contr. Syst. Mag.*, vol. 18, no. 3, pp. 33–40, 1998.

[4] S. Dormido, S. Dormido-Canto, R. Dormido-Canto, J. Sanchez, and N. Duro, "The role of interactivity in control learning," *Int. J. Eng. Edu.*, vol. 21, no. 6, pp. 1122–1133, 2005.

[5] J.L. Guzmán, K.J. Åström, S. Dormido, T. Hägglund, and Y. Piguet, "Interactive learning modules for PID control," in *Proc. 7th IFAC Symp. Advances in Control Education,* Madrid, Spain 2006 {CD-ROM].

[6] R.C. Garcia and B.S. Heck, "Enhancing classical controls education via interactive GUI design," *IEEE Contr. Syst. Mag.,* vol. 19, no. 3 , pp. 77–82, 1999.

[7] V. Kroumov, K. Shibayama, and A. Inoue, "Interactive learning tools for enhancing the education in control systems," in *Proc. 33rd ASEE/IEEE Frontiers in Education Conf.*, Boulder, CO, session T4E, 2003, pp. 23–27.

[8] Sysquake [Online]. Available: http://www.calerga.com/products/Sysquake/index.html

[9] D.M. Bourg, *Excel Scientific and Engineering Cookbook*, 1st ed. O'Reilly, 2006

[10] C. Rives and D.J. Lacks, "Teaching process control with a numerical approach based on spreadsheets," *Chem. Eng. Edu.*, vol. 36, no. 4. pp. 242–248, 2002.

[11] Chehab and A. El-Hajj, "Spreadsheet applications in electrical engineering: A review," *J. Comput. Applicat. Eng. Edu.*, vol. 20, no. 6, pp. 902–908, 2004.

[12] S.A. Oke, "Spreadsheet applications in engineering education: A review," *J. Comput. Applicat. Eng. Edu.*, vol. 20, no. 6, pp. 893–901, 2004.

[13] M. Schumack, "Use of a spreadsheet package to demonstrate fundamentals of computational fluid dynamics and heat transfer," *J. Comput. Applicat. Eng. Edu.*, vol. 20, no. 6, pp. 974–983, 2004.

[14] T.S. Hanania and I. Levin, "Spreadsheet-based logic controller for teaching fundamentals of requirements engineering," *Int. J. Eng. Edu.*, vol. 20, no. 6, pp. 939–948, 2004.

[15] A. El-Hajj, K.Y. Kabalan, and S. Khoury, "A linear control system simulation toolbox using spreadsheets," *IEEE Contr. Syst. Mag.*, vol. 20, no. 6, pp. 8–14, 2000.

[16] A. El-Hajj, S. Karaki, and K.Y. Kabalan, "Digital control systems simulation using

spreadsheets," *J. Comput. Applicat. Eng. Edu.*, vol. 11, no. 1, pp. 6–12, 2003.
[17] A. 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.
[18] COCT workbook [Online]. Available: http://www.esp.uem.es/aliane/coct/coct.zip

AUTHOR INFORMATION

Nourdine Aliane (nourdine.aliane@uem.es) 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.

Advice to Young Researchers

MALCOLM D. SHUSTER

A wise man can learn from anyone,
even from a fool, but a fool can
learn from no one, not even
from a wise man.
—Jewish proverb

It is easier to give advice than to
bear one's own problems.
—Euripides (480–406 BCE)

A new researcher in any discipline is often at sea without a compass. He may have demonstrated tremendous skill already as a graduate student in the ability to achieve new results in research, but is unaware of the ethical requirements of research or of many activities that can improve the quality of his work and protect that work to a degree from serious errors. I am sorry to say that too many supposedly seasoned researchers are unaware of these errors or choose to ignore them. For the engineer there is the additional requirement that his research should have some connection to the real world, particularly on how one finds research problems in it. The present essay is not intended as a survivor's manual. It has loftier goals for those young creative researchers who, to steal a phrase from Faulkner [1], will not only survive but prevail, whether in industry, government, or academia. The present article is adapted from part of an earlier article [2].

Digital Object Identifier 10.1109/MCS.2008.929065

A BASIC RULE

Work hard and do good work! There is no substitute for this nor compensating factors. Also, to do really good work, you must be a little crazy. As the Romans said: *nullum magnum ingenium sine mixtura dementiae fuit* (there has not been great genius without an admixture of insanity), which need not imply that if you are a bit crazy, you will become a great genius, but it helps. But genius is not enough. My own career is certainly proof that persistence and hard work can compensate for lack of genius, unless one accepts Edison's definition of genius as one percent inspiration and ninety-nine percent perspiration, although my own ratio is certainly much poorer than Edison's.

RESEARCH

Be focused! It is important, certainly, that your research have breadth. If you are just starting out, breadth will get you better job offers, or present you with a greater range of areas in which to obtain research funding, but proven depth is more likely to get you fame. If you are a young academician, both will help you get tenure. It is important that there be at least one area where you have real depth, although it may take a long time to develop.

Put not thy trust in drawings! Drawings are very helpful in research. They excite our visual perception of the problem at hand, give us new insights, and are excellent ancillary devices for communication and teaching as well as mnemonics for important results. But drawings are also fraught with dangers, since it is often very difficult to interpret signs correctly from drawings. The misinterpretations can sometimes be very subtle. When my calculation almost "works" except for a sign inconsistency that I think I can fix later, that is usually the time for me to start doing the algebra microscopically. A word to the wise…

Put not thy trust in others! Never trust a published formula. Always rederive it yourself. Not only do published formulas sometimes contain errors (*mea culpa*), but we often understand the range of application of a formula only by deriving it ourselves and seeing then the hidden assumptions. If an article is not particularly well written, be prepared that there may be communication problems in the results as well. Always save your notes.

Be real! The best research often comes from real problems. I think that much academic research is, well, academic. If you are a young academician, form close ties with a local company or with a government facility. You may find that your best ideas, even general theoretical ideas, arise from their needs. (My most important research has often been research and development.) These contacts may also become a good source of research funding, summer salaries, and jobs for your students.

Don't always be practical! Sometimes impractical research can be very