

10-5-2005

Nonlinear Control Systems Simulation Using Spreadsheets

Ali El-Hajj

American University of Beirut, Lebanon, elhajj@aub.edu.lb

Sami Karaki

American University of Beirut, Lebanon

Karim Y. Kabalan

American University of Beirut, Lebanon

Follow this and additional works at: <http://epublications.bond.edu.au/ejsie>



This work is licensed under a [Creative Commons Attribution-Noncommercial-No Derivative Works 4.0 License](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Recommended Citation

El-Hajj, Ali; Karaki, Sami; and Kabalan, Karim Y. (2005) Nonlinear Control Systems Simulation Using Spreadsheets, *Spreadsheets in Education (eJSiE)*: Vol. 1: Iss. 2, Article 2.

Available at: <http://epublications.bond.edu.au/ejsie/vol1/iss2/2>

This Regular Article is brought to you by the Bond Business School at [ePublications@bond](mailto:epublications@bond.edu.au). It has been accepted for inclusion in Spreadsheets in Education (eJSiE) by an authorized administrator of [ePublications@bond](mailto:epublications@bond.edu.au). For more information, please contact [Bond University's Repository Coordinator](#).

Nonlinear Control Systems Simulation Using Spreadsheets

Abstract

In this paper, a method for simulating nonlinear control systems using spreadsheets is presented. Various nonlinear blocks are simulated using graphics and cell formulas, and are generated by clicking on specially developed toolbar buttons. These blocks can be connected to one another using a simple and intuitive procedure again based on graphics and toolbar buttons. A complete nonlinear system can thus be created by generating and connecting its constituting basic blocks, using the simple graphics interface provided. The corresponding data may then be entered in the familiar manner as illustrated, and finally the system can be simulated literally at the click of a button. Such a system can be analyzed by calculating its time response to any input signal or by using other methods such as phase-plane trajectories. The simulation is characterized by its availability, flexibility, and simplicity. The paper provides several examples to illustrate the simulation capabilities available. The first example considers a servo with a dead-zone and a saturation amplifier, the second illustrates the steps required to obtain a phase-plane trajectory, and the third example considers a nonlinear system having a PI controller and nonlinearity consisting of soft saturation. The final example illustrates a relay-controlled servo system.

Keywords

spreadsheets, nonlinear systems, control, circuits

Distribution License



This work is licensed under a [Creative Commons Attribution-Noncommercial-No Derivative Works 4.0 License](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Nonlinear Control Systems Simulation Using Spreadsheets

Ali El-Hajj, Sami Karaki, and Karim Y. Kabalan
Electrical and Computer Engineering Department
Faculty of Engineering and Architecture
American University of Beirut
P.O.Box 11-0236, Beirut, Lebanon 1107-2020
Tel: 961-1-353465 ext 3530,
Fax : 961-1-744462,
Email: elhajj@aub.edu.lb

October 5, 2005

Abstract

In this paper, a method for simulating nonlinear control systems using spreadsheets is presented. Various nonlinear blocks are simulated using graphics and cell formulas, and are generated by clicking on specially developed toolbar buttons. These blocks can be connected to one another using a simple and intuitive procedure again based on graphics and toolbar buttons. A complete nonlinear system can thus be created by generating and connecting its constituting basic blocks, using the simple graphics interface provided. The corresponding data may then be entered in the familiar manner as illustrated, and finally the system can be simulated literally at the click of a button. Such a system can be analyzed by calculating its time response to any input signal or by using other methods such as phase-plane trajectories. The simulation is characterized by its availability, flexibility, and simplicity. The paper provides several examples to illustrate the simulation capabilities available. The first example considers a servo with a dead-zone and a saturation amplifier, the second illustrates the steps required to obtain a phase-plane trajectory, and the third example considers a nonlinear system having a PI controller and nonlinearity consisting of soft saturation. The final example illustrates a relay-controlled servo system.

Submitted October 2003; revised and accepted January 2004.

Keywords: Spreadsheets, nonlinear systems, control, circuits.

1 Introduction

Automatic control is an essential part of industrial processes, robotic systems, power generation, transportation and many other modern systems, which are nonlinear to various extents. In some cases, when variations around an operating point of a system are small, the system can be linearized over a small range of change. Other systems with

EL HAJJ, KARAKI AND KABALAN

low degree of nonlinearity can be analyzed using a method known as describing-function analysis. However there are systems with high-degree of nonlinearity that may be inherent like saturation, hysteresis, and dead-zones, or intentionally introduced to improve the performance of such systems like on-off relays. Procedures for finding solutions of such systems are extremely complicated, and thus their behavior is best predicted by simulation methods.

A method for simulating linear and nonlinear systems using spreadsheets was introduced in [1], where a system is divided into its basic building blocks, which are simulated as boxes with inputs and outputs related by formulas. Two blocks are connected by writing in the input cell of one block a formula equal to the output cell address of the other, and by drawing a line that connects the centers of the two cells. A complete system can thus be simulated by generating and connecting its building blocks.

The same method was later used to simulate logic networks and linear control systems. The basic blocks used in simulating logic networks [2] are logic gates, flip-flops, the clock and some medium scale integration circuits. This method was applied to the simulation of combinational, sequential, synchronous and asynchronous networks. The basic blocks used in simulating linear control systems are an adder, an integrator, a first and a second order system [3]. Any transfer function $F(s)$ in the s -domain can be obtained by connecting a number of these blocks with appropriate scaling coefficients. A large control system is thus simulated by connecting its constituent transfer functions and calculating its time response for any input signal. A digital control system toolbox was introduced in [4] to simulate a sampled data control system using its transfer function $T(z)$ in the z -transform domain. The basic blocks used are a delay z^{-1} , a first order system, and a second order system. Any transfer function in the z -domain can thus be obtained by connecting a number of these blocks with appropriate coefficients. Each of the basic blocks can be obtained by clicking a corresponding button on the toolbar, and then by writing in appropriate cells the block coefficients values. Toolbar buttons similar to the linear control case are used for connecting these blocks and adding or subtracting the outputs of two or more blocks. Other toolbar buttons are also used to initialize the worksheet or to run it for a certain number of iterations.

In a recent work [5], a nonlinear control toolbar was constructed to graphically simulate many nonlinear devices. The basic blocks used are relays, saturation amplifiers, dead-zones, and backlash nonlinearities. Each of these nonlinear blocks can be obtained by clicking a corresponding button on the toolbar. This method has the following advantages.

1. It is of low cost since spreadsheets are software packages that are widely available.
2. It is easy to learn and apply by a wide audience of users, especially students who are familiar with spreadsheet programs.
3. It allows the study of the behavior of a system by running the worksheet step-by-step or many steps at a time. It also permits the study of the influence of changing one or more parameters on the system behavior.

NONLINEAR CONTROL SYSTEMS SIMULATION USING SPREADSHEETS

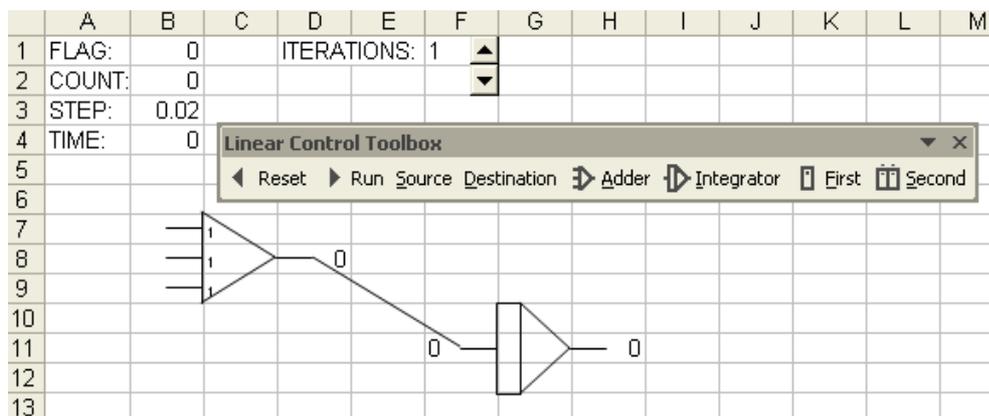


Figure 1: “Linear control toolbox” Toolbar

4. It is suitable for quick simulation when one does not have the time or means to write a sophisticated program or access a sophisticated package.
5. It is readily extensible as new nonlinear devices can be simulated by simply editing or adding to the formulas used for the existing blocks.
6. It allows a graphical display of the results using the full power of the Excel chart toolbar.
7. Any task can be recorded in a macro, thus reducing programming efforts to a minimum.

This method is suitable for graduate and undergraduate students taking courses in control systems, and has been used by students in assignments to study the behavior of control systems via modeling and simulation. Generally it is very helpful, especially in situations with limited budget when more expensive simulation packages cannot be afforded.

In this work, the linear and nonlinear toolbars will be presented and applied to analyze nonlinear systems in the time domain and in the phase plane. The process of generating the basic blocks in the linear control toolbox and connecting them using spreadsheet tools is first explained in some detail in Section 2. The nonlinear control toolbox along with the various devices it presently implements is presented in Section 3. Four simulation examples to illustrate the various nonlinear devices and the study capabilities of the nonlinear toolbox are then presented in Section 4, and the results of these simulations are compared with published ones. Concluding remarks are finally given in Section 5.

2 The “Linear Control Toolbox” Toolbar

Several simulation tools introduced in [3] are used in simulating linear control systems. These tools are attached to buttons shown on the “linear control toolbox” toolbar in Figure 1. These buttons perform the following functions:

1. **Reset and Run buttons.** Studying a simulated system requires initializing some variables and running this system for a certain number of iterations (i.e., worksheet calculations). This can be done using an initialization flag stored in cell B1, an iteration counter stored in cell B2, an integration step stored in cell B3, and the required number of iterations stored in cell F1. The initialization flag is set to zero to indicate the initial state of the simulated system. It is used by some blocks to set their state to an initial condition. The worksheet is calculated a number of times indicated in cell F1 only when this flag is set to one. The iteration counter is initialized to zero and incremented after each worksheet calculation by writing in cell B2 the formula: $=IF(\$B\$1=0,0,B2+1)$. The time is calculated in cell B4 by multiplication of the iteration counter and the integration step using the formula: $=\$B\$2*\$B\3 . The required number of iterations can be set using a spinner in cells F1:F2. Initialization is automated using the “Reset” button. This button resets the initialization flag to zero. When initializing a worksheet is done for the first time, the spinner is generated and initialized to a default value. The “Run” button sets the initialization flag to one and calculates the worksheet a number of times indicated by the spinner. It is possible to run the worksheet many consecutive times using different number of iterations values.
2. **Source and destination buttons.** The simulation requires connecting a source cell to a destination one. This is done by drawing a line that connects the centers of the two cells and by writing in the destination cell a formula equal to the source cell address. This is done by locating the source cell and clicking the “Source” button, then by locating the destination cell and clicking the “Destination” button. As an example, to connect the two blocks shown in Figure 1, cell D8 is made active and the “Source” button is clicked. Cell F11 is then made active and the “Destination” button is clicked.
3. **The basic blocks buttons.** These buttons are used to generate adders, integrators, first order, and second order transfer functions. The “Adder” button causes a three-input adder with scaling coefficients to be generated. The scaling coefficients are assigned default values of one, which can be directly changed on the worksheet by the user. The Integrator button causes an integrator to be generated. The first order and second order buttons cause the generation of first order or second order transfer function in the s -domain with default parameter values, which can be directly changed on the worksheet by the user. As an example, to generate the adder with inputs B7, B8, and B9 shown in Figure 1, cell B7 is made active and the “Adder” button is clicked. The scaling coefficients in cells C7, C8, and C9 are assigned default values of one, which can be directly changed on the worksheet

NONLINEAR CONTROL SYSTEMS SIMULATION USING SPREADSHEETS

by the user. The output D8 contains the formula: $=B7*C7 + B8*C8 + B9*C9$. As another example, to generate the integrator with input F11 and output H11 shown in Figure 1, cell F11 is made active and the “Integrator” button is clicked.

Every described button is attached to a macro. In order to minimize the programming effort while constructing the toolbar, a macro is first recorded to perform partially or completely the required task, then edited if necessary to complete the task. As an example, the macro attached to the “Adder” button can be obtained in the following way:

1. The drawing toolbar is used to draw an adder with three inputs and one output.
2. Formulas are used to calculate the adder output as function of the inputs and scaling coefficients.
3. The previous two steps are recorded in a macro. The macro code, generated in Visual Basic Applications (VBA), is edited to ensure that all drawings and references are relative to the current active worksheet cell that is considered as the first input cell.
4. The macro is then attached to the “Adder” toolbar button.

The same method used to simulate the adder is used to simulate the integrator, the first order and the second order blocks [3]. The same method will also be used in the next section to model and simulate the operation of nonlinear devices.

The “Nonlinear Control Systems” Toolbar

The nonlinear devices presented in this work are relays, saturation amplifiers, dead-zones, and backlash nonlinearities [5], and their input-output characteristics are shown in Figures 2–5. For example, relays are used in servo-motor positioning systems, or in industrial processes for the control of temperature or flow. Saturation is a common property of amplifiers, widely used in control systems, when their input signal exhibits large variation. Dead zones are used to represent the insensitivity of some components to small signals. Backlash or hysteresis, for example, is used to model slack in the meshing of gears, residual magnetism effects in motors or transformers, and static or Coulomb friction. In some systems hysteresis is intentionally introduced to reduce chatter in relays and stabilize their operation.

Consider the relay shown in Figure 6 with input A6 and output C6. Its output levels are stored in cells B5 and B7, which are assigned default values of +1 and –1 that may be changed directly on the worksheet by the user. The formula of its output in cell C6 is as follows:

$$= \text{IF}(A6 < 0, B7, \text{IF}(A6 > 0, B5, 0))$$

The block simulation approach explained in Section 2 above is used for the relay. The actions of drawing the relay and writing its formulas are recorded in a macro, which

EL HAJJ, KARAKI AND KABALAN

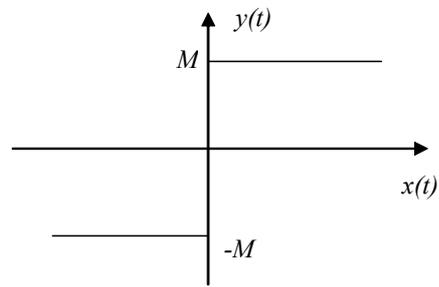


Figure 2: Relay

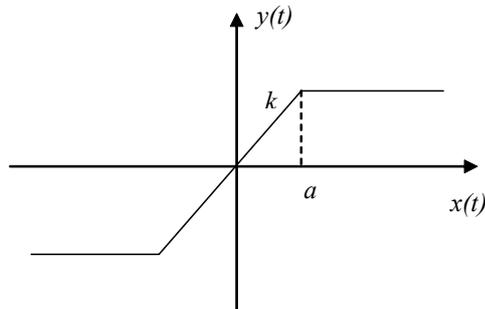


Figure 3: Saturation Amplifier

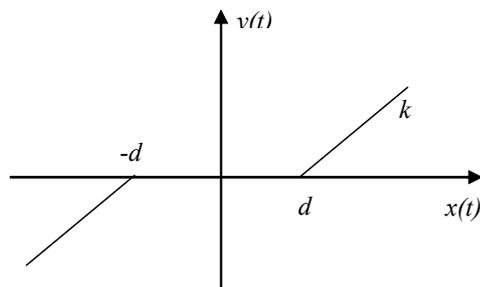


Figure 4: Dead-zone

NONLINEAR CONTROL SYSTEMS SIMULATION USING SPREADSHEETS

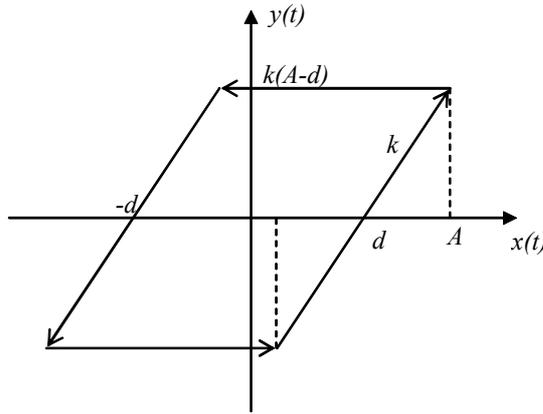


Figure 5: Backlash

is edited, to make cell addresses and formulas relative, and is then attached to a “Relay” button created on the “Nonlinear Control Systems” toolbar. In order to generate a relay at a certain location, the cell of the input is made active and the “Relay” button is clicked. A similar approach is used to simulate a saturation amplifier with input E6 and output G6 shown also in Figure 6. The slope k is stored in cell F6, and the saturation range parameter a in cell F7. The output G6 contains the following formula:

$$=IF(E6 < -F7, -F7*F6, IF (E6 > F7, F7*F6, E6*F6))$$

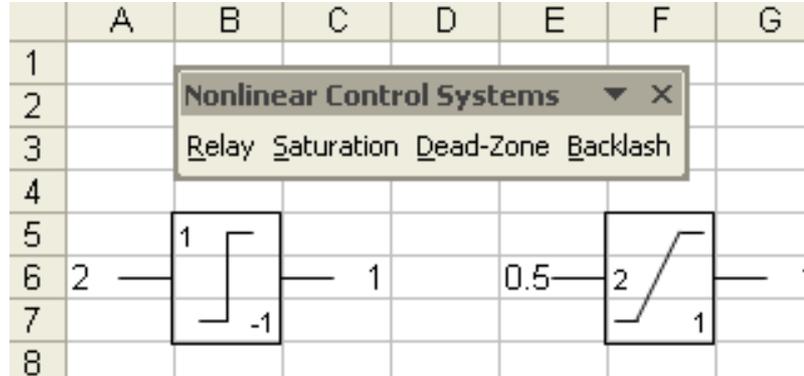


Figure 6: Relay and Saturation Simulation

As in the relay case, the saturation amplifier generation process is automated using a specially created “Saturation” button. A dead-zone simulation is shown in Figure 7, where the slope k and width parameter d are stored in cells C5 and C7, respectively. Its output is in cell D6 that contains the following formula:

$$=IF(B6 < -C7, C5*(B6 + C7), IF(B6 > C7, C5*(B6 - C7), 0))$$

The dead-zone generation process is also automated using a “Dead-Zone” button.

EL HAJJ, KARAKI AND KABALAN

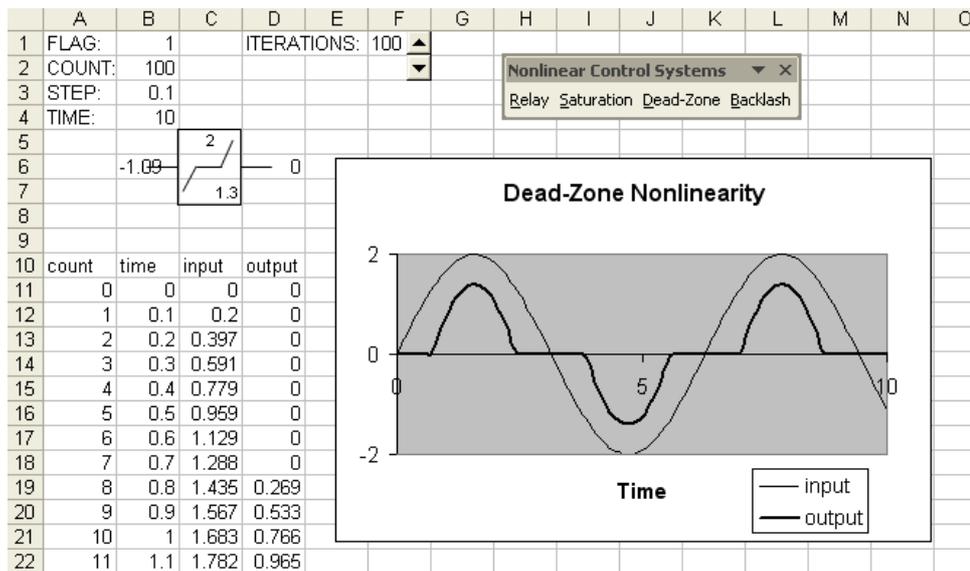


Figure 7: Dead-zone Simulation

The response of a dead-zone to the function $f(t) = 2\sin(t)$ is calculated for $k = 2$ and $d = 1.3$. This input is applied by writing in cell B6 the formula $=2*\text{SIN}(B4)$. To plot the results for a study of duration 10 seconds, a total of 100 iterations were carried out with a time step of 0.1 seconds and the range A11:A111 is filled with values from 0 to 100. The time values are obtained in range B11:B111 by writing in cell B11 the formula $=A11*\$B\3 and copying it to B12:B111. The successive input values are transferred from cell B6 to C11:C111 by writing in cell C11 the following formula:

$$=IF(\$B\$1=0, 0, IF(\$B\$2 = A11, \$B\$6, C11))$$

This formula is copied into C12:C111, and in the same way the output values are transferred from cell D6 to the range D11:D111. The spreadsheet plotting facilities are then used to plot the results.

The backlash simulation is shown in Figure 8, where the slope k , width d , and parameter A are stored in cells D8, D9 and D10, respectively. In order to calculate the output, the current and previous input values are needed. The input used in the previous iteration is stored in cell C8 using the formula: $=C9$. In this case, due to the worksheet order calculation by rows, C8 is assigned the previous-input before a new value is assigned to C9. The output in cell E9 contains a formula that implements the following pseudocode:

NONLINEAR CONTROL SYSTEMS SIMULATION USING SPREADSHEETS

If (previous-input < current-input)
 If (current-input < $2d - A$) then output is $-k(A - d)$
 Else output is $k(\text{current-input} - d)$
 Else
 If (current-input > $-2d + A$) then output is $k(A - d)$
 Else output is $k(\text{current-input} + d)$

The spreadsheet formula that implements the above pseudocode in cell E9 is given by:

`=IF(C8<C9, IF(C9<C10, -E8, D8*C9-D8*D9), IF(C9>-C10, E8, D8*C9+D8*D9))`

The value of $2d - A$ is calculated in cell C10 using the formula: `=2*D9-D10`, and the value of $k(A - d)$ is calculated in cell E8 using the formula: `=D8*(D10-D9)`.

In a similar manner, the implementation of a backlash is shown in Figure 8 with parameters $k = 1$, $d = 1.5$, and $A = 2$ and an input function $f(t) = 2\sin(t)$.

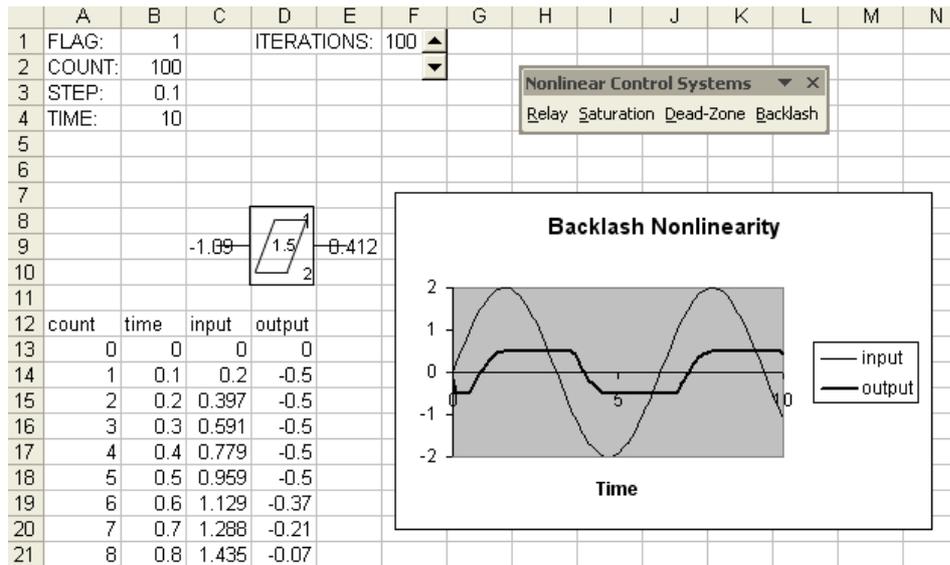


Figure 8: Backlash Simulation

Similarly, it is possible extend the toolbar to implement and simulate additional non-nonlinear devices. In some cases, a nonlinear function can be used by direct implementation in the required cell using the spreadsheet Function Editor, or they can be defined as a set of points defined in tables that are accessed using spreadsheet lookup functions.

EL HAJJ, KARAKI AND KABALAN

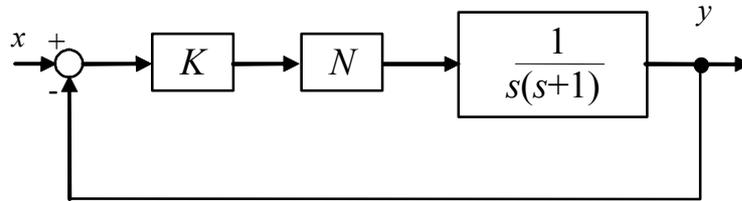


Figure 9: Block diagram of a servo system with non-linearity

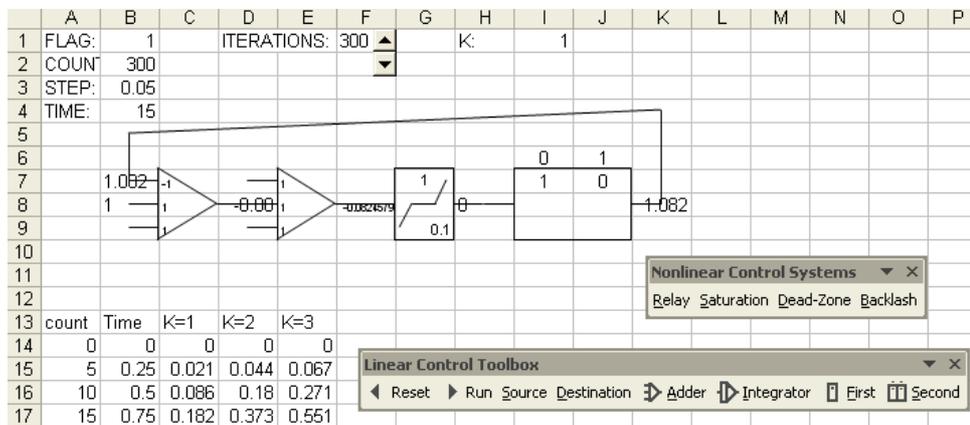


Figure 10: Spreadsheet simulation of the servo system.

NONLINEAR CONTROL SYSTEMS SIMULATION USING SPREADSHEETS

3 System Simulation Examples

In this section, three nonlinear control systems are simulated, and time responses and phase-plane trajectories are calculated. As a first example, consider the block diagram of a servo system with nonlinearity N representing a dead-zone with $d = 0.1$ and a slope $k = 1$, as shown in Figure 9. The spreadsheet simulation of this system is shown in Figure 10. In this simulation, the “Linear Control Toolbox” toolbar was used to generate the two adders and the second order transfer function. The unit step response of the system for different values of the gain K is plotted in Figure 11. The results are in agreement with those obtained in [6].

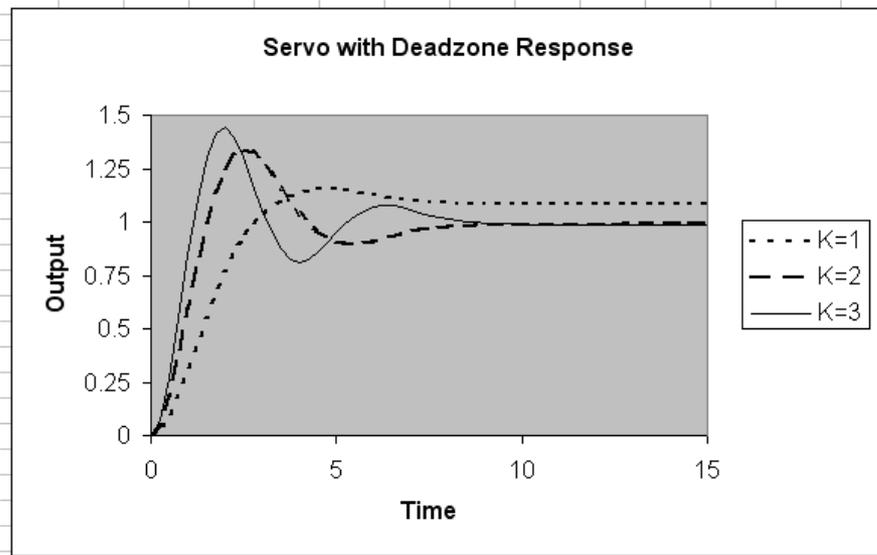


Figure 11: Step response of the servo system.

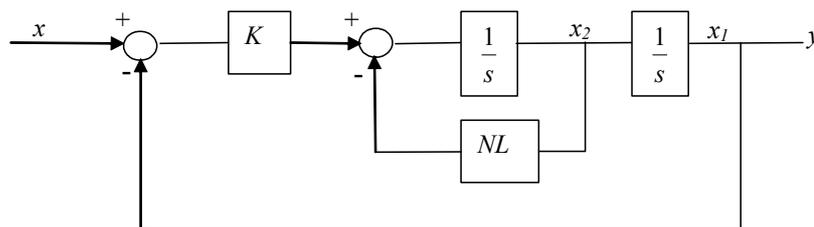


Figure 12: Block diagram of horizontal mass-spring system with friction

As a second example, consider the block diagram of a horizontal mass-spring system with friction as shown in Figure 12. In this system, friction is represented by a relay NL

EL HAJJ, KARAKI AND KABALAN

with levels -1 and $+1$. The spreadsheet simulation of this system is shown in Figure 13. This system is studied and in the phase-plane (x_1, x_2) , where the variables x_1 and x_2 are represented by cells M15 and J11. As in the first example, the path followed in simulating each feedback loop takes into consideration the row-order of worksheet calculation. In order to obtain a phase-plane trajectory, two steps are required. In the first step, in order to initialize at a certain location in the plane (x_1, x_2) , the input is assigned a value of 4 (or any other value) and the worksheet is calculated a number of times equal to 59 with an integration step of 0.05. In the second step, the input is set to zero, and the worksheet is calculated until the final point is reached. The values of x_1 and x_2 obtained in the second step are recorded and plotted in Figure 14. The results are in agreement with the ones obtained in [7].

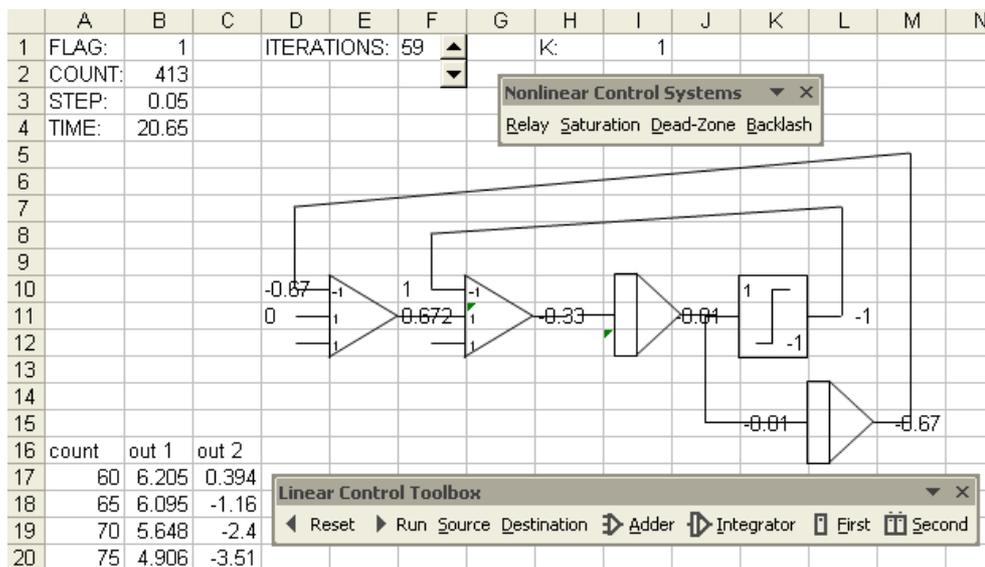


Figure 13: Spreadsheet simulation of the mass-spring system.

As a third example, consider the nonlinear system of Figure 15 which consists of a PI controller and a second-order Type-0 plant. The nonlinearity N consists of a soft saturation function shown in Figure 16, which is not available in the toolbar but can be obtained by direct implementation in the required cell using the spreadsheet Function Editor. The example is simulated in Figure 17, and the soft saturation is obtained by writing in cell K9 the following formula:

$$=IF(I9<1.83, 1.75*I9, 0.175*I9+2.7)$$

The response to the input indicated in Figure 18 is in agreement with that obtained in [6]. A smooth saturation can be simulated by directly entering its modeling non-linear function in cell K9, however, the piece-wise linear function was used for comparison purposes only.

The relay-controlled position servo system shown in Figure 19 is finally simulated in

NONLINEAR CONTROL SYSTEMS SIMULATION USING SPREADSHEETS

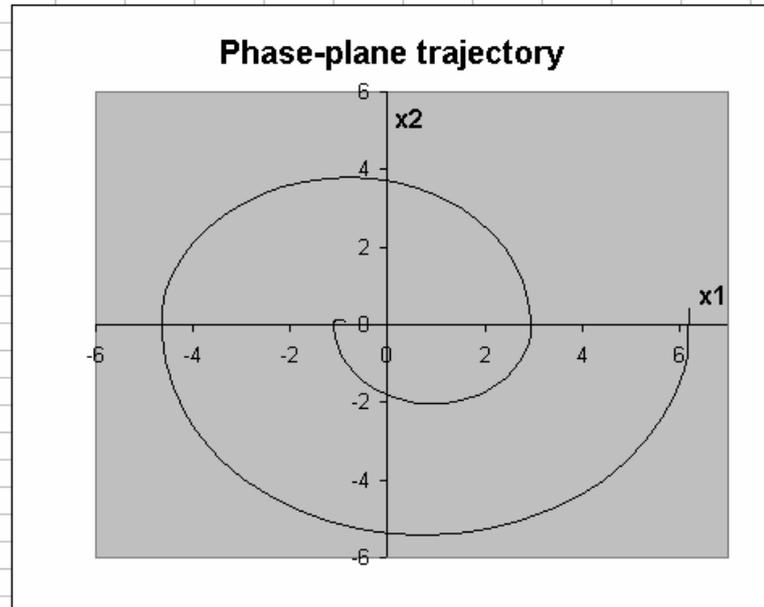


Figure 14: Phase-Plane plots of the mass-spring system.

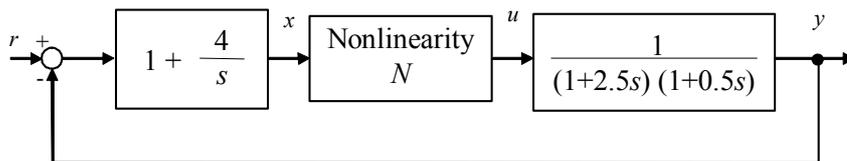


Figure 15: Block diagram of a PI-controlled second-order system

EL HAJJ, KARAKI AND KABALAN

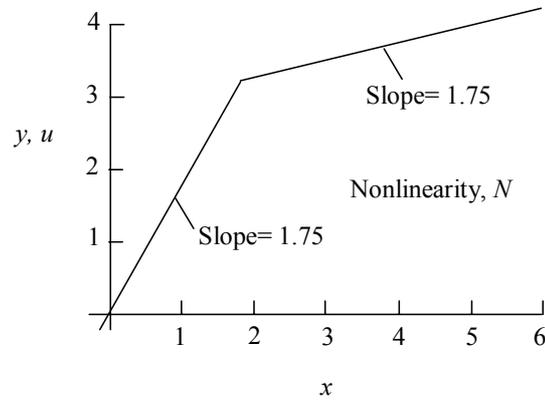


Figure 16: Soft saturation function

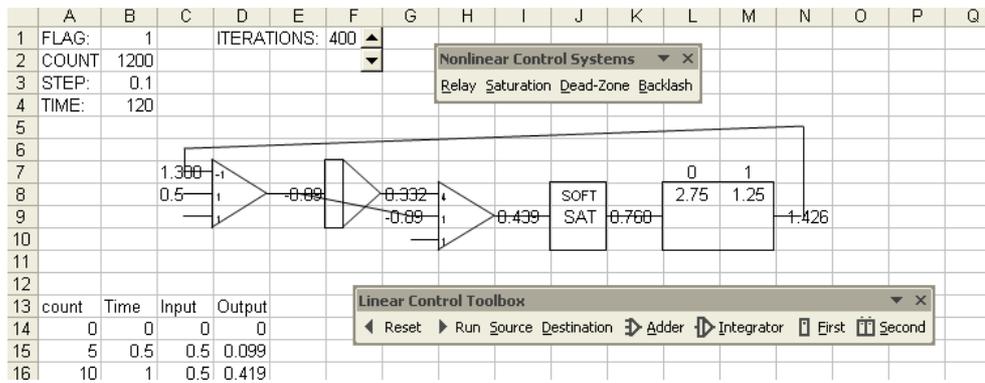


Figure 17: Spreadsheet simulation of the PI-controlled second-order system.

NONLINEAR CONTROL SYSTEMS SIMULATION USING SPREADSHEETS

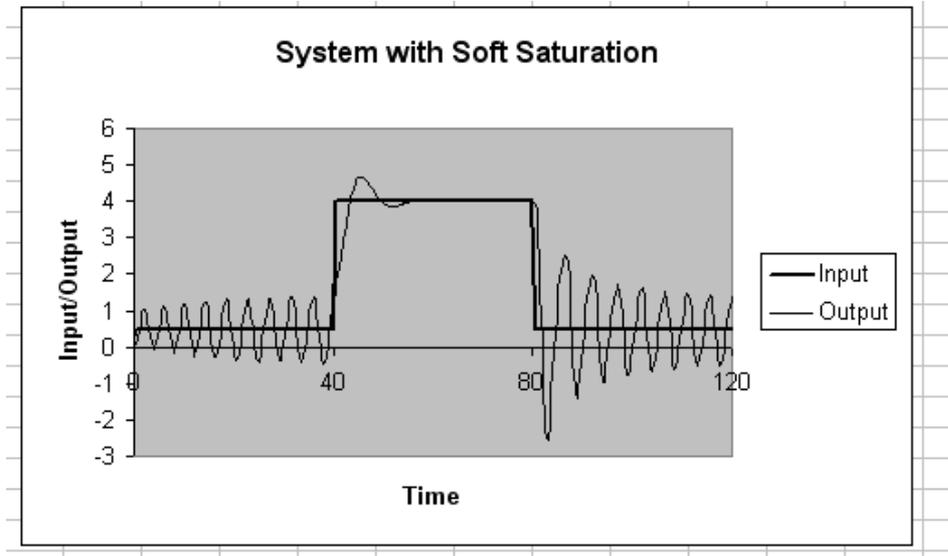


Figure 18: Responses of the PI-controlled second-order system

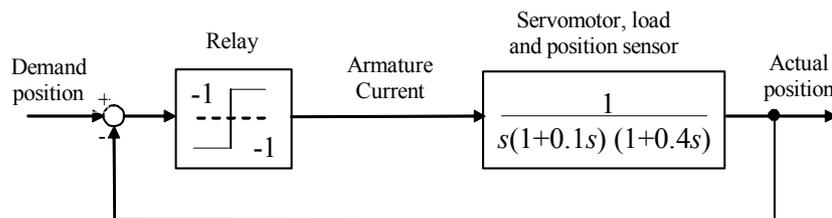


Figure 19: Block diagram of a relay-controlled position servo-system

EL HAJJ, KARAKI AND KABALAN

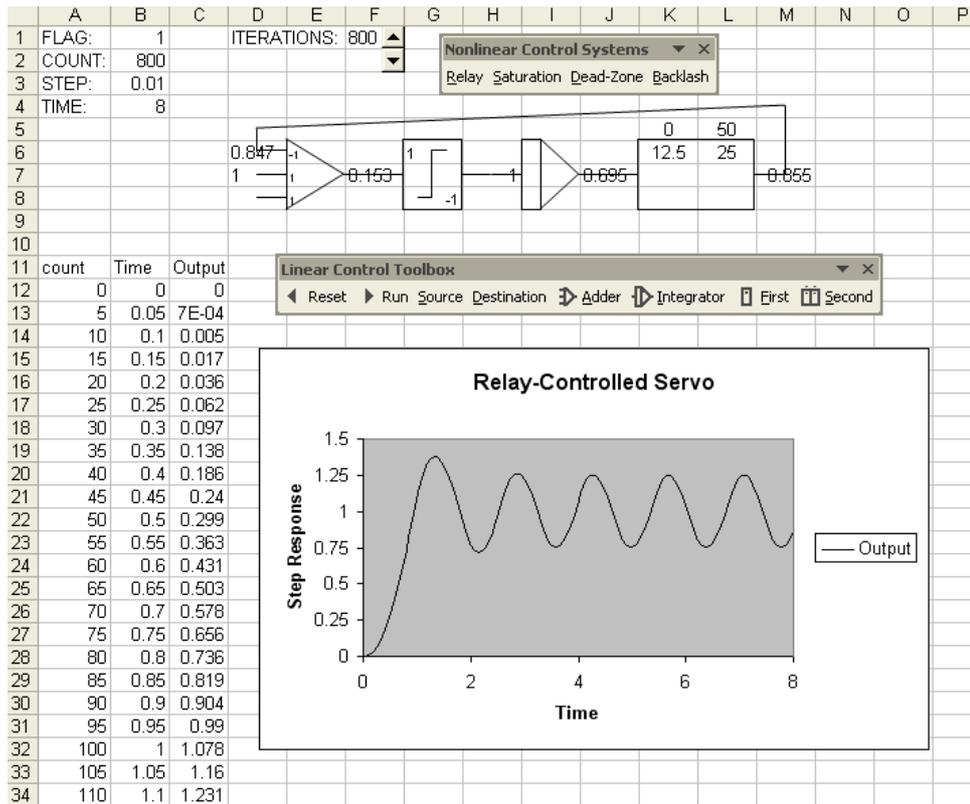


Figure 20: Spreadsheet simulation of the relay-controlled position servo-system.

NONLINEAR CONTROL SYSTEMS SIMULATION USING SPREADSHEETS

Figure 20. The relay switches a current of ± 1 A into the armature of the motor. The step response of the closed-loop system to a unit demand is also shown in Figure 20. Its step response with the resulting limit cycle is also in agreement with the results obtained in [6].

4 Conclusion

A method for simulating nonlinear control systems using modern spreadsheet programs has been presented. This method is simple and easy to learn, and can readily be extended to model additional nonlinear devices. It has been used by students in assignments to study the behavior of control systems through modeling and simulation. In general, it can be used for quick simulation checks of simple to moderate systems when one does not have the time or means to write a sophisticated program or does not have access to a sophisticated package. The built in ability to readily change system parameters and carry out “what if” type of analysis, will help the student or professor develop insight of the problem being solved. It allows running a system step by step, or in many steps at a time. The user can input, collect, and analyze data using all spreadsheets features. Several examples were presented to illustrate the available simulation capabilities. The examples illustrated various types of nonlinearities including a dead-zone, a saturation amplifier, the steps required to obtain a phase-plane trajectory, and nonlinear system having a PI controller and soft saturation. A final example illustrated a relay-controlled position servo system. The ease of use of the developed tools was clearly evident and the results obtained were identical to those obtained in the respective references using more sophisticated packages.

References

- [1] El-Hajj, A. (1999). Functional Simulation Using Spreadsheets. *SIMULATION*, **73**(2): 80–90.
- [2] El-Hajj, A., and Hazim, M. (1998). On Using Spreadsheets for Logic Networks Simulation. *IEEE Transactions on Education*, **41**(4): 311–319.
- [3] El-Hajj, A., Kabalan, K.Y., and Khoury, S. (2000). A Linear Control System Simulation Toolbox Using Spreadsheets. *IEEE Control Systems Magazine*, **20**(6): 8–14.
- [4] El-Hajj, A., Karaki, S., and Kabalan, K.Y. (2003). Digital Control Systems Simulation Using Spreadsheets. *Journal of Computer Applications in Engineering Education*, **11**(1): 6–12.
- [5] El-Hajj, A. (2001). A Nonlinear Control System Simulation Toolbox Using Spreadsheets. *IASTED International Conference on Modelling and Simulation 2001*, 83–87, May 16–18, 2001, Pittsburgh, U.S.A.

EL HAJJ, KARAKI AND KABALAN

- [6] Golten, J., and Verwer, A. (1991). *Control System Design and Simulation*, London: McGraw-Hill, 352–368.
- [7] Quenec'hdu, Y., and Viault, D. (1978). *Systemes asservis non lineaires*, Ecole Supérieure d'Electricite, No.2581.