Enhancing Student Understanding of Concepts in Engineering and other STEM Areas using Spreadsheets

Authors:

Martin R. Varley, University of Central Lancashire, UK mrvarley@uclan.ac.uk
Michael S. Peak, University of Central Lancashire, UK mpeak@uclan.ac.uk
Philip D. Picton, The University of Northampton, UK phil.picton@northampton.ac.uk
Fraser Hatfield, University of Central Lancashire, UK fraser.hatfield@btinternet.com

Abstract —It is widely recognised that practical experience reinforces students' understanding of concepts introduced in the classroom. This is particularly applicable to engineering education, where students' firsthand experience of design, analysis and simulation, at various levels appropriate to the particular topic, can assist their deeper and more complete understanding of concepts in engineering and other STEM (Science, Technology, Engineering and Mathematics) areas. Whilst the use of specialist software packages can lead to quick simulation results, the authors have found that this approach can often enable, and even encourage, students to bypass the technical detail, and can therefore compromise their in-depth understanding. In several areas of engineering, the authors have found significant benefits in using Microsoft Excel spreadsheets with students to develop design, analysis and simulation results for a number of different application areas. Students start from the theory (e.g. basic equations describing a particular concept), and create increasingly sophisticated spreadsheets which can then be used to investigate different aspects of the particular application area. The intuitive interface and scalability of the spreadsheets enables reasonably large-scale simulations to be developed without the students having to generate many complicated formulae or programs, and graphical results are also readily available. This approach enhances students' understanding and appreciation of the theoretical concepts, as well as opening their eyes to the potential of simple spreadsheet usage. This paper describes the development of Excel-based simulations, using various examples in the areas of electronic engineering, robotics, basic physics and mathematics. Various examples are described in detail along with descriptions of the main steps involved in creating the spreadsheets. It is envisaged that this work will encourage engineering educators to develop similar simulations to foster in-depth understanding on the students' part, alongside use of specialist software for larger-scale simulations.

Index Terms —Computer-based teaching, Microsoft Excel, Spreadsheets, Transferable skills

INTRODUCTION

For many areas of education, students benefit greatly from practical experience, which helps them develop a deeper and more complete understanding and appreciation of the details of the particular topic. In areas such as engineering and physics, this experience can be in the form of laboratory-based work, and/or computer-based simulations. This paper focuses on the use of computer-based simulations for various areas of engineering and physics. Whilst special-purpose software or high-level development tools may be available for such areas, the authors, as tutors of various modules in engineering and physics, have found that students can come to regard these as a 'magic box', enabling them to readily and rapidly achieve simulation results for a particular application, without the need for them to actually understand, or even consider, the details of the theoretical aspects. The alternative approach presented in this paper is to encourage the students to use Microsoft Excel to develop simulations of some particular topics encountered in undergraduate engineering and physics courses. The authors regularly use this approach in undergraduate modules at various levels, covering a wide range of topics incuding analogue electronics, digital electronics, equations of motion, artificial neural networks, robotics, and signal processing [1, 2]. Initially, students are shown spreadsheet simulations for a particular example in one of these areas. These are directly and clearly related to the equations describing the particular application, and therefore the students are encouraged to consider the equations more deeply and, for example, to think about how the different variables involved can affect the outcome of the simulation. Students are then guided through the development of their own simulation spreadsheets for related simulations, and/or encouraged to develop a particular given simulation further (for example to create a simulation of a 3-section robot arm from a 2-section robot arm simulation provided by the tutor). As well as developing the depth of their understanding of the technical concepts, this approach also raises the students' awareness of the potential for simple spreadsheet simulations, and often motivates them to experiment further.

This paper first describes some of the ideas of spreadsheet development, including some of the particularly useful aspects such as the use of macros and on-screen buttons to initiate certain functions. Subsequent sections of the paper go on to present a few example application areas drawn from engineering and physics modules taught by the authors. The ICEE2010 conference presentation will cover spreadsheet creation and development, as well as presenting some animated examples of spreadsheet simulations. Interested readers are encouraged to contact the authors directly for further information and discussion.

SPREADSHEET DEVELOPMENT

In many of the physics/maths/engineering simulations developed by the authors, the spreadsheet software may easily be used to create a table of simulation results. This is facilitated by the cut-and-paste, or select-and-drag, facilities in Excel, whereby row and column indices are automatically updated. These automatic updates may be avoided if necessary, by using a '$' prefix for the row and/or column reference as appropriate. It is a very straightforward matter to create a large table of results, which may then be plotted to produce a graph (referred to as a 'chart' in Excel terminology). Values from a particular row of the table may be referenced using the VLOOKUP function, and this data may then used to plot another graph using the same axes of the main graph [1]. Simple macros can then be created such that the rows of the table are accessed one at a time in sequence (or in some other order if required), and running these macros then produces an animated display. A very useful feature of Microsoft Excel [3] is that the macros may be recorded rather than written, thereby allowing students to create their own functions without the need for any programming knowledge. The macros may be linked to keyboard shortcuts and/or on-screen buttons, allowing easy access to the associated functions, and an effective user interface to the simulation. These aspects will be demonstrated in the ICEE2010 conference presentation using Microsoft Excel 2007.

EXAMPLES OF SPREADSHEET SIMULATIONS

This section includes some details of a few of the areas in which the authors have found the use of spreadsheets to be effective. For each example, a summary of the relevant theoretical and/or design aspects is provided, followed by a brief summary of spreadsheet development for the particular application.

Basic Physics: Equations of Motion

Simple equations of motions readily lend themselves to simulation in Excel, enabling students to better appreciate their significance and quickly develop plots showing trajectories, and/or particles moving along trajectories. The relevant equations of motion which may be used for particles under constant acceleration are often referred to as SUVAT equations, and may be visualised in Excel in various ways, two of which are discussed below:

- **Particle trajectory in two dimensions**: In this simulation a particle is launched with a particular initial velocity and elevation angle, and this may be resolved into initial velocities u_x and u_y in the horizontal and vertical directions respectively. Different accelerations may apply in these directions: for the simulation shown in Figure 1, the
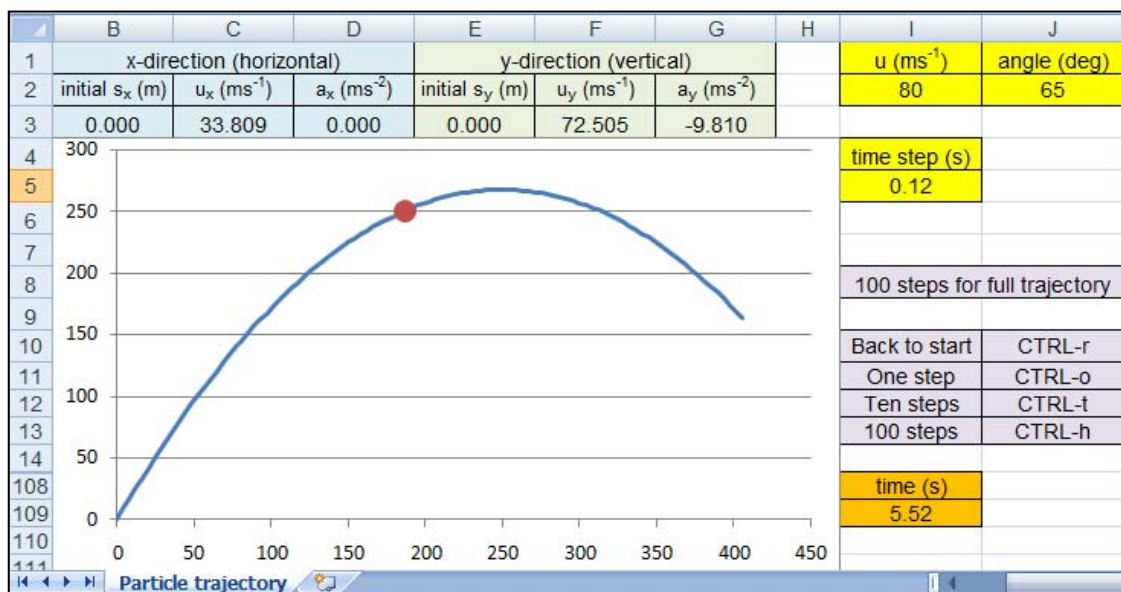

Figure 1 Spreadsheet simulation of particle trajectory including animated particle

horizontal acceleration is zero, whilst the vertical acceleration is 9.81 ms^{-2} downwards, i.e. acceleration due to gravity. It is a simple matter to produce a table of values in Excel, which may be used to plot a line showing the trajectory of the particle. A second plot using the same axes may be used to plot the position of the particle, and this may be updated over time using macros.

- **Bouncing ball**: The particle trajectory simulation may be developed further into a bouncing ball simulation, by including formulae to detect when the particle hits (or passes) some limit (for example a limit of $s_y = 0$ could denote the floor), and then to reverse the direction of the particle, i.e. the ball bounces. The particle's energy after bouncing is a user-selected fraction of its energy (i.e. kinetic energy, since potential energy would be zero at that point) before collision with the floor. This may be used to determine the initial upward velocity of the ball immediately after the bounce.

Basic Physics / Engineering: Addition of Sinusoidal Signals

There are many aspects of physics, engineering, and mathematics where it is useful to represent a system or signal using simple sinusoidal components. Examples include representation of signals using Fourier series and Fourier transforms (spectral analysis), simple harmonic motion (e.g. an undamped oscillatory system), analysis of audio signals etc. Such analysis is easily and readily simulated using Microsoft Excel. Over a number of years, the authors have found this approach particularly useful to support students' understanding and appreciation of spectral analysis of signals. Students carry out assignment work in an undergraduate 'Signal Analysis and Processing' module, using Microsoft Excel to develop increasingly sophisticated spreadsheets for Fourier synthesis and analysis of periodic and quasi-periodic signals.

The example shown in Figure 2 is a very simple spreadsheet which students can create for themselves in only a few minutes under the guidance of the tutor. The spreadsheet generates two separate sinusoidal signals with frequencies and amplitudes that can be selected by the user. These are then added and the resulting signal is plotted (shown on the right in Figure 2), allowing the students to easily experiment with concepts such as beat frequencies and modulation.
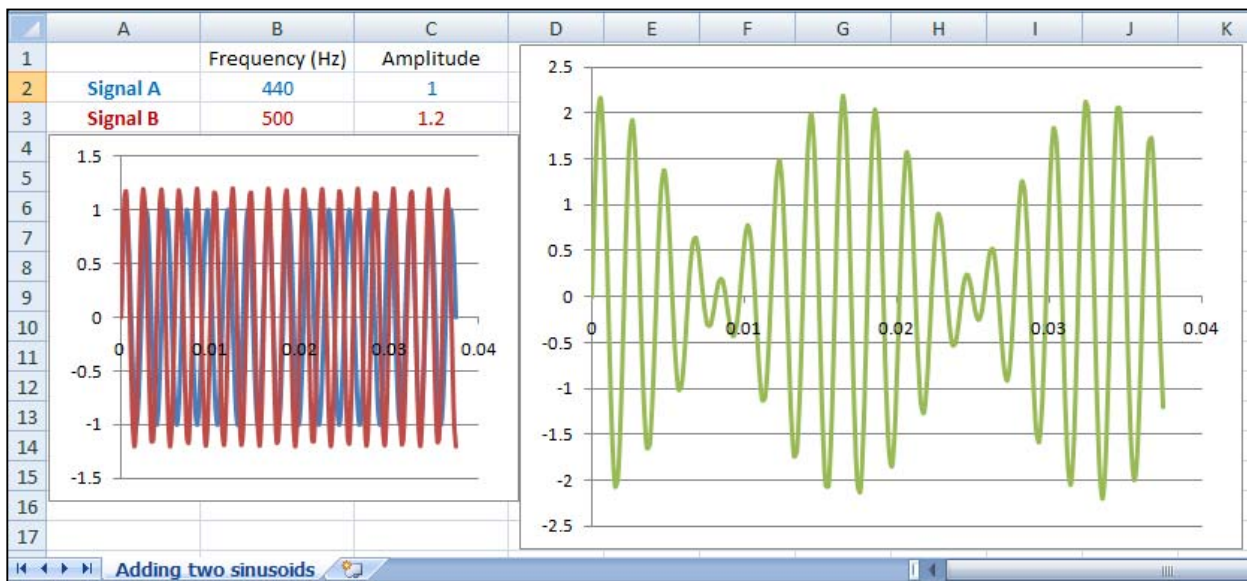

Figure 2 Simple spreadsheet showing the addition of two independent sinusoidal components

Digital Electronics: Combinational and Sequential Logic

This example builds on a previous single-digit Binary Coded Decimal (BCD) counter [1] by extending the simulation to two BCD digits Each is implemented using four D-type flip-flops, with all eight flip-flops connected to a common clock signal i.e. a synchronous counter simulation. A signal from the least significant digit enables the most significant digit to remain unchanged, or to increment or decrement as appropriate, depending on the current value of the least significant digit and the mode of the counter (UP count or DOWN count). For each digit, a seven-segment display is included in the simulation, as well as an indication of the corresponding four flip-flop outputs. Each digit of the seven-segment display is controlled by a combinational logic circuit with four inputs and seven outputs, and conditional formatting is used to light each segment as appropriate.

This simulation includes combinational logic design (i.e. the output logic controlling the seven-segment displays, and the next state logic) and sequential logic design (i.e. use of flip-flops to represent the present state of a circuit, leading

International Conference on Engineering Education ICEE-2010   July 18–22, 2010, Gliwice, Poland.

3

to a particular next state depending on the present state and the mode of the counter). A screenshot of the two-digit BCD counter is shown in Figure 3.
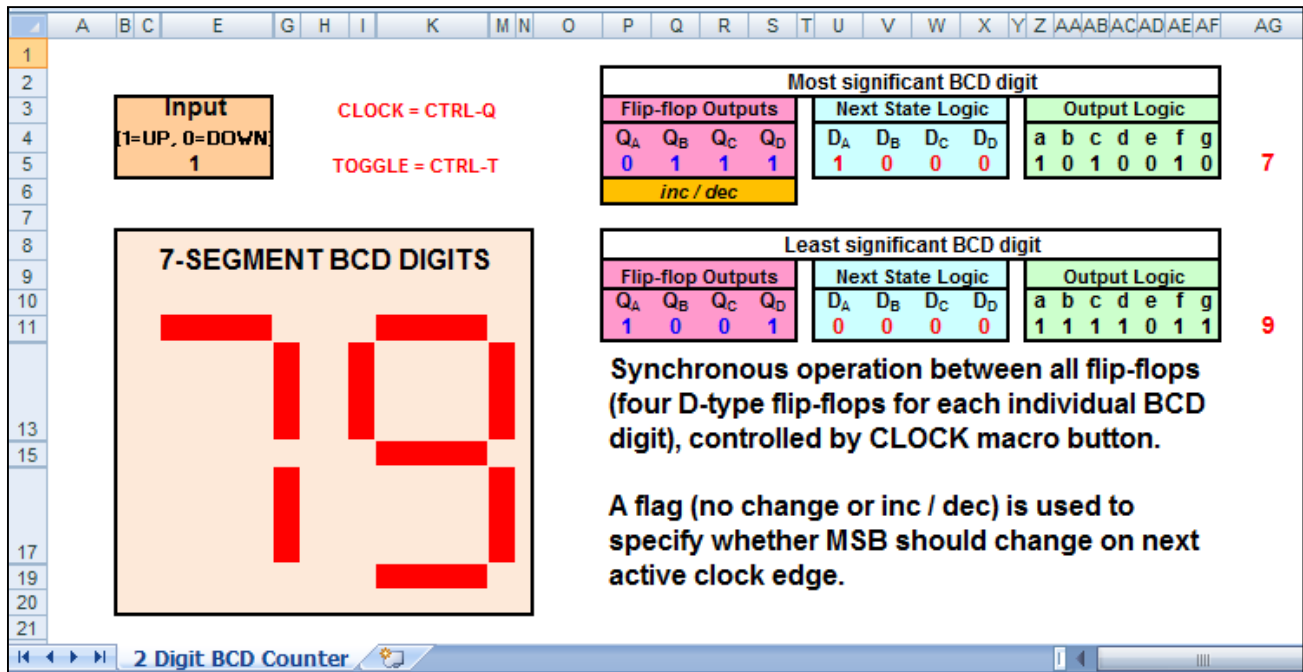

Figure 3 Spreadsheet simulation of a two-digit BCD counter

Students can develop these ideas in other directions, and by incorporating different logical functions into their simulations they can readily develop their own simulations of different and more complex digital systems.

Robotic Systems: Simulation of a Robot Arm

At the University of Central Lancashire, undergraduate course provision in the area of Electronic and Computer Engineering (ECE) is increasingly moving in the direction of robotic systems. Our practical facilities include several robot systems, and also a tele-immersive manufacturing facility which includes various stations for automatic testing, analysis, and mainpulation. One of the key areas covered in our 'Robotic Systems' modules is kinematics. Students are initially introduced to a basic simplified robot arm model, involving two or thre sections. This is presented alongside a physical robot arm (Neuronics' Katana robot arm [4]) which corresponds to the simplified model, but also has further aspects such as rotation about the fixed base, and a rotating gripper.

In this case, students are provided with a basic spreadsheet which plots the three sections of the robot arm in a two-dimensional plane, based on the (x, y) coordinates of the end of each section, with the base of the first section fixed at location $(0, 0)$. Using parameters in particular cells of the spreadsheet, students can specify the length L_i of the ith section of the arm, and its angle θ_i relative to an appropriate reference. Simple trigonometry may be used to determine equations for the coordinates (x_i, y_i) of the end of each of the three sections, and thereby the tip of the arm (i.e. the end of the final section):

$$x_1 = L_1 cos\theta_1 \qquad\qquad x_2 = x_1 + L_2 cos(\theta_1 + \theta_2) \qquad\qquad x_3 = x_2 + L_3 cos(\theta_1 + \theta_2 + \theta_3)$$
$$y_1 = L_1 sin\theta_1 \qquad\qquad y_2 = y_1 + L_2 sin(\theta_1 + \theta_2) \qquad\qquad y_3 = y_2 + L_3 sin(\theta_1 + \theta_2 + \theta_3)$$

The simulated arm moves over 100 incremental steps, from a specified start position to a specified end position. Macros are included in the spreadsheet such that the position of the arm can be reset to the start position, or a single increment can be applied, or 100 increments can be applied. The spreadsheet includes the equations for the first two sections of the arm, but it is left to the students to independently consider and input the appropriate equations for the third section. Parameters defining the start position and end position for each of the 100 increments can also be entered by the students. An example spreadsheet showing an intermediate position of the 2-section arm is shown in Figure 4. In this example, the trajectory of the robot arm is simulated by adding the same increment for each iteration, although this could be done differently by the students to simulate different trajectories from start point to end point.
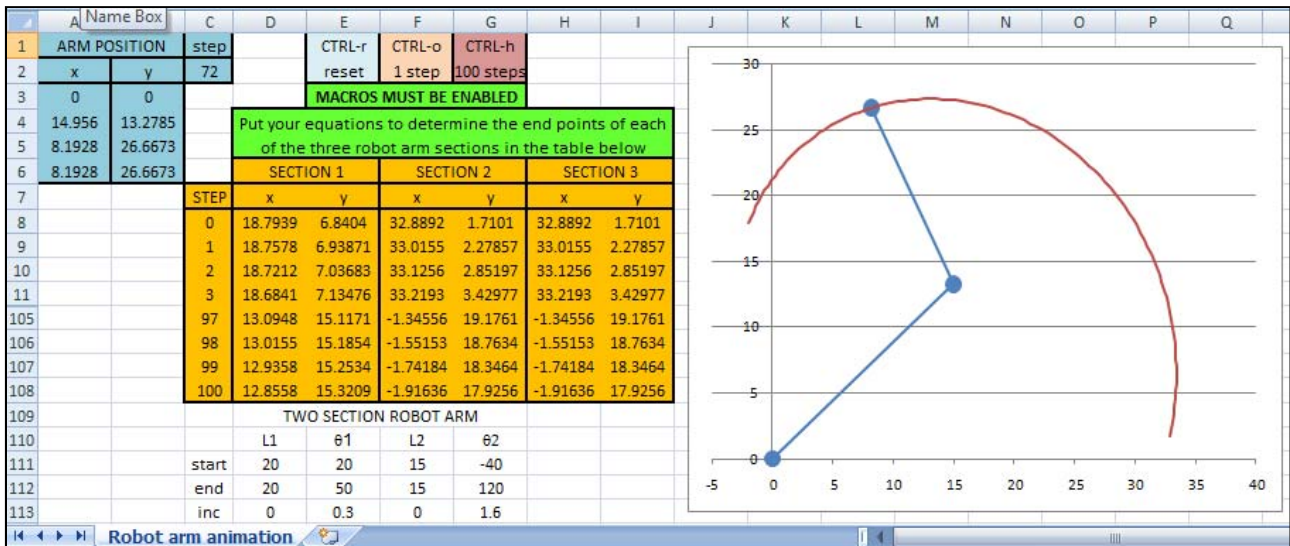
Figure 4 Spreadsheet simulation of 2-section robot arm in two dimensions

This simulation introduces students to the subject of kinematics, and helps them to develop an understanding of forward kinematics, in which the position of each section of the robot arm is determined using the relevant lengths and angles, i.e. it is a many-to-one mapping in which more than one combination of the variable parameters ('many') can lead to the same position of the tip of the robot arm ('one'). In practical situations, a particular position, and perhaps orientation, of the tip is required, and this corresponds to the much more complex problem of inverse kinematics, a one-to-many mapping where a single requirement may have more than one solution, and appropriate solutions are not easily determined. Further spreadsheet simulations have also been developed which use neural network models to provide an iterative solution to determining appropriate values for the variable parameters such that the tip of the robot arm is at the required position.

Digital Signal Processing: Chirp *z*-transform

There are various areas within analogue and digital signal processing where the authors have found spreadsheets to be very effective in helping students to visualise and understand particular concepts. These include Fourier series for signal synthesis and analysis, time-domain analysis (e.g. impulse response analysis) of digital filters including digital oscillators, and simulations of pole/zero analysis in the *s*-plane [1] and the *z*-plane to illustrate complex-plane analytical methods.

The particular example described in this section enables the students to experiment with the chirp *z*-transform (CZT). This is a system analysis technique whereby a *z*-plane contour may be defined as a set of points z_k. The first point of the contour, z_0, is specified by its magnitude A_0 and phase θ_0, and subsequent points are related to the preceding point by parameters W_0 and ϕ_0. The z-plane contour is defined as:

$$z_k = AW^{-k} \qquad\qquad \text{for } k = 0, 1, 2, \cdots, M\text{ - }1$$

where $\quad A = A_0 e^{j\theta_0}$ and $\qquad\qquad W = W_0 e^{-j\phi_0}$

With appropriate parameters, the CZT contour can be made to be the *z*-plane unit circle, in which case CZT analysis becomes the same as signal/system analysis using the discrete Fourier transform (DFT). However the CZT is much more flexible than the DFT, as the contour can be adjusted so as to pass close to the system's poles or zeros, and thereby identify them more accurately.

The particular spreadsheet simulation used by the authors in the undergraduate module 'Digital Signal Processing' allows the students to specify three complex conjugate pairs of poles. These are plotted on a *z*-plane diagram, along with the CZT contour, also specified by the students using parameters A_0, θ_0, W_0 and ϕ_0. This *z*-plane analysis contour can be adjusted by the user, and by trial and error (i.e. experimentation with the CZT contour parameters), the user can 'explore' the z-plane and thereby estimate the characteristics of an unknown system. For the particular example shown in Figure 5, the number of points in the CZT contour is set at $M = 128$, although this could be changed by the students. The parameters of the system and analysis contour are specified in the cells towards the top of the screen, with the *z*-plane plot on the bottom left, and the magnitude of the *z*-transform along the CZT contour at the bottom right.

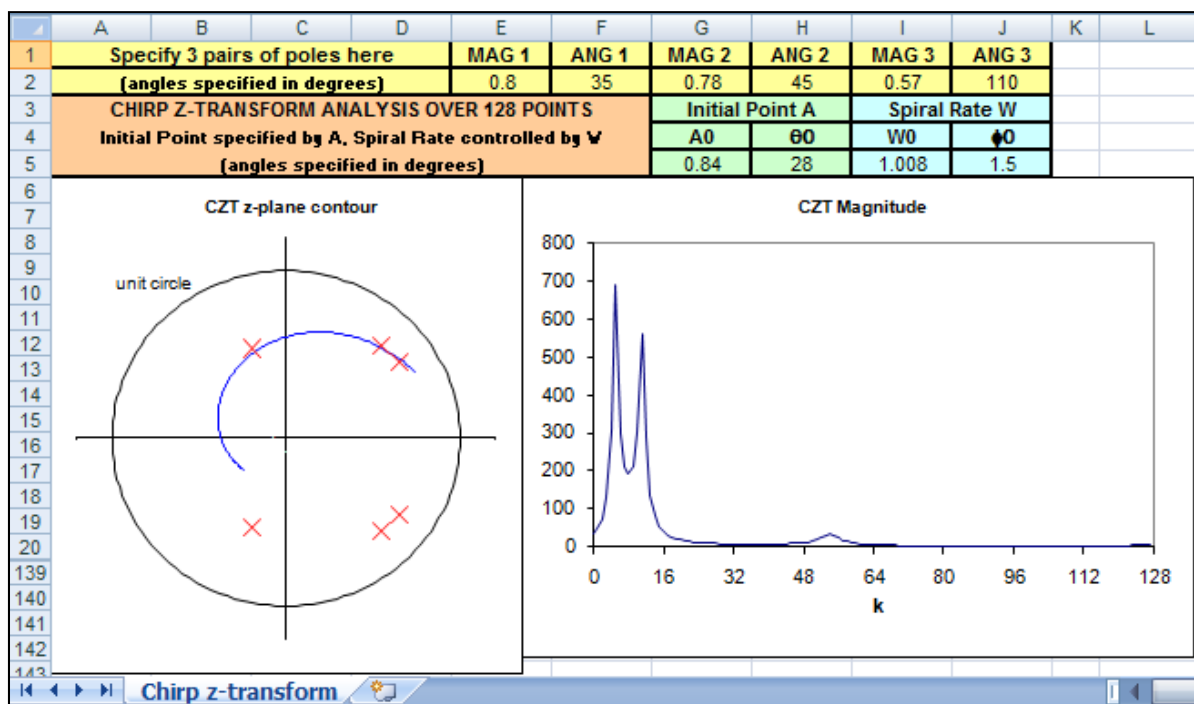|	A	B	C	D	E	F	G	H	I	J	K	L
1	Specify 3 pairs of poles here				MAG 1	ANG 1	MAG 2	ANG 2	MAG 3	ANG 3		
2	(angles specified in degrees)				0.8	35	0.78	45	0.57	110		
3	CHIRP Z-TRANSFORM ANALYSIS OVER 128 POINTS						Initial Point A		Spiral Rate W			
4	Initial Point specified by A, Spiral Rate controlled by W						A0	θ0	W0	φ0		
5	(angles specified in degrees)						0.84	28	1.008	1.5		

Figure 5 Spreadsheet simulation of chirp z-transform analysis for a digital filter with six poles

It can be seen that when the CZT contour passes close to a *z*-plane pole of the system, a significant peak appears in the CZT magnitude plot. Students are encouraged to experiment with this simulation spreadsheet to gain a better understanding of *z*-plane analysis methods in general, and the CZT in particular.

CONCLUDING REMARKS

In this paper, the authors have described the use of Microsoft Excel to aid students in developing a more complete and thorough understanding of concepts in engineering and physics. Indicative examples have been included to illustrate some of the useful features in Excel which facilitiate the development of the simulations, including easy generation of large tables of simulation data, the VLOOKUP function for accessing data row by row, and recordable macro functions to enable animated simulations. These functions will be explained and demonstrated in the ICEE2010 conference presentation.

The authors will continue these developments in the future and extend the use of spreadsheet simulations into other areas within engineering and physics. It is intended that these simulations will continue to complement, rather than be a substitute for, practical experimentation and more complex simulations involving specialist software tools.

ACKNOWLEDGEMENT

REFERENCES

[1] Varley, M.R., Peak, M.S., Picton, P.D. and Heys, J.D., "The Use of General Purpose Software as a Teaching Tool in Electronic Engineering", *Proceedings of The International Conference on Innovation, Good Practice and Research in Engineering Education 2006 (EE2006)*, Liverpool, England, 2006, pp24-29.
Also available online at http://www.ee2006.info/docs/02.pdf (article) and http://www.ee2006.info/presentations.html#esposters (slides), last accessed 26 June 2010.

[2] Peak, M.S., Varley, M.R., and Picton, P.D., "Spreadsheet Software as a Teaching Tool in Technology", *Proceedings of The International Symposium on Modern Applied Technology and Management*, Beijing, China, 2005, pp341-346.

[3] Liengme, B.V., *A Guide to Microsoft Excel 2007 for Scientists and Engineers,* Academic Press, 2009.

[4] Neuronics AG Research, "*Intelligent and personal robotics: Katana – automation made easy*", available online at http://www.neuronics.ch/cms_en/web/index.php?id=244&s=katana, last accessed 26June 2010.