

Spreadsheet Software as a Teaching Tool for Concepts in Electronic Engineering

M.R. VARLEY, M.S. PEAK, J.D. HEYS, G. COLLINS, A.J. KONSTANTARAS

Department of Technology
University of Central Lancashire
Preston PR1 2HE
UNITED KINGDOM

<http://www.uclan.ac.uk/technology>

F. VALLIANATOS

Technological Educational Institute of Crete
3 Romanou Str., Chania GR-73133, Crete
GREECE

<http://www.chania.teicrete.gr>

P.D. PICTON

School of Applied Sciences
University College Northampton
St George's Avenue, Northampton NN2 6JD
UNITED KINGDOM

<http://www.northampton.ac.uk>

Abstract: - It is well-known that practical experience aids and reinforces students' understanding of concepts within engineering. Use of high-level simulation and design tools has become commonplace in many areas of electronic engineering. Whilst this may enable designs and simulation results to be readily obtained, it is often at the cost of in-depth understanding on the part of the student. In this paper, the authors describe the use of spreadsheets to enable students to develop simulation results directly from basic theory for a number of different application areas within electronic engineering. This approach enhances students' understanding and appreciation of the theoretical concepts, and complements the use of high-level design and analysis tools.

Key-Words: - Electronic Engineering, Analysis, Simulation, Computers in Engineering Education, Spreadsheet

1 Introduction

In many of the technical areas encountered in electronic engineering, teaching and learning are greatly enhanced by practical demonstrations by the teacher, and hands-on practical experience and experimentation on the part of the learner. Various software packages, such as PROTEUS or MATLAB, are available which can readily produce quick results for a wide range of technical areas covered within the electronic engineering curriculum. Whilst use of such packages is clearly valuable in engineering education, it can in some cases be counter-productive since, in the experience of the authors, students may sometimes use the availability of such high-level software as a replacement for their own in-depth understanding. In this paper the authors present, and discuss their experience with the use of ordinary, widely

available, spreadsheet software in the teaching of electronic engineering. A range of technical areas is presented in which students are encouraged to develop their own simulation results directly from the appropriate theoretical models presented in the lectures and course notes. The authors have developed this approach across several modules at all levels within an undergraduate Bachelor of Engineering honours (BEng (Hons)) degree course. In each of these modules, students are guided through the construction of a spreadsheet incorporating the basic mathematical formulae representing a particular application. Example spreadsheets are made available to the students, and discussed within the lectures so as to clarify the implementation of aspects of the theory within the spreadsheet. The simple graph plotting facilities available within the spreadsheet then enable ready

access to a graphical description, which can easily be made interactive and animated using on-screen buttons and macros. As well as encouraging the students to develop their understanding by producing simulation results from the theory, the use of readily available general purpose software makes this approach more readily accessible to a significantly wider audience than is currently the case when using specialist software.

2 Development of Simulations

In this section, a brief explanation is given of how a spreadsheet [1] is created and used to produce dynamic graphical results. The example presented in this section simulates a simple AC theory application, the technical aspects of which are discussed in Section 3.1. The first few rows of the spreadsheet are used for parameters which may be varied by the user, such as (in this example)

component values, frequency range limits and a sample frequency value. Different columns of the spreadsheet are used to evaluate the relevant formulae for many different analysis parameters (in this case this corresponds to a 1001-row table evaluating the component voltages for different frequencies over the specified range, which are then plotted). Figure 1 shows a screenshot of the spreadsheet with the plots reduced in size so as to make the table of calculations, normally hidden behind the plots, visible. Macros are used to implement repeated processes enabling dynamic display: in this case a macro selects successive rows from the 1001-row table and plots the voltages as a moving phasor diagram, and as a moving marker on the frequency response plot(s). These macros may be associated with text boxes, thereby creating clickable on-screen buttons which may be used to control the operation of the spreadsheet.

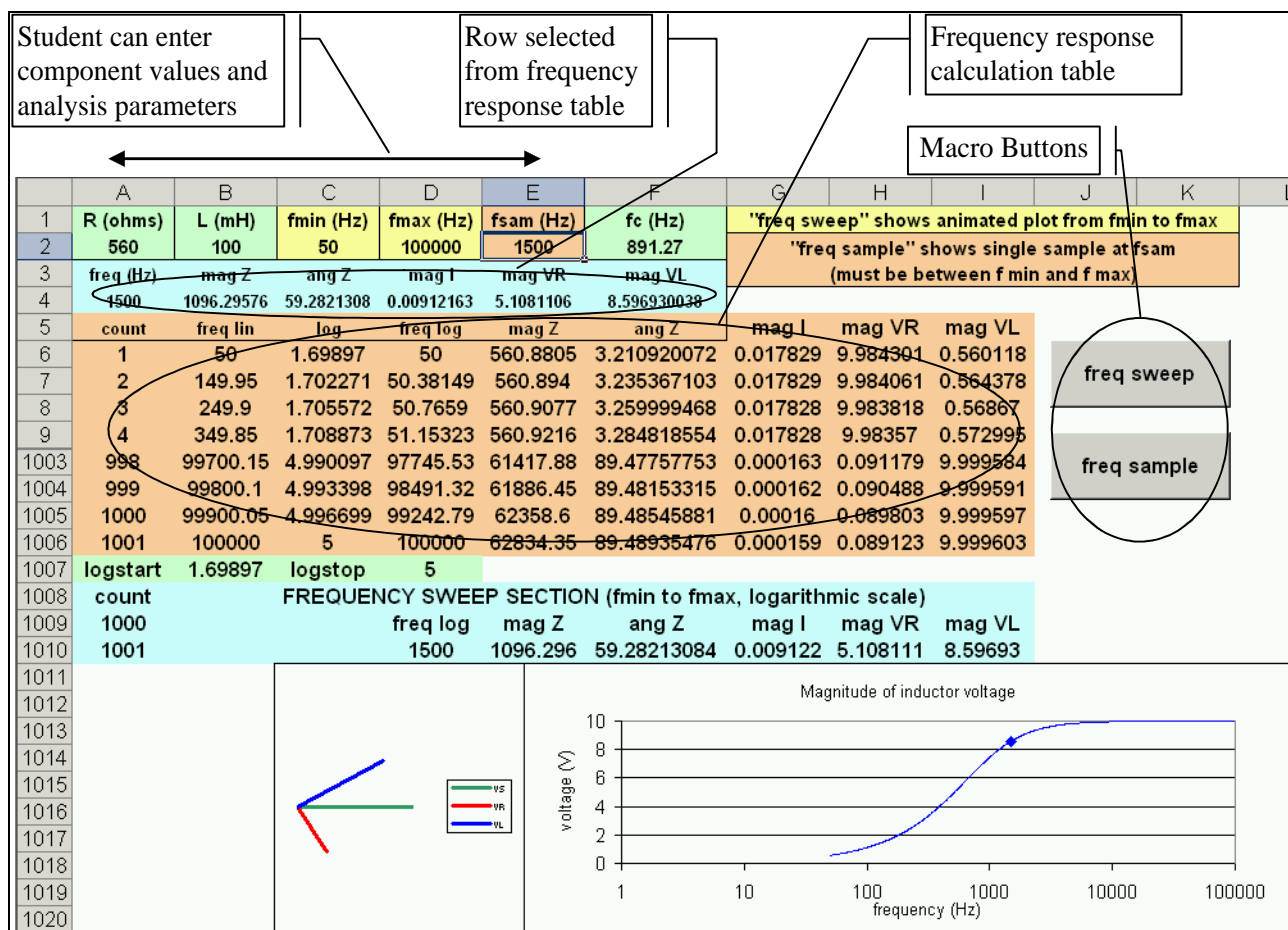


Figure 1 Example screenshot showing different areas of the spreadsheet

3 Concepts in Electronic Engineering

This section presents four separate modules from the undergraduate degree course, in which the authors have found this spreadsheet-based approach fruitful. Spreadsheets are introduced in the course during the first year, for simulation of basic AC theory within the level 1 module ‘Analogue Electronics’, as described in section 3.1. The students’ knowledge and experience in this area are further developed by using spreadsheets in the level 2 module ‘Signal Analysis and Processing’ to aid the explanation of areas including s-domain system analysis and Fourier series. Section 3.2 presents an example of s-plane analysis using this approach. Another level 2 module benefiting from the use of this approach is ‘Artificial Neural Networks’ (described in section 3.3), in which the training of single neurons and of various neural network models are presented. Further spreadsheet analysis is included in the level 3 module ‘Digital Signal Processing’, where this approach readily and effectively demonstrates issues including digital filters, z-plane analysis, and the chirp z-transform, as described in section 3.4.

3.1 Basic AC Theory

The level 1 module ‘Analogue Electronics’ introduces a number of aspects of AC (alternating current) theory, including resistor-inductor (RL) and

resistor-capacitor (RC) networks, phasor representations, and resonant circuits [2]. The example shown in this sub-section presents the simulation of a series RL circuit with a fixed supply voltage over a range of input frequencies. In this circuit, the voltage across the resistor decreases as the input frequency is increased, whilst the voltage across the inductor increases as the input frequency is increased, due to the increasing inductive reactance. Figure 2 shows a screenshot of a spreadsheet which evaluates and plots the voltage across each of the two components, R and L, assuming a supply voltage of 10 V. A colour-coded phasor diagram illustrates the phasor relationship between the supply voltage V_S , the resistor voltage V_R and the inductor voltage V_L . The user can specify the values for components R and L, the range of frequency analysis f_{\min} and f_{\max} , and also a single value of frequency f_{sam} . The on-screen buttons select a dynamic ‘frequency sweep’ from f_{\min} to f_{\max} , or a single ‘frequency sample’ at f_{sam} . The dynamic sweep shows a moving phasor diagram, and corresponding colour-coded moving markers on the frequency response plots. Having been guided through the elements of the RL simulation spreadsheet, the students are encouraged to develop similar spreadsheets for RC circuits and series resonant (RLC) circuits.

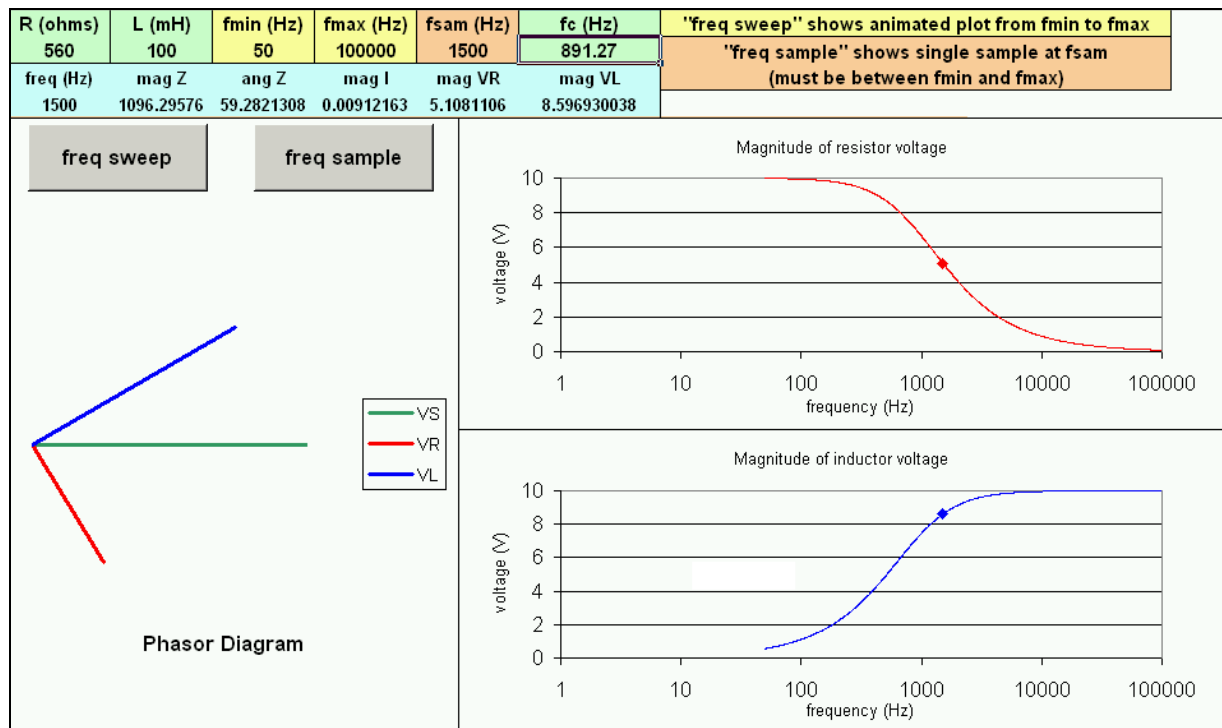


Figure 2 Frequency Response Analysis of RL circuit including Phasor Diagram

3.2 System Analysis using the s-plane

The level 2 module ‘Signal Analysis and Processing’ develops the ideas introduced at level 1, by using spreadsheets to illustrate areas including s-domain analysis and Fourier series representations. The example shown in Figure 3 illustrates a simulation of s-plane analysis, in which the user may specify up to four s-domain poles p_i and four zeros z_j (magnitude and angle) of a transfer function $H(s)$, and a scale factor K . The pole/zero locations are plotted on an s-plane diagram. $H(s)$ is defined as:

$$H(s) = K \times \frac{\prod_{j=1}^4 (s - z_j)}{\prod_{i=1}^4 (s - p_i)} \quad (1)$$

If the user requires fewer than four poles and/or zeros, then some poles and/or zeros may be disabled by specifying an angle greater than 360° , in which case these poles/zeros are ignored in the subsequent analysis. As with the AC theory example in section 3.1, frequency range and frequency sample values (f_{\min} , f_{\max} and f_{sam}) may be specified by the user. Frequency analysis is carried out in the s-domain using the distances between the poles/zeros and the frequency point of interest (on the imaginary axis in the s-plane) for the magnitude response (equation 2), which is calculated and plotted.

$$|H(s)| = K \times \frac{\prod_{j=1}^4 d_{zj}}{\prod_{i=1}^4 d_{pi}} \quad (2)$$

The spreadsheet shows these distances as $dz1$, $dp1$, etc., allowing students to perform calculations manually and check their results against the spreadsheet. Similarly, the angles from poles/zeros to the frequency point of interest (given as $angz1$, $angp1$ etc.) are used to determine and plot the phase response according to equation 3.

$$\angle H(s) = \sum_{j=1}^4 \text{ang}_{zj} - \sum_{i=1}^4 \text{ang}_{pi} \quad (3)$$

Three plots are shown: the s-plane diagram, the magnitude response and the phase response, and on-screen buttons may be used to select a dynamic ‘frequency sweep’ from f_{\min} to f_{\max} , or a single ‘frequency sample’ at f_{sam} . The dynamic sweep shows moving markers on the s-plane and frequency response plots. Again, students are encouraged to experiment with this spreadsheet for investigating topics and examples covered in lectures (e.g. Butterworth low-pass filter design), and to develop similar spreadsheets for higher order systems.

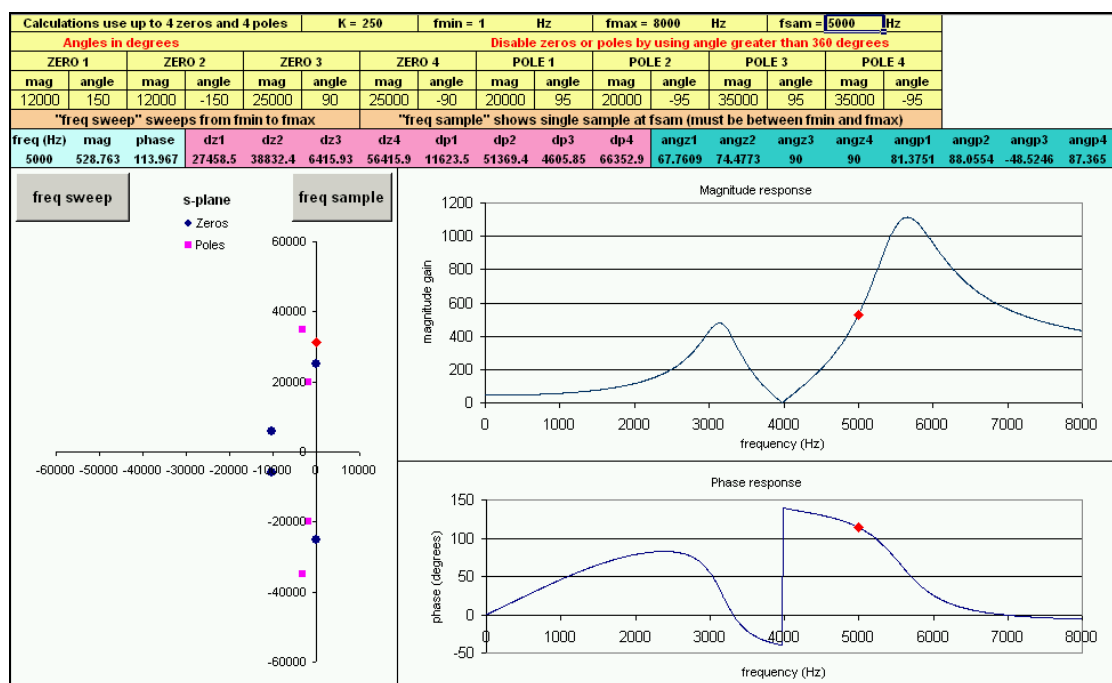


Figure 3 System Analysis using s-plane Poles and Zeros

3.3 Artificial Neural Networks

The authors have developed several spreadsheets to demonstrate a number of concepts [3] for the level 2 module ‘Artificial Neural Networks’, including: training algorithms for McCulloch Pitts (MCP) neurons, error backpropagation for multi-layer networks, the design and analysis of Hopfield (single layer) networks, and the application of Hopfield networks in the implementation of an image compression system. The example presented here shows a spreadsheet demonstrating the Delta Rule training algorithm for MCP neurons.

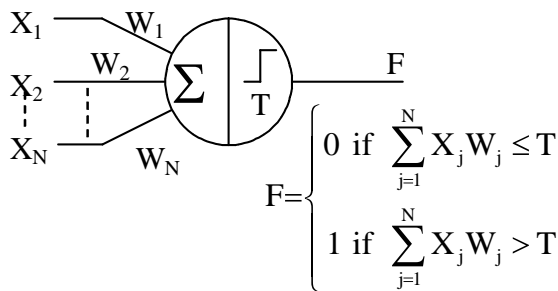


Figure 4 A McCulloch Pitts neuron

The Delta Rule training algorithm is implemented by sequential application of the training patterns and updating of the weights and threshold as defined by equations 4 and 5:

$$W_{i(\text{new})} = W_{i(\text{old})} + \alpha \delta X_i \quad (4)$$

$$T_{(\text{new})} = T_{(\text{old})} + \alpha \delta \quad (5)$$

where α is a user-specified learning rate factor, and δ is the difference between the required output and the actual output, i.e. $\delta = (R - F)$, for the current input pattern. The sequence is restarted after applying an input pattern for which $\delta \neq 0$, and training is complete when $\delta = 0$ for all patterns.

The example in Figure 5 shows the completed training of a two input MCP neuron trained to perform the logical AND function. The training set is fixed. The user may specify initial values for the neuron’s weights and threshold and the learning rate factor α . These values are then copied to the working part of the spreadsheet using the RESET macro. The user may then single-step the algorithm (STEP) or run the algorithm to the next point where the weights and threshold are updated (RUN TO CHANGE). The RUN TO END macro resets the system and then runs the algorithm until training is complete. This spreadsheet-based approach has the clear advantage, over a procedural programming language such as C, of allowing the students to see the training process iteration by iteration and the corresponding changes in the decision line position as training progresses, and enabling them to readily assess the effects of different starting conditions and learning rates.

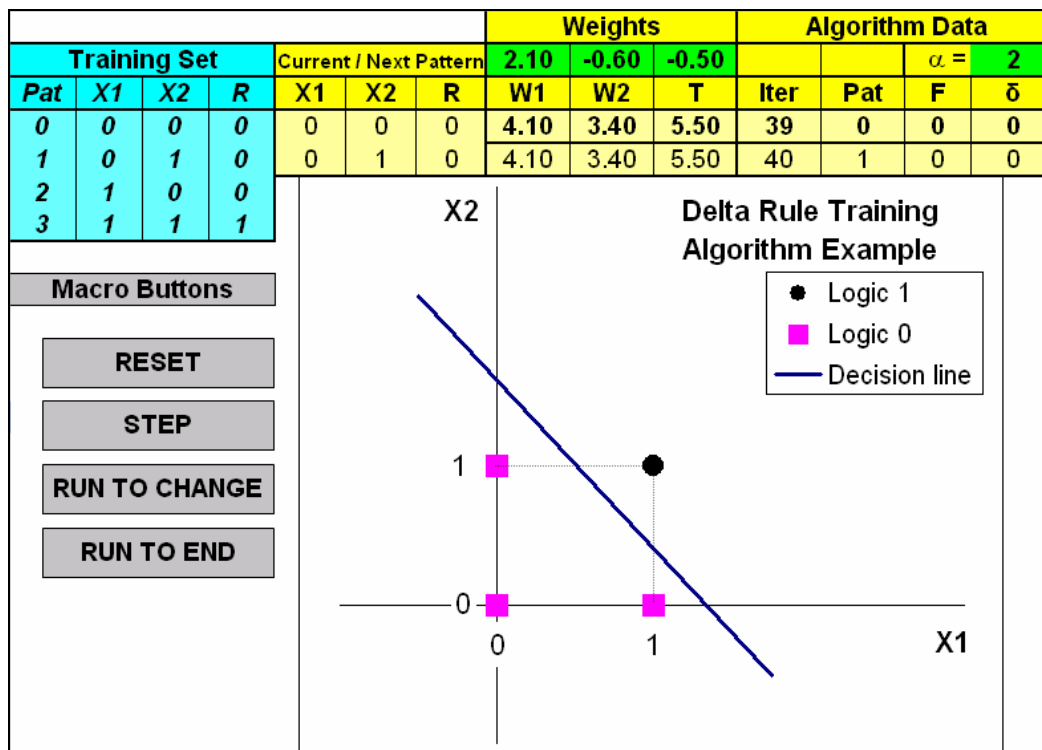


Figure 5 MCP Neuron Training using the Delta Rule Training Algorithm

3.4 Digital Signal Processing

The level 3 module ‘Digital Signal Processing’ extends and develops the signal analysis concepts described in section 3.2 to discrete-time signals, and includes concepts such as z-plane analysis, digital oscillator design, and the chirp-z-transform (CZT). Figure 6 illustrates a CZT [4] simulation, in which the user may specify up to three complex conjugate pairs of poles of a z-domain transfer function $H(z)$:

$$H(z) = \frac{1}{\prod_{i=1}^3 (z - p_i)(z - p_i^*)} \quad (6)$$

where * denotes complex conjugate.

The spreadsheet analyses the magnitude of $H(z)$ over a z-plane contour defined by the parameters

$A = A_0 \angle \theta_0$ (defining the start point of the contour) and $W = W_0 \angle -\phi_0$ (defining the spiral rate of the contour). The z-plane contour incorporates 128 points z_k defined by:

$$z_k = AW^{-k} \quad k = 0, 1, 2, \dots, 127 \quad (7)$$

Figure 6 shows an example result from this spreadsheet simulation, with the z-plane plot on the left indicating the positions of the six poles and the user-specified CZT contour. It can be seen that the contour passes quite close to the first two poles, resulting in significant separate peaks in the magnitude plot, shown on the right of Fig. 6. Students can experiment with different transfer functions and CZT contours, enabling a deeper understanding and better appreciation of the CZT.

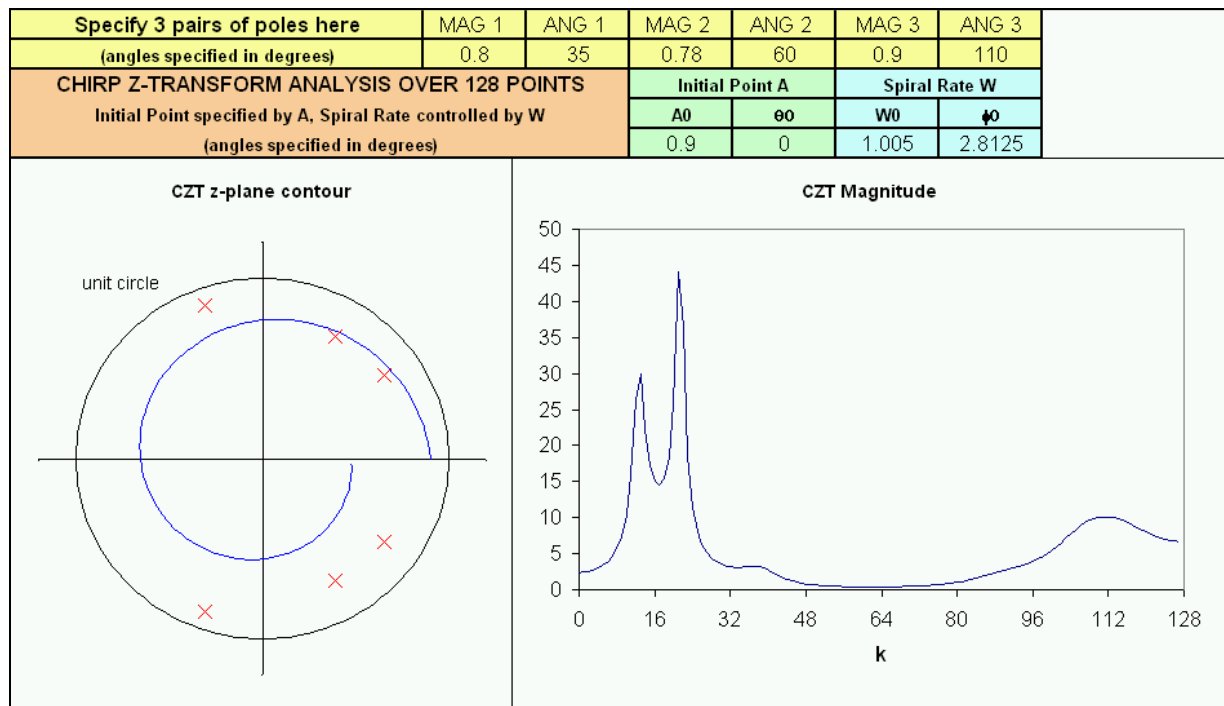


Figure 6 Discrete-Time System Analysis using Chirp z-Transform Algorithm

4 Concluding Remarks

This paper has presented, with illustrative screenshots, a discussion of the use of spreadsheets for a number of different application areas within electronic engineering. It has been found that this approach is helpful and constructive across all levels of an undergraduate honours degree course, and that a deeper and more complete understanding of the theory is achieved by the students, compared to the use of high-level tools alone. The use of high-level tools is still maintained, and complements this spreadsheet-based approach.

References:

- [1] Liengme, B.V., A Guide to Microsoft Excel for Scientists and Engineers, Arnold, 1997
- [2] Hughes, E., Electrical Technology, 7th edition, Longman, 1997
- [3] Picton, P.D., Neural Networks, 2nd edition, Palgrave Macmillan, 2000
- [4] Oppenheim, A.V., Schafer, R.W. and Buck, J.R., Discrete-Time Signal Processing, 2nd edition, Prentice Hall 1999