POLES AND ZEROS, AND MATLAB®, OH MY!

Thad B. Welch†, Cameron H.G. Wright‡, and Michael G. Morrow†

†Department of Electrical Engineering
U.S. Naval Academy, MD

‡Department of Electrical Engineering
U.S. Air Force Academy, CO

1. Introduction

Digital signal processing (DSP) using MATLAB® is being taught at the undergraduate level all around the world[1-8]. Even with the tremendous computational capabilities of MATLAB®, the significance of the pole/zero plot remains a mystery to many of our students. The ability to predict the effect of pole/zero location on either the magnitude or phase plot can be significantly enhanced using computer software. Software programs already exist that allow a student to calculate the magnitude and phase response associated with the arbitrary placement or movement of real and complex poles/zeros upon the complex plane. Unfortunately, these programs all have significant limitations.

2. Discussion

Our program adds several features that are either totally unavailable, or only partially available from these other programs. Specifically, we use the familiar MATLAB® environment to add a graphical user interface (GUI), see Fig. 1, that allows for easy interactive pole/zero placement, relocation, and/or deletion. This GUI includes both real and complex conjugate pairs of both poles and zeros. After a satisfactory pole/zero plot is constructed (e.g., the notched filter shown in Fig. 2), clicking on the *Plot mag/phase* button causes the magnitude and phase plots to be calculated and displayed in a separate figure (e.g., Fig. 3). The transfer function

$$H(z) = \frac{B(z)}{A(z)}$$

is also calculated and returned to the MATLAB® workspace using the familiar numerator coefficient variable **B** and the denominator coefficient variable, **A**. Rapid updates are possible and there is no need for a command line interface. Additionally, clicking the *Load/run DSK* button with the mouse downloads the calculated filter coefficients to an attached TMS320C31 DSK and executes this filter. Since the order of a filter designed using this technique is not expected to be large, implementation using only direct form II (DF-II) is provided.

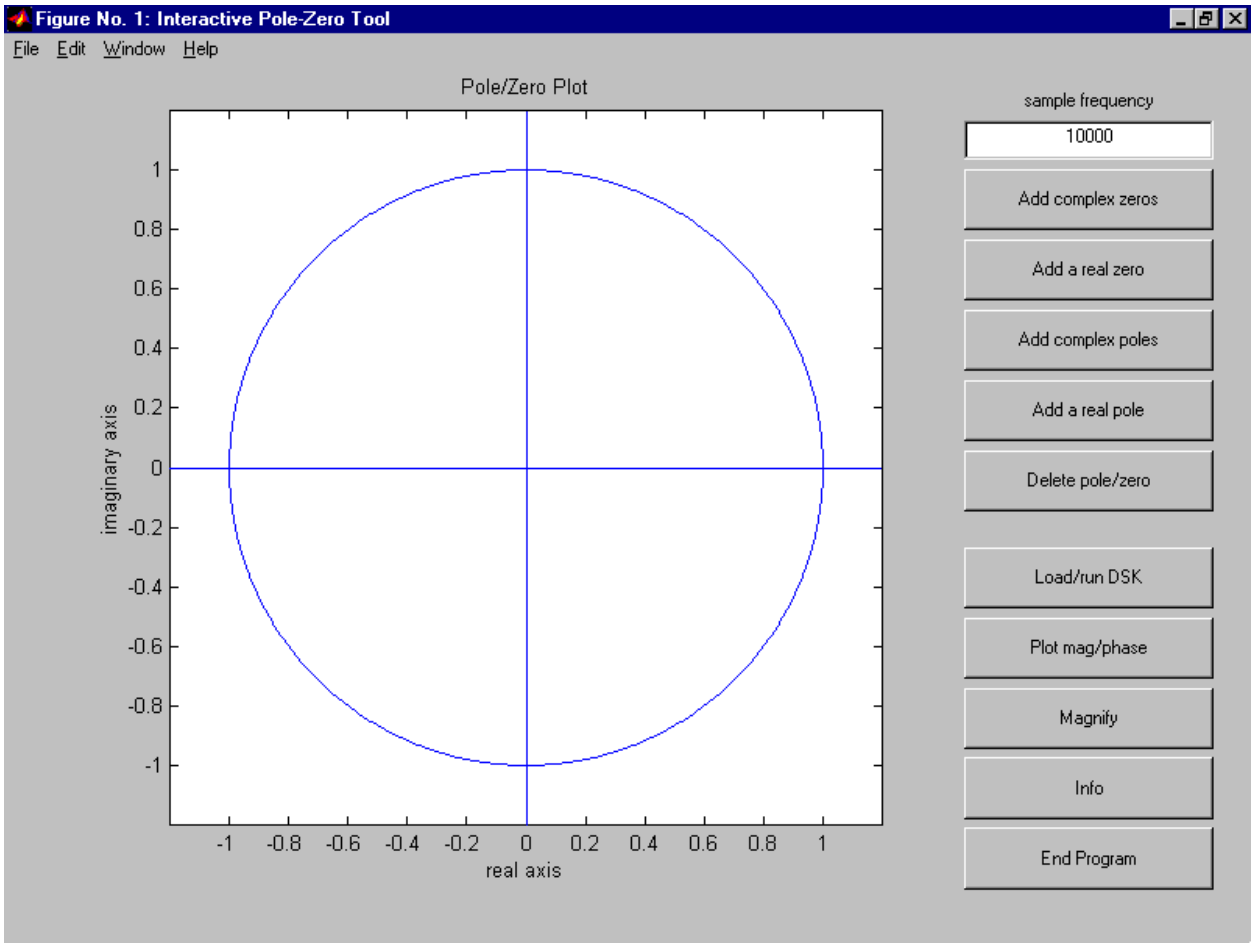

Fig. 1. Interactive Pole-Zero Graphical User Interface.

We often need to remind the student that the movement of one or more poles outside of the unit circle will result in an unstable system (only causal systems are considered). A special warning about the stability of the system is required since MATLAB® does not check for this condition. The displayed information is meaningless, since the frequency response for an unstable system is undefined. An example of this warning message, with a red warning background, is provided in Fig. 4. The fact that such a filter is unstable can easily be verified using the MATLAB® command,

stem(filter(B,A,[1 zeros(1,99)]))

This command calculates and stem plots the first 100 terms of the impulse response of the filter being designed. Theoretically, unstable filters, once excited, have output signals that grow without bound. In a hardware system, noise or any input signal will excite the system. The TMS320C31 DSK, however, will either saturate or oscillate. In both cases, the output in no way corresponds to the output expected from a stable system.

More elaborate control of the DSP hardware and a discussion of the software link between the MATLAB® workspace variables and the DSP hardware are also available[9].

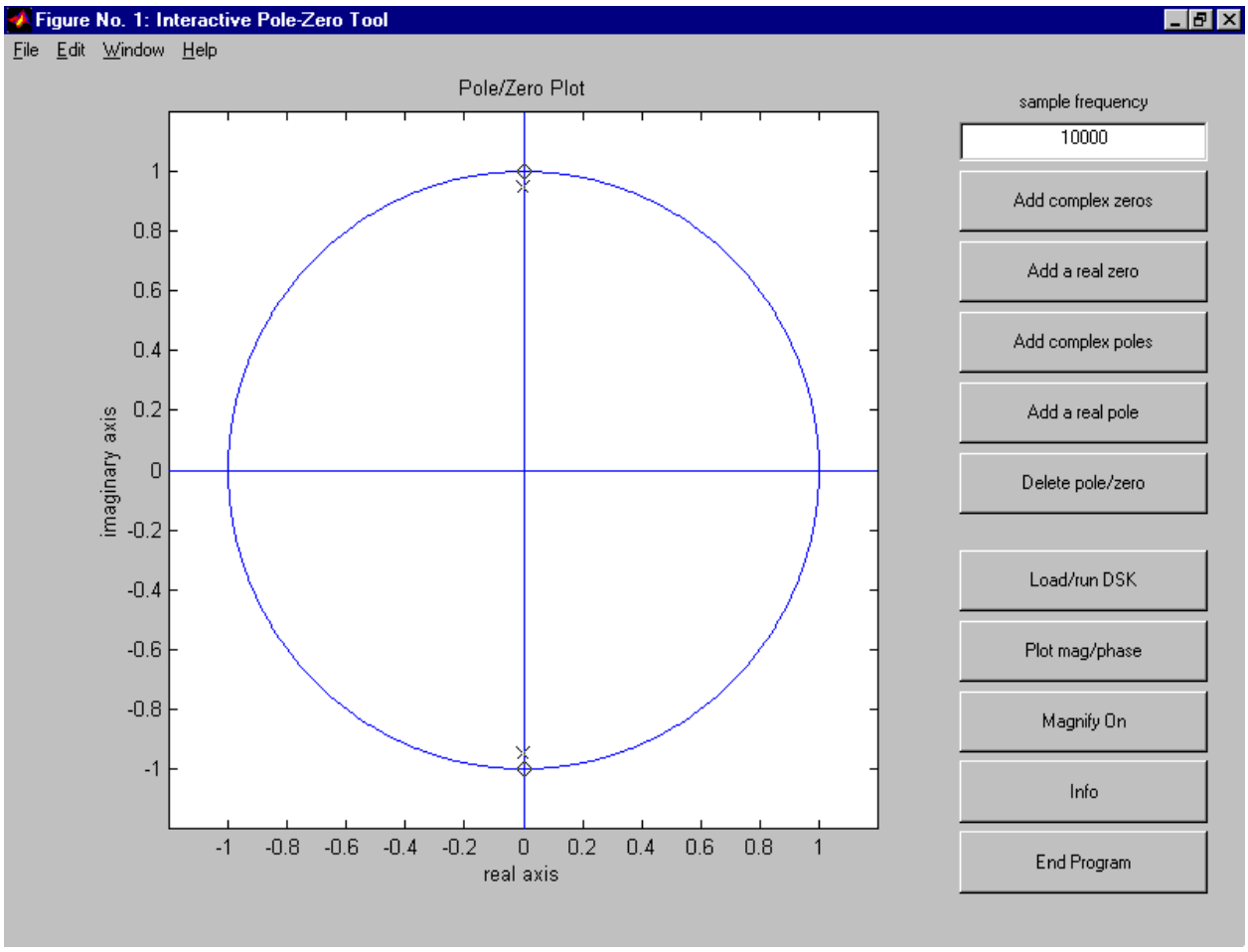

Fig. 2. Interactive pole-zero graphical user interface.

3. A Simple Design Example

A notched filter represents one of the easiest digital filters to design and understand. With this ease of design also comes an opportunity for the student to explore the effects of pole and zero placement on the associated magnitude and phase plots. In Fig. 2, a notched filter was used without derivation. This filter, which was an elementary digital filter design and implementation exercise, will now be more thoroughly discussed.

Since the ultimate goal is to implement the filter in a hardware device, a sample frequency that is supported by the DSP device must be selected. The TMS320C31 DSK has a default sample frequency, f_s of about 10 kHz, and this frequency will therefore be used. With the unit circle now being used as a drawing board for digital filter design, it is imperative that the student understand that 0, $f_s/4$ (2500 Hz), and $f_s/2$ (5000 Hz) map to 1, j, and -1 respectively, on the z-plane. In this design example, the student is tasked with designing a digital filter that will *notch out* 2500 Hz with minimal effect on all the other frequencies.

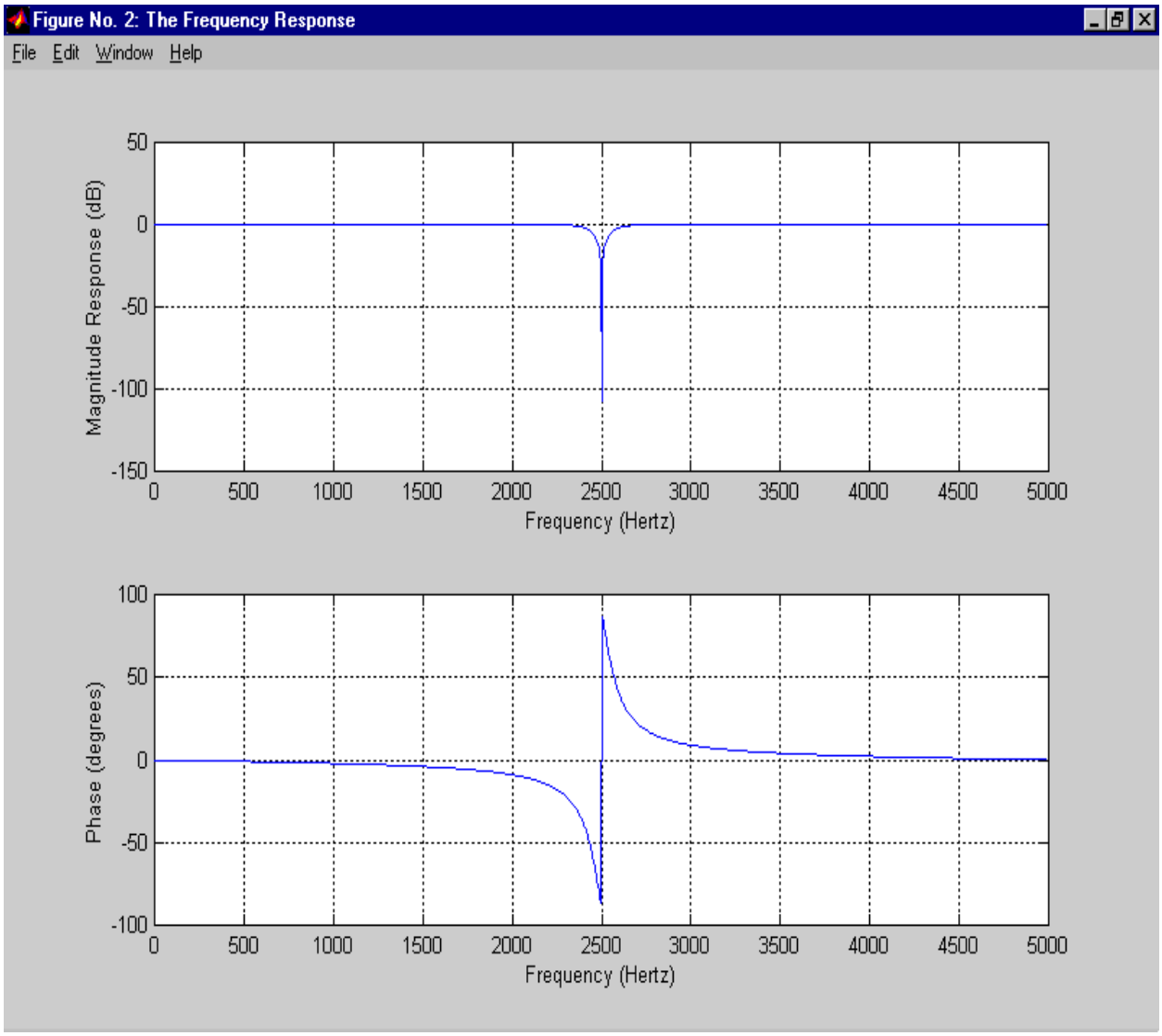

Fig. 3. Magnitude and phase plots associated with the notched filter of Fig. 2.

This design can take place mathematically, e.g.,

$$\frac{(z+j)(z-j)}{(z+j0.95)(z-j0.95)} = \frac{z^2+1}{z^2+0.95^2}.$$

With the transfer function now known, the numerator and denominator polynomial coefficients variables can be created in MATLAB®,

B = [1 0 1];
A = [1 0 0.95^2];

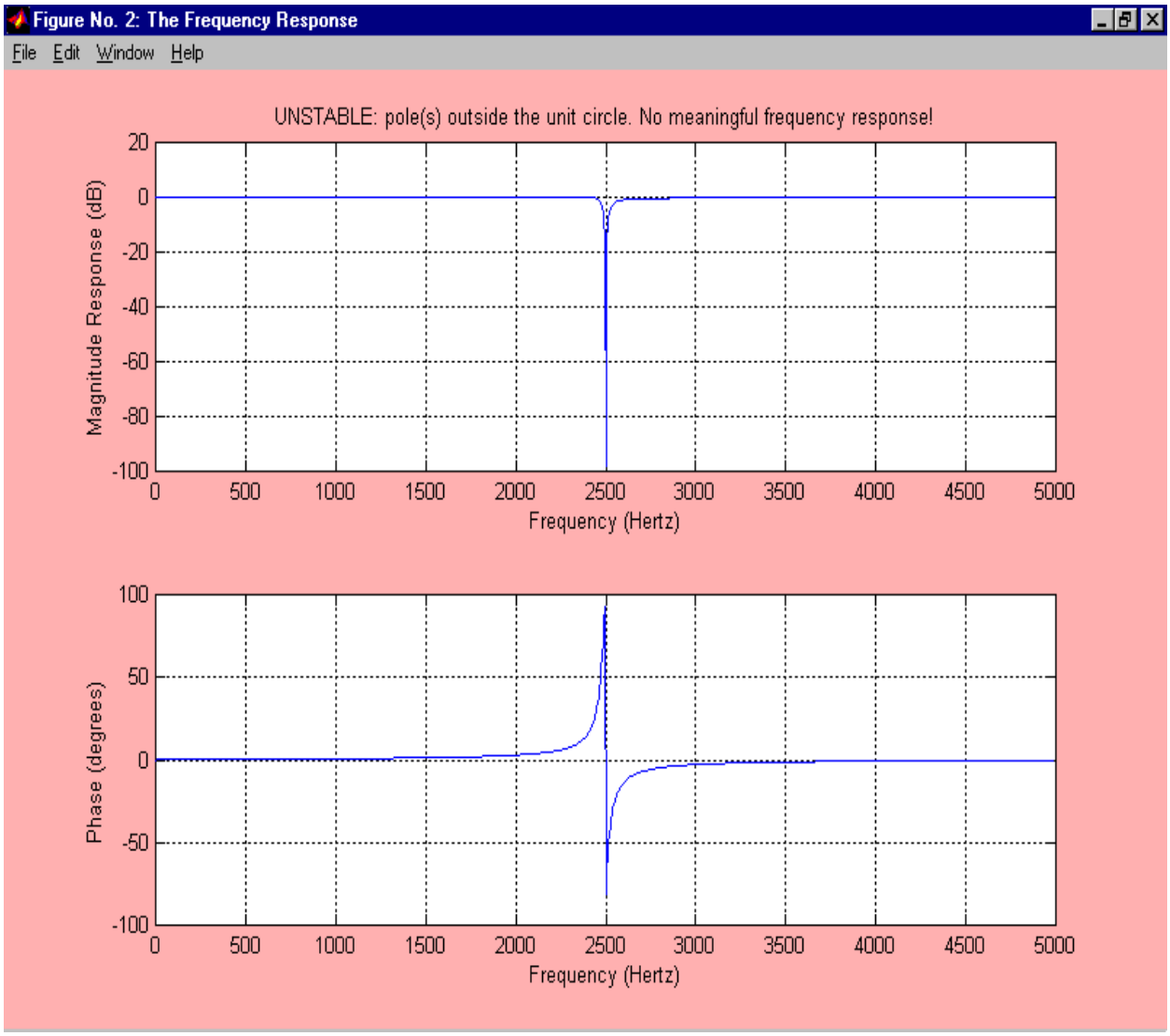

Fig. 4. Magnitude and phase plots associated with an unstable filter.

Filter analysis can now begin with MATLAB® commands such as,

zplane(B, A)
freqz(B, A)

These two commands plot the poles and zeros and the frequency response (magnitude and phase) of the transfer function. Even with the assistance of the computer to expand the polynomials, an iterative design is a lengthy process. The idea of,

- designing a filter on paper,
- algebraically computing the transfer function,
- extracting the numerator and denominator coefficients,
- plotting the location of the poles and zeros,

- plotting the frequency response, and
- iterating the process ...

is a useful exercise to learn the programming language, but adds very little to the student's knowledge of digital filter design. The GUI process removes the need for a command line interface and shortens the process to,

- designing a filter graphically in a familiar MATLAB® figure window,
- apply the design (click a button) to view all of the plots,
- iterating the process ...

With the tedium of the design process now removed, our students have spent additional hours exploring the "what ifs..." of digital filter design, even after the required assignments were completed!

4. Conclusions

With this program, the student now has a free, powerful, easy to learn and use software tool that allows for graphical filter design, hardware implementation, and exploration of the "what happens if ..." we all desire our students to explore! This program has been successfully used for both classroom demonstrations and filter design projects in both a Junior level *signals and systems* class as well as in a Senior level *digital signal processing* class. Student comments support our original assumptions that graphical user interfaces are easy to learn and use and promote intellectual curiosity.

All of this software is freely available for downloading via the www site http://wseweb.ew.usna.edu/ee/LINKS/EE_Links.htm (should the URL be changed, from the Naval Academy home page, select Academics, Academic Divisions and Departments, Electrical Engineering, Links).

References

[1] Kubichek, R. F., "Using MATLAB® in a Speech and Signal Processing Class," *Proceedings of the 1994 ASEE Annual Conference*, pp. 1207–1210, June 1994.

[2] Burrus, C. S., "Teaching Filter Design Using MATLAB®," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 20–30, April 1993.

[3] Jacquot, R. G., Hamann, J. C., Pierre, J. W., and Kubichek, R. F., "Teaching Digital Filter Design Using Symbolic and Numeric Features of MATLAB®," *ASEE Computers in Education Journal*, vol. VII, no. 1, pp. 8–11, January-March 1997.

[4] Porat, B., *A Course in Digital Signal Processing*, John Wiley & Sons, Inc., 1997.

[5] Ingle, V. K., and Proakis, J. G., *Digital Signal Processing Using MATLAB® V.4*, PWS Publishing, 1997.

[6] Mitra, S. K., *Digital Signal Processing: A Computer-Based Approach*, McGraw-Hill, 1998.

[7] Ambardar, A., and Borghesani, C., *Mastering DSP Concepts Using MATLAB®*, Prentice-Hall, 1998.

[8] Burrus, C. S., McClellan, J. H., Oppenheim, A. V., Parks, T. W., Schafer, R. W., and Schuessler, H. W., *Computer-Based Exercises for Signal Processing Using MATLAB®*, Prentice-Hall, 1994.

[9] Wright, C. H. G., Welch, T. B., Morrow, M. G., and Gomes, W. J., "Teaching Real-World DSP Using MATLAB and the TMS320C31 DSK," *Proceedings of the 1999 ASEE Annual Conference*, session 1320-06 [CD-ROM], June 1999.

Commander THAD B. WELCH, PhD, is an Assistant Professor in the Department of Electrical Engineering at the U.S. Naval Academy (from 1994-1997 he was an Assistant Professor in the Department of Electrical Engineering at the U.S. Air Force Academy). His research interests include multicarrier communication systems analysis and signal processing. He is a member of ASEE, IEEE, and Eta Kappa Nu. Email: t.b.welch@ieee.org

Major CAMERON H.G. WRIGHT, PhD, PE, is an Associate Professor in the Department of Electrical Engineering at the U.S. Air Force Academy. His research interests include signal and image processing, biomedical instrumentation, communication systems, and laser/electro-optics applications. He is a member of ASEE, IEEE, SPIE, NSPE, Tau Beta Pi, and Eta Kappa Nu. Email: c.h.g.wright@ieee.org

Lieutenant Commander MICHAEL G. MORROW, PE, is an Instructor in the Department of Electrical Engineering at the U.S. Naval Academy. His research interests include real-time digital systems, power system automation, and software engineering. Email: morrow@nadn.navy.mil