Demonstrating in the Classroom the Ideas of Frequency Response

Dr. Mark A. Hopkins, Assoc. Prof.

Electrical Engineering Dept., Rochester Institute of Technology

Abstract

One of the most persistently difficult concepts to communicate to students is that of "frequency response", because it spans both the frequency-domain and the time-domain. This paper presents a Matlab®-based graphical method of demonstrating several important relationships among pole/zero locations, Bode plot (*i.e.*, Fourier transform), and time-domain sinusoidal response.

The author has been using this method successfully for several years in a variety of systems-related courses to help students understand these relationships. The software is also easily available to students, so they may reconstruct classroom demonstrations, and do much more.

In the demonstrations, Bode magnitude and phase plots are programmatically linked to plots of the pole/zero map and of the time-domain sinusoidal response. The frequency of the stimulus sinusoid can easily be changed in any one of the plots, with the change automatically propagated to all the other plots. All of the plots have interactive capability, and display much more information than simple line graphs.

These plots are used to enhance discussion of (1) minima and maxima in the Bode magnitude plot and their relationship to pole/zero locations, (2) graphical (vector) analysis of the Fourier transform, (3) steady-state response versus transient response, (4) steady-state magnitude and phase and their relationship to the Bode plots, and (5) phase-lead and phase-lag.

Complete details of the method are presented. The software is available on the web, and is free if used only for educational purposes.

1. Introduction

The ability to use computers at the lectern, enabled by relatively cheap projection equipment, has tremendous potential for computer-aided teaching. This is not a reference to being able to present PowerPoint slides, using computers in essentially the same way we once used overhead projectors to show transparencies. Rather, we now have the ability to create accurate mathematical plots on-the-fly, and dynamically manipulate graphical content to emphasize points of discussion.

This paper is about realizing that potential for the purpose of teaching frequency-response concepts. This is an area that requires a teacher to present several different types of plots – time-domain response plots, frequency-response (*e.g.*, Bode) plots, and pole/zero maps in the *s*-plane and *z*-plane – and to discuss their inter-relatedness.

Here are some of the problems using computers to create plots "on-the-fly" in the classroom:

- delays result from taking time to create plots,
- delays result from making mistakes while creating plots,
- plots are not readable from everywhere in a classroom,
- plots are not interactively modifiable, or it is not easy to modify them,
- plots do not clearly and easily illustrate the points of a lecture, and
- the lecture is difficult, or even impossible, for students to re-create from their notes.

This paper presents a Matlab® add-on toolbox, named **`pzgui`**, that helps overcome these difficulties, and gives teachers a unified graphical environment in which to demonstrate many valuable insights into the difficult topic of frequency response. The name **`pzgui`** is the (copyright-protected) acronym of "**p**ole/**z**ero **g**raphical **u**ser **i**nterface".

2. Plots Used in Teaching Frequency-Response Concepts

In linear system analysis, frequency-response is fundamental. The plots typically used to demonstrate most of the concepts are:

- a pole/zero map that, together with the gain, specifies the model of a linear system,
- a time-domain plot showing the sinusoidal response of a linear system, and
- steady-state magnitude and phase of sinusoidal response, versus frequency (Bode plots).

Examples of these plot types were generated using the **`pzgui`** toolbox, and are shown in Fig. 1 through Fig. 4, which have been manipulated so that there are not any *obvious* "linkages" among them, and only a few of the **`pzgui`** user controls are shown.

Fig. 5 through Fig. 8 show the same set of plots after a new frequency, 3 Hz, has been selected. The selected frequency can be changed in any one of the four plots, as discussed below. Also, in Fig. 5 through Fig. 8, all of the **`pzgui`** user controls are visible.

Figure 1. Pole/zero map showing the model of a linear system (not including gain).

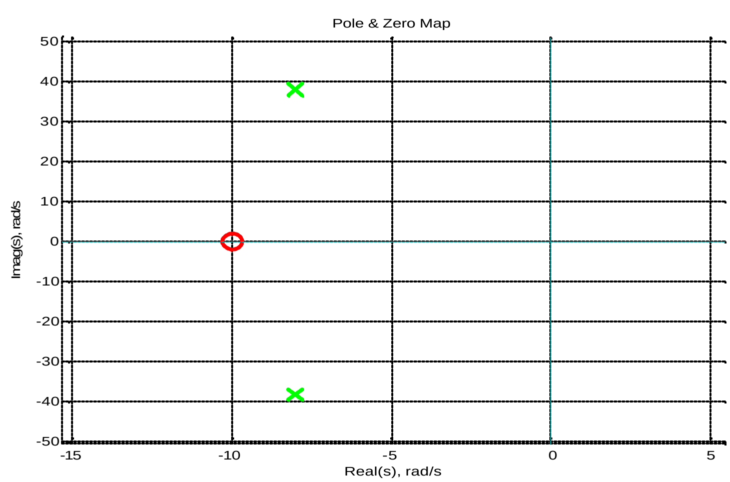


Figure 2. Time-domain plot of the sinusoidal response to a 2.4 Hz sinusoid.

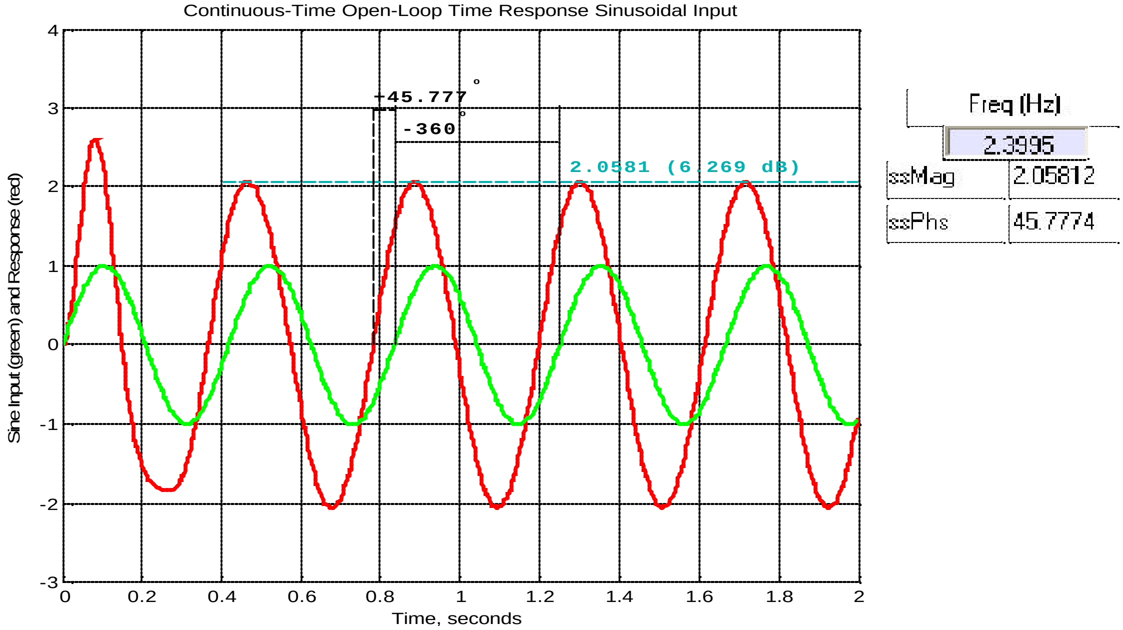


Figure 3. Bode magnitude plot of the system.

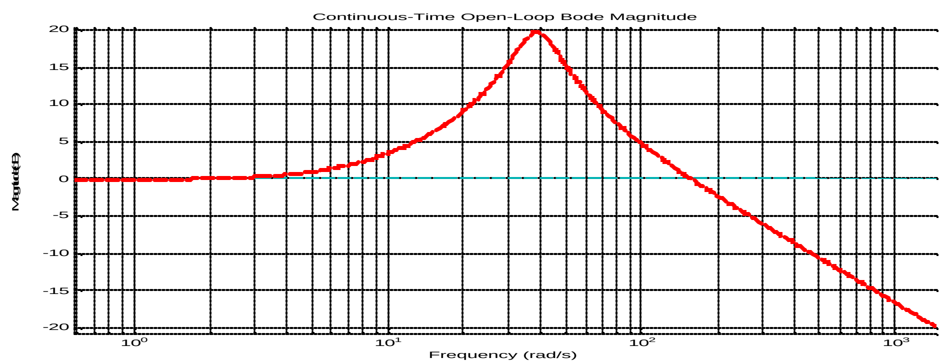
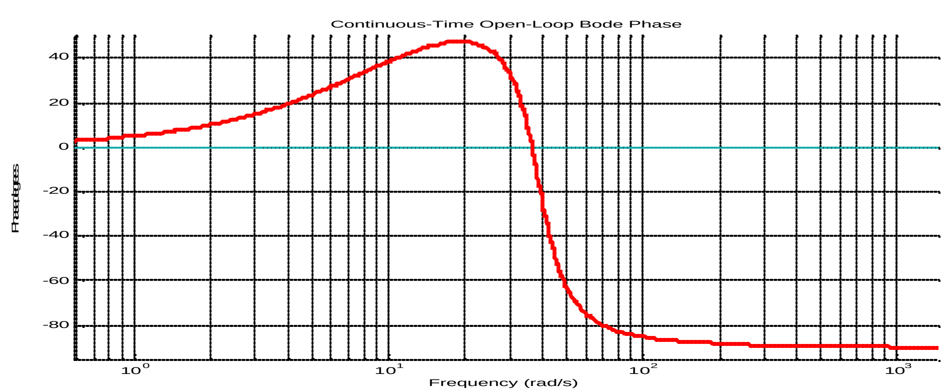


Figure 4. Bode phase plot of the system.

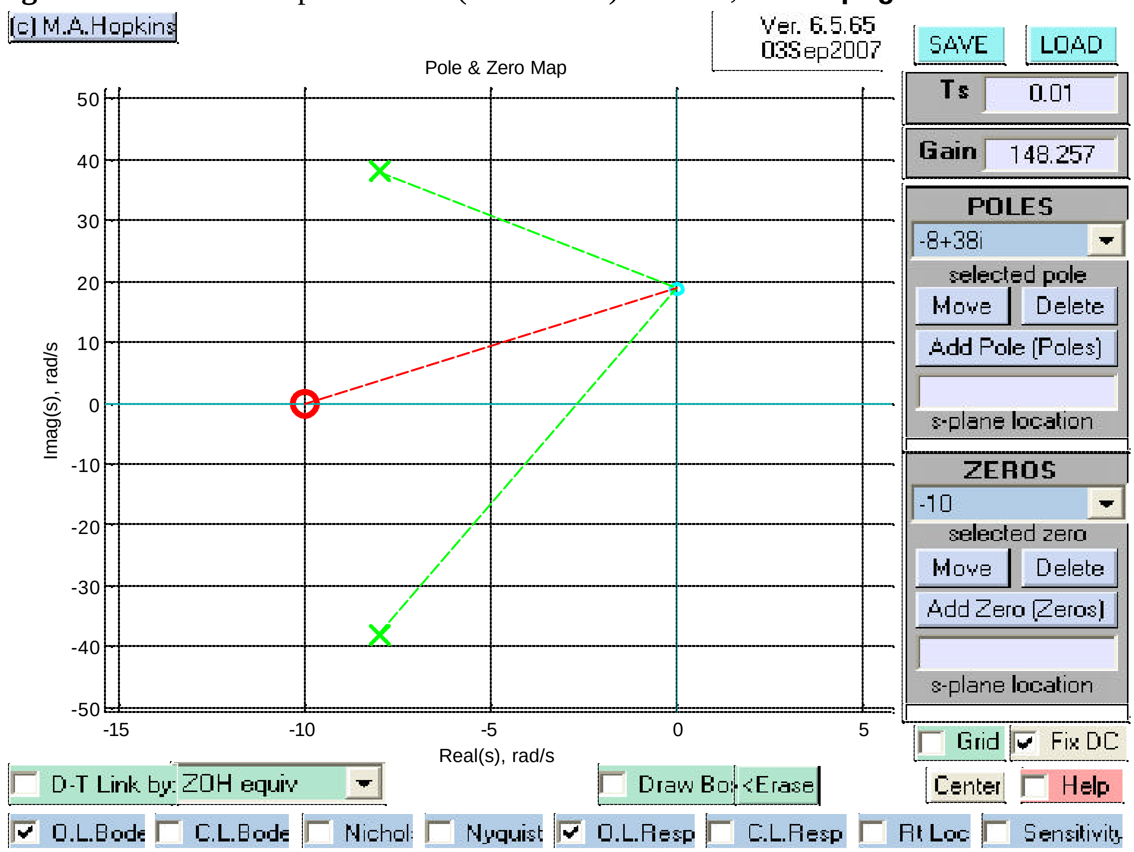


2.1. The Pole/Zero GUI

The models in Figs. 1 and 5 are in zero/pole/gain (Z/P/K) format, $H(s) = \dfrac{K(s-z_1)\cdots(s-z_m)}{(s-p_1)\cdots(s-p_n)}$.

Fig. 5 shows the full set of controls in the pole/zero GUI. Some of the controls are not germane to this discussion, but most are. The controls along the bottom allow the user to easily create various open-loop (O.L.) and closed-loop (C.L.) plots related to the system being modeled. In particular, notice that there are checkmarks in the boxes labeled "**O.L.Resp**" and "**O.L.Bode**". Left-clicking on those controls creates the three plots shown in Fig. 6 through Fig. 8.

Figure 5. Pole/zero map with 3 Hz (18.85 rad/s) selected, and all **`pzgui`** controls visible.



Notice the pole/zero map in Fig. 5 is annotated with vectors that terminate at a selected frequency on the jw-axis and originate at the various poles and zeros in the model. A different frequency may easily be selected by left-clicking on some other point along the jw-axis. When a different frequency is selected, the other plots are automatically updated accordingly, too.

The vector annotations help demonstrate

- the relationship between Laplace and Fourier transforms,
- the meaning of the Fourier transform, in terms of frequency response plots,
- why an extremum occurs in the magnitude plot when a pole (zero) is near the imaginary axis, and why the extremum occurs near the imaginary part of that pole (zero), and

- the concept that a rational polynomial is a product of complex-plane vectors in the numerator and in the denominator.

In Fig. 5, the location of a pole can be changed easily, in either of two ways. If the mouse cursor is positioned above a pole, then it may be dragged-and-dropped at a new location by right-clicking on it. A more accurate change may be made by entering the new pole location numerically in the box labeled "**s-plane location**" in the frame marked "**POLES**", and then left-clicking on the button labeled "**Move**". The pole that will be moved is the one currently selected in the drop-down menu, which lists all of the poles in the model. A similar description applies to changing the location of a zero, except those controls are in the frame labeled "**ZEROS**".

In Z/P/K form, the gain parameter *K* is usually not equal to the DC-gain of the model. When poles and zeros are moved, added, or deleted, if *K* is held constant, then the DC-gain typically changes. In the **`pzgui`**, the user has an option to maintain the same DC-gain when poles and zeros are changed. This option is put into force by checking the box labeled "**Fix DC**", just below the "**ZEROS**" frame. This option is checked by default, when the pole/zero GUI is created. When DC-gain is fixed, the parameter *K* is recomputed so as to maintain the same DC-gain, any time poles or zeros are changed (unless the DC-gain is infinite or zero).

In Fig. 5, poles and zeros can easily be added to or deleted from the model, except that the number of zeros is not allowed to exceed the number of poles. The "**Delete**" button deletes the selected pole (or zero). The "**Add Pole**" and "**Add Zero**" buttons use the numerical entry from the corresponding "**s-plane location**" window (in their respective frames) to place a new pole or zero. Complex-conjugate pairing is automatically enforced in all changes to the poles and zeros.

2.2. The Time-Response Plot

Fig. 6 shows the GUI controls associated with the time-response plot. The frequency of the sinusoid is shown in the display window labeled "**Freq (Hz)**". This is an interactive element, by means of which a different frequency may be specified. When the frequency is changed, the other plots are automatically updated accordingly.

The simulation input is always a unit-magnitude sine-function, plotted in green. The output (system response) is plotted in red. The maximum simulation time may be changed using the editable window labeled "**Max Time**". The user controls below the "**Max Time**" window are related to time-domain performance measures, typically applied to a system's step-response (*i.e.*, not usually related to frequency response).

Provided that the simulation time extends far enough into the steady-state condition, annotations of steady-state magnitude and phase automatically will be shown in the plot, as in Figs. 2 and 6.

In Fig. 6, the magnitude annotations (lines and text) are designed to help demonstrate

- the concept of output steady-state,
- the magnitude relationship of the steady-state output sinusoid to the input sinusoid,
- the relationship between the steady-state output and the Bode-magnitude plot, and
- the relationship between raw magnitude and decibels.

In Fig. 6, the phase annotations (lines and text) are designed to help demonstrate

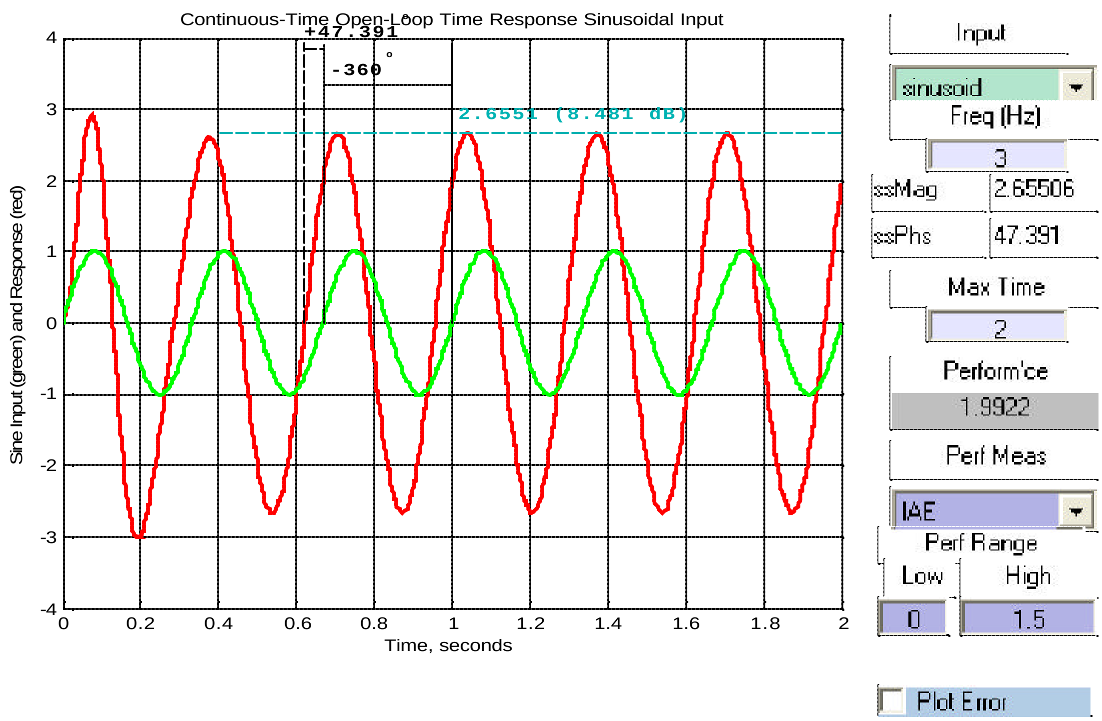
- the phase relationship of the steady-state output sinusoid to the input sinusoid,

- the meaning of phase-lead and phase-lag,
- the meaning of 360 degrees in terms of elapsed-time, and
- the relationship between the steady-state output and the Bode-phase plot.

2.3. The Bode Magnitude and Phase Plots

Fig. 7 shows the Bode magnitude plot with annotations related to the selected frequency, which is indicated by a small cyan-colored circle, and a text description in the frequency-axis label. The selected frequency can easily be changed, by positioning the cursor over the magnitude plot line at the desired frequency, and left-clicking the mouse. When the selected frequency is changed, the other plots are automatically updated accordingly.

Figure 6. Simulated response to a 3 Hz sine wave, with all **`pzgui`** controls visible.



The three controls at the lower-left allow the plot to be used to help demonstrate

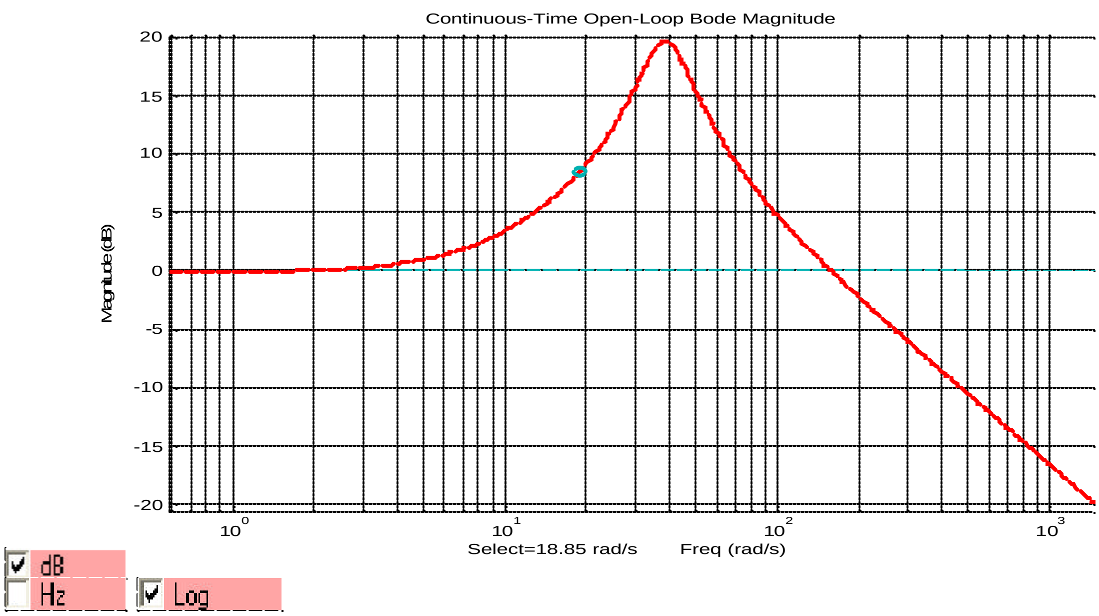
- the relationship between raw magnitude and decibels,
- the reasons why frequency response magnitude is usually plotted in decibels,
- the difference between plotting frequency in hertz and plotting it in radians/second, and
- the reasons why frequency response is usually plotted versus the log of frequency.

Fig. 8 shows the Bode phase plot with annotations related to the selected frequency, which is indicated by a small cyan-colored circle, and a text description in the frequency-axis label.

The selected frequency can easily be changed, by positioning the cursor over the phase plot line at the desired frequency, and left-clicking the mouse. When the selected frequency is changed, the other plots are automatically updated accordingly.

The two controls at the lower-left allow the plot to be used to help demonstrate the difference between plotting frequency in hertz and plotting it in radians/second, and to switch between linear and log frequency-axis.

Figure 7. The Bode magnitude of the Z/P/K model, with all **`pzgui`** controls visible.



In Fig. 7 or 8, when the user changes the frequency-axis scaling (log or linear) or "zooms" to a different range of frequency, the same change is made to the scaling or range of the other figure. That way, the magnitude plot and the phase plot are always directly comparable because they display the same frequency range, and use the same scaling along the frequency axis. This linkage of Fig. 7 and Fig. 8 saves time during discussion of the relationships between the magnitude and phase plots.

3. Conclusion

In order to create the plots of Fig. 1 through 4 "on-the-fly", ultimately, program instructions must be entered into a computer. Although applications such as Matlab® have made plot creation fairly easy, it is nevertheless challenging to quickly create and format plots in such a way that they are easily readable in a classroom, and also easily and clearly demonstrate important points of a discussion. Making those plots dynamically and easily recomputable, as in Fig. 5 through 8, is even more challenging.

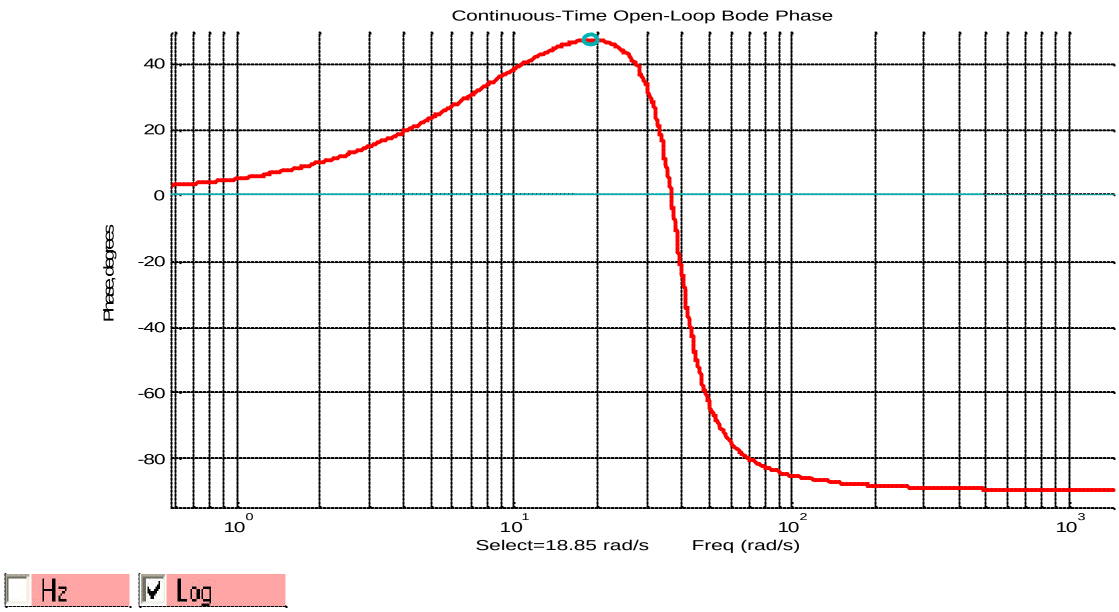
The **`pzgui`** toolbox quickly and unobtrusively handles the low-level programming details of creating and updating the interlinked plots of Fig. 5 through 8, and servicing their user controls. Taken altogether, these four dynamically interactive plots provide a rich set of tools that can be effectively used to illustrate all of the essential ideas in discussions of frequency response.

Furthermore, students can produce the same plots on their own computers, or in a school's computer labs, if they have the **`pzgui`** toolbox and access to Matlab® and the Matlab® Controls Toolbox. Provided with a simple "script", students can even re-create an entire interactive demonstration. Perhaps equally important, **`pzgui`** provides a "*playground*" in which students are able to do much more than brief classroom demonstrations can show.

The **`pzgui`** toolbox comprises a set of m-files that are available at **`people.rit.edu/maheee`** in a zip-file. This toolbox is free to faculty and students (provided that it's used only for educational purposes), and it runs on Matlab® ver.6.5.0, and all later versions. The zip-file also includes a brief *User's Manual* in PDF format.

The required software infrastructure for **`pzgui`** is quite accessible. The Student Version of Matlab® (which includes the Controls Toolbox, Simulink, and several other toolboxes) is available directly from its publisher, *The Mathworks*, for about $100 US, and is also stocked in some university bookstores. *The Mathworks* also provides relatively cheap site-licensing for educational institutions, many of which have enough "roaming" licenses available in their computer labs that students effectively have "24/7" on-demand access to Matlab® in those labs.

Figure 8. The Bode phase of the Z/P/K model, with all **`pzgui`** controls visible.



Biographical Information

MARK A. HOPKINS
An Associate Professor in the Electrical Engineering Department, Kate Gleason College of Engineering, Rochester Institute of Technology, Dr. Hopkins earned his Ph.D. in Electrical Engineering from Virginia Tech, in 1988. His main research interests are in the modeling and control of large flexible structures. He has an abiding interest in the computer-aided teaching of linear systems, modeling, and control. He has been developing **`pzgui`** since 1996.