

Interactive Multi-Domain System Analysis and Design

G.E. Smid and Ka C. Cheok

Department of Electrical & Systems Engineering, Oakland University
Rochester Hills, Michigan 48309-4401, USA.

Abstract

This paper presents a MATLAB tool for interactive transfer function analysis which can be used to enhance students and engineers understanding of system characteristics and behaviour. It allows a user to develop an intuitive feeling for the relationship between time and frequency domains, and locations of roots of a transfer function.

Keywords : System characteristics, Interactive Visualization, MATLAB.

1 Introduction

It has become a common practice to use numerically highly intensive algorithms to design and analyze control systems. Iterative, recursive and search methods are part of the standard toolbox for a control system designer. Examples are the increasing use of MATLAB (which is after all a numerical tool), μ -synthesis for Robust control (in which the non-convex problem is solved using D-K-iterations [4]), applications of genetic algorithms and learning schemes for neural networks. This trend is capable of training a generation of engineers who can design control systems on the fly, an yet do not have a full understanding of the dynamics and relations that account for a systems behaviour.

On the other hand there are control system scientists, who solve for controllers in a highly mathematical fashion, not rarely only in the frequency domain. Also here the scientists have often lost the touch with a physical system behaviour.

Indications from R&D centers in industry

point out that there is an increasing need for a *systems* approach towards design. A top down approach and thus a comprehensive understanding of the system urges to be the winning method for modern control engineering.

The MATLAB software tool described in this paper allows for interactively designing a system described by a linear transfer function. During the design process in either the Laplace domain, the frequency domain or the time domain, the system characteristics will be simulated on-line in the other domains. The effect and applications for interactive design methodology will be discussed afterwards. Similar work has been done by [2].

2 Interactive Design

Figure 1 shows the MATLAB graphical interface window for the interactive design of a linear system, represented by the transfer function

$$G(s) = K \frac{(s + z_1)(s + z_2) \dots (s + z_p)}{(s + r_1)(s + r_2) \dots (s + r_q)} \quad (1)$$

$$= \frac{b_0 s^m + b_1 s^{m-1} + \dots + b_m}{s^n + a_1 s^{n-1} + \dots + a_n} \quad (2)$$

The transfer function equations are displayed in a text window. Three graphs are displayed for this system, that present the systems characteristics. They are the pole-zero plot, the Bode diagram and the step response. The curves can be generated with MATLAB, once the numerator and denominator are known.

A user can interact with the software using keyboard and mouse to

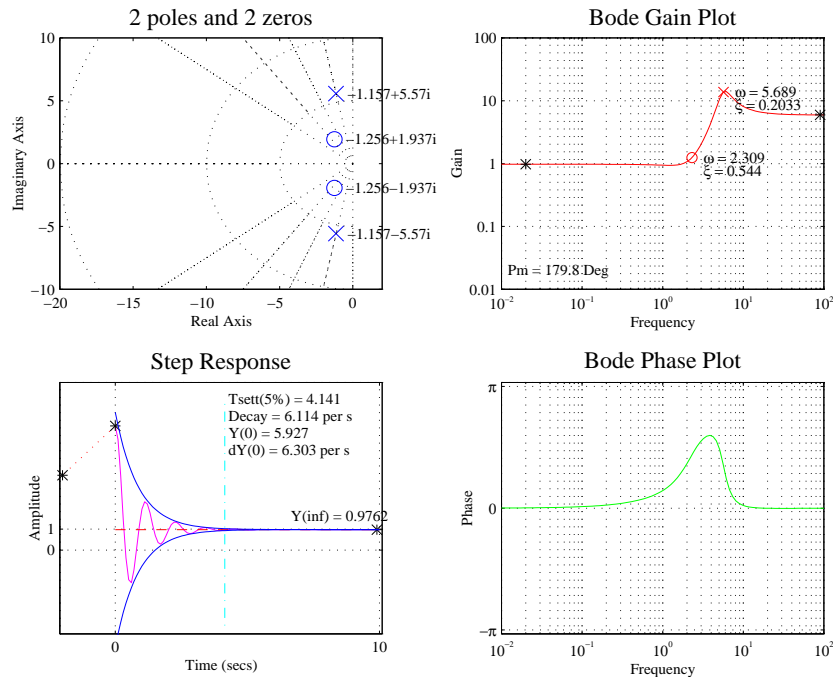


Figure 1: Screen Shot of the main interface window in the MATLAB Simulator.

1. design in the complex domain by moving, adding or deleting poles and zeros on the s-plane,
2. design in the frequency domain by modifying the shape of the gain in the Bode diagram,
3. design in the time domain by modifying the shape of the transient in the step response.

The software will draw a root-locus in the Laplace domain when roots are changed, due to interaction in the frequency or time domain. Additional functionality is listed by typing `help idesign` on the MATLAB command line.

We must point out that the effect of interactive tools can hardly be explained in words. The appreciation is beyond doubt after hands-on experience with the software. A copy can be received by sending to `smid@oakland.edu`. A live demo will be part of the presentation of this paper.

In the next sections it is explained how the sequence for interaction in either domain is

actually achieved. In each domain it is the objective to derive a new transfer function, that can be simulated in the three domains. Most of the derivations can be found in control textbooks like [1].

2.1 Laplace Domain

Dragging a cross or circle in the pole-zero plane will relocate the poles and zeros of the system respectively. Pressing `z` will add a zero at the current mouse location, pressing `p` will add a pole and pressing `space` will delete either.

When the imaginary parts of two complex conjugate poles or zeros are very small, they will be merged to one. Similarly, when a real pole or zero is dragged off the real axis, it will be split into a complex conjugate pair.

Obviously, changing the poles and zeros will automatically return the new coefficients for the numerator and denominator.

2.2 Frequency Domain

To represent poles and zeros on the frequency axis, their natural frequency ω_i must be derived from their distance in the Laplace plane, to the origin. So, for a pole p_i ,

$$\omega_i = |p_i| = \sqrt{\text{Re}\{p_i\}^2 + \text{Im}\{p_i\}^2}. \quad (3)$$

The damping ratio ζ_i for a pole is represented by its angle with the real axis. Generally, for a complex pole p_i ,

$$\zeta_i = -\cos \angle p_i = -\frac{\text{Re}\{p_i\}}{|p_i|}. \quad (4)$$

The steady state gain can be found from Equation (2) as

$$K_{DC} = \lim_{s \rightarrow 0} G(s) = \frac{b_m}{a_n} \quad (5)$$

If $m < n$, the system is strictly proper and will thus attenuate for high frequencies. The HF gain can be calculate as

$$K_{HF} = \lim_{s \rightarrow \infty} G(s) = \frac{b_0}{|s^{n-m}|}. \quad (6)$$

Poles in a system represent phase lags in the Bode diagram. A single real pole (a lag) will cause the gain-frequency curve to decrease with a slope of -20 dB/dec. A complex double pole causes a slope of -40 dB/dec. A complex double pole also shows a narrow band resonance gain, proportional to its damping. Zeros represent phase leads and will cause an increase in gain for an increasing frequency, in a similar fashion.

Poles and zeros can be relocated to change either the frequency or damping. A set of complex conjugate poles or zeros will be merged to a real pole or zero, if their damping ratios ζ_i becomes 1. Similarly, if a real zero is moved down or a real pole is moved up, it will be split into a double zero or pole.

For example, suppose the system to be analyzed has one zero and two poles, $z_1 = 0.5$, $p_1 = 0.4 + 4i$ and $p_2 = 0.4 - 4i$. This system is represented by $G(j\omega)$ in Figure 2, which is the superposition of the zero (given by $Z(j\omega)$)

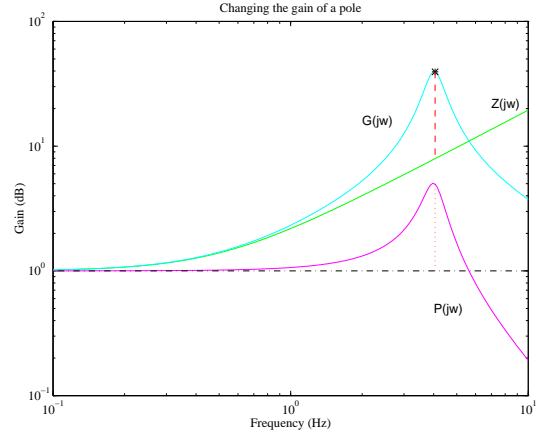


Figure 2: The System is factorized and superposed, when a pole or zero is changed.

and of the complex poles (given by $P(j\omega)$). Notice that the dotted gain for the resonance peak of the double pole is equal to the gain of the total system relative to the trend of the zero.

To change a pole p_m of the system, the following sequence of actions will take place.

1. Eliminated the pole from the transfer function
2. Calculate the damping from the desired gain relative to the gain of the remaining system $H(s)$.
3. From this gain and the desired frequency, derive a pole or double pole
4. Add the new pole or pole-pair to $H(s)$ by convoluting it in the denominator.

where

$$\begin{aligned} H(s) &= G(s)(s + p_m), \text{ if real} & (7) \\ &= G(s)(s + p_m)(s + \bar{p}_m), \text{ if cplx} & (8) \end{aligned}$$

The gain offset, relative to $H(s)$ is given by

$$\partial H = \frac{G_m}{|H(s)|}, \quad (9)$$

where G_m is the desired gain, which is just the amplitude, pointed at with the mouse in the Bode diagram, hence the y-position of the

mouse. If $\partial H < 1$, then a real pole is assigned at the current frequency. Thus

$$G(s) = H(s) \frac{1}{(s + \omega_m)}, \quad (10)$$

where ω_m is the frequency which the mouse points at, hence the x-position of the mouse. To attenuate further for this frequency, the pole should be moved to a lower frequency or a zero should be added.

If $\partial H > 1$, the mouse points above the system gain curve. Then the damping can be calculated as $\zeta_m = \arctan(\partial H)$ and the complex pole becomes

$$p_m = -\omega_m (\cos(\zeta_m) + i \sin(\zeta_m)) \quad (11)$$

The desired system is calculated as

$$G(s) = H(s) \frac{1}{(s + p_m)(s + \bar{p}_m)} \quad (12)$$

The similar derivation for changing a zero holds, however the conditions for ∂H are inverse.

Phase and Gain margins are printed in the Bode diagram, when ‘i’ is pressed on the keyboard. They are given by

$$M_{\text{gain}} = 20 \log \left| \frac{1}{G(j\omega)} \right| \Big|_{\angle G(j\omega) = \pm\pi} \quad (13)$$

and and the phase margin by

$$M_{\text{phase}} = \pi + \angle G(j\omega) \Big|_{|G(j\omega)|=1}. \quad (14)$$

2.3 Time Domain

The unit step response $g(t)$ is used to illustrate the system response characteristics in the time-domain. Depending on the system, three characteristics can be changed interactively in the step-response. They are the initial value, the initial slope and the steady state gain K_{DC} .

For the first two characteristics we make use of the initial value theorem. Since the step response is the integral of the unit impulse response, the initial value and the initial slope can be determined from

$$\lim_{t \rightarrow 0} g(t) = \lim_{s \rightarrow \infty} G(s), \quad \lim_{t \rightarrow 0} \dot{g}(t) = \lim_{s \rightarrow \infty} sG(s). \quad (15)$$

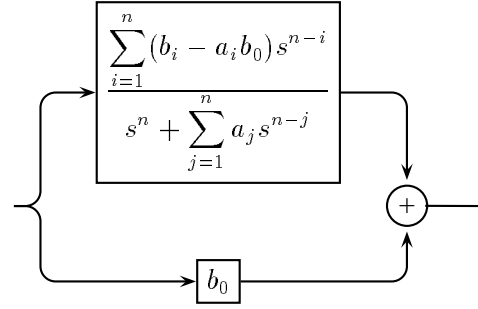


Figure 3: Separation of terms of a non-strictly proper system

The transfer function for a non-strictly proper system with $m = n$, can be split in terms as

$$G(s) = b_0 + G'(s), \quad (16)$$

with

$$G'(s) = \frac{\sum_{i=1}^n (b_i - a_i b_0) s^{n-i}}{s^n + \sum_{j=1}^n a_j s^{n-j}}. \quad (17)$$

which is schematically represented in Figure 3. The impulse response for such a system will be b_0 times the impulse input plus the impulse response of $G'(s)$. Therefore the slope at $t = 0$ is infinity. The slope at $t = 0^+$, however, can be found as

$$\lim_{t \rightarrow 0^+} \dot{g}(t) = \lim_{s \rightarrow \infty} sG'(s). \quad (18)$$

From (16) and (18) it can be seen that the unit step response of a system with $m = n$ has an initial value given by $\lim_{t \rightarrow 0} g(t) = b_0$ and an initial slope given by $\lim_{t \rightarrow 0^+} \dot{g}(t) = b_1 - a_1 b_0$. For strictly proper systems with $n = m - 1$, the initial value for the step response will be zero, but the initial slope is calculated as $\lim_{t \rightarrow 0} g(t) = \lim_{s \rightarrow \infty} G(s) = b_0$. For systems with $m < n - 1$, the initial value and the initial slope will be zero, $\lim_{t \rightarrow 0} g(t) = 0$ and $\lim_{t \rightarrow 0} \dot{g}(t) = 0$.

The envelope $E(t)$ for any proper system $F(s)$ can be found by considering the inverse Laplace transform

$$f(t) = \frac{1}{j2\pi} \int_{\sigma-j\infty}^{\sigma+j\infty} F(s) e^{-ts} \delta s \quad (19)$$

which can be solved using the residue theorem. By summing the residues with real exponent, the envelope $E(t)$ of $G(t)$ can be found as

$$E(t) = K_{DC} \pm \sum_{p=1}^m |R| e^{-\zeta_p \omega_n t} \quad (20)$$

where

$$R = (s + p_p)G(s) \Big|_{s=p_p}. \quad (21)$$

The steady state gain can be found using the Final Value Theorem as shown in Equation (5).

With the above relations between the coefficients of the system transfer function and the slope and initial value, new values for the numerator and denominator can be derived interactively.

3 General Applications

Because it provides a graphical interface for intuitive interaction, this MATLAB software serves well for controls courses. Concepts like stability, gain and phase margins, settling and rise times are integrated in one design step. Especially for students, the effect of interactive tools for control education has been proved to be successful [3].

Controller design specifications for classical control, pole placement, model reference, optimal control, etc are applications for this tool. A time domain constraint like settling time or maximum value can be set, and the set of poles and zeros can be derived.

As an application of advanced control, the tool could be used to shape the D-scalings that may be used to bound the SSV in the frequency domain during the process of μ -analysis [4]. In the same fashion it could be used to shape filters of any shape and order. The software can also be readily extended for optimal control design by adding indications of performance indexes.

A minor modification will allow the software to display a Fourier transform of an arbitrary signal in the frequency graph, in order for a designer to fit a frequency behaviour of any kind interactively.

The size and accessibility of the code allows any engineer make other modifications to interactively design in the Nyquist domain or the impulse response as well.

4 Conclusion

Effective use of numerical tools can help in understanding systems. The interaction with system characteristics and the ability to see the effect of changing characteristics in the different domains dynamically can enhance designers' decision making.

References

- [1] R. C. Dorf. *Modern Control Systems*. McGraw Hill, 1995.
- [2] W. K. Durfee, M. B. Wall, and et al, "Interactive Dynamics and Control Collection Software" (for UNIX), M.I.T. Website.
- [3] J. S. Glower. "Design of Computer Simulation Experiments for an Introductory Controls Course", *American Control Conference*, pages 569–573, 1995.
- [4] H. Rotstein and O. Kidron. "On Fitting a Transfer Function to Magnitude Data", *American Control Conference*, pages 2079–2080, 1995.