Using MATLB for stability analysis in Controls engineering

Cyrus Hagigat Ph.D., PE
College of Engineering
University of Toledo, Toledo, Ohio

Abstract

Analyses of control systems require solution of differential equations. Such solutions demonstrate performance in time domain and are used for determining stability boundaries. However, solving the differential equations for moderately complex control systems without use of numerical techniques are at times impossible.

MATLAB can be used to solve differential equations and MATLAB can be used to quickly construct a Root-Locus plot for a control system that determines the stability boundaries for the control system without the need to solve the differential equations. This article demonstrates use of MATLAB for both solving system differential equations and constructing Root-locus plots.

Introduction

The concepts discussed in this article are using MATLAB to generate Root-Locus plots for control systems and using the plots to determine critical boundaries for control system analysis, and using MATLAB to determine the time response of control systems.

The application of Root-Locus plots in control system analysis are determination of parameters that lead to an unstable behavior of a control system and determining boundaries where the control system behavior changes in the stable region. Root-Locus plots can be generated by manual techniques. However, the manual techniques are time consuming. MATLAB can be used to generate Root-Locus plots quickly.

One of the goals of a control system analysis is determination of the response of the system in time domain. MATLAB can be used to determine the time domain response of a control system in most instances. However, there can be situations where MATLAB cannot be used for this purpose because the results while mathematically correct are not in a readily usable format. In this article application of MATLAB and its limitations for time response analysis of control systems are discussed.

Nomenclature

G(S): Transfer function of a closed loop control system in Laplace domain.

H(S): Feedback multiplier of a control system in Laplace domain.

R(S): Input to a control system in Laplace domain.

C(S): Output of a control system in Laplace domain.

K: Constant that can be varied to change control system behavior.

S: Parameter in Laplace domain used to describe elements of the control system.

f(t): Control system response in time domain.

Technical Discussion

A closed loop control system in its simplest form in Laplace domain is represented by figure 1.

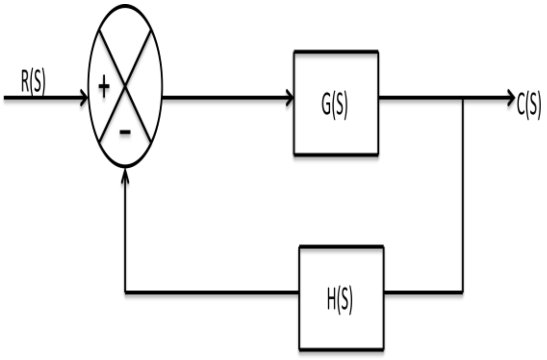

Figure 1: Representation of a closed loop control system in its simplest form in Laplace domain

The open loop equivalent of the system of figure 1 is shown in figure 2.

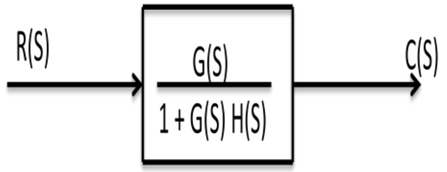

Figure 2: open loop representation of the Closed loop system of figure 1

An example of a system of the type of figure 1 is shown in figure 3 [1]. Figure 3 represents a preliminary design that contains a gain factor K which is yet to be determined.

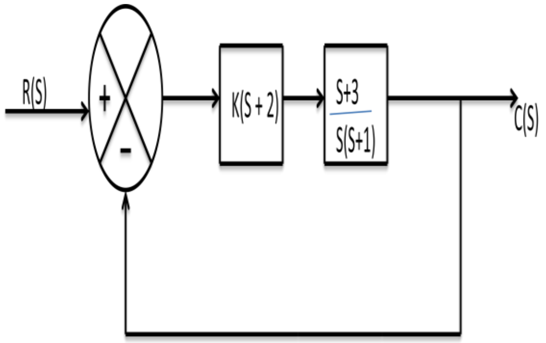

Figure 3: Example of a typical controls block diagram
having a gain factor of K

in order to determine the numerical values for use in MATLAB for getting a Root-Locus diagram the "G(S)H(S)" of the closed loop control block without the gain factor K must be determined.
For the block diagram of figure 3 the "G(S)H(S)" factor is shown in equation (1).

$$\frac{(S+2)(S+3)}{S(S+1)} \quad (1)$$

Equation (1) once simplified becomes equation (2).

$$\frac{S^2 + 5S + 6}{S^2 + S} \quad (2)$$

In order to plot the Root-Locus plot for the expression of equation (2), the following sequence of codes must be used in MATLAB.

```
num= [1 5 6];

den= [1 1 0];

rlocus(num,den);

v=[-4 1 -1 1]; axis(v)
```

MATLAB generates the Root-Locus plot shown in figure 4.

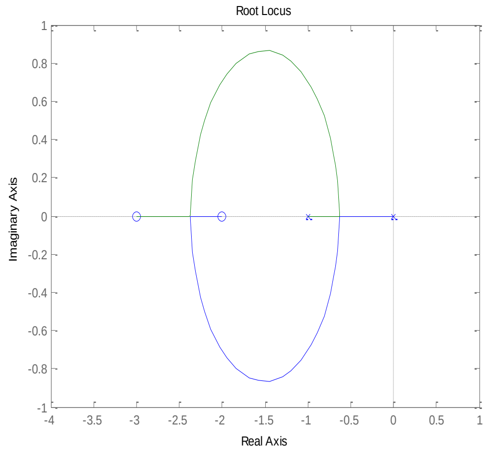

Figure 4: Root-Locus plot of system of figure 3

Since the entire Root-Locus plot of figure 4 is to the left of the imaginary axis, it is concluded that the system of figure 3 is stable for all values of K. The type of Root-Locus plot shown in figure 4 points to critical system performance differences based on the location where the plot crosses the real axis.

By using the formula “1 + G(S)H(S) = 0”, an expression for K for the system of figure 3 can be calculated [2]. Using this technique, K for system of figure 3 is determined to be as shown in equation (3).

$$K= \frac{-S(S+1)}{(S+2)(S+3)} \qquad (3)$$

Setting dk/ds=0 results in a polynomial equation that when solved results in two values for S. They are

$S_1 = -0.634$ & $S_2 = -2.366$.

Substituting the values of S_1 & S_2 into equation (3) results in $K_1 = 0.0718$ & $K_2 = 14$. K_1 & K_2 are the K values that correspond to the Root-Locus plot of figure 4, and therefore they are the K values that form the boundaries where the system of figure 3 behaves differently. Based on the values of K_1 & K_2 there are three distinct ranges of interest for K. They are: $K<0.0718$, $0.0718 <K< 14$ & $K>14$.

In order to evaluate the three ranges for K, the following three values corresponding to the three ranges are chosen for K.

K= 0.05

K= 7

K= 15

Using the formula of figure 2, the open loop transfer function for the system of figure 3 becomes

$$\frac{K(S+2)(S+3)}{S^2 + S + K(S^2 + 5S + 6)} \qquad (4)$$

Substituting the K values of 0.05, 7 and 15 into equation (4) results in formulas (5), (6) & (7) for the transfer function.

$$\frac{0.05\ S^2 + .25\ S + .3}{1.05\ S^2 + 1.25\ S + .3} \qquad (5)$$

$$\frac{7\ S^2 + 35\ S + 42}{8\ S^2 + 36\ S + 42} \qquad (6)$$

$$\frac{15\ S^2 + 75\ S + 90}{16\ S^2 + 76\ S + 90} \qquad (7)$$

Equations (5), (6) & (7) can be used in MATLAB to obtain the system response for various inputs.

The following is the MATLAB script for getting an impulse response for the system represented by equation (5).

```
num= [.05 .25 .3];
den= [1.05 1.25 .3];
impulse (num, den)
```

Figure 5 is the impulse response for system of equation (5) generated by MATLAB.

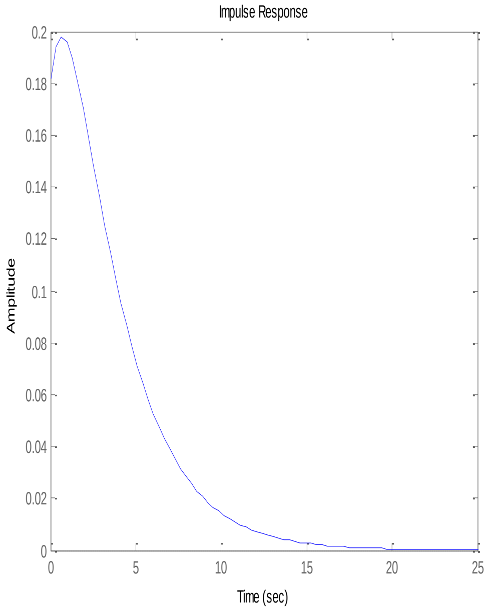

Figure 5: Impulse response of system of equation (4) for a K value of .05

The following is the MATLAB script for getting an impulse response for the system represented by equation (6).

```
num= [7 35 42];
den= [8 36 42];
impulse (num, den)
```

Figure 6 is the impulse response for system of equation (6) generated by MATLAB.

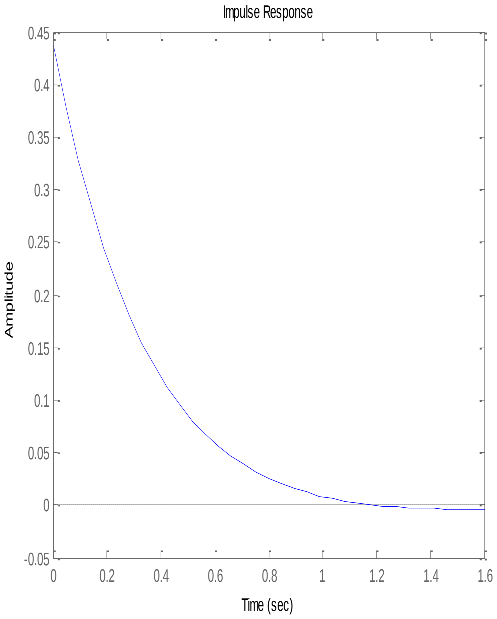

Figure 6: Impulse response of system of equation (4) for a K value of 7

The following is the MATLAB script for getting an impulse response for the system represented by equation (7).

```
num= [15 75 90];
den= [16 76 90];
impulse (num, den)
```

Figure 7 is the impulse response for system of equation (7) generated by MATLAB.

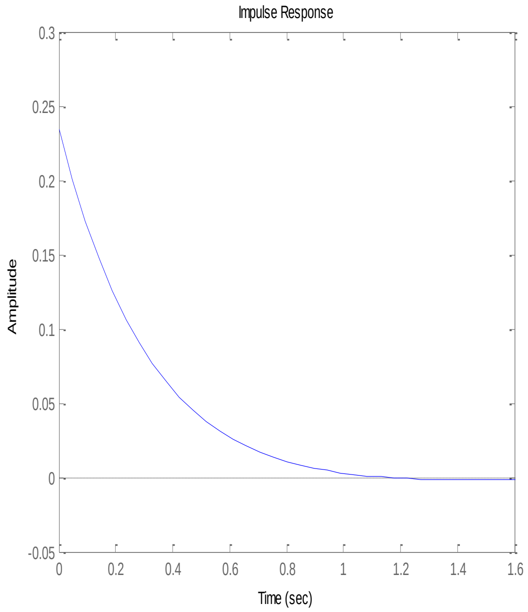

Figure 7: Impulse response of system of equation (4) for a K value of 15

A comparison of figures 5, 6 & 7 shows that the system responds differently to a unit impulse input for different values of K. The parameters of interest in the case of a unit impulse input are the initial displacement and the settling time for the system to return to equilibrium.

The following is the MATLAB script for getting a unit step input response for the system represented by equation (5).

num= [.05 .25 .3];
den= [1.05 1.25 .3];
step (num, den)

Figure 8 is the system response for a unit step input for system of equation (5) generated by MATLAB.

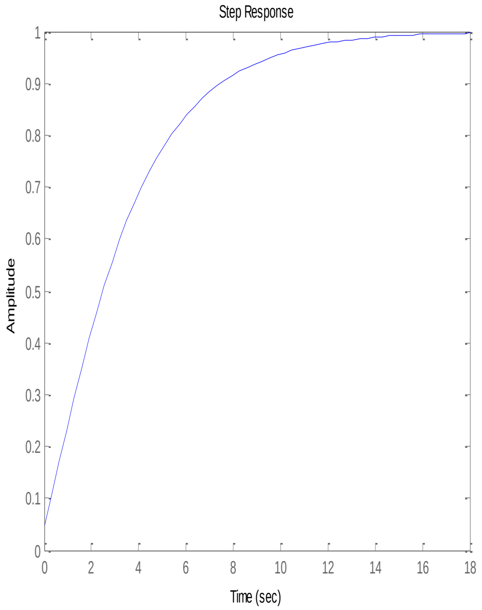

Figure 8: Response of system of equation (4) for a K value of .05

The following is the MATLAB script for getting a unit step input response for the system represented by equation (6).

num= [7 35 42];
den= [8 36 42];
step (num, den)

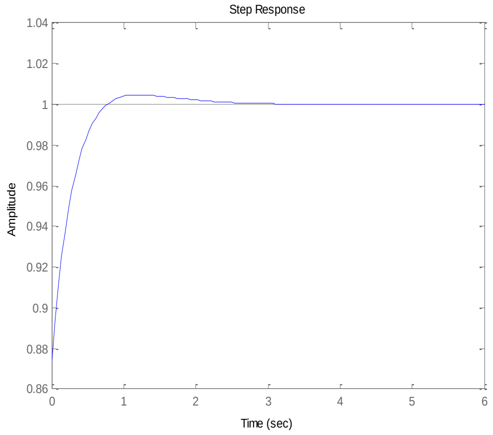

Figure 9: Response of system of equation (4) for a K value of .7

The following is the MATLAB script for getting a unit step input response for the system represented by equation (7).

```
num= [15 75 90];
den= [16 76 90];
step (num, den)
```

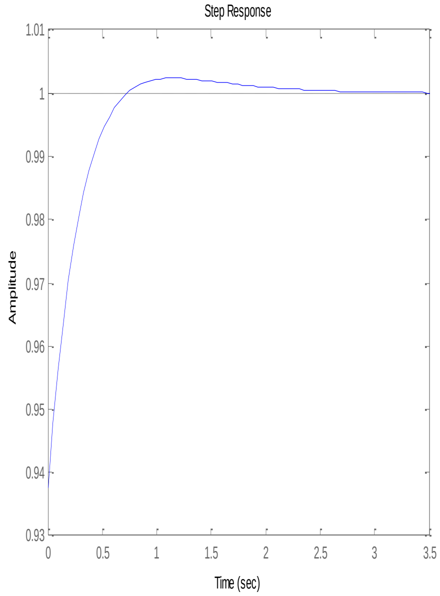

Figure 10: Response of system of equation (4) for a K value of 15

A comparison of figures 8, 9 & 10 shows that the system responds differently to a unit step input for different values of K. The parameters of interest in the case of a unit step input are the initial displacement and the settling time for the system to reach the value of unity which is guided by the unit step input.

In order to find the response to a sinusoidal input for the three selected K values the transfer functions for equations (5), (6) & (7) for a sinusoidal input must be calculated.

Assume a sinusoidal input of "2 sin (3t)" where the frequency of vibration is in units of rad/sec.

The Laplace transform for the sinusoidal input is "6/ S^2 + 9". [3]

Transfer functions for equations (5), (6) & (7) for a sinusoidal input of "2 sin (3t)" are as shown in equations (8), (9) & (10).

$$\frac{0.286\,S^2 + 1.428\,S + 1.714}{S^4 + 1.19\,S^3 + 9.23\,S^2 + 10.71\,S + 2.57} \qquad (8)$$

$$\frac{5.25\,S^2 + 26.25\,S + 31.5}{S^4 + 4.5\,S^3 + 14.24\,S^2 + 40.5\,S + 46.875} \qquad (9)$$

$$\frac{5.625\,S^2 + 28.125\,S + 33.75}{S^4 + 4.75\,S^3 + 14.625\,S^2 + 42.75\,S + 50.625} \qquad (10)$$

To get the system response for transfer functions shown in equations (8), (9) & (10) their inverse Laplace transform must be calculated. MATLAB cannot be effectively used to find the inverse Laplace transforms because the numerical techniques used by it do not present the solutions in a usable format. Equation (11), (12) & (13) are the inverse Laplace transforms calculated for equations (8), (9) & (10) by manual calculation techniques. [5]

$$f(t) = 0.26\ e^{-0.34t} - 0.14\ e^{-0.86t} + 0.083 \sin (3t) - 0.12 \cos (3t) \qquad (11)$$

$$f(t) = e^{-2.25t} (-0.38 \sin 0.43t + 0.15 \cos 0.43t) + 1.9 \sin 3t\ - 0.15 \cos 3t \qquad (12)$$

$$f(t) = 0.37\ e^{-2.5t} - 0.3\ e^{-2.25t} + 1.95 \sin(3t) - 0.07 \cos(3t) \qquad (13)$$

Further simplification of equations (11), (12) & (13) result in equations (14), (15) & (16).

$$f(t) = 0.26\ e^{-0.34t} - 0.14\ e^{-0.86t} + 0.15 \sin(3t - 0.6) \qquad (14)$$

$$f(t) = 0.17\ e^{-2.25t} \sin(0.43t - 1.19) + 1.9 \sin(3t - 1.49) \qquad (15)$$

$$f(t) = 0.37\ e^{-2.5t} - 0.3\ e^{-2.25t} + 1.95 \sin(3t - 1.53) \qquad (16)$$

Reviews of equations (14), (15) & (16) show that equation (15) indicates a different type of system characteristic compared to equations (14) and (16). Equation (15) shows that for values of K between 0.0718 & 14 the system behaves differently compared to outside of this range. Figures 11 through 13 illustrate this point graphically. These results show that the root-locus technique is able to identify the segments of the system that behave differently from a mathematical point of view. However, a comparison of figures 12 and 13 also demonstrate that from a practical point of view there may not be a difference between systems that mathematically appear different.

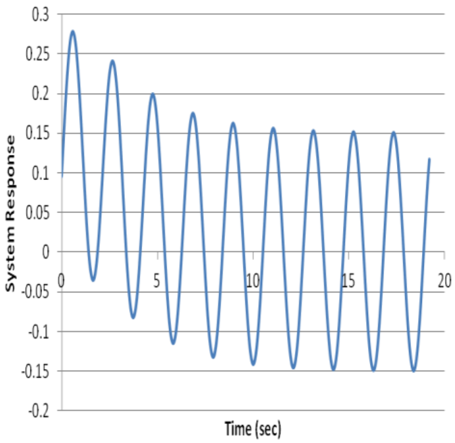

Figure 11: Response of system of equation (5) to a sinusoidal input. This plot represents equation (14)

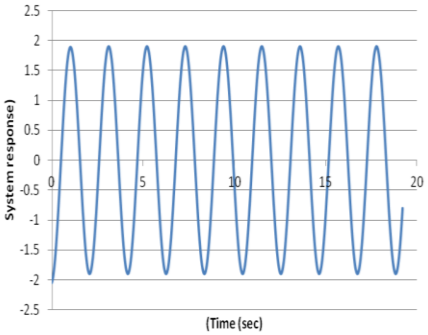

Figure 12: Response of system of equation (6) to a sinusoidal input. This plot represents equation (15)

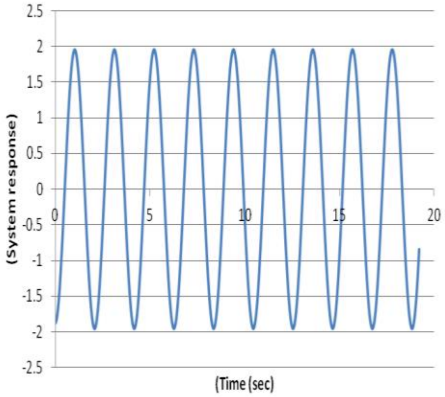

Figure 13: Response of system of equation (7) to a sinusoidal input. This plot represents equation (16)

Summary & Conclusion

In this article MATLAB was used to generate a Root-Locus plot for a closed loop control system that was represented by its block diagram in Laplace domain. The primary purpose of a Root-Locus plot is stability analysis of a control system. All the portions of a Root-Locus plot that are to the left of the imaginary axis represent the stable portion of the system and the plot portions to the right of the imaginary axis represent the unstable portion of the system. For the example presented in this article the system was completely stable.

In addition to stability analysis, a Root-Locus plot can be used to determine the critical areas where the control system behavior changes significantly. These areas can be determined by calculating the parameters where the plot crosses the real axis. In the example presented in this article, the plot crossed the real axis at two locations, and the constant values where the plot crossed the real axis were shown to be the boundaries where the system behavior changed mathematically and physically.

Once the parameters for a stable control system are determined, the control system must be analyzed for standard inputs. Among standard inputs are impulse, step function and sinusoidal inputs.

In this article MATLAB was used to determine the control system response for an impulse input and a step input. Additionally, it was shown that the MATLAB output is not usable for a sinusoidal input. Classical techniques were used to determine the output for a sinusoidal input. Comparisons of the results obtained by classical techniques for the regions identified by the Root-Locus plot were made. The results obtained by the classical techniques confirmed that the Root-Locus plot does indeed identify boundaries where the system performance changes.

References:

1. Modern Control Engineering; Fifth Edition; Katsuhiko Ogata; problem A-6-1.
2. Introduction to Control Systems ; D.K. Anand; page 203.
3. Transform Circuit Analysis for Engineering and Technology; William D. Stanley; Appendix B.
4. Modern Control Systems; Third Edition; Richard C. Dorf; Robert H. Bishop; page 87.
5. Technical Calculus. Special Edition ISBN 0-536-98727-0.