Invited Paper

Computer program for root locus plots
G.J. Lastman, a N.K. Sinha b
a Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1
b Department of Electrical & Computer Engineering, McMaster University, Hamilton, Ontario, Canada L8S 4L7

ABSTRACT

A computer program has been developed for plotting the root locus of the characteristic polynomial of a closed-loop system, either in the s-plane or the z-plane. The program can be run on an IBM-compatible personal computer with EGA or VGA colour graphics. The roots of the characteristic polynomial are calculated for a sequence of values of a parameter, each root is assigned to the appropriate branch of the root locus, and the root locus is plotted. The plot can be examined in detail by a movable cursor. The program can be run in either a test mode or a production mode. The test mode automatically executes a set of 20 instructive example problems either in the s-plane or the z-plane. The root loci of user-defined transfer functions can be obtained in the production mode. Six empirical constants, which can be changed by the user, are used for the numerical computation of the root locus.

INTRODUCTION

The root locus is an important tool in the study of the stability and design of a linear system as a parameter varies from zero to infinity. Manually obtaining a root locus can be a time-consuming computation. The task is made much easier if the computations and graph plotting are done by a digital computer. This paper describes a computer program that has been developed for plotting the root locus of the characteristic polynomial of a closed-loop system, either in the s-plane or the z-plane. The program can be run on an IBM-compatible personal computer with EGA or VGA colour graphics.
Consider a unity-feedback system described by the open-loop transfer function

\[ G(x) = \frac{P(x)}{Q(x)} \]

where \( P(x) \) and \( Q(x) \) are polynomials with real coefficients, and \( x \) can be either \( s \) or \( z \). The root locus is the graph of the roots of the characteristic polynomial

\[ KP(x) + Q(x) \]

as the parameter \( K \) varies from zero to infinity. Because the roots are functions of \( K \) we can denote them by \( x(K) \). To obtain \( x(K) \), a monotonically increasing sequence of \( K \)-values is computed in the interval \( 0 \leq K \leq K_{\text{max}} \), where \( K_{\text{max}} \) depends on \( P(x) \) and \( Q(x) \). Additional finite values of \( K \), corresponding to points where \( \frac{dx}{dK} \rightarrow \infty \) are inserted into the sequence of \( K \)-values to ensure that any break-away/re-entry points appear on the root locus. For each \( K \) in the sequence the roots \( x(K) \) are calculated, each root is assigned to its appropriate branch of the root locus, each finite zero of \( G(x) \) is assigned to its branch, and the root locus is plotted. The root locus graphs can be examined in detail by a movable cursor along any branch of the locus.

Note that although it appears from equation (1) that \( K \) is simply the open-loop gain of the system, in fact it is possible to rearrange the characteristic polynomial so that \( K \) may represent either a pole or a zero of the open-loop transfer function (for example, see reference 1). This makes it possible to use the root locus for the design of compensators for a control system.

Because the points on the root locus are obtained by numerical computation and utilized for plotting, it is necessary to use select six empirical constants in the program. These are necessary in order to be able to get smooth curves from a finite number of computations of roots of the polynomial for different values of \( K \). The significance of these constants is described in detail in the sections on “Generation of the Root Locus”, “Graphs” and “Empirical Constants Used in the Program”. The user can change these constants. The default values of the empirical constants gave satisfactory results for all the built-in example problems (see the section on Test Mode, below) and many other problems. However, there may be transfer functions for which one or more empirical constants might need to be changed.

**RUNNING THE PROGRAM**

The program can be run in either of two modes, a test mode or a production mode.

(a) **Test Mode**

This mode is invoked by typing

\[ \text{Locus } m \ n \]

at the DOS prompt. The command-line parameters \( m \) and \( n \) are character strings that do not contain blanks. If \( n = z \) (or \( Z \)), then the \( z \)-plane is used. If \( n \) is omitted, or if \( n = s \) (or \( S \)), then the \( s \)-plane is used. When the character string \( m \) is not an integer, the program executes a built-in set of 20 example problems, in the order 1, 2, \ldots, 20. If \( m \) is the string form of the integer \( M \), then the example problem set is executed in the order \( L, L+1, \ldots, 20, 1, 2, \ldots, L-1 \), where \( L = \text{Max}(1, \text{Min}(20, M)) \). The program terminates when the last problem is completed.
(b) Production Mode

This mode is obtained by typing

\[ \text{Locus} \]

at the DOS prompt. Production mode allows a user to obtain root locus plots for different transfer functions. Fig. 1 shows the introduction screen in this mode.

GENERATION OF THE ROOT LOCUS

In the program we assume that \( \deg[Q(x)] \geq \deg[P(x)] \). If \( \deg[Q(x)] < \deg[P(x)] \), we interchange \( P(x) \) and \( Q(x) \), and replace \( K \) by \( 1/K \). The user is unaware of this interchange because the original \( P(x) \), \( Q(x) \), and \( K \) are displayed on the screen. Example 13 in the Test Mode is of this type.

The program uses the following steps to generate the root locus.

(a) **Find all poles and all finite zeros of** \( G(x) \).

Roots of polynomials are calculated by Laguerre’s method.

(b) **Find all discontinuity K-points**

A possible discontinuity \( K \)-point, \( K_d \), is found by setting

\[ K_d P'(x) + Q'(x) = 0 \]

where the prime denotes differentiation with respect to \( x \). Because \( K_d P(x) + Q(x) = 0 \), \( x = x_d \) is a root of

\[ P(x)Q'(x) - Q(x)P'(x) \]

Some of these roots may not be on the root locus. For those that are on the root locus we have

\[ K_d = \frac{|Q(x_d)|}{|P(x_d)|} \quad (3) \]

If we obtain \( K_d \) for each \( x_d \), there are at most

\[ \deg[P(x)Q'(x) + Q(x)P'(x)] \leq \deg[Q(x)] + \deg[P(x)] - 1 \]

special \( K \)-values at which we should compute the root locus. These \( K \)-values include all possible break-away/re-entry points.

(c) **Find** \( K_{\text{max}} \)

Although the root locus exists for each non-negative \( K \), we need an upper bound on \( K \), \( K_{\text{max}} \), to terminate the computations. \( K_{\text{max}} \) is determined as follows.

(i) If \( G(x) \) has finite zeros then \( K_{\text{max}} \) is determined by examining the branches of the root locus that terminate at the finite zeros. \( K_{\text{max}} \) is a value of \( K \) for which each of these finite-zero-terminating branches is sufficiently close to its zero. This is done by calculating the root locus at \( K_{\text{max}} = \text{LowerLimit} \) (an empirical constant). On the other hand, if \( K_{\text{max}} = \text{UpperLimit} \) (an empirical constant), then we accept \( K_{\text{max}} \); otherwise, if, for each finite-zero-terminating branch of the root locus, the distance from the root locus at \( K_{\text{max}} \) to the zero is less than or equal to

\[ \text{ClosenessFactor} \times \text{Max(magnitude of the finite zero, 1)} \]
then we accept $K_{\text{max}}$; otherwise, set

$$K_{\text{max}} = \min (\text{ExpansionFactor} \times K_{\text{max}}, \text{UpperLimit})$$

and recalculate. ($\text{ClosenessFactor}$ and $\text{ExpansionFactor}$ are empirical constants.)

(ii) If $G(x)$ only has zeros at infinity, then $K_{\text{max}}$ is a value of $K$ at which each branch of the root locus has left a circle, of given radius, centred at each pole. The root locus is first calculated at $K_{\text{max}} = \text{LowerLimit}$. If $K_{\text{max}} = \text{UpperLimit}$ or if, for each branch of the root locus the distance from the root locus to each pole, $x_p$ of $G(x)$ is greater than or equal to $r$, where

$$r = \begin{cases} 2h, & h > 0 \\ 5, & h = 0 \end{cases}$$

$$h = \max \{|x_p|\}$$

we accept $K_{\text{max}}$; otherwise, set

$$K_{\text{max}} = \min (\text{ExpansionFactor} \times K_{\text{max}}, \text{UpperLimit})$$

and recalculate.

(d) Generation of the sequence of $K$-values

A sequence of $K$-values is generated as

$$K_i = A \times K_{i+1}, \quad 1 \leq i \leq N-1,$$

where $A = (K_{\text{max}}/K_{\text{initial}})^{1/(N-1)}$, $N$ is the number of data points (currently 350), $K_0 = K_{\text{initial}}$ and $K_{\text{initial}}$ is an empirical constant. This method to generate the $K_i$ values has proven to be very effective.

The final set of $K$-values is obtained by adjoining $K = 0$ and any discontinuity $K$-points, $K_x$ to the sequence $\{K_i\}$. This set of $K$-values is then reordered from least magnitude to greatest magnitude.

(e) Construction of the branches of the root locus.

Each branch starts at a pole of $G(x)$, corresponding to $K = 0$. The roots of the characteristic polynomial are calculated for each $K$ in the set of $K$-values. For each root calculated at $K$ we determine the branch that is closest to it, and then adjoin it to that branch. The finite zeros of $G(x)$, corresponding to $K = \infty$, are then adjoined to their appropriate branches.

GRAPHS

The branches of the root locus are displayed graphically. Although the graphs in both the $s$-plane and the $z$-plane are similar, the unit circle, $|z| = 1$, also is shown in the $z$-plane. Furthermore, the calculation of the damping ratio along the locus is different in the $z$-plane.

After the root locus is plotted, the user can invoke a movable cursor that can be used to travel up or down a branch. The cursor moves from a stored data point to its successor or predecessor data point on a branch when the "up arrow" or "down arrow" keys are pressed. The "page up" and "page down" keys are used to jump the cursor five data points up or down the branch. In addition, the cursor can be moved to other branches by use of the "+" or "-" keys. The value of $K$, the damping ratio,
and the coordinates of the cursor position are displayed when the cursor is activated. There are some problems that could arise in plotting the graph of the root locus. The first of these problems concerns the closeness of the spacing of the $K_i$ for small values of $i$, where several consecutive root locus points could be mapped onto the same screen pixel; the computed root locus is scanned to eliminate these extraneous consecutive points. The second problem could occur for large values of $i$ if consecutive root locus points map onto different screen pixels that are relatively far apart, making the graph of the root locus look "kinky"; if such root locus points are more than PixelSeparation (an empirical constant) pixels apart, then extra root locus points are computed to give a smoother graph. Thirdly, because of the finite value of $K_{max}$ and the finite extent of the graph boundaries, a branch of the root locus that tends to infinity may not be shown as extending to the boundary of the graph plotting area. Finally, because of the discrete nature of determining the $K$-values the root locus may be distorted in regions where $|dx/dK|$ is very large; changing the empirical constants may help.

EMPIRICAL CONSTANTS USED IN THE PROGRAM

Since the root locus plot is based on calculating the roots of the characteristic polynomial for a finite number of values of $K$, it is necessary to use a set of empirical constants. These allow the program to determine the initial and final values of $K$, as well as the separation between different values of $K$ so that a smooth curve is obtained for each case.

The six empirical constants used in the program are LowerLimit, UpperLimit, ExpansionFactor, ClosenessFactor, $K_{init}$ and PixelSeparation. The usage is described in the sections on "Generation of the Root Locus (c) and (d)", and "Graphs". Although the default values (see Fig. 8) of these constants give satisfactory results for a large set of example problems, the user can easily change the values. It may be necessary to do so for some transfer functions. For example, graph smoothness might be improved by setting PixelSeparation $= 2$, its minimum value.

GENERAL INFORMATION ABOUT THE PROGRAM

The main menu in the production mode (Fig. 2) allows the user to change either the numerator or the denominator polynomial, plot the root locus, change the empirical constants, or exit the program.

(a) Changing the numerator or denominator polynomial

To change either the numerator or denominator polynomial, the user presses the appropriate key and a window appears on the screen to allow input of a new polynomial. The user is provided with prompts at each step in the process. Fig. 3 shows a sample input for the numerator polynomial. When all the factors have been entered a new window appears showing each of the entered factors (Fig. 4). At this point the user can correct any errors. When the user decides that all factors are correct, the main menu (Fig. 2) reappears.

(b) Plotting the root locus

Selection of this option produces a plot of the root locus of the stored transfer
function. Fig. 5 shows a root locus plot in the s-plane, while Fig. 6 shows the same root locus with the movable cursor activated. Fig. 7 shows a root locus plot in the z-plane.

(c) Changing the empirical constants

When this option is selected the screen changes to that shown in Fig. 8. An on-line explanation can be accessed for each empirical constant.

RESTRICTIONS ON THE POLYNOMIALS \( P(x) \) AND \( Q(x) \)

The following restrictions on the polynomials \( P(x) \) and \( Q(x) \) are checked in the computer program.

(a) Each polynomial can be entered in the factored form, where the degree of each factor is at most 10. In addition, we must have \( 0 \leq \text{deg}[P(x)] \leq 10, \) \( 0 \leq \text{deg}[Q(x)] \leq 10, \) and \( \text{Max}(\text{deg}[P(x)],\text{deg}[Q(x)]) \geq 1. \)

(b) If \( \text{deg}[P(x)] = \text{deg}[Q(x)] \), then the leading coefficients of \( P(x) \) and \( Q(x) \) must have the same sign. This is to avoid having \( |x| \to \infty \) for a finite positive value of \( K. \)

(c) \( P(x) \) and \( Q(x) \) must be coprime; that is, they have no roots in common.

CONCLUSIONS

Although the root locus method was proposed almost forty-five years ago, and rules were developed for sketching the root locus approximately, there have always been practical problems in obtaining accurate plots, except in very simple cases. Hence, although the method is quite attractive for analysis and design of control systems due to the nice graphical presentation, it has not been used extensively. The object of the present program is to remove the drudgery involved in obtaining accurate plots of root loci and thereby make them more useful for application in design. Root loci in both the s-plane and the z-plane can be obtained with this program. Although the rules for plotting the loci are the same in both cases, the display of the damping ratio in the z-plane along the locus requires some additional calculations, which are included in the program so that the user is not burdened with these. We hope that the program is easy to understand and use, and that it is user-friendly.

We believe that the computer program described above will enable engineering students as well as practising engineers to use the root locus more effectively as a tool for analysis and design of control systems.

REFERENCES

Root Locus Program

This program finds the root locus for a characteristic polynomial. The characteristic equation \( K P(x) + Q(x) = 0 \) defines the root locus, and we write

\[ G(x) = \frac{K P(x)}{Q(x)} \]

where \( 0 \leq \deg(P(x)) \leq 10 \) and \( 0 \leq \deg(Q(x)) \leq 10 \). Each of the polynomials \( P(x) \) and \( Q(x) \) can be entered in factored form, where each factor is of degree at most 10 and the number of factors is at most 10. The user should ensure that \( P(x) \) and \( Q(x) \) are relatively prime; that is, they have no roots in common.

The variable \( x \) can be \( s \) or \( z \), so that either an \( s \)-plane or a \( z \)-plane plot can be examined.

The root locus is plotted with the poles of \( G(x) \) marked by \( X \) and the zeros by \( O \). The user can move a cursor along any branch of the root locus graph or switch branches.

Use the \( s \)-plane? (Y or N) Y
Do you want to do the built-in example problem? (Y or N)

Figure 1. The introduction message
The user has answered "Yes" to the first question, but has not yet answered the second question.

Root Locus Plotter
\( s \)-plane

Figure 2. The main menu
Root Locus Plotter

The new numerator is to have two factors. The first factor has degree one. The coefficient $a[1]$ has been entered; $a[0]$ has just been typed.

Figure 3.

The factors of the new numerator are displayed to allow the user to correct any error.

Figure 4.
The Root Locus

\[ G(s) = \frac{s^2 + 1.5s + 1.5}{(s - 0.75)(s + 0.25)(s + 1.25)(s + 2.25)} \]

Figure 5. A root locus plot in the s-plane

Figure 6. A root locus plot in the s-plane, with the movable cursor activated.
Figure 7. A root locus plot in the z-plane, with the movable cursor activated.

Change Empirical Constants

<table>
<thead>
<tr>
<th>Empirical Constant</th>
<th>Default Value</th>
<th>Current Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LowerLimit</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>UpperLimit</td>
<td>1.000E+0010</td>
<td></td>
</tr>
<tr>
<td>ExpansionFactor</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>ClosenessFactor</td>
<td>0.675</td>
<td>0.675</td>
</tr>
<tr>
<td>Kinitial</td>
<td>1.000E-0008</td>
<td>1.000E-0008</td>
</tr>
<tr>
<td>PixelSeparation</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

1. To move the highlight bar, press <↑> or <↓>.
2. To change a highlighted value, type in the new value and press <Enter>.
3. To reset a highlighted value to the default value, just press <Enter>.
4. To accept all current values, press <Esc>.
5. To obtain an explanation of a highlighted empirical constant, press <H>.

Figure 8. Changing the empirical constants