Nigerian Journal of Technology (NIJOTECH)
Vol. 33. No. 1. January 2014, pp.**60 – 71**

University of Nigeria, Nsukka, ISSN: 1115-8443
www.nijotech.com
http://dx.doi.org/10.4314/njt.v33i1.9

COMPUTER-AIDED ROOT-LOCUS NUMERICAL TECHNIQUE

A. R. Zubair[1,*], A. Olatunbosun[2]
[1, 2] DEPARTMENT OF ELECTRICAL/ELECTRONIC ENGINEERING, UNIVERSITY OF IBADAN, NIGERIA

E-mail addresses: [1]*ar.zubair@mail.ui.edu.ng*, [2]*aolatunbosun@yahoo.com*

Abstract

The existing technique of manual calculation and plotting of root-locus of a control system for stability studies is laborious and time consuming. A computer-aided root-locus numerical technique based on simple fast mathematical iteration is developed to eliminate this rigour. Alternative iteration formulas were formed and tested for convergence. A pair of complementary formulas with highest rate of convergence was selected. The process of automatic determination of roots and plotting of root-locus were divided into steps which were analyzed, simplified and coded into computer programs. The geometric properties of the root-loci obtained with this technique are found to conform to those described in the literature. Root-loci are drawn to scale instead of rough sketches.

Keywords: control system, stability, transient response, root-locus, iteration

1. Introduction

A control system is the means by which any parameter of interest in a machine, mechanism or equipment is maintained or altered in accordance with desired manner. Introduction of feedback into a control system has the advantages of reducing sensitivity of system performance to internal variations in system parameters, improving transient response and minimizing the effects of disturbance signals. However, feedback increases the number of components, increases complexity, reduces gain and introduces the possibility of instability [1 – 6].

A system is stable if its response to a bounded input vanishes as time t approaches ∞. An unstable system cannot perform any control task. A stable system with low damping is also not desirable. Therefore, a stable system must also meet the specifications on relative stability which is a quantitative measure of how fast the transients die out with time in the system. The location of roots of system's characteristic equation determines the stability of the system [1 -5, 7, 8]. The roots of the system's characteristic equation are the same as the poles of the closed-loop system.

Ideally, a desired performance can be achieved by a control system by adjusting the location of its roots in the s-plane by varying one or more system parameters. Root-locus Method is a linear time invariant control system design technique that determines the roots of the characteristic equation (or closed-loop poles) when the open-loop gain-constant K is increased from zero to infinity [9, 10]. The root-locus shows the complete dynamic response of the system and is a measure of the sensitivity of the roots to variation in the parameter (open-loop gain-constant K) [11]. It yields frequency response of the system and can also be used to solve problems in the time domain. The basic task in root-locus technique is to determine the closed-loop pole configuration from the configuration of the open-loop poles and zeros.

The locus of the roots or closed-loop poles are plotted in the s-plane. This is a complex plane, since $s = \sigma + j\omega$. The real part, σ, is the index in the exponential term of the time response, and if positive will make the system unstable. Any locus in the right-hand side of the plane therefore represents an unstable system. The imaginary part, ω, is the frequency of transient oscillation. When a locus crosses the imaginary axis, $\sigma = 0$, the control system is on the verge of instability; where transient oscillations neither increase, nor decay, but remain at a constant value. The design method requires the closed-loop poles to be plotted in the s-plane as K is varied from zero to infinity, and then a value of K selected to provide the necessary transient response as required by

* Corresponding author, Tel: +234-8023278605

the performance specification. The loci always commence at open-loop poles (denoted by x) and terminate at open-loop zeros (denoted by o) when they exist [1 – 5, 9, 10, 12].

In the literature, there exist eight rules based on elementary geometric properties of root-locus which lead to rough sketch of root locus. These rules govern the important features of the root-locus method such as asymptotes, roots condition on the real axis, breakaway points, and imaginary axis crossover [13 – 16]. These rules or steps involve some calculations but avoid determination of actual roots. There are other approaches which use complex analytic or semi-analytic representation that involve the use of equations of the loci [14, 17 – 25]. There is also a computer aided root-locus method based on complex conversion of solution of a parameterized family of algebraic problems into solution of a set of associated differential equations [12]. In this paper, a computer-aided root-locus numerical technique is developed based on simple and fast mathematical iteration.

2.0 Root-Locus Numerical Technique

Figure 1 shows a closed-loop system with negative feedback; G(s) is the forward-part gain. H(s) is the feedback gain. The transfer function of the system is given by Eqn (1) [1 - 5]. The denominator of the transfer function set to zero is referred to as the characteristic equation: $1+G(s)H(s)=0$.

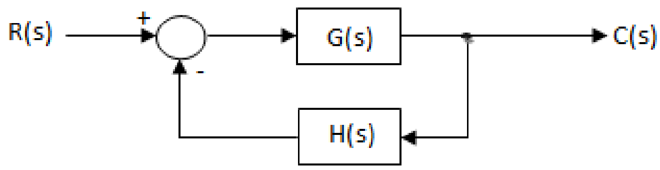


Figure 1: Closed-Loop System with Negative Feedback

$$\frac{C(s)}{R(s)}=\frac{G(s)}{1+G(s)H(s)} \qquad (1)$$

The open-loop transfer function is given as in Eqn. (2).

$$G(s)H(s)=K\frac{B}{A}$$
$$=K\frac{[s-z(1)][s-z(2)]...[s-z(m-1)][s-z(m)]}{[s-p(1)][s-p(2)]...[s-p(n-1)][s-p(n)]} \qquad (2)$$

z(1), z(2), z(3), ..., z(m-1) and z(m) are m open-loop zeros and p(1), p(2), p(3), ..., p(n-1) and p(n) are n open-loop poles. Given the open-loop poles and zeros, the task is to deduce the configuration of the closed-loop poles as K varies from zero to infinity. The characteristic equation becomes:

$$1+K\frac{B}{A}=0;\ \mathrm{A}+\mathrm{KB}=0 \qquad (3)$$

The root-locus technique presented in this paper is divided into four steps.

2.1 Step 1: Expansion for the Formation of Characteristic Equation

The first step is the formation of polynomials A and B from the given open-loop poles and zeros respectively. This step involves expansion. A and B are given as

$$A=[s-p(1)][s-p(2)][s-p(3)]...$$
$$[s-p(n-1)][s-p(n)]=\mathrm{a(n+1)s}^{n}$$
$$+a(n)s^{n-1}+...+a(3)s^{2}+a(2)s+a(1)$$

and

$$B=[s-z(1)][s-z(2)][s-z(3)]...$$
$$[s-z(m-1)][s-z(m)]$$
$$=\mathrm{b(m+1)s}^{m}+b(m)s^{m-1}$$
$$+...+b(3)s^{2}+b(2)s+b(1)$$

Suppose an N^{th} order polynomial $C(s)=c(N+1)s^{N}+c(N)s^{N-1}+...+c(2)s+c(1)$ is multiplied with $(s-r)$ where r is a real open-loop pole or zero. The product is a $(N+1)^{th}$ order polynomial $D(s)=d(N+2)s^{N+1}+d(N+1)s^{N}+d(N)s^{N-1}+...+d(2)s+d(1)$ such that

$$\mathrm{d(y)}=\begin{cases}c(y-1) & \text{for y = (N + 2)}\\ c(y-1)-rc(y) & \text{for y = 2 to (N + 1)}\\ -rc(y) & \text{for y = 1}\end{cases} \qquad (4)$$

Suppose an N^{th} order polynomial $C(s)=c(N+1)s^{N}+c(N)s^{N-1}+...+c(2)s+c(1)$ is multiplied with $[s-(r+jh)][s-(r-jh)]$ where (r+jh) and (r-jh) are complex conjugate open-loop poles or zeros. The product is a $(N+2)^{th}$ order polynomial

$$D(s)=d(N+3)s^{N+2}+d(N+2)s^{N+1}$$
$$+d(N+1)s^{N}+...+d(2)s+d(1)$$

such that:

$$d(y)=\begin{cases}\mathrm{t(3)}c(y-2) & \text{for y = (N + 3)}\\ \mathrm{t(3)}c(y-2)+t(2)c(y-1) & \text{for y = (N + 2)}\\ \mathrm{t(3)}c(y-2)+t(2)c(y-1)+t(1)c(y) & \text{for y = 3 to (N + 1)}\\ t(2)c(y-1)+t(1)c(y) & \text{for y = 2}\\ t(1)c(y) & \text{for y = 1}\end{cases} \qquad (5)$$

where

$$[s-(r+jh)][s-(r-jh)]=t(3)s^{2}+t(2)s+t(1)$$
$$t(3)=1;\ \ t(2)=-2r;\ \ t(1)=r^{2}+h^{2} \qquad (6)$$

Eqns. (4) and (5) give the general models for expansion involving a single real pole or zero and a pair of conjugate complex pole or zero respectively. As illustrated with the flow chart of Figure 2, the expansion of A starts with $C(s)=1$ (order zero) multiplied with $s-r$ for the first real pole. The product D obtained is feedback to the multiplication process as C and the process is repeated for other real poles. The new product D is feedback as C which is multiplied with $[s-(r+jh)][s-(r-jh)]=t(3)s^2+t(2)s+t(1)$ for the first pair of conjugate poles. The process is repeated for other pairs of conjugate poles. A is set equal to the final product D. The whole process is repeated for B using zeros instead of poles. With the substitution of A, B and a value for K, the characteristic Eqn. (3) is given as in Eqn. (7). The characteristic equation is said to be of order N and has N roots.

$$E(s)=e(\mathrm{N}+1)\mathrm{s}^N+e(N)s^{N-1}+\ldots+e(3)s^2+e(2)s+e(1)=0 \tag{7}$$

2.2 Step 2: Determination of Roots for a Value of K

Having obtained the characteristic equation, the next step is to determine the roots for a value of K. This step is classified into three sub-steps.

2.2.1 Sub-Step 2.1: Iteration

For N=1, the root is:

$$r=-\frac{e(1)}{e(2)} \tag{8}$$

For N=2, one of the roots is

$$r=\frac{-e(2)-\sqrt{e(2)^2-4e(1)e(3)}}{2e(3)} \tag{9}$$

For N>2, a root of the characteristic Eqn. (7) is obtainable by iterative technique. Iteration computes a sequence of progressively accurate iterates to approximate the solution of an equation [26,27]. Iteration formula is obtainable from Eqn. (7) by making s the subject of formula. Retaining the term e(2)s in Eqn. (7) on the left hand side (LHS), moving all other terms to the right hand side (RHS) and dividing both sides by e(2) lead to the iteration formula of Eqn. (10).

$$s=\begin{bmatrix}-e(1)-e(3)s^2-e(4)s^3-\ldots\\ -e(N)s^{N-1}-e(N+1)s^N\end{bmatrix}/\,e(2) \tag{10}$$

Retaining the term $e(N+1)s^N$ in Eqn. (7) on LHS, moving all other terms to RHS, dividing both sides by e(N+1) and evaluating $\left(\frac{1}{N}\right)^{\text{th}}$ root of both sides lead to the even version of the iterative formula of Eqn. (11). This is used if N is even. Retaining the term $e(N+1)s^N$ in Eqn. (7) on LHS, moving all other terms to RHS, dividing both sides by e(N+1)s and evaluating $\left(\frac{1}{N-1}\right)^{\text{th}}$ root of both sides lead to the odd version of the iterative formula of Eqn. (11). This is used if N is odd (N-1 is even). Eqns. (11) and (12) are the same except that the result of $\left(\frac{1}{N}\right)^{\text{th}}$ root and $\left(\frac{1}{N-1}\right)^{\text{th}}$ root are taken to be positive in Eqn. (11) but negative in Eqn. (12). For example, the result of $16^{1/4}$ is either +2 or -2.

$$s=\begin{cases}+\left[\left(-e(1)-e(2)s-e(3)s^2-\ldots-e(N)s^{N-1}\right)/\,e(N+1)\right]^{\frac{1}{\mathrm{N}}} & \text{for even N}\\ +\left[\left(-e(1)s^{-1}-e(2)-e(3)s-\ldots-e(N)s^{N-2}\right)/\,e(N+1)\right]^{\frac{1}{\mathrm{N-1}}} & \text{for odd N}\end{cases} \tag{11}$$

$$s=\begin{cases}-\left[\left(-e(1)-e(2)s-e(3)s^2-\ldots-e(N)s^{N-1}\right)/\,e(N+1)\right]^{\frac{1}{\mathrm{N}}} & \text{for even N}\\ -\left[\left(-e(1)s^{-1}-e(2)-e(3)s-\ldots-e(N)s^{N-2}\right)/\,e(N+1)\right]^{\frac{1}{\mathrm{N-1}}} & \text{for odd N}\end{cases} \tag{12}$$

Eqns. (13) to (16) are obtainable from Eqn. (11) by breaking down the $\left(\frac{1}{N}\right)^{\text{th}}$ root into two $\left(\frac{1}{\sqrt{N}}\right)^{\text{th}}$ roots or breaking down the $\left(\frac{1}{N-1}\right)^{\text{th}}$ root into two $\left(\frac{1}{\sqrt{N-1}}\right)^{\text{th}}$ roots on the RHS. For example $+16^{1/4}$ is the same as $[(16)^{\frac{1}{2}}]^{\frac{1}{2}}$. Eqns. (13), (14), (15) and (16) are the same except that the two $\left(\frac{1}{\sqrt{N}}\right)^{th}$ roots or the two $\left(\frac{1}{\sqrt{N-1}}\right)^{th}$ roots are taken to be $(--)$, $(-+)$, $(++)$ and $(+-)$ respectively.

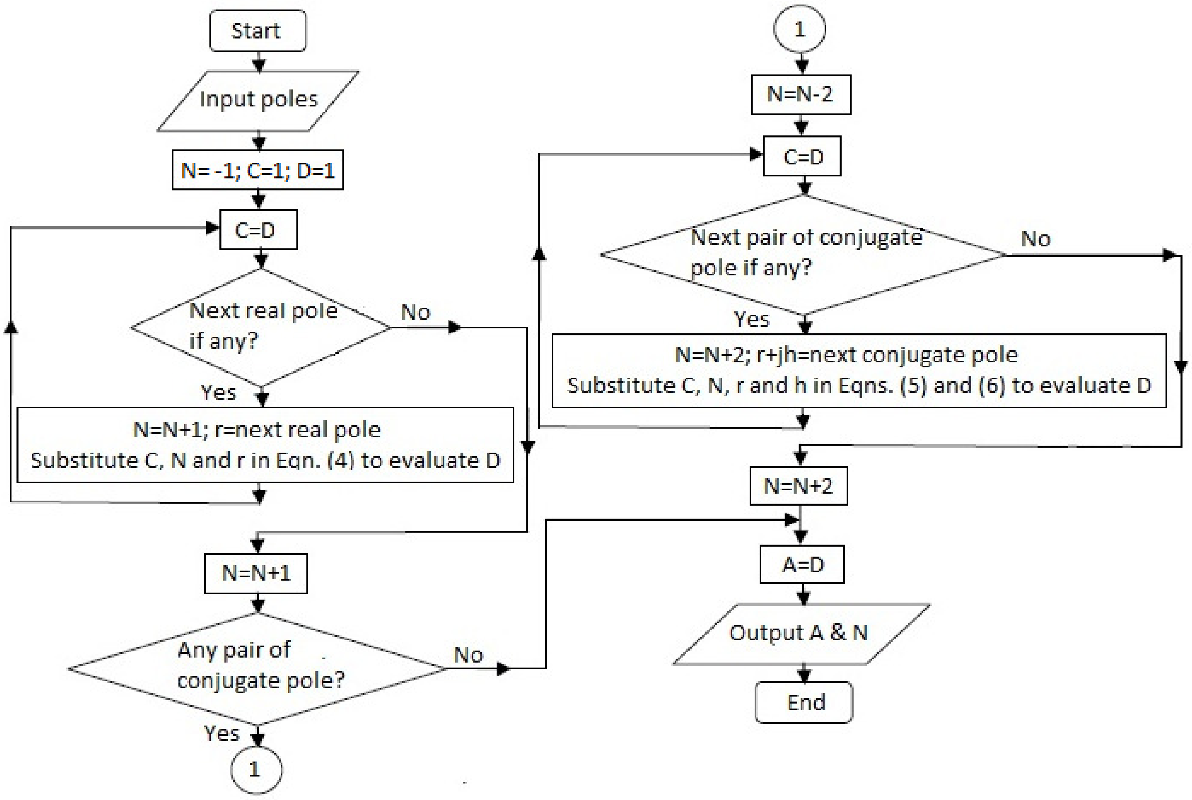


Figure 2: Flow Chart for Expansion Process

$$s = \begin{cases} -\left[-\left[\left(-e(1)-e(2)s-e(3)s^2-\ldots-e(N)s^{N-1}\right)/e(N+1)\right]^{\frac{1}{\sqrt{N}}}\right]^{\frac{1}{\sqrt{N}}} & \text{for even N} \\ -\left[-\left[\left(-e(1)s^{-1}-e(2)-e(3)s-\ldots-e(N)s^{N-2}\right)/e(N+1)\right]^{\frac{1}{\sqrt{N-1}}}\right]^{\frac{1}{\sqrt{N-1}}} & \text{for odd N} \end{cases} \quad (13)$$

$$s = \begin{cases} -\left[+\left[\left(-e(1)-e(2)s-e(3)s^2-\ldots-e(N)s^{N-1}\right)/e(N+1)\right]^{\frac{1}{\sqrt{N}}}\right]^{\frac{1}{\sqrt{N}}} & \text{for even N} \\ -\left[+\left[\left(-e(1)s^{-1}-e(2)-e(3)s-\ldots-e(N)s^{N-2}\right)/e(N+1)\right]^{\frac{1}{\sqrt{N-1}}}\right]^{\frac{1}{\sqrt{N-1}}} & \text{for odd N} \end{cases} \quad (14)$$

$$s = \begin{cases} +\left[+\left[\left(-e(1)-e(2)s-e(3)s^2-\ldots-e(N)s^{N-1}\right)/e(N+1)\right]^{\frac{1}{\sqrt{N}}}\right]^{\frac{1}{\sqrt{N}}} & \text{for even N} \\ +\left[+\left[\left(-e(1)s^{-1}-e(2)-e(3)s-\ldots-e(N)s^{N-2}\right)/e(N+1)\right]^{\frac{1}{\sqrt{N-1}}}\right]^{\frac{1}{\sqrt{N-1}}} & \text{for odd N} \end{cases} \quad (15)$$

$$s = \begin{cases} +\left[-\left[\left(-e(1)-e(2)s-e(3)s^2-\ldots-e(N)s^{N-1}\right)/e(N+1)\right]^{\frac{1}{\sqrt{N}}}\right]^{\frac{1}{\sqrt{N}}} & \text{for even N} \\ +\left[-\left[\left(-e(1)s^{-1}-e(2)-e(3)s-\ldots-e(N)s^{N-2}\right)/e(N+1)\right]^{\frac{1}{\sqrt{N-1}}}\right]^{\frac{1}{\sqrt{N-1}}} & \text{for odd N} \end{cases} \quad (16)$$

The alternative iteration formulas Eqns. (10) to (16) were subjected to tests to determine the most reliable or effective. The results and conclusion of the tests are presented and discussed in section 3.1. A guess value of s is substituted in the right side of the iteration formula; a new value of s is computed and substituted back in the right side of the iteration formula and the process is repeated T times. s will converge to a root r of the characteristic equation. The higher the T, the closer the result of iteration to the root and the more time the process takes. Therefore T should not be too small and should not be too large. In some cases, s may not converge to the root [26, 27]. There is need to validate the result obtained as a root.

2.2.2 Sub-Step 2.2: Root Validation Test

The final value of s ($s=r$) obtained by iteration is tested by substituting the value of s in the test equation, Eqn. (17) which is adapted from the characteristic equation Eqn. (7). rvt is computed. If $|rvt|$ is equal to 0 or less than 0.00004, s is accepted and recorded as a root; otherwise it is rejected.

$$rvt = e(\mathrm{N}+1)\mathrm{s}^{N} + e(N)s^{N-1} + \ldots + e(3)s^{2} + e(2)s + e(1) \quad (17)$$

2.2.3 Sub-Step 2.3: Grouping of Accepted Root into a Sub-branch of Root-Locus

Accepted root is plotted as a standalone dot (.) on a root-locus dots-plot. Root-locus for a system with n open-loop poles has n branches. A branch of root-locus starts at an open loop pole usually marked with 'x' and ends at an open loop zero usually marked with 'o' or at infinity. At breakaway points, each single branch breaks into sub-branches. For the purpose of joining the dots with a curve, each accepted root is checked and grouped into a sub-branch of the root-locus. Based on the root-locus dots-plot generated by the algorithm, a number of computer program conditional statements guide the placement of the accepted root into the right sub-branch. An example is presented section 3.3 and illustrated in figure 5.

2.2.4 Sub-Step 2.4: Division Process for Next Root

Having obtained a root, whether the root is acceptable or not in Sub-Step 2.2, there is need to divide Eqn. (7) to obtain a lower order characteristic equation for the next root. For a real root r, Eqn. (7) is divided by $(s-r)$ and the order is reduced by 1. A complex root $r=\sigma+j\omega$ predicts it's conjugate, $r=\sigma-j\omega$, and vice versa. Therefore, for a complex root r, Eqn. (7) is divided by $(s-\sigma+j\omega)(s-\sigma-j\omega)$ and the order is reduced by 2.

Suppose an Nth order polynomial $E(s)=e(N+1))s^{N}+e(N)s^{N-1}+e(N-1)s^{N-2}+\ldots+e(2)s+e(1)$ is divided by $s-r$ where r is a real root. The quotient a (N-1)th order polynomial $D(s)=d(N)s^{N-1}+d(N-1)s^{N-2}+d(N-2)s^{N-3}+\ldots+d(2)s+d(1)$ such that

$$d(y)=\begin{cases} e(y+1) & \text{for y = N} \\ e(y+1)+rd(y+1) & \text{for y = (N - 1) to 1} \end{cases} \quad (18)$$

Suppose a Nth order polynomial $E(s)=e(N+1))s^{N}+e(N)s^{N-1}+e(N-1)s^{N-2}+\ldots+e(2)s+e(1)$ is divided by a pair of conjugate roots, $(s-\sigma+j\omega)(s-\sigma-j\omega)$. The quotient is a (N-2)th order polynomial :

$$D(s)=d(N-1)s^{N-2}+d(N-2)s^{N-3}+d(N-3)s^{N-4}+\ldots+d(2)s+d(1)$$

such that

$$d(y)=\begin{cases} e(y+2) & \text{for y = N - 1} \\ e(y+2)-td(y+1) & \text{for y = N - 2} \\ e(y+2)-td(y+1)-pd(y+2) & \text{for y = (N - 3) to 1} \end{cases} \quad (19)$$

where

$(s-\sigma+j\omega)(s-\sigma-j\omega)=s^{2}+ts+p$;

$t=-2\sigma;\quad p=\sigma^{2}+\omega^{2}$ (20)

Eqns. (18) and (19) give the general models for the division process involving a single real root and a pair of conjugate complex root respectively.

2.2.5 Repetition of Sub-Steps 2.1 to 2.4 for Next Root

The output D of the division process (Sub-Step 2.4) is feedback as E (characteristic equation) to the iteration process (Sub-Step 2.1) which is repeated to obtain the next root. The next root is subjected to validation test (Sub-Step 2.2); if it is a valid root, it is plotted as a standalone dot (.) on root-locus dots-plot and it is grouped into a sub-branch (Sub-Step 2.3). Sub-Step 2.4 is also repeated to update D. The sub-steps 2.1, 2.2, 2.3 and 2.4 are repeated until all the N roots have been obtained and the polynomial D is reduced to $D(s)=1$.

2.3 Step 3: Plotting of Root-Locus

K is varied at some intervals from zero to infinity. The N roots corresponding to each selected value of K are determined as explained in Step 2. Figure 3 shows the flow chart for the determination and plotting of the root-locus.

2.4 Adding Details to the Root-Locus

Having plotted the root-locus, the asymptotes, breakaway points and imaginary axis crossover can be added. For large values of K, root-locus branches are parallel to lines called asymptotes. There are (n-m) asymptotes [1 - 5]. Point of intersection of asymptotes on the real axis is given as in Eqn. (21) [1 - 5].

$$P_{asx} = \frac{\sum(\text{real part of open-loop poles}) - \sum(\text{real part of open-loop zeros})}{(n-m)} \qquad (21)$$

The angles between asymptotes and the real axis are given as in Eqn. (22) [1 - 5].

$$\theta(q) = \frac{(2q+1)\pi}{(n-m)} \quad \text{where } q = 0, 1, 2, \ldots, (n-m-1) \qquad (22)$$

Breakaway points are obtained as the solution of $\frac{dK}{ds} = 0$ [1 - 5]. But $1 + K\frac{B}{A} = 0$,

Therefore,

$$\frac{dK}{ds} = \frac{d(-A/B)}{ds} = \frac{-B\frac{dA}{ds} + A\frac{dB}{ds}}{B^2} = 0 .$$

.i.e $\quad -B\frac{dA}{ds} + A\frac{dB}{ds} = 0 \qquad (23)$

Eqn. (23) is solved for the breakaway points as illustrated in Figure 4.

The points of intersection of the root-locus with the imaginary axis are called imaginary axis crossover. At these points, the system is said to be marginally stable. Routh-Hurwitz stabilty criterion is used to obtain value(s) of K for marginal stability. Roots of the characteristic equation corresponding to these values of K include the imaginary axis crossover points.

The various steps and sub-steps are coded into computer programs. The required inputs to these programs are the open-loop poles and zeros and the output is the root-locus.

3. Tests and Results

3.1 Convergence of Alternative Iteration Formulas

The alternative iteration formulas Eqns. (10) to (16) discussed in Sub-Step 2.1 are tested with an open-loop transfer function. For each iteration formula, the number of successful iterations which produced valid roots and number of failed iterations which produced non-valid roots are recorded and presented in Table 1. Percentage success which is the ratio of successful iterations to total iterations expressed in percentage for each formula is also listed in Table 1. Eqns. (12) and (14) are found to be similar and displayed the same performance. Eqn. (12) is found to be most successful with 83.41% success followed by Eqn. (13) with 14.29% success.

In another experiment, Eqns. (12) and (13) were used complementarily such that Eqn. (12) is used normally and Eqn. (13) is only used when Eqn. (12) failed to produce valid root. 97.27% success was achieved as listed in Table 1. It is therefore concluded that complementary use of iteration formulas Eqns. (12) and (13), is the choice for best performance.

The flowchart of Figure 3 is actually based on complementary use of Eqns. (12) and (13) as iteration formulas.

The effect of use of versions of iteration formulas for even and odd N are tested for two different open-loop transfer functions and the results are summarised in Table 2. Complementary use of both versions produced the best results. It is therefore concluded that, when N is odd, version for odd N should be used and when N is even, version for even N should be used.

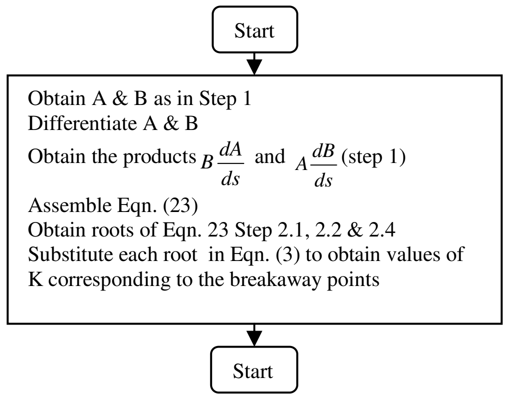


Figure 4: Flow Chart for the Determination of Breakaway Points.

3.2 Sample Root-Locus Plots

Sample root-loci plots for a number of systems are obtained with this computer-aided root-locus numerical technique. These plots are presented in Table 3.

3.3 Adding Details to the Root-Locus Plot

The root-locus dots-plot obtained for a system with open-loop transfer function $G(s)H(s) = \frac{K}{s(s+4)(s^2+4s+20)}$ is shown in Figure 5(a). The details for this system are obtained and are presented in Figure 5(b) and Table 4.

It is not necessary to find these details before plotting the root-locus. However if values of K for breakaway points are determined first, it will assist the choice of intervals in the values of K to be used. For this open-transfer function, K is varied from 0 to 62 at incremental interval of 2. K is varied from 63.5 to 64.5 at the incremental interval of 0.05. K is varied from 66 to 98 at the incremental interval of 2. K is varied from 99.5 to 100.5 at the incremental interval of 0.05. Finally K is varied from 102 to 5000 with varying incremental interval 2 (initially) to 100 (later) and then finally 500. K must vary very slowly near breakaway points.

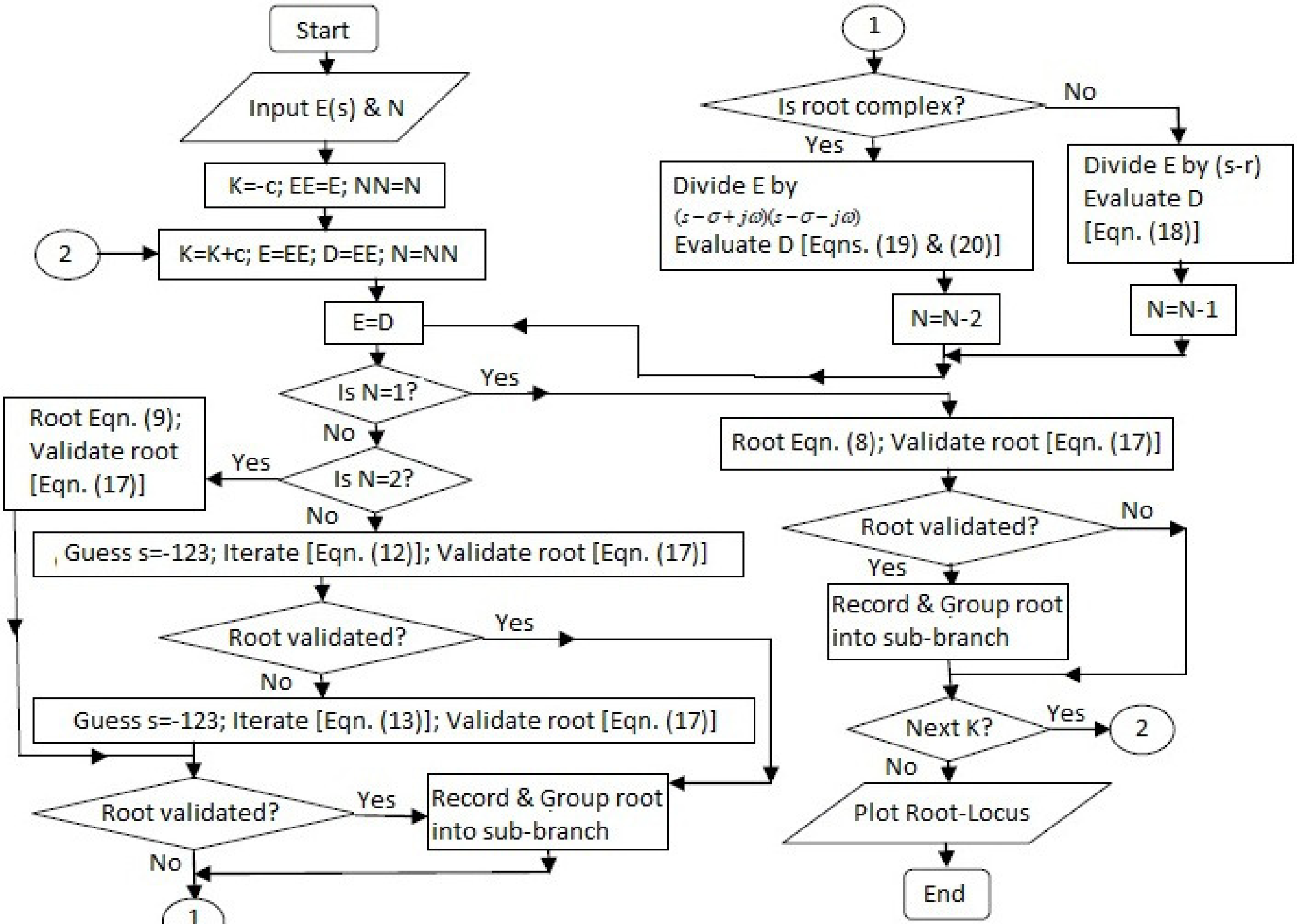


Figure 3: Flow Chart for the Determination and Plotting of Root-Locus as K Varies from 0 to ∞

Table 1: Convergence of Alternative Iteration Formulas for open-loop transfer function

$$G(s)H(s) = \frac{K}{s(s+4)(s^2+4s+20)}$$

Equation	No of successful iterations	No of failed iterations	% success	Remark
(10)	0	868	0%	not reliable
(11)	0	868	0%	not reliable
(12)	724	144	83.41%	most reliable
(13)	124	744	14.29%	reliable near break away point
(14)	724	144	83.41%	same as Eqn. (12)
(15)	0	868	0%	not reliable
(16)	16	852	1.84%	not reliable
(12) & (13)	844	24	97.27%	Eqn. (13) is used when Eqn. (12) fails **The choice for best performance**

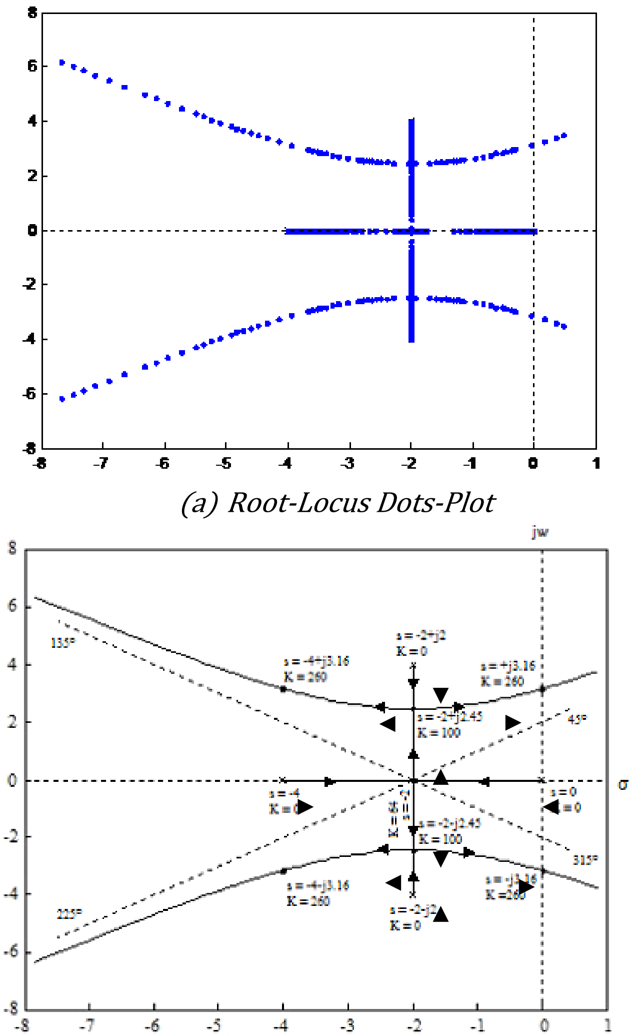
Table 2: Convergence of Even and Odd Versions of Alternative Iteration Formulas

Equation	No of successful iterations	No of failed iterations	% success	Remark
		$G(s)H(s)=\dfrac{K}{s(s+4)(s^2+4s+20)}$		
(12) & (13) even version only	595	273	68.55%	Only version for even N was used for both even and odd N
(12) & (13) odd version only	0	868	0%	Only version for odd N was used for both even and odd N
(12) & (13) both versions	844	24	97.27%	Both versions were used as appropriate
		$G(s)H(s)=\dfrac{K(s+1)(s+3)(s^2+2s+2)}{s(s+2)(s+4)(s^2+4s+13)}$		
(12) & (13) even version only	1	1088	0.0009%	Only version for even N was used for both even and odd N
(12) & (13) odd version only	185	904	16.99%	Only version for odd N was used for both even and odd N
(12) & (13) both versions	873	216	80.17%	Both versions were used as appropriate

There are four branches starting from the four poles and ending at ∞. For the purpose of grouping into sub-branches, six sub-branches are identified. Sub-branch 1 is on the real axis. Sub-branch 2 is parallel to the imaginary axis at $s=-2$. Sub-branches 3 and 4 are above the real axis and are to the right and left of Sub-branch 2 respectively. Sub-branches 5 and 6 are below the real axis and are to the right and left of Sub-branch 2 respectively. Figure 6 shows the flow chart for grouping of root of this system into a sub-branch.

3.4 Comparison with Existing Technique

One of the existing techniques is the 'rlocus' code in Matlab [25]. The results obtained in this work are compared with the results obtained with the existing Matlab 'rlocus' code. Table 5 shows that the roots obtained by iteration in this work are exactly the same as those obtained using an existing Matlab 'rlocus' Code for three different open-loop transfer functions. The selected pair of iteration formulas developed in this work can be adapted to determine roots of equations for other applications besides root-locus.



(a) Root-Locus Dots-Plot

(b) Complete Root-Locus with details

Figure 5: Root-Locus with details for $G(s)H(s)=\dfrac{K}{s(s+4)(s^2+4s+20)}$

Table 3: Sample Root-Locus Plots Obtained

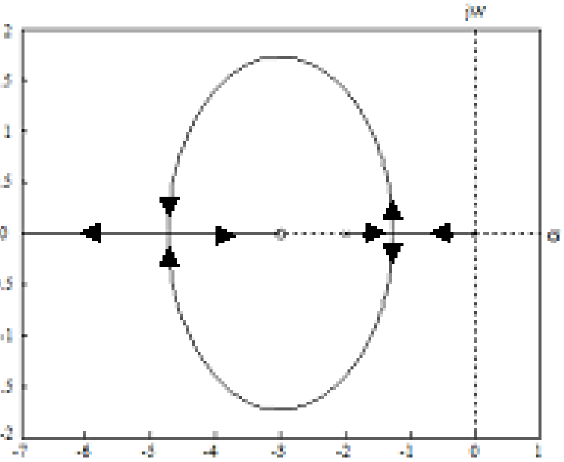
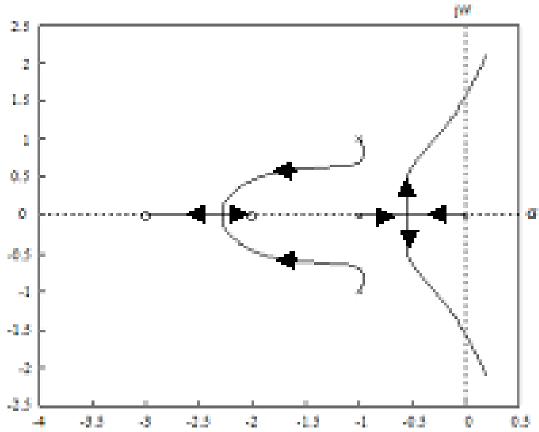
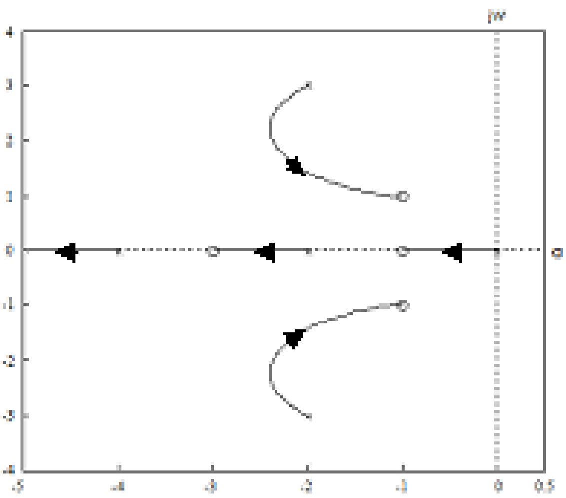
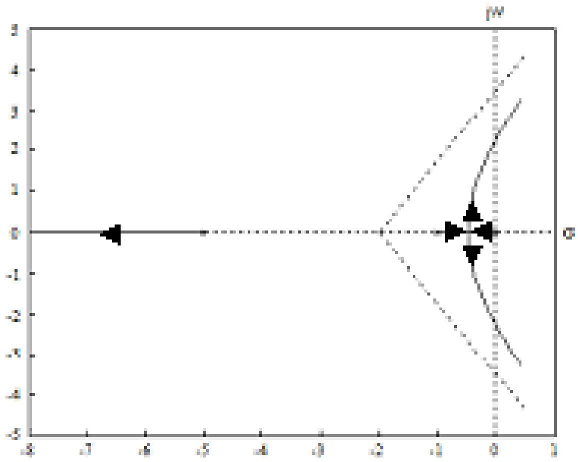
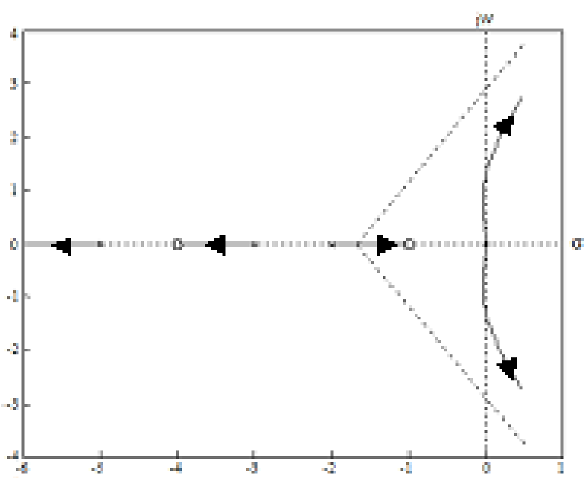
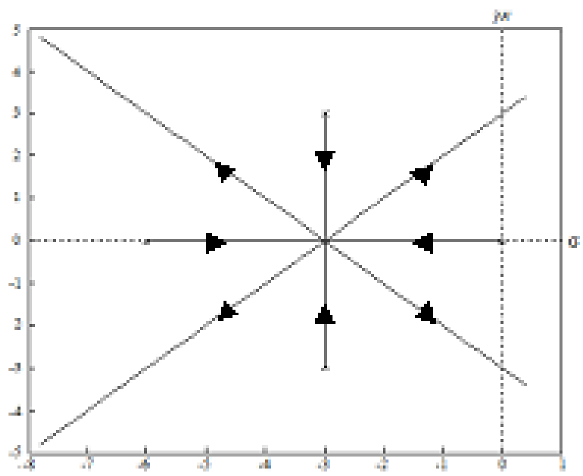
S/N	Sample Plot	S/N	Sample Plot
1.	$G(s)H(s)=\frac{K(s+3)}{s(s+2)}$ 	2.	$G(s)H(s)=\frac{K(s+2)(s+3)}{s(s+1)(s^2+2s+2)}$ 
3.	$G(s)H(s)=\frac{K(s+1)(s+3)(s^2+2s+2)}{s(s+2)(s+4)(s^2+4s+13)}$ 	4.	$G(s)H(s)=\frac{K}{s(s+1)(s+5)}$ 
5.	$G(s)H(s)=\frac{K(s+1)(s+4)}{s^2(s+2)(s+3)(s+5)}$ 	6.	$G(s)H(s)=\frac{K}{s(s+6)(s^2+6s+18)}$ 

Table 4: Details for $G(s)H(s) = \dfrac{K}{s(s+4)(s^2+4s+20)}$

S/N	Description	Results
1	Open-loop poles	$p_1 = 0; p_2 = -2$; $p_3 = -2 - j4; p_4 = -2 + j4$
2	Open-loop zeros	Nil
3	Point of intersection of asymptotes	$s = -2$
4	Angles between the four asymptotes and real axis	45°, 135°, 225° and 315°
5	Breakaway points	$s = -2$ ($K = 64$); $s = -2 - j2.45$ ($K = 100$) $s = -2$ ($K = 64$); and $s = -2 + j2.45$ ($K = 100$)
6	Value of K obtained for marginal stability	$K = 260$
7	Imaginary crossover	$s = -j3.16$ and $s = +j3.16$
8	Additional roots for $K = 260$	$s = -4 - j3.16$ and $s = -4 + j3.16$

Table 6 compares the root-locus plots obtained in this work and those obtained using an existing Matlab 'rlocus' Code for three different open-loop transfer functions. The plots are the same in terms of shape and values of roots. The plots obtained in this work are magnified as the range of the plots along the real axis is controlled and limited to $x \le real(s) \le +0.5$. x is negative and is less than the real part of each of the open-loop poles and zeros. This is sufficient as the system is unstable in the range $real(s) \ge +0.5$ and the plot is predictable in the range $-\infty \le real(s) \le x$. However, the user of the Matlab 'rlocus' code do not have such convenient control over the range of the plot along real axis.

The input to the algorithm developed in this work can be either open-loop poles and zeros or transfer function expressed as the ratio of two polynomials. This is because it included an expansion process (section 2.1). This is an advantage over the existing Matlab 'rlocus' code which only accept transfer function expressed as the ratio of two polynomials. Furthermore, the developed algorithm adds details of asymptotes, breakaway points and imaginary axis crossover to the root-locus plot (section 2.4). This is another advantage over the existing Matlab 'rlocus' code.

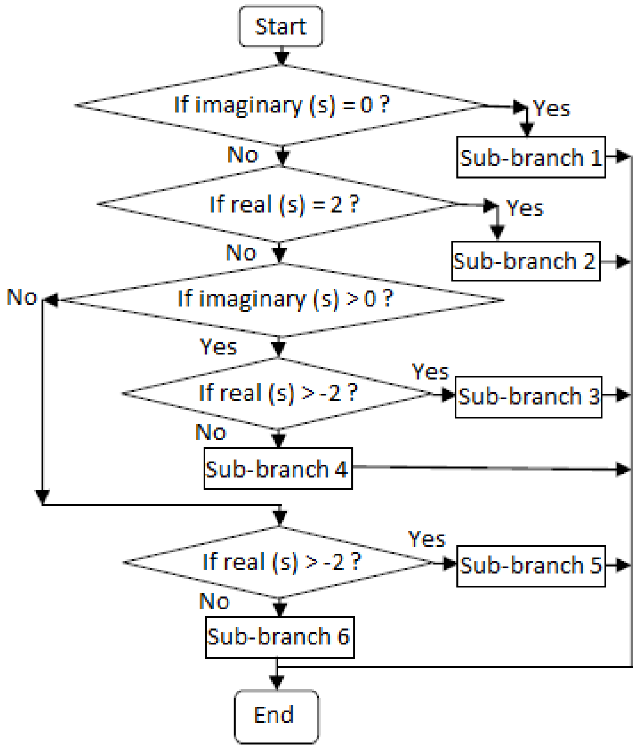
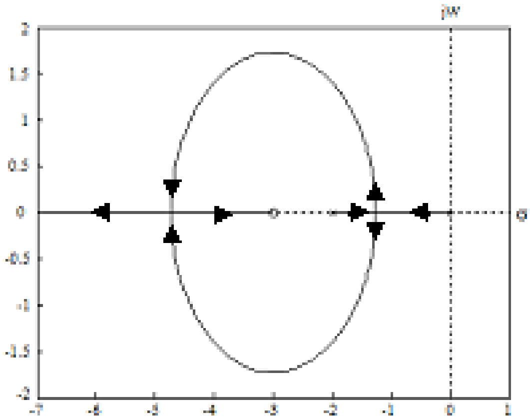
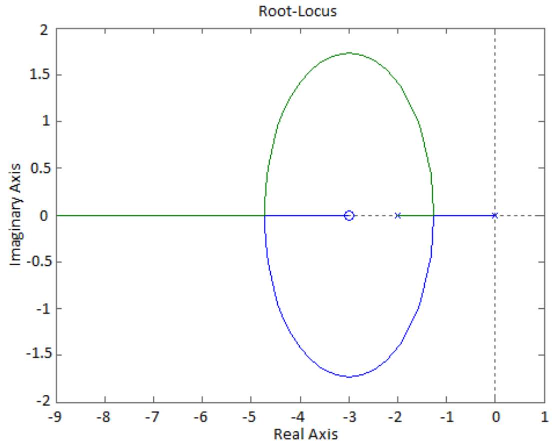
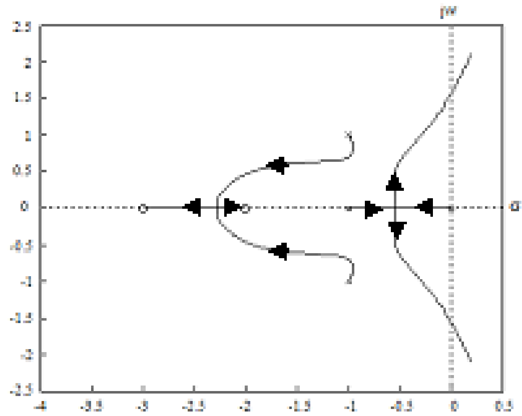
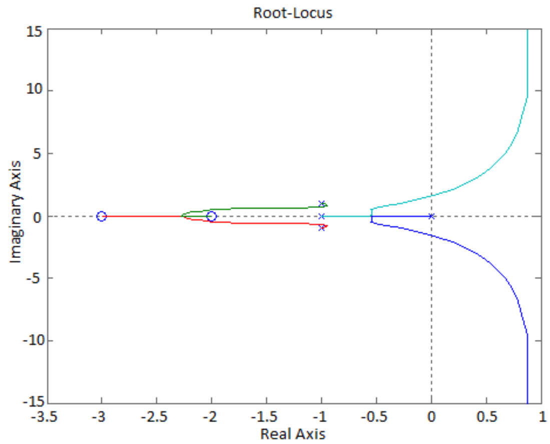
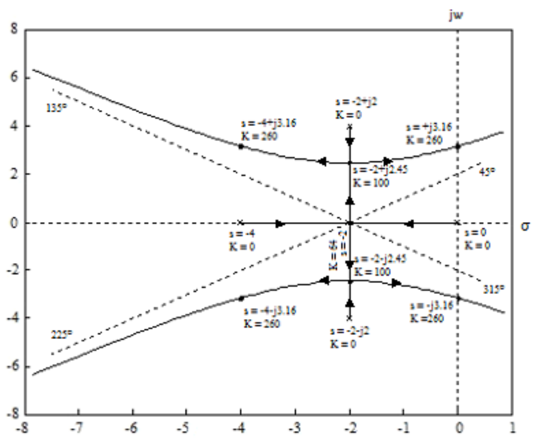
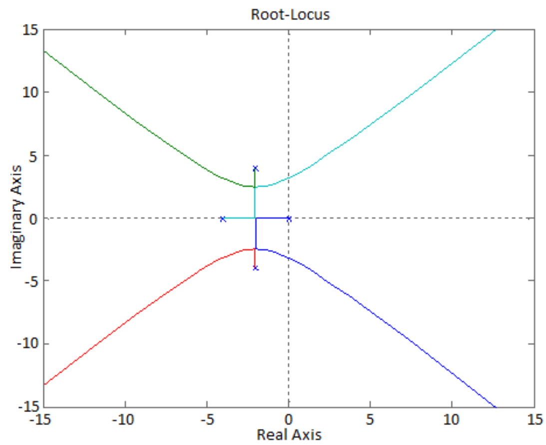


Figure 6: Flow chart for Grouping of a Root of $G(s)H(s) = \dfrac{K}{s(s+4)(s^2+4s+20)}$ *into a Sub-Branch.*

Table 5: Comparison of roots obtained in this work and those obtained using existing Matlab 'rlocus' code for three different open-loop transfer functions.

	$G(s)H(s) = \dfrac{K(s+3)}{s(s+2)}$ K = 1.2		$G(s)H(s) = \dfrac{K(s+2)(s+3)}{s(s+1)(s^2+2s+2)}$ K = 0.2		$G(s)H(s) = \dfrac{K}{s(s+4)(s^2+4s+20)}$ K = 3.55	
	Iteration Results	Existing Matlab 'rlocus' Code	Iteration Results	Existing Matlab 'rlocus' Code	Iteration Results	Existing Matlab 'rlocus' Code
1st root	-1.6+j1.02	-1.6+j1.02	-1.061+j0.67	-1.061+j0.67	2.0+j3.978	2.0+j3.978
2nd root	-1.6-j1.02	-1.6-j1.02	-1.061-j0.67	-1.061-j0.67	2.0-j3.978	2.0-j3.978
3rd root			-0.439+j0.755	-0.439+j0.755	-3.955	-3.955
4th root			-0.439+j0.755	-0.439+j0.755	-0.045	-0.045

Table 6: Comparison of root-locus plots obtained in this work and those obtained using existing Matlab 'rlocus' code for three different open-loop transfer functions.

S/N	Root-Locus Plot Obtained in This Work	Root-Locus Plot Obtained With Existing Matlab 'Rlocus' Code
1.	$G(s)H(s)=\dfrac{K(s+3)}{s(s+2)}$ 	$G(s)H(s)=\dfrac{K(s+3)}{s(s+2)}$ 
2.	$G(s)H(s)=\dfrac{K(s+2)(s+3)}{s(s+1)(s^2+2s+2)}$ 	$G(s)H(s)=\dfrac{K(s+2)(s+3)}{s(s+1)(s^2+2s+2)}$ 
3.	$G(s)H(s)=\dfrac{K}{s(s+4)(s^2+4s+20)}$ 	$G(s)H(s)=\dfrac{K}{s(s+4)(s^2+4s+20)}$ 

4. Conclusion

A computer aided root-locus numerical technique based on simple fast mathematical iteration has been developed. The technique has been tested successfully. The geometric properties of the root locus obtained with this technique are found to conform to those described in the literature. Using this technique, root-locus is drawn to scale instead of rough sketches. This technique overcome some of the limitations of the existing Matlab 'rlocus' code and will facilitate linear time invariant control system design.

5. References

[1] Chen, C. T. *Analog and Digital Control System Design*, Saunders College, Orlando, Fla, USA, 1993.

[2] D'Azzo, J. J. and Houpis, C. H. *Linear Control System Analysis and Design: Conventional and Modern*, McGraw-Hill, New York, NY, USA, 1995.

[3] Dorf, R. C. and Bishop, R. H. *Modern Control Systems*, Addison-Wesley, Reading, Mass, USA, 1998.

[4] Nagrath, I.J. and Gopal, M. *Control Systems Engineering*, Wiley Eastern Ltd, New Delhi, India, 1975.

[5] Ogata, K. *Modern Control Engineering*, Prentice-Hall, Upper Saddle River, NJ, USA, 2001.

[6] Anih, L.U. and Amah, O. "The performance characteristics of a closed loop, one axis electromechanical solar tracker", *Nigerian Journal of Technology*, Vol. 25, Number 2, 2006, pp. 66-76.

[7] Oti, S.E. and Okoro, O.I. "Stability analysis of static slip-energy recovery drive via eigenvalue method", *Nigerian Journal of Technology*, Vol. 26, Number 2, 2007, pp. 13-29.

[8] Itaketo, U.T., Ogbogu, S.O.E. and Chukwudebe, G.A. "Apllication Of Lyapunov's Second Method In The Stability Analysis Of Oil/Gas Separation Process", *Nigerian Journal of Technology*, Vol. 21, Number 1, 2002, pp. 60-71.

[9] Evans, W. R. "Control system synthesis by root locus method", *Transactions of the American Institute of Electrical Engineers*, Vol. 69, Number 1, 1950, pp. 66–69.

[10] Evans, W. R. "Graphical analysis of control systems," *Transactions of the American Institute of Electrical Engineers*, Vol. 67, Number 1, 1948, pp. 547–551.

[11] Krall, A. M. "The root locus method: a survey", *SIAM Review*, Vol. 12, Number 1, 1970, pp. 64–72.

[12] Pan, C.T. and Chao K.S. ''A Computer-Aided Root-Locus Method'', *IEEE Transactions on Automatic Control*, Vol. AC-23, Number 5, October 1978, pp 856-860.

[13] Berman G. and Stanton, R. G. "The asymptotes of the root locus", *SIAM Review*, Vol. 5, Number 3, 1963, pp. 209–218.

[14] Byrnes, C. I., Gilliam, D. S. and He, J. "Root-locus and boundary feedback design for a class of distributed parameter systems", *SIAM Journal on Control and Optimization*, Vol. 32, Number 5, 1994, pp. 1364–1427, 1994.

[15] Eydgahi, A. M. and Ghavamzadeh, M. "Complementary root locus revisited", *IEEE Transactions on Education*, Vol. 44, Number 2, 2001, pp. 137–143.

[16] Monteiro, L. H. A. and Da Cruz, J. J. "Simple answers to usual questions about unusual forms of the Evans' root locus plot", *Controley Automacao*, Vol. 19, Number 4, 2008, pp. 444–449.

[17] Bahar, E. and Fitzwater, M. "Numerical technique to trace the loci of the complex roots of characteristic equations", *SIAM Journal on Scientific and Statistical Computing*, Vol. 2, Number 4, 1981, pp. 389–403.

[18] Krall, A. M. "A closed expression for the root locus method", *SIAM Journal on Applied Mathematics*, Vol. 11, Number 3, 1963, pp. 700–704.

[19] Merrikh-Bayat, F. and Afshar, M. "Extending the root-locus method to fractional-order systems", *Journal of Applied Mathematics*, Vol. 2008, Article ID 528934, 2008, pp1-13.

[20] Merrikh-Bayat, F., Afshar, M. and Karimi-Ghartemani, M. "Extension of the root-locus method to a certain class of fractional-order systems", *ISA Transactions*, Vol. 48, Number 1, 2009, pp. 48–53.

[21] Spencer, D. L., Philipp, L. and Philipp, B. "Root loci design using Dickson's technique", *IEEE Transactions on Education*, Vol. 44, Number. 2, 2001, pp. 176–184.

[22] Teixeira, M. C. M., Assuncao, E. and Machado, E. R. M. D. "A method for plotting the complementary root locus using the root-locus _positive gain_ rules", *IEEE Transactions on Education*, Vol. 47, Number 3, 2004, pp. 405–409.

[23] Teixeira, M. C. M., Assuncao, E., Cardim, R., Silva, N.A.P. and Machado,E. R. M. D. "On Complementary Root Locus of Biproper Transfer Functions", *Journal of Applied Mathematics*, Vol. 2009, Article ID 727908, 20091, pp 1-4.

[24] Anyaegunam, A. J. "Simple formulae for the evaluation of all the exact roots (real and complex) of the general cubic", *Nigerian Journal of Technology*, Vol. 27, Number 1, 2008, pp. 48-53.

[25] MathWorks (Matlab), Documentation for MathWorks products retrieved on January 1, 2009 from http://www.mathworks.com/access/helpdesk/help/helpdesk.shtml

[26] Kelley, C. T. *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia 1995.

[27] Onyia, O. T., Madueme, T. C. and Omeje, C. O. "An improved voltage regulation of a distribution network using facts - devices", *Nigerian Journal of Technology*, Vol. 32, Number 2, 2013, pp. 304-317.