الأسم : عبدالعزيز السيد عبدالعزيز راشد                    الفرقة : الثانية
رقم الجلوس : 40                    القسم : الحاسبات والنظم

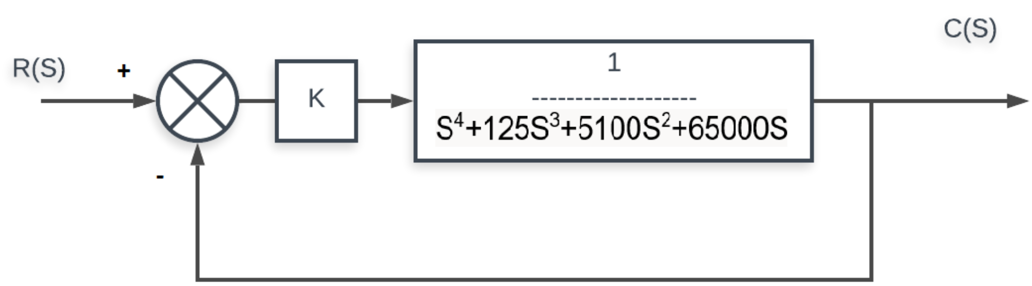
# Root Locus Drawer

Project Link Here

## Problem Statement

Given the following open loop transfer function with four poles at S = 0, S= -25, S=-50 + j10 and S = -50 –j10 and no zeroes.

It is required to write a program to draw the root locus following the rules.




## Used Data Structures :

Point :

A data structure for storing Pole properties (X,Y and angle of departure).

LineSegment :

A data structure to store line segment between two critical frequencies (shown with rules).

ArrayList :

For storing poles, zeros, Asymptotes angles and line segments between two critical frequencies.

## Classes :

Controller :

The main class responsible for drawing root locus by the rules and changing the value of K in the characteristics equation.

KController:

A class for controlling the window of changing K properties

Line Segment, Pole , Zero :
used as data structures described above in data structure section
Utility :

A class used for utilities algorithms ( Getting the roots of a given function)

## Things done outside the program :

Finding Break point : by solving $\frac{d}{ds}(C(s)) = 0$
Finding intersection with imaginary axis : using Routh criteria

## Algorithms and functions :

Class Controller :

public void initialize()
Initializer method for controller Javafx
We set scaller and drawing axis X and Y in it

public void setScale()
setting the scale of canvas

public void View()
Responsible for viewing whether with rules or with changing K value or both

private void viewVaryOfK()
Responsible for getting and drawing the root locus by changing the value of K in the characteristics equation
Uses class Utility to find the roots
The K value changes depending on some properties (Start value of K (ex. 1) , step (ex : 1000) and number of iterations (ex 1000))

$$1 + k_i\, G(s)H(s) = 0$$

Where I goes from 1 to the number of steps and $k_{i+1} = k_i + step$

public void viewUsingRules()
a function for drawing the root locus using the rules, inside of it we get the number of poles , zeros
and the angle of Asymptotes using the formula $\theta_i = \frac{(2q+1)*180}{m-n}$
where m = number of poles and n = number of zeros
The centroid point = $\frac{\sum_{i=0}^{m} p_i - \sum_{i=0}^{n} Z_i}{m-n}$
Drawing line segments and calculating the angle of departure for every pole then Complete the draw
private double angleOfDeprture(Point polee,ArrayList<Point> poles,ArrayList<Point> zeros)

a function for calculating the angle of departure for every pole
using the formula $\varphi = 180 - \sum \varphi + \sum \theta$

## Note :

The remaining functions in the class are only some helper functions for drawing

Class Utility :

`public static Complex64F[] findRoots(double... coefficients)`

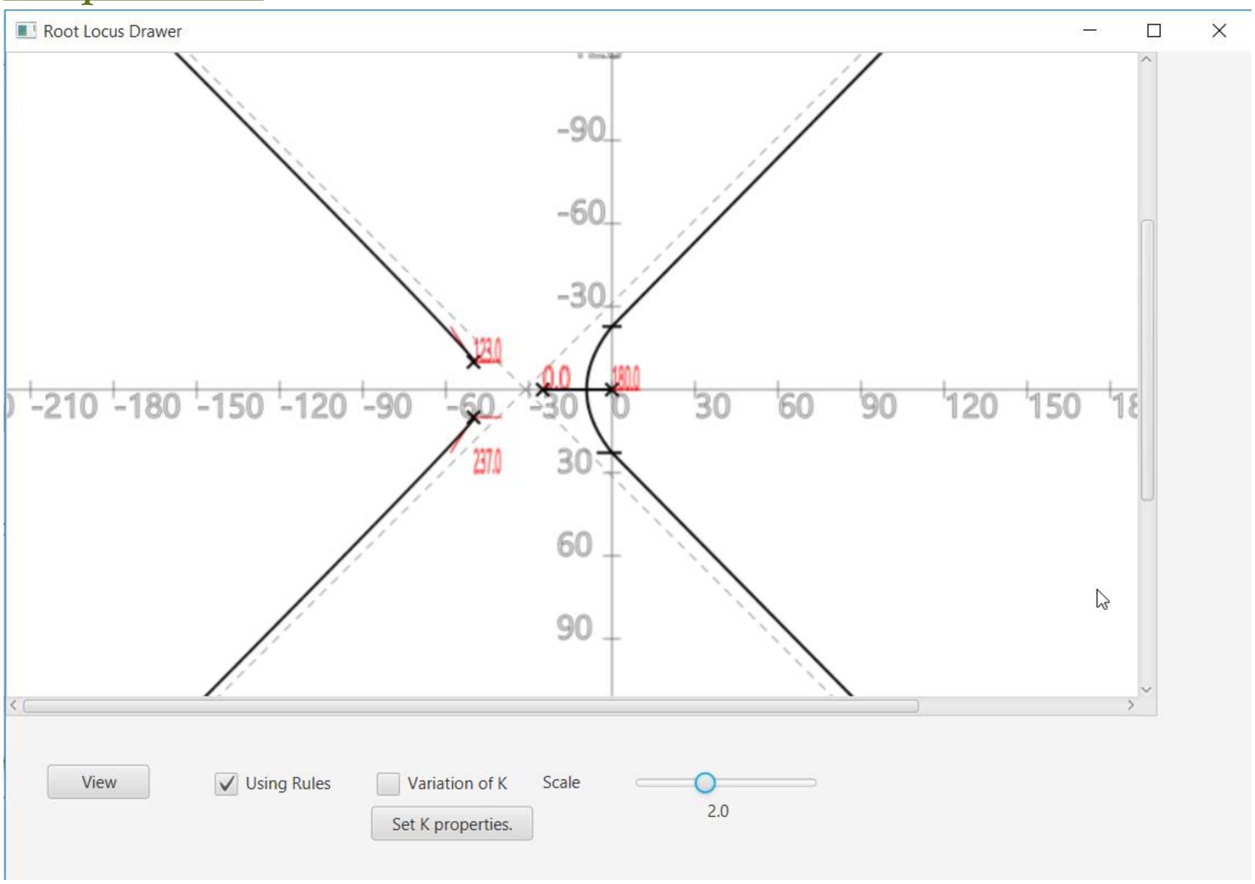
A function for finding the root of a given f(x)

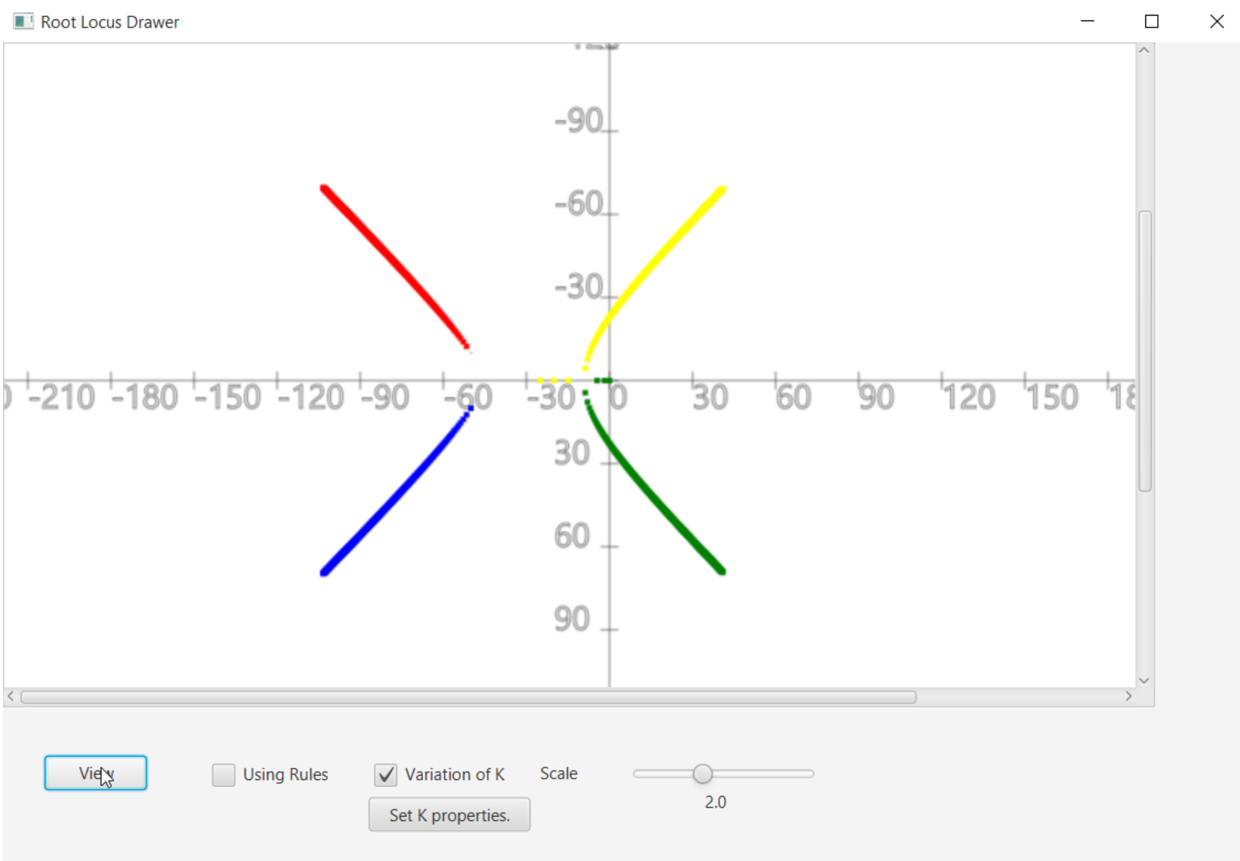
Using the the Algorithm

$$R = eig(A)$$

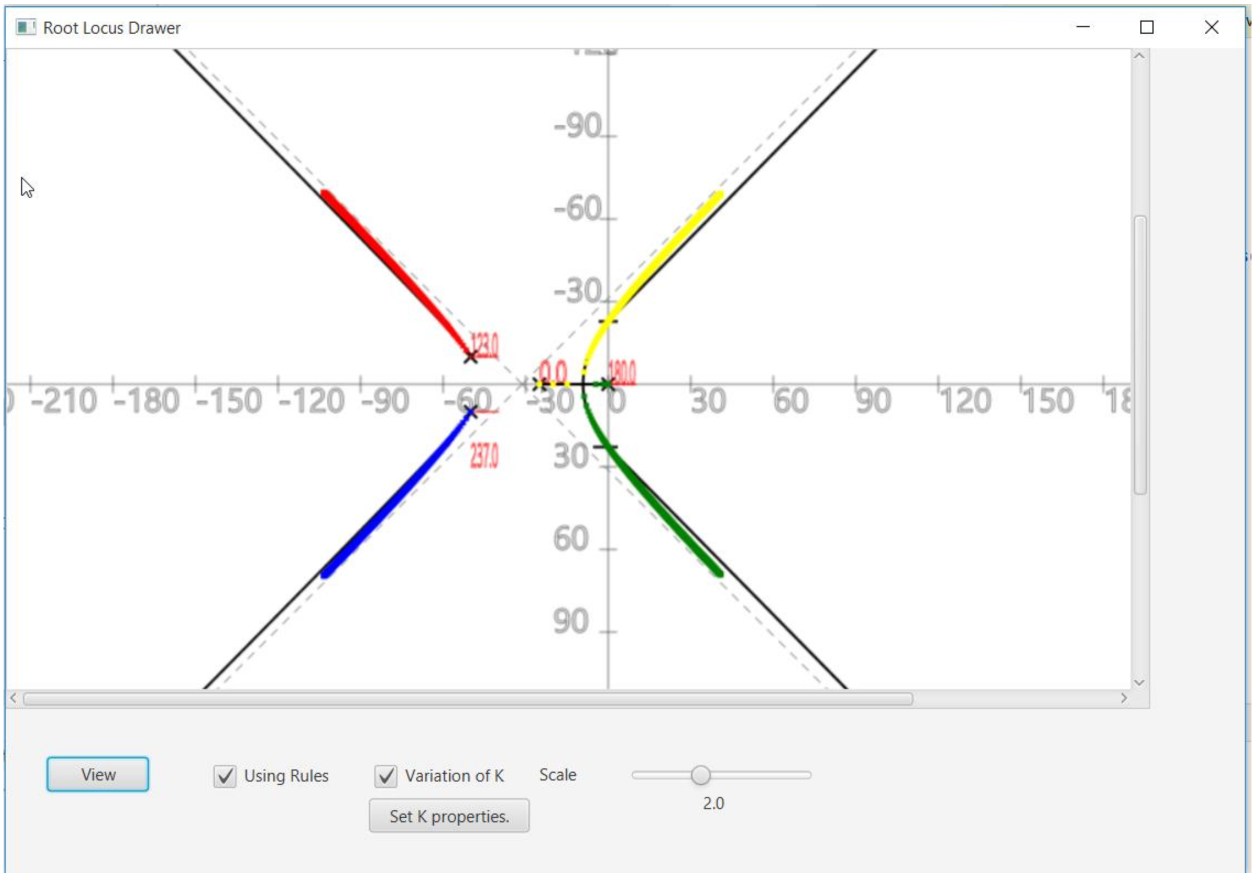
Where A is the companion matrix of f(x) and eig(A) is the eigenvalues of A

# Sample Runs :





Root Locus Drawer
-90
-60
-30
-210 -180 -150 -120 -90 -60 -30 0 30 60 90 120 150
30
60
90
View
Using Rules
Variation of K
Scale
2.0
Set K properties.




Note :

All needed external libraries will be found in “lib” folder