ROOT LOCUS 7

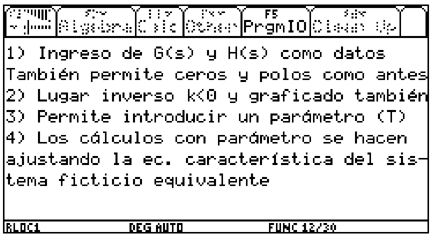
RESOLUCIÓN DE PROBLEMAS DE LUGARES DE RAÍCES PARA LA

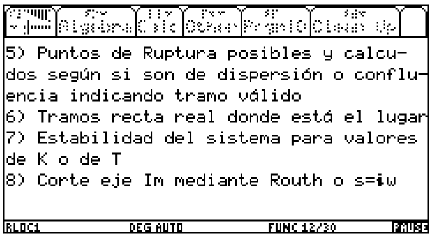
TEXAS INSTRUMENTS 92 PLUS.

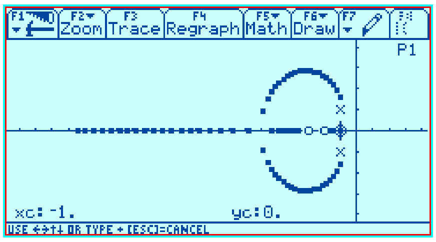
por

José Manuel Gómez Vega

gomezvega@hotmail.com







...es otro programa paso a paso...

ÍNDICE

Anterior	**1. Historia y origen de Root Locus 6.**	Siguiente

Chadd L. Easterday (página web: http://easterday.home.mindspring.com , correo electrónico: easterday@mindspring.com) creó un programa para calcular el lugar de las raíces. Además hizo algunas funciones que acompañaban al programa. La versión era Root Locus 6. Estaba escrito en inglés. Después de mejorar este programa, he contactado con el autor de Root Locus 6 ya que he aprovechado algunas características de dicho programa, aunque lo he transformado para ser capaz de hacer todos los cálculos requeridos en un problema escrito, y me ha dado su autorización para publicarlo como una continuación a su aportación bastante mejorada.

Anterior	**2. Tipo de Calculadora, instalación.**	Siguiente

Introducción

Chadd L. Easterday realizó las funciones y programas que se detallan a continuación. José Manuel Gómez (yo) he reprogramado Root Locus para transformarlo en un programa "Paso a Paso". En definitiva me he centrado en el programa principal *Rlocus7* (la versión anterior era *Rlocus6*). También he modificado parcialmente *Rlplot* (ahora es *Rlplot2*) y he corregido dos bugs (errores) en la función *azeros.* El programa Ruptura() es nuevo. No he comprobado minuciosamente todas las funciones, pues algunas no se precisan en Rlocus7. Aún así las he incluido todas.

Instalación

Después de instalar, no hay que mover, renombrar o borrar ninguno de los programas o funciones. Si se omite esta advertencia puede que los programas o funciones operen de forma extraña o no lo hagan indicando errores. Use TI GraphLink o TI Connect para transferir los ficheros a la calculadora. Los ficheros se envían automáticamente a la carpeta Rlocus. Se recomienda crear una carpeta nueva con *Newfold* nombredecarpeta, por ejemplo, *Newfold* Rloc10 para el problema 10 y de esa forma tendremos dos ventajas:

1) Podremos borrar directamente las variables cuando estén en memoria sin confundir con las variables que haya en *Rlocus.*

2) Podremos tener varios problemas simultáneamente en memoria y abrir el problema que se quiera cada vez, sin necesidad de introducir los datos.

Tipo Calculadora y S.O

El programa se ha realizado para la Texas Instruments 92 Plus. Aunque no lo he probado, debe valer igualmente para la Voyage 200 pues no existen diferencias salvo en recursos de memoria. Sin embargo, para la Texas Instruments 89 no debe funcionar por problemas de dimensión de ancho y alto de columnas en funciones *Text*, *Disp*, *Output*, etc y graficado de funciones que habría que modificar para hacerlo compatible.

Lo he probado en sistemas operativos AMS 2.05, 2.08 y 2.09 sin problemas de cuelgues o fallos. Los archivos cuando se instalan en la calculadora están archivados. De esta forma, si se envían a la calculadora mediante TI Connect o TI Graph Link, van preparados para ejecutarse con rapidez, ocupando poca memoria. Si se usa el emulador y se envía, se cargan pero con la protección *Lock* (¡pero sin archivar!), por lo que los programas correrán más despacio. Se recomienda realizar *Unlock rlocus7* que desbloquea el programa principal, y seguidamente cargar dicho programa (en la carpeta Rlocus), mediante *rlocus7()*. Seguidamente pulsar *ON* y esperar hasta que pasen unos 20 segundos hasta que el programa haga *BREAK.* Acto seguido se archiva el programa con *Archive rlocus7().* De esta forma se consigue que dicho programa arranque automáticamente cuando se le llame con *rlocus7()*. En teoría habría que hacer esto con todos los demás programas cuando se usa el emulador, pero he observado que dado que los demás programas se ejecutan más rápidamente puede optarse a no hacerlo (eso es decisión del usuario).

Contenido carpeta Control7

La carpeta Control 7 contiene todos los archivos agrupados antes de enviarse a la calculadora.

Carpeta de programas y funciones en calculadora: rlocus

Programa principal: rLocus7() - Root Locus versión 7 (ejecutado llama a los demás programas y funciones involucrados).

Tipo Archivos: Programas y Funciones (en TI Basic).

Materia: Colección de funciones y programas de Ingeniería de Control.

Ejecución programa principal: rlocus\rLocus7() (desde cualquier carpeta).

Otras carpetas: No.

Archivos necesarios (se incorporan todos en el archivo de instalación):

La carpeta rlocus\ debe contener los siguientes programas y funciones : rlocus7(), ruptura(), asa(), ecar(), rlopause(), rlplot2(), mrowdel(), azeros(), cofpoly(), compbar(), cpoles(), lsa(), os(), poles(), poly2cof(), polydeg(), polydiv(), rh(), rtspoly(), sortr(), ss2tf(), ss2tf2(), tf(), tp(), zpk(), delelem(), interval(), iscmplxn(), isolve(), listswap(), memberq(), mor(), replcstr(), rmdup(), sort() y varlist()

Todas las variables creadas pueden borrarse una vez que se ha calculado un programa. Si se quiere ejecutar un programa con variables en memoria de un problema y se borran de forma manual parcialmente, a la hora de ejecutarse es posible que dé errores el programa, lógicamente.

Breve descripción funciones y programas

rlocus\aZeros() - Todos los ceros

Tipo archivo: Función

Sintaxis: rlocus\aZeros(expr, var)

Objetivo: Encontrar todas las raíces de expr respecto a var, dando todas las raíces repetidas. Mejora la función del C.A.S zeros

rlocus\cof2Poly() - Conversión de Coeficientes a Polinomio

Tipo archivo: Función

Sintaxis: rlocus\cof2Poly(coefList)

Archivos requeridos: rlocus\sortR()   (No confundir sort con sortR)

Objetivo: Da el polinomio que resulta de los coeficientes del polinomio puestos en coefList en forma de lista.

rlocus\compBar() - Completando la Barra

Tipo archivo: Programa

Sintaxis: rlocus\compBar(tamaño, {x, y}, %Completo, %previo)

Objetivo: Muestra barra para señalizar porcentaje de ejecución ya completada. Se emplea en Rlplot2., mostrando el porcentaje de tiempo ya calculado en el lugar.

rlocus/cPoles() - Polos complejos

Tipo archivo: Función

Sintaxis: rlocus\cPoles(FuncionTransferencia, variable)

Objetivo: Devuelve los polos complejos de FuncionTransferencia respecto a variable.

rlocus/lsa() - Nuevo tamaño lista

Tipo archivo: Función

Sintaxis: rlocus\lsa(list, newSize, shiftDir)

Objetivo: Se usa en polyDiv(). Da otro tamaño a la lista en la dirección de shiftDir.

rlocus/os() - Sobreoscilación

Tipo archivo: Función

Sintaxis: rlocus\os(z)

Objetivo: Dada z=ζ, se calcula $Mp=e^{\frac{-z.\pi}{\sqrt{(1-z^2)}}}$

rlocus/Poles() - Polos

Tipo archivo: Función

Sintaxis: rlocus\poles(expr, var)

Objetivo: Da los polos de expr respecto a var.

rlocus/poly2Cof() - Conversión de Polinomio a Coeficientes

Tipo archivo: Función

Sintaxis: relocus\poly2Cof(poly, var)

Archivos requeridos: rlocus\sortR()

Objetivo: Da una lista de los coeficientes de poly en términos de var.

rlocus/polyDeg() - Grado de un polinomio

Tipo archivo: Función

Sintaxis: rlocus\polyDeg(poly, var)

Objetivo: Da el grado de poly en términos de var.

rlocus/polyDiv() - División larga de un polinomio

Tipo archivo: Función

Sintaxis: rlocus\polyDiv(numCoefList, denCoefList, #Terms)

A**rchivos requeridos:** rlocus\lsa()

Objetivo: División larga de polinomios.

rlocus/rh() - Tabla de Routh-Hurwitz

Tipo archivo: Función

Sintaxis: rlocus\rh(coefList)

Objetivo: Devuelve la matriz de Routh-Hurwitz del polinomio característico dado por coefList que es una lista de los coeficientes del polinomio.

rlocus/rts2Poly() - Obtención de polinomio mediante la lista de raíces

Tipo archivo: Función

Sintaxis: rlocus\rts2Poly(rootList)

Objetivo: Devuelve el polinomio dado por rootList.

rlocus/sortR() - Lista revertida

Tipo archivo: Función

Sintaxis: rlocus\sortR(list)

Objetivo: Da una lista en orden invertido.

rlocus/ss2tf() y ss2tf2() - Cambio de Espacio-Estado a Función de transferencia

Tipo archivo: Función

Sintaxis: rlocus\ss2tf(A, B, C)

Objective: Da la función de transferencia de tiempo continuo dado un espacio-estado en forma de matrices A, B, y C.

El autor refiera la misma descripción para ambas funciones.

rlocus/tf() - Función de Transferencia

Tipo archivo: Función

Sintaxis: rlocus\tf(numCoefList, denCoefList)

Objetivo: Devuelve la función de transferencia en tiempo continuo.

rlocus/tp() - Tiempo de pico

Tipo archivo: Función

Sintaxis: rlocus\tp(wn, zeta)

Objective: Devuelve el tiempo de pico dado por la frecuencia natural (wn) y el factor de amortiguamiento (zeta=ζ).

rlocus/zpk() - Función de transferencia entrando zeroList, poleList y gain

Tipo archivo: Función

Sintaxis: rlocus\zpk(zeroList, poleList, gain)

Archivos Requeridos: rlocus\rts2Poly()

Objetivo: Devuelve la función de transferencia en tiempo continuo.

rlocus/rlocus7() - Programa principal

Tipo archivo: Programa

Sintaxis: rlocus\rLocus7()

Archivos necesarios: rlocus\aZeros(), rlocus\poly2Cof(), rlocus\rh(), \rlocus\rts2Poly(), rlocus\rlpic89, rlocus\rlpic92

Objetivo: Diseñado para analizar el lugar de las raíces partiendo de una función de transferencia.

rlocus/ruptura() - Programa secundario rlocus7()

Tipo archivo: Programa

Sintaxis: No se debe ejecutar directamente; acompaña a rlocus7().

Objetivo: Subprograma para calcular puntos de ruptura.

rlocus/rlpic - Gráfico Función de Transferencia

Tipo archivo: Dibujo.

Sintaxis: Ninguna

Objetivo: Ver las funciones de transferencia y la ganancia junto con la realimentación simbólicamente.

rlplot2() - Dibujo del lugar de las raíces

Tipo archivo: Programa

Sintaxis: rlocus\rlplot(charEqn, initialGain, finalGain, gainStep)

Objetivo: Dibuja el lugar de las raíces desde initialGain a finalGain siendo incrementadas por gainStep.

rlocus/mrowdel() - Borra una fila de una matriz

Tipo archivo: Función. (descrita en Tip List 10).

Sintaxis: rlocus\mrowdel(mat,fil)

Objetivo: Borra fila fil de matriz mat.

delelem(), interval(), iscmplxn(), listswap(), memberq(), mor(), replcstr(), rmdup(), sort() y varlist() - Funciones necesarias para que funcione isolve (no se describen)

rlocus/isolve() - Resuelve inecuaciones

Tipo archivo: Función.

Sintaxis: rlocus\isolve(ineq,var)

Objetivo: Resuelve la inecuación ineq respecto a la variable var. No es multivariable (solo calcula si hay una variable simbólica en la expresión)

Anterior	3. ¿Qué hace Root Locus 7?.	Siguiente

3. ¿Qué hace Root Locus 7?.

BREVE DESCRIPCIÓN. PROBLEMAS QUE RESUELVE

Root Locus 7 realiza cálculos de lugares y contornos de las raíces en problemas de Ingeniería de Regulación y Control para sistemas con funciones de transferencia en lazos de realimentación. Sirve para comprobar si dichos sistemas son o no estables observando la evolución de los polos en el tiempo mediante un cálculo doble (analítico y gráfico) completo, dependiendo de los valores de la ganancia K o de un parámetro T, que pueden ser ajustados tras el estudio.

- **Para PROBLEMAS DE LUGARES DE LAS RAÍCES:**

Se introduce una función de transferencia G(s) en lazo abierto junto con una ganancia K simbólica. Se da el dato de H(s) que es la función de transferencia en lazo cerrado; se da el dato de si es realimentación negativa, positiva y si es para el lugar directo (K>0) o inverso (K<0)...¡y eso es todo! El programa se encarga de forma secuencial y según la opción que se elija del cálculo analítico y gráfico del lugar de las raíces ofreciendo todos los cálculos intermedios para llegar a los resultados.

- **Para PROBLEMAS DE CONTORNO DE LAS RAÍCES:**

En estos casos se estudia para un parámetro (el parámetro debe ser t ó T); las funciones G(s) y H(s) también se piden como dato, pero en este caso K es un valor numérico. Se resuelve el contorno de las raíces para la función según $1+T.G.H_1=0$, calculándose $G.H_1$.

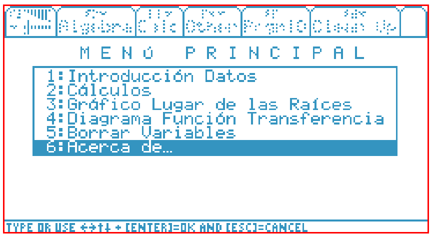
El sistema de presentación de datos es ideal para comprobar problemas completos, ya sean manuscritos o referidos en libros. Se han corregido muchos bugs. El último ha sido referente a la presentación en pantalla de los tramos del lugar en el eje real para polos o ceros de multiplicidad mayor o igual que tres: está corregido, así como los puntos de ruptura para polos-ceros múltiples que también se ha corregido y comprobado en varios problemas. Sin embargo, pueden existir errores si hay un polo y un cero del mismo valor (esto no lo he comprobado totalmente pues no he visto problemas resueltos con esta peculiaridad).

REPRESENTACIÓN GRÁFICA DEL LUGAR

A la hora de dibujar el gráfico, se debe tener en cuenta que se pide Ki, Kf y el paso. El gráfico cambiará en magnitudes relativas de acuerdo a estos parámetros. Es bueno consultar para qué s se tiene una K determinada. Para ello, está la posibilidad de elegir esta opción, que de paso nos informa analíticamente de dicha correspondencia de valores. El gráfico no es perfecto, en el sentido de que si se quiere una buena visualización, que sea completa, etc., habrá que probar varios valores de s que conocemos porque tanto ceros como polos y puntos de ruptura los hemos calculado y serán los indicadores de las K más acertadas. Para introducir los valores de K es necesario que Ki>Kf y además: si es el lugar directo que ambos valores sean positivos o uno que sea 0, y si es en el lugar inverso, que ambos valores sean negativos o uno que sea 0.

MENÚS

El menú principal contiene 6 items y es el siguiente:



Menú principal.

Menú Introducción Datos.

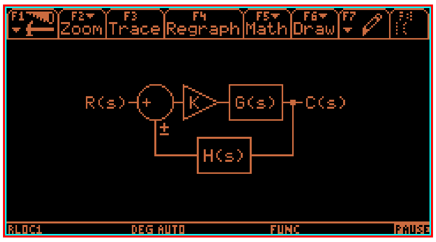
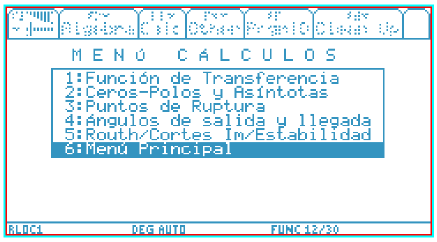


Diagrama Función de Transferencia.



Menú Cálculos.

1) INTRODUCCIÓN DE DATOS: permite introducir datos para numerador y denominador tanto de G(s) como H(s). Si nos dan la función como G(s)H(s) pueden ponerse tanto el numerador como el denominador de H(s) como 1. Si se trata de un problema de Lugar de las Raíces, K será simbólico (en el menú de datos viene preconfigurado K como simbólico y aparece ya, aunque puede cambiarse poniendo un número); si se trabaja con contorno de las raíces con un parámetro T, K tendrá un valor que se localiza en la expresión de G(s), siendo el valor numérico que acompaña al numerador, por lo que donde pone K se introduce dicho valor cambiando la K que viene.

Tanto la realimentación (positiva, negativa o ninguna) como el dominio forman parte de una selección en el diálogo que hay que elegir adecuadamente.

2) CÁLCULOS: Conduce al menú cálculos.

3) GRÁFICO LUGAR DE LAS RÁICES: grafica el lugar; se introducen Ki, Kf y paso; se puede optar por introducir como datos los tres parámetros directamente o bien ajustarlos mediante puntos s conocidos. Se puede graficar: el lugar con los polos y ceros; el lugar sin los polos ni ceros; sólo los polos y ceros sin el lugar.

4) DIAGRAMA FUNCIÓN DE TRANSFERENCIA: dibuja la función G(s) en realimentación con H(s) y la ganancia K.

5) BORRAR VARIABLES: borra las variables del problema.

6) ACERCA DE...: información del autor de Rlocus7 y el de Rlocus6.

Anterior	***4. Mejoras versión 7 frente a 6.***	Siguiente

Las características añadidas o mejoradas de la versión 7 respecto a la versión 6 y llevadas a cabo por mí son:

1. Permite **introducir el dato de las funciones de transferencia**, y no sólo de los ceros y polos de las mismas. De esta forma no hay que calcularlos previamente si no se conocen. Introducir polos y ceros en lugar de las funciones puede dar errores. Téngase en cuenta que las raíces de un polinomio pueden diferir en una constante, por ejemplo

$F(s) = s^2+2s+1 = (s+1)^2$ >> Raíces: s = -1 doble

$J(s) = 2s^2+4s+2 = 2.(s+1)^2$ >> Raíces: s = -1 doble (a pesar de tener un término multiplicativo)

Si pusiésemos en la entrada de datos los polos y ceros en lugar de las funciones, deberíamos añadir el término multiplicativo para que el programa interpretara la función J(s) correctamente. He mantenido esta entrada de datos por hacerla testimonial, ya que antes estaba así en la versión 6. Sin embargo es mucho mejor introducir las funciones tal cual vienen enunciadas.

2. Ahora no solo permite **cálculos para** K>0 sino también para **K<0**. Se han tenido que reprogramar ciertas partes.

3. Posibilidad de **ingresar parámetros (T)**, aparte de K, es decir, problemas de **contorno de raíces**. El programa calcula la función de transferencia real y la ficticia; es capaz de hacer el cálculo como se hace en un libro. Se ha realizado para que el programa sepa qué debe hacer y qué información debe mostrar, dando todos los cálculos intermedios en pantalla. Todo esto ha sido programado minuciosamente para hacer el cálculo secuencial.

4. Los **puntos de ruptura** no calculaban antes más que posibles máximos y mínimos globales que incluían además valores no posibles (como infinito), es decir los candidatos, sin especificar su naturaleza. Sin embargo, esto no es así pues los máximos y mínimos son locales en intervalos válidos de la recta real (aquellos en que existe desplazamiento de los polos). **Se dividen en puntos de dispersión y en puntos de confluencia** según la naturaleza máximo-mínimo del punto. Se da una información precisa de todo esto.

5. Se puede **establecer la estabilidad** del sistema analizando la tabla de Routh, **dando las desigualdades en que hay estabilidad para K o para el parámetro T** (opera también con contornos de raíces). Se calculan **puntos de corte con el eje imaginario** de dos formas a elegir. Antes esto no se hacía.

6. Todo el **proceso** es **"paso a paso"**. Antes no se daba ninguna información, salvo asíntotas, polos-ceros y ángulos de salidas-llegadas, donde había errores en la versión previa para polos-ceros múltiples (se han corregido).

7. Se muestran los **intervalos válidos en el eje real para el lugar**. Antes no se daban intervalos. Se aplican según si se trata de lugar directo o inverso.

8. El dibujo del lugar de las raíces no se ha tocado, que es preciso y animado, sin embargo se ha mejorado la introducción de datos para su dibujo, de tal forma que **es posible conocer para un valor entrado de s que K le corresponde**.

9. Se han corregido dos bugs en la función *azeros()* escrita por el autor indicado arriba (mejora la función del CAS de Texas Instruments zeros(), pues obtiene ceros múltiples). Uno de ellos era introducir en la función el cálculo con approx. Esto producía un error y generaba polos múltiples repetidos, por lo que lo suprimí. Otro era un excesivo tiempo de cálculo en determinados ceros de funciones. Añadí una salida con Return si la dimensión de los ceros calculados internamente con czeros() era igual a la dimensión del polinomio-función de transferencia, pues entonces la multiplicidad de todos los ceros estaba hecha.

10. Aparte de las funciones incorporadas de Chadd L. Easterday, se han introducido las funciones necesarias para trabajar con *isolve()* , solucionador de inecuaciones que mejora el C.A.S (Computer Algebraic System) de la TI, extraídas del programa de rutinas y funciones Matemáticas MathTools de **Bhuvanesh Bhatt:**

(página web: http://triton.towson.edu/users/bbhatt1/ti/ ,

correo electrónico: bbhatt1@towson.edu).

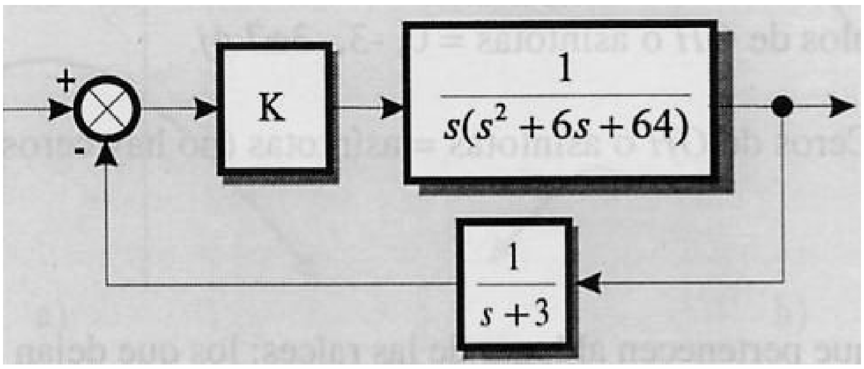
En concreto dichas funciones son: *delelem(), interval(), iscmplxn(), isolve(), listswap(), memberq(), mor(), replcstr(), rmdup(), sort() y varlist().*

Anterior	**5.Problemas resueltos con Root Locus 7.**	Siguiente

"Control de sistemas continuos. Problemas Resueltos", de Antonio Barrientos y otros autores, Mac Graw Hill.

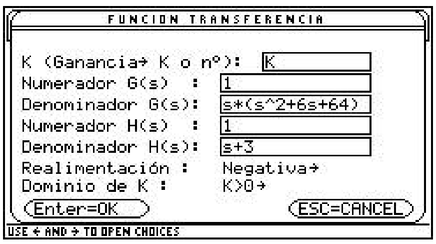
Anterior	**Problema 8.1 Lugar de Raíces.**	Siguiente Problema

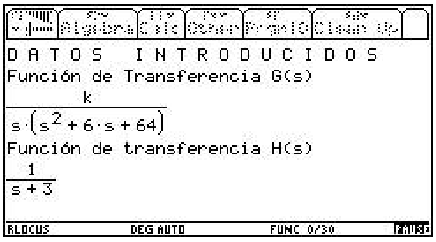
Dibujar la evolución de los polos del sistema para *K>0* y *K<0* comentando el comportamiento dinámico del sistema al variar *K.*

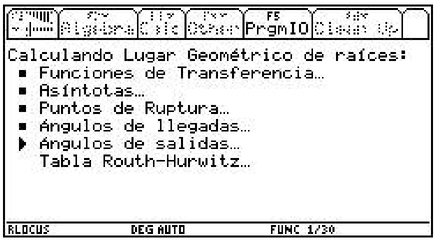


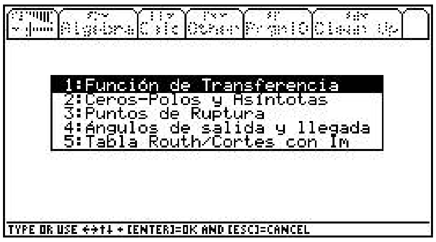
Las funciones de transferencia se observan en las pantallas, por lo que no se repiten en el enunciado. Al tratarse de un problema del tipo "Lugar de las Raíces", K es incógnita y lo que se busca fundamentalmente son los intervalos para alcanzar el sistema estabilidad para un diseño dado.

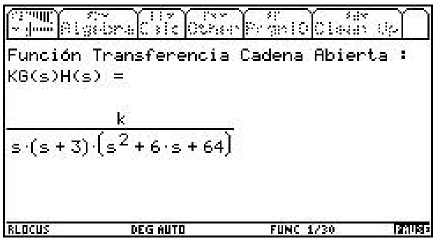
Dominio K>0

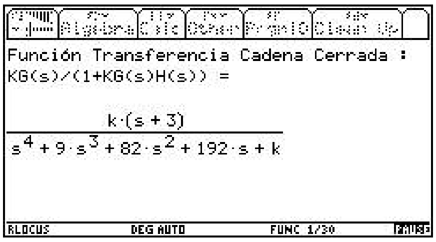


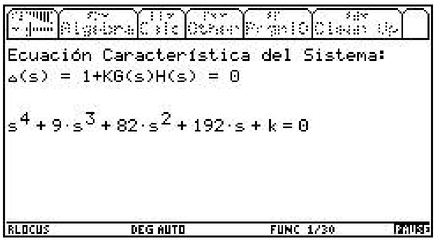


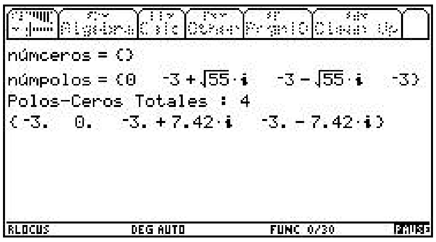


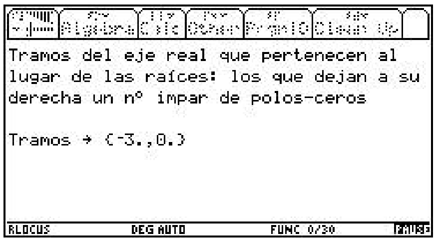




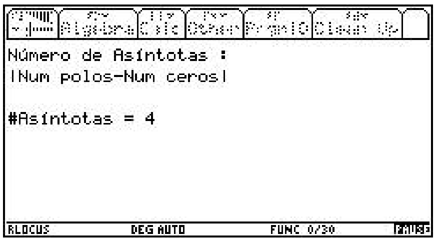


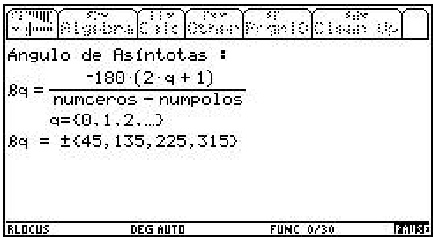


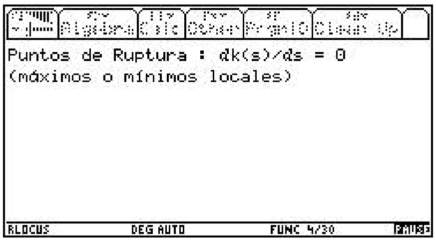


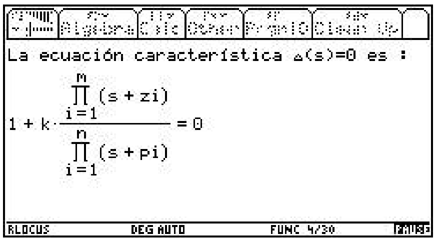


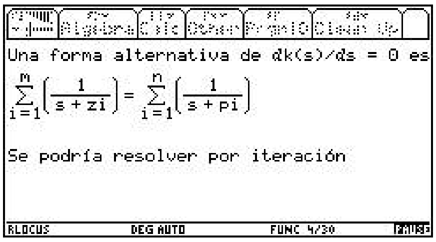
El programa sabe que es el tramo que deja a su derecha un nº impar porque se introdujo K>0.

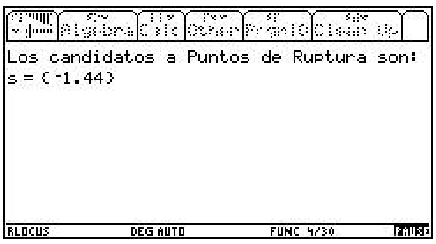




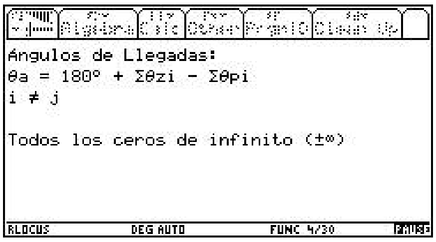


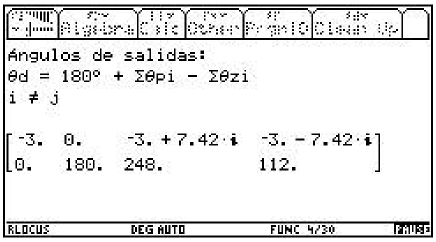


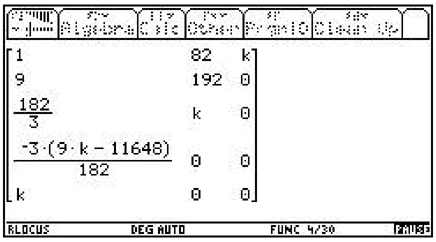


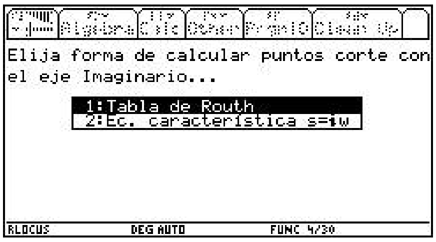


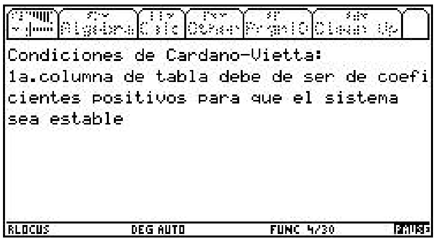
Se han indicado algunas formas de calcular los puntos de ruptura. Primero se hallan los candidatos. Luego se comprueba si lo son, de qué tipo y porqué y el valor de K en dicho punto.

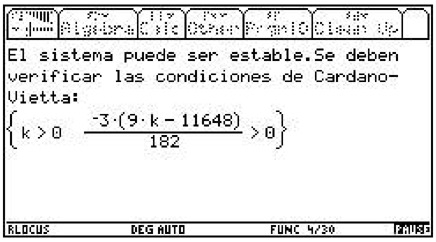


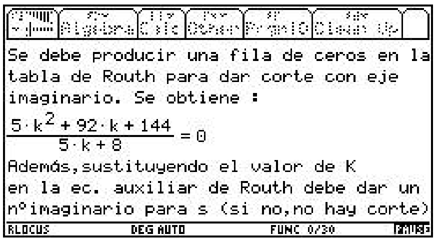


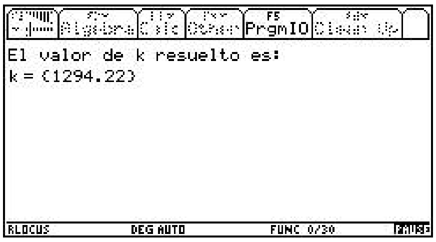


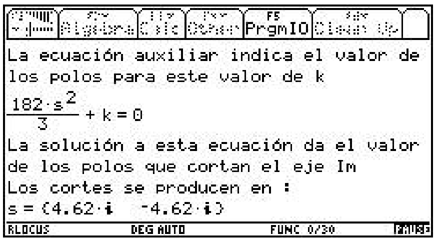


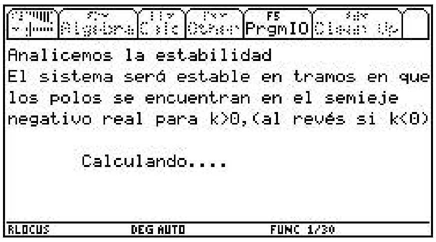


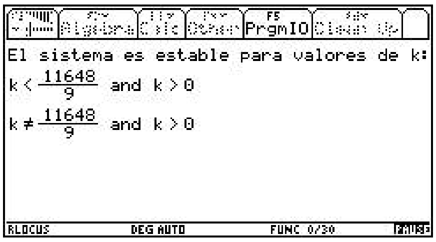




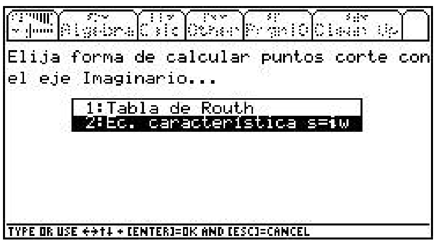


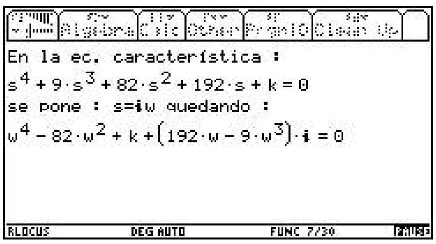


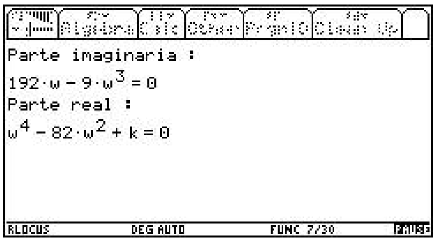


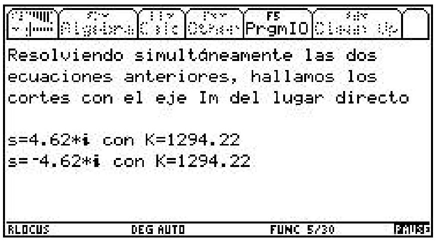


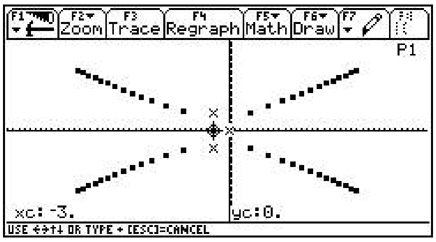
A partir de 11648/9=1294.2 , el sistema no será estable. La última desigualdad sobra, sin embargo la función isolve() opera así, cuando resuelve 2 desigualdades completamente calculadas. He descubierto este bug en esta función y me he puesto en contacto con el autor de isolve para ver si puede arreglarse. En concreto falla para valores de cálculo entre dos desigualdades en las que un valor numérico sea mayor o igual que 1000, cuando se introducen desigualdades como la anterior totalmente expandidas.

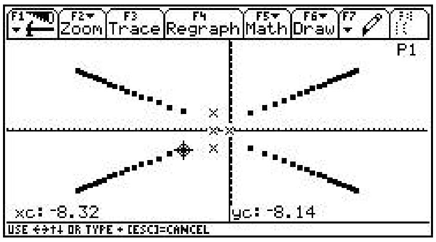




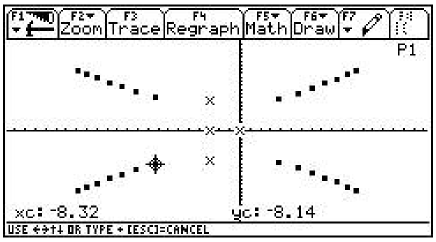


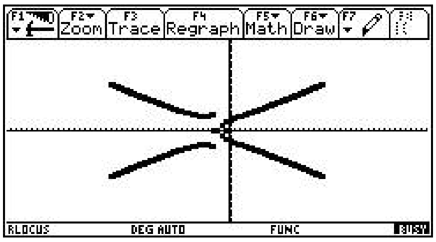






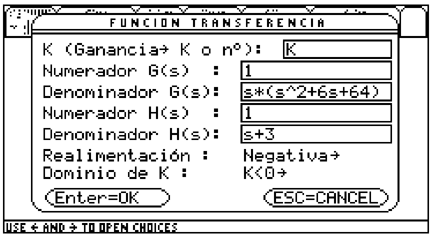
Para dibujar la gráfica hay que definir valores de K y de escala de paso. El lugar de arriba a la izquierda parece muy alejado, se han definido un intervalo de escala muy grande para que el proceso sea más rápido.

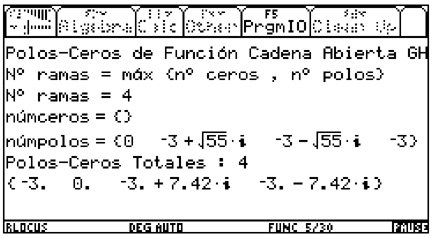


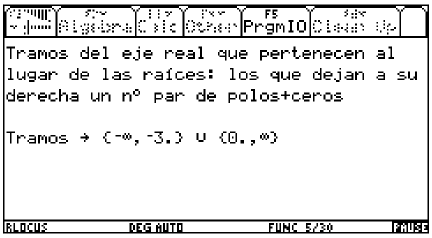


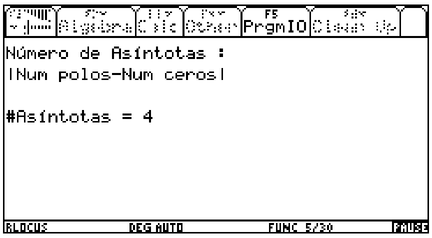
Finalmente, el último dibujo presenta un mejor aspecto. Además, la gráfica está animada y los puntos se mueven según si son ceros o polos en la dirección adecuada. Esto no se aprecia en una imagen congelada.

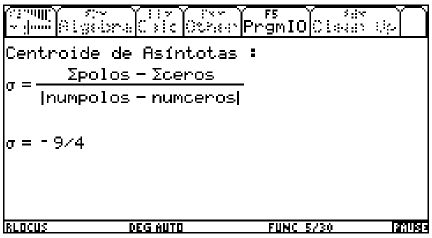
Dominio K<0

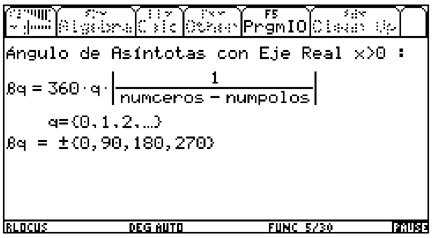


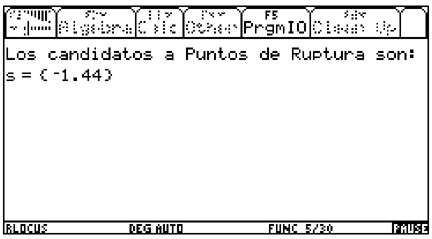


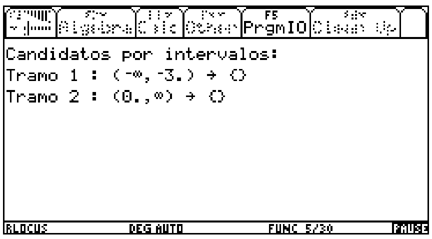












No hay puntos de dispersión/confluencia

Ángulos de Llegadas (Ceros):
θa = 360° + Σθzi − Σθpi

Todos los ceros de infinito (±∞)

Ángulos de salidas (Polos):
θd = 360° + Σθpi − Σθzi

$$\begin{bmatrix} \text{"s"} & -3. & 0. & -3.+7.42\cdot i & -3.-7.42\cdot i \\ \text{"θ°"} & 180. & 0. & 68. & 292. \end{bmatrix}$$

En la ec. característica :
$s^4 + 9\cdot s^3 + 82\cdot s^2 + 192\cdot s + k = 0$
se pone : s=iw quedando :
$w^4 - 82\cdot w^2 + k + (192\cdot w - 9\cdot w^3)\cdot i = 0$

Parte imaginaria :
$192\cdot w - 9\cdot w^3 = 0$
Parte real :
$w^4 - 82\cdot w^2 + k = 0$

Resolviendo simultáneamente las dos ecuaciones anteriores, hallamos los cortes con el eje Im del lugar inverso

No hay corte con el eje imaginario.
ya que w ∈ Im → s ∈ Re

Comprobemos si en la fila 4 de Routh
$$\begin{bmatrix} \frac{-3\cdot(9\cdot k - 11648)}{182} & 0 & 0 \end{bmatrix}$$
podemos obtener el valor de corte con Im
$$\frac{-3\cdot(9\cdot k - 11648)}{182} = 0$$
obteniendo fila de ceros y valores de s imaginarios

Se produce fila de ceros en fila : 4
El valor de k resuelto es: k=1294.22
Este valor de K corresponde al lugar directo (no vale)

Comprobemos si en la fila 5 de Routh
$[k \;\; 0 \;\; 0]$
podemos obtener el valor de corte con Im
$k = 0$
obteniendo fila de ceros y valores de s imaginarios

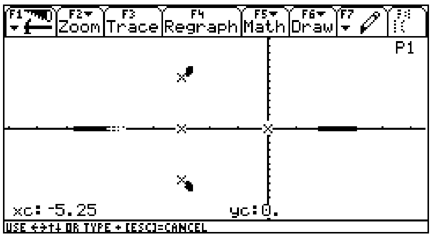
Se produce fila de ceros en fila : 5
El valor de k resuelto es: k=0.00
La ecuación auxiliar indica el valor de los polos para el valor k=0.
$$\frac{-3\cdot(9\cdot k - 11648)\cdot s}{182} = 0$$
$192.\cdot s = 0$

La solución a esta ecuación da el valor de los polos que cortan al eje Im si son valores imaginarios (ó s=0)
Valor k=0.
Los cortes se producen en:
s = {0}

(2) Condiciones de Routh
La 1a. columna de Routh debe de ser de coeficientes positivos para ser estable
El sistema puede ser estable si:
$k < \frac{11648}{9}$ and $k > 0$
Resolviendo queda:
$k \neq 1.29E3$ and $k > 0.$

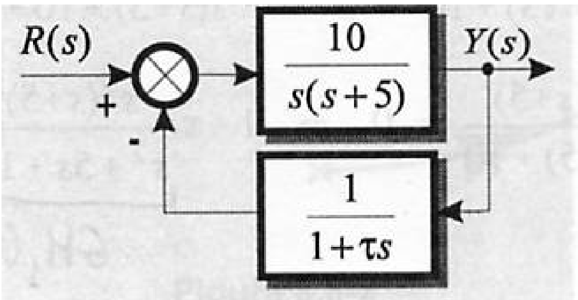
Intervalos de Estabilidad
Conjuntando las condiciones de Cardano-Vietta junto con la condición de la 1a. columna de la tabla de Routh da:

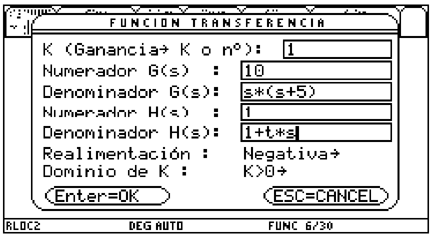
$k \neq 1.29E3$ and $k > 0.$



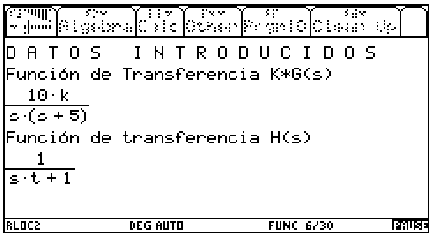
Problema 8.6 Contorno de Raíces.

Estudiar el efecto que tiene la variación del parámetro t para t>0.

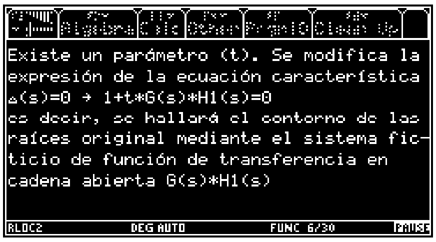




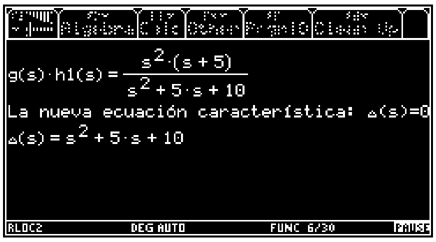
FUNCION TRANSFERENCIA
K (Ganancia→ K o n°): 1
Numerador G(s) : 10
Denominador G(s): s*(s+5)
Numerador H(s) : 1
Denominador H(s): 1+t*s
Realimentación : Negativa→
Dominio de K : K>0→
Enter=OK ESC=CANCEL



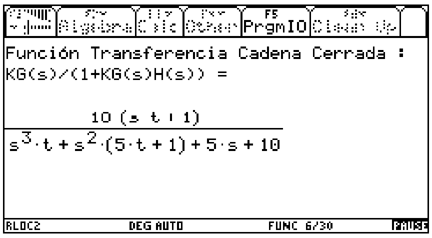
D A T O S I N T R O D U C I D O S
Función de Transferencia K*G(s)
$\frac{10\cdot k}{s\cdot(s+5)}$
Función de transferencia H(s)
$\frac{1}{s\cdot t+1}$



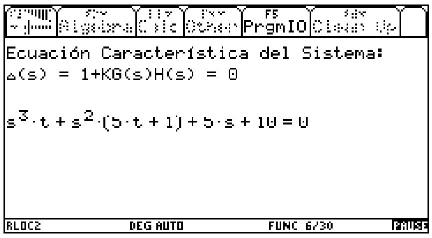
Existe un parámetro (t). Se modifica la expresión de la ecuación característica ⊿(s)=0 → 1+t*G(s)*H1(s)=0 es decir, se hallará el contorno de las raíces original mediante el sistema ficticio de función de transferencia en cadena abierta G(s)*H1(s)



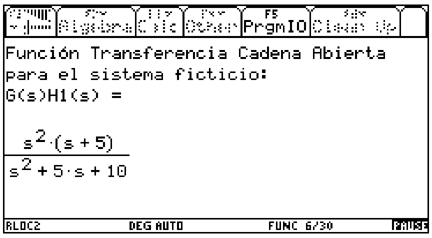
$g(s)\cdot h1(s)=\frac{s^2\cdot(s+5)}{s^2+5\cdot s+10}$
La nueva ecuación característica: ⊿(s)=0
$⊿(s)=s^2+5\cdot s+10$



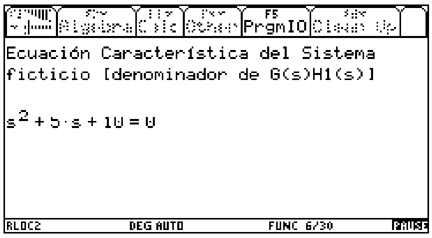
Función Transferencia Cadena Cerrada :
KG(s)/(1+KG(s)H(s)) =
$\frac{10\,(s\cdot t+1)}{s^3\cdot t+s^2\cdot(5\cdot t+1)+5\cdot s+10}$



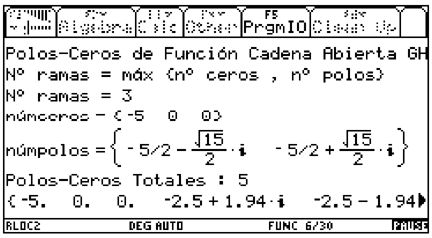
Ecuación Característica del Sistema:
⊿(s) = 1+KG(s)H(s) = 0
$s^3\cdot t+s^2\cdot(5\cdot t+1)+5\cdot s+10=0$



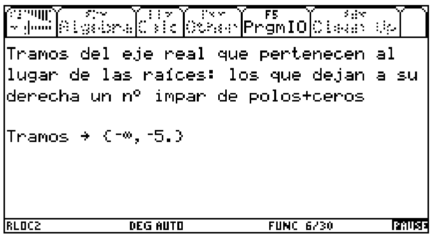
Función Transferencia Cadena Abierta para el sistema ficticio:
G(s)H1(s) =
$\frac{s^2\cdot(s+5)}{s^2+5\cdot s+10}$



Ecuación Característica del Sistema ficticio [denominador de G(s)H1(s)]
$s^2+5\cdot s+10=0$

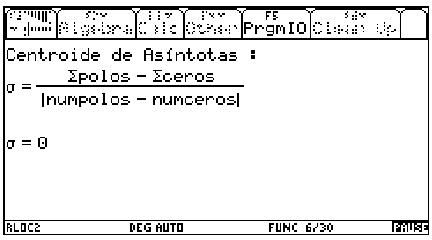


Polos-Ceros de Función Cadena Abierta GH
N° ramas = máx (n° ceros , n° polos)
N° ramas = 3
númceros = {-5 0 0}
$\text{númpolos}=\left\{-5/2-\frac{\sqrt{15}}{2}\cdot i \quad -5/2+\frac{\sqrt{15}}{2}\cdot i\right\}$
Polos-Ceros Totales : 5
{-5. 0. 0. -2.5+1.94·i -2.5-1.94

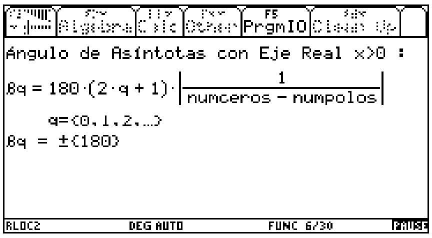


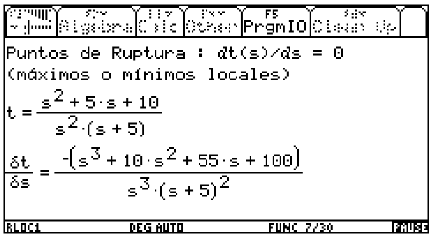
Tramos del eje real que pertenecen al lugar de las raíces: los que dejan a su derecha un n° impar de polos+ceros
Tramos → (-∞, -5.)

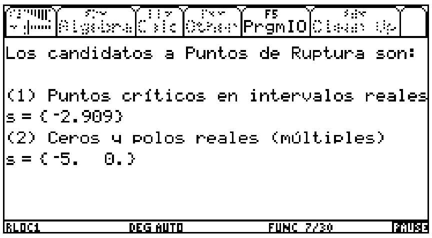
Número de Asíntotas :
|Num polos-Num ceros|
#Asíntotas = 1

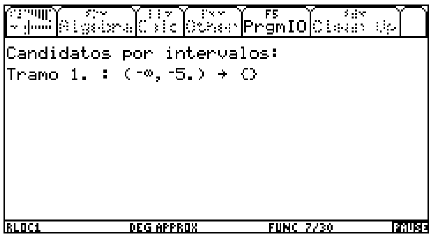


Centroide de Asíntotas :
$\sigma=\frac{\Sigma\text{polos}-\Sigma\text{ceros}}{|\text{numpolos}-\text{numceros}|}$
$\sigma=0$



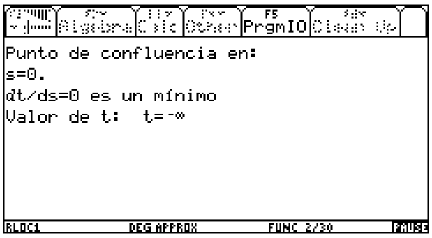


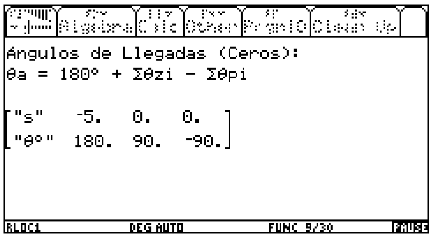


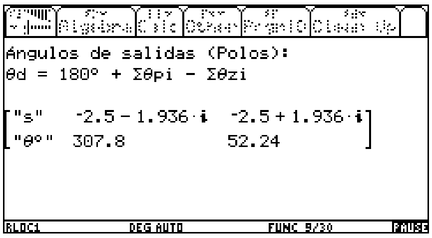


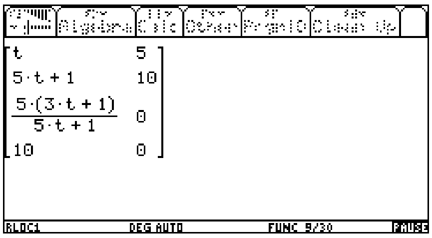
Se dividen los candidatos a puntos de ruptura en dos tipos: (1) los que están en intervalos reales y (2) los ceros y polos reales. En el caso (2) 0 es cero doble. Realmente aunque se listan todos los ceros y polos reales, los candidatos a punto de ruptura dentro de éstos serán únicamente los múltiples, pues el caso (1) recoge también los que no son múltiples. No se ha dividido esto por una razón especial, únicamente es por la programación para calcularlos, que emplea métodos distintos, dentro de las rutinas empleadas.

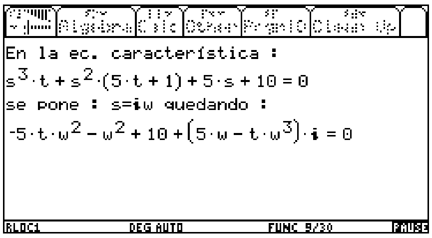
Si nos damos cuenta, al partir de la expresión para t y derivar, da igual que la expresión vaya precedida de un (-) o no; nada más hay que observar los puntos críticos y se puede ver que es indiferente. Sin embargo, a la hora de realizar el cálculo para t en la expresión con el valor de s que es punto de ruptura, como $1+t.GH_1=0$, habría que proceder a poner dicho signo (-), y es por ello que t sale -infinito, al sustituir en la función. Los problemas del libro se resuelven usando el criterio del módulo y convirtiendo la expresión antedicha (en función de t o K), según la distancia medida de cierto punto a los ceros y polos, por lo que los sumandos en el numerador y denominador sufren ciertas transformaciones. Este criterio es universal y se deduce de las ecuaciones, al igual que el criterio del argumento. Sin embargo, a la hora de efectuar una programación sistemática, es preferible partir de la ecuación de partida en la pantalla primera de los puntos de ruptura, tomar la derivada y hallar los puntos críticos; luego verificar si es máximo o mínimo bastando comprobar que en un intervalo pequeño la función en dicho punto es máxima o mínima, tomando cierto valor arriba y abajo y verificando dichas desigualdades. Este cálculo se completa con todos aquellos polos+ceros reales múltiples. Para este caso es preciso que la función lleve el signo (-). Esto lo he comprobado a la hora de hacer ejercicios de este tipo, con ceros, polos múltiples.

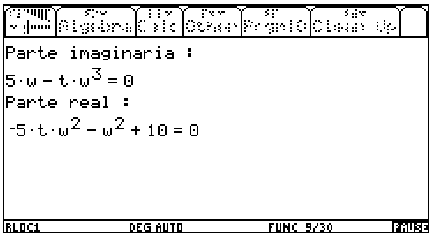


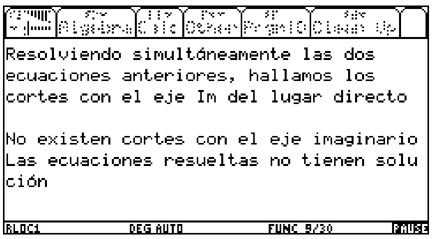


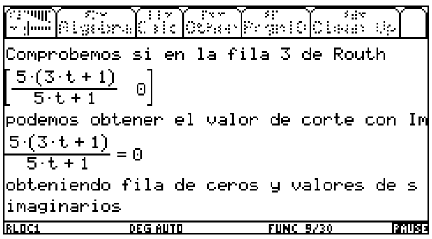


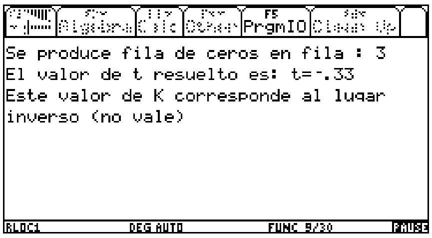


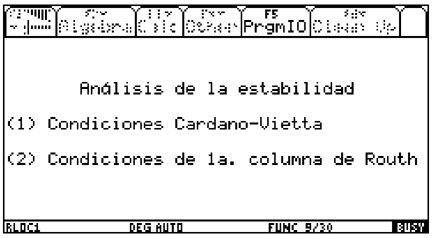


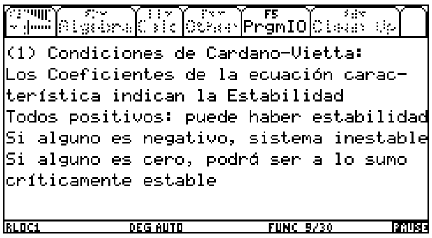


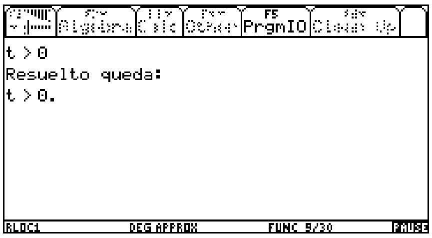


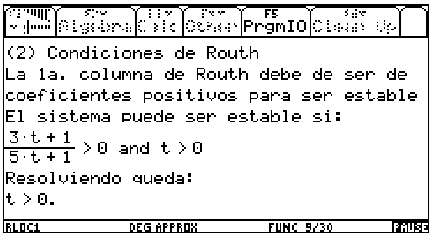


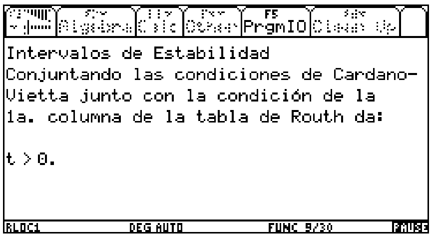


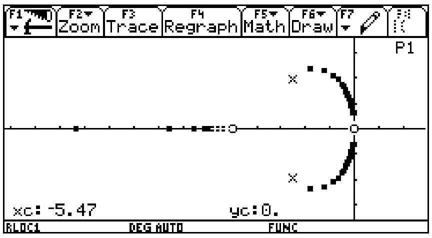










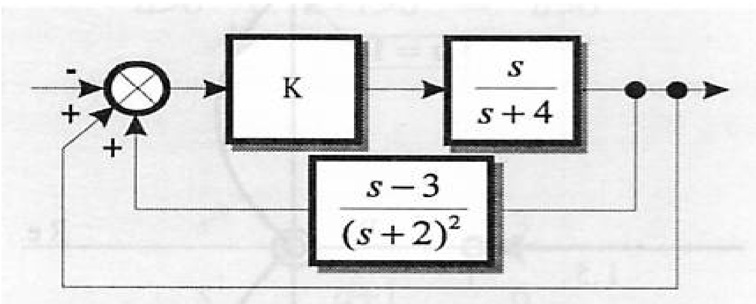


[Anterior Problema]

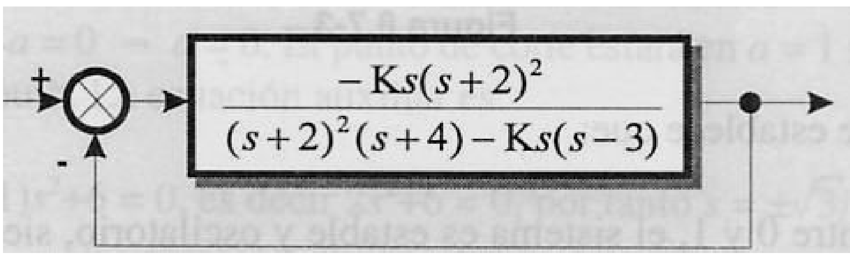
Problema 8.8 Contorno de Raíces.

[Siguiente]

Estudiar el comportamiento dinámico del sistema para *K>0* y *K<0* .



El planteamiento del problema en el libro realiza una realimentación negativa poniendo un signo (-) y quedando otra realimentación negativa que se ha cambiado (+) gracias al signo del numerador. Aquí se pondrá un sistema con realimentación (+), y luego el lazo externo sin función de transferencia también será de realimentación (+). El resultado es el mismo, como se verá. Sólo cambia el signo del numerador y la realimentación externa.



FUNCION TRANSFERENCIA

K (Ganancia→ K o nº): K

Numerador G(s) : s*(s+2)^2

Denominador G(s): (s+2)^2*(s+4)-

Numerador H(s) : 1

Denominador H(s): 1

Realimentación : Positiva→

Dominio de K : K>0→

Enter=OK ESC=CANCEL

USE ← AND → TO OPEN CHOICES

Existe un parámetro (k). Se modifica la expresión de la ecuación característica
△(s)=0 → 1-k*G(s)*H1(s)=0
es decir, se hallará el contorno de las raíces original mediante el sistema ficticio de función de transferencia en cadena abierta G(s)*H1(s)

Sin embargo, la expresión debería ser:
△(s)=0 → 1+k*G(s)*H1(s)=0
con un (+) en vez de un (-)
Haciendo el cambio de variable:
k'=-k, el estudio dinámico del sistema para 0<k<∞ del lugar directo se invierte

---

Entonces G(s)H1(s) cambia a G'(s)H1(s):

$$g'(s)\cdot h1(s)=\frac{-s\cdot\left(s^{2}+5\cdot s+1\right)}{(s+2)^{2}\cdot(s+4)}$$

La nueva ecuación característica:△'(s)=0

$$△'(s)=s^{3}+8\cdot s^{2}+20\cdot s+16$$

NOTAS: En adelante las variables con ' (prima) no figurarán así
Al cambiar -K por K', se usará la ecuación característica del sistema original para hacer la tabla de Routh tomando ese cambio

---

Función Transferencia Cadena Abierta :
KG(s)H(s) =

$$\frac{s\cdot\left(s^{2}+5\cdot s+1\right)}{s^{3}+8\cdot s^{2}+20\cdot s+16}$$

Función Transferencia Cadena Cerrada :
KG(s)/(1-KG(s)H(s)) =

$$\frac{-k\cdot s\cdot(s+2)^{2}}{(k-1)\cdot s^{3}+(5\cdot k-8)\cdot s^{2}+(k-20)\cdot s-16}$$

Ecuación Característica del Sistema:
△(s) = 1-KG(s)H(s) = 0

$$(k+1)\cdot s^{3}+(5\cdot k+8)\cdot s^{2}+(k+20)\cdot s+16=0$$

Función Transferencia Cadena Abierta para el sistema ficticio:
G(s)H1(s) =

$$\frac{s\cdot\left(s^{2}+5\cdot s+1\right)}{s^{3}+8\cdot s^{2}+20\cdot s+16}$$

Ecuación Característica del Sistema ficticio [denominador de G(s)H1(s)]

$$s^{3}+8\cdot s^{2}+20\cdot s+16=0$$

Polos-Ceros de Función Cadena Abierta GH
Nº ramas = máx (nº ceros , nº polos)
Nº ramas = 3

$$\text{númceros}=\left\{0 \quad \frac{\sqrt{21}-5}{2} \quad \frac{-\left(\sqrt{21}+5\right)}{2}\right\}$$

númpolos = {-4 -2 -2}
Polos-Ceros Totales : 6
{-4.791 -4. -2. -2. -.2087 0.}

Tramos del eje real que pertenecen al lugar de las raíces: los que dejan a su derecha un nº par de polos+ceros

Tramos → (-∞,-4.791) ∪ (-4.,-.2087) ∪ (

Tramos del eje real que pertenecen al lugar de las raíces: los que dejan a su derecha un nº par de polos+ceros

s → (-∞,-4.791) ∪ (-4.,-.2087) ∪ (0.,∞)

Número de Asíntotas :
|Num polos-Num ceros|

#Asíntotas = 0

Centroide de Asíntotas :

$$\sigma=\frac{\Sigma polos-\Sigma ceros}{|numpolos-numceros|}$$

σ = -∞

Ángulo de Asíntotas con Eje Real x>0 :

$$\theta q=360\cdot q\cdot\left|\frac{1}{numceros-numpolos}\right|$$

q={0,1,2,...}
θq = ±()

Puntos de Ruptura : dk(s)/ds = 0
(máximos o mínimos locales)

$$k=\frac{s^3+8\cdot s^2+20\cdot s+16}{s\cdot(s^2+5\cdot s+1)}$$

Los candidatos a Puntos de Ruptura son:

(1) Puntos críticos en intervalos reales
s = {-7.199 -3.357 -2. -.1103}
(2) Ceros y polos reales (múltiples)
s = {0. -.2087 -4.791}

Candidatos por intervalos:
Tramo 1 : (-∞,-4.791) → {-7.199}
Tramo 2 : (-4.,-.2087) → {-3.357,-2.}
Tramo 3 : (0.,∞) → {}

Punto de confluencia en:
s=-7.199
dk/ds=0 es un mínimo
en el tramo : (-∞,-4.791)
Valor de k: k=-.7136

Punto de dispersión en:
s=-3.357
dk/ds=0 es un máximo
en el tramo : (-4.,-.2087)
Valor de k: k=-.0781

Ángulos de Llegadas (Ceros):
θa = 360° + Σθzi - Σθpi

$$\begin{bmatrix} \text{"s"} & 0. & -.2087 & -4.791 \\ \text{"}\theta°\text{"} & 0. & 180. & 180. \end{bmatrix}$$

En la ec. característica :

$$(k+1)\cdot s^3+(5\cdot k+8)\cdot s^2+(k+20)\cdot s+16=0$$

se pone : s=iw quedando :

$$16-(5\cdot k+8)\cdot w^2+\left((k+20)\cdot w-(k+1)\cdot w^3\right)\cdot i$$

Parte imaginaria :

$$(k+20)\cdot w-(k+1)\cdot w^3=0$$

Parte real :

$$16-(5\cdot k+8)\cdot w^2=0$$

Resolviendo simultáneamente las dos ecuaciones anteriores, hallamos los cortes con el eje Im del lugar inverso

No existen cortes con el eje imaginario
Las ecuaciones resueltas no tienen solución

Comprobemos si en la fila 3 de Routh

$$\begin{bmatrix} \frac{5\cdot k^2+92\cdot k+144}{5\cdot k+8} & 0 \end{bmatrix}$$

podemos obtener el valor de corte con Im

$$\frac{5\cdot k^2+92\cdot k+144}{5\cdot k+8}=0$$

obteniendo fila de ceros y valores de s imaginarios

Se produce fila de ceros en fila : 3
El valor de k resuelto es: k=-1.73
La ecuación auxiliar indica el valor de los polos para el valor k=-1.727

$$(5\cdot k+8)\cdot s^2+16=0$$

$$16-.6369\cdot s^2=0$$

La solución a esta ecuación da el valor de los polos que cortan al eje Im si son valores imaginarios (ó s=0)
Valor k=-1.727
No hay cortes, soluciones de s reales:
s = {-5.012 5.012}

Comprobemos si en la fila 3 de Routh

$$\begin{bmatrix} \frac{5\cdot k^2+92\cdot k+144}{5\cdot k+8} & 0 \end{bmatrix}$$

podemos obtener el valor de corte con Im

$$\frac{5\cdot k^2+92\cdot k+144}{5\cdot k+8}=0$$

obteniendo fila de ceros y valores de s imaginarios

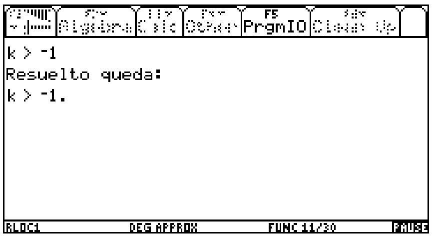
Se produce fila de ceros en fila : 3
El valor de k resuelto es: k=-16.67
La ecuación auxiliar indica el valor de los polos para el valor k=-16.67

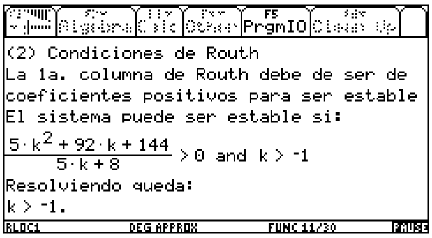
$$(5\cdot k+8)\cdot s^2+16=0$$

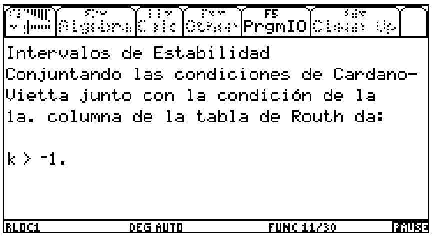
$$16-75.36\cdot s^2=0$$

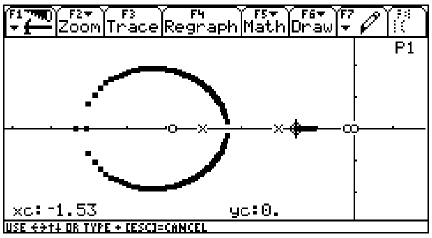
La solución a esta ecuación da el valor de los polos que cortan al eje Im si son valores imaginarios (ó s=0)
Valor k=-16.67
No hay cortes, soluciones de s reales:
s = {-.4608 .4608}

(1) Condiciones de Cardano-Vietta:
Los Coeficientes de la ecuación característica indican la Estabilidad
Todos positivos: puede haber estabilidad
Si alguno es negativo, sistema inestable
Si alguno es cero, podrá ser a lo sumo críticamente estable









Anterior	6. El autor de Root Locus 7.	Ir al Principio

Soy estudiante de la Escuela Técnica Superior de Ingenieros Industriales, especialidad Mecánica de Máquinas por la U.N.E.D (Universidad Nacional de Educación a Distancia), universidad a distancia española. Simultaneo estudios con trabajo (tengo 35 años) y desde hace casi tres años descubrí esta calculadora. Llevo más de un año de experiencia como programador, partiendo de cero, aunque ya conocía la estructura del Basic. No he podido hasta ahora publicar ningún programa, debido a la falta de tiempo, sobre todo, para hacer manuales.

Estoy *enfrascado* en varios proyectos importantes en el área de Cálculo de Estructuras que me absorben la mayor parte del tiempo que dedico a la programación. Me gustaría que este programa (Rlocus7) fuera útil y libre de errores (bugs). Agradezco a **Chadd L. Easterday** su idea al crear Rlocus 6. Espero haber mejorado este programa pues quizás le faltaban algunas cosas que yo he tratado de añadir para completar el estudio del lugar de las raíces. Igualmente agradezco al magnífico **Bhuvanesh Bhatt** por sus aportaciones a la comunidad TI, y por sus funciones del Mathtool para calcular desigualdades que mejoran el sistema operativo de la calculadora para este tipo de operaciones.

Cualquier error del programa, sugerencia o comentario, no dudes en planteármelo en:

gomezvega@hotmail.com

Visita la página: http://members.fortunecity.es/etsii/ para otros recursos de Ingeniería Industrial y programas de la Texas Instruments 92 plus y Voyage 200.