# Tutorial del Lugar Geométrico de las Raíces

**Asignatura**:

# Teoría de Control

**Nivel**: 4º Año
**Carrera**: Ingeniería en Sistemas de Información

Esta guía contiene una explicación paso a paso de los comandos a ingresar tanto en el software **MATLAB** como en el software **SCILAB**, utilizados para resolver ejercicios prácticos del Lugar Geométrico de las Raíces.

**Docente:** Ing. Dominga Concepción Aquino
**Autor:** Daniel Eduardo Trnovsky

## MatLab y SciLab

# Lugar Geométrico de las Raíces utilizando MATLAB

En esta guía vamos a desarrollar un ejemplo de cálculo del Lugar Geométrico de las Raíces, y mediante el software MATLAB comprobaremos partes de las resoluciones.

Dada la siguiente ecuación característica del sistema:

$$1 + k\frac{s + 4{,}5}{s^3 + 6s^2 + 45s} = 0$$

**a) Cálculo de polos y ceros:**

Ingresamos los coeficientes de las ecuaciones del numerador y del denominador de la función de transferencia de lazo abierto en forma de vectores (ambos términos deben estar completos y expresados en potencias descendentes).

```
>> num=[1 4.5];
>> den=[1 6 45 0];
```

Luego procedemos a calcular los polos y ceros de la ecuación mediante el comando **roots**.

```
>> ceros=roots(num)

ceros =

   -4.5000

>> polos=roots(den)

polos =

   0.0000 + 0.0000i
  -3.0000 + 6.0000i
  -3.0000 - 6.0000i
```

A partir de estos comandos determinamos que el cero de la ecuación se encuentra en el punto -4,5 y los polos se encuentran en los puntos 0, -3+6j y -3-6j.

También es posible realizar el proceso inverso con **MATLAB**, es decir, obtener la función de transferencia de lazo abierto a partir de los polos y ceros.

Se utiliza el comando **zp2tf** y pasamos como parámetros un vector con los ceros, otro vector con los polos, y **k** que en este caso va a tomar el valor 1. Es importante recordar que el vector que contiene los ceros debe ser traspuesto colocando una comilla al final, luego del corchete de cierre.

```
>> n=[-4.5]'

n =

   -4.5000

>> d=[0 -3+6i -3-6i]

d =

   0.0000 + 0.0000i  -3.0000 + 6.0000i  -3.0000 - 6.0000i

>> [N,D]=zp2tf(n,d,1)

N =

         0         0    1.0000    4.5000


D =

     1     6    45     0
```

Finalmente para poder ver la función de transferencia de forma más amigable, se utiliza la función **tf** como se muestra a continuación:

```
>> sys=tf(N,D)

sys =

       s + 4.5
  ------------------
  s^3 + 6 s^2 + 45 s

Continuous-time transfer function.
```

**b) Punto donde nacen las asíntotas**

$$\sigma = \frac{\sum Polos - \sum Ceros}{n - m} = \frac{\big(0 + (-3 + 6j) + (-3 - 6j)\big) - (-4{,}5)}{3 - 1} = \frac{-6 + 4{,}5}{2} \quad => \sigma = -0{,}75$$

Para realizar este cálculo con **MATLAB**, primero creamos una función a la cual posteriormente llamamos enviando como parámetros el número de polos (**n**), el número de ceros (**m**) y dos vectores con los valores de los polos (**p**) y los ceros (**c**).

```
function [punto] = punto_donde_nacen_asintotas(n,m,p,c)
        o=1;
        q=1;
        sumapolos=0;
        sumaceros=0;
        while o <= n
            sumapolos=sumapolos+p(o);
            o=o+1;
        end
        while q <= m
            sumaceros=sumaceros+c(q);
            q=q+1;
        end
        punto = (sumapolos-sumaceros)/(n-m);
end
```

```
>> punto=punto_donde_nacen_asintotas(3,1,[0 -3+6i -3-6i],[-4.5])

punto =

   -0.7500
```

Vemos que el punto donde nacen las asíntotas es igual a -0,75.

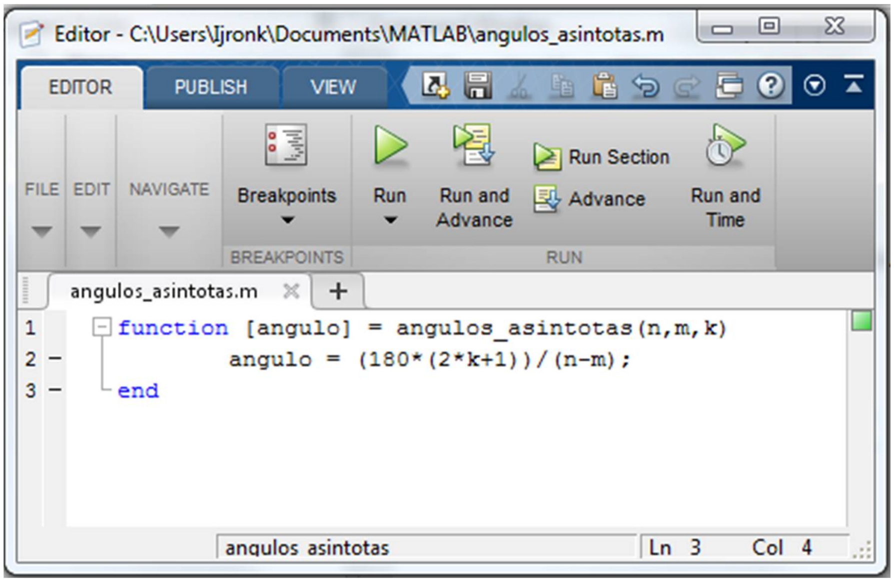
**c) Ángulos de las asíntotas**

$$\alpha = \frac{180^o(2k+1)}{n-m}$$

$$k = 0 \quad \alpha_1 = \frac{180^o(2*0+1)}{3-1} = 90^o$$

$$k = 1 \quad \alpha_2 = \frac{180^o(2*1+1)}{3-1} = 270$$

Para realizar la comprobación primero creamos la siguiente función en **MATLAB**:




Luego llamamos a la función anteriormente creada colocando como parámetros el número de polos (**n**), número de ceros (**m**) y el valor del contador **k**.

```
>> angulo=angulos_asintotas(3,1,0)

angulo =

    90

>> angulo2=angulos_asintotas(3,1,1)

angulo2 =

   270
```

Con el resultado obtenido vemos que **90º** y **270º** son los ángulos de las asíntotas.

**d) Puntos de Ruptura**

$$1+\frac{k(s+4{,}5)}{s^3+6s^2+45s}=0$$

$$k=-\frac{s^3+6s^2+45s}{s+4{,}5}$$

$$k=\frac{-s^3-6s^2-45s}{s+4{,}5}$$

$$\frac{\delta k}{\delta s}=0=>\frac{(s+4{,}5)*(-3s^2-12s-45)-(-s^3-6s^2-45s)*1}{(s+4{,}5)^2}=0$$

$$-3s^3-12s^2-45s-13{,}5s^2-54s-202{,}5+s^3+6s^2+45s=0$$

$$-2s^3-19{,}5s^2-54s-202{,}5=0$$

$$s_1=-7{,}96$$

$$s_2=-0{,}897+3{,}45j$$

$$s_3=-0{,}897-3{,}45j$$

En este ejercicio en particular, no existen puntos de ruptura, porque ninguna de las raíces del polinomio pertenece al Lugar Geométrico de las Raíces.

Con **MATLAB** tenemos la posibilidad de calcular la derivada de k con respecto a **s** de la siguiente manera:

- ✓ Guardamos la función igualada a k en una variable.

```
>> k=(-s^3-6*s^2-45*s)/(s+4.5)

k =

-(s^3 + 6*s^2 + 45*s)/(s + 9/2)
```

- ✓ Derivamos la función.

```
>> derivada=diff(k)

derivada =

(s^3 + 6*s^2 + 45*s)/(s + 9/2)^2 - (3*s^2 + 12*s + 45)/(s + 9/2)
```

- ✓ Simplificamos la expresión obtenida anteriormente.

```
>> d=simplify(derivada)

d =

-(2*(4*s^3 + 39*s^2 + 108*s + 405))/(2*s + 9)^2
```

- ✓ Transformamos el resultado de la derivación a una forma más amigable.

```
>> pretty(d)

            3        2
    2 (4 s  + 39 s  + 108 s + 405)
  - ------------------------------
                          2
                 (2 s + 9)
```

- ✓ Luego hay que hallar las raíces del numerador para determinar si son o no puntos de ruptura.

Otra forma de calcular con **MATLAB** la derivada de k con respecto a **s** es de la siguiente forma:

- ✓ Una vez que despejamos k, guardamos los coeficientes del numerador en un vector, en orden descendente y completo, y hacemos lo mismo con los coeficientes del denominador en otro vector.

```
nsd =

    -1    -6   -45     0

>> dsd=[1 4.5]

dsd =

    1.0000    4.5000

>> [dn,dd]=polyder(nsd,dsd)
```

- ✓ Como resultado de aplicar la función **polyder** se obtiene lo siguiente:

```
dn =

   -2.0000  -19.5000  -54.0000 -202.5000


dd =

    1.0000    9.0000   20.2500
```

Donde el vector **dn** contiene los coeficientes del polinomio numerador de la derivada del cociente, y el vector **dd** contiene los coeficientes del polinomio denominador, el cual no interesa debido a que $\frac{\delta k}{\delta s} = 0$

- ✓ Transformamos el resultado de la derivación a una forma más amigable.

```
>> sys=tf(dn,dd)

Transfer function:
-2 s^3 - 19.5 s^2 - 54 s - 202.5
--------------------------------
       s^2 + 9 s + 20.25
```

- ✓ A continuación se hallan las raíces del polinomio numerador para obtener los valores de los posibles puntos de ruptura.

```
>> r = roots(dn)

r =

  -7.9559
  -0.8970 + 3.4528i
  -0.8970 - 3.4528i
```

**e) Gráfica del Lugar Geométrico de las Raíces**

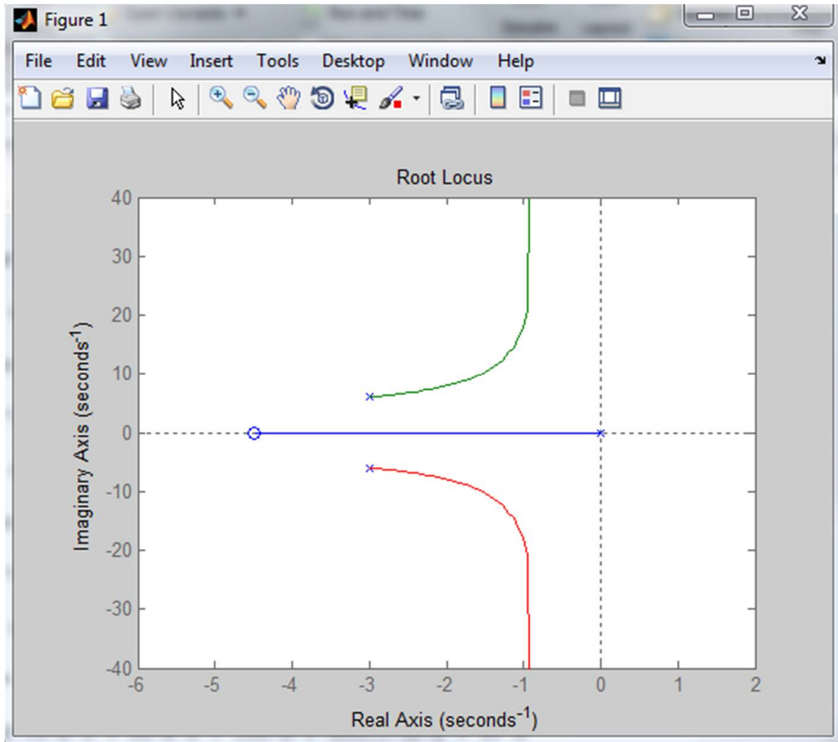
Para Graficar el Lugar Geométrico de las Raíces, usamos el comando **rlocus** que puede recibir como parámetros dos vectores con los valores de los coeficientes del numerador y del denominador respectivamente, o bien la función de transferencia de lazo abierto.

- ✓ Utilización de **rlocus** para graficar recibiendo como parámetros dos vectores.

```
>> num=[1 4.5];
>> den=[1 6 45 0];
>> rlocus(num,den)
```

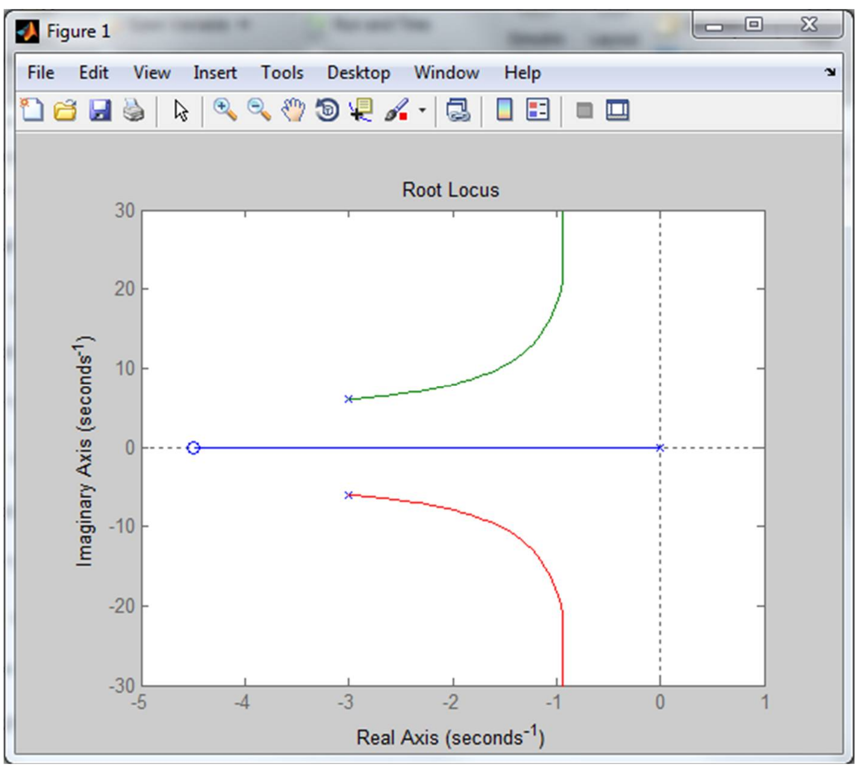
- ✓ Utilización de **rlocus** para graficar recibiendo como parámetro la función de transferencia de lazo abierto.

```
>> s=tf('s');
>> G=(s+4.5)/(s^3+6*s^2+45*s)

G =

        s + 4.5
  ------------------
  s^3 + 6 s^2 + 45 s

Continuous-time transfer function.

>> rlocus(G)
```



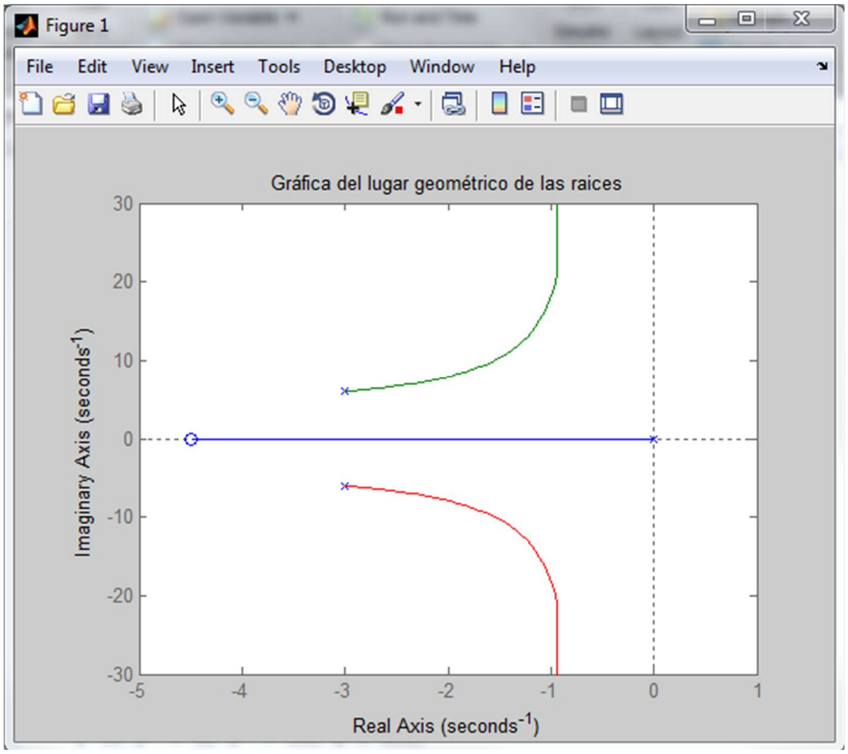

✓ Empleo de **axis** para establecer los valores mínimos y máximos de los ejes de coordenadas, según nuestra conveniencia para analizar mejor el gráfico.

```
>> v=[-5 1 -30 30]; axis(v)
```

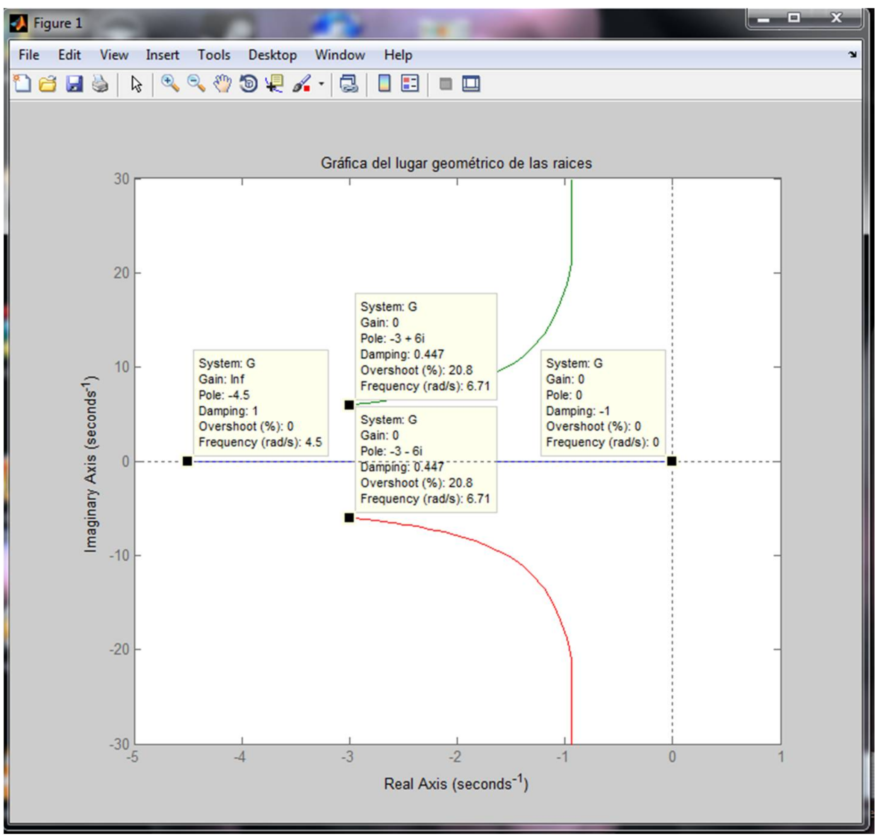



✓ Colocación de un título a la gráfica.

```
>> title('Gráfica del lugar geométrico de las raices')
```



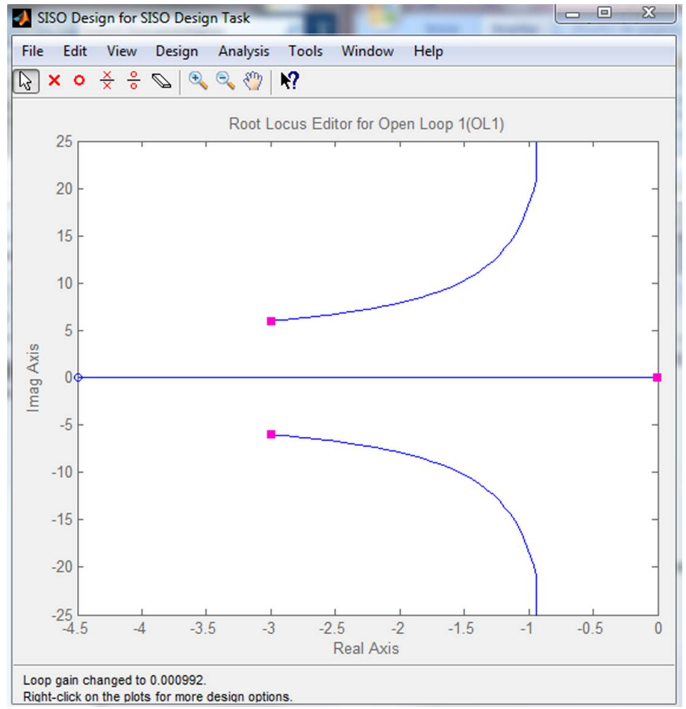

✓ Cabe destacar que ésta gráfica nos permite visualizar los valores de los polos y ceros haciendo clic con el mouse sobre la singularidad que nos interese.

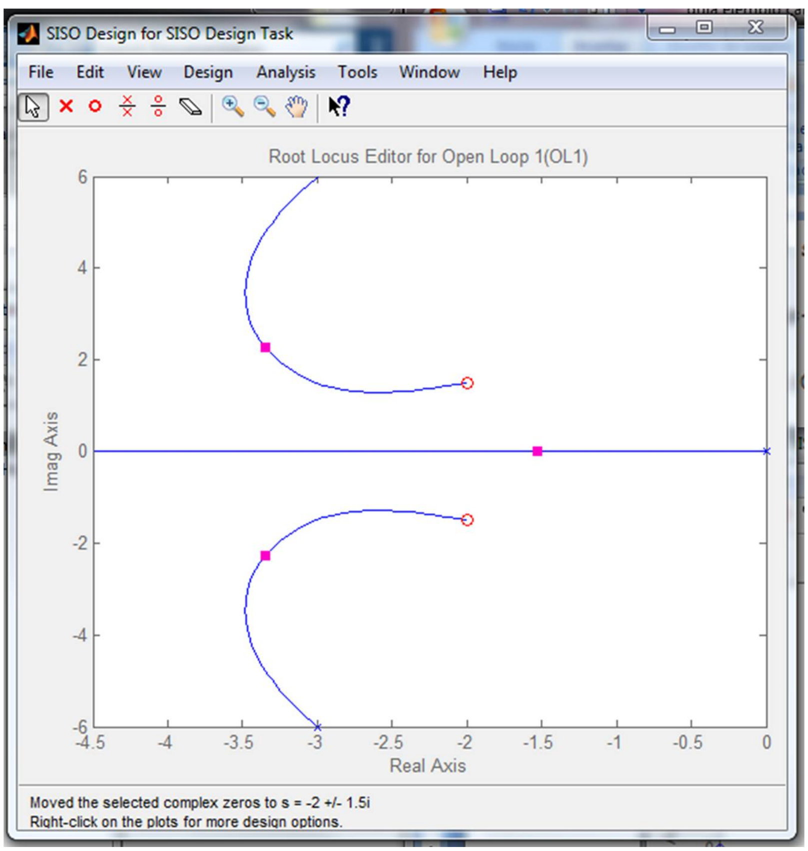



✓ Otro comando muy útil en **MATLAB** es **rltool**, el cual nos permite agregar polos y ceros a la gráfica y ver como varía ésta al ir moviéndolos de lugar.

✓ En este caso se va a ingresar como parámetro la función de transferencia de este ejercicio, y luego se van a agregar dos ceros complejos para visualizar qué sucede con la gráfica.

```
>> s=tf('s');
>> G=(s+4.5)/(s^3+6*s^2+45*s)

G =

       s + 4.5
  ------------------
  s^3 + 6 s^2 + 45 s

Continuous-time transfer function.

>> rltool(G)
```




Como se puede apreciar en esta última gráfica, se agregaron dos ceros complejos en las posiciones -2+1,5j y -2-1,5j observándose los cambios que se produjo en la gráfica con respecto a la anterior.

## Lugar Geométrico de las Raíces utilizando SCILAB

En esta guía vamos a desarrollar un ejemplo de cálculo del Lugar Geométrico de las Raíces y mediante el programa **SCILAB** comprobaremos partes de las resoluciones.

Dada la siguiente ecuación del sistema:

$$1 + k\frac{s + 4{,}5}{s^3 + 6s^2 + 45s} = 0$$

**a) Cálculo de polos y ceros:**

Primero debemos definir los polinomios del numerador y denominador para posteriormente calcular los polos y los ceros de la ecuación. Los polinomios pueden definirse de dos formas diferentes utilizando el comando **poly()**:

Se puede crear polinomios a través de sus coeficientes, en este caso se usa el comando "**poly**", ingresando como parámetros un vector con los coeficientes del polinomio en forma ascendente (de forma inversa a **Matlab**), el nombre de la variable del polinomio 's' y la letra 'c' que significa coeficientes.

```
-->num=poly([9/2 1],'s','c')
 num  =

    4.5 + s

-->den=poly([0 45 6 1],'s','c')
 den  =

             2    3
    45s + 6s + s
```

Para hallar las raíces de un polinomio utilizamos el comando **roots()** indicándole como parámetro el polinomio en cuestión.

```
-->ceros=roots(num)
 ceros  =

  - 4.5

-->polos=roots(den)
 polos  =

  - 3. + 6.i
  - 3. - 6.i
    0
```

A partir de estos comandos determinamos que el cero de la ecuación se encuentra en el punto -4.5 y los polos se encuentran en los puntos 0, -3+6j y -3-6j.

También es posible realizar el proceso inverso con **SCILAB**, es decir obtener los polinomios del numerador y denominador a partir de los ceros y polos respectivamente.

Se utiliza el comando **poly** y pasamos como parámetros un vector con los ceros o polos y la variable incógnita del polinomio.

```
-->num=poly([-4.5],'s')
 num  =

    4.5 + s

-->den=poly([-3+6*%i -3-6*%i 0],'s')
 den  =

                2    3
    45s + 6s + s
```

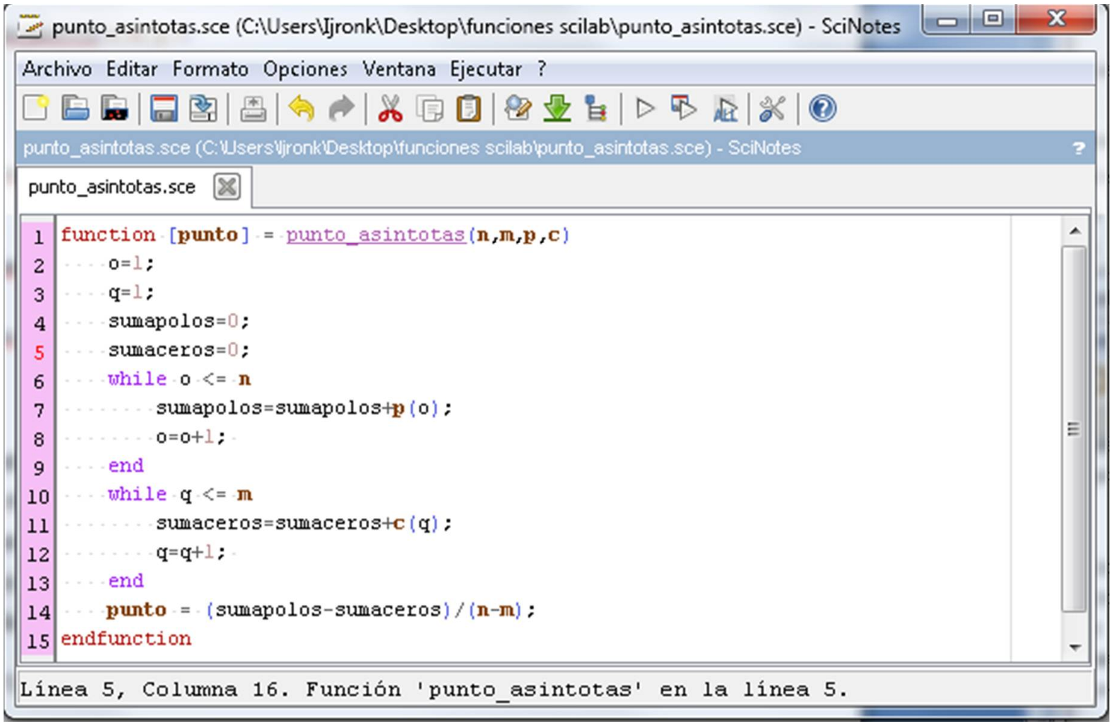
**b) Punto donde nacen las asíntotas**

$$\sigma = \frac{\sum Polos - \sum Ceros}{n - m}$$

$$\sigma = \frac{\big(0 + (-3 + 6j) + (-3 - 6j)\big) - (-4{,}5)}{3 - 1} = \frac{-6 + 4{,}5}{2} = -0{,}75$$

Para realizar este cálculo con **SCILAB** primero creamos una función a la cual posteriormente llamamos enviando como parámetros el número de polos (n), ceros (m) y dos vectores con los valores de los polos (p) y los ceros (c).

```
function [punto]=punto_asintotas(n, m, p, c)
  o=1;
  q=1;
  sumapolos=0;
  sumaceros=0;
  while o <= n
    sumapolos=sumapolos+p(o);
    o=o+1;
  end
  while q <= m
    sumaceros=sumaceros+c(q);
    q=q+1;
  end
  punto = (sumapolos-sumaceros)/(n-m);
endfunction
```




Para ejecutar esta función desde la consola necesitamos ingresar el comando **exec** enviando como parámetro la ruta del archivo guardado, y un **-1** para que no aparezca un "**eco**" de todo lo que se haya escrito en la función, o un **1** para que cada línea que exista en la función aparezca escrita en la consola. El **exec** solo hay que realizarlo una vez.

```
-->exec('C:/Users/Ijronk/Desktop/funciones scilab/punto_asintotas.sce', -1)
```

Ejecutamos el comando **punto_asintota**:

```
-->punto=punto_asintotas(3,1,[0 -3+6*%i -3-6*%i],[-9/2])
 punto  =

  - 0.75
```

Vemos que el punto donde nacen las asíntotas es igual a -0,75.

Como se puede apreciar en **SCILAB** para ingresar un número complejo se debe agregar la notación "*%" antes de la letra **i**.

**c) Ángulo de las asíntotas**

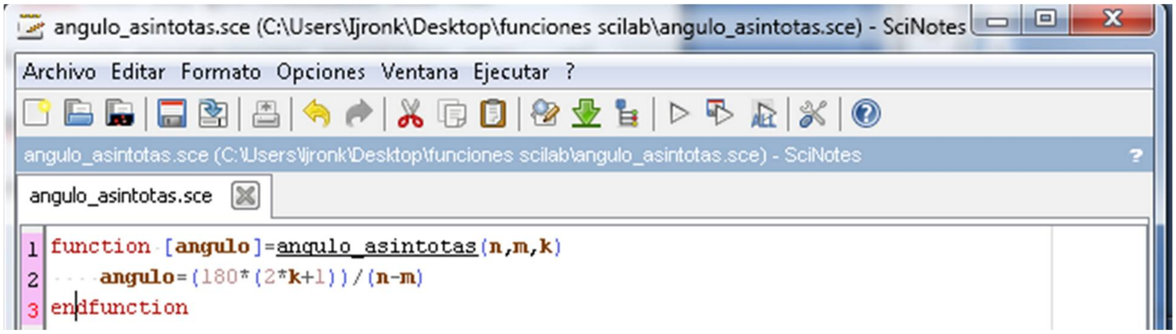
$$\alpha = \frac{180^o(2k + 1)}{n - m}$$

$$k = 0 \quad \alpha_1 = \frac{180^o(2 * 0 + 1)}{3 - 1} = 90^o$$

$$k = 1 \quad \alpha_2 = \frac{180^o(2 * 1 + 1)}{3 - 1} = 270$$

Para realizar la comprobación primero creamos la siguiente función en **SCILAB**:

```
function [angulo]=angulo_asintotas(n, m, k)
   angulo=(180*(2*k+1))/(n-m)
endfunction
```




Cargamos la función

```
-->exec('C:/Users/Ijronk/Desktop/funciones scilab/angulo_asintotas.sce', -1)
```

Luego llamamos a la función anteriormente creada y cargada colocando como parámetros el número de polos (n), número de ceros (m) y el valor del contador k.

```
-->angulo=angulo_asintotas(3,1,0)
 angulo  =

    90.

-->angulo=angulo_asintotas(3,1,1)
 angulo  =

    270.
```

Con el resultado obtenido vemos que 90º y 270º son los ángulos de las asíntotas.

**d) Puntos de Ruptura**

$$1 + \frac{k(s + 4{,}5)}{s^3 + 6s^2 + 45s} = 0$$

$$k = -\frac{s^3 + 6s^2 + 45s}{s + 4{,}5}$$

$$k = \frac{-s^3 - 6s^2 - 45s}{s + 4{,}5}$$

$$\frac{\delta k}{\delta s} = \frac{(s + 4{,}5) * (-3s^2 - 12s - 45) - (-s^3 - 6s^2 - 45s) * 1}{(s + 4{,}5)^2} = 0$$

$$-3s^3 - 12s^2 - 45s - 13{,}5s^2 - 54s - 202{,}5 + s^3 + 6s^2 + 45s = 0$$

$$-2s^3 - 19{,}5s^2 - 54s - 202{,}5 = 0$$

$$s_1 = -7{,}96$$

$$s_2 = -0{,}897 + 3{,}45j$$

$$s_3 = -0{,}897 - 3{,}45j$$

No existe punto de ruptura porque ninguna de las raíces pertenece al Lugar Geométrico.

Con **SCILAB** tenemos la posibilidad de calcular la derivada de k con respecto a s de la siguiente manera:

```
-->s=poly(0,'s')
 s  =

    s

-->k=(-s^3-6*s^2-45*s)/(s+4.5)
 k  =

               2    3
  - 45s - 6s - s
    ------------
      4.5 + s

-->derivat(k)
 ans  =

                           2     3
  - 202.5 - 54s - 19.5s - 2s
    ------------------------
                           2
         20.25 + 9s + s
```

Calculamos las raíces del polinomio resultante de la derivación

```
-->raices=roots(poly([-202.5 -54 -19.5 -2],'s','c'))
 raices  =

  - 7.9559109
  - 0.8970446 + 3.4527812i
  - 0.8970446 - 3.4527812i
```

**e) Gráfica del Lugar Geométrico de las Raíces**

Para Graficar el Lugar Geométrico usamos el comando **evans**.

- ✓ Creación de los polinomios del numerador y denominador de la función de Transferencia de Lazo Abierto.

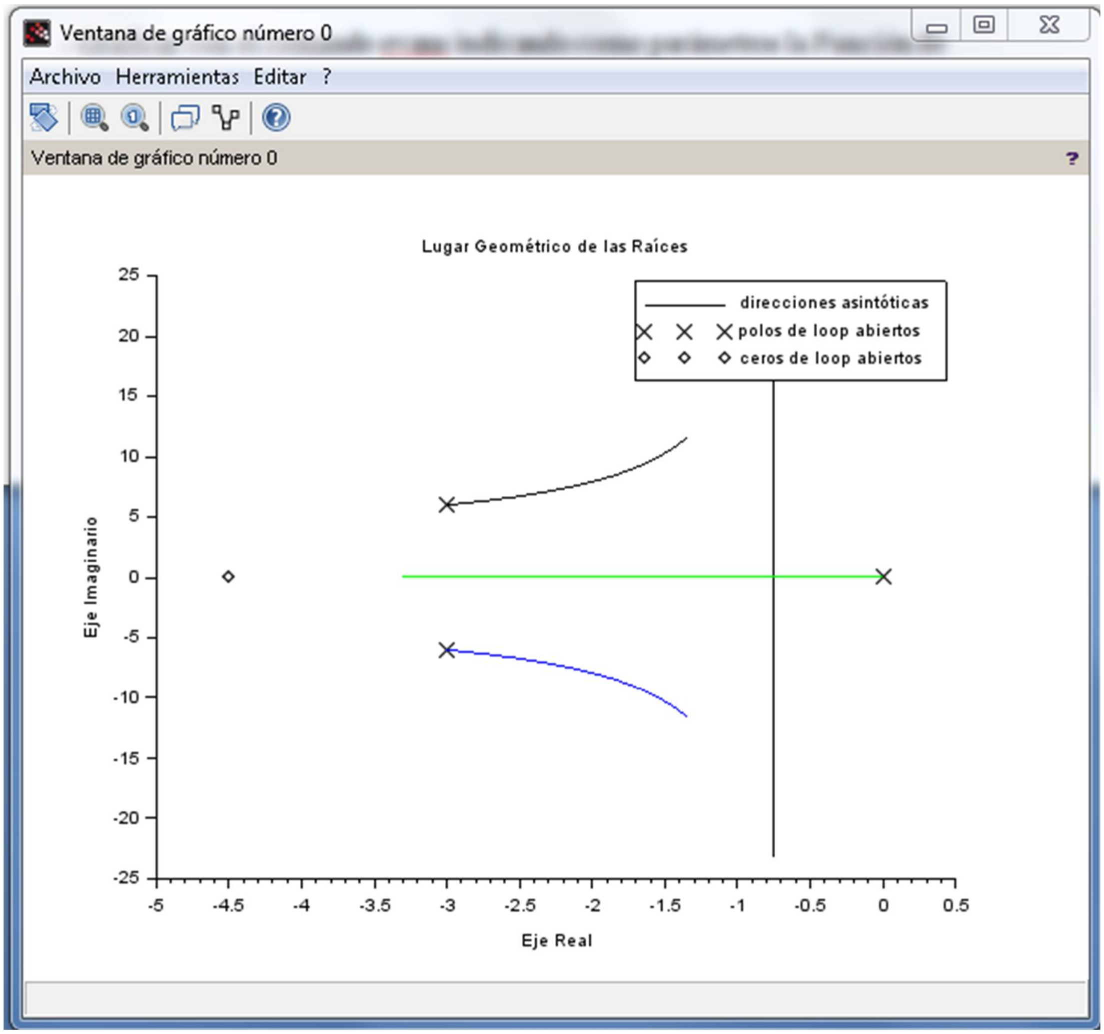
```
-->s=poly(0,'s');

-->num=poly([4.5 1],'s','c')
 num  =

    4.5 + s

-->den=poly([0 45 6 1],'s','c')
 den  =

               2    3
    45s + 6s + s
```

- ✓ Formación de la función de Transferencia de Lazo Abierto.

```
->sis=syslin('c',num,den)
sis  =

     4.5 + s
   -----------
            2    3
   45s + 6s + s
```

✓ Graficar con el comando **evans** indicando como parámetros la Función de Transferencia y la ganancia máxima que se desea.

```
-->evans(sis,100)

-->title('Lugar Geométrico de las Raíces')
```




**f) Cálculo de k y ω críticos**

**SCILAB** nos provee del comando **kpure** para calcular el parámetro **k** y la frecuencia ω cuando el sistema pasa de ser estable a inestable.

```
-->[k,w]=kpure(sis)
 w  =

     []
 k  =

     []
```

En el ejercicio actual no existe un **k** y un ω (omega) críticos, por lo tanto el lugar geométrico de las raíces no atraviesa el eje imaginario.