

Índice General

1	Introducción	1
2	Lugar de las Raíces	1
2.1	Función <code>rlocus</code>	1
2.2	Función <code>rlocfind</code>	2
2.3	Función <code>sgrid</code>	2
2.4	Ejemplos	2
3	Respuesta en Frecuencia	5
3.1	Función <code>bode</code>	5
3.2	Función <code>Nyquist</code>	6
4	Diseño de control	7
4.1	Función <code>rltool</code>	7

1 Introducción

En este apunte veremos algunas funciones de MATLAB relacionadas con lugar de las raíces, gráfico de Bode y de Nyquist. Entre ellas nos centraremos en detallar las siguientes:

Comando	Breve explicación
<code>rlocus</code>	Calcula y grafica el lugar de las raíces de un sistema LTI SISO
<code>rlocfind</code>	Encuentra la ganancia del lugar de las raíces para polos dados
<code>rltool</code>	Permite diseñar un compensador SISO usando lugar de las raíces
<code>zmap</code>	Calcula y grafica polos y ceros de un sistema LTI
<code>bode</code>	Respuesta en frecuencia de Bode para modelos LTI
<code>nyquist</code>	Respuesta en frecuencia de Nyquist para modelos LTI

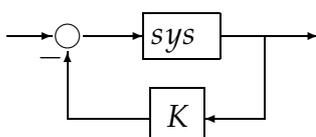
Tabla 1: Comandos de Matlab

2 Lugar de las Raíces

A continuación detallaremos el uso de las funciones de MATLAB relacionadas con el gráfico del lugar de las raíces. En donde citaremos algunos ejemplos para clarificar.

2.1 Función `rlocus`

Calcula y grafica el lugar de las raíces de un sistema LTI SISO. El gráfico del lugar de las raíces se utiliza para analizar el lazo de realimentación negativa de la Figura 1 y muestra las trayectorias de los polos a lazo cerrado cuando la ganancia K varía de 0 a ∞ .



$$G(s) = \frac{sys(s)}{1 + Ksys(s)}$$

Figura 1: Realimentación negativa y su función de transferencia

La función puede utilizarse de la siguiente forma:

```
R=rlocus(sys,K)
[R,K]=rlocus(sys)
```

donde en el parámetro R matricial devuelve el lugar complejo de la raíz para la ganancia K . En el primer caso indicamos qué intervalo de valores de ganancias estamos interesados, mientras que si no le ingresamos dicho parámetro, K varía de 0 a ∞ . El sistema puede ser ingresado como función transferencia, con el comando `tf` o `zpk`, o simplemente pasándole el numerador y el denominador del sistema, es decir `rlocus(num,den)`.

2.2 Función `rlocfind`

Esta función nos permite hallar la ganancia del gráfico del lugar de raíces correspondiente para un conjunto de polos dados. El comando ingresado de la siguiente forma

```
[K,polos]=rlocfind(sys);
```

se utiliza para seleccionar la ganancia del lugar de las raíces generado por `rlocus` en forma interactiva. Luego de ejecutado dicho comando, aparece una cruz sobre el gráfico del lugar de las raíces con el que seleccionaremos el lugar deseado para los polos a lazo cerrado. La ganancia asociada con el punto seleccionado es la que este comando devuelve en la variable K y todos los polos a los que le corresponde esa ganancia los devuelve en la variable $polos$.

2.3 Función `sgrid`

Genera una grilla en el plano complejo " s " para un lugar de las raíces ya existente o un mapa de polos y ceros. Se dibujan líneas de amortiguamiento (ζ) y frecuencia natural (ω_n) constantes. El comando se utiliza de la siguiente forma:

```
sgrid(z,wn)
```

2.4 Ejemplos

Ejemplo 1. Dada la siguiente función de transferencia a lazo abierto

$$G(s)H(s) = \frac{K}{s(s+2)},$$

vamos a calcular el lugar de las raíces. Para ello ingresamos los siguientes comandos desde el *WorkSpace* de MATLAB

```
sys=zpk([], [0 -2], 1);           o   sys=tf(1, [1 2 0]);
rlocus(sys);
```

obteniendo la Figura 2

Si tomamos el mismo sistema pero le agregamos un polo en $p = -3$ y otro en $p = -4$, e ingresamos nuevamente los comandos correspondientes, obtenemos los gráficos de la Figura 3 respectivamente, en donde observamos como el gráfico se ve forzado hacia la derecha debido a cada polo que agregamos.

Ejemplo 2. Supongamos ahora una planta con la siguiente función de transferencia:

$$G(s)H(s) = \frac{K(s+1)}{s^2(s+\alpha)}, \quad \text{donde } \alpha = 10, 9, 8 \text{ y } 3$$

Si ejecutamos los mismos comandos que en el ejemplo anterior, es decir, ingresamos primero el sistema y luego le calculamos el lugar de raíces, obtenemos los resultados que se observan en la Figura 4 y 5

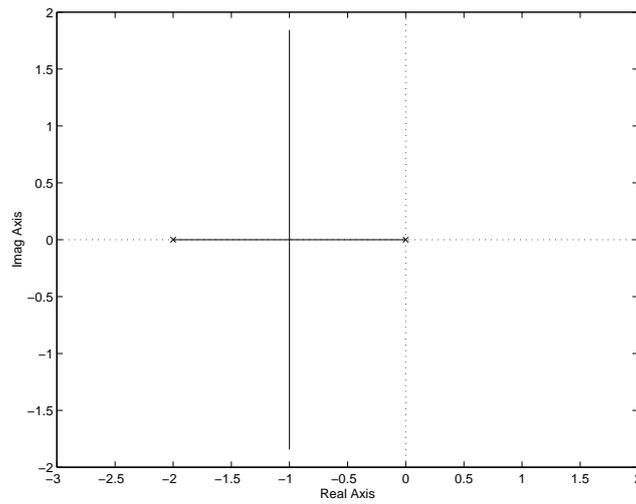


Figura 2: Gráfica del lugar de las raíces

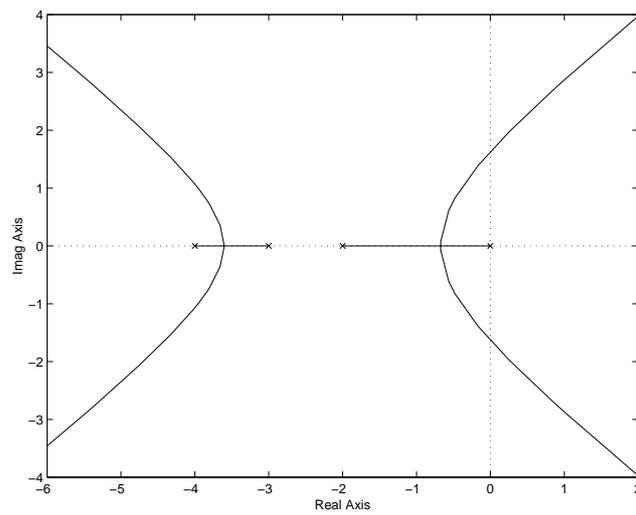


Figura 3: Gráfica del lugar de las raíces

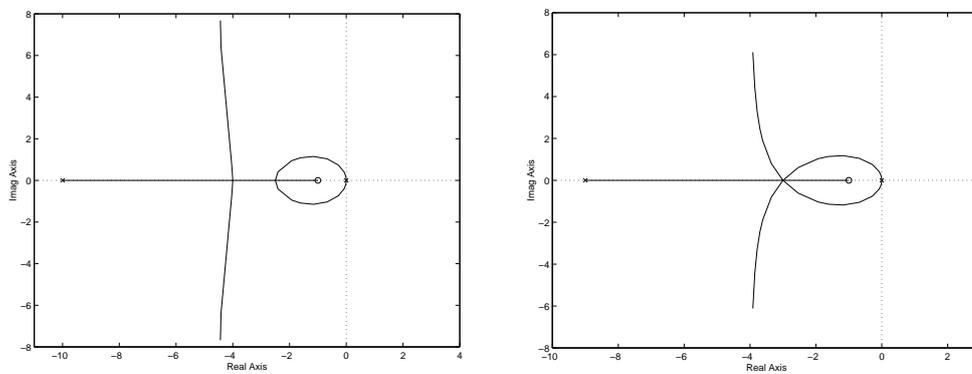
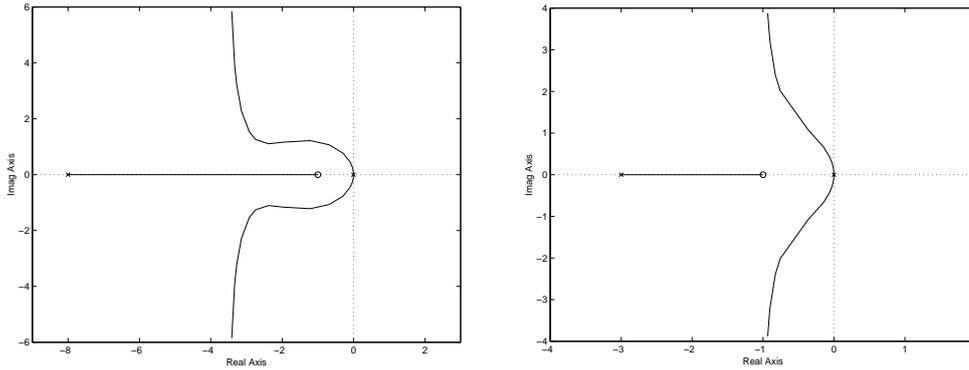


Figura 4: Gráfica del lugar de las raíces para $\alpha = 10$ y $\alpha = 9$, respectivamente


 Figura 5: Gráfica del lugar de las raíces para $\alpha = 8$ y $\alpha = 3$, respectivamente

Existen casos donde queremos analizar el comportamiento de un sistema a lazo cerrado (LC), pero donde el parámetro variable no aparece como un factor multiplicativo (como lo es la ganancia K en los ejemplos anteriores) del sistema a lazo abierto (LA). En esos casos tenemos que reescribir la ecuación característica de forma tal que el parámetro variable aparezca como un factor multiplicativo de $G(s)H(s)$ y así podremos utilizar el comando `rlocus`. Los ejemplos a continuación ilustran cómo podemos proceder en distintos casos.

Ejemplo 3. Grafiquemos el lugar de las raíces del diagrama de la Figura 6. La función de transferen-

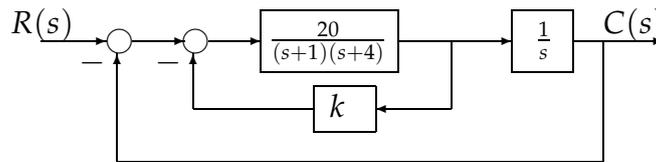


Figura 6: Sistema de control

cia a LA resulta

$$G(s)H(s) = \frac{20}{s(s+1)(s+4) + 20ks}$$

donde la ecuación característica es $s^3 + 5s^2 + 4s + 20 + 20ks = 0$. Como la variable ajustable no aparece como factor multiplicativo, llamemos $K = 20k$ y reescribamos la ecuación característica. Si ahora dividimos ambos miembros por el polinomio que no tiene términos con K , tenemos

$$1 + \frac{Ks}{s^3 + 5s^2 + 4s + 20} \quad \text{que es del tipo} \quad 1 + K \frac{\text{num}(s)}{\text{den}(s)}.$$

Ahora podemos graficar el lugar de las raíces utilizando la sentencia `rlocus(num, den)`.

Ejemplo 4. Consideremos una planta con función transferencia $G_0(s)$ y un controlador en realimentación con función transferencia $C(s)$, donde

$$G_0(s) = \frac{1}{(s+1)(s+2)} \quad \text{y} \quad C(s) = 4 \frac{(s+\alpha)}{s}.$$

Queremos conocer cómo varía la ubicación de los polos a lazo cerrado para α variando en \mathbb{R}^+ . Veamos que los polos a lazo cerrado del sistema son los ceros de

$$1 + 4 \frac{s+\alpha}{s(s^2+s-2)} = \frac{s(s^2+s-2) + 4s + 4\alpha}{s(s^2+s-2)} = 0 \quad \Rightarrow \quad s(s^2+s+2) + 4\alpha = 0$$

Si dividimos ambos miembros por $s(s^2 + s + 2)$, convertimos la ecuación a la forma $1 + kG(s)$, donde

$$k = 4\alpha \quad \text{y} \quad G(s) = \frac{1}{s(s^2 + s + 2)}$$

Lo único que queda ahora es ingresar el comando `rlocus` para obtener la Figura 7

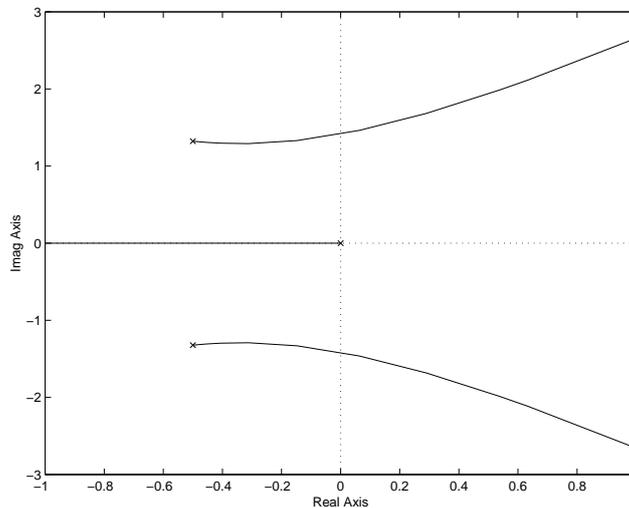


Figura 7: Ubicación de los polos a LC cuando el cero del controlador varía

3 Respuesta en Frecuencia

En esta sección detallaremos las funciones de MATLAB para graficar la respuesta en frecuencia, es decir para obtener los gráficos de Bode y de Nyquist, con las funciones `bode` y `nyquist` respectivamente.

3.1 Función `bode`

Esta función nos permite obtener la respuesta en frecuencia de Bode para modelos LTI. Entre las formas más comunes de utilizar esta función se encuentran:

- `bode(sys)`: dibuja el gráfico de Bode del modelo LTI `sys` (creado con `tf` o `zpk`). El rango de frecuencia y el número de puntos que tomará para graficar los elige en forma automática.
- `bode(sys, wmin, wmax)`: dibuja el gráfico de Bode para frecuencias entre w_{min} y w_{max} (en rad/seg).
- `bode(sys, w)`: utiliza el vector `w` de frecuencias propuesto para calcular el Bode. Dado que el vector `w` debe estar en escala logarítmica, existe en MATLAB la función `logspace` que genera un vector de frecuencias en forma logarítmica.
- `bode(sys1, sys2, ..., w)`: dibuja el gráfico de Bode de varios modelos LTI en una sola figura. El parámetro `w` es opcional, también se puede especificar color, tipo de línea y marcadores como se los utiliza con el comando `plot`.
- `[mag, fase]=bode(sys, w)` o `[mag, fase, w]=bode(sys)`: devuelve la magnitud y la fase en grados. Este comando no dibuja en pantalla, `mag(:,k)` y `fase(:,k)` determina la respuesta en $w(k)$. Para obtener magnitudes en dB, debemos calcular `magdb=20*log10(mag)`.

Ejemplo 5. Dada la siguiente transferencia, queremos obtener el gráfico de Bode

$$G(s) = \frac{2500}{s(s+5)(s+50)}$$

Para ello ejecutemos los siguientes comandos desde el *workspace*:

```
G=zpk([], [0 -5 -50], 2500);
bode(G);
```

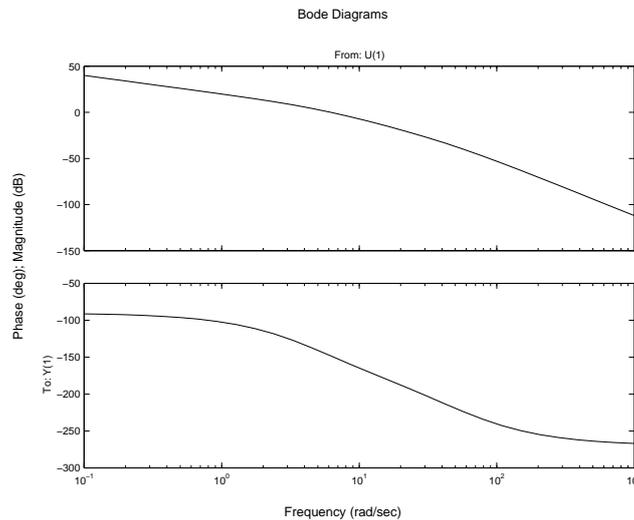


Figura 8: Gráfico de Bode obtenido con el comando `bode(G)`

3.2 Función Nyquist

Esta función nos permite obtener la respuesta en frecuencia de Nyquist para modelos LTI. Entre las formas más comunes de utilizar esta función se encuentran:

- `nyquist(sys)`: dibuja el gráfico de Nyquist de `sys` que es un modelo LTI creado con los comandos `tf` o `zpk`. El rango de frecuencia y el número de puntos que utilizará para graficar son elegidos en forma automática.

Las distintas alternativas para este comando son las misma que para la función `bode`, excepto el últimos de los ítems que se reemplaza por:

- `[Re, Im]=nyquist(sys, w)` o `[Re, Im, w]=nyquist(sys)`: devuelve la parte real e imaginaria de la respuesta en frecuencia, a lo largo de `w`. La respuesta a la frecuencia $w(k)$ está dada por $Re(:,k)*Im(:,k)$

Ejemplo 6. Dado el sistema cuya función transferencia viene dada por

$$G(s) = \frac{1}{(s+1)^2}$$

obtener el diagrama de Nyquist.

Para ello ingresamos los siguientes comandos

```
G=zpk([], [-1 -1], 1);
nyquist(G)
```

y obtuvimos la Figura 9

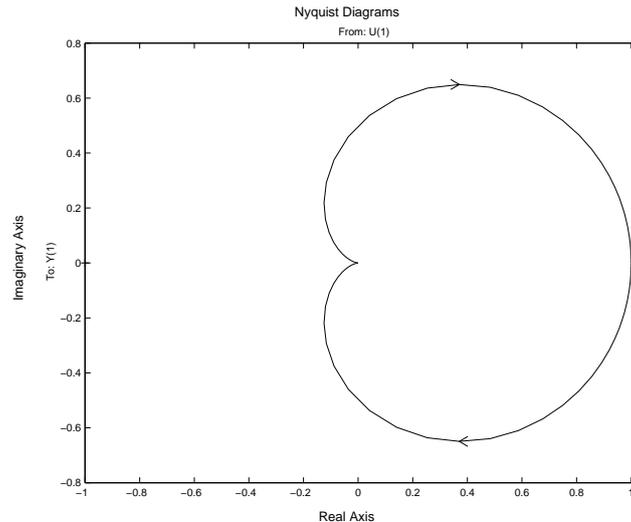


Figura 9: Gráfico de Nyquist obtenido con el comando `nyquist(G)`

4 Diseño de control

Presentaremos ahora una herramienta de diseño interactivo llamada `rltool` que puede utilizarse para:

- Analizar el lugar de las raíces para los sistemas de control LTI SISO.
- Especificar los parámetros de un compensador de realimentación: polos, ceros y ganancia.
- Examinar cómo cambiando los parámetros del compensador, cambia el lugar de las raíces y varias respuestas a lazo cerrado, como la respuesta al escalón unitario, respuesta al impulso unitario, diagramas de Bode y/o Nyquist entre otros.

4.1 Función `rltool`

Para explicar mejor el uso de esta herramienta, seguiremos un ejemplo que involucra un servomecanismo electro-hidráulico, que esencialmente es un amplificador de potencia electro-hidráulico, controlado por una válvula piloto y un actuador, Figura 10. Estos servomecanismos son muy pequeños y se utilizan para controlar posición. Un modelo a lazo cerrado para el control de posición puede representarse como aparece en la Figura 10 (detalles del modelado matemático del mecanismo y su linealización ver [2, pag. 91] o [1, pag. 43]).

En la Figura 10, $K(s)$ representa el compensador que deseamos diseñar. Este compensador puede ser tanto una ganancia como un sistema LTI. La planta linealizada viene dada por

$$G_{servo}(s) = \frac{4 \cdot 10^7}{s(s + 250)(s^2 + 40s + 9 \cdot 10^4)}$$

Para este ejemplo, queremos diseñar un controlador de forma tal que la respuesta al escalón a lazo cerrado cumpla con las siguientes especificaciones de diseño:

- El tiempo de establecimiento menor a 0.05 seg.
- El sobrevalor máximo menor al 5%.

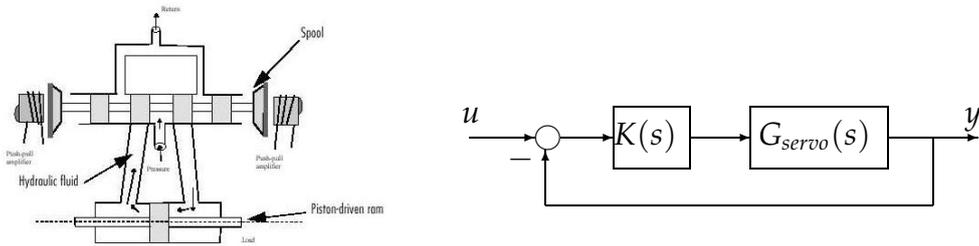


Figura 10: Servomecanismo electro-hidráulico y el diagrama de bloques del sistema más el compensador, respectivamente.

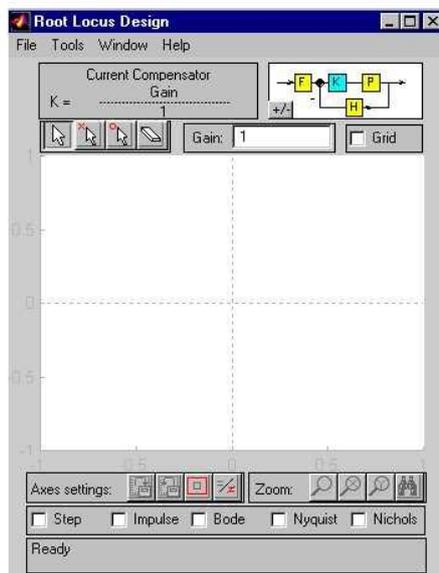


Figura 11: Ventana de la herramienta rltool

Para comenzar, desde el *workspace* de MATLAB, ingresemos el comando `rltool`, con el que abrimos una nueva ventana, Figura 11, en la que encontramos:

- Una barra de menú, entre los que encontramos, por ejemplo, una opción para importar/exportar modelos y para editarlos.
- Un gráfico de bloques que es la estructura de realimentación que utilizaremos; si hacemos *click* sobre el bloque K , podemos ver o editar el compensador, y sobre P , F o H para ver las características de diseño del modelo.
- Un botón para cambiar entre realimentación negativa y positiva.
- Una descripción del compensador que configuraremos. Por defecto toma el valor de ganancia igual a 1 ($K = 1$).
- Cuatro botones que utilizaremos para agregar polos o ceros del compensador, borrarlos o moverlos.
- Un cuadro de texto para editar la ganancia que modificará el lugar de los polos a LC.
- Botones para editar los ejes.
- Botones para modificar el *zoom* del gráfico.
- *Check boxes* para abrir herramientas de análisis de respuesta del sistema.
- Una barra de *status* que provee información.

Una vez abierta la ventana, debemos importar el modelo del sistema para el que queremos diseñar un compensador. Existen cuatro formas para importar un modelo LTI, y éstas son:

1. Cargar el modelo desde el *workspace* de MATLAB, con el comando `rltool(sys, comp)`, donde *sys* es la función transferencia del sistema ingresada como tal, y *comp* la del compensador (opcional).
2. Cargar el modelo desde un archivo de extensión *.mat* de un disco.
3. Cargar bloques LTI SISO desde un diagrama de SIMULINK.
4. Crear los modelos utilizando `tf`, `ss` o `zpk`.

Para este ejemplo, importe nuestro modelo del servomecanismo desde el *workspace*. Para ello debemos previamente ingresar el modelo con la función `tf` o `zpk` y guardarla en una variable, por ejemplo *Gservo*. Una vez cargado el modelo, lo podemos importar desde el menú *Import Model* del menú *File*. En la ventana de importación que se desplegará encontraremos

- Un diagrama correspondiente a la estructura de realimentación que utilizaremos.
- Un botón que conmuta entre las dos posibles estructuras de realimentación que hay configuradas.
- Una lista para seleccionar desde dónde se importará el modelo.
- Una lista de sistemas LTI disponibles o bloques.
- Un cuadro de texto para editar el nombre del diseño.
- Tres botones con flechas para transferir el modelo seleccionado de la lista al componente de diseño del modelo, ya sea P , F o H .

- Tres cuadros de textos para los nombres de los componentes del diseño. Desde este recuadro podemos también definir funciones de transferencias para cada componente, utilizando tF , ss o zpk .

Ya sea en cualquiera de las dos configuraciones, cada componente representa: F - prefiltro; P - modelo de la planta; H - dinámica de la planta; K - compensador a diseñar.

Siguiendo con nuestro ejemplo, carguemos el modelo lineal en P , seleccionándolo desde la lista de *workspace* y luego haciendo un click sobre la flecha que señala a P , o simplemente tipeando en el cuadro de texto $Gservo$ (nombre de nuestro modelo). Luego de seleccionar OK, en la región de gráfico de la ventana de diseño, aparecerá el gráfico del lugar de las raíces del modelo ingresado, Figura 12. Los pequeños cuadros rojos sobre él corresponden a los polos a lazo cerrado correspondientes al valor de la ganancia del compensador.

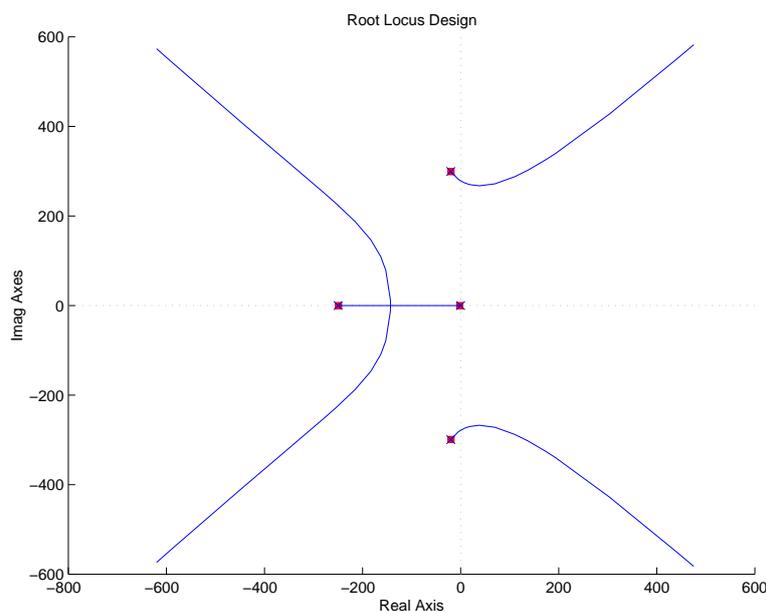


Figura 12: Lugar de las raíces que resulta cuando cargamos el modelo

Veamos hasta qué valor de ganancia podemos aplicarle al compensador para que se mantenga estable a lazo cerrado, es decir, hasta que los polos a lazo cerrado se mantengan en el semi-plano izquierdo del plano complejo. Este límite lo podemos calcular de la siguiente forma:

1. Mover el puntero del mouse sobre un de los cuadrados rojos, donde aparecerá una mano en lugar del puntero. Arrastrar dicho cuadrado hasta lo más cerca del eje imaginario. Observar que el valor de la ganancia se va modificando.
2. Como no podemos saber si los polos se encuentran exactamente sobre el eje imaginario, utilizar el *zoom* para acercarlos mejor. Una vez alcanzado el eje imaginario, quitar el *zoom* con el botón de los binoculares.
3. Por último, verificar que la ganancia correspondiente para que los polos a lazo cerrado sean imaginarios puros es aproximadamente 43.5 (*ganancia crítica*).

El valor de la ganancia podría haberse modificado directamente desde el recuadro correspondiente a la ganancia del compensador, hasta que los polos a lazo cerrado se ubiquen sobre el eje imaginario.

Antes de diseñar el compensador, quisiéramos conocer cómo se comporta el sistema a lazo cerrado para un determinado valor de ganancia. Para ello, debemos seleccionar el *check box* que aparece

en la parte inferior de la ventana la opción del gráfico que no interesa conocer. Para este ejemplo las especificaciones de diseño vienen dadas sobre la respuesta al escalón, por lo que seleccionaremos dicha opción, que desplegará una ventana de LTIVIEWER. Si editamos el valor de la ganancia, por ejemplo 20, y presionamos ENTER, veremos como la respuesta al escalón de la ventana LTIVIEWER se modificó. Recordemos que en dicha ventana, cuando hacemos *click* sobre la figura con el botón derecho del mouse, podemos seleccionar que nos muestre las características que necesitamos, en este caso el sobrevalor y el tiempo de establecimiento. Podemos observar que en este caso por más que modifiquemos la ganancia, las especificaciones nunca se alcanzan. Por lo que para que se cumplan debemos diseñar un compensador.

Para diseñar el compensador, debemos conocer dónde queremos que se ubiquen los polos a lazo cerrado para que las especificaciones se cumplan. Por eso vamos a elegir del menú *Tools*, la opción *Add Grid/Boundary*, que realizará rectas donde se especifiquen, en este caso le indicaremos el tiempo de establecimiento igual a 0.05 seg. y el factor de amortiguamiento igual a 0.7 (ver [1, contratapa]). Una vez seleccionados dichos valores, veremos que en el gráfico aparecen las dos rectas correspondientes a las características ingresadas. Ahora sabemos que para que se cumplan las especificaciones de diseño, los polos dominantes del sistema a lazo cerrado deberán ubicarse en la intersección de dichas rectas. Para ello, debemos "mover" el lugar de las raíces agregando polos y ceros al compensador. Para ello debemos proceder de alguna de las siguientes formas:

- **Utilizando los botones de agregar, o quitar, polos o ceros:** seleccionar el botón correspondiente, luego con el puntero del mouse indicar dónde se ubicará el polo y hacer un *click* en dicho lugar.

Para nuestro ejemplo, ubiquemos un par de polos complejos conjugados por debajo y a la derecha de los polos a lazo abierto. Agreguemos un par de ceros complejos conjugados "cerca" de los polos a lazo abierto. Observemos cómo se modificó el lugar de las raíces. Si todavía no pasa por la intersección de las rectas correspondientes a las características de diseño, mover los polos y/o ceros. Una vez conseguido esto, modificar la ganancia para que los polos a lazo cerrado se ubiquen donde queremos, Figura 13. Por último nos queda verificar con la respuesta al escalón que se cumplen las especificaciones dadas, Figura 14.

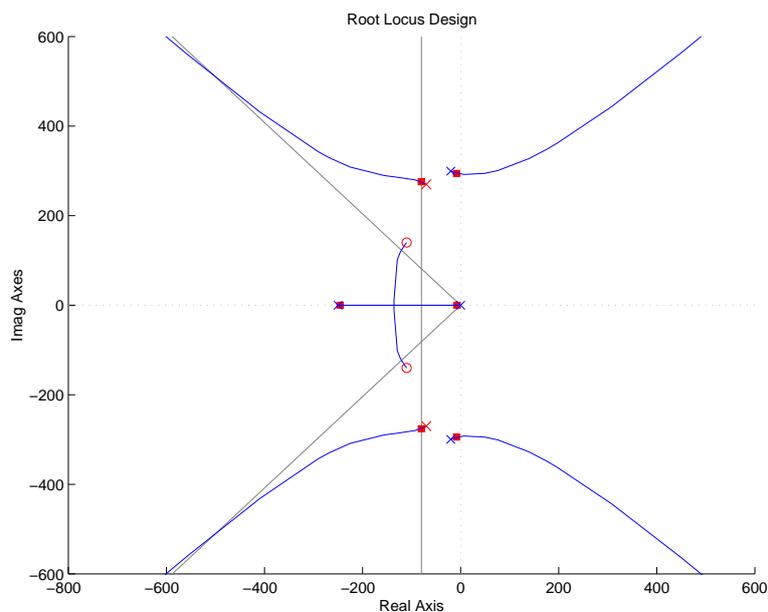


Figura 13: Ubicación deseada de los polos a LC.

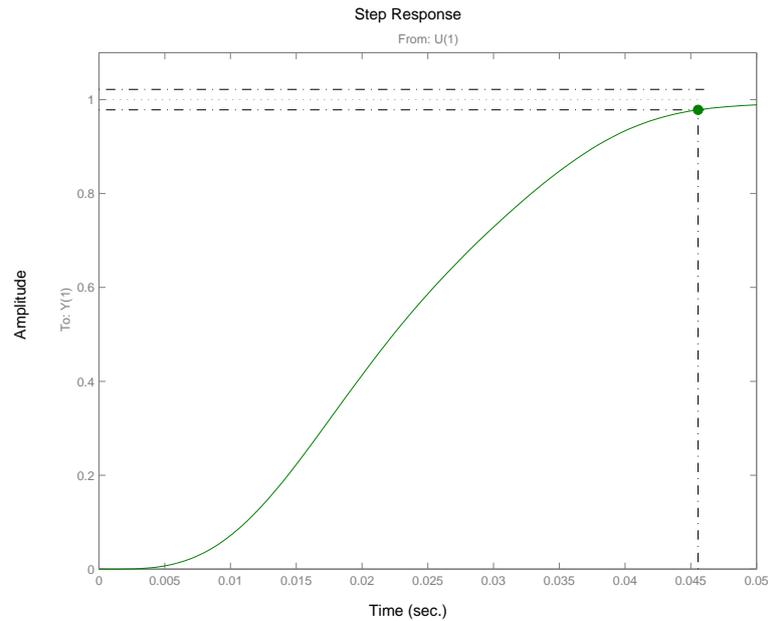


Figura 14: Respuesta al escalón del sistema a LC.

- Utilizando el menú *Edit compensator* del menú *Tools*, o haciendo un *click* sobre el compensador del gráfico de bloques de la derecha. Si ya realizamos el procedimiento anterior, sabemos que los polos, ceros y ganancia del compensador se puede aproximar con

$$K = 9.7$$

$$\text{Polos} = -110 \pm 140i$$

$$\text{Ceros} = -70 \pm 270i$$

Una vez encontrado el compensador que lleva a nuestro sistema a que cumpla con las especificaciones dadas, podemos guardar los parámetros desde el menú *File*, con la opción *Export* podemos llevarlo a un disco, o al *workspace* de MATLAB.

Referencias

- [1] Gene F. Franklin, J.David Powel, and Abbas Emami-Naeini. *Control de Sistemas Dinámicos con Retroalimentación*. Addison-Wesley Iberoamericana, 1991.
- [2] Katsiuko Ogata. *Control Moderno*. Pentice Hall, 1998.