

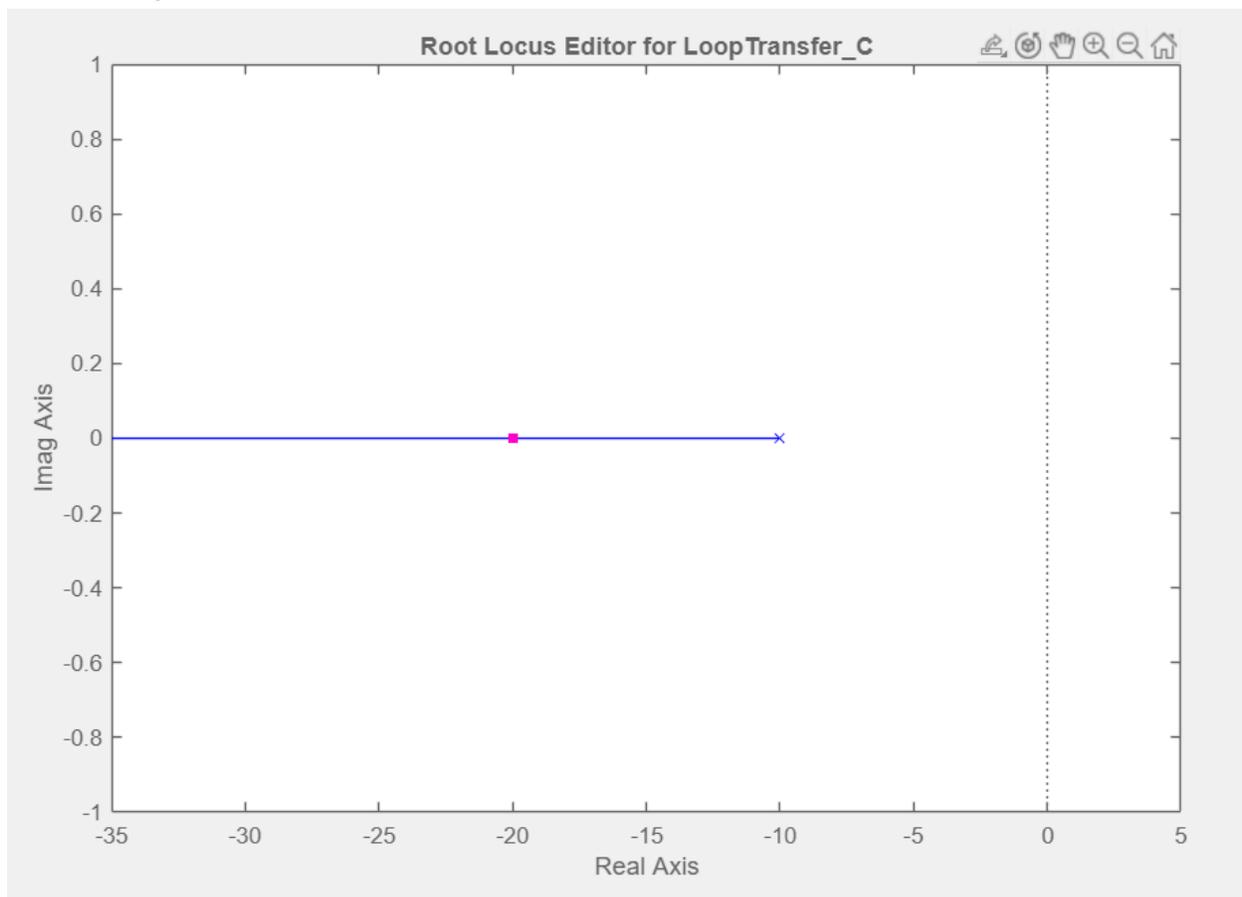
P3 Aux 3.5

1. Grafique el LGR de la planta $G(s) = 10/(s+10)$. Proponga un controlador simple usando cancelación de polos y que asegure cero error en estado estacionario a entrada escalón. Ajuste la ganancia del controlador para obtener distintas frecuencias naturales (por lo menos 2). Verifique el resultado de sus controladores implementándolos en Simulink.

Primero que nada creamos un archivo .m en matlab. Luego de esto definimos el sistema de la siguiente forma:

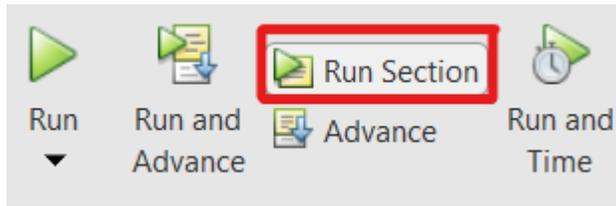
```
%% 1)
% Se define el sistema a partir de los coeficientes del numerador y
% denominador
num= 10;
den= [1 10];
sys= tf(num,den);
```

Luego abrimos el lugar de la raíz con el comando `rltool()`. Esto nos entrega el lugar de la raíz que es el siguiente:

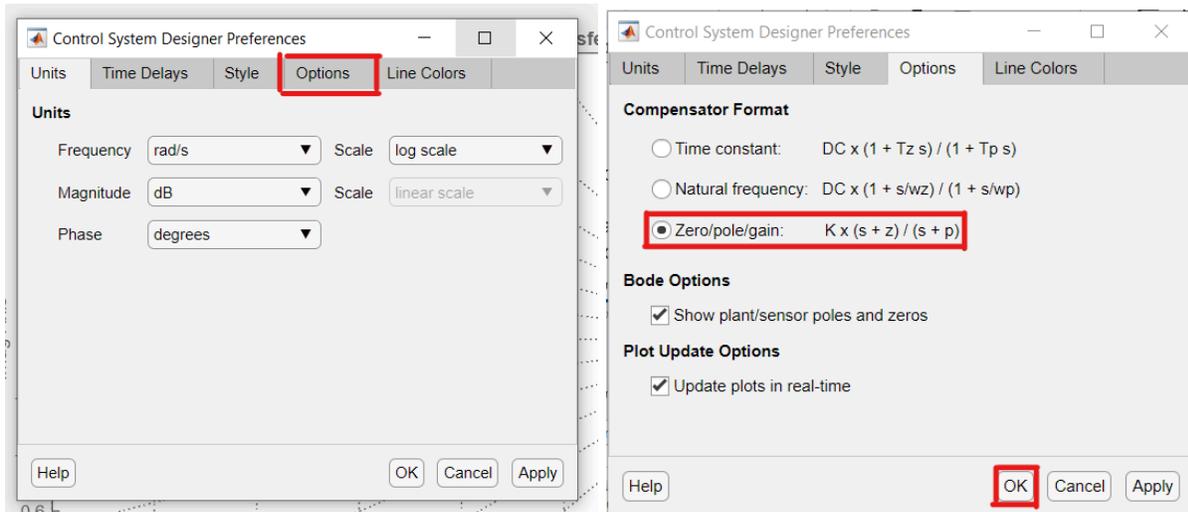
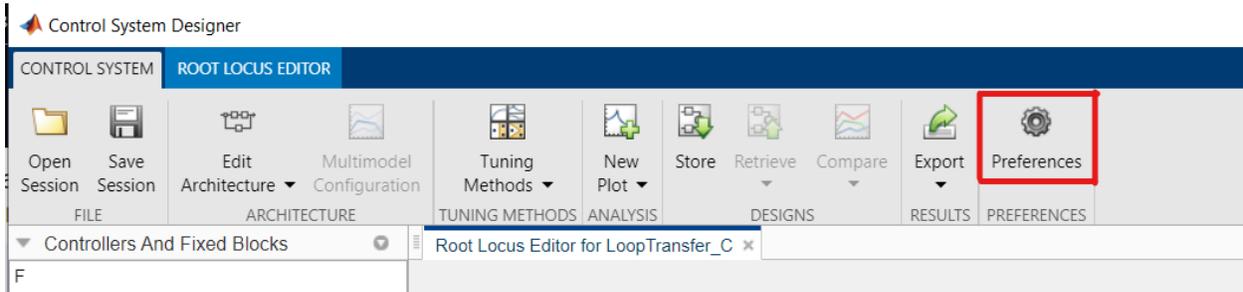


Nota: Para correr solo ciertos segmentos de un código de matlab (archivos .m) se puede poner una sección entre “%%”, utilizándolo en la primera línea para iniciar la sesión y luego en la última línea de esa para terminarla. Luego se puede correr esa sección utilizando “Run

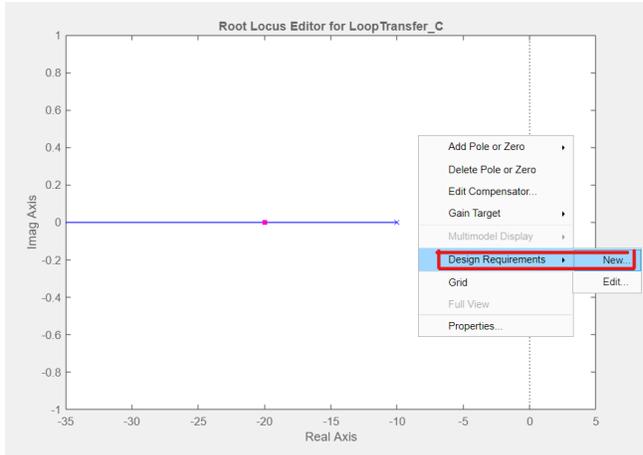
Section”.



Ahora para diseñar el controlador, primero (de manera opcional) cambiamos las preferencias para que el diseño del controlador quede de la forma ceros, polos y ganancia.



Luego apretamos click derecho en el plano del lugar de la raíz para incluir los requerimientos de diseño (Frecuencia natural, coeficiente de amortiguamiento, tiempo de establecimiento, sobrepaso, etc.) que en nuestro caso nos piden al menos 2 frecuencias naturales. Vamos a utilizar 50 [rad/s] en este caso.



New Design Requirement

Design requirement type: Natural frequency

Design requirement parameters

Natural frequency at least 50 rad/s

OK Close Help

Ahora hay varias formas de continuar, se puede ir a la sección “Root Locus Editor” para agregar los polos y ceros de forma manual, o utilizando el click derecho nuevamente. Para este caso vamos a usar la opción “Edit Compensator”, a la que se llega después de apretar click derecho **en la parte blanca del LGR**, ya que si aprietan en la parte amarilla se abre la configuración de los requerimientos de diseño. Luego se muestra el siguiente menú:

Root Locus Editor for LoopTransfer_C

Compensator Editor

Compensator

C = 1

Pole-Zero Parameter

Dynamics

Edit Selected Dynamics

Select a single row to edit values

Right-click to add or delete poles/zeros

Add Pole or Zero

Delete Pole or Zero

Real Pole

Complex Pole

Integrator

Real Zero

Complex Zero

Differentiator

Lead

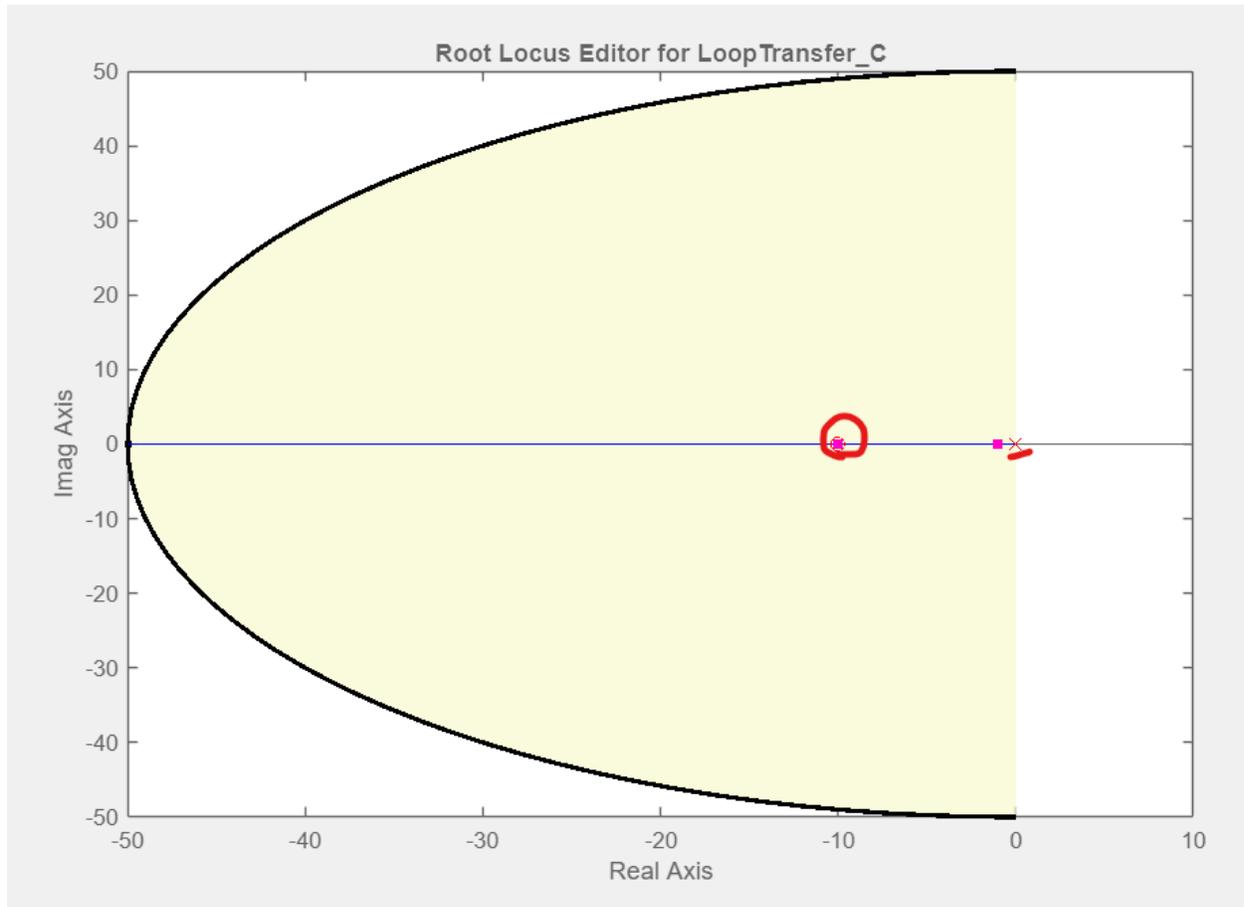
Lag

Notch

Help

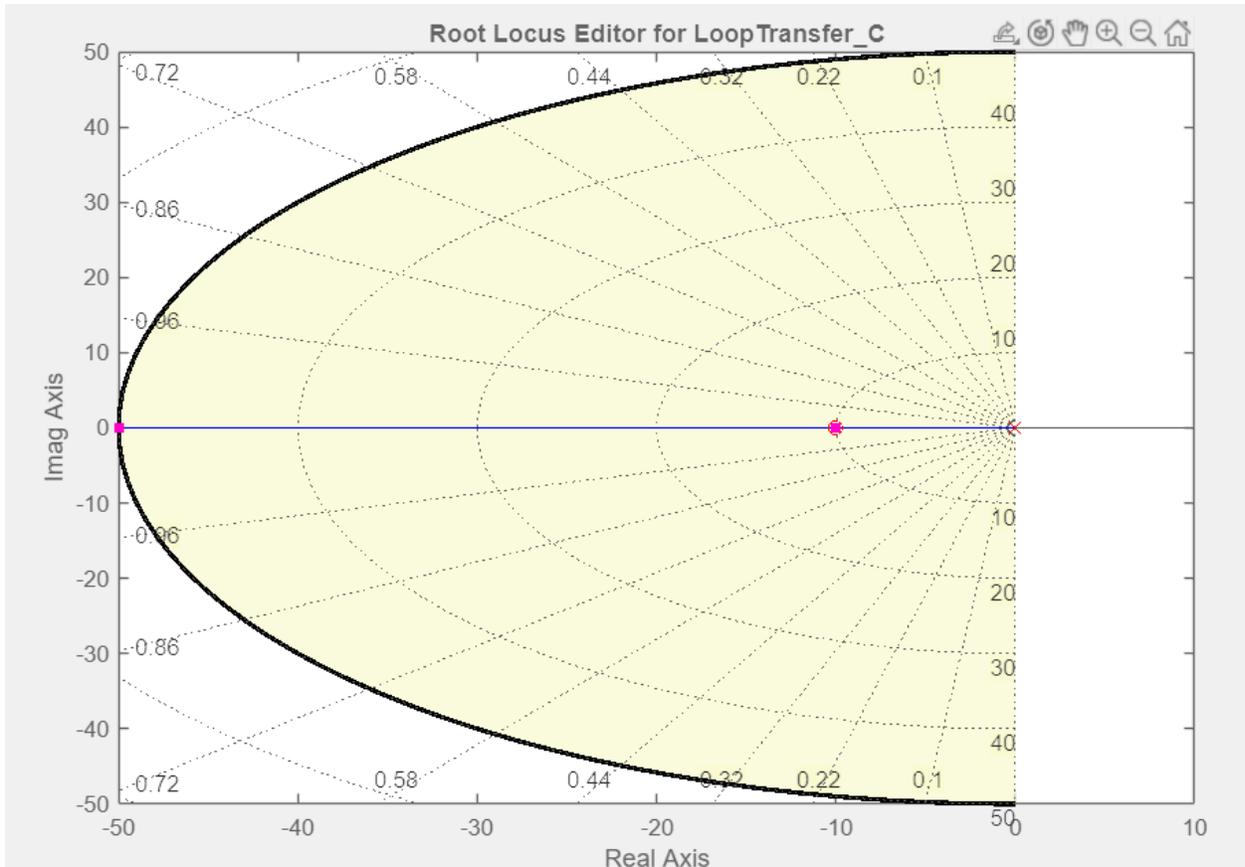
En la parte de arriba se muestra el controlador (se ve que es solo una ganancia unitaria), en la que podemos ajustar el valor de la ganancia de este. En la parte inferior se pueden agregar los polos y ceros, tanto reales como complejos. Apretando click derecho aparecen las opciones.

También en esta sección se mostrarán polos y ceros del controlador ya creados. Para nuestro caso nos piden control por cancelación, por lo que tenemos que agregar un cero que cancele el polo de la planta, es decir, que se ubique en -10. También nos piden cero error en estado estacionario por lo que tenemos que agregar un integrador. Habiendo hecho esto, así se ve el lugar de la raíz.

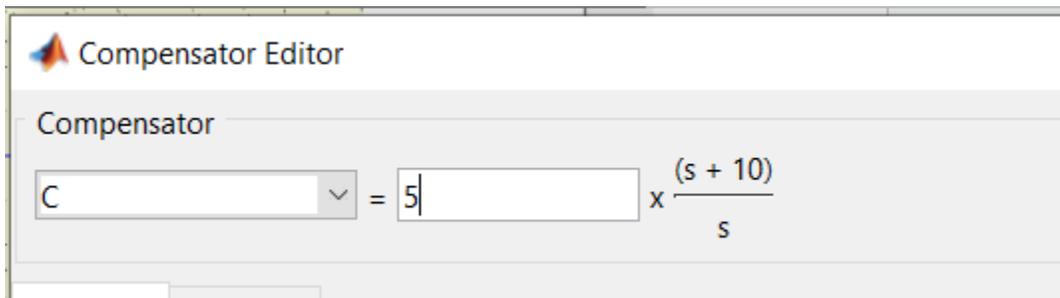


La “x” es el polo del integrador, mientras que “o” es el cero que está cancelando al polo de la planta. El cuadrado rosado es la ubicación del polo de lazo cerrado con la ganancia actual.

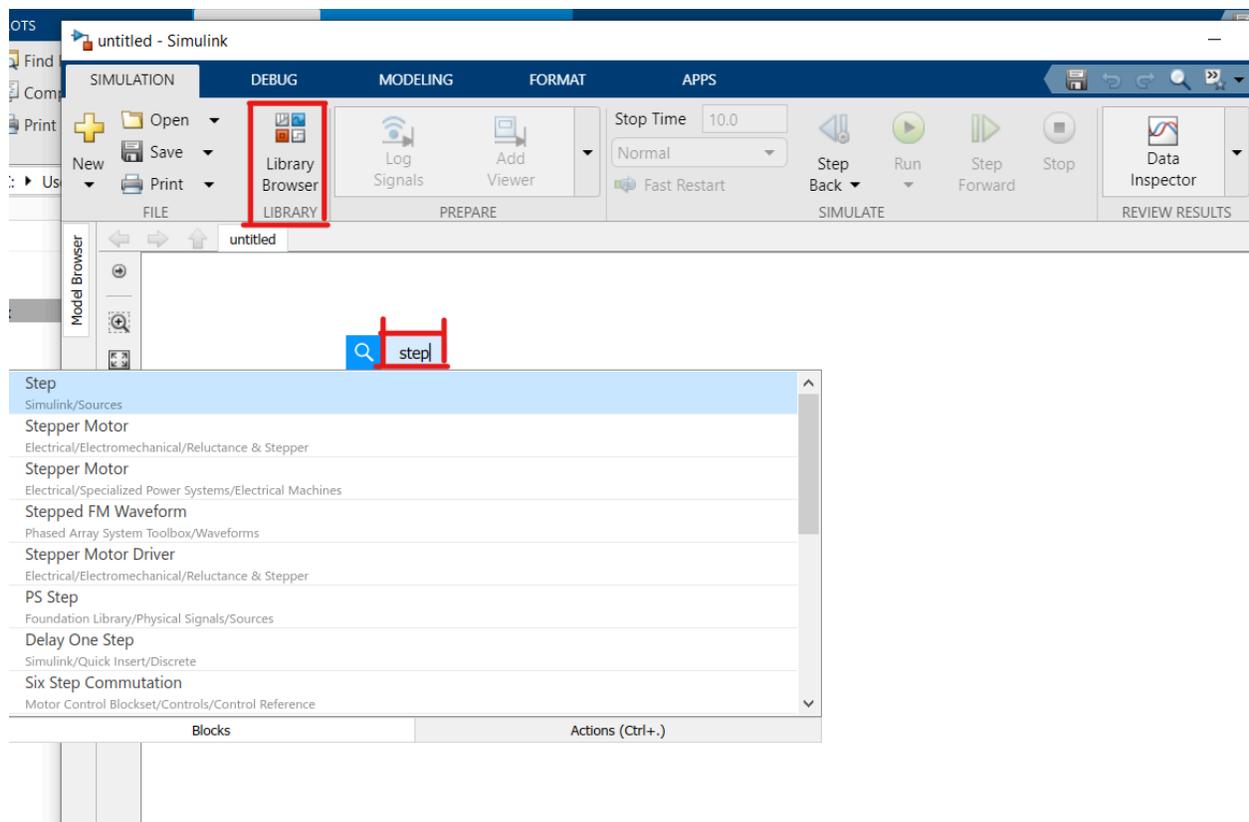
Podemos “agarrar” el polo y llevarlo al punto de diseño que nos indica el borde negro (frecuencia natural), ya que esto cambia la ganancia del controlador. También podemos hacer esto en el menú anterior. Si queremos una mejor vista podemos poner la grilla con click derecho en en LGR (también se pueden cambiar los límites de los ejes del LGR con click derecho -> Propiedades). Con esto el lugar de la raíz queda de la siguiente forma:



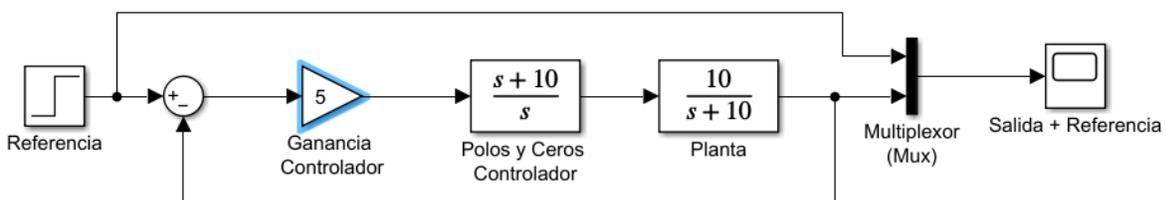
Volviendo al menú anterior vemos que con una ganancia de 5 el controlador llega al punto de diseño que elegimos. Con esto nuestro controlador queda de la forma $G_c(s) = 5 \cdot (s+10)/s$



Ahora para implementarlo en simulink, primero creamos un archivo en blanco. Luego con el "Library browser" podemos buscar los componentes que necesitamos para la simulación (se arrastran al canvas). También podemos hacer doble click en el canvas para poder buscar los componentes.

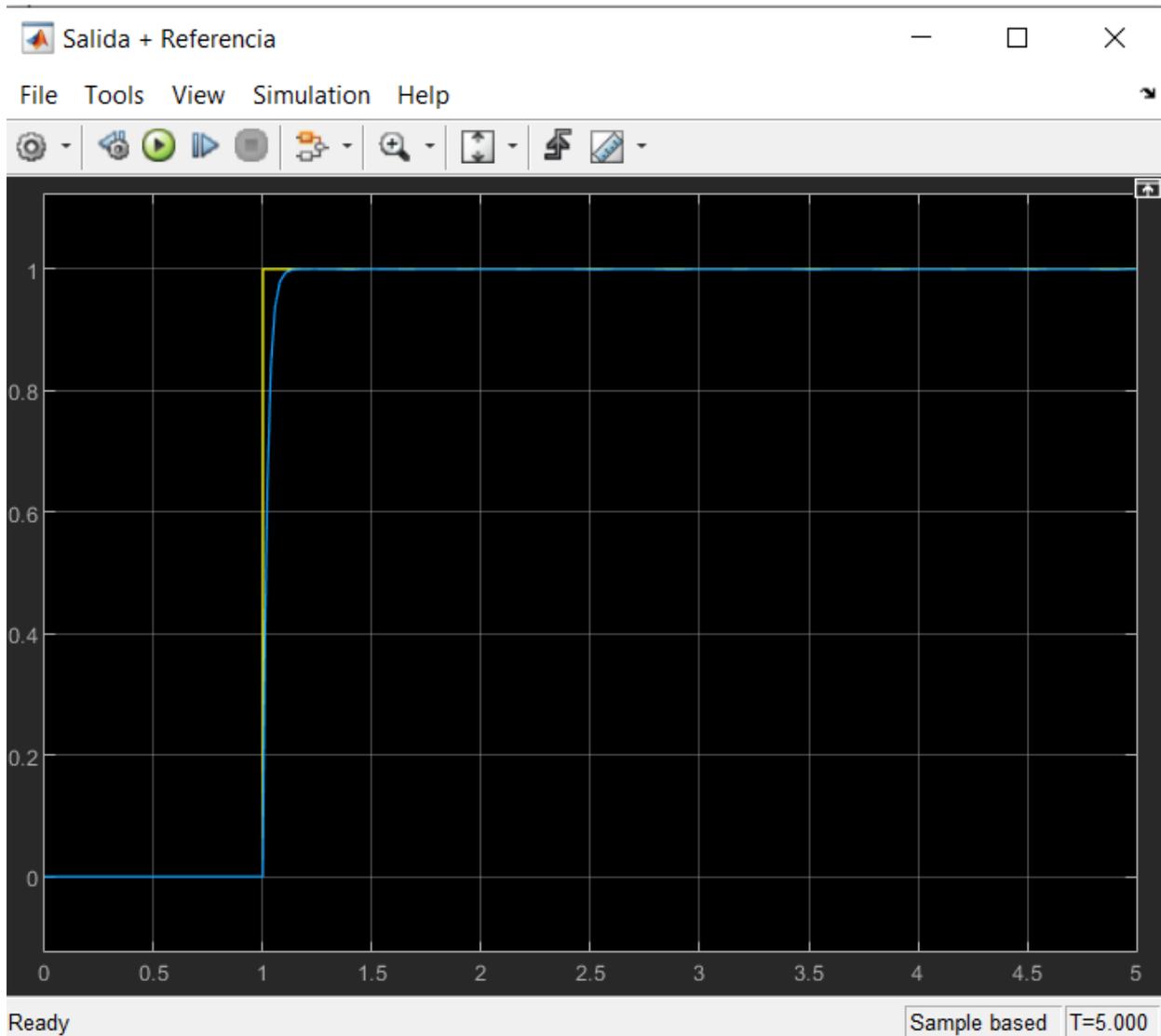


Vamos a necesitar un escalón (Step) para la referencia del sistema, un bloque suma/resta (Sum) para calcular el error del sistema, una ganancia (Gain) para la ganancia del controlador, dos funciones de transferencia (Transfer Fcn.) para el controlador y la planta y un Scope para ver la salida. Armamos el sistema para que quede de la siguiente forma (yo le agregue un Mux para que se vea la entrada y la salida):



También se puede agregar un bloque “To Workspace” para poder guardar los datos de simulación, conectando la salida del sistema a este bloque. De forma alternativa, se puede configurar el Scope para que guarde los datos, en la configuración de este, activando “Log data to workspace” (Block parameters -> tuerquita arriba a la izquierda -> Logging).

Corriendo la simulación por 5 segundos, el Scope nos debe entregar lo siguiente:



Podemos ver que la línea amarilla es la referencia mientras que la línea azul es la salida del sistema. Si eligen una frecuencia natural menor se debería demorar más el sistema en llegar a la referencia.

2) Grafique el LGR de la planta $G1(s)$ de la pregunta 2. Compruebe los puntos de corte al eje imaginario.

Para este caso, ya que tenemos los polos, ceros y ganancia del sistema vamos a utilizar `zpk()` para definir el sistema, lo que se muestra a continuación:

```
%Se define el sistema a partir de la ganancia y la ubicación de los polos y  
%ceros  
gain= 1;  
zeros = [];  
poles= [0,-2,-3];  
sys2= zpk(zeros,poles,gain);
```

Luego de la misma forma que en la parte 1 obtenemos el lugar de la raíz del sistema. Para verificar los límites de la planta simplemente “agarramos” los polos de lazo cerrado y los llevamos al límite. Después en “Edit compensator” pueden ver la ganancia límite. También se puede ver la respuesta del sistema en los gráficos de “Step response” que muestra `rltool()`.

Se adjuntan los archivos utilizados en el aux.