Universidad
Rey Juan Carlos

Área de Tecnología Electrónica

Práctica 3

Polos y ceros de una función de transferencia, respuesta transitoria de un sistema de control con MATLAB y LTIViewer, precisión y error en estado estacionario

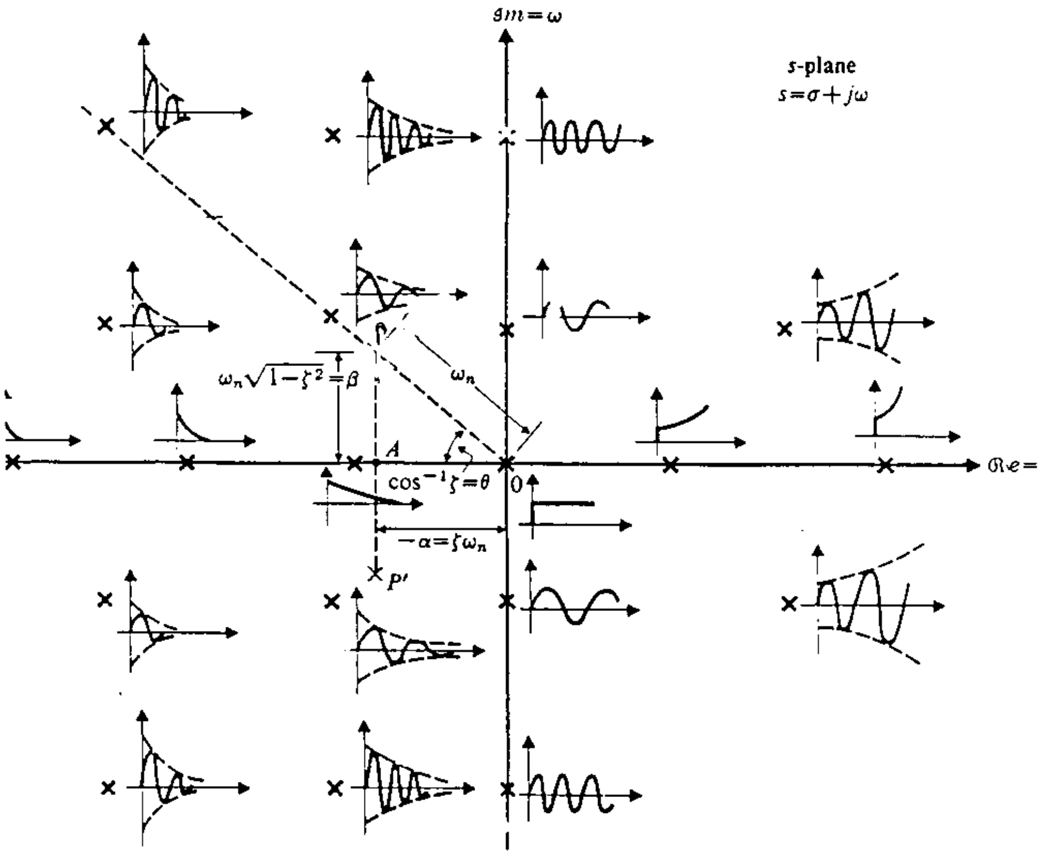
2022-2023

ÍNDICE

1. Polos y ceros de una función de transferencia con MATLAB

En las clases de teoría hemos visto que la respuesta temporal y(t) de un sistema de control depende esencialmente de la posición en el plano complejo (plano s) de los **polos (raíces del denominador)** de la función de transferencia en lazo cerrado $G_{LC}(s)$ y del tipo de entrada r(t) del sistema. Los ceros (raíces del denominador) también influyen en dicha respuesta, aunque en menor medida, ya que sólo afectan a los coeficientes de cada término en el dominio del tiempo.

En resumen, en la relación entre la respuesta transitoria y la posición de los polos viene dada por:



Existen diferentes maneras de extraer numéricamente los polos y los ceros de una función de transferencia G(s) mediante MATLAB. Los comandos más habituales son:

>> **polos = pole (G)**

>> **ceros = zero (G)**

Para dibujar el mapa de polos y ceros de un sistema de control en el plano complejo se utiliza el comando **pzmap(G)**. Al ejecutarlo, los polos aparecen representados con cruces (**x**) y los ceros con círculos (**o**).

- **pzmap (G1,G2,…)** dibuja los polos y ceros de varias funciones de transferencia en el mismo plan complejo, utilizando diferentes colores.

Con **pzmap** también se pueden extraer los valores numéricos de polos y ceros, agrupados en vectores columna, sin más que ejecutar el comando "*pzmap*" utilizando argumentos de salida de la siguiente manera:

>> [polos, ceros] = pzmap (G)

Ejercicio práctico 1. Extracción y dibujo de polos y ceros de funciones de transferencia

1. Extrae los valores numéricos de los polos y ceros de las siguientes funciones de transferencia:

$$G_1(s) = \frac{1}{s^2+4} \qquad G_2(s) = \frac{1}{s^4+4s^3+6s^2+4s+1}$$

$$G_3(s) = \frac{s^2+2s+1}{s^2-s} \qquad G_4(s) = \frac{s^2-1}{s^6+5s^5+11s^4+25s^3+34s^2+20s+24}$$

2. Para cada una de las funciones de transferencia, realiza una gráfica del plano complejo donde aparezcan dichos polos y ceros.

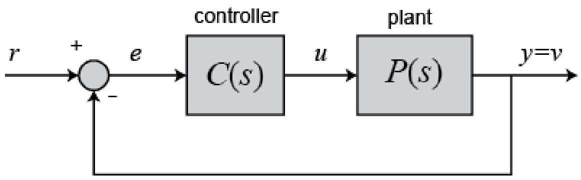
Nota: resuelve los ejercicios prácticos utilizando scripts (.m) o LiveScripts (.mlx) de MATLAB.

Ejercicio práctico 2. Polos y ceros en lazo abierto y en lazo cerrado en función de K

1. Realiza una gráfica del plano complejo con los polos y ceros de P(s):

$$P(s) = \frac{s+1}{(s-1)(s+2)(s+4)}$$

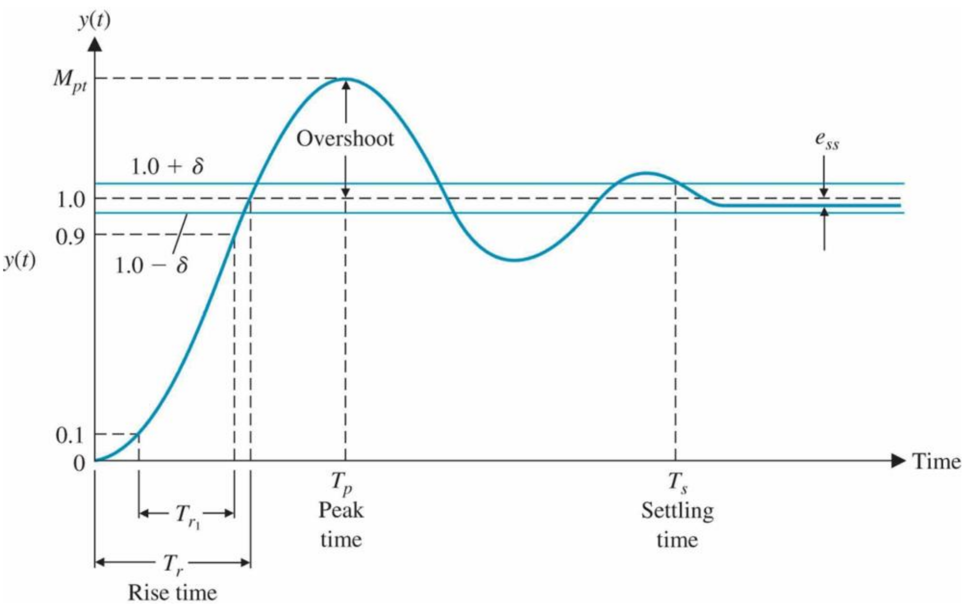
2. Realiza ahora una <u>única gráfica</u> de los polos y ceros de la función de transferencia en lazo cerrado $G_{LC}(s)$ del sistema de la figura, con la misma P(s) y con C(s) = K, para tres valores de la constante K, K=1, K=3 y K=10.



¿Influye el valor de K en la posición de los polos, de los ceros o de ambos en el sistema en lazo cerrado? Justifica tu respuesta

2. Estudio de la respuesta transitoria de sistemas de control

En este apartado vamos a aprender los comandos del *Control System Toolbox* de MATLAB que nos permiten conocer la respuesta temporal de un sistema de control y calcular los parámetros característicos del transitorio, que se han visto en teoría y que quedan resumidos en la siguiente gráfica:



2.1 Respuesta temporal a entradas impulso y escalón unitarios

El comando "*impulse*" nos permite visualizar gráficamente la respuesta de un sistema de control definido por una función de transferencia G(s) a una entrada impulso unitario, mientras que el comando "*step*" hace lo propio para una entrada escalón unitario:

>> **impulse (G)**

>> **step (G)**

MATLAB decide automáticamente la duración del intervalo de tiempo a mostrar, comenzando siempre desde t = 0. Si se quiere controlar la duración del intervalo de tiempo de interés para la visualización, su valor final debe aparecer como un argumento de entrada. Por ejemplo:

>> **step (G, 20)** % *Respuesta al escalón para un intervalo desde t = 0 s hasta t = 20 s.*

También se puede forzar un intervalo específico, con un tiempo inicial, final y paso temporal determinados, mediante:

>> **intervalo = 10:0.1:20;**

>> **step (G, intervalo)** % *Respuesta al escalón desde t = 10 s hasta t = 20 s en pasos de 0.1 s.*

En todos los casos anteriores se puede almacenar tanto el vector temporal como el vector de salida tanto de *"impulse"* como de *"step"* incluyendo argumentos de salida de la siguiente manera:

>> [y, t] = step (G) % *Almacenamiento de la curva de salida en los vectores y,t.*

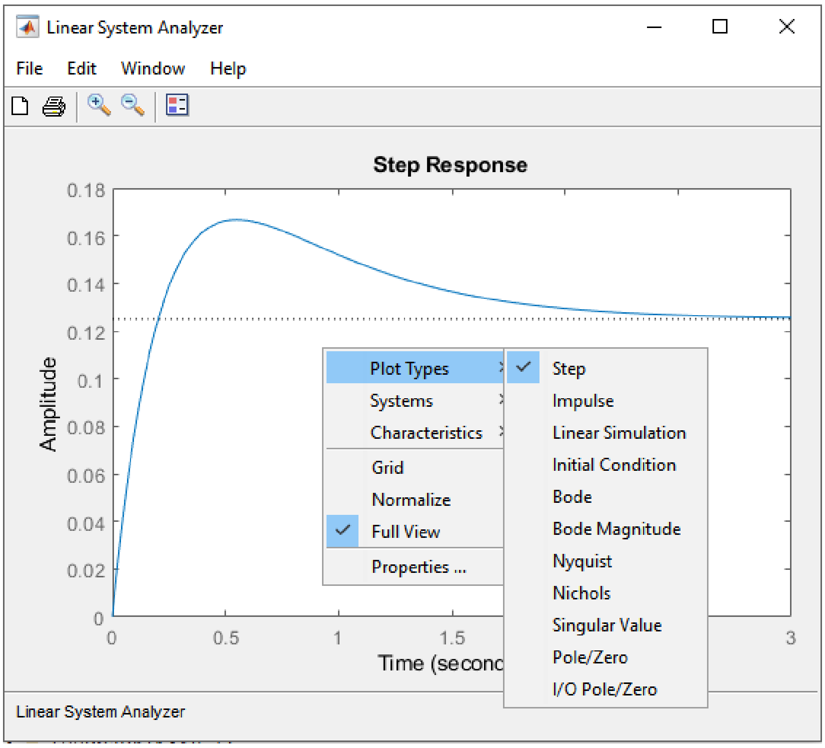
Por último, indicar que tanto *"impulse"* como *"step"* admiten varias funciones de transferencia como argumentos. Por ejemplo:

>> impulse (G1, G2, G3) % *Respuesta al impulso unitario de G1, G2 y G3*

Ejercicio práctico 3. Respuesta temporal de sistemas de primer orden

1. Construye cinco funciones de transferencia de primer orden situando sus polos en s = -0.5, s=-1, s = -3, s = -5 y s = -10 respectivamente.
2. Realiza una gráfica que incluya la respuesta al impulso unitario de todas ellas hasta un valor final de t = 5 s. Incluye la leyenda. Justifica el resultado obtenido.
3. Realiza ahora una gráfica que incluya la respuesta al escalón unitario de todas ellas hasta un valor final de t = 10 s. Incluye la leyenda. Justifica el resultado obtenido.
4. Modifica la ganancia estática de cada una de las funciones anteriores para la respuesta al escalón de todas ellas tenga el mismo valor final igual a 1.

Una forma alternativa de acceder a la gráfica tanto al mapa de polos y ceros como a la respuesta temporal de un sistema es a través del *"Linear System Analyzer"* de MATLAB:



Para acceder al "*Linear System Analyzer*" de MATLAB se debe usar el siguiente comando:

>> ltiview (G)

Una vez abierta la ventana del analizador, se puede modificar la visualización mostrada por defecto haciendo clic con el botón derecho del ratón, y cambiando la selección en el menú "*Plot Types*".

Ejercicio práctico 4. Respuesta temporal de sistemas de segundo orden

1. Construye cuatro funciones de transferencia de segundo orden de tal manera que todas tengan una K = 1, una frecuencia natural de 2 rad/s y una coeficiente de amortiguamiento igual a 0, 0.25, 1 y 5, respectivamente.
2. Extrae el valor numérico y realiza una única gráfica con la posición de los polos y ceros de las cuatro funciones de transferencia.
3. Realiza una gráfica que incluya la respuesta al escalón unitario de todas ellas hasta un valor final que permita que todas las respuestas se vean con comodidad. Incluye la leyenda. Indica el *tipo* de cada una de las respuestas temporales (*oscilatoria, subamortiguada, con amortiguamiento crítico o sobreamortiguada*).
4. Calcula manualmente la posición de los polos para cada uno de los tipos de respuesta temporal al escalón, comprueba tus resultados y justifica las respuestas.

   Nota: mediante el comando **ord2()** también se pueden definir funciones de transferencia de segundo orden. Consulta su sintaxis mediante **doc ord2**.

2.2 Extracción de parámetros característicos de la respuesta temporal

Existen dos opciones principales para extraer los parámetros característicos que definen la respuesta temporal a una entrada escalón unitario, que son:

- Tiempo de asentamiento o establecimiento (*settling time*, t_s).
- Tiempo de subida (*rise time*, t_r)
- Valor final del estacionario, $y(\infty)$

En caso de respuestas subamortiguadas, a los anteriores se añaden:

- Sobreelongación (*overshoot*, M_p)
- Tiempo de pico (*peak time*, t_p)

Para acceder a dichos datos numéricamente debemos hacer uso del comando "*stepinfo*" de MATLAB.

>> respuesta_G = stepinfo (G)

De esa manera, la salida del comando se guarda en la variable "*respuesta_G*", que es de tipo "*struct*" (*structure array* o array de estructura). Un **array de estructura** es un tipo de datos de MATLAB que agrupa datos relacionados mediante contenedores de datos denominados **campos**. Cada campo puede contener cualquier tipo de datos.

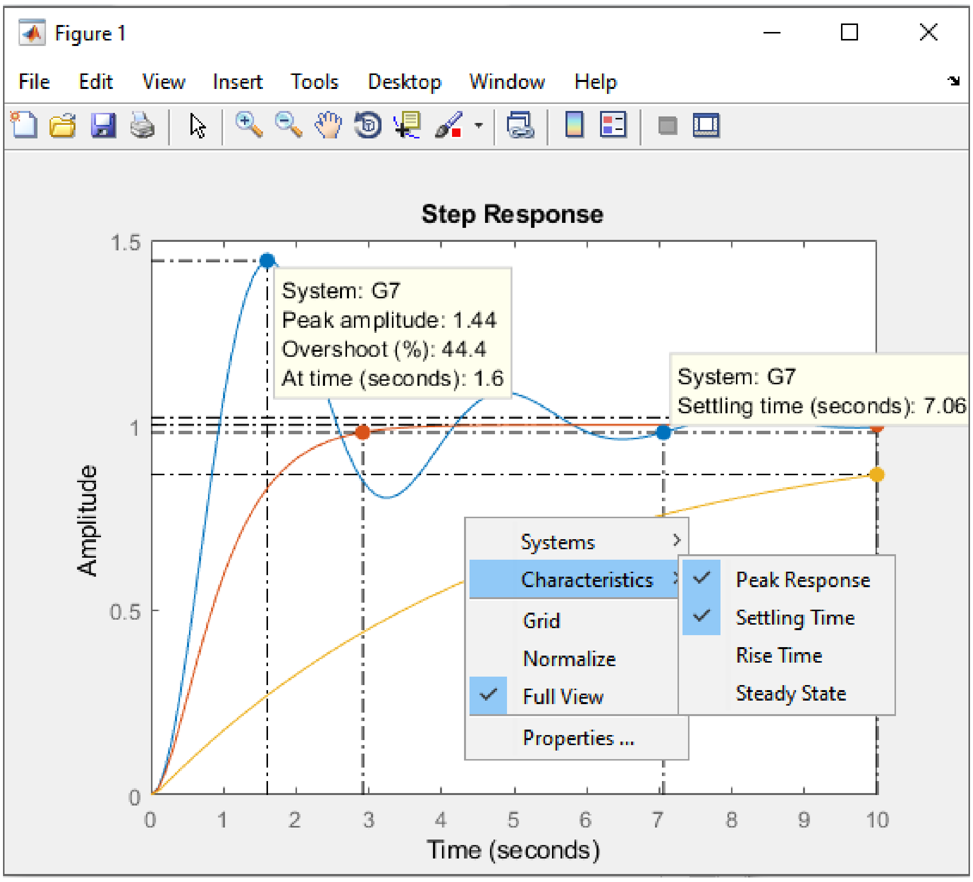
Para acceder a los datos almacenados en un campo de un array de estructura se debe utilizar la notación de puntos similar a la usada en programación orientada a objetos, es decir:

variable = structName.fieldName

Se puede llamar al comando stepinfo utilizando los argumentos '*SettlingTimeThreshold*' y/o '*RiseTimeThreshold*' que permiten modificar la definición del tiempo de establecimiento o del tiempo de subida, respectivamente. Por ejemplo, para calcular el tiempo de establecimiento en una banda del 0.5% y el tiempo de subida del 5 al 95% haríamos:

>> stepinfo(G, 'SettlingTimeThreshold', 0.005, 'RiseTimeThreshold', [0.05 0.95])

Aparte del comando "*stepinfo*", los parámetros característicos de la respuesta temporal de un sistema se pueden extraer de la gráfica de la respuesta a un escalón unitario o del "*Linear System Analyzer*", haciendo clic con el botón derecho del ratón sobre la gráfica, como se observa en la siguiente figura:



Ejercicio práctico 5. Extracción de los parámetros de la respuesta temporal

1. Extrae los valores numéricos del tiempo de subida y del tiempo de asentamiento para los sistemas de primer orden del ejercicio práctico 3, guardándolos en variables independientes. Comprueba dichos valores con los extraídos de la gráfica de la respuesta al escalón unitario
2. Repite el apartado anterior para las cuatro funciones de transferencia de sistemas de segundo orden especificadas en el ejercicio práctico 4.
3. Contrasta algunos tiempos de asentamiento con los valores teóricos esperados.

Ejercicio práctico 6. Diseño de sistemas de control

1. Utilizando comandos de control de flujo (*for, while, if*) realiza un script de MATLAB que calcule el valor que debe tener el polo de un sistema de primer orden para que su tiempo de subida (del 10 al 90%) sea igual a 1 ms.
2. Repite el apartado anterior para extraer la frecuencia natural de un sistema de segundo orden con respuesta críticamente amortiguada que permite que el tiempo de asentamiento del sistema sea igual a 10 ms.

2.3 Respuesta a otro tipo de entradas: rampa unitaria, aceleración unitaria, etc.

Para estudiar la respuesta de un sistema de control a una entrada de cualquier otro tipo (rampa unitaria, aceleración unitaria, suma de diferentes entradas, etc.) tenemos a nuestra disposición un comando muy importante, denominado "*lsim*". La sintaxis más habitual de "*lsim*" es la siguiente:

>> lsim (G, entrada, tiempo)

Donde:

1. "tiempo" es un vector que debe contener el intervalo de valores temporales equiespaciados para la simulación, definido por ejemplo de la forma:

   >> tiempo = 0 : dt : Tfinal

2. "entrada" es un vector que debe contener los valores de la señal de entrada para cada uno de los datos temporales del vector tiempo. Por ejemplo, para una señal de entrada de tipo aceleración unitaria, tendríamos:

   >> aceleracion = tiempo.^2

Por último, comentar que "*lsim*":

- Permite simular la respuesta de varios sistemas de control simultáneamente (en la misma gráfica).
- Se puede extraer el valor numérico de la respuesta especificando argumentos de salida de la misma manera que se vio con "*step*".

Ejercicio práctico 7. Respuesta de sistemas de 1º orden a entradas rampa y aceleración

1. Realiza una gráfica con la respuesta temporal de los sistemas de primer orden del ejercicio práctico 3 (una vez ajustadas las K) a una entrada rampa unitaria.
2. Repite el apartado anterior para una rampa de pendiente igual a 3.
3. Realiza ahora una gráfica con la respuesta temporal de dichos sistemas a una entrada aceleración unitaria

Ejercicio práctico 8. Respuesta de sistemas de 2º orden a entradas rampa y aceleración

1. Repite los tres puntos del apartado anterior para las cuatro funciones de transferencia de sistemas de segundo orden especificadas en el ejercicio práctico 4.

3. Respuesta transitoria de sistemas de orden superior

Hasta ahora se ha estudiado la respuesta temporal de sistemas de primer y segundo orden (es decir, con uno o dos polos en lazo cerrado y sin ceros). Bajo determinadas condiciones, los sistemas de orden superior (con tres o más polos) y/o los sistemas con ceros **se comportan como un sistema de primer o de segundo orden sin ceros**, donde su dinámica viene determinada por sólo uno o dos polos, denominados "**polo o polos dominantes"**, que son los que encuentran más cerca del eje imaginario (polos lentos).

En este caso se suele hablar de "aproximación mediante polos dominantes", y la respuesta temporal de dicho sistema se puede aproximar por la de un sistema de primer o de segundo orden, sin ceros, sólo con el polo o los dos polos dominantes.

3.1 Respuesta de un sistema de tercer orden. Polos dominantes

¿Qué condiciones se tienen que dar en un sistema de tercer orden para que se pueda realizar la aproximación de su respuesta temporal mediante los polos dominantes?

Vamos a estudiarlo mediante un ejercicio práctico:

Ejercicio práctico 9. Respuesta de sistemas de 3º orden. Polos dominantes

1. Construye una función de transferencia de primer orden G1(s) donde la posición de su único polo sea configurable mediante la variable P (s = -P).

$$G_1(s) = \frac{1}{(1+\frac{s}{P})} = \frac{P}{(s+P)}$$

2. Construye la siguiente función de transferencia de segundo orden G2(s). Extrae la posición (fija) de sus dos polos:

$$G_2(s) = \frac{10}{s^2 + 2s + 10}$$

3. Define una nueva función de transferencia de tercer orden, G3(s), como el producto (cascada) de G1·G2. Realiza dos gráficas, por un lado una con posición de los polos de G1 y G2 (que serán los polos de G3) y por otro la respuesta al escalón de las tres funciones de transferencia, G1, G2 y G3. Activa la leyenda de la gráfica.

4. Da valores a P desde 0,1 hasta 20 y relaciona la respuesta al escalón de G3 con las respuestas al escalón de G1 y G2, a través de la posición de sus polos. ¿Qué polos dominantes tiene G3 en cada caso? Razona tu respuesta

5. ¿A partir de qué valor de P podrías decir que la respuesta de G3 es equivalente a la de G2, es decir, que el sistema de tercer orden se comporta como si tuviera sólo dos polos dominantes (los de G2)? Coteja tu respuesta en la bibliografía de la asignatura.

Ejercicio práctico 10. Respuesta de sistemas de 4º orden

1. En base al ejercicio anterior, añade a la función G3(s) del ejercicio anterior otro polo simple en posición s = -P2, construyendo una nueva función de 4º orden G4 = G1·G1b*G2 y comprueba si tus conclusiones del ejercicio anterior siguen siendo válidas para un sistema de 4º orden:

$$G_{1b}(s) = \frac{1}{(1+\frac{s}{P_2})} = \frac{P_2}{(s+P_2)}$$

4. Precisión y error en estado estacionario de un sistema de control

4.1 Error de posición en estado estacionario

Para calcular el error verdadero en estado estacionario de un sistema de control utilizando MATLAB es necesario:

a) Comprobar que el sistema es estable. El concepto de error en estado estacionario sólo tiene sentido para que tienen una respuesta temporal que se estabiliza, es decir, para sistemas con todos sus polos en el semiplano complejo negativo.

b) Calcular la diferencia entre la entrada del sistema y la salida en estado estacionario, en términos absolutos o relativos

Por ejemplo, para calcular el error verdadero de posición en estado estacionario ante una entrada escalón de una amplitud determinada, para un sistema cuya función de transferencia en lazo cerrado viene dada por GLC, debemos hacer:

>> Amplitud = 2; *% Amplitud (configurable) del escalón*

>> [y,t] = step (Amplitud * GLC, t_final) *% Almacenamos respuesta hasta t_final en el vector y*

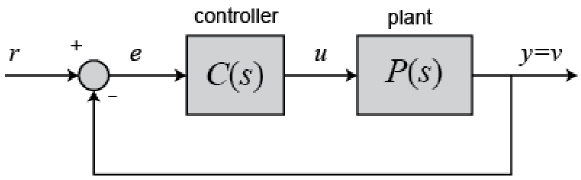
>> ess_pos_abs = Amplitud – y(end) *%Error absoluto, con su signo*

>> ess_pos_rel = 100*(Amplitud –y(end)) / Amplitud *% Error relativo, en porcentaje, con signo*

Para obtener valores de los errores absoluto y relativo precisos, el tiempo t_final de evaluación del escalón debe ser sensiblemente superior al tiempo de asentamiento (comienzo del estacionario) del sistema.

Ejercicio práctico 11. Precisión y error de posición en estado estacionario

1. Construye el sistema en lazo cerrado de la figura, donde C(s) es igual a 1 y P(s) es un sistema sin ceros y con dos polos en s= -3 y s= -1



2. ¿De qué TIPO es el sistema? Calcula su constante de error de posición y el error verdadero de posición en estado estacionario con las expresiones vistas en teoría.
3. Comprueba dichos cálculos extrayendo los errores de posición absoluto y relativo utilizando MATLAB, mediante un escalón de amplitud igual a 2.
4. Repite el ejercicio asumiendo que ahora C(s) es igual a 50. ¿Ha cambiado el error de posición? ¿Se ha visto afectada la respuesta transitoria?

4.2 Tipos de sistemas frente al error. Error en estado estacionario de velocidad

En las clases de teoría hemos visto que, con respecto al error en estado estacionario, los sistemas de control se pueden definir por su TIPO, que viene dado por el número de polos en el origen de su función de transferencia en lazo abierto G(s)H(s) (error de control) o G(s) error verdadero:

$$\lim_{s\to 0} G(s).H(s) = \lim_{s\to 0} \frac{K.\prod_{i=1}^{z}\left(\frac{s}{z_1}+1\right)}{s^n.\prod_{j=1}^{p}\left(\frac{s}{z_p}+1\right)} = \lim_{s\to 0} \frac{K}{s^n}$$

Valor de "n" = Tipo del sistema

El error verdadero del sistema depende de la entrada según su tipo, como se resume en la siguiente tabla:

	Constantes			Error para diferentes entradas		
Tipo	k_p	k_v	k_a	Salto	Rampa	Parábola
0	K	0	0	M/(1+K)	∞	∞
1	∞	K	0	0	M/K	∞
2	∞	∞	K	0	0	M/K

Ejercicio práctico 12. Aumento del tipo de sistema frente al error.

1. Repite el ejercicio 11 introduciendo en C(s) las modificaciones necesarias que hagan aumentar en una unidad el TIPO del sistema frente al error.
2. Vuelve a calcular el error de posición con MATLAB y justifica el resultado obtenido

Ejercicio práctico 13. Precisión y error de velocidad en estado estacionario

1. Calcula ahora la constante de precisión de velocidad y el error verdadero en estado estacionario de velocidad del sistema que has definido ejercicio 12, mediante las expresiones teóricas.
2. Utilizando el comando “lsim” y la metodología de cálculo aprendida para los errores de posición, determina el error en estado estacionario de velocidad utilizando MATLAB
3. Realiza la gráfica de la respuesta temporal del sistema ante una entrada de velocidad y explica los resultados obtenidos.
4. ¿En qué unidades crees que debe darse el error de velocidad absoluto?