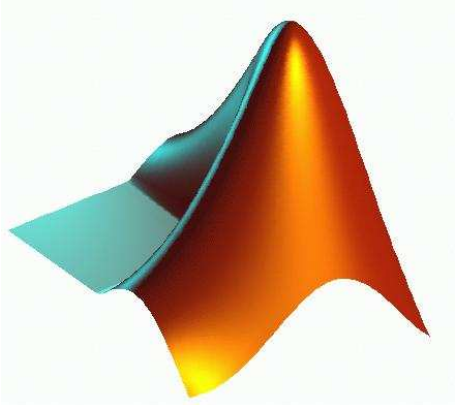
ME55A

Matlab y Control Automático



R. H. Hernández
Depto. Ing. Civil Mecánica - U. de Chile

13 de mayo de 2010

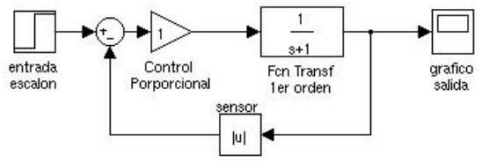


Figura 1: Sistema de lazo cerrado creado en Simulink. La señal de referencia es un escalón unitario. El controlador es proporcional. La planta es de primer orde y el sensor entrega el valor absoluto de la salida.

Controlador PID

Detalles

La función del transferencia del controlador PID se puede escribir como:

$$K_P + \frac{K_I}{s} + K_D s = \frac{1}{s} K_D s^2 + K_P s + K_I$$

donde K_P, K_I, K_D son la ganancia Proporcional, la ganancia Integral y la ganancia Derivativa respectivamente. Primero veamos cómo el controlador de PID trabaja dentro de un sistema de lazo-cerrado que usa el esquema anterior. La variable e representa el error, la diferencia entre el valor de entrada deseado R y el valor de salida actual Y. Esta señal de error e se enviará al controlador PID, y el controlador calcula la derivada y la integral de esta señal de error. La señal u que sale del controlador es luego igual a:

$$u = K_P\, e + K_I \int e\, dt + K_D \frac{de}{dt}$$

Esta señal u se enviará a la planta, y se obtendrá una nueva salida Y actualizada. El controlador tomará sistematicamente los nuevos errores para calcular una nueva salida Y, hasta converger a un error mínimo.

Caraterísticas de controladores P, I y D

Un controlador proporcional K_P producirá el efecto de reducir el tiempo de subida t_r y además reducirá el error estacionario, aunque no podrá eliminarlo completamente. Un control integral K_I producirá el efecto de eliminar el error estacionario, pero puede empeorar la respuesta transiente del sistema. Un control derivativo K_D tendrá el efecto de aumentar la estabilidad del sistema, reduciendo el overshoot, y mejorando la respuesta transiente. Los efectos de cada uno de los controladores K_P, K_D, y K_I en un sistema de lazo cerrado se resumen a continuación.

Resp.	T_r	M_p	t_s	Error $e(\infty)$
K_P	Disminución	Aumento	cambio chico	Disminución
K_I	Disminución	Aumento	Aumento	Elimina
K_D	cambio chico	Disminución	Disminución	cambio chico

Note que estas correlaciones no son siempre precisas, porque K_P, K_I, y K_D no necesariamente son independientes entre ellos. De hecho, al cambiar una de estas variables puede que cambie el efecto de las otras dos.

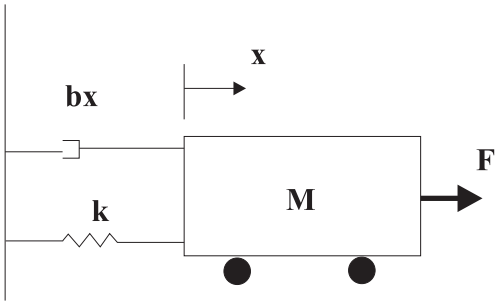


Figura 2:

Por esta razón, esta tabla debe usarse sólo como referencia cuando usted esté ajustando los valores K_I, K_Py K_D. Lo más apropiado es ajustar los valores de acuerdo al criterio de Zieger&Nichols.

Un ejemplo simple

Suponga que nosotros tenemos una masa conectada a un resorte de constante k y a un amortiguador de constante b.
La ecuación del este sistema es

$$m\ddot{x} + b\dot{x} + kx = f \tag{1}$$

Tomando la TL de la ecuación 1

$$ms^2X(s) + bsX(s) + kX(s) = F(s) \tag{2}$$

La función de transferencia entre el desplazamiento $X(s)$ y la entrada $F(s)$ es

$$\frac{X(s)}{F(s)} = \frac{1}{ms^2 + bs + k}$$

Sea $m = 1$ [kg], $b = 10$ [N.s/m], $k = 20$ [N/m] y $F(s) = 1$, entonces la función de transferencia es:

$$\frac{X(s)}{F(s)} = \frac{1}{s^2 + 10s + 20}$$

El objetivo de este problema es mostrar cómo influye o contribuye cada uno de los coeficientes K_P, K_Iy K_D en la obtención de : (a) Un tiempo de subida corto, (b) un overshoot mínimo y (c) un error estacionario nulo.

Respuesta de lazo abierto

Primero veamos la respuesta de lazo abierto. En MATLAB, debe crear un nuevo archivo *m-file* y escriba en él el código siguiente: $num = 1; den = [1\,10\,20]; elstep(num, den)$.
Al ejecutar este archivo en Matlab, la ventada gráfica debe mostrarle el gráfico de la figura 3:
La ganancia DC de la función de transferencia de la planta es $1/20$, de manera que $0{,}05$ es el valor final de la salida como respuesta a un escalón unitario. Esto corresponde a un error estacionario de $0{,}95$, de hecho, bastante grande no es cierto ?. Además, el tiempo de subida es aproximadamente un segundo, y el tiempo de

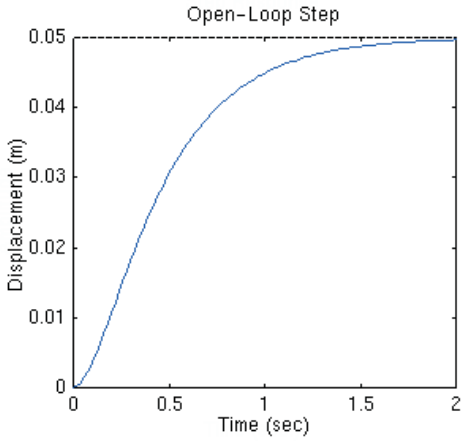


Figura 3:

establecimiento es aproximadamente $1{,}5$segundos. Tratemos de diseñar un controlador que reduzca el tiempo de subida, que reduzca el tiempo del establecimiento, y que elimine el error estacionario.

Controlador Proporcional

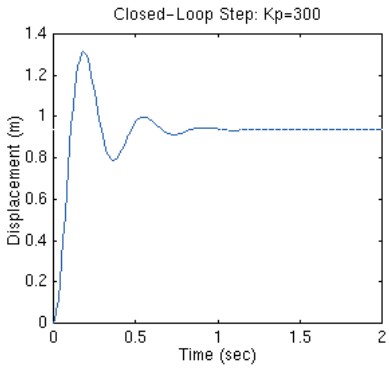
En la tabla anterior vimos que el controlador K_P proporcional reduce el tiempo de subida, aumenta el overshoot, y reduce el error estacionario. La función de transferencia de lazo cerrado con un controlador proporcional es la sgte:

$$\frac{X(s)}{F(s)} = \frac{K_P}{s^2 + 10s + (20 + K_P)}$$

Imponga una ganancia proporcional $K_P = 300$ y cambie el archivo Matlab por lo siguiente:

- $Kp = 300; num = [K_P]; den = [1\ 10\ 20 + K_P];$
- $t = 0 : 0{,}01 : 2; step(num, den, t);$

Ejecutando este archivo en Matlab la ventana gráfica le mostrará el gráfico de la figura .



Nota: La función de Matlab llamada $cloop$ puede usarse para obtener la función de transferencia de lazo abierto directamente a partir de la función de transferencia de lazo cerrado (en lugar de obtener la función de transferencia de lazo cerrado a la mano). El archivo Matlab siguiente usa el comando $cloop$ y esto debe producir un gráfico idéntico a anterior:

- $num = 1; den = [1\,10\,20]; Kp = 300;$
- $[numCL, denCL] = cloop(K_P * num, den);$
- $t = 0:0{,}01:2; step(numCL, denCL, t);$

El gráfico anterior muestra que el controlador proporcional reduce a la vez el tiempo de subida y el error estacionario, aumenta el overshoot y además hace decrecer el tiempo de establecimiento en un porcentagealgo insignificante.

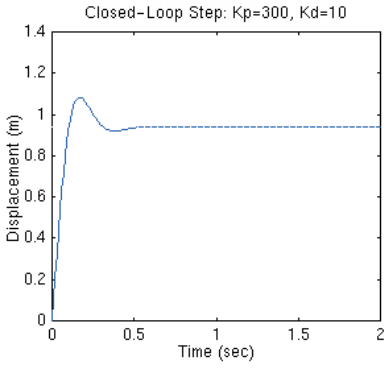
Controlador Proporcional-Derivativo

Ahora, echemos una mirada a un controlador PD. De la tabla anterior vemos que el controlador derivativo K_D reduce a la vez, el overshoot y el tiempo de establecimiento. La función de transferencia de lazo cerrado del sistema con un controlador PD es:

$$\frac{X(s)}{F(s)} = \frac{K_D s + K_P}{s^2 + (10 + K_D)s + (20 + K_P)}$$

Imponga $K_P = 300$ como antes y haga $K_D = 10$. Ingrese los comandos sgtes en un archivo Matlab y ejecútelo:

- $K_P = 300; K_D = 10; num = [K_D K_P];$
- $den = [1\,10 + K_D\,20 + K_P]; t = 0:0{,}01:2; step(num, den, t)$



Este gráfico de la figura muestra que el controlador derivativo redujo a la vez el overshoot y el tiempo de establecimiento, y tuvo un efecto pequeño en el tiempo de subida y en el error estacionario.

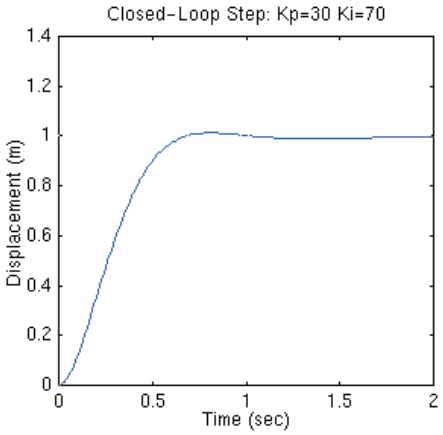


Figura 4:

Control Proporcional-Integral

Antes de usar un controlador PID, echemos una mirada a un controlador PI. De la tabla anterior vemos que el controlador integral K_I disminuye el tiempo de subida, aumenta a la vez el overshoot y el tiempo de establecimiento, además elimina el error estacionario. Entonces, para nuestro sistema, la función de transferencia en lazo cerrado con un control PI es:

$$\frac{X(s)}{F(s)} = \frac{K_P s + K_I}{s^3 + 10s^2 + (20 + K_P)s + K_I}$$

Reduzcamos el K_P a 30, y hagamos K_I igual a 70. Cree un nuevo archivo Matlab e ingrese los comandos sgtes:

- $K_P = 30; K_I = 70; num = [K_P K_I]; den = [1\,10\,20 + K_P\,K_I];$
- $t = 0 : 0{,}01 : 2; step(num, den, t)$

Ejecute este archivo. Ud. debe conseguir el gráfico de la figura 4.
Hemos reducido la ganancia proporcional K_P porque el controlador integral también reduce el tiempo de subida y aumenta el overshoot como los hace el controlador proporcional (efecto doble). Las respuestas anteriores indican que el controlador integral eliminó el error estacionario.

Control Proporcional-Integral-Derivativo

Echemos una mirada a un controlador PID. La función de transferencia de lazo cerrado con control PID es:

$$\frac{X(s)}{F(s)} = \frac{K_D s^2 + K_P s + K_I}{s^3 + (10 + K_D)s^2 + (20 + K_P)s + K_I}$$

Después de varios ensayos (inténtelo), las ganancias $K_P = 350$, $K_I = 300$, y $K_D = 50$ nos entregna la respuesta deseada. Para confirmarlo, ingrese los comandos sgtes a un archivo Matlab y ejecútelo. Debe obtener la repuesta al escalón ($step$) de la figura 5.

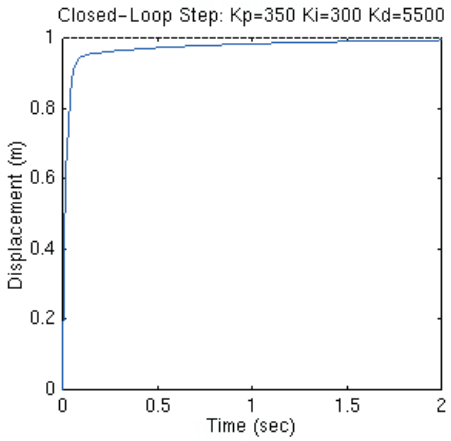


Figura 5:

- $K_P = 350; K_I = 300; K_D = 50; num = [K_D\, K_P\, K_I];$
- $den = [1\, 10 + K_D\, 20 + K_P\, K_I]; t = 0 : 0{,}01 : 2; step(num, den, t);$

Hemos obtenido un sistema de control sin overshoot, con tiempo de subida rápido y sin error estacionario.

Ideas generales para diseñar un controlador PID

Cuando quiera diseñar un controlador PID para un sistema dado, siga los pasos siguientes para obtener una respuesta deseada.

- 1.Obtenga una respuesta de lazo abierto y determine lo que necesita ser mejorado
- 2.Agregue un control proporcional para mejorar el tiempo de subida.
- 3.Agregue un control derivativo para mejorar el overshoot
- 4.Agregue un control integral para eliminar el error estacionario.
- 5.Ajuste cada una de las ganancias K_P, K_I, y K_D hasta que obtenga una respuesta global óptima (Ziegler&Nichols).

Por último, por favor tenga presente que no necesita implementar los tres controladores en un mismo sistema siempre (proporcional,derivativo, e integral) si no estrictamente necesario. Por ejemplo, si un controlador PI da un buen resultado (como el ejemplo anterior), entonces usted no necesita usar un cntrol derivativo. Mantenga el controlador tan simple como sea posible.

LGR

Estudiamos aquí el método de diseño: Lugar Geométrico de la Raíces (LGR). Los comandos de Matlab más importantes que se usarán en esta guía son: cloop, rlocfind, rlocus, sgrid, step.

Polos del lazo cerrado

El LGR (root locus, en inglés) de una función de transferencia $G(s)$ (en loop abierto) corresponde a todos los posibles polos de lazo cerrado usando un controlador de ganancia K con retro–alimentación unitaria (figura 6).

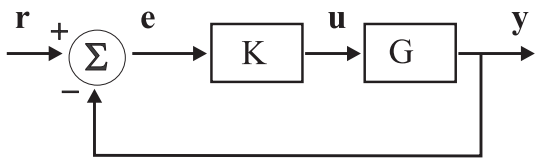


Figura 6: Sistema de lazo cerrado. Señal de referencia r, Controlador proporcional K, Planta G y salida y

La función de transferencia de lazo cerrado es:

$$\frac{Y(s)}{R(s)} = \frac{KG(s)}{1 + KG(s)}$$

y así los polos del sistema de lazo cerrado son valores de s tal que

$$1 + KG(s) = 0.$$

Si escribimos $G(s) = b(s)/a(s)$, entonces esta ecuación tiene la forma:

$$\begin{aligned} a(s) + Kb(s) &= 0 \\ \frac{a(s)}{K} + b(s) &= 0 \end{aligned}$$

Sea n el orden de $a(s)$ y m el orden de $b(s)$ (el orden de un polinomio es la potencia más alta de s que aparece en él).
Nosotros consideraremos todos los valores positivos de K. En el límite $K \to 0$, los polos del sistema de lazo cerrado son $a(s) = 0$ o los polos de $G(s)$. En el límite $K \to \infty$, los polos del sistema de lazo cerrado son $b(s) = 0$ o los ceros de $G(s)$.
Independientemente del valor de K, el sistema de lazo cerrado siempre debe tener n polos, donde n es el número de polos de $G(s)$. El LGR debe tener tantas ramas como polos tenga $G(s)$. Cada rama parte desde un polo de $G(s)$ y va hacia un cero de $G(s)$. Si $G(s)$ tiene más polos que ceros (como es a menudo el caso), $m < n$ y decimos que $G(s)$ tiene ceros en el infinito. En este caso, el límite de $G(s)$ con $s \to \infty$ es cero. El número de ceros en el infinito es $n - m$, el número de polos menos el número de ceros, y corresponde al número de ramas del LGR que van a infinito, osea las asíntotas.
Puesto que el LGR es el lugar geométrico de todos los posibles polos de lazo cerrado, nosotros podemos seleccionar una ganancia tal que nuestro sistema de lazo cerrado cumpla con los requerimientos que nosotros le impongamos. Si cualquiera de los polos seleccionados está en el lado derecho del plano complejo, el sistema

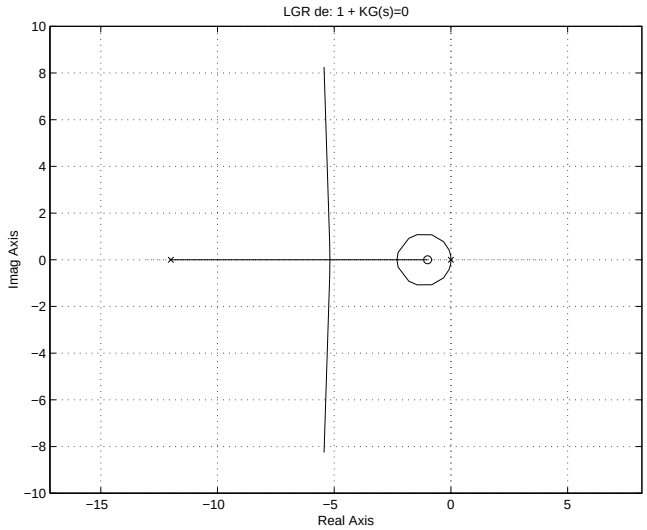


Figura 7: LGR, uso de *rlocus*

será inestable. Los polos más cercanos al eje imaginario tienen la influencia más grande en la respuesta de lazo cerrado, con lo cuál si el sistema tiene tres o cuatro polos, se puede comportar como un sistema de segundo orden o incluso de primer orden dependiendo de la ubicación del polo (s) dominante.

Graficamos el LGR

Considere un sistema de lazo abierto que tiene una función de transferencia $G(s)$:

$$G(s) = \frac{(s+1)}{s^2(s+12)}$$

Diseñemos el sistema de control usando el LGR. Digamos que nuestro criterio es $5\,\%$ de overshoot, y tiempo de subida $t_r = 1/2$ segundo. Creamos un archivo de Matlab de nombre $lgr.m$. Ingresamos la función de transferencia, y el comando para trazar el LGR:

- $num = [1\ 1]$;
- $den = [1\ 12\ 0\ 0]$
- $rlocus(num, den)$
- $title('LGRde : 1 + KG(s)')$
- $axis('equal')$

Si ejecutamos el script lgr.m, obtendremos el gráfico de la figura 7.

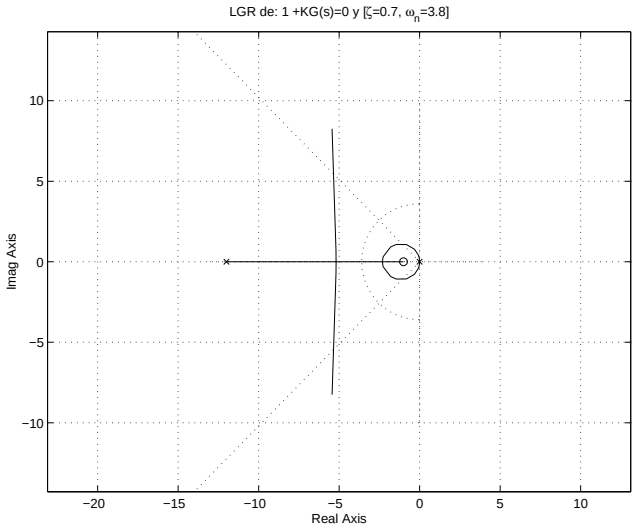


Figura 8: Uso de *sgrid*

Cómo Escoger K con el LGR

El gráfico (figura 7) muestra el lugar geométrico de los polos de lazo cerrado para un controlador proporcional de ganancia K. Obviamente no todos los polos de lazo cerrado satisfacen nuestro criterio. Para determinar qué parte del sitio es aceptable, podemos usar la función $sgrid(\zeta, \omega_n)$ para trazar las líneas de amortiguamiento ζ constante y frecuencia natural ω_n constante. Los argumentos de esta función pueden pueden ser vectores, si usted quiere observar un rango de valores aceptables. En nuestro caso, necesitamos un overshoot menor que $5\,\%$ (significa que $zeta \geq 0{,}7$) y un tiempo de subida de $0{,}5$ segundos (significa $\omega_n > 3{,}6$). En la misma ventana principal de Matlab, ejecute:

- $zeta = 0{,}7$;
- $Wn = 3{,}6$;
- $sgrid(zeta, Wn)$

Ud. obtendrá un gráfico como el de la figura 8.
Note que t_r y ω_n estan relacionados por la ecuación que vimos en el curso ME55A:

$$t_r \cong \frac{1{,}8}{\omega_n}$$

Además el máximo overshoot M_p está relacionado con el parámetro de amortiguamiento ζ del sistema a través de la ecuación:

$$M_p = e^{-\pi\zeta/\sqrt{1-\zeta^2}}$$

En este caso $M_p = 0{,}05$ y $\omega_n = 1{,}8/t_r$.
En el gráfico de la figura 8 las dos líneas punteadas blancas (más o menos a 45 ^{o}) indican la ubicación de polos con $\zeta = 0{,}7$; Entre estas líneas, los polos tendrán $\zeta > 0{,}7$ y fuera de las líneas $\zeta < 0{,}7$. El semicírculo

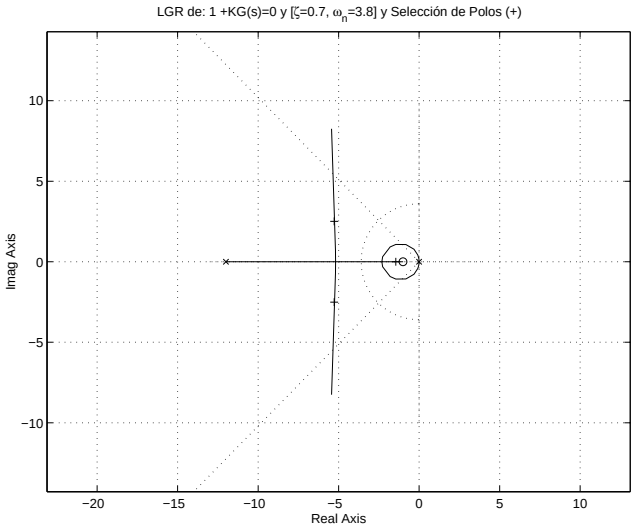


Figura 9: Uso de *rlocfind*

indica posiciones del polo con una frecuencia natural $\omega_n = 3{,}6$; Dentro del círculo, $\omega_n < 3{,}6$ y fuera del círculo $\omega_n > 3{,}6$.

Ahora, volviendo a nuestro problema, para obtener un overshoot menor que $5\,\%$, los polos tienen que estar entre las dos líneas punteadas blancas, y para que el tiempo de subida sea más corto que $1/2$ segundo, los polos tienen que estar fuera del semicírculo punteado blanco. Vemos que la parte del LGR que nos sirve está fuera del semicírculo pero entre las dos líneas. Ojo que todos los polos de este caso están en el lado izquierdo del plano complejo: sistema estable.

Ahora debemos elegir una ganancia K del controlador proporcional, que nos mueva los polos al lugar deseado del LGR. Esto lo puede hacer en Matlab usando la función $rlocfind$, escogiendo el lugar donde Ud. quiere los polos o si quiere aprender más de control, elija los polos en el LGR que cumplan además del criterio impuesto, la condición angular que vimos en el curso. Ejecute el siguiente comando en la ventana principal de Matlab.

$$[K, polos] = rlocfind(num, den)$$

Desplace el Mouse sobre la figura anterior y haga click sobre el lugar donde desea que se encuentre el polo de lazo cerrado pero que satisfaga el criterio especificado anteriormente. Verá modificarse la figura anterior (figura 9)

Note que el LGR puede tener más de una rama, y cuando usted selecciona un polo (y por ende K), usted debe saber donde se han desplazado los demás polos. Recuerde que ellos afectan la respuesta del sistema. En la figura 9 vemos que los polos marcados con $(+)$ son razonables. La función $rlocfind$ nos entrega como resultado los nuevos polos y la ganancia K del proporcional asociada a ellos.

Respuesta de lazo abierto

Para encontrar la respuesta en lazo abierto, debemos conocer la función de transferencia en lazo cerrado. Usted la puede calcular a partir del diagrama de bloque, o pedirle a Matlab que lo haga por Ud.

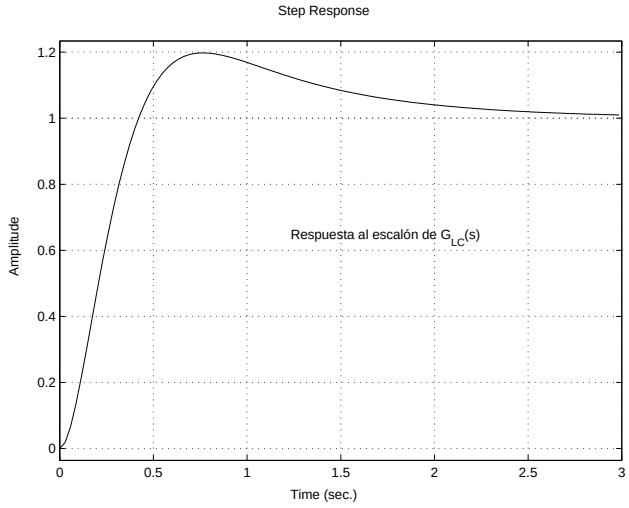


Figura 10: Respuesta al escalón, uso de *step*

$$[numLC, denLC] = cloop(K * num, den)$$

Los dos argumentos de $cloop$ son el numerador y denominador del sistema de lazo abierto. No olvide incluir la ganancia proporcional que usted escogió. Se asume retro–alimentación unitaria. Si usted tiene una situación de retro–aliemntación no–unitaria, use el help de Matlab para la función $feedback$, que entrega la función de transferencia en lazo cerrado con ganancia en retro distinta de 1.
Compruebe la respuesta de su sistema de lazo cerrado con:

$$step(numLC, denLC)$$

En la figura 10, aparece la respuesta esperada: overshoot menor que $5\,\%$ y tiempo de subida $t_r = 0{,}5$ segundos. Si Ud. es consciente e incrédulo (como cualquier buen estudiante de esta escuela), puede identificar la función de transferencia de lazo cerrado a partir del diagrama de bloque usando la regla de Mason,

$$G_{LC}(s) = \frac{Y(s)}{R(s)} = \frac{K(s+1)}{(s^3 + 12s^2 + Ks + K)}$$

Así identifique los coeficientes de las potencias de s, en orden decreciente, del numerador $b(s)$: K, K y del denominador $a(s)$: $1, 12, K, K$. Cree los vectores $num = [K\ K]$ y $den = [1\ 12\ K\ K]$. Ahora ejecute el mismo comando anterior $step(num, den)$ para verificar que Matlab utilizó correctamente la función $cloop$. El gráfico obtenido debe ser idéntico al anterior.