Análisis de las respuestas de un SLI de tiempo continúo con MatLab.

María del Rosario Vázquez Fuentes

Enero 2017

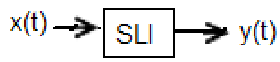
Índice General

1.1. Objetivo de aprendizaje

Que el alumno adquiera los conocimientos para el análisis de los sistemas lineales e invariantes en el tiempo (SLI) de tercer orden utilizando una herramienta de cómputo avanzado.

1.2. Introducción teórica

Un sistema físico con una entrada y una salida está representado por el diagrama de bloque:



Donde x (t) es la señal de entrada, y (t) es la señal de salida.

El modelo matemático normalizado del SLI de orden "n" está dado por:

$$\frac{d^n}{dt^n}y(t) + a_{n-1}\frac{d^{n-1}}{dt^{n-1}}y(t) + \cdots + a_1\frac{d}{dt}y(t) + a_0 y(t) = b_m\frac{d^m}{dt^m}x(t) +,,, + b_1\frac{d}{dt}x(t) + b_0 x(t)$$

a forma equivalente es

$$\sum_{1=0}^{n}\left(a_1\,\frac{d^i}{dt^i}\,y(t)\right) = \sum_{1=0}^{m}\left(b_i\frac{d^i}{dt^i}x(t)\right)$$

Respuesta de un sistema SLI

1.2.1. Respuesta libre (zero input)

La genera el sistema cuando la entrada x(t) igual a cero y las condiciones iniciales son diferentes de cero $\mathcal{Y}_{\mathrm{zi}}(\mathrm{t})$.

$$\sum_{1=0}^{n}\left(a_1\,\frac{d^i}{dt^i}\,y(t)\right) = 0$$

1.2.2. Respuesta forzada (zero state)

La genera el sistema cuando la entrada x(t) es diferente de cero y las condiciones iniciales iguales a cero $\mathcal{Y}_{\mathrm{zs}}(\mathrm{t})$

$$\sum_{1=0}^{n}\left(a_1\,\frac{d^i}{dt^i}\,y(t)\right) = \sum_{1=0}^{m}\left(b_i\frac{d^i}{dt^i}x(t)\right)$$

1.2.3. Respuesta total

La genera el sistema cuando las condiciones iniciales y entrada x(t) son diferentes de cero o bien, sumando la respuesta de entrada cero más la respuesta de estado cero.

$$\mathcal{Y}_{tot}(t) = \mathcal{Y}_{\mathrm{zi}}(\mathrm{t}) + \mathcal{Y}_{\mathrm{zs}}(\mathrm{t})$$

1.2.4. Respuesta permanente

También llamada de estado estable, la produce el sistema cuando la respuesta se ha estabilizado después de que ha transcurrido un cierto tiempo. Algebraicamente, es el límite cuando el tiempo tiende a infinito de la respuesta total.

$$\mathcal{Y}_{perm}(\mathrm{t}) = \lim_{t\to\infty}\mathcal{Y}_{tot}\,(t)$$

1.2.5. Respuesta transitoria

La produce el sistema antes de alcanzar el estado permanente.

$$\mathcal{Y}_{trans}(t) = \mathcal{Y}_{tot}(t) - \mathcal{Y}_{perm}(t)$$

1.2.6. Función de transferencia y patrón de polos y ceros

La función de transferencia es un modelo matemático de un SLI definido como la relación de la transformada de Laplace de la salida $Y(s)$ y la transformada de Laplace de la entrada $X(s)$. Esta función de transferencia se obtiene de la ecuación diferencial invariante en el tiempo dado por el sistema.

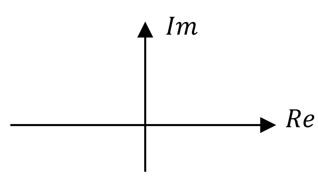
$$H(s) = \frac{Y(s)}{X(s)} \qquad \text{Con condiciones iniciales iguales a cero}$$

Las raíces del polinomio $Y(s)$ son los ceros y las raíces del polinomio $X(s)$ son los polos del sistema.

$$H(s) = \frac{y(s)}{x(s)} = \frac{(s-z_1)\,(s-z_2)\ldots(s-z_m)}{(s-p_1)(s-p_2)\ldots(s-p_n)} \qquad \text{Con } m \leq n \text{ para un sistema real.}$$

Un cero es el valor de **s** de $Y(s)$ que hace que la función de transferencia sea cero.
Un polo es el valor de **s** de $X(s)$ que hace que la función de transferencia tienda a infinito.

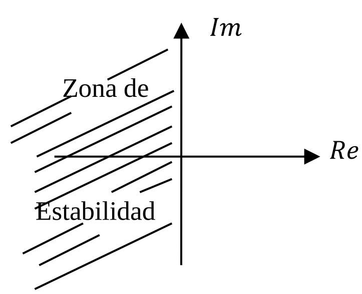
El patrón de polos y ceros de un SLI es una gráfica en el plano complejo que muestra la ubicación de los polos indicados por una cruz "×" y los ceros indicados por un círculo "∘". La ubicación de los polos es de fundamental importancia para determinar la estabilidad del sistema.



1.27. Estabilidad de SLI

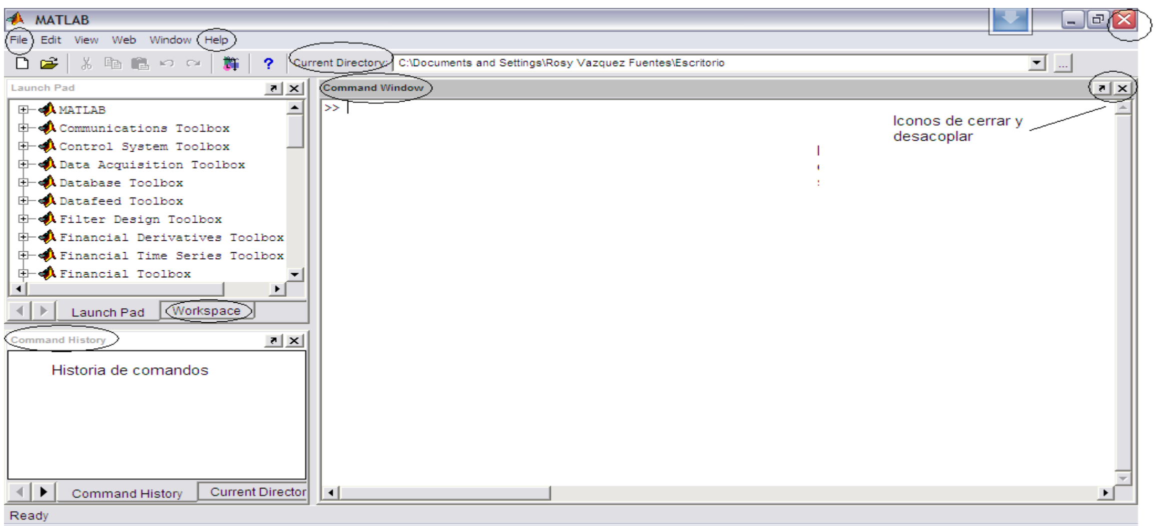
Un sistema es estable con función de transferencia $H(s)$ si todos los polos se encuentran en el semiplano izquierdo abierto es decir, si todos los polos cumplen con

$$Re\{p_i\} < 0$$



1.2.8. Ambiente de Matlab

En la mayoría de los casos, cuando el sistema es de orden superior, el resultado del análisis puede ser comprobado aplicando Matlab para obtener la repuesta libre, forzada, total, transitoria y permanente. Y el patrón de polos y ceros pudiéndose interpretar a través de ellas el comportamiento del sistema.
Matlab es un paquete de programación numérica con una gran capacidad para resolver problemas de aplicaciones en diversas áreas de la ingeniería, química, en economía y en muchas aplicaciones de diseño, basado en un software de matrices para el análisis, de ahí su nombre (**Mat**rix **Lab**oratory).



Command Windows: (ventana de comandos) donde se ejecutan todas la instrucciones y programas, en Matlab, para ejecutarla teclear enter.

Command History: (Historia de los comandos), Muestra los comandos ejecutados en el orden tecleado. Se puede recuperar cualquier comando al seleccionarlo para poder ejecutarlo en la ventana de comandos.

Current Directory: (Directorio actual) Muestra ficheros de Matlab y ejecuta operaciones de ficheros tales como abrir y buscar contenidos.

Launch Pad: Ejecuta herramientas y documentación de acceso para todos los directorios de Math Works instalados en el computador.

Comandos más utilizados para esta práctica.

dsolve('ec','ci',…'v')	Resuelve la ecuación diferencial sujeta a las condiciones iniciales
plot(t,x(t)'color')	Grafica el conjunto de puntos de la función x (t) en un intervalo de valores previamente definido.
xlabel('texto')	Sitúa el texto al lado del eje x en gráficas 2-D y 3-D
ylabel('texto')	Sitúa el texto al lado del eje vertical y en graficas 2-D y 3-D
Title('texto')	Añade texto en la parte superior de la gráfica en 2-D y 3-D
Grid on	Coloca rejilla en la grafica
FT=tf(num,den)	Crea la función de transferencia en tiempo continuo con numerador y denominador dados y muestras de tiempo Ts unitaria
Roots(A)	Da las raíces del polinomio cuyos coeficientes construyen el vector A
pzmap	Mapea polo/cero
Quit o exit	Salir de Matlab

Respuestas de un sistema de tercer orden.

Se tiene el sistema dinámico lineal e invariante en el tiempo (SLI) con modelo matemático

$$\frac{d^3}{dt^3}y(t) + 6\frac{d^2}{dt^2}y(t) + 11\frac{d}{dt}y(t) + 6y(t) = 10x(t)$$

Con una entrada escalón dada por $x(t) = 10u_{-1}(t)$ y condiciones iniciales:

$$y(0) = 2 \qquad y'(0) = 0 \qquad y''(0) = 0$$

Se puede programar de diferentes maneras los comandos de Matlab para obtener las expresiones matemáticas y respuestas gráficas de las respuestas del sistema, aquí se presenta una manera muy sencilla para obtener las gráficas

1.3. Material y equipo

Una computadora PC con el paquete de programación de Matlab instalado.

1.4. Actividad de Investigación previa

Desarrolle el análisis teórico del SLI dado por su modelo matemático para obtener:
1. La respuesta libre
2. La respuesta forzada
3. La respuesta total
4. La respuesta permanente
5. La respuesta permanente
6. El patrón de polos y ceros
7. analice la estabilidad del sistema

1.5. Desarrollo

Compruebe el resultado de su desarrollo teórico dado por la actividad 1.4 y escriba los códigos para los siguientes experimentos y compare con sus resultados simulados. Obtenga las gráficas correspondientes de cada respuesta e interprete su comportamiento. Exprese sus conclusiones.

1.5.1. Experimento 1

```
close all
clear all
% Respuesta libre
y_libre=dsolve('D3y+6*D2y+11*Dy+6*y=0','y(0)=2','Dy(0)=0','D2y(0)=0','t');
t=0:0.1:10;
z=eval(vectorize(y_libre));
plot(t,z,'r')
xlabel('Tiempo')
ylabel('Amplitud')
title('Respuesta libre')
grid
```

1.5.2. Experimento 2

```
close all
clear all
%Respuesta forzada
y_forzada=dsolve('D3y+6*D2y+11*Dy+6*y=10','y(0)=0','Dy(0)=0','D2y(0)=0','t')
t=0:0.01:10;
z=eval(vectorize(y_forzada));
plot(t,z,'r')
xlabel('Tiempo')
ylabel('Amplitud')
```

```
title('Respuesta forzada')
grid
```

1.5.3. Experimento 3

```
clear all
close all
 %Respuesta total=Respuesta_ libre+Respuesta _forzada o bien
% Con entrada y condiciones iniciales diferentes de cero.
 y_total=dsolve('D3y+6*D2y+11*Dy+6*y=10','y(0)=2','Dy(0)=0','D2y(0)=0','t')
 t=0:0.1:10;
 z=eval(vectorize(y_total));
 plot(t,z,'b')
 xlabel('Tiempo')
 ylabel('Aplitud')
 title('Respuesta total')
 grid
```

1.5.4. Experimento 4

```
close all
clear all
%Respuesta permanente
syms t
y_total=dsolve('D3y+6*D2y+11*Dy+6*y=10','y(0)=2','Dy(0)=0','D2y(0)=0','t');
y_perm=limit(y_total,t,inf);
t=6:0.001:20;
z=eval(vectorize(y_perm))
plot(t,z,'r')
xlabel('Tiempo')
ylabel('Aplitud')
title('Respuesta permanente')
grid
```

1.5.5. Experimento 5

```
close all
clear all
%Respuesta transitoria
syms t
y_total=dsolve('D3y+6*D2y+11*Dy+6*y=10','y(0)=2','Dy(0)=0','D2y(0)=0','t');
y_perm=limit(y_total,t,inf)
y_transitoria=y_total-y_perm;
t=0:0.01:6;
z=eval(vectorize(y_transitoria));
plot(t,z,'r')
 xlabel('Tiempo')
 ylabel('Aplitud')
 title('Respuesta transitoria')
 grid
```

1.5.6. Experimento 6

```
%Obtención de la función de transferencia
 num=[10];
 den=[1,6,11,6];
 sys=tf(num,den)
```

1.5.7. Experimento 7

```
ppyc=tf([10],[1 6 11 6]);
pzmap(ppyc)
```

A partir del resultado anterior justifique si el sistema es o no estable.

1.5.8. Experimento 8

Para el sistema analizado de tercer orden, realice un programa para obtener todas las respuestas a las entadas indicadas en los incisos a, b, y c, la función de transferencia, el patrón de polos y ceros e indique si son o no estables. Reporte los programas y las gráficas correspondientes.

a.- $x(t) = 5e^{-3t}u_{-1}(t)$

b.- $x(t) = 2tu_{-1}(t)$

c.- $x(t) = \cos(3t)u_{-1}(t)$

1.5.9. Experimento 9

Escriba en un archivo M-file el siguiente código de Matlab contenido en el anexo y reporte las conclusiones.

1.6. Bibliografía

César Pérez MATLAB y sus aplicaciones en las ciencias y la ingeniería
Prentice Hall.

Holly Moore. MATLAB para ingenieros. Editorial Pearson- Prentice hall

Kamen Edward W., Heck Bonnie S. Fundamentos de señales y sistemas usando la Web y MATLAB tercera edición. Editorial Pearson-Prentice Hall

Thanos Antoulas, Richard Baraniuk, Steven J. Cox, Benjamin Fite, Roy Ha, Michael Haag, Don Johnson, Ricardo Radaelli_Sanchez, Justin Romberg, Phil Schiniter, Melissa Selik, JP Slavinsk, Ricardo von Borries Señales y Sistemas, connexions Rice University, Houston, Texas,Online: <http://cnx.org/content/col10373/1.2>