Práctica 6
Modelos Matemáticos en Ecuaciones Diferenciales

Objetivo. El objetivo de esta práctica es dar una introducción al uso del comando `ode` y sus variantes para definir y resolver ecuaciones diferenciales mediante Scilab.

Introducción.

El modelo matemático más natural para representar el comportamiento de un sistema dinámico lo constituyen las ecuaciones diferenciales. Las ecuaciones diferenciales permiten representar sistemas dinámicos analógicos de muy diversos tipos: lineales y no lineales, variantes e invariantes en el tiempo, con parámetros concentrados y con parámetros distribuidos.

Debido a la dificultad de establecer una teoría general para tratar tanta variedad de sistemas, la teoría clásica de control se restringe solamente a los SLIT (sistemas lineales e invariantes en el tiempo) cuyo modelado se cubre perfectamente con la noción de función de transferencia que se trató en la práctica No. 4.

En general, un sistema SISO puede ser modelado mediante una ecuación diferencial, no necesariamente lineal.

Por ejemplo, consideremos el caso de una varilla rígida de masa m actuada en su extremo superior por un motor que puede imprimirle un par para hacerla girar respecto a ese mismo extremo (ver figura (6.1).

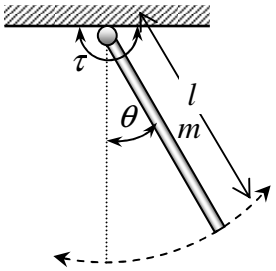

Figura 6.1.- Varilla rígida actuada por uno de sus extremos.

Aplicando la segunda ley de Newton para movimiento rotacional respecto al eje de giro:

$$\sum \tau_{ext} = J\alpha \tag{6.1}$$

Donde: $\sum \tau_{ext}$ es el par resultante externo aplicados al cuerpo en movimiento

J es el momento de inercia respecto al eje de giro del cuerpo en movimiento

$\alpha = \dfrac{d^2\theta}{dt^2}$ es la aceleración angular del cuerpo respecto al eje de giro

En el caso de la varilla los pares externos aplicados son: el par del motor τ, el par debido a la

fricción τ_f que siempre se opone al movimiento de la varilla y el par debido a la fuerza de gravedad τ_g. Ver el diagrama del cuerpo libre en la figura 6.2. en el cual se supone que la masa de la varilla está concentrada en su centro.

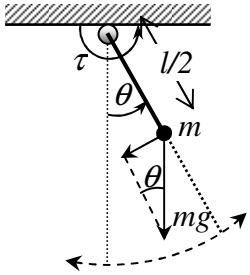

Figura 6.2.- Diagrama del cuerpo libre del centro de masa de la varilla.

El momento de inercia respecto al extremo de giro es $J = \frac{1}{3}ml^2$, por lo tanto, la ecuación (6.1) queda como sigue

$$\tau - \tau_f - \tau_g = \frac{1}{3}ml^2\frac{d^2\theta}{dt^2} \tag{6.2}$$

Del diagrama del cuerpo libre (figura 6.2) se observa que el par debido a la fuerza de gravedad es producido por la componente del peso perpendicular a la varilla, por lo tanto, $\tau_g = mg\frac{l}{2}\sin\theta$. Además, si consideramos sólo el efecto de la fricción viscosa, el par debido a la fricción será proporcional a la velocidad tangencial, es decir, $\tau_f = \mu l\frac{d\theta}{dt}$, donde μ es el coeficiente de fricción viscosa. Sustituyendo en (6.2) se obtiene

$$\tau - \mu\frac{d\theta}{dt} - mg\frac{l}{2}\sin\theta = \frac{1}{3}ml^2\frac{d^2\theta}{dt^2} \tag{6.3}$$

o bien, en notación compacta

$$\frac{1}{3}ml^2\ddot{\theta} + \mu\dot{\theta} + mg\frac{l}{2}\sin\theta = \tau \tag{6.4}$$

Despejando la máxima derivada de θ se obtiene

$$\boxed{\ddot{\theta} = -a_2\dot{\theta} - a_1\sin\theta + b_0\tau} \tag{6.5}$$

donde: $a_2 = \frac{3\mu}{ml^2}$, $a_1 = \frac{3g}{2l}$ y $b_0 = \frac{3}{ml^2}$

La ecuación (6.5) es un modelo matemático para la varilla actuada por el motor, cuya entrada es el par del motor τ y cuya salida es el ángulo θ de la varilla respecto a la vertical, como se muestra en la figura 6.3.

Figura 6.3.- Modelo entrada-salida de la varilla actuada por el motor.

El modelo en Espacio de Estado.

Obsérvese que la ecuación diferencial obtenida (6.5) es una ecuación no lineal, por lo tanto no puede ser expresada mediante una función de transferencia. En este caso, la alternativa de modelo más natural es la propia ecuación diferencial (6.5), o alguna variante de ésta.

La ecuación (6.5) es una ecuación de diferencial segundo orden. Esta ecuación puede ser reescrita como dos ecuaciones diferenciales de primer orden, definiendo dos ***variables de estado*** x_1, x_2 como sigue:

$$x_1 = \theta,\ x_2 = \dot{\theta} \tag{6.6}$$

sustituyendo estas variables en la ecuación (6.5), obtenemos

$$\boxed{\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -a_2 x_2 - a_1 \sin x_1 + b_0 \tau \end{aligned}} \tag{6.7}$$

al sistema de ecuaciones diferenciales de primer orden dado por (6.7) se le llama modelo en variables de estado o en espacio de estado o simplemente ***ecuaciones de estado***.

El espacio de estado es la representación n-dimensional (n=2 en este caso) cuyos ejes coordenados son las variables de estado. En el ejemplo de la varilla actuada, como solo hay dos variables de estado, el espacio de estado es un plano de una variable de estado respecto a la otra (x_2 respecto a x_1 o viceversa).

El comando `ode` de Scilab.

Scilab permite obtener la respuesta de modelos en variables de estado mediante el comando `ode` (Ordinary Differential Equation). `ode` es un comando que permite ejecutar distintos métodos numéricos (***solvers***) para la solución de las ecuaciones de estado. Los diferentes *solvers* disponibles se seleccionan mediante el primer parámetro **`type`** (opcional) con el que se invoca `ode`. En la siguiente tabla se describen los *solvers* disponibles.

Parámetro `type`	Método numérico	Tipo de problema	Exactitud	Cuando usar
`vacío`	LSODA del paquete ODEPACK	Rígido/norígido	media	Es el default y selecciona automá-ticamente entre el método predictor de Adams y la diferenciación hacia atrás.
`"adams"`	Predictor de Adams	No rígido	media	
`"rk"`	Runge-Kutta 4º orden adaptivo	No rígido	media	Problemas moderadamente rígidos. Tolerancia al error holgada
`"rkf" o "fix"`	Runge-Kutta 4º y 5º orden	No rígido/medianamente rígido	media	Problemas moderadamente rígidos. Tolerancia al error holgada
`"root"`	Runge-Kutta 4º y 5º orden	Sistemas de ecuaciones Algebraico diferenciales (DAE)	- - -	Solo aplica para sistemas de ecuaciones diferenciales con condiciones algebraicas
`"discrete"`	Recursión simple	Ecuaciones de diferencias	alta	Solo aplica para sistemas discretos.

Rigidez. Se dice que el problema numérico de resolver un sistema de ecuaciones diferenciales es rígido si es numéricamente inestable (produce errores grandes ante pequeños cambios), excepto para pasos de integración extremadamente pequeños.

Sintaxis de `ode`.- La forma para invocar el comando **`ode`** desde la consola de Scilab es la siguiente:

```
x = ode(x0, t0, t, odefun)
[x,w,iw] = ode([type,] x0, t0, t [,rtol [,atol]], odefun [,jac] [,w, iw])
```

donde:

`x0`	Vector o matriz de Condiciones iniciales.
`t0`	Tiempo inicial (escalar)
`t`	Es un escalar especificando el tiempo final de la simulación (paso de integración adaptable), ó un vector renglón especificando los instantes de tiempo en que la solución es calculada. Es decir, si se desea calcular la solución en instantes específicos, se deberán especificar dichos instantes: `t = [t0,t1,...,tf]`
`odefun`	Función vectorial o lista de objetos que define el lado derecho de las ecuaciones de estado $\dot{x} = f(t,x)$.
`type`	Parámetro opcional para seleccionar el solver: "adams", "stiff", "rk", "rkf", "fix", "discrete", "roots"
`rtol`	Tolerancia relativa*: Escalar o vector del mismo tamaño que x
`atol`	Tolerancia absoluta*: Escalar o vector del mismo tamaño que x
`jac`	Jacobiano de la función f
`w,iw`	Vectores (Entrada/Salida)
`x`	`x` es un vector renglón o una matriz cuyos renglones corresponden a cada variable de estado y cuyas columnas corresponden al instante de tiempo de cada columna de `t`.

Descripción

En su forma más simple: `x = ode(x0, t0, t, f)` integra el sistema de ecuaciones diferenciales $\dot{x} = f(t,x)$ definido dentro de la función `odefun` con las condiciones iniciales dadas `x0`, calculando la solución en el intervalo dado por `t`.

La función `odefun` deberá definirse como $f(t,x)$ con la instrucción `function` para un escalar `t` y un vector columna `x`, esta función debe producir el vector columna $\dot{x}$. Además, si la función requiere parámetros extra, estos deberán pasarse al comando `ode` como una lista de objetos junto con la función `f`.

***Tolerancias**:

Cuando se usa paso de integración adaptable, las tolerancias `rtol` and `atol` son umbrales que controlan los errores relativos y absolutos estimados. El error estimado para el estado x_i es:

$\hat{e}_i = rtol_i |x_i| + atol_i$ y la integración es realizada mientras este error se mantiene pequeño para todos los componentes del vector de estado x. Si rtol y/o atol son escalares, cada componente `rtol`(i) y/o `atol`(i) es puesto al valor escalar especificado. Los valores por default para `rtol` y `atol` son respectivamente `rtol`=1x10^{-7} and `atol`=1x10^{-9} para la mayoría de los *solvers* y `rtol`=1.x10^{-3} y `atol`=1x10^{-4} para "`rfk`" y "`fix`".

El Vector %ODEOPTIONS.

Este vector permite controlar la exactitud y otras variantes de la manera en que se ejecuta el solver. El contenido del vector **%ODEOPTIONS** es el siguiente:

[itask, tcrit, h0, hmax, hmin, jactyp, mxstep, maxordn, maxords, ixpr, ml, mu].

y su valor por default es: `[1,0,0,%inf,0,2,500,12,5,0,-1,-1]`

El único valor de este vector que usaremos en este curso es el primero **itask** de acuerdo a lo siguiente:

itask= 1: Selecciona cálculo normal (Los instantes de integración son especificados por el vector de instantes de tiempo que se deberá proporcionar a la función `odefun`)

itask= 2: Selecciona el cálculo con paso de integración variable (Los instantes de integración son calculados de manera adaptable para respetar las tolerancias especificadas por rtol y atol. En este caso en lugar de un vector de instantes de tiempo, solo se requiere proporcionar el instante final de tiempo de integración a la función `odefun`). El vector de instantes de tiempo generado es devuelto en el primer renglón de x.

Solución de un sistema de ecuaciones de estado <u>sin entrada</u> mediante ode45.

Para ilustrar como se utiliza **ode** para obtener la solución de un conjunto de ecuaciones de estado a continuación definimos las ecuaciones de estado (6.7) para el ejemplo del sistema de la varilla <u>no</u> actuada considerando los siguientes valores para las constantes involucradas

$$m = 0.1 Kg\ ,\ \ \mu = 0.01\ Nt\ seg\ ,\ l = 0.5 mt\ ,\ g = 9.81\ mt/seg^2$$

y se considera el movimiento natural de la varilla (con el actuador desconectado), es decir, la entrada de par es cero, $\tau = 0\ Nt$. Para ello debemos escribir la función `odefun` siguiente:

```
function dx=varilla(t,x)
// Ecuaciones de estado de una varilla rígida no actuada
//Constantes del sistema:
m=0.1;
mu=0.01;
L=0.5;
g=9.81;
a2=3*mu/m/L^2;
a1=3*g/2/L;
b0=3/m/L^2;
// Entrada al sistema (cero)
```

```
tao=0;
////////////////////////////////////////////////////
// Ecuaciones de Estado:
dx(1) = x(2);
dx(2) = -a2*x(2)-a1*sin(x(1))+ b0*tao;
endfunction
```

Para obtener la solución de las ecuaciones de estado, guardamos la función anterior en el archivo `varilla.sci` y a continuación usamos el comando **ode** como sigue:

```
exec varilla.sci; //Carga función varilla.sci
x0=[*%pi/2; 0]; //vector columna de condiciones iniciales
%ODEOPTIONS=[2,0,0,%inf,0,2,500,12,5,0,-1,-1];//Selecciona paso adaptable
x=ode(x0,0,10,varilla); //Calcula la solución de 0 a 10 seg
//// en el intervalo de 0 a 10 con condiciones iniciales (pi/2,0)
t=x(1,:)'; //Obtiene vector de tiempo y lo convierte a columna
x1=x(2,:)'; x2=x(3,:)'; //Obtiene los estados en forma de columna
f0=scf(0); plot(t,x1,t,x2); //Grafica los dos estados respecto a t
xgrid(1);
legend('x1=ángulo','x2=velocidad angular');
f1=scf(1); plot(x1,x2) //grafica la solución en el plano de estado
xgrid(1);
clear %ODEOPTIONS; //Restablece a valores default
```

En la figura 6.4 se observa el comportamiento de los dos estados respecto al tiempo. Como puede verse, la dinámica corresponde a lo esperado en un péndulo simple con fricción, el cual comienza oscilando durante un tiempo y al final se estabiliza en la posición vertical de reposo ($\theta = 0$). El mismo comportamiento se visualiza en la trayectoria de la solución en el plano de estado de la figura 6.5.

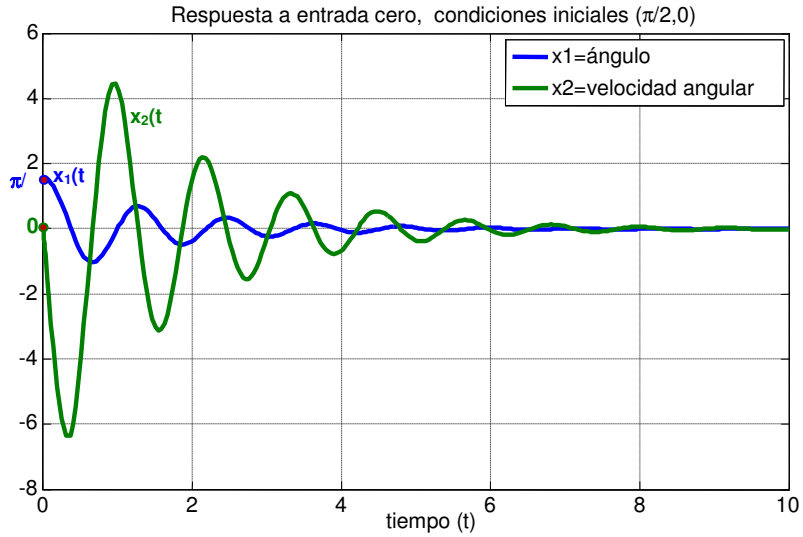

Figura 6.4.- Respuesta de la varilla con el actuador desconectado.

Se puede checar cuantos puntos de la solución fueron generados mediante el comando size:

```
size(x)
ans  =
   3.    331.
```

Es decir, se calcularon 331 puntos de la solución por cada estado del sistema.

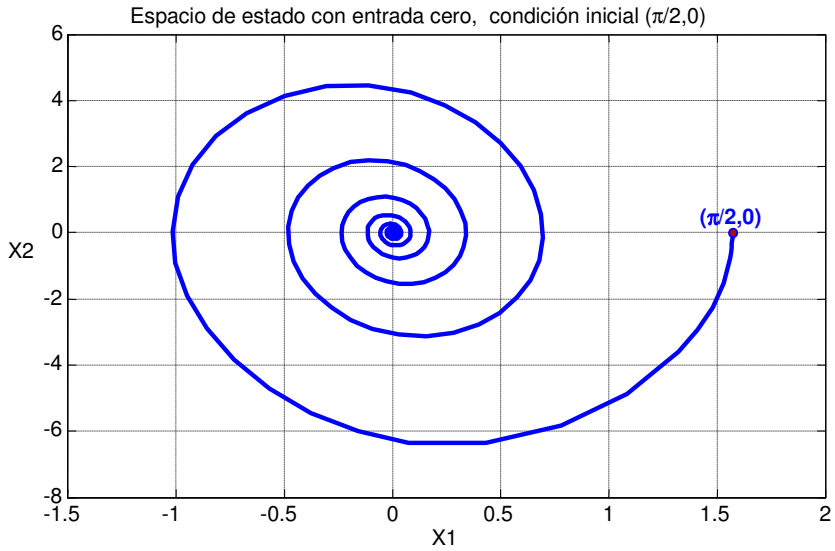

Figura 6.5.- Respuesta de la varilla sin actuador en el plano de estado.

Se puede aumentar la exactitud del método de integración, o disminuirla si lo que se quiere es reducir el tiempo de cálculo cambiando los parámetros de tolerancia como sigue

```
rtol=1e-5; //Aumenta tolerancia relativa (menor exactitud)
atol=1e-6;//Aumenta toleracia absoluta (menor exactitud)
%ODEOPTIONS=[2,0,0,%inf,0,2,500,12,5,0,-1,-1];//paso adaptable
x=ode(x0,0,10,rtol,atol, varilla); //Calcula la solución
size(x)
ans  =
    3.    138.
```

La ejecución resulta más rápida, pero como puede verse ahora solamente se generaron 138 puntos de la solución por cada estado del sistema y las gráficas correspondientes son un poco más burdas.

Solución de un sistema de ecuaciones de estado con entrada mediante `ode`.

Una manera sencilla de considerar funciones del tiempo como entradas a un sistema definido en una función `odefun` es calcularlas dentro del código de la misma función `odefun`, sin embargo, esto no es lo más adecuado, pues se supone que una entrada debe provenir de fuera del sistema.

Por ejemplo, para considerar una entrada sinusoidal de frecuencia 1 hertz, en el ejemplo de la varilla, simplemente reemplazamos la línea donde se define la entrada `tao=0` por:

```
tao=sin(2*pi*t); // Entrada sinusoidal de 1 hertz
```

La única manera de que la función `odefun` en la cual se define el sistema de ecuaciones de estado acepte una entrada variable desde afuera de la función es enviársela como un parámetro adicional.

El nuevo parámetro puede ser una función que calcule los valores de la entrada en cada instante de tiempo de integración.

Ejemplo: La función varilla se reescribirá para aceptar como parámetro una función del tiempo denominada `entrada` como sigue

```
function dx=varilla1(t,x,entrada)
// entrada es el vector de valores de entrada en los instantes
// de tiempo dados en t1
// Constantes del sistema:
m=0.1;
mu=0.01;
L=0.5;
g=9.81;
a2=3*mu/m/L^2;
a1=3*g/2/L;
b0=3/m/L^2;
// Entrada al sistema (vector de valores de entrada en cada instante t1)
tao= entrada(t); //Calcula los valores de la señal de entrada en cada
//instante de //tiempo de integración t
//////////////////////////
// Ecuaciones de Estado:
dx(1) = x(2);
dx(2) = -a2*x(2)-a1*sin(x(1))+ b0*tao;
```

Para obtener la respuesta a una función de entrada primero se definirá la función de entrada deseada, por ejemplo a continuación se considera una entrada pulso de amplitud 1 Nt y de duración 0.5 seg que aparece en el instante t=0.5 seg

```
//primero se define la función que calcula la entrada en cada instante t
function u=entrada(t)
  u=(t>0.5 & t<1)*1; //Entrada pulso de amplitud 1 y duración 0.5
endfunction
/////
exec varilla1.sci;
x0=[0.0001;0];// Condiciones iniciales casi cero.
%ODEOPTIONS=[2,0,0,%inf,0,2,500,12,5,0,-1,-1];//paso adaptable
x=ode(x0,0,10,list(varilla1,entrada)); //Calcula la solución
t=x(1,:)'; //Obtiene vector de tiempo y lo convierte a columna
u=entrada(t);//Obtiene valores de entrada en los puntos de integración
x1=x(2,:)'; x2=x(3,:)'; //Obtiene los estados en forma de columna
f0=scf(0); subplot(2,1,1);
plot(t,x1,t,x2); //Grafica los dos estados respecto a t
xgrid(1);
legend('x1=ángulo','x2=velocidad angular');
subplot(2,1,2)
plot(t,u,'r');xgrid;//Grafica la señal de entrada
legend('par aplicado por el motor');
clear %ODEOPTIONS; //Restablece a valores default
```

En la figura 6.6 se muestran las gráficas de la respuesta de los dos estados del sistema así como la entrada de par aplicada.

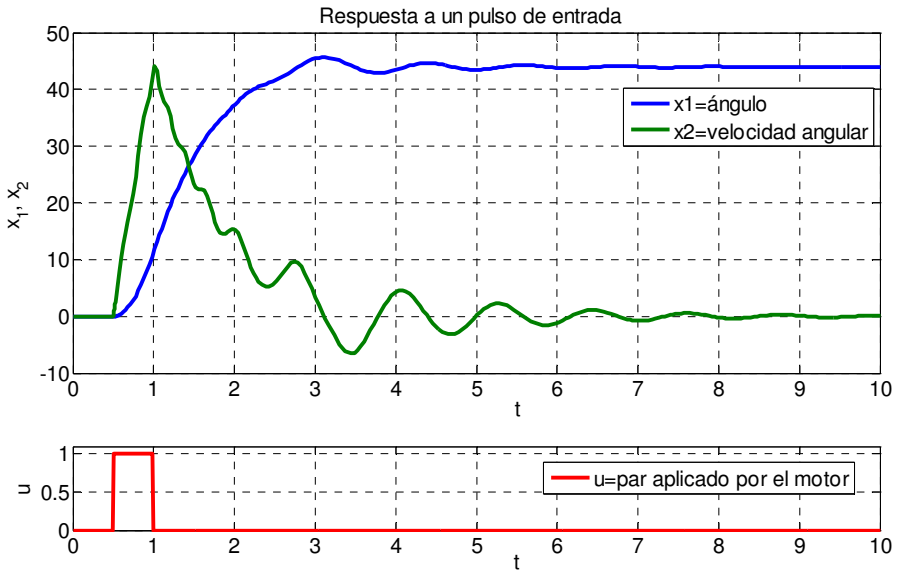

Figura 6.6.- Respuesta de la varilla a un pulso de par aplicado por el motor.

Caso lineal.

Cuando un sistema SISO de entrada $u(t)$ y salida $y(t)$ es un SLIT, se puede representar por un modelo lineal como la ecuación diferencial lineal de orden n siguiente,

$$a_n \overset{(n)}{y} + a_{n-1} \overset{(n-1)}{y} + \ldots + a_1 \dot{y} + a_0 y = b_m \overset{(m)}{u} + b_{m-1} \overset{(m-1)}{u} + \ldots + b_1 \dot{u} + b_0 u \tag{6.8}$$

Aunque no siempre es sencillo, como se explicó en la práctica No. 4, es posible elegir un conjunto de variables de estado para transformar (6.8) en n ecuaciones diferenciales lineales de primer orden de la forma

$$\dot{x} = Ax + bu \tag{6.9}$$

además de una ecuación de salida dada por

$$y = cx + du \tag{6.10}$$

Donde x es el vector de estados, A es una matriz cuadrada $n \times n$ de coeficientes constantes, b es un vector columna de coeficientes constantes, c es un vector renglón de coeficientes constantes y d es un escalar constante (en la mayoría de los casos $d = 0$), es decir,

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix}, \quad A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \quad c = \begin{bmatrix} c_1 & c_2 & \ldots & c_n \end{bmatrix}$$

En este caso las ecuaciones de estado y la ecuación de salida (6.9) se pueden transformar de manera directa a su Función de Transferencia equivalente aplicando Transformada de Laplace a ambos lados de (6.9), suponiendo condiciones iniciales cero, obtenemos

$$\begin{aligned} sX(s) &= AX(s) + bU(s) \\ Y(s) &= cX(s) + dU(s) \end{aligned} \tag{6.11}$$

despejando el vector de estado

$$X(s) = (sI - A)^{-1} bU(s) \tag{6.12}$$

sustituyendo en la ecuación de salida

$$Y(s) = \left[c(sI - A)^{-1} b + d \right] U(s) \tag{6.13}$$

Es decir, la función de transferencia correspondiente a (6.9) es

$$G(s) = \frac{Y(s)}{U(s)} = c(sI - A)^{-1} b + d \tag{6.14}$$

Scilab provee varios comandos que fueron explicados en la práctica No. 4 para definir modelos en el espacio de estado en el ***caso lineal***: syslin, abcd, ss2tf y tf2ss. Por lo tanto, en el caso lineal no es necesario el uso del comando **ode**.

Ejemplo: Modelo Lineal del sistema de la varilla actuada.

Como se mencionó arriba, el ejemplo de la varilla actuada por un extremo nos condujo al modelo en ecuación diferencial no lineal (6.5), el cual se pudo escribir como un modelo en ecuaciones de estado no lineales (6.7). Este modelo se puede *aproximar* por un modelo lineal si observamos que el único término no lineal en las ecuaciones de estado (6.7) es la función $\sin(x_1)$.

Si expresamos la función $\sin(x_1)$ por su fórmula de Taylor en las cercanías de $x_1 = 0$, es decir, para movimientos de la varilla cercanos a su punto de reposo, obtenemos

$$\sin(x_1) = x_1 - \frac{1}{3!} x_1^{\,3} + \frac{1}{5!} x_1^{\,5} - \frac{1}{7!} x_1^{\,7} + \ldots \tag{6.15}$$

Si truncamos los términos no lineales, dejando solo la parte lineal

$$\sin(x_1) \approx x_1 \tag{6.16}$$

Sustituyendo esta aproximación en el modelo de estado (6.7), obtenemos el modelo aproximado en ecuaciones de estado

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -a_2 x_2 - a_1 x_1 + b_0 \tau \end{aligned} \tag{6.17}$$

y con la ecuación de salida

$$y = \theta = x_1 \tag{6.18}$$

las ecuaciones (6.17) y (6.18) se pueden escribir en la forma matricial (6.9), donde

$$A = \begin{bmatrix} 0 & 1 \\ -a_1 & -a_2 \end{bmatrix},\ b = \begin{bmatrix} 0 \\ b_0 \end{bmatrix},\ c = \begin{bmatrix} 1 & 0 \end{bmatrix},\ d = 0$$

O bien, podemos obtener la función de transferencia correspondiente, mediante la expresión (6.14)

$$G(s)=\frac{Y(s)}{U(s)}=c(sI-A)^{-1}b=\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}s & -1\\ a_1 & s+a_2\end{bmatrix}^{-1}\begin{bmatrix}0\\ b_0\end{bmatrix}$$

calculando la matriz inversa obtenemos

$$G(s)=\frac{\begin{bmatrix}1 & 0\end{bmatrix}}{s^2+a_2s+a_1}\begin{bmatrix}s+a_2 & 1\\ -a_1 & s\end{bmatrix}\begin{bmatrix}0\\ b_0\end{bmatrix}$$

Haciendo las operaciones

$$G(s)=\frac{\begin{bmatrix}1 & 0\end{bmatrix}}{s^2+a_2s+a_1}\begin{bmatrix}b_0\\ sb_0\end{bmatrix}=\frac{b_0}{s^2+a_2s+a_1} \qquad (6.19)$$

Para validar el modelo lineal aproximado (6.19) a continuación escribimos el siguiente código para comparar los resultados del modelo lineal contra el modelo exacto no lineal (6.7).

```
m=0.1; mu=0.01; L=0.5; g=9.81; //// Constantes del sistema
a2=3*mu/m/L^2;a1=3*g/2/L;
b0=3/m/L^2;
num=poly([b0],'s','c');//Numerador de la FT
den=poly([a1 a2 1],'s','c');//Denominador de la FT
G=syslin('c',num,den) //// F.T. del sistema, de acuerdo a (6.19)
G  =
          120
    ---------------
                   2
    29.43 + 1.2s + s

//Se define la función que calcula la entrada en cada instante t
function u=entrada(t)
  u=(t>0.5 & t<0.6)*1;; //// Define entrada pulso, amplitud 1, ancho 0.1
endfunction

// Obtiene la respuesta del modelo no lineal a la entrada pulso:
exec varilla1.sci;
t=0:0.01:10; //vector de tiempo de simulación
x0=[0.001;0];// Condiciones iniciales casi cero.
%ODEOPTIONS=[1,0,0,%inf,0,2,500,12,5,0,-1,-1];//paso fijo
x=ode(x0,0,t,list(varilla1,entrada)); //Calcula la solución
subplot(2,1,1);
plot(t',x'); // Grafica los dos estados del modelo no lineal
xgrid(1);

// Obtiene la respuesta del modelo lineal a la entrada pulso:
u=entrada(t);//Calcula valores de la señal de entrada
[y,x]=csim(u,t,tf2ss(G),x0);
plot(t',y','r--'); // Grafica la salida de la F.T.
legend('x1 exacta','x2exacta','y=x1 aprox lineal');
subplot(2,1,2);
plot(t,u,'r');// Grafica el par de entrada al sistema.
xgrid(1);
```

En la figura 6.7 se muestra la respuesta del código anterior, ligeramente arreglada para una mejor presentación.

Figura 6.7.- Respuesta de la varilla a un pulso de ancho 0.1 y amplitud 1, calculada mediante el modelo no lineal exacto (6.7) y el modelo lineal aproximado (6.19).

Ejercicios:

1) Generar las figuras 6.6 y 6.7.
2) Modificar la figura 6.7 para que en la gráfica superior solo muestre la salida exacta y la salida aproximada.
3) Después del punto anterior, modificar la entrada para que sea un pulso de amplitud 1 y duración 0.4 seg. que inicie desde el instante inicial. y volver a genera la figura correspondiente.

Desarrollo de la Práctica.

1. Probar todos los ejemplos propuestos por el profesor conforme los va explicando.
2. Realizar todos los ejercicios propuestos.
3. Contestar el cuestionario de evaluación de la práctica.

Reportar:

1) El código utilizado para resolver el ejercicio (3) con comentarios adecuados.
2) Describe con tus propias palabras el movimiento que se observaría en la varilla si el experimento del ejercicio (3) se realizara en forma real. ¿Cuántas vueltas completas daría la varilla?

3) Para el circuito de la figura 6.8 cuya F.T. $\frac{V_C(s)}{V_i(s)}$ se obtuvo en el reporte de la práctica No. 4. Obtén las ecuaciones de estado en términos de los valores de R,L y C, eligiendo los estados siguientes: $x_1(t) = V_c(t)$, $x_2(t) = \dot{V}_c(t)$.

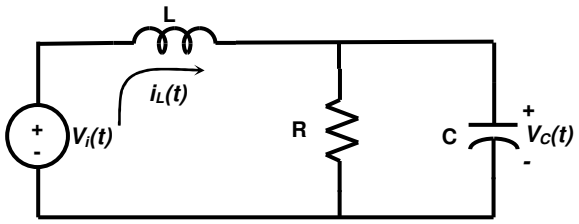

Figura 6.8.- Circuito para el problema 3 del reporte.