

## 1 Módulo 1

- Realización de trabajo colaborativo en simulación de procesos dinámicos mediante Colaboratory y GitHub.
- Funciones de control básicas para cómputo numérico en Matlab/Octave.
- Funciones para cálculo simbólico Symbolic Math Toolbox en Octave.
- Funciones avanzadas en Matlab/Octave de cálculo simbólico, alternativo a Symbolic Math Toolbox.
- El uso de Octave es optativo, si se prefiere otro de los programas como Python, R o Scilab, puede emplearse en su lugar.

### Ejercicios del Módulo 1

**Se debe redactar un informe que debe realizarse de manera individual por cada estudiante. Dicho informe debe contener:**

- 1- todos los resultados correctos de las consignas dadas que pueden generarse en grupo.**
- 2- un resumen de las lecciones aprendidas**
- 3- detalles de problemas que aparecieron, las fuentes de datos, enlaces etc., generando así Recomendaciones finales o Conclusiones parciales de la actividad.**

**Una vez finalizado, titular el archivo del informe del modo Apellido\_Nombre\_M1.pdf y subir un único archivo en la solapa correspondiente con los ejercicios resueltos.**

#### Contenidos

1	Módulo 1 .....	1
	Ejercicio 1. Asignación de variables y verificación con simulación numérica .....	1
	Ejercicio 2. Cálculo simbólico en de linealización de sistemas dinámicos.....	4
	Ejercicio 3. Linealización de un sistema no lineal y verificación en simulación numérica .....	4

#### Ejercicio 1. Asignación de variables y verificación con simulación numérica

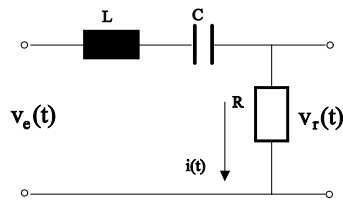


Fig. 1-1. Esquemático del circuito RLC.

Sea el sistema eléctrico de la Fig. 1-1, se requiere obtener la evolución temporal de la corriente para diferentes tensiones de entrada.

Análisis: para determinar la evolución de la corriente del circuito se pueden plantear las ecuaciones diferenciales del mismo como,

$$\begin{cases} \frac{di_t}{dt} = -\frac{R}{L} i_t - \frac{1}{L} v_{c(t)} + \frac{1}{L} v_{e(t)} \\ \frac{dv_{c(t)}}{dt} = \frac{1}{C} i_t \end{cases} \quad (1-1)$$

Las Ec. (1-1) se pueden expresar en una ecuación matricial-vectorial con las representaciones en variables de estado definiendo a  $i$ ,  $v_c$  como variables de estado y a  $\mathbf{x}$  como vector de estado, es decir  $\mathbf{x} = [i \ v_c]^T$  se convierte en

$$\dot{\mathbf{x}} = \mathbf{A} \mathbf{x}(t) + \mathbf{b} u(t) \quad (1-2)$$

$$y = \mathbf{c}^T \mathbf{x}(t) \quad (1-3)$$

donde las matrices contienen a los coeficientes del circuito,

$$\mathbf{A} = \begin{bmatrix} -R/L & -1/L \\ 1/C & 0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 1/L \\ 0 \end{bmatrix}, \quad (1-4)$$

$$\mathbf{c}^T = [R \ 0]. \quad (1-5)$$

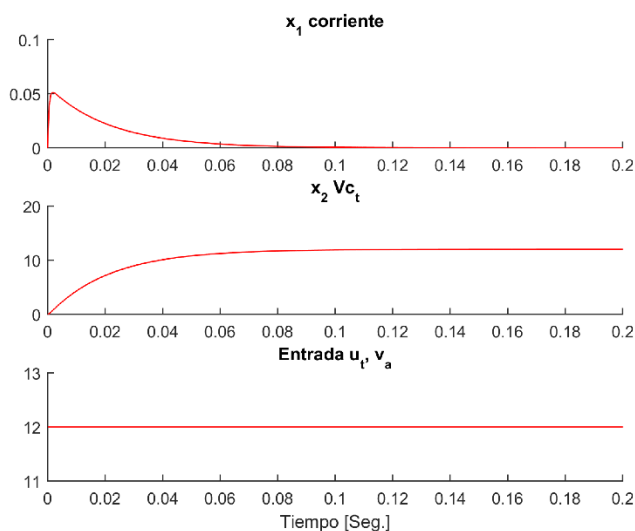


Fig. 1-2. Curvas del circuito RLC para una entrada de 12V.

- 1- Asignar valores a  $R=2,2K\Omega$ ,  $L=10\mu H$ , y  $C=100nF$ . Obtener simulaciones que permitan estudiar la dinámica del sistema, con una entrada de tensión escalón de 12V.
- 2- Asumir que la tensión de entrada cambia de signo cada 1ms.
- 3- Asignar valores a  $R=10K\Omega$ ,  $L=2\mu H$ , y  $C=10nF$ ; repetir lo anterior para comparar el resultado y verificar la correcta simulación.

<https://www.wolframalpha.com/input?i=use+Euler+method+y%27+%3D+-2++y%2C+y%281%29+%3D+2%2C+from+1+to+5>

## Ejercicio 2. Actividad integradora

Sabiendo que un sistema lineal en variables de estado tiene la representación

$$\begin{aligned}\dot{\mathbf{x}}_t &= \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t \\ \mathbf{y}_t &= \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t\end{aligned}\quad (1-6)$$

donde  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{u} \in \mathbb{R}^e$ ,  $\mathbf{y} \in \mathbb{R}^s$ , con  $\mathbf{x}_0 = \mathbf{x}(0)$ , donde A, B, C y D son las matrices de estados, entrada, salida y acople directo respectivamente, se propone escribir en ésta forma al siguiente sistema de ecuaciones lineales (1-7).

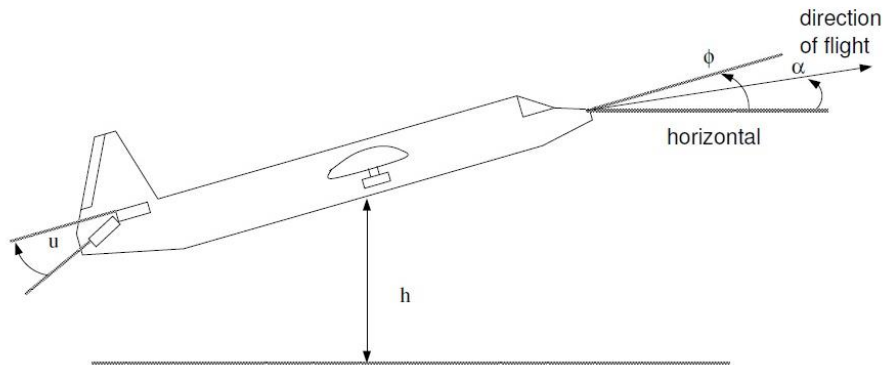


Fig. 1-3. Modelo de sistema de altitud en un avión, extraído de [1].

Para el caso de la Fig. 1-3, modelo válido sólo para pequeños ángulos, se tiene

$$\begin{cases} \dot{\alpha} = a(\phi - \alpha) \\ \ddot{\phi} = -\omega^2(\phi - \alpha - b \cdot u) \\ \dot{h} = c\alpha \end{cases}\quad (1-7)$$

donde  $\omega > 0$  representa la frecuencia natural, y los coeficientes  $a$   $b$  son constantes positivas,  $u$  es la variable manipulada y es proporcional a la posición de los elevadores,  $\phi$  (ángulo de cabeceo) en radianes, vuela a  $c$  metros por segundo, su trayectoria de

vuelo forma un ángulo  $\alpha$  con la horizontal (si  $\alpha > 0$  sube, si  $\alpha < 0$  desciende) Elegir  $x_1 = \alpha$ ,  $x_2 = \phi$ ,  $x_3 = \dot{\phi}$  y  $x_4 = h$ . Se pide,

- 1- Obtener el sistema lineal en variables de estado para el equilibrio  $\mathbf{x} = [0 \ 0 \ 0 \ 0]^T$ .
- 2- Obtener la solución numérica del sistema lineal para evaluar cuantitativamente el comportamiento con intención de verificar el correcto planteo. Para hacerlo, se le asignan los valores siguientes a los parámetros, son  $\omega = 0,2$ ;  $a = 0,01$ ;  $b = 2$ ;  $c = 100$  m/s, (es decir, 360Km/h),  $\Delta t = 10^{-3}$ ; y el tiempo de simulación de 5 segundos.
- 3- Obtener la solución numérica del sistema lineal para  $c = 50$  m/s, (es decir, 180Km/h),  $\Delta t = 10^{-3}$ ; y el tiempo de simulación de 20 segundos.

## Ejercicio 2. Cálculo simbólico en de linealización de sistemas dinámicos

Se propone implementar simbólicamente las expresiones de Taylor para el caso de un sistema dinámico con varias variables de estado.

Sea el sistema dinámico no lineal expresado en variables de estado dado por

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (1-8)$$

para la ecuación de estados, y

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}, \mathbf{u}, t) \quad (1-9)$$

para la ecuación de salida, se pretenden hallar una expresión lineal del mismo.

Para ello, si las funciones  $\mathbf{f}$  y  $\mathbf{g}$  fuesen vectoriales, lo cual significa que  $\mathbf{Y} \in \mathbb{R}^m$ ,  $\mathbf{X} \in \mathbb{R}^n$  y  $\mathbf{U} \in \mathbb{R}^r$  tienen varias componentes, entonces se emplean las siguientes expresiones para el cálculo el modelo linealizado

$$\begin{aligned} \dot{\mathbf{x}}_i &= \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) + \sum_{j=1}^n \left. \frac{\partial f_i(\mathbf{x}, \mathbf{u}_t)}{\partial x_j} \right|_{\mathbf{x}_0, \mathbf{u}_0} (\mathbf{x}_j - \mathbf{x}_{0j}) + \sum_{j=1}^r \left. \frac{\partial f_i(\mathbf{x}, \mathbf{u}_t)}{\partial u_j} \right|_{\mathbf{x}_0, \mathbf{u}_0} (\mathbf{u}_j - \mathbf{u}_{0j}), \quad i = 1, 2, \dots, n. \\ \mathbf{y}_h &= \mathbf{g}(\mathbf{x}_0, \mathbf{u}_0) + \sum_{j=1}^n \left. \frac{\partial g_h(\mathbf{x}, \mathbf{u}_t)}{\partial x_j} \right|_{\mathbf{x}_0, \mathbf{u}_0} (\mathbf{x}_j - \mathbf{x}_{0j}) + \sum_{j=1}^r \left. \frac{\partial g_h(\mathbf{x}, \mathbf{u}_t)}{\partial u_j} \right|_{\mathbf{x}_0, \mathbf{u}_0} (\mathbf{u}_j - \mathbf{u}_{0j}), \quad h = 1, 2, \dots, m. \end{aligned} \quad (1-10)$$

La (1-10) toma la forma

$$\begin{aligned} \dot{\mathbf{x}}_t &= \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t \\ \mathbf{y}_t &= \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t \end{aligned} \quad (1-11)$$

con  $\mathbf{x}_0 = \mathbf{x}(0)$ , donde  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  y  $\mathbf{D}$  son matrices.

## Ejercicio 3. Linealización de un sistema no lineal y verificación en simulación numérica

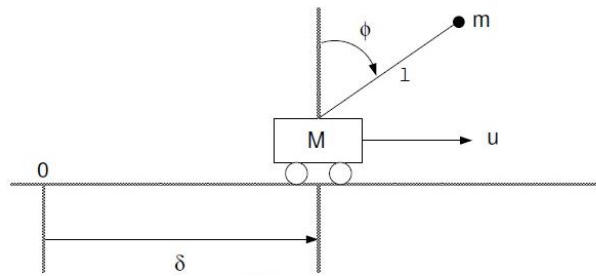


Fig. 1-4. Sistemas para modelar, extraído de [1].

Para el caso del esquema del péndulo invertido de la Fig. 1-4 se tiene,

$$\begin{cases} (M+m)\ddot{\delta} + m l \ddot{\phi} \cos \phi - m l \dot{\phi}^2 \sin \phi + F \dot{\delta} = u \\ l \ddot{\phi} - g \sin \phi + \ddot{\delta} \cos \phi = 0 \end{cases} \quad (12)$$

donde el sistema lineal en variables de estado  $x = [\delta \quad \dot{\delta} \quad \phi \quad \dot{\phi}]^T$ , con los valores de los coeficientes de  $m=0,1$ ;  $F=0,1$ ;  $l=0,6$ ;  $g=9,8$ ;  $M=0,5$  y  $\Delta t=10^{-4}$  seg, tomando un tiempo de simulación de 5 segundos con  $u=0$ .

```

fi_pp=
      Fricc*p_p + M*fi*g - fi*fi_p^2*long*m + fi*g*m - u
      -----
                    M*long
p_pp=
      -Fricc*p_p + fi*fi_p^2*long*m - fi*g*m + u
      -----
                    M

```

Fig. 1-5. Respuesta del Octave en el equilibrio inestable cuando se usa el paquete >> pkg load symbolic.

Se pide:

1. Obtener la versión linealizada en los equilibrios del sistema empleando el paquete de cálculo simbólico, tomando como referencia las instrucciones del Script de la Tabla 1-1. Se sugiere realizar la aproximación de las funciones coseno y seno por su variable correspondiente en el punto de operación, y luego mediante el soft obtener las expresiones que se muestran en la Fig. 1-5, donde  $fi\_pp$  es  $\ddot{\phi}$  y  $p\_pp$  es  $\ddot{\delta}$ .
- 2- Obtener simulaciones del sistema (12) en las condiciones iniciales  $x_0 = [0 \ 0 \ -0,01 \ 0]^T$  y  $x_0 = [0 \ 0 \ 3,2 \ 0]^T$ , empleando una integración Euler con  $\Delta t=10^{-4}$ . El tiempo de simulación será de 10 segundos en cada caso, con  $u=0$ .
- 3- Modificar la masa  $m$  al doble y repetir la operación.
- 4- Obtener la representación lineal en variables de estado para el equilibrio estable, para lo que se sugiere obtener la versión simplificada con las funciones seno y coseno linealizadas, y luego sí despejar las derivadas segundas como muestra la Fig. 1-6, donde  $fi\_pp$  es  $\ddot{\phi}$  y  $p\_pp$  es  $\ddot{\delta}$ .

- 5- Obtener la solución numérica de los dos sistemas, del lineal y del no lineal para evaluar cuantitativamente la equivalencia, modificando  $m$  de 0,1 a 0,01 y la longitud  $l$  a 1,2m.
- 6- Obtener el sistema lineal para el equilibrio estable  $x = [0 \ 0 \ \pi \ 0]^T$ .
- 7- Obtener la solución numérica de los dos sistemas, del lineal y del no lineal para evaluar cuantitativamente la equivalencia en el equilibrio estable modificando  $m$  de 0,1 a 0,01 y cambiar la longitud  $l$  a 1,2m.

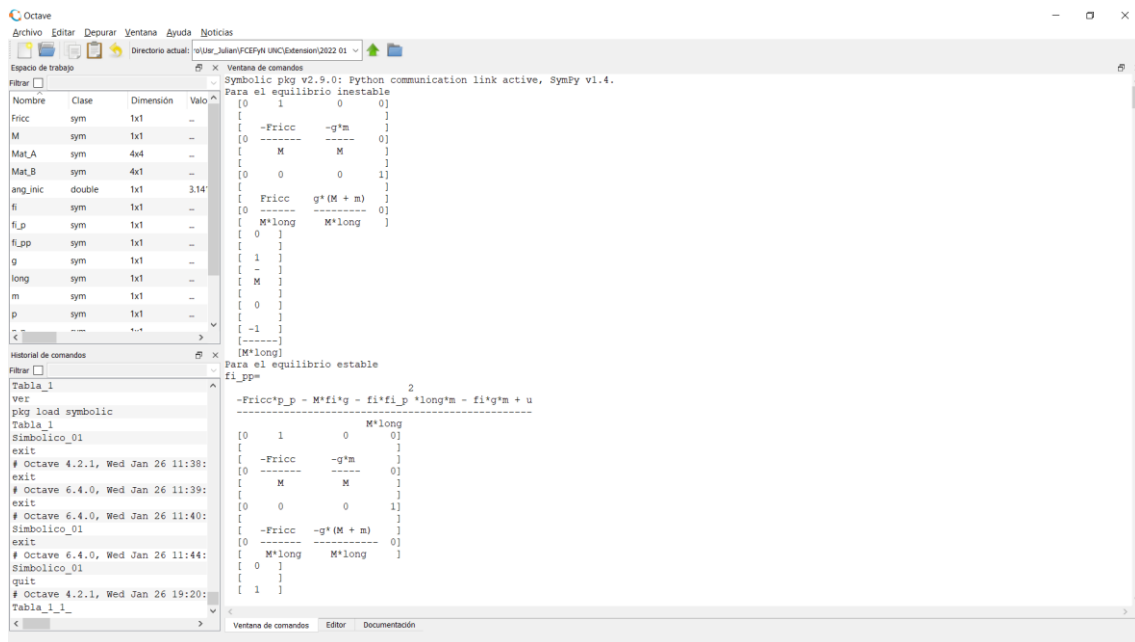


Fig. 1-6. Respuesta del Octave en el equilibrio inestable (y Estable abajo) cuando se usa el paquete `>> pkg load symbolic`.

```
pkg load symbolic %Primera vez que se corre el paquete simbólico de Python.
clear all; clc;
syms fi fi_p fi_pp p p_p p_pp M m u long Fricc g;
disp('Para el equilibrio inestable')
ang_inic=0;
p_pp=(1/(M+m))*(u-m*long*fi_pp+m*long*fi_p^2*fi-Fricc*p_p); %Pequeños angulos
% fi_pp=(1/long)*(g*sin(fi)-p_pp*cos(fi)); %Expresión completa
% fi_pp=(1/long)*(g*(fi)-p_pp); %Pequeños angulos para fi~0, sin(fi)~fi, cos(fi)~1
fi_pp=solve(fi_pp==(1/long)*(g*fi-p_pp),fi_pp);
%disp('fi_pp='); pretty(simplify(fi_pp));
p_pp=subs(p_pp,'fi_pp',fi_pp);
%disp('p_pp='); pretty(simplify(p_pp))
Mat_A=[[0 1 0 0];
        [subs(subs(subs(subs(diff(p_pp, p), p,0),p_p,0),fi,ang_inic),fi_p,0), ...
        subs(subs(subs(subs(diff(p_pp, p_p), p,0),p_p,0),fi,ang_inic),fi_p,0), ...
        subs(subs(subs(subs(diff(p_pp, fi), p,0),p_p,0),fi,ang_inic),fi_p,0), ...
        subs(subs(subs(subs(diff(p_pp, fi_p), p,0),p_p,0),fi,ang_inic),fi_p,0)]];
[0 0 0 1];
[subs(subs(subs(subs(diff(fi_pp, p), p,0),p_p,0),fi,ang_inic),fi_p,0),...
subs(subs(subs(subs(diff(fi_pp, p_p), p,0),p_p,0),fi,ang_inic),fi_p,0),...
subs(subs(subs(subs(diff(fi_pp, fi), p,0),p_p,0),fi,ang_inic),fi_p,0),...
subs(subs(subs(subs(diff(fi_pp, fi_p), p,0),p_p,0),fi,ang_inic),fi_p,0)]];
Mat_B=[0;
        subs(subs(subs(subs(diff(p_pp, u), p,0),p_p,0),fi,ang_inic),fi_p,0);...
```

```

0;
subs(subs(subs(subs(diff(fi_pp, u), p,0),p_p,0),fi,ang_inic),fi_p,0)];
pretty(simplify(Mat_A))
pretty(simplify(Mat_B))
syms fi fi_p fi_pp p p_p p_pp M m u long Fricc g;
disp('Para el equilibrio estable')
ang_inic=pi;
p_pp=(1/(M+m))*(u+m*long*fi_pp-m*long*fi_p^2*fi-Fricc*p_p);
%fi_pp=(1/long)*(-g*(fi)+p_pp); %Pequeños angulos para fi~pi sin(fi)~-fi, cos(fi)=-1
fi_pp=solve(fi_pp==(1/long)*(-g*fi+p_pp),fi_pp);
disp('fi_pp='); pretty(simplify(fi_pp));
p_pp=subs(p_pp,'fi_pp',fi_pp);
Mat_A=[0 1 0 0];
[subs(subs(subs(subs(diff(p_pp, p), p,0),p_p,0),fi,ang_inic),fi_p,0), ...
subs(subs(subs(subs(diff(p_pp, p_p), p,0),p_p,0),fi,ang_inic),fi_p,0), ...
subs(subs(subs(subs(diff(p_pp, fi), p,0),p_p,0),fi,ang_inic),fi_p,0), ...
subs(subs(subs(subs(diff(p_pp, fi_p), p,0),p_p,0),fi,ang_inic),fi_p,0)]; ...
[0 0 0 1];...
[subs(subs(subs(subs(diff(fi_pp, p), p,0),p_p,0),fi,ang_inic),fi_p,0),...
subs(subs(subs(subs(diff(fi_pp, p_p), p,0),p_p,0),fi,ang_inic),fi_p,0),...
subs(subs(subs(subs(diff(fi_pp, fi), p,0),p_p,0),fi,ang_inic),fi_p,0),...
subs(subs(subs(subs(diff(fi_pp, fi_p), p,0),p_p,0),fi,ang_inic),fi_p,0)]];
Mat_B=[0;
subs(subs(subs(subs(diff(p_pp, u), p,0),p_p,0),fi,ang_inic),fi_p,0);...
0;
subs(subs(subs(subs(diff(fi_pp, u), p,0),p_p,0),fi,ang_inic),fi_p,0)];
pretty(simplify(Mat_A))
pretty(simplify(Mat_B))

```

Tabla 1-1. Código orientativo de operación simbólica para Octave compatible con Matlab ®.

- [1] Sontag. Mathematical control theory 1998. Pag 104. <http://www.sontaglab.org>.  
[2] <https://es.wikipedia.org/wiki/FreeMat>