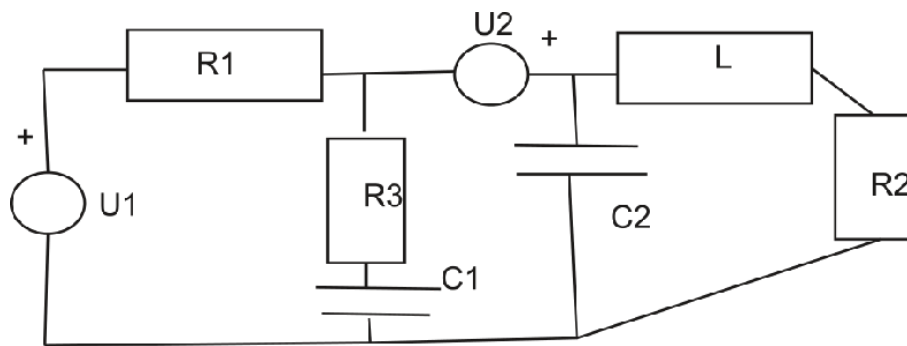


Modelado circuito eléctrico

(c) 2022 A. Sala, Universitat Politècnica de Valencia. Todos los derechos reservados.

Vídeo-presentación disponible en: <http://personales.upv.es/asala/YT/V/cir1.html>



declaremos los símbolos a usar:

```
syms U1 U2 VR1 iR1 dVC1 VC1 iC1 dVC2 VC2 iC2 diL VL VR2 iL VR3 iU2 real
```

y los parámetros constantes:

```
R1=100; R2=100; R3=100; C1=1e-3; C2=2e-3; L=2e-2;
```

Introduzcamos las ecuaciones de la física (*Nota: La sintaxis es diferente en otras versiones del Symbolic Toolbox; en este caso se ha usado Matlab R2016a*):

```
dinamica=[dVC1==iC1/C1; dVC2==iC2/C2; diL==VL/L];  
resistencias=[VR1==iR1*R1; VR2==iL*R2; VR3==iC1*R3];  
balances=[iR1==iU2+iC1; iU2==iL+iC2; U1==VC1+VR3+VR1; VR3+VC1==VC2-U2; VL+VR2==VC2];  
Modelo =[dinamica; resistencias ; balances];  
Modelo
```

Modelo =

$$\begin{pmatrix} dVC_1 = 1000 iC_1 \\ dVC_2 = 500 iC_2 \\ diL = 50 VL \\ VR_1 = 100 iR_1 \\ VR_2 = 100 iL \\ VR_3 = 100 iC_1 \\ iR_1 = iC_1 + iU_2 \\ iU_2 = iC_2 + iL \\ U_1 = VC_1 + VR_1 + VR_3 \\ VC_1 + VR_3 = VC_2 - U_2 \\ VL + VR_2 = VC_2 \end{pmatrix}$$

Comprobemos cuántas ecuaciones tiene el modelo:

```
length(Modelo)
```

```
ans = 11
```

y cuántas variables simbólicas aparecen:

```
symvar(Modelo)
```

```
ans = (U1 U2 VC1 VC2 VL VR1 VR2 VR3 dVC1 dVC2 diL iC1 iC2 iL iR1 iU2)
```

```
length(symvar(Modelo))
```

```
ans = 16
```

Se pueden "despejar", por tanto, 11 símbolos en función de otros cinco.

Para obtener la representación interna (EDO) normalizada:

```
Incognitas=[iR1, VR1, VR2, VR3, diL, VL, iC1, dVC1, iC2, dVC2, iU2];  
length(Incognitas)
```

```
ans = 11
```

Podemos resolver el sistema de 11 ecs con 11 incognitas, en función de las cinco letras restantes que son (U1, U2, VC1, VC2, iL)

```
sol=solve(Modelo, Incognitas);
```

Como no ha dado ningún error, el modelo, era, en efecto, un sistema de ecuaciones algebraico-diferencial de índice 1 dado que hemos podido despejar todo en función de las variables de estado y las entradas:

```
%Entradas  
VectorEntradas=[U1;U2];  
% vbles estado, aquellas bajo signo derivada:  
VectorEstado=[VC1;VC2;iL];
```

Las ecuaciones de estado normalizadas son:

```
ecs_estado=[sol.dVC1;sol.dVC2;sol.diL]
```

```
ecs_estado =
```

$$\begin{pmatrix} 10 VC_2 - 10 VC_1 - 10 U_2 \\ 5 U_1 + 10 U_2 + 5 VC_1 - 10 VC_2 - 500 iL \\ 50 VC_2 - 5000 iL \end{pmatrix}$$

Las ecuaciones de salida son cualesquiera de las incógnitas de interés... por ejemplo:

```
%solo nos interesa VC2, la intens por R1, y la intens. por fuente U2  
ecs_salida=[VC2; sol.iR1;sol.iU2]
```

```
ecs_salida =
```

$$\begin{pmatrix} VC_2 \\ \frac{U_1}{100} + \frac{U_2}{100} - \frac{VC_2}{100} \\ \frac{U_1}{100} + \frac{U_2}{50} + \frac{VC_1}{100} - \frac{VC_2}{50} \end{pmatrix}$$

Nota: VC2 no necesita "despejarse" en función del estado y entradas porque directamente ES una vble de estado. (poner sol.VC2 sería un error, porque NO es incógnita).

En forma matricial:

```
A=jacobian(ecs_estado,VectorEstado)
```

A =

$$\begin{pmatrix} -10 & 10 & 0 \\ 5 & -10 & -500 \\ 0 & 50 & -5000 \end{pmatrix}$$

```
B=jacobian(ecs_estado,VectorEntradas)
```

B =

$$\begin{pmatrix} 0 & -10 \\ 5 & 10 \\ 0 & 0 \end{pmatrix}$$

```
C=jacobian(ecs_salida,VectorEstado)
```

C =

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & -\frac{1}{100} & 0 \\ \frac{1}{100} & -\frac{1}{50} & 0 \end{pmatrix}$$

```
D=jacobian(ecs_salida,VectorEntradas)
```

D =

$$\begin{pmatrix} 0 & 0 \\ \frac{1}{100} & \frac{1}{100} \\ \frac{1}{100} & \frac{1}{50} \end{pmatrix}$$

Con lo que podríamos expresar el modelo normalizado:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

Siendo x un vector de longitud 3 (VectorEstado), u de longitud 2 (VectorEntradas), e y de longitud 3 ($y=(VC2, iR1, iU2)$).

Vamos a simular, por ejemplo, el transitorio con alimentación CC ($U1, U2$ constantes):

```
modelocircuito=ss(eval(A),eval(B),eval(C),eval(D));  
% eval es para pasar de simbólico (cadena de caracteres/texto) a numérico  
modelocircuito.OutputName={'VC2','iR1','iU2'};  
modelocircuito.InputName={'U1','U2'};  
step(modelocircuito)
```

