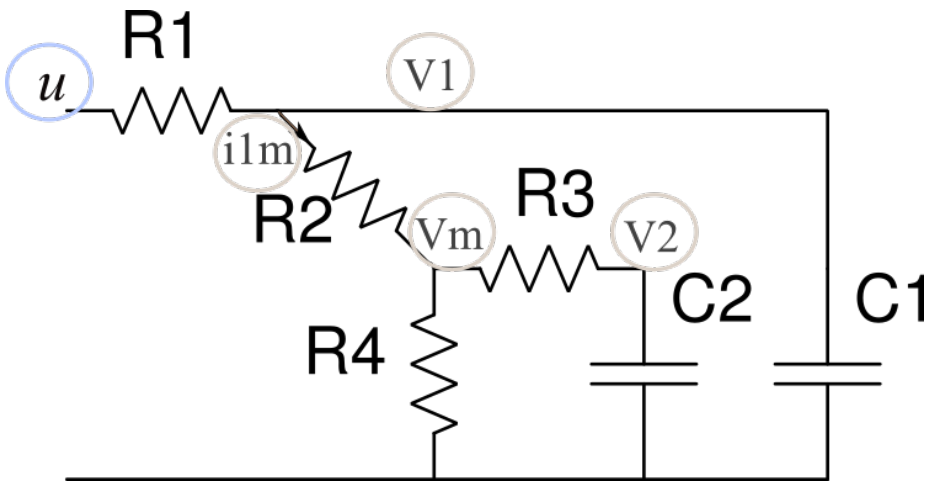
Modelado de un circuito eléctrico (LFT) con DOS parámetros físicos inciertos.

Vídeo-presentación en https://youtu.be/jDPdPuJyeA0



Parámetros nominales:

```
C1_0=1e-3;C2=2e-3; R1=8;R3=16; R4=32; R2_0=4;
syms dv1 dv2 V1 V2 incR2 incC1 Vm u i1m
```

standard model

La resistencia "real" R2 es R2_0+incR2, el condensador "real" es C1_0+incC1 modelo es:

```
R2=R2_0+incR2;C1=C1_0+incC1;
 Model1=[C1*dv1==(u-V1)/R1-i1m; ...
   C2*dv2==(Vm-V2)/R3; ...
   i1m==(Vm-V2)/R3+Vm/R4; (V1-Vm)==R2*i1m ]
```

```
Model1 =
```

$$\begin{pmatrix} \mathrm{dv}_1\left(\mathrm{incC}_1+\frac{1}{1000}\right)=\frac{u}{8}-\mathrm{i1m}-\frac{V_1}{8} \\ \frac{\mathrm{dv}_2}{500}=\frac{\mathrm{Vm}}{16}-\frac{V_2}{16} \\ \mathrm{i1m}=\frac{3\,\mathrm{Vm}}{32}-\frac{V_2}{16} \\ V_1-\mathrm{Vm}=\mathrm{i1m}\,(\mathrm{incR}_2+4) \end{pmatrix}$$

```
Unknowns=[dv1,i1m,Vm,dv2]; %remaining V1,V2,incR2,u
SolvedModel=solve(Model1,Unknowns)
```

```
SolvedModel = struct with fields:
```

```
dv1: (125*(16*V2 - 68*V1 + 44*u - 3*V1*incR2 + 3*incR2*u))/((3*incR2 + 44)*(1000*incC1 + 1))
i1m: (3*V1 - 2*V2)/(3*incR2 + 44)
 Vm: (2*(16*V1 + 4*V2 + V2*incR2))/(3*incR2 + 44)
dv2: -(125*(36*V2 - 32*V1 + V2*incR2))/(4*(3*incR2 + 44))
```

```
StateEqs=simplify([SolvedModel.dv1; SolvedModel.dv2])
```

```
StateEqs =
```

$$\begin{pmatrix} \frac{125\ (16\ V_2 - 68\ V_1 + 44\ u - 3\ V_1\ \mathrm{incR}_2 + 3\ \mathrm{incR}_2\ u)}{(3\ \mathrm{incR}_2 + 44)\ (1000\ \mathrm{incC}_1 + 1)} \\ -\frac{125\ (36\ V_2 - 32\ V_1 + V_2\ \mathrm{incR}_2)}{4\ (3\ \mathrm{incR}_2 + 44)} \end{pmatrix}$$

```
A=jacobian(StateEqs,[V1,V2])
```

```
A =
```

$$\begin{pmatrix} -\frac{125\ (3\ \mathrm{incR}_2 + 68)}{(3\ \mathrm{incR}_2 + 44)\ (1000\ \mathrm{incC}_1 + 1)} & \frac{2000}{(3\ \mathrm{incR}_2 + 44)\ (1000\ \mathrm{incC}_1 + 1)} \\ \frac{1000}{3\ \mathrm{incR}_2 + 44} & -\frac{125\ (\mathrm{incR}_2 + 36)}{4\ (3\ \mathrm{incR}_2 + 44)} \end{pmatrix}$$

```
B=jacobian(StateEqs,[u])
```

```
B =
```

$$\begin{pmatrix} \frac{125}{1000\ \mathrm{incC}_1 + 1} \\ 0 \end{pmatrix}$$

Introduzcamos ecuación de salida... por ejemplo, se dispone de sensor/objetivos de control en V1 y Vm

```
OutputEqs=[V1;SolvedModel.Vm];
C=jacobian(OutputEqs,[V1 V2])
```

```
C =
```

$$\begin{pmatrix} 1 & 0 \\ \frac{32}{3\ \mathrm{incR}_2 + 44} & \frac{2\ (\mathrm{incR}_2 + 4)}{3\ \mathrm{incR}_2 + 44} \end{pmatrix}$$

```
D=jacobian(OutputEqs,[u])
```

```
D =
```

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

El modelo "nominal" sería:

```
incR2=0;incC1=0;
Anum=eval(A)
```

```
Anum = 2×2
 -193.1818   45.4545
```

```
   22.7273  -25.5682
```

```
Bnum=eval(B)
```

```
Bnum = 2×1
   125
     0
```

```
Cnum=eval(C)
```

```
Cnum = 2×2
    1.0000         0
    0.7273    0.1818
```

```
Dnum=eval(D)
```

```
Dnum = 2×1
     0
     0
```

```
circuitnominal=ss(Anum,Bnum,Cnum,Dnum);
```

introducing changes in the model to incorporate LFT

Redeclaremos incR2 como simbólica (borremos su valor de cero de antes), y introduzcamoslo en forma LFT

```
syms incR2 incC1 w_deltaR w_deltaC  Ana   %extra symbolic variables
Model_sinR2C1=[ dv1==Ana; ...
   C2*dv2==(Vm-V2)/R3; ...
   i1m==(Vm-V2)/R3+Vm/R4;];
EqResistor2_LFT= ( (V1-Vm)==R2_0*i1m+w_deltaR  );
EqC1_LFT= ( C1_0*Ana+w_deltaC==(u-V1)/R1-i1m );
Eq_Uncertainty = [w_deltaR==incR2*i1m; w_deltaC==incC1*Ana];
```

La parte lineal "cierta" del modelo es esta:

```
Model2=[Model_sinR2C1; EqResistor2_LFT; EqC1_LFT]
```

```
Model2 =
```

$$\begin{pmatrix} \mathrm{dv}_1 = \mathrm{Ana} \\ \frac{\mathrm{dv}_2}{500} = \frac{\mathrm{Vm}}{16} - \frac{V_2}{16} \\ \mathrm{i1m} = \frac{3\,\mathrm{Vm}}{32} - \frac{V_2}{16} \\ V_1 - \mathrm{Vm} = 4\,\mathrm{i1m} + w_{\mathrm{deltaR}} \\ \frac{\mathrm{Ana}}{1000} + w_{\mathrm{deltaC}} = \frac{u}{8} - \mathrm{i1m} - \frac{V_1}{8} \end{pmatrix}$$

El modelo nominal será w_deltaR=w_deltaC=0.

El modelo completo será (equiv. a Model 1):

```
Model3=[Model2; Eq_Uncertainty]
```

```
Model3 =
```

$$\begin{pmatrix} \mathrm{dv}_1 = \mathrm{Ana} \\ \frac{\mathrm{dv}_2}{500} = \frac{\mathrm{Vm}}{16} - \frac{V_2}{16} \\ \mathrm{i1m} = \frac{3\,\mathrm{Vm}}{32} - \frac{V_2}{16} \\ V_1 - \mathrm{Vm} = 4\,\mathrm{i1m} + w_{\mathrm{deltaR}} \\ \frac{\mathrm{Ana}}{1000} + w_{\mathrm{deltaC}} = \frac{u}{8} - \mathrm{i1m} - \frac{V_1}{8} \\ w_{\mathrm{deltaR}} = \mathrm{i1m}\,\mathrm{incR}_2 \\ w_{\mathrm{deltaC}} = \mathrm{Ana}\,\mathrm{incC}_1 \end{pmatrix}$$

```
Unknowns3=[Unknowns w_deltaR w_deltaC Ana ];
SolvedModel3=solve(Model3, Unknowns3)
```

```
SolvedModel3 = struct with fields:
         dv1: (125*(16*V2 - 68*V1 + 44*u - 3*V1*incR2 + 3*incR2*u))/(44000*incC1 + 3*incR2 + 3000*incC1*in
         i1m: (3*V1 - 2*V2)/(3*incR2 + 44)
          Vm: (2*(16*V1 + 4*V2 + V2*incR2))/(3*incR2 + 44)
         dv2: -(125*(36*V2 - 32*V1 + V2*incR2))/(4*(3*incR2 + 44))
    w_deltaR: (3*V1*incR2 - 2*V2*incR2)/(3*incR2 + 44)
    w_deltaC: (125*incC1*(16*V2 - 68*V1 + 44*u - 3*V1*incR2 + 3*incR2*u))/(44000*incC1 + 3*incR2 + 3000*in
         Ana: (125*(16*V2 - 68*V1 + 44*u - 3*V1*incR2 + 3*incR2*u))/(44000*incC1 + 3*incR2 + 3000*incC1*in
```

Que resulta, en tanto a dv1, dv2, i1m, Vm, exactamente igual que el Model1. Comprobemos, por ejemplo A

```
LaMismaA=simplify(jacobian([SolvedModel3.dv1, SolvedModel3.dv2],[V1 V2]))
```

```
LaMismaA =
```

$$\begin{pmatrix} -\frac{125\ (3\,\mathrm{incR}_2 + 68)}{(3\,\mathrm{incR}_2 + 44)\ (1000\,\mathrm{incC}_1 + 1)} & \frac{2000}{(3\,\mathrm{incR}_2 + 44)\ (1000\,\mathrm{incC}_1 + 1)} \\ \frac{1000}{3\,\mathrm{incR}_2 + 44} & -\frac{125\ (\mathrm{incR}_2 + 36)}{4\ (3\,\mathrm{incR}_2 + 44)} \end{pmatrix}$$

```
A-LaMismaA
```

```
ans =
```

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Modelado LFT: NO usar Eq_Uncertainty, con lo que w_delta es una entrada "virtual"

```
SolvedModelLFT=solve(Model2,[Unknowns Ana])
```

```
SolvedModelLFT = struct with fields:
    dv1: (500*V2)/11 - (2125*V1)/11 + 125*u - 1000*w_deltaC + (750*w_deltaR)/11
    i1m: (3*V1)/44 - V2/22 - (3*w_deltaR)/44
     Vm: (8*V1)/11 + (2*V2)/11 - (8*w_deltaR)/11
```

```
dv2: (250*V1)/11 - (1125*V2)/44 - (250*w_deltaR)/11
Ana: (500*V2)/11 - (2125*V1)/11 + 125*u - 1000*w_deltaC + (750*w_deltaR)/11
```

El resultado, claro, depende de w_deltaR y w_deltaC

```
StateEqsLFTModel=[SolvedModelLFT.dv1;SolvedModelLFT.dv2]
```

StateEqsLFTModel =

$$\begin{pmatrix} \frac{500\,V_2}{11} - \frac{2125\,V_1}{11} + 125\,u - 1000\,w_{\mathrm{deltaC}} + \frac{750\,w_{\mathrm{deltaR}}}{11} \\ \frac{250\,V_1}{11} - \frac{1125\,V_2}{44} - \frac{250\,w_{\mathrm{deltaR}}}{11} \end{pmatrix}$$

Representación interna completa como upper LFT

```
A=jacobian(StateEqsLFTModel,[V1,V2])
```

A =

$$\begin{pmatrix} -\frac{2125}{11} & \frac{500}{11} \\ \frac{250}{11} & -\frac{1125}{44} \end{pmatrix}$$

```
GeneralisedInput=[w_deltaR, w_deltaC, u];
B_lft=jacobian(StateEqsLFTModel, GeneralisedInput)
```

B_lft =

$$\begin{pmatrix} \frac{750}{11} & -1000 & 125 \\ -\frac{250}{11} & 0 & 0 \end{pmatrix}$$

```
GeneralisedOutputEq=[SolvedModelLFT.i1m;SolvedModelLFT.Ana;V1;SolvedModelLFT.Vm]; %añad
C_lft=jacobian(GeneralisedOutputEq,[V1,V2])
```

C_lft =

$$\begin{pmatrix} \frac{3}{44} & -\frac{1}{22} \\ -\frac{2125}{11} & \frac{500}{11} \\ 1 & 0 \\ \frac{8}{11} & \frac{2}{11} \end{pmatrix}$$

```
D_lft=jacobian(GeneralisedOutputEq,GeneralisedInput)
```

D_lft =

$$\begin{pmatrix} -\frac{3}{44} & 0 & 0 \\ \frac{750}{11} & -1000 & 125 \\ 0 & 0 & 0 \\ -\frac{8}{11} & 0 & 0 \end{pmatrix}$$

Comprobemos que la interconexión LFT da el resultado correcto (modelo original):

```
incR2=0; incC1=0;
sysuncert=diag([incR2;incC1]);
sys2=ss(eval(A),eval(B_lft),eval(C_lft),eval(D_lft));
circuito_otra_vez=lft(sysuncert,sys2);
norm(circuito_otra_vez-circuitnominal,'inf')
```

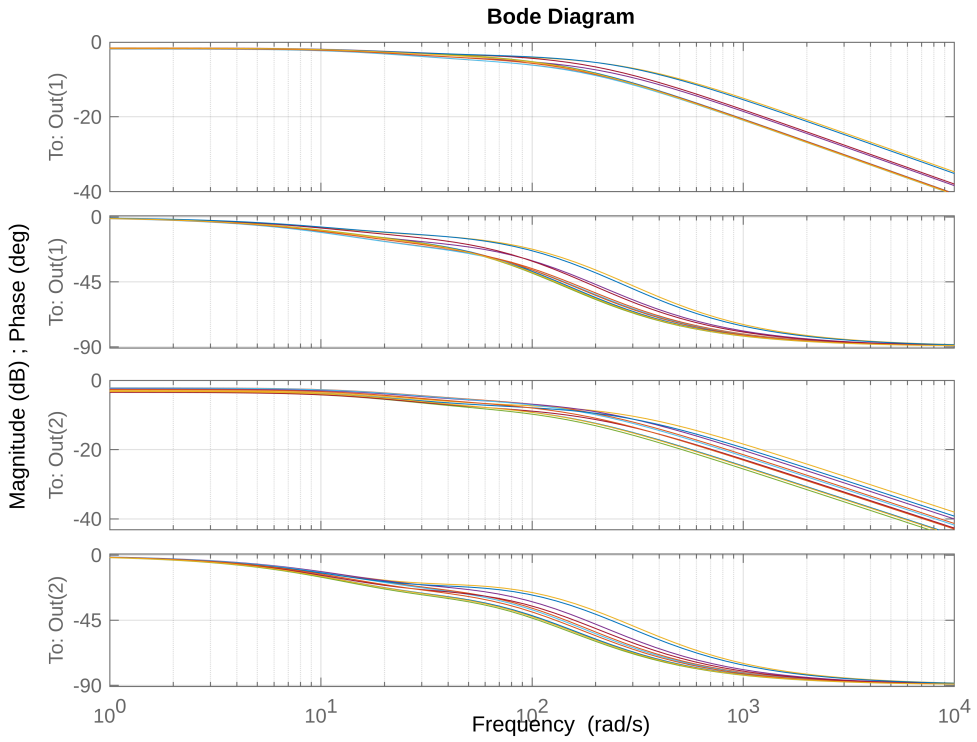
```
ans = 2.7756e-17
```

Simulemos variaciones de resistencia y capacidades al azar:

```
for i=1:10
    incR2=8*(rand()-.5); incC1=8e-4*(rand()-.5);
    sysuncert=diag([incR2;incC1]);
    circmodif=lft(sysuncert,sys2);
    bode(circmodif)
    hold on
    [incR2 incC1*1e4 norm(circmodif-circuitnominal,'inf')]
end
```

```
ans = 1×3
    2.5178    3.2463    0.1334
ans = 1×3
   -2.9841    3.3070    0.1025
ans = 1×3
    1.0589   -3.2197    0.1437
ans = 1×3
   -1.7720    0.3751    0.0458
ans = 1×3
    3.6601    3.7191    0.1571
ans = 1×3
   -2.7391    3.7647    0.1107
ans = 1×3
    3.6573   -0.1170    0.0771
ans = 1×3
    2.4022   -2.8649    0.1211
ans = 1×3
   -0.6259    3.3259    0.1063
ans = 1×3
    2.3377    3.6759    0.1432
```

```
hold off, grid on
```



Márgen de estabilidad ante incertidumbre no estructurada (R2 no lineal, con retardo, dinámica...)

```
mgst=1/norm(sys2(1:2,1:2),'inf')
```

```
mgst = 9.9768e-04
```

```
```

El resultado da un margen "pequeño"... problema: se ha mezclado incR2 con incC1, tienen unidades distintas... Necesaria "ponderación" (normalización) para que los márgenes tengan significado, mgst no se sabe las unidades que tiene.