# Anàlisi de circuits amb Octave

**Circuits dinàmics**

Pere Palà

Octubre de 2011

En aquest document expliquem com podem ajudar-nos amb Octave per analitzar circuits dinàmics. Octave *no* és un programa d'anàlisi de circuits, però es pot fer servir per als càlculs que cal dur a terme. Octave o Matlab, la versió no lliure en què s'inspira, són **les** eines matemàtiques que fan servir els enginyers electrònics i/o de telecomunicació.

## 1 Càlculs freqüents

En `octave`, podem representar un polinomi mitjançant un vector. El conveni és que els vectors comencen pel terme de major grau. Així, el polinomi

$$P(s) = s^4 + 3s^3 + 4s^2 + 5s + 1 \tag{1}$$

pot ser descrit mitjançant un vector, de la forma:

```
octave:1> P=[1 3 4 5 1]
P =
   1   3   4   5   1
```

Per calcular les arrels d'aquest polinomi, tenim l'ordre `roots`:

```
octave:2> roots(P)
ans =
  -2.12074 + 0.00000i
  -0.32072 + 1.37110i
  -0.32072 - 1.37110i
  -0.23781 + 0.00000i
```

Observem que aquest polinomi té dues arrels reals negatives i un parell d'arrels complexes conjugades.

Per saber el valor d'un polinomi en un punt qualsevol (real o complexe) podem fer servir l'ordre `polyval`. Per exemple, per avaluar el valor del polinomi anterior a $s = 2+2j$, escriurem

```
octave:3> polyval(P,2+2*j)
ans = -101 +  90i
```

El resultat d'aquestes ordres queda emmagatzemat per defecte a la variable `ans`, però ho podem assignar a qualsevol variable que ens interessi guardar per operacions posteriors.

Per exemple, per comprovar que la primera de les arrels que hem calculat és, efectivament, una arrel, podem fer

```
octave:7> r=roots(P);
octave:8> polyval(P,r(1))
ans =  1.2434e-14
```

Però també podem comprovar totes les arrels simultàniament. Donat que `octave` és una eina orientada a vectors, la majoria de funcions operen sobre vectors igual que sobre escalars:

```
octave:9> polyval(P,r)
ans =
  1.2434e-14 + 0.0000e+00i
  -8.6597e-15 + 1.3101e-14i
  -8.6597e-15 - 1.3101e-14i
  0.0000e+00 + 0.0000e+00i
```

Veiem, doncs, que les arrels s'han calculat amb una precisió de l'ordre de $10^{-14}$.

De cara a la presentació dels resultats, podem jugar amb la comanda `format`. Si busqueu informació sobre la comanda, `doc format`, trobareu informació al respecte. De moment, podeu provar què passa amb les ordres següents

```
octave:1> pi
ans =  3.1416
octave:2> format short
octave:3> pi
ans =  3.1416
octave:> format short e
octave:4> pi
ans =  3.1416e+00
octave:5> format long e
octave:6> pi
ans =  3.14159265358979e+00
octave:7> format bank
octave:8> pi
ans = 3.14
```

## 2 Descomposició en fraccions simples

Considerem una funció $F(s)$ com la següent

$$F(s) = \frac{s^2 + 4.5s + 2}{s^4 + 5s^3 + 10s^2 + 10s + 4} \tag{2}$$

Després d'emplenar els vectors `num` i `den` adequadament, fem servir la comanda `residue` per calcular els anomenats *residus* corresponents als pols de $F(s)$:

```
octave:1> [r,p,k,e]=residue(num,den)
r =
  1.5000e+00 - 6.9389e-16i
  9.9920e-16 - 1.2500e+00i
  9.2635e-16 + 1.2500e+00i
  -1.5000e+00 + 3.0866e-17i
p =
  -2.00000 + 0.00000i
  -1.00000 + 1.00000i
  -1.00000 - 1.00000i
  -1.00000 + 0.00000i
k = [](0x0)
```

```
e =
   1
   1
   1
   1
```

És a dir, el resultat ens està dient que

$$F(s) = \frac{s^2 + 4.5s + 2}{s^4 + 5s^3 + 10s^2 + 10s + 4} = \frac{1.5}{s+2} + \frac{-1.25j}{s+1-j} + \frac{1.25j}{s+1+j} + \frac{-1.5}{s+1} \tag{3}$$

En aquesta forma, la transformada inversa és molt més senzilla de realitzar. El vector `e` conté informació sobre la multiplicitat del pol corresponent. Podem comprovar-ho amb un exemple senzill, on el denominador és $(s+1)^2 = s^2 + 2s + 1$:

$$F(s) = \frac{1}{s^2 + 2s + 1} \tag{4}$$

```
octave:1> [r,p,k,e]=residue(1,[1 2 1])
r =
   0
   1
p =
  -1
  -1
k = [](0x0)
e =
   1
   2
```

El resultat ens està dient que la descomposició en fraccions simples és, com ja sabem:

$$F(s) = \frac{1}{s^2 + 2s + 1} = \frac{0}{(s+1)^1} + \frac{1}{(s+1)^2} \tag{5}$$

Finalment, cal esmentar el vector `k`, format per la divisió dels polinomis, quan el grau del numerador és igual o superior al del denominador. Per exemple, si

$$F(s) = \frac{s+2}{s+1} \tag{6}$$

aleshores $F(s)$ es pot escriure com

$$F(s) = 1 + \frac{1}{s+1} \tag{7}$$

Si provem de fer la descomposició en fraccions simples mitjançant `octave`, obtindrem el següent resultat

```
octave:1> [r,p,k,e]=residue([1 2],[1 1])
r =  1
p = -1
k =  1
e =  1
```

Si ens hi fixem atentament, veurem que aquest és precisament el resultat de la equació 7. En general, `k` és un vector que representa el polinomi resultant de fer la divisió. En general, pels circuits *ben comportats*, el grau del numerador és inferior o, com a molt, igual al denominador. Per tant, com a molt tindrem una constant (que tindrà com a transformada inversa de Laplace una funció delta de Dirac.

## 3 Gràfic de la transformada inversa de Laplace

Si un sistema té una funció de xarxa $H(s)$, la sortida, en el domini transformat de Laplace, vindrà donada per $V_O(s) = H(s)V_I(s)$. Per tornar al domini del temps, caldrà calcular la transformada inversa de $V_O(s)$. Si calculem la transformada inversa de $H(s)$, això equival a calcular la sortida quan $V_I(s) = 1$, és a dir, quan $v_I(t) = \delta(t)$. Per tant, la transformada inversa de $H(s)$ s'anomena resposta impulsional, perquè és la resposta a l'impuls unitari.

Quan vulguem visualitzar una transformada inversa, podem fer servir la funció `impulse`, que calcula la resposta impulsional d'un sistema, és a dir, la resposta quan l'entrada val 1 al domini $s$. La forma d'emprar aquesta funció, a partir d'un numerador i un denominador donats és

```
octave:12> impulse(tf(num,den))
```

on `tf` és necessari per convertir el format "quocient de polinomis" en el format adequat.

Si afegim

```
octave:13> [y,t]=impulse(tf(num,den));
```

tindrem els vectors `t` i `y` disponibles. El ";" al final, fa que el resultat no es mostri a la pantalla. Per exemple, ara podem calcular el màxim de `y`

```
octave:14> max(y)
ans =  0.43562
```

i trobar l'instant de temps en què es produeix:

```
octave:15> t(y==max(y))
ans =  0.85714
```