

APPUNTI del CORSO di TEORIA dei CIRCUITI 2

Circuiti Dinamici del I Ordine - Esempi

docente: Stefano Pastore

November 16, 2004

1 Circuito con Diodo e Condensatore in Parallelo

Il circuito di Fig. 1 è descritto dalla seguente equazione (MNA):

$$\frac{v(t) - v_S(t)}{R} + I_0 \left(e^{\frac{v(t)}{V_T}} - 1 \right) + C \dot{v}(t) = 0 \quad (1)$$

da cui, si ricava la seguente equazione differenziale nonlineare del I ordine:

$$\dot{v}(t) = -\frac{v(t)}{RC} - \frac{I_0}{C} e^{\frac{v(t)}{V_T}} + \frac{I_0}{C} + \frac{v_S(t)}{RC} \quad (2)$$

Poniamo i seguenti valori numerici:

- $I_0 = 10^{-9}$ A, $1/V_T = 40$ V⁻¹
- $R = 1$ kΩ, $C = 1$ mF
- $v_S(t) = 2$ V, $v(0) = 0.3$ V

In Fig. 2(a) è riportato l'andamento di $v_C(t)$ ottenuto con la routine ODE45 di Matlab: si vede che gli errori numerici comportano una oscillazione della soluzione che non dovrebbe esserci. Il codice Matlab è il seguente:

```
FILE "Circuito1.m"  
% ODE per circuito con diodo e condensatore  
[t,y]=ode45('Diodo_cond1',[0 1],[0.3]);  
plot(t,y,'k');  
grid;  
xlabel('\fontsize{14} time \it{t} \rm[s]');  
ylabel('\fontsize{14} voltage \it{v} \rm[V]');
```

```
FILE "Diodo_cond1.m"  
% Circuito con diodo e condensatore  
function dy = Diodo_cond1(t,y)  
RCinv = 1;  
IOC = 1e-6;  
vs = 2;  
dy = [-RCinv*y-IOC*exp(40*y)+IOC+RCinv*vs];
```

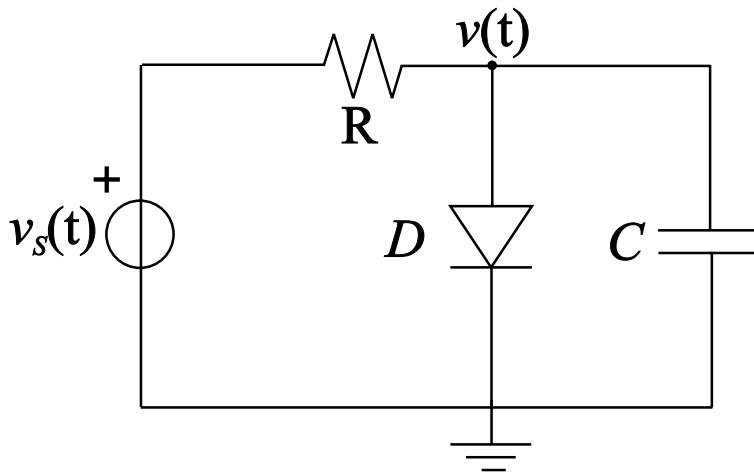


Fig. 1: Circuito con diodo e condensatore.

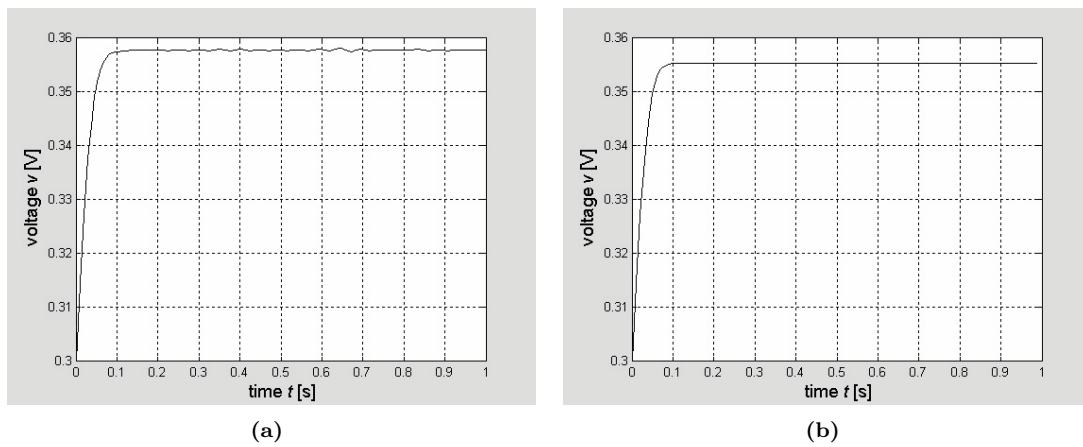


Fig. 2: (a) Integrazione con ODE45. (b) Integrazione PWL.

Se invece approssimiamo il diodo esponenziale con una caratteristica PWL a 20 segmenti:

```

MOD1 PWL (II) (-15.000000,0.00000000E+00)
(0.00000000E+00,0.00000000E+00) (0.24999999E-01,0.17182816E-08)
(0.49999997E-01,0.63890546E-08) (0.74999996E-01,0.19085531E-07)
(0.99999994E-01,0.53598132E-07) (0.12499999,0.14741310E-06)
(0.14999999,0.40242861E-06) (0.17499998,0.10956326E-05)
(0.19999999,0.29799562E-05) (0.22499999,0.81020780E-05)
(0.24999999,0.22025450E-04) (0.27499998,0.59873091E-04)
(0.29999998,0.16275364E-03) (0.32499999,0.44241195E-03)
(0.34999996,0.12026019E-02) (0.37499997,0.32690126E-02)
(0.39999998,0.88860989E-02) (0.42499998,0.24154920E-01)
(0.44999999,0.65659881E-01) (0.44999999,10) END

```

Possiamo integrare le equazioni differenziali lineari ottenute "lavorando" in ogni singola regione lineare attraversata dalla soluzione. In particolare, la soluzione parte dal segmento 14 e arriva al segmento 16 della caratteristica del diodo, dove si trova il punto di equilibrio $v_{CQ} = 0.355348$ V. Si ottiene l'andamento di Fig. 2(b) e si nota che ora non ci sono errori numerici di integrazione.

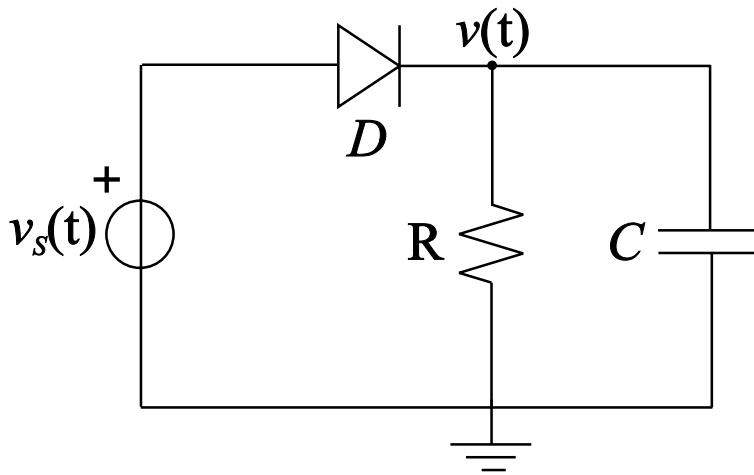


Fig. 3: Circuito con diodo e condensatore.

1.1 Punto di Lavoro con l'Algoritmo di Newton-Raphson

Se poniamo $\dot{v}(t) = 0$ nell'equazione (2), troviamo i punti di equilibrio del circuito. Questa operazione equivale a sostituire il condensatore con un corto circuito. Si ricava l'equazione:

$$f(V_{CQ}) = V_{CQ} + 10^{-6} e^{40 V_{CQ}} - 10^{-6} - 2 = 0 \quad (3)$$

Possiamo usare il metodo di Newton-Raphson per trovare lo zero della funzione $f(\cdot)$. Ricordando che la derivata (jacobiano) è: $f'(V_{CQ}) = 1 + 40 \cdot 10^{-6} e^{40 V_{CQ}}$, si può eseguire il seguente codice scritto in Matlab:

```
FILE "Circuito_equil1.m"
% Example of application of Newton method to find a zero of a circuit
% with a diode in parallel with a capacitor
x0 = 0.4; % make a starting guess at the solution
options = optimset('Display','iter','TolFun',10^(-9)); % option to display output
% Note the option TolFun: function tolerance
[x, fval] = fsolve(@diodo_cond_eq, x0, options); % call optimizer
sprintf('solution: v = %15.10f', x)
%
% the jacobian matrix is given in close form
x0 = 0.4; % make a starting guess at the solution
disp('jacobian on');
options = optimset('Display','iter','Jacobian','on','TolFun',10^(-9)); % option to display output
[xj, fvalj] = fsolve(@diodo_cond_eq, x0, options); % call optimizer
sprintf('solution: v = %15.10f', xj)

FILE "Diodo_cond_eq.m"
% Function for Newton method of circuit for newton_diodoC1.m
function [F,J] = f_diodo_cond_eq(x)
F = x + 10^(-6)*exp(40*x)-10^(-6)-2; % function evaluated at x
if nargin > 1 % two output arguments (J: jacobian)
    J = 1 + 40*10^(-6)*exp(40*x) % Jacobian of the function evaluated at x
end
```

Si trova che, in entrambi i casi, il punto di equilibrio è: $V_{CQ} = 0.3577888688$ V. Differisce con il punto trovato con l'approssimazione lineare a tratti nella terza cifra decimale.

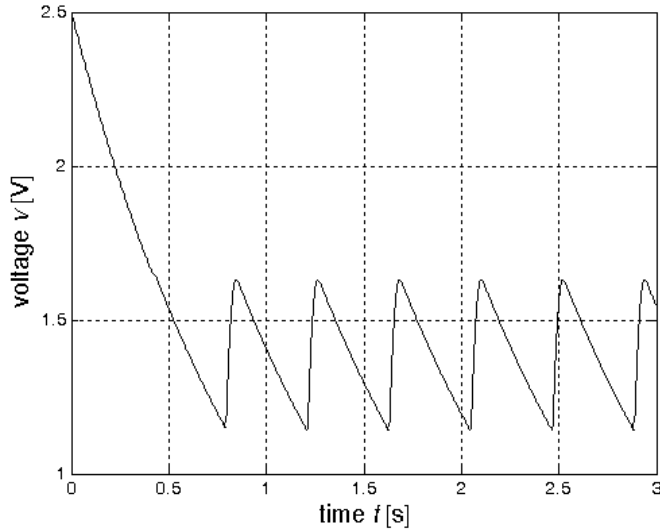


Fig. 4: Integrazione PWL.

2 Circuito con Condensatore e Diodo in Serie

Il circuito di Fig. 3 è descritto dalla seguente equazione (MNA):

$$\frac{v(t)}{R} - I_0 \left(e^{\frac{v_S(t) - v(t)}{V_T}} - 1 \right) + C \dot{v}(t) = 0 \quad (4)$$

da cui, si ricava la seguente equazione differenziale nonlineare del I ordine:

$$\dot{v}(t) = -\frac{v(t)}{RC} + \frac{I_0}{C} e^{\frac{v_S(t) - v(t)}{V_T}} - \frac{I_0}{C} \quad (5)$$

Poniamo i seguenti valori numerici:

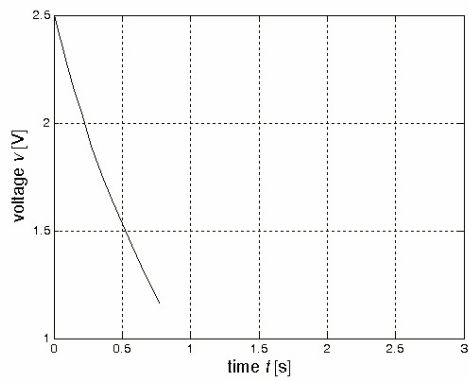
- $I_0 = 10^{-9}$ A, $1/V_T = 40$ V⁻¹
- $R = 1000$ Ω, $C = 1$ mF
- $v_S(t) = 2 \cos(15t)$ V, $v(0) = 2.5$ V

Adottando la approssimazione PWL del diodo vista sopra, otteniamo l'andamento di Fig. 4.

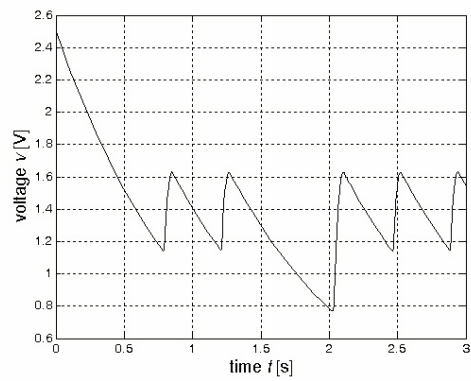
Integrando, invece, con la routine ODE45 di Matlab (Fig. 5(a)), vediamo che gli errori numerici sono tali che non si ottiene la soluzione completa. Infatti, questo è un circuito che lavora su grandi segnali, detto *stiff*: le "costanti di tempo", ottenute linearizzando il circuito per piccoli segnali punto per punto, variano notevolmente durante il funzionamento. Questo fatto genera degli errori di integrazione tali che l'algoritmo numerico non converge. Usando la routine ODE15S, costruita apposta per circuiti stiff, otteniamo l'andamento di Fig. 5(b). Gli errori numerici causano comunque degli errori nella soluzione, anche se l'algoritmo ora converge.

Le routines Matlab sono:

```
FILE "Circuito2.m"
% ODE per circuito con diodo e condensatore
% [t,y]=ode45('Diodo_cond2',[0 3],[2.5]);
[t,y]=ode15s('Diodo_cond2',[0 3],[2.5]);
grid;
xlabel('\fontsize{14} time \it{t} \rm[s]');
```



(a)



(b)

Fig. 5: (a) Integrazione con ODE45. (b) Integrazione con ODE15S.

```
ylabel('\fontsize{14} voltage \it{v} \rm[V]');
```

```
FILE "Diodo_cond2.m"
% Circuito con diodo e condensatore
function dy = Diodo_cond2(t,y)
RCinv = 1;
IOC = 1e-6;
vs = 2*cos(15*t);
dy = [-RCinv*y+IOC*exp(40*(vs-y))-IOC];
```