



**MISSION
INNOVATION**

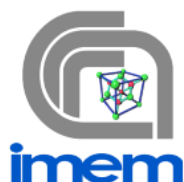
accelerating the clean energy revolution

POA MATERIALI AVANZATI PER L'ENERGIA

**PROGETTO IEMAP - Piattaforma Italiana Accelerata per i Materiali per
l'Energia**

Strumenti di modelling complessivo dei dispositivi ibridi fotovoltaico + accumulo

Stefano Rampino, Edmondo Gilioli, Francesco Pattini



Software- Strumenti di modelling complessivo dei dispositivi ibridi fotovoltaico + accumulo
Stefano Rampino (CNR IMEM), Edmondo Gilioli (CNR IMEM), Francesco Pattini (CNR IMEM)

Maggio 2022

Report MISSION INNOVATION

Ministero della Transizione Ecologica - ENEA

Mission Innovation 2021-2024 - I annualità

Progetto: IEMAP, Piattaforma Italiana Accelerata per i Materiali per l'energia

Work package: WP4 – Materiali per il Fotovoltaico

Linea di attività: LA4.16

Responsabile del Progetto: Massimo Celino, ENEA

Responsabile della LA: Stefano Rampino, CNR

Indice

| | |
|---|----|
| SOMMARIO | 5 |
| 1 INTRODUZIONE | 6 |
| 2 DESCRIZIONE DELLE ATTIVITÀ SVOLTE E RISULTATI | 6 |
| 2.1 F_GAMMA_2.M | 6 |
| 2.2 E1_CARICA_CONDENSATORE.M | 9 |
| 2.3 E2_RLC_NONLINEARE.M | 10 |
| 2.4 E4_DCDC.M | 11 |

Sommario

In questa attività sono stati utilizzati strumenti per la modellizzazione top-down del funzionamento di un potenziale dispositivo ibrido ottenuto integrando una cella fotovoltaica ed un accumulatore a in modo che abbiano terminali elettrici in comune.

In questo documento sono presentati i codici Matlab che a partire da un modello circuitale e da una configurazione specifica del convertitore sia in grado di determinare con notevole precisione le performance del sistema.

1 Introduzione

La simulazione circuitale dei sistemi fotovoltaici per la valutazione del corretto funzionamento dei sistemi di conversione/accumulo richiede una serie di accorgimenti e lo sviluppo di algoritmi specifici.

Di seguito presenteremo brevemente il software prodotto tenendo conto degli elementi chiave che sono:

- Modello circuitale *one-diode* per la descrizione della cella/modulo fotovoltaico (con dipendenza dei parametri dalla temperatura e dall'irradianza)
- Sistema alle equazioni di stato per la descrizione di un tipico sistema di conversione
- Simulazione di un convertitore *DC-DC*

2 Descrizione delle attività svolte e risultati

Il principale risultato di questa linea di attività è stato lo sviluppo della metodologia di modellizzazione "top-down" dei dispositivi ibridi fotovoltaico più conversione/accumulo.

Il software è composto da codici per Matlab che permette di determinare, per approssimazioni successive, i parametri di input ottimali in funzione dei target di prestazione del dispositivo ibrido

I codici che compongono il software deliverables sono i seguenti:

- *f_gamma_2.m*: calcola i cinque parametri a partire dai dati del datasheet fornito dai produttori delle celle/pannelli solari
- *E1_Carica_condensatore.m*: per simulare con Eulero la carica del condensatore da parte di un modulo fotovoltaico
- *E2_RLC_NonLineare.m*: per risolvere il problema di un circuito RLC non lineare
- *E4_DCDC.m*: simulatore per il modulo fotovoltaico connesso ad DC-DC

2.1 *F_gamma_2.m*

```
function [ gamma ] = f_gamma_2( n, phi )
```

```
% #####  
% ##### INITIALIZE #####  
% #####
```

```
Voc = phi(1);  
Isc = phi(2);  
Vmp = phi(3);  
Imp = phi(4);  
Ns = phi(5);
```

```
T = 273.15 + 25.0;  
k = 1.3806503E-23;  
q = 1.602E-19;  
Vt = k*T/q;
```

```
P1 = Vmp .* Imp;  
P2 = (Voc - Vmp) .* Imp;  
P3 = (Voc - Vmp) .* (Isc - Imp);  
P4 = Vmp .* (Isc - Imp);
```

```

ExpSC = @(Rs)    exp((Isc.*Rs)./(Ns.*n.*Vt));
ExpOC = @(Rs)    exp(Voc./(Ns*n*Vt));
ExpMP = @(Rs)    exp((Vmp+Imp*Rs)./(Ns.*n.*Vt));

A1 = @(Rs)    Vmp + Rs .* Imp - Voc;
A2 = @(Rs)    Voc - Rs .* Isc;
A3 = @(Rs)    Rs .* Isc - Rs .* Imp - Vmp;

% #####
% #####      COMPUTE RS      #####
% #####

f1 = @(Rs) (((P2 - P1)*ExpSC(Rs)) + ...
    (P1 - P4)*ExpOC(Rs) + ...
    ((P1 - P3) * (Imp * Rs - Vmp))/(n* Ns * Vt) + ...
    (P4- P2)) * ExpMP(Rs)) / ExpOC(Rs);

% f2 = @(Rs) (P2 - P1)*ExpSC(Rs) + (P1 - P4)*ExpOC(Rs) + ...
%   ((P1 - P3) * (Imp * Rs - Vmp))/(n* Ns * Vt) + (P4- P2)) * ExpMP(Rs);

options = optimoptions('fsolve','Display','off');

[Rs, ~, exitflag] = fsolve(f1,0.1, options);

if exitflag < 0
    Rs = NaN;
end

% #####
% #####      NaN NON-PHYSICAL RS      #####
% #####

RsMAX = Vmp/Imp + (Ns*n*Vt/Imp)*(1+lambertw(-1,-exp( (Voc-n*Ns*Vt-2*Vmp))/(Ns*n*Vt))));
if Rs > RsMAX | Rs < 0
    Rs = NaN;
end

% #####
% #####      CALCULATE MODEL      #####
% #####

Gsh = (ExpOC(Rs).*(Imp-Isc)+ExpMP(Rs).*(Isc)-
ExpSC(Rs).*Imp)./(A1(Rs).*ExpSC(Rs)+A2(Rs).*ExpMP(Rs)+A3(Rs).*ExpOC(Rs));
Rsh = 1 ./ Gsh;
IO = (Voc.*(Isc-Imp) - Vmp.*Isc)./(A1(Rs).*ExpSC(Rs)+A2(Rs).*ExpMP(Rs)+A3(Rs).*ExpOC(Rs));
Iirr = (Isc.*Voc.*(ExpMP(Rs) -1) + Isc.*Vmp.*(1-ExpOC(Rs) ) + Imp.*Voc.*(1-ExpSC(Rs) )) ./
(A1(Rs).*ExpSC(Rs)+A2(Rs).*ExpMP(Rs)+A3(Rs).*ExpOC(Rs));

```

```
gamma(1) = Rs;
gamma(2) = Rsh;
gamma(3) = Iirr;
gamma(4) = I0;
gamma(5) = n;
```

```
end
```

Per simulare il comportamento del modulo fotovoltaico al variare delle condizioni ambientali è importante riferirsi agli studi sulla dipendenza dei cinque parametri dalla temperatura e irradianza. C'è un'ampia bibliografia a riguardo; di seguito ci limiteremo ad indicare i modelli di riferimento che sono stati implementati, fatta eccezione per l'ideality factor n che verrà considerato costante.

La corrente di saturazione inversa I_0 è parte della corrente inversa di un diodo a semiconduttore ed è causata dalla diffusione delle cariche minoritarie dalle zone neutre alle zone di svuotamento. Tale corrente risulta quasi del tutto indipendente dalla tensione di saturazione del diodo. Per un diodo p-n ideale è data dalla seguente formula:

$$I_0 = qAn_i^2 \left(\frac{1}{N_D} \sqrt{\frac{D_P}{t_P}} + \frac{1}{N_A} \sqrt{\frac{D_N}{t_n}} \right)$$

Dove q è la carica elementare, A è la sezione trasversale dell'area, D_P e D_N sono i coefficienti di diffusione di lacune ed elettroni, N_D e N_A sono le concentrazioni di donori e accettori nelle zone n e p delle giunzioni, n_i è la concentrazione intrinseca dei portatori nel materiale semiconduttore e t_p e t_n sono rispettivamente i coefficienti di vita media di lacune ed elettroni.

La variazione di I_0 con la temperatura T può essere studiata con la formula che segue:

$$I_0 = A_0 T^3 e^{\frac{-E_g}{nKt}}$$

Dove A_0 è una costante che dipende dal materiale del semiconduttore che nel caso del silicio vale $2.22e3$, e il termine E_g indica l'energy gap del silicio che può essere studiato mediante la formula di Varshni (eq. 6), e rappresentato nel grafico in figura 4 per il caso del silicio:

$$E_g = E_g(0K) - \frac{\alpha T^2}{T + \beta}$$

La dipendenza della resistenza serie è descritta nell'equazione:

$$R_s(T) = R_{s0} + A_{0Rs} e^{\frac{T-273}{B_0}}$$

La dipendenza della resistenza Shunt con temperatura ed irradianza è stata molto discussa e studiata. Il modello che è ritenuto più efficiente è quello dovuto a De Soto:

$$R_{sh} = \frac{G_{Ref}}{G} \cdot R_{sh,ref}$$

La Photocurrent I_{ph} è la corrente prodotta dall'esposizione di un fotodiodo ad una radiazione luminosa. La corrente di fotodiodo è dunque generata dagli effetti fotoelettrico o fotovoltaico. Risulta uno dei parametri più importanti per una cella fotovoltaica.

Se ci poniamo nel caso di luce monocromatica allora la corrente I_{ph} può essere calcolata come:

$$I_{ph} = A \cdot G \cdot SR(\lambda_0) \quad (10)$$

Dove G è l'irradianza normale alla cella e A l'area della superficie del dispositivo. Nel modello a cinque parametri è stato osservato come la I_{ph} dipende dall'irradianza effettiva G_{eff} , dalla temperatura della cella T_c , dal coefficiente di temperatura della corrente di cortocircuito α_{Isc} e dal modificatore di massa d'aria M_{am} . Esiste anche un'equazione che rappresenta la I_{ph} a qualsiasi condizione operativa, facendo però uso dell'irradianza G e dell'irradianza e temperatura in condizioni standard:

$$I_{ph} = \frac{G}{G_{ref}} [I_{ph,ref} + \alpha_T (T - T_{ref})]$$

2.2 E1_Carica_condensatore.m

```
clear; clc;
```

```
C = 1e-3;  
Rs = .1;  
Rsh = 10;  
n = 2;  
I0 = 1e-6;  
Iirr = 1;  
Vt = 26e-3;
```

```
ipv = @(vpv) - (n*Vt/Rs) * lambertw( (Rs*I0*Rsh)/((n*Vt)*(Rsh+Rs)) * ...  
    exp(Rsh/(Rs+Rsh)*(vpv+Rs*(Iirr+I0))/(n*Vt))) + ((Iirr + I0)*Rsh - vpv)/(Rsh + Rs);
```

```
dx1 = @(x1) 1/C*ipv(x1);
```

```
% dx1 = @(x1) 1/C*(0.71455-x1)/Rs;
```

```
x1_vect(1) = 0;
```

```
t = linspace(0,.003,100);  
dt = max(t)/length(t);
```

```
for ct_t = 1:length(t)-1
```

```
    x1_vect(ct_t+1) = x1_vect(ct_t) + dt*dx1(x1_vect(ct_t));
```

end

plot(t,x1_vect)
hold on

in cui: C è la capacità del condensatore accoppiato al modulo fotovoltaico; Rs è la resistenza serie del modulo; Rsh è la resistenza di shunt del modulo; n indica il fattore d'idealità; I0 è la corrente di buio, Iirr è la corrente fotogenerata dal modulo e Vt è un valore costante che descrive il voltaggio termico.

2.3 *E2_RLC_Nonlineare.m*

lear; clc;

C = 0.2;

L0 = 0.1;

Li = 0.5;

I0 = .5;

L = @(i) L0+Li*(exp(-i.^2./I0));

R = .1;

E = 3;

dx1 = @(x1,x2) 1/C*(x2);

dx2 = @(x1,x2) 1/L(x2)*(-R*x2+E-x1);

t = linspace(0,3,200);

dt = max(t)/length(t);

x1_vect(1) = 1;

x2_vect(1) = -2;

for ct_t = 1:length(t)-1

 x1_vect(ct_t+1) = x1_vect(ct_t) + dt*dx1(x1_vect(ct_t),x2_vect(ct_t));

 x2_vect(ct_t+1) = x2_vect(ct_t) + dt*dx2(x1_vect(ct_t),x2_vect(ct_t));

end

figure

comet(x1_vect,x2_vect)

grid on

```
grid minor
xlabel('Vc')
ylabel('iL')
```

```
hold on;
```

```
%%
```

```
while(1)
```

```
    [x1_vect(1), x2_vect(1)] = ginput(1);
```

```
for ct_t = 1:length(t)-1
```

```
    x1_vect(ct_t+1) = x1_vect(ct_t) + dt*dx1(x1_vect(ct_t),x2_vect(ct_t));
```

```
    x2_vect(ct_t+1) = x2_vect(ct_t) + dt*dx2(x1_vect(ct_t),x2_vect(ct_t));
```

```
end
```

```
    comet(x1_vect,x2_vect)
```

```
end
```

Per simulare un dispositivo fotovoltaico inserito in un circuito di conversione dell'energia, purtroppo non basta tracciare la curva I-V, in quanto la presenza di elementi a memoria richiede la risoluzione di un problema differenziale. Il metodo più semplice (e più efficace) di integrazione numerica di una equazione differenziale ordinaria del primo ordine è il metodo di Eulero descritta in dettaglio al paragrafo 2.3 del file D4.38.

2.4 E4_DCDC.m

```
clear; clc;
```

```
% Parametri DCDC
```

```
C = 33e-6;
```

```
L = 240e-6;
```

```
Cin = 33e-6;
```

```
R = 1;
```

```
Vforw = 0.5;
```

```
Rd = 0.05;
```

```
Rds = 0.01;
```

```
% Parametri PV
```

```
Ns = 32;
```

```
n_guess = 0.903; % Valore di n_guess massimo, risposta migliore su Voc@NOCT
alpha = 0.05/100; % alpha [%/K]
dsheet = [24.5 6.0 18.2 5.7 Ns];
```

```
gamma_ref = f_gamma_2(n_guess, dsheet);
```

```
kJ = 1.380648521e-23;
q = 1.60217662e-19;
Vt = @(T) kJ*(T)/q;
kEv = 8.61673324e-5;
Vt = 26e-3;
```

```
Eg = @(T) 1.66 - 4.73e-4 * T.^2/(T+636);
```

```
Duty = 0.2;
pwm_period = 1e-5;
```

```
samps_per_period = 30;
n_periods = 200;
```

```
t = linspace(0, n_periods*pwm_period, n_periods*samps_per_period);
pwm_wave = mod(t,pwm_period)/pwm_period < Duty;
```

```
% Parametri GT
g_vect = 1000*ones(size(t));
t_vect = 298.15*ones(size(t));
```

```
g_vect(end/3:end) = 1300;
t_vect(end/3:end) = 330;
```

```
g_vect(2*end/3:end) = 400;
t_vect(2*end/3:end) = 290;
```

```
x1_vect(1) = 0;
x2_vect(1) = 0;
x3_vect(1) = 0;
```

```
dt = t(end)/length(t);
```

```
for ct_time = 1:length(t)-1
    clc;
    disp(ct_time);
    disp(length(t));
```

```
% Attualizzo il modello a 1diode e preparo la funzione ipv(vpv)
```

```
G = g_vect(ct_time);
```

```
T = t_vect(ct_time);
```

```
gammaGT = @(G, T, th) [th(1)
    (1000/G)*th(2)
    th(3)*(G/1000)*(1+alpha*(T-298.15))
    th(4)*(T/298.15).^3*exp(Eg(298.15)/(kEv*298.15) - Eg(T)/(kEv*T))
    th(5)];
```

```
gamma = gammaGT(G,T,gamma_ref);
```

```
Rs = gamma(1);
Rsh = gamma(2);
lirr = gamma(3);
I0 = gamma(4);
n = Ns*gamma(5);
```

```
ipv = @(vpv) - (n*Vt/Rs) * lambertw( (Rs*I0*Rsh)/((n*Vt)*(Rsh+Rs)) * ...
    exp(Rsh/(Rs+Rsh)*(vpv+Rs*(lirr+I0))/(n*Vt))) + ((lirr + I0)*Rsh - vpv)/(Rsh + Rs);
```

```
% Equazioni
```

```
dx1_on = @(x1,x2,x3) 1/C*(-x1/R+x2);
dx2_on = @(x1,x2,x3) 1/L*(-x1+x3-Rds*x2);
dx3_on = @(x1,x2,x3) 1/Cin*(ipv(x3)-x2);
```

```
dx1_off = @(x1,x2,x3) 1/C*(-x1/R+x2);
dx2_off = @(x1,x2,x3) 1/L*(-Vforw-x1-Rd*x2);
dx3_off = @(x1,x2,x3) 1/Cin*(ipv(x3));
```

```
if pwm_wave(ct_time)
```

```
    x1_vect(ct_time+1) = x1_vect(ct_time) + dt*dx1_on(x1_vect(ct_time),x2_vect(ct_time),
x3_vect(ct_time));
    x2_vect(ct_time+1) = x2_vect(ct_time) + dt*dx2_on(x1_vect(ct_time),x2_vect(ct_time),
x3_vect(ct_time));
    x3_vect(ct_time+1) = x3_vect(ct_time) + dt*dx3_on(x1_vect(ct_time),x2_vect(ct_time),
x3_vect(ct_time));
```

```
else
```

```
    x1_vect(ct_time+1) = x1_vect(ct_time) + dt*dx1_off(x1_vect(ct_time),x2_vect(ct_time),
x3_vect(ct_time));
    x2_vect(ct_time+1) = x2_vect(ct_time) + dt*dx2_off(x1_vect(ct_time),x2_vect(ct_time),
x3_vect(ct_time));
    x3_vect(ct_time+1) = x3_vect(ct_time) + dt*dx3_off(x1_vect(ct_time),x2_vect(ct_time),
x3_vect(ct_time));
```

```
    x2_vect(ct_time+1) = max(0, x2_vect(ct_time+1));
```

```
end
```

end

```
subplot(3,1,1)
plot(t,x1_vect);
grid on
grid minor
xlabel('time [s]');
ylabel('v(C) [V]');
hold on
```

```
subplot(3,1,2)
plot(t,x2_vect);
grid on
grid minor
xlabel('time [s]');
ylabel('i(L) [V]');
hold on
```

```
subplot(3,1,3)
plot(t,x3_vect);
grid on
grid minor
xlabel('time [s]');
ylabel('v(C_{IN}) [V]');
hold on
```

Come evidente all'inizio del listato in esame, dimensionando in maniera plausibile i componenti del DC-DC e il pannello PV, si ottiene l'andamento previsto. I parametri ottimizzati per la simulazione sono i seguenti:

```
% Parametri DCDC
C = 33e-6;
L = 240e-6;
Cin = 33e-6;
R = 1;
Vforw = 0.5;
Rd = 0.05;
Rds = 0.01;
dsheet = [24.5 6.0 18.2 5.7 Ns];
Duty = 0.8;
pwm_period = 1e-5;
```

I risultati della simulazione permettono di definire il tempo di assestamento dei parametri del sistema come tensione e corrente del supercapacitore, tensione del pannello solare nell'ordine di alcuni secondi. A partire da queste condizioni al contorno è possibile anche considerare una variazione delle condizioni ambientali. Per esempio, possiamo vedere come reagisce un DC-DC ad un transiente molto rapido di irradianza (vedi i dettagli nel report D4.38).

In definitiva, con questo strumento abbiamo la possibilità di monitorare accuratamente i transistori, e scoprire, per esempio, se siamo a rischio di sovracorrenti/sovratensioni che potrebbero danneggiare i dispositivi di switch.