POLITECNICO DI BARI

SISTEMI PROGRAMMABILI

L.M. INGEGNERIA INFORMATICA

A.A. 2010/11

"HOW TO ACQUIRE THE DIODE'S VALUES TO DETERMINE VOLTAGE-CURRENT CHARACTERISTIC CURVE"

SALATINO ANGELO ANTONIO a.salatino@studenti.poliba.it

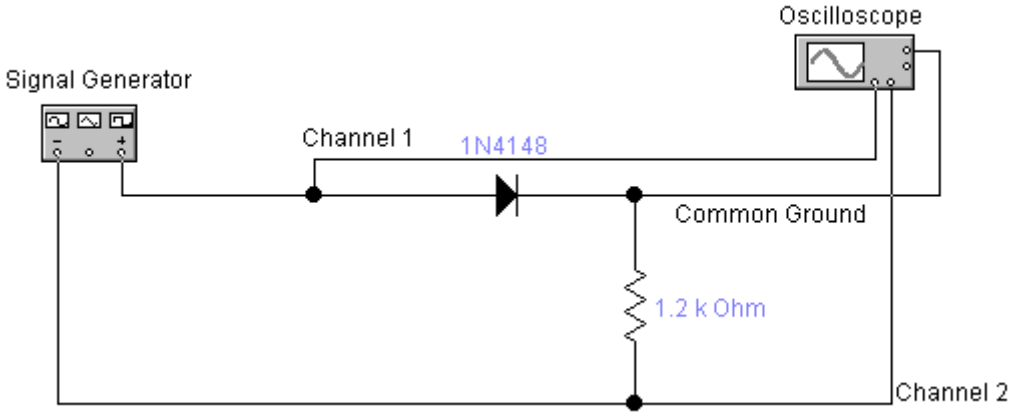
DI LEO CARLO ing.carlo88@gmail.com

Obiettivi

Questo documento ha come compito principale, quello di mostrare come in ambiente MATLAB sia possibile pilotare un generatore di segnali, acquisire dati dall'oscilloscopio ed elaborali successivamente attraverso una qualsiasi operazione di analisi numerica. Verrà mostrato e commentato uno script contente comandi Matlab, che consente l'interfacciamento con gli strumenti dai quali, nello specifico, acquisisce i valori di tensione ai capi di un diodo per poi elaborare la curva caratteristica di tensione e corrente.

Apparato Sperimentale

Strumenti e accessori: Oscilloscopio digitale HP54603, Generatore di funzioni HP3312, Computer (su cui è installato il software MATLAB, compreso di driver per gli strumenti), basetta bread-board, diodi IN4148 e IN4007, resistore da 1200Ω.



Essendo l'oscilloscopio a massa comune, quest'ultima deve necessariamente essere inserita nell'unico punto in comune tra le due misure da effettuare. Il canale 1 preleverà la tensione ai capi del diodo, mentre il canale 2 preleverà la tensione ai capi del resistore, dal quale otterremo il valore della corrente (legge di Ohm) con segno negativo (legge di Kirchhoff).

Codice

Il codice è composto dalle seguenti componenti:

Istanziazione degli oggetti

Apertura degli oggetti e settaggi vari

Reset degli strumenti

Creazione vettore frequenza

Acquisizione dati

Salvataggio dati

Plotting dei dati

Chiusura e cancellazione oggetti

Istanziazione degli oggetti

Gli strumenti vengono visti dall'elaboratore come degli oggetti, nonché enormi contenitori di variabili che permettono allo strumento stesso la possibilità di memorizzare dati. Le variabili vengono definite anche attributi.

```
osc=gpib('ni',0,1); % oggetto oscilloscopio
gen=gpib('ni',0,2); % oggetto gen. segnale
```

Apertura degli oggetti e settaggi vari

Dopo aver istanziato gli oggetti, con un opportuno metodo si deve aprire la comunicazione con gli strumenti. Tuttavia è necessario che sull'oggetto dell'oscilloscopio il valore dell'attributo InputBufferSize sia maggiore o uguale al numero di punti che si vuole richiedere (default 512 pt).

```
set(osc,'InputBufferSize',5000)
fopen(osc);
fopen(gen);
instrfind
```

Il commando `instrfind` consente di sapere quali oggetti sono tutt'ora in memoria e qual è il loro stato. Successivamente per entrambi gli oggetti si devono impostare i valori End Of String cosicché Matlab possa interpretare i caratteri terminatori di riga per controllare e distinguere il flusso da e verso lo strumento.

```
set(gen,'EOSMode','read&write')
set(gen,'EOSCharCode','LF') % Line Feed -> ASCII = 10
set(osc,'EOSMode','read&write')
set(osc,'EOSCharCode','LF')
```

Reset degli strumenti

```
fprintf(osc,'*rst; *cls')
fprintf(gen,'*rst; *cls')
```

Creazione vettore frequenza

Con le seguenti istruzioni cerchiamo di ottenere un vettore di frequenze con le quali poi pilotare il generatore di segnali e acquisire i dati. I valori del vettore non hanno un andamento lineare bensì esponenziale in modo tale da avere un range più ampio di basse frequenze e un range meno ampio di alte frequenze. Successivamente sono state eliminate alcune componenti perché risultano essere a bassissima frequenza (0 – 96Hz) e inutili ai fini dell'esercitazione. E' possibile, tuttavia, implementare un qualsiasi metodo per ottenere un proprio vettore delle frequenze.

```
x=[1:100];
freq=1.1482.^x;
freq=freq(33:100);
freq=uint64(freq);
```

Come valore di base è stato scelto 1.1482 perché elevato a 100 fornisce 1MHz.

Acquisizione dati

L'acquisizione dei dati si compone fondamentalmente con il riempimento due matrici (Vdiodo e Idiodo) aventi stessa dimensione con numero di righe pari al numero di valori acquisiti (2000) e colonne pari al numero di acquisizioni (68). Le due matrici hanno rispettivamente i valori di tensione e corrente al variare della frequenza.

Prima di procedere con l'acquisizione dei dati si devono impostare il valore picco-picco del segnale, il suo offset e il numero di punto da richiedere all'oscilloscopio.

```
Vpp_gen=15;%Ampl
c_gen=0;%offset
N=2000; %numero di punti da acquisire
```

L'acquisizione sarà composta fondamentalmente da due cicli `for` dove nel primo si acquisirà il valore di tensione sul diodo, nel secondo si acquisirà il valore di tensione sul resistore diviso per lo stesso e cambiato di segno.

Per una visione più decorosa del codice nei cicli for, il dettaglio di ogni comando avverrà per mezzo di commenti (`% commento`)

```
for k=1:68
    f_gen=freq(k); % memorizzazione del k-esimo valore di frequenza in f_gen
    s=sprintf('apply:sin %f, %f, %f', f_gen, Vpp_gen, c_gen) %creaz. Commandline
    fprintf(gen,s); % scrittura del comando sull'oggetto gen di segnali

%L'onda è stata generata, con le sonde opportunamente connesse si dovrebbero
%notare i risultato sull'oscilloscopio e quaindi si è pronti per l'acquisizione.

% impostazioni forma d'onda
fprintf(osc,':autoscale'); % autoscale dell'oscilloscopio
fprintf(osc,':display:connect off'); %no interpolation
fprintf(osc,':waveform:points %d',N); %numero di punti da chiedere
fprintf(osc,':aquire:complete 100'); % percentuale di punti totali da acquisire

fprintf(osc,':waveform:source channel1'); % impostazione del canale da dove
%prendere I punti
fprintf(osc,':digitize channel1');
% acquisizione di forma d'onda sul canale 1
fprintf(osc,':waveform:data?'); %comando per la richiesta dati
samples = fread(osc, N+11,'uchar'); %lettura dati
samples = samples(11:end-1);
% acquisizione dei valori di tensione e di tempo canale 2
%y=(samples?code_ref )*Q + y _ orig
%tensione
Q = query(osc, ':wav:yinc?','%s','%f'); % passo di quantizzazione
y_orig = query(osc, ':wav:yor?','%s','%f'); % valore di tensione nell'origine
code_ref = query(osc, ':wav:yref?','%s','%d');%valore del codice corrispondente
%al centro dello schermo
Vdiodo(:,k) = (samples - code_ref)*Q + y_orig;
end
```

Qualche chiarimento utile a supporto del semplice commento risulta doveroso quando invochiamo il comando:

```
samples = samples(11:end-1);
```

perché negli N+11 punti richiesti abbiamo che i primi 10 formano un header e l'ultimo è il LF impostato con il comando:

```
set(gen,'EOSCharCode','LF');
```

in definitiva si eliminiamo gli 11 valori riottenendo i dati originalmente chiesti `samples <4000x1>.`

Il comando:

```
Vdiodo(:,k) = (samples - code_ref)*Q + y_orig;
```

serve ad adattare i valori ottenuti (samples), compresi tra 0 e 255, ai valori originali di tensione relazionati al passo di quantizzazione e all'offset.

Nel secondo ciclo for ripetiamo la stessa procedura con la differenza che il canale sorgente è il canale 2 e alla fine di ogni ciclo il vettore delle tensioni misurato viene diviso per -1200 (-1*1200) ottenendo i valori di corrente rispettando le leggi di Ohm e di Kirchhoff

```
for k=1:68
    % molto simile a quanto già fatto per la prima acquisizione
    f_gen=freq(k);
    s=sprintf('apply:sin %f, %f, %f', f_gen, Vpp_gen, c_gen)
    fprintf(gen,s);

% impostazioni forma d'onda
fprintf(osc,':autoscale');
fprintf(osc,':display:connect off'); %no interpolation
fprintf(osc,':waveform:points %d',N);
fprintf(osc,':aquire:complete 100');

% ch2
fprintf(osc,':waveform:source channel2'); %la sorgente è il canale 2
fprintf(osc,':digitize channel2');
% acquisizione di forma d'onda sul canale 2
fprintf(osc,':waveform:data?');
samples = fread(osc, N+11,'uchar');
samples = samples(11:end-1);
% acquisizione dei valori di tensione e di tempo canale 2
%tensione
Q = query(osc, ':wav:yinc?','%s','%f'); % passo di quantizzazione
y_orig = query(osc, ':wav:yor?','%s','%f'); % valore di tensione nell'origine
code_ref = query(osc, ':wav:yref?','%s','%d');%valore del codice corrispondente
al centro dello schermo
disp('y=(samples?code_ref )*Q + y _ orig');
Idiodo(:,k) = (samples - code_ref)*Q + y_orig; %adattamento del valore
Idiodo(:,k) = Idiodo(:,k)/(-1200); %1200 è il valore del resistore
end
```

Salvataggio dati

Viene predisposto un comando che effettua il salvataggio delle matrici della tensione e della corrente e del vettore frequenza per eventuali elaborazioni future.

```
save dati.mat Vdiodo Idiodo freq
```

Il suo complementare è:

```
load dati.mat
```

Plotting dei dati

Con i successivi comandi si effettua un plot completo:

- Tensione
- Corrente
- Caratteristica tensione corrente

Si deve impostare il valore dell'indice (`index`)ricordando che in questo caso è un valore compreso tra 1 e 68 (numero max di misure effettuate) che a sua volta indica quale valore di frequenza prelevare e successivamente plottare.

```
index=1; % 1<=index<=68
subplot(2,2,1), plot(Vdiodo(:,index));
grid
title(strcat('Frequenza = ', num2str(freq(index)), ' Hz'));
xlabel('Time [s]');
ylabel('Volt [V]');
subplot(2,2,2), plot(Idiodo(:,index));
grid
xlabel('Time [s]');
ylabel('Current [I]');
subplot(2,2,3), plot(Vdiodo(:,index),Idiodo(:,index));
grid
xlabel('Volt [V]');
ylabel('Current [I]');
```

Chiusura e cancellazione oggetti

Qui di seguito si trovano i comandi per chiudere le comunicazioni con gli strumenti e di conseguenza eliminarli dal workspace.

```
fclose(osc);   %chiusura oggetto oscilloscopio
fclose(gen);

delete(osc); %cancellazione oggetto oscilloscopio
delete(gen);
delete(instrfind);
clear %cancellazione di tutto il workspace
```

Accorgimenti

Per evitare errori di calcolo si consiglia di controllare la sonda in uso. Nel caso in cui si utilizzino sonde con un fattore di scala a 10x si deve riportare la scala a 1x oppure moltiplicare tutti i valori Vdiodo e Idiodo per 10.

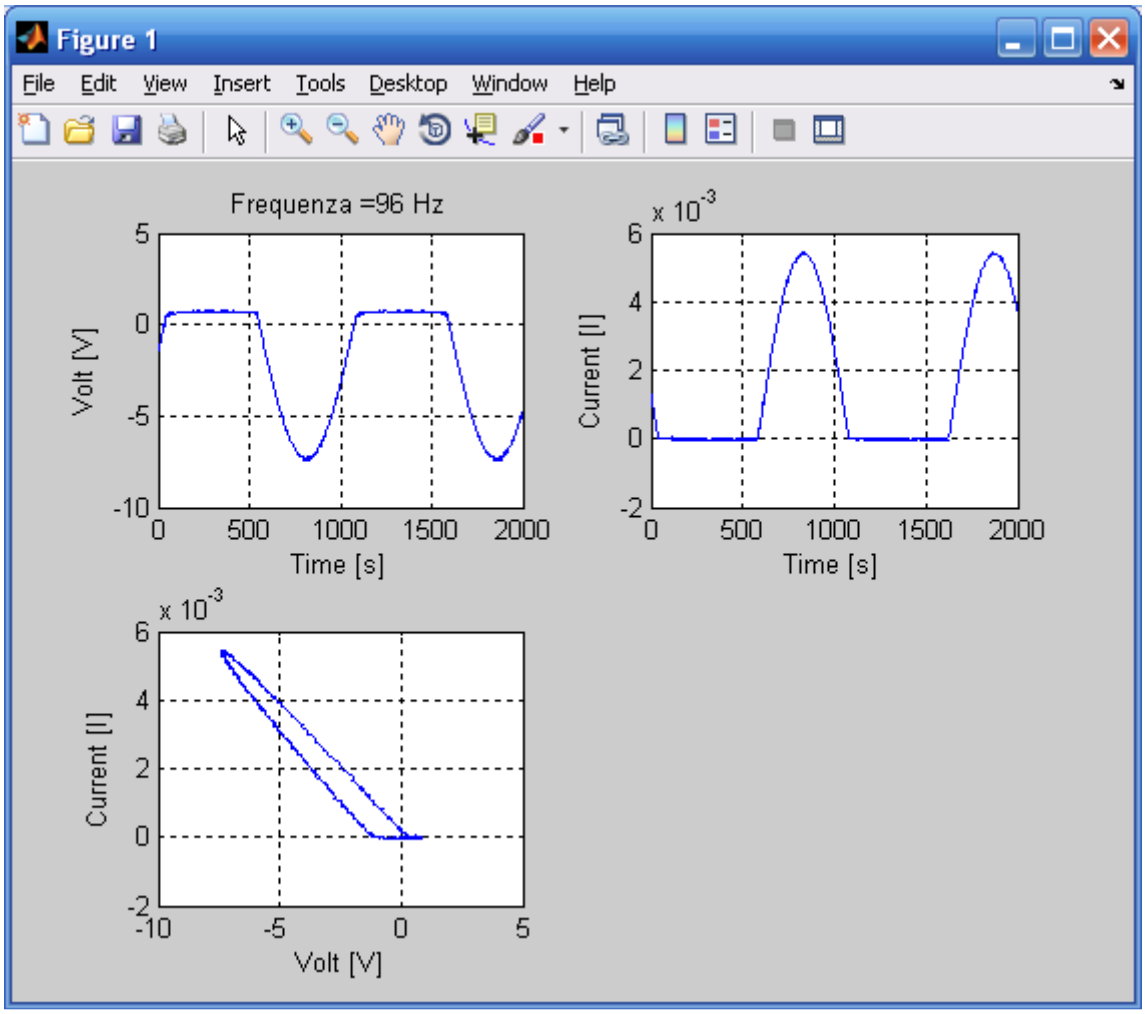
Per un risparmio di tempo si poteva includere la seconda acquisizione (Idiodo) in coda alla prima (Vdiodo) eseguendo cosi un unico ciclo for, ma in fase di autoscale il segnale del canale 2 a determinate frequenze finiva a ridosso del margine basso facendo in modo che tra i valori acquisiti ci fosse anche il 10. E' bene ricordare che siccome dall'oscilloscopio si acquisiscono stringhe di `'uchar'` dove ogni uchar (unsigned char) può assumere valore tra 0 e 255, il valore 10 in ASCII è considerato come il LF (Line Feed). Acquisendo il 10 come valore all'interno della stringa e poiché LF determina il punto in cui termina la

stringa essi risulterebbero di dimensione inferiore ad N (2000), ottenendo cosi un errore da parte di Matlab nella fase di assegnazione. Il problema potrebbe essere aggirato in tanti modi, ma il più comodo al fine dell'esercitazione è stato quello di adoperare due cicli for distinti.

Alcuni risultati

Per il diodo IN4007 si mostreranno tre risultati acquisiti:

Index = 1;



Index=15;

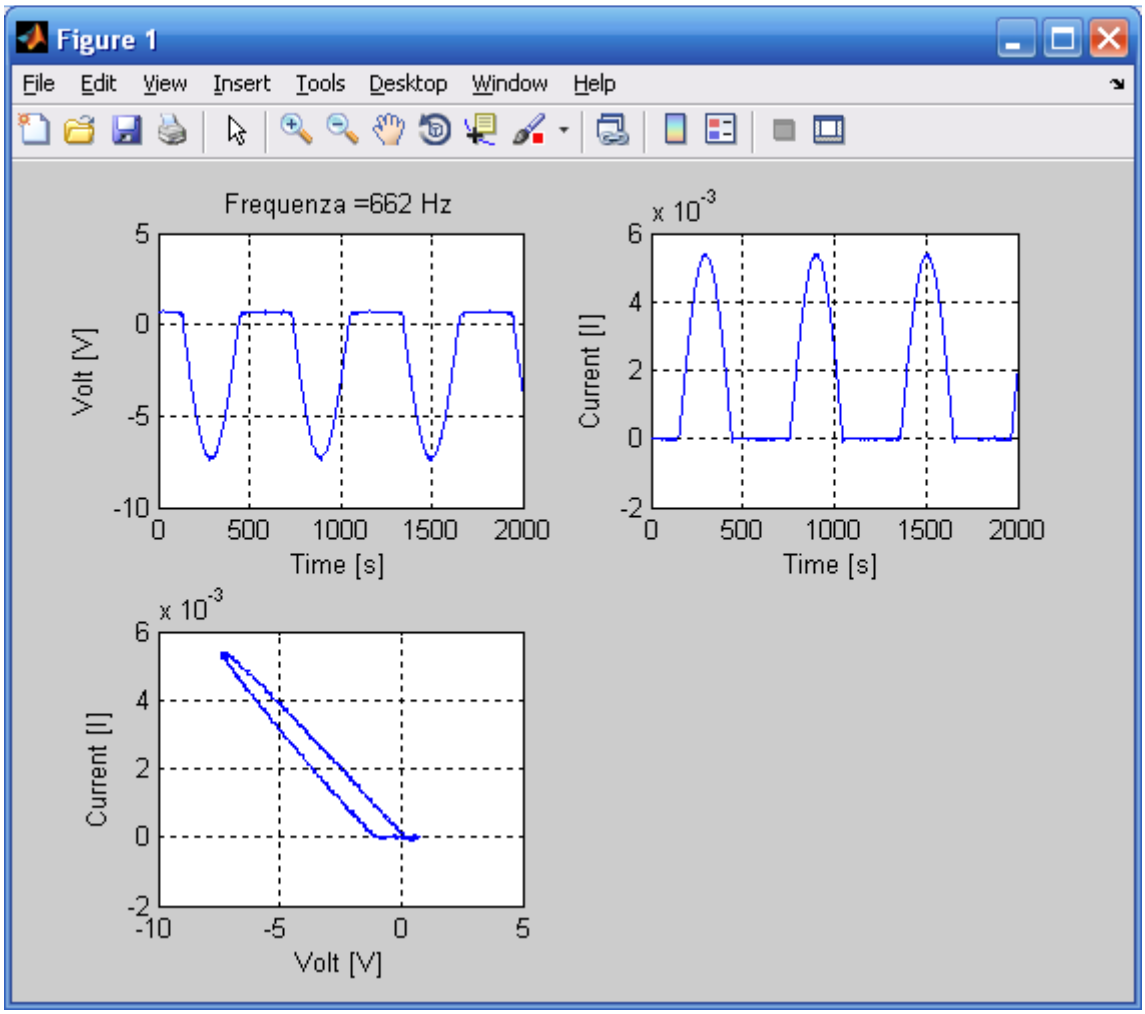


Figure 1
File Edit View Insert Tools Desktop Window Help
Frequenza =662 Hz
Volt [V]
Time [s]
x 10^-3
Current [I]
Time [s]
x 10^-3
Current [I]
Volt [V]

Index = 68;

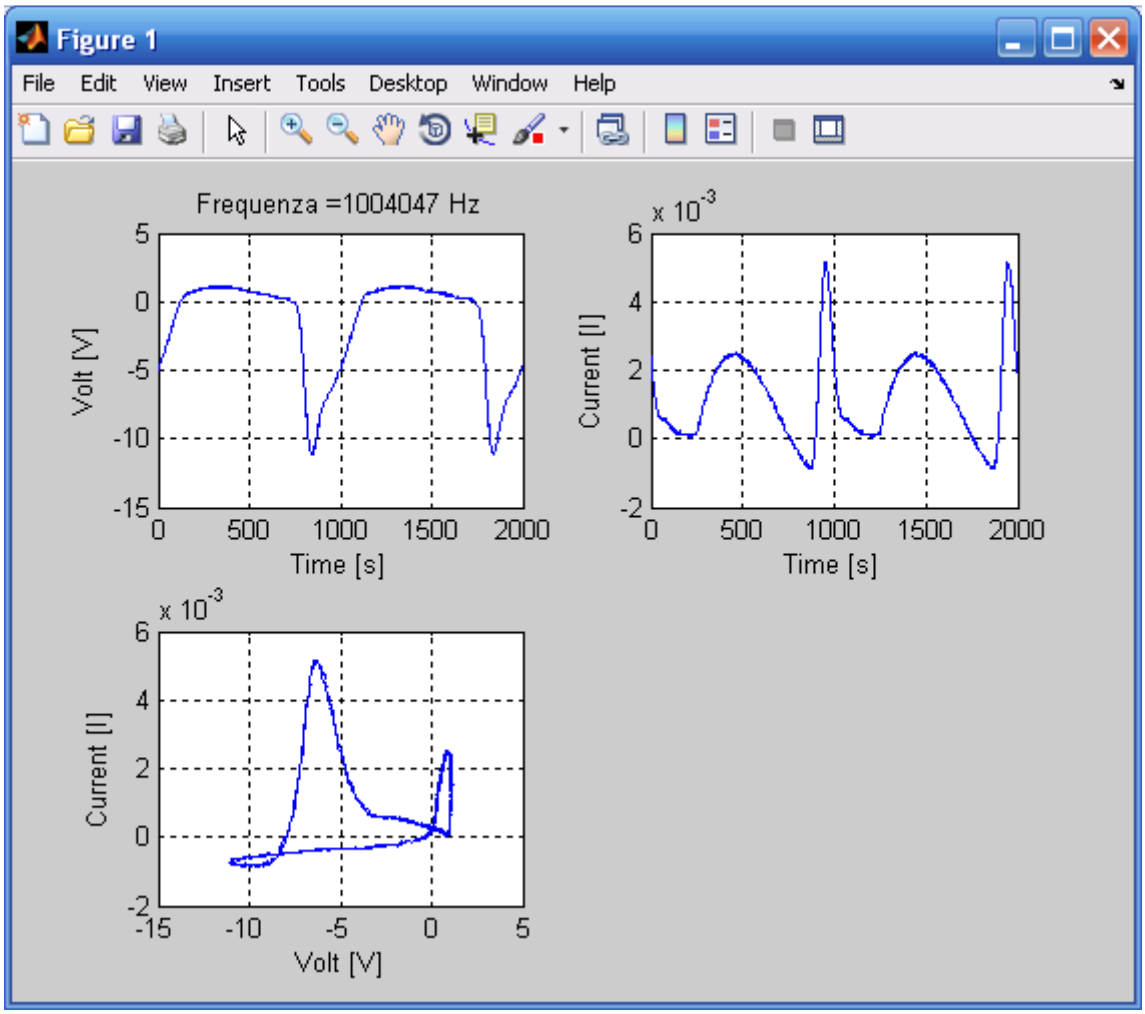


Figure 1
File Edit View Insert Tools Desktop Window Help
Frequenza =1004047 Hz
Volt [V]
Time [s]
x 10^-3
Current [I]
Time [s]
x 10^-3
Current [I]
Volt [V]