Graficar datos de ltspice

Unknown Author

August 31, 2013

1 Datos de tiempo

En general esto no presenta mayor dificultad, basta con graficar las variables del circuito que se quieren utilizar en python, despues en la ventana de resultados usar el menu file->export lo cual genera un archivo txt donde la información se de divide en columnas, donde la primera fila son las etiquetas de las variables. El eje x de la grafica siempre es la primera columna

1.1 Estructura del archivo

Etiquetas

Tiempo

Variable 1

. . .

Variable n

Datos

0

0.159898

. . .

-14454

1

-0.159898

. . .

0.14454

. . .

. . .

. . .

. . .

n

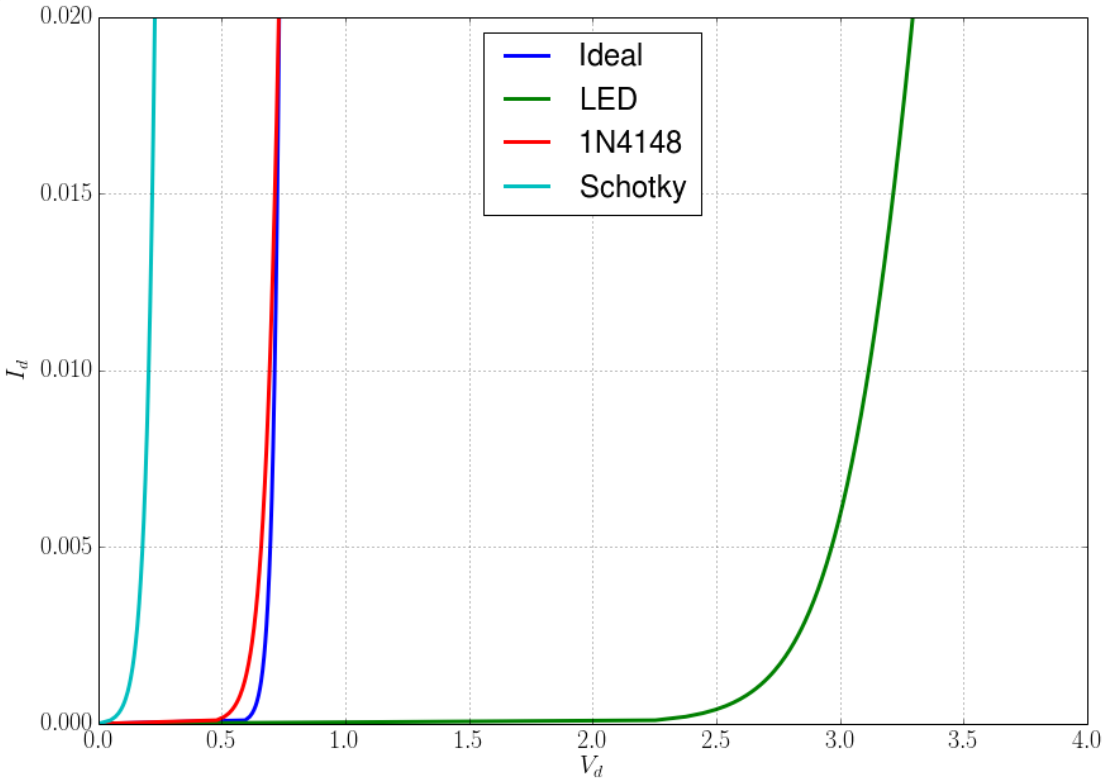
0

...

0

1.2 Ejemplo

In [82]:
```
import numpy as np
import pylab
varia=np.loadtxt('curvasdiodos.txt', skiprows=1)
matplotlib.rcParams.update({'font.size': 20,'text.usetex': True})
figure(figsize=(14,10), dpi=150)
xlabel('$V_{d}$')
ylabel('$I_{d}$')
xlim(0,4)
ylim(0,0.02)
plot(varia[:,1],varia[:,0],linewidth=3,label='Ideal')
plot(varia[:,2],varia[:,0],linewidth=3,label='LED')
plot(varia[:,3],varia[:,0],linewidth=3,label='1N4148')
plot(varia[:,4],varia[:,0],linewidth=3,label='Schotky')
legend(loc=9)
grid()
```



In [24]:
```
import numpy as np
import pylab
from scipy.signal import argrelextrema

filename = 'modulardoam[1].txt'

# Using the newer with construct to close the file automatically.
with open(filename) as f:
    data = f.readlines()

data2=[]
```

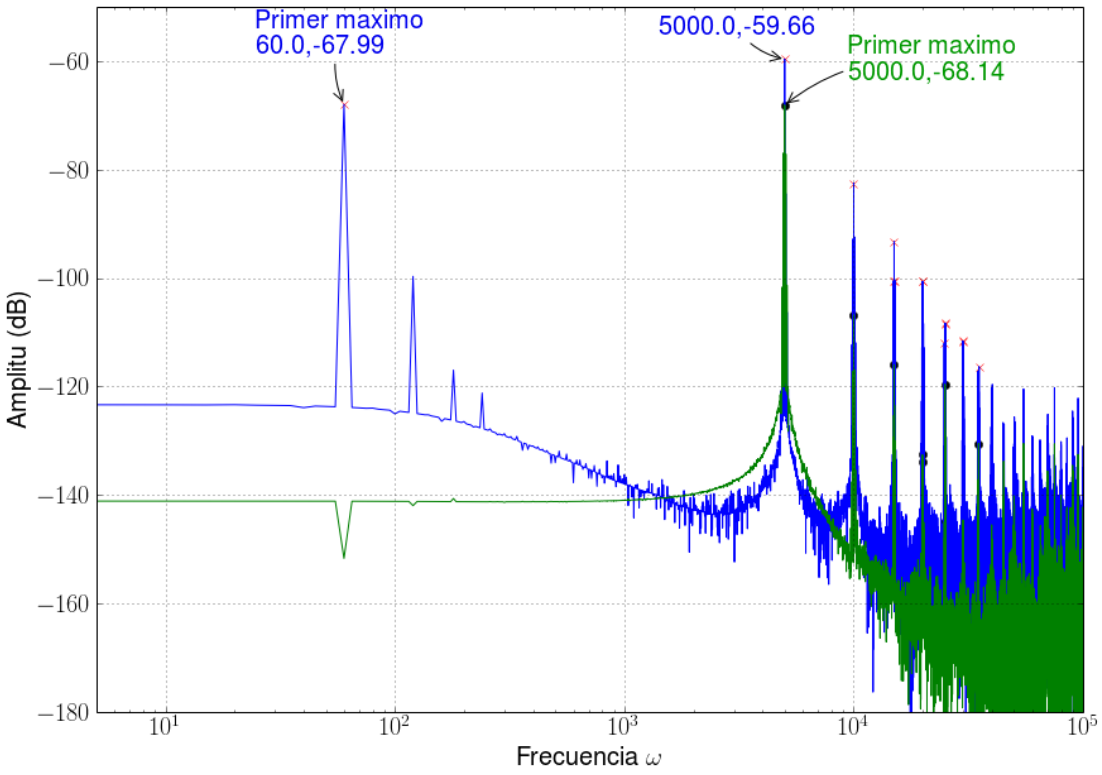
```
for i in data[1:len(data)]:
    data2.append(i.strip())
```

In [3]:
```
t=[]
i1=[]
i2=[]
for i in data2:
    s=i.split('\t')
    t.append(s[0])
    i1.append(s[1].split(','))
    i2.append(s[2].split(','))

i2n=np.asarray(i2,dtype=np.float32)
i1n=np.asarray(i1,dtype=np.float32)
t=np.asarray(t,dtype=np.float)
```

In [45]:
```
i1f=i1n[:,1]+1j*i1n[:,0]
i2f=i2n[:,1]+1j*i2n[:,0]
i1max=np.array([])
i2max=np.array([])
tmax=np.array([])
for i in range(0,len(t)):
    if 20*np.log10(np.abs(i1f[i])) > -120:
        i1max=np.append(i1max,i1f[i])
        i2max=np.append(i2max,i2f[i])
        tmax=np.append(tmax,t[i])
index=argrelextrema(np.abs(i1max), np.greater)
indexm=np.argmax(np.abs(i1max))
index2=argrelextrema(np.abs(i2max), np.greater)
indexm2=np.argmax(np.abs(i2max))
```

In [84]:
```
matplotlib.rcParams.update({'font.size': 20,'text.usetex': True})
figure(figsize=(14,10), dpi=150)
xlim(5,1e5)
ylim(-180,-50)
xlabel('Frecuencia $\omega$')
ylabel('Amplitu (dB)')
semilogx(tmax[index],20*np.log10(np.abs(i1max[index])),'rx')
semilogx(tmax[index2],20*np.log10(np.abs(i2max[index2])),'ko')
semilogx(t,20*np.log10(np.abs(i1f)))
semilogx(t,20*np.log10(np.abs(i2f)))
annotate(str(tmax[indexm])+','+str(round(20*np.log10(np.abs(i1max[indexm])),2)),
         xy=(tmax[indexm],20*np.log10(np.abs(i1max[indexm]))),
         xycoords='data',xytext=(-100, 20), textcoords='offset points',
         arrowprops=dict(arrowstyle="->",connectionstyle="arc3,rad=.2"),
         color = '#0000ee')
annotate('Primer maximo \n'+str(tmax[1])+','+str(round(20*np.log10(np.abs(i1max[1])),2
         xy=(tmax[1],20*np.log10(np.abs(i1max[1]))),  xycoords='data',xytext=(-70, 40)
         textcoords='offset points',arrowprops=dict(arrowstyle="->",connectionstyle="a
         color = '#0000ee')
annotate('Primer maximo \n'+str(tmax[indexm2])+','+str(round(20*np.log10(np.abs(i2max[
         xy=(tmax[indexm2],20*np.log10(np.abs(i2max[indexm2]))),
         xycoords='data',xytext=(50, 20), textcoords='offset points',
         arrowprops=dict(arrowstyle="->",connectionstyle="arc3,rad=.2"),
         color = '#009900')
grid()
savefig('images/espectroam.png')
```


Primer maximo
60.0,-67.99
5000.0,-59.66
Primer maximo
5000.0,-68.14
−60
−80
−100
−120
−140
−160
−180
Amplitu (dB)
10^1
10^2
10^3
10^4
10^5
Frecuencia ω

In [ ]: