Circuitos usando Diodos

Unknown Author

August 31, 2013

1 Curvas Caracteristicas

Con un circuito simple se puede utilizar el circuito que se muestra acontinuación:

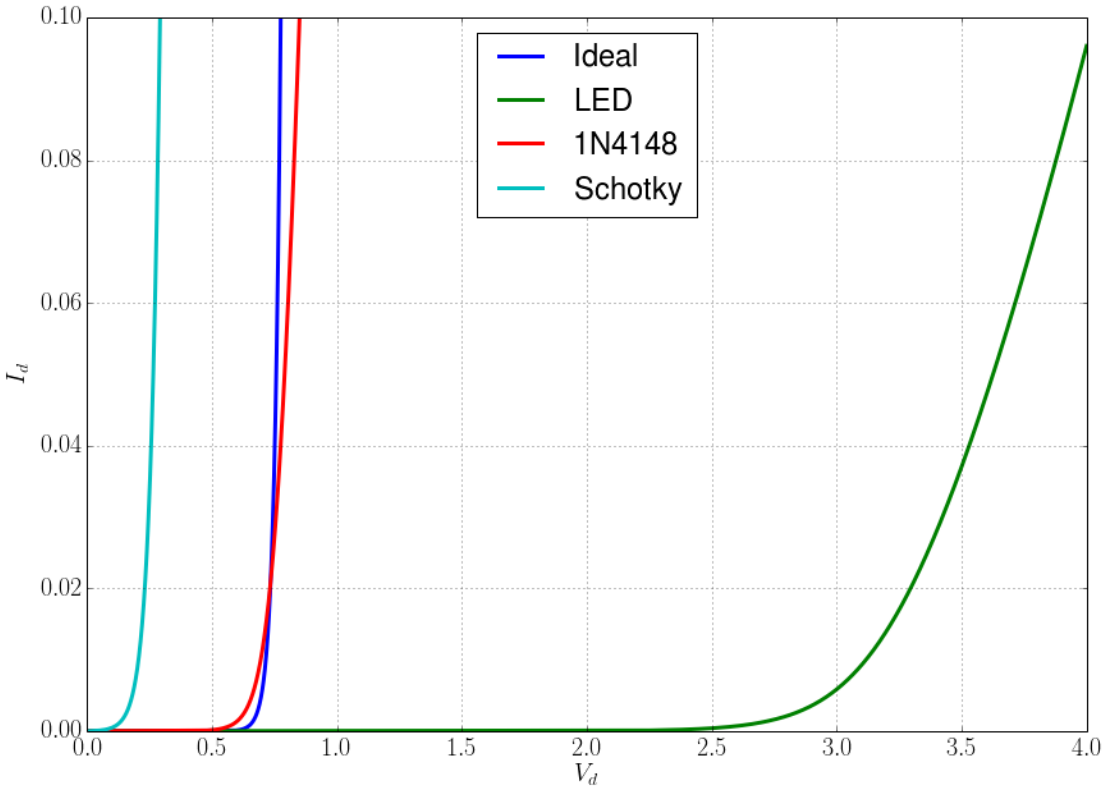
Figure 1: Circuito Básico

dado que la fuente de corriente puede tener cualquier valor de voltaje (teoricamente), si controlamos la corriente I_D entonces obtendremos una variación en el voltaje del diodo V_D, la cual debe de ser similar al expresado por la expresión:

$$I_D = I_s \left(e^{\frac{V_D}{nV_T}} - 1 \right) \tag{1}$$

Utilizando el program LTspice para obtener el comportamiento de diferentes diodos se obtiene la siguiente gráfica. (Los archivos son: diodos_curvas.asc el archivo de LTspice y el de curvasdiodos.txt contiene los datos que se usan en el ejemplo.

In [9]:
```
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(0,0.1)
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()
```



2 Circuito Rectificador de media onda

El circuito más simple con un diodo es el conocido como rectificador de media onda el cual consiste de una fuente de voltaje V_i, un resistencia R y un diodo D en seria, como se muestra en la siguiente figura

/files/images/diodosimple.png

Figure 2: Circuito con un diodo

Utilizando la ley de Kircchof de voltajes y la ley de Ohm se tiene

$$V_i = I_D R + V_D \tag{2}$$

dado que los elementos estan en serie se tiene que corriente es la misma, por lo tanto es posible encontrar la corriente total con la expresión para la corriente del diodo

$$I_D = I_s \left(e^{\frac{V_D}{nV_T}} - 1 \right) \tag{3}$$

sustituyendo (3) en (2)

$$V_i = I_s \left(e^{\frac{V_D}{nV_T}} - 1 \right) R + V_D \tag{4}$$

La expresión (4) es una ecuación trascendente (Una ecuación trascendente es una igualdad entre dos expresiones matemáticas en las que aparecen una o más incógnitas relacionadas mediante operaciones matemáticas, que no son únicamente algebraicas, y cuya solución no puede obtenerse empleando solo las herramientas propias del álgebra.) lo cual no permite encontrar una solución analitica.

Para resolver una ecuación trascendente se utilizan dos métodos que son: por estimación y gráfico.

2.1 Estimación

Considerese una expresión del tipo

$$y = f(x, p) \tag{5}$$

donde y es una salida conocida (Variable dependiente), x es una entrada (Variable independiente) y p son un conjunto de parámetros. Si $f(x, p)$ es una ecuación que no tiene solución analitica, entonces no se puede calcular su solución exacta. Sin embargo, es posible estimar una solución aproximada $\hat{y}$ con el método de estimación. En este método se asignan valores a la incognita hasta que el error (E_y) entre la solución exacta y y la solución estimada $\hat{y}$ es menor a un cierto valor establecido. Esto se puede expresar como:

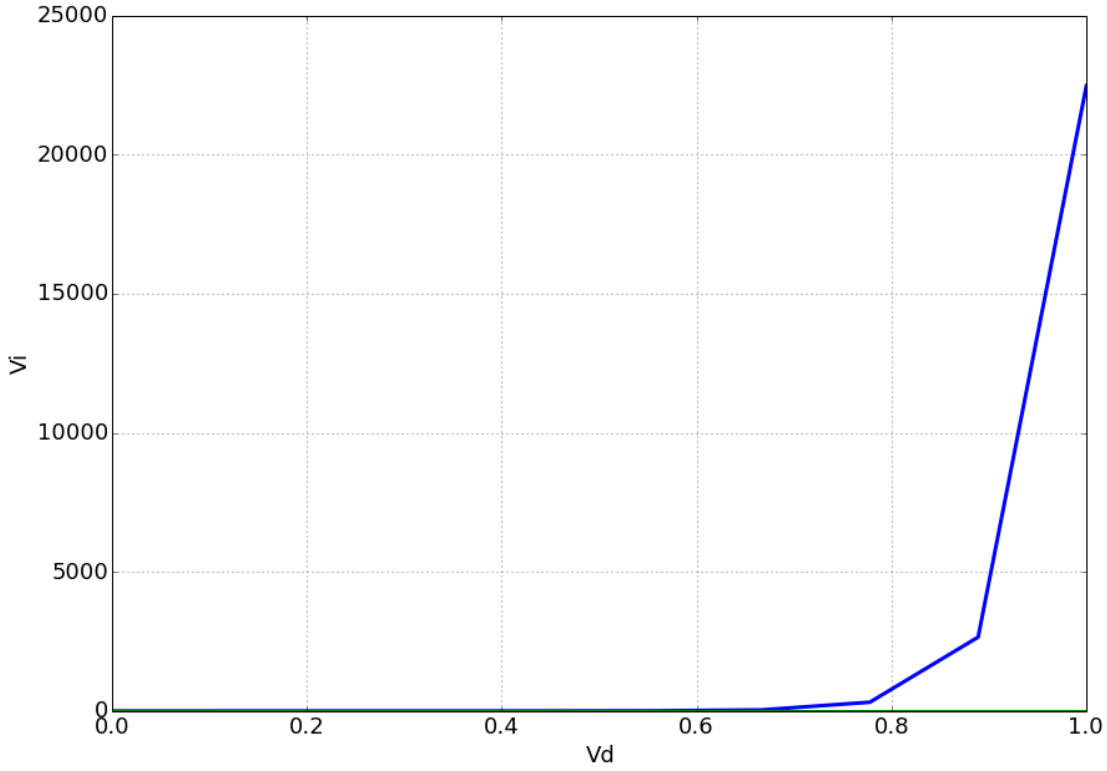
$$E_y = |y - \hat{y}| \leq k$$

Por lo que este metodo requiere que los coeficientes de la expresión a evular sea conocidos.

Ejemplo

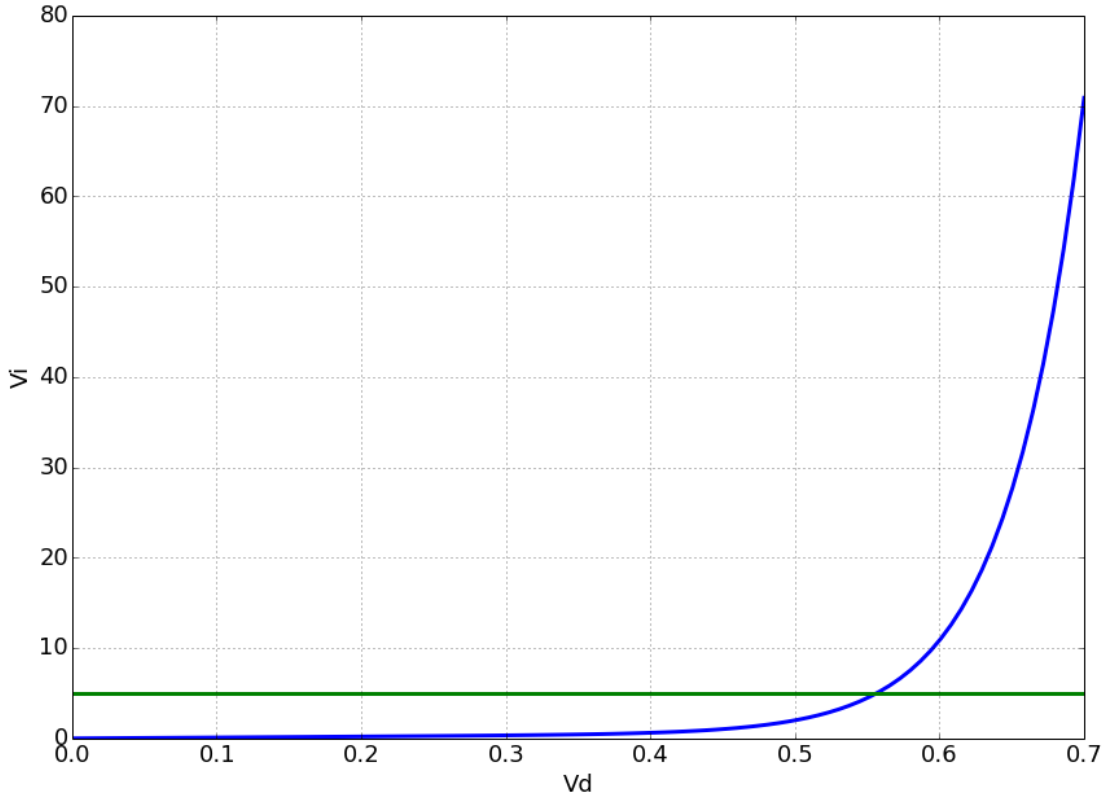
en el caso de la expresión (4) consideramos que $R = 2k\Omega$, $V_i = 5$, $n = 2$, y $V_T = 26mV$. En la siguiente gráfica se muestra el valor estimado de la expresión (4) al sustituir valores de los parámetros y considerar $0 \leq V_D \leq 1$

In [2]:
```
R=np.array([2e3])
n=np.array([2])
Vt=np.array([26e-3])
Is=np.array([.05e-6])
Vd=np.linspace(0, 1, 10)
Vi=np.ones(len(Vd))*5
yp=Is*(np.exp(Vd/(n*Vt))-1)*R+Vd
matplotlib.rcParams.update({'font.size': 18})
figure(figsize=(14,10), dpi=150)
xlabel('Vd')
ylabel('Vi')
plot(Vd,yp,linewidth=3)
plot(Vd,Vi,linewidth=3)
grid()
```



De la gráfica anterior se observa que para valores de V_D mayores a 0.8, la salida estimada es mucho mayor que la salida esperada, por lo que haciendo una nueva iteración $0 \leq V_D \leq 0.7$, se obtiene la siguiente gráfica

In [3]:
```
Vd=np.linspace(0, .7, 100)
Vi=np.ones(len(Vd))*5
yp=Is*(np.exp(Vd/(n*Vt))-1)*R+Vd
matplotlib.rcParams.update({'font.size': 18})
figure(figsize=(14,10), dpi=150)
xlabel('Vd')
ylabel('Vi')
plot(Vd,yp,linewidth=3)
plot(Vd,Vi,linewidth=3)
grid()
```

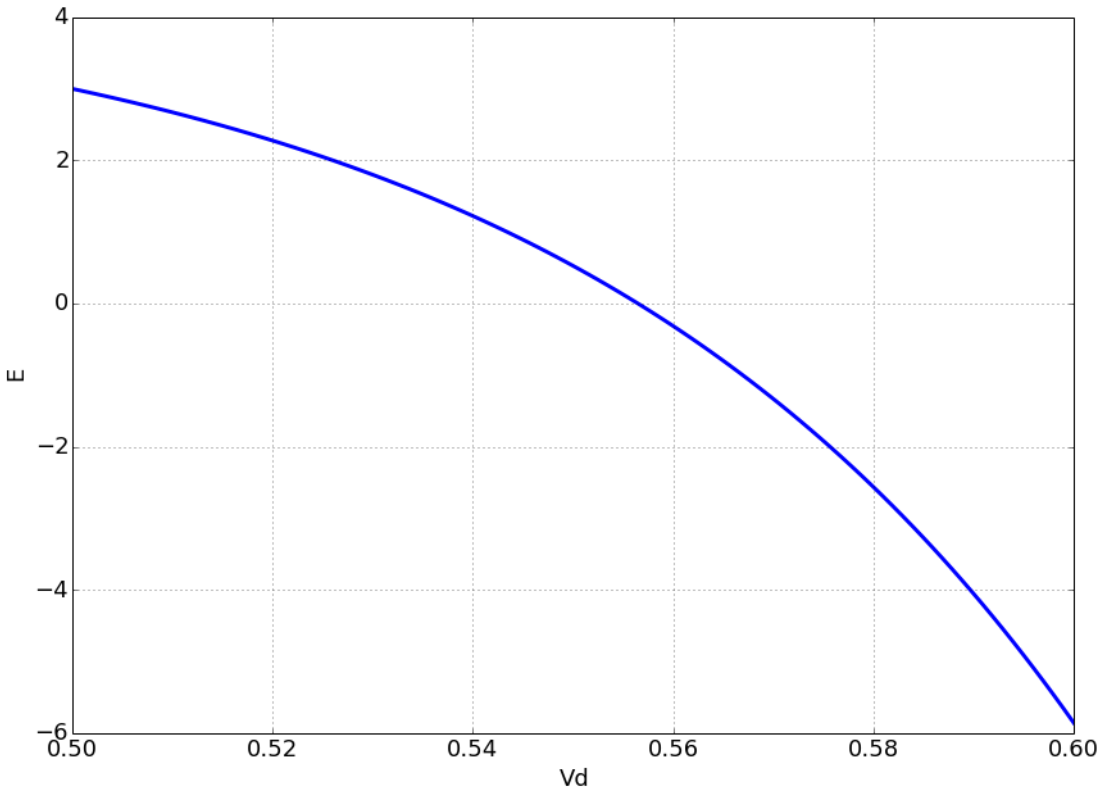


De los resultados mostrados en la gráfica anterior se puede concluir que el valor estimado se aproxima al valor deseado caundo $0.5 \leq V_D \leq 0.6$, para realizar una mejor aproximación se utiliza el error de estimacion

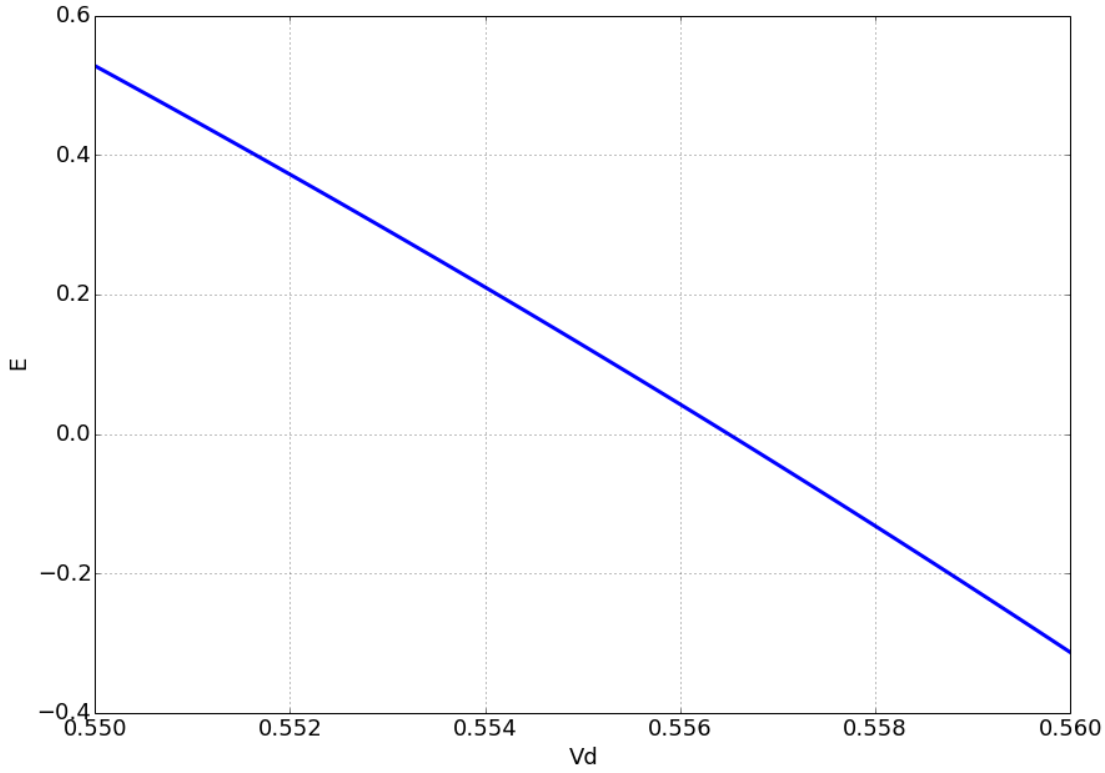
$$E = V_i - \hat{V}_i \tag{6}$$

donde $\hat{V}_i$ es el valor estimado a través de la expresion (4)

In [4]:
```
Vd=np.linspace(0.5, .6, 100)
Vi=np.ones(len(Vd))*5
yp=Is*(np.exp(Vd/(n*Vt))-1)*R+Vd
E=5-yp;
matplotlib.rcParams.update({'font.size': 18})
figure(figsize=(14,10), dpi=150)
xlim(0.5,0.6)
xlabel('Vd')
ylabel('E')
plot(Vd,E,linewidth=3)
grid()
```



In [5]:
```
Vd=np.linspace(0.55, .56, 100)
Vi=np.ones(len(Vd))*5
yp=Is*(np.exp(Vd/(n*Vt))-1)*R+Vd
E=5-yp;
matplotlib.rcParams.update({'font.size': 18})
figure(figsize=(14,10), dpi=150)
xlabel('Vd')
ylabel('E')
plot(Vd,E,linewidth=3)
grid()
```



In [6]:
```
Vd=0.556495
Vi=5
yp=array([])
yp=Is*(np.exp(Vd/(n*Vt))-1)*R+Vd
print("El valor estimado es de:{0}".format(yp))
```

```
El valor estimado es de:[ 5.00009139]
```

In [6]: