Circuitos usando Diodos

Manuel Moisés Miranda Velasco

September 11, 2013

1 Curvas Caracteristicas

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

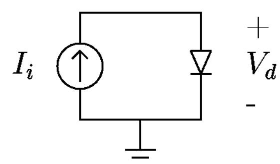


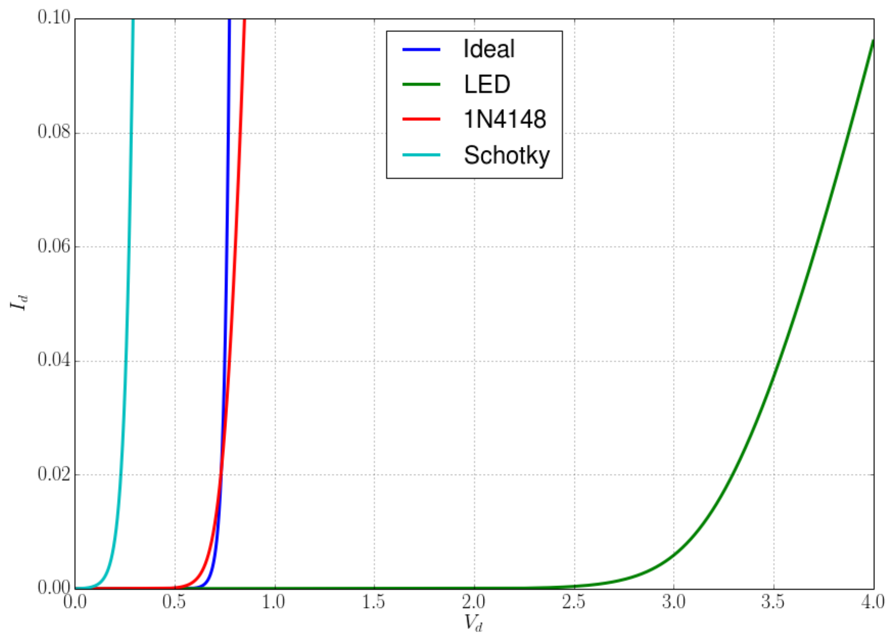
Figure 1: Circuito Básico

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 [1]:
```
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

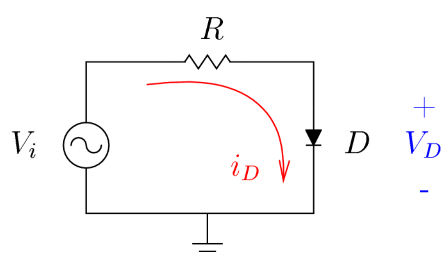


Figure 2: Circuito con un diodo

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:

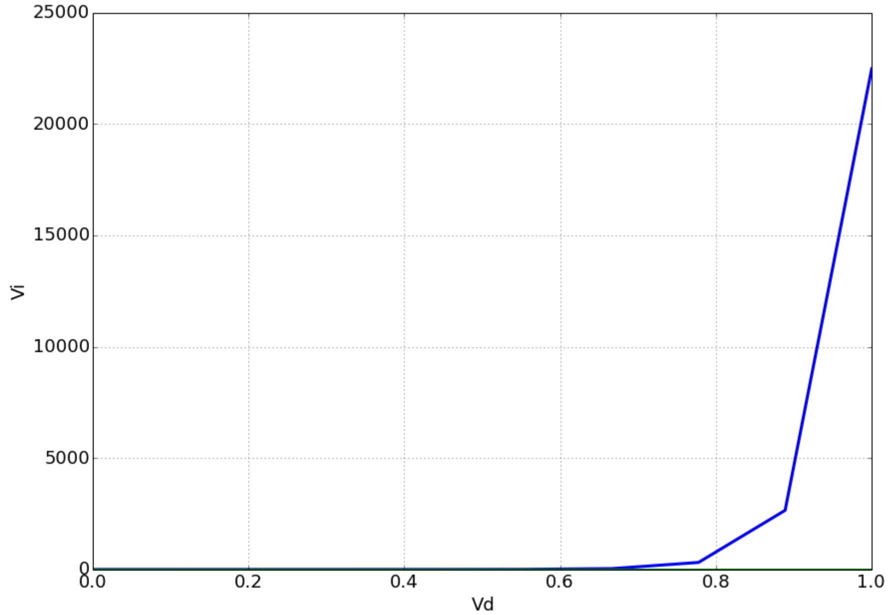
$$E_y = |y - \hat{y}| \leq k$$

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

Ejemplo 1

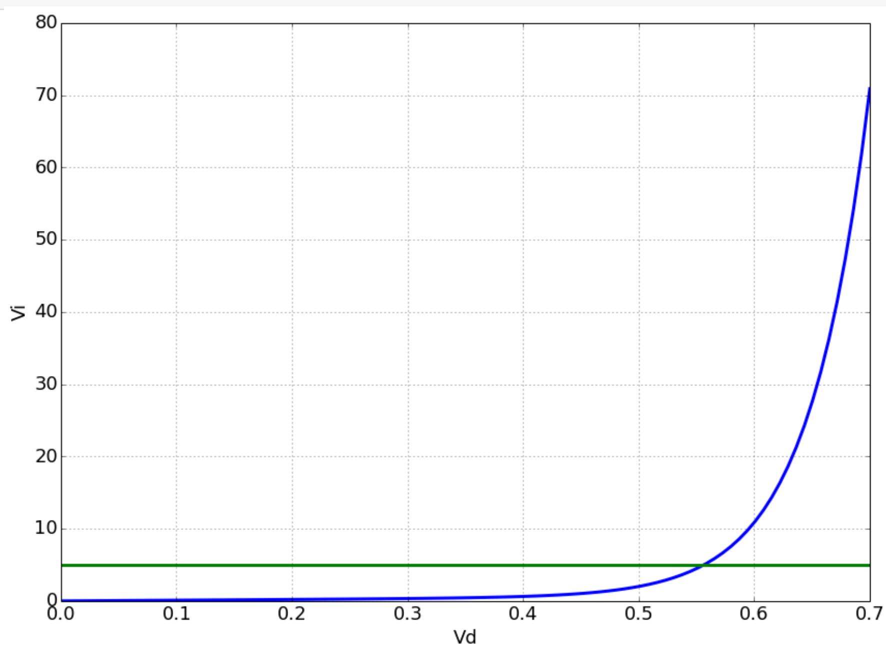
en el caso de la expresión (4) consideramos que $R = 2k\Omega$, $V_i = 5$, $n = 2$, y $V_T = 26mV$. En la siguiente gráfica se muestra el valor estimado de la expresión (4) al sustituir valores de los parámetros y considerar $0 \leq V_D \leq 1$

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()
```

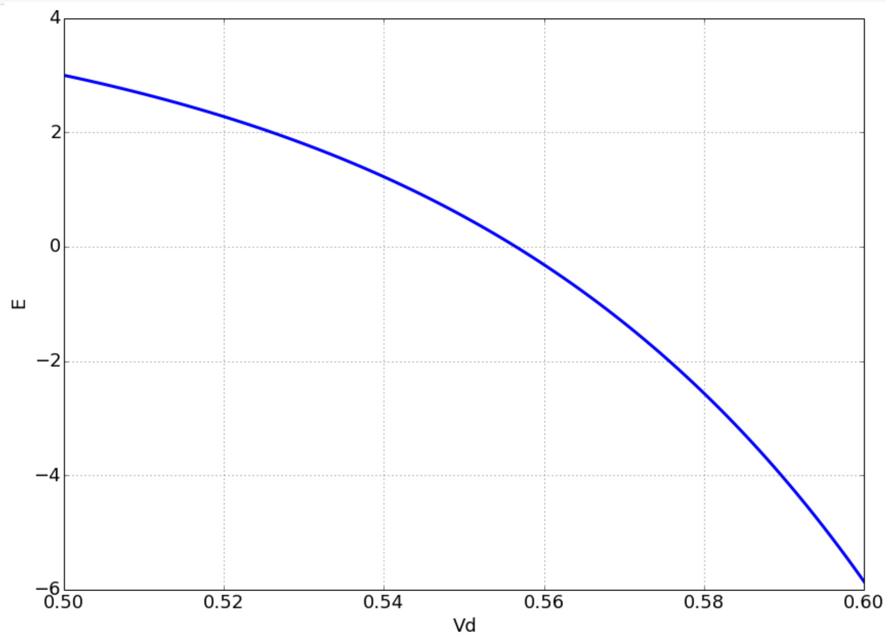


De los resultados mostrados en la gráfica anterior se puede concluir que el valor estimado se aproxima al valor deseado caundo $0.5 \leq V_D \leq 0.6$, para realizar una mejor aproximación se utiliza el error de estimacion

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

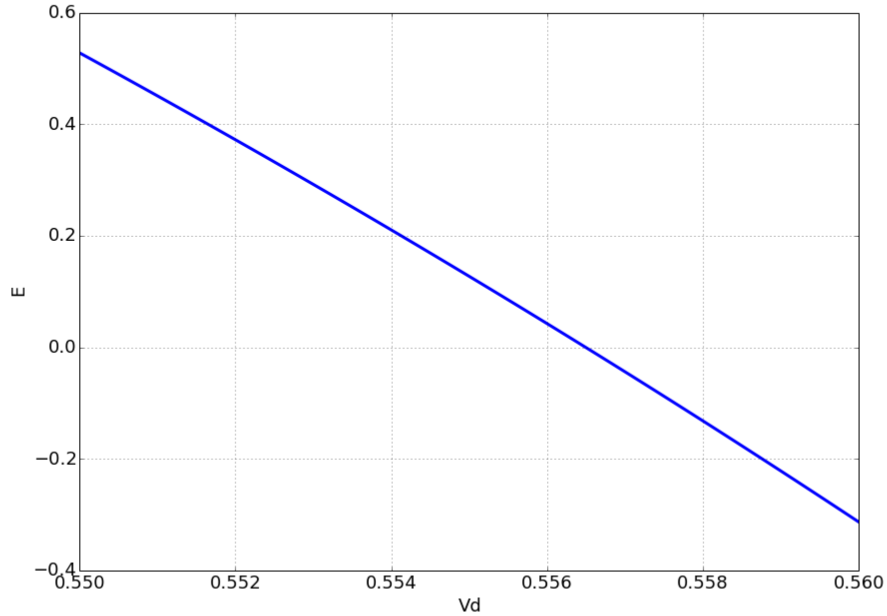
donde $\hat{V}_i$ es el valor estimado a través de la expresion (4)

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()
```



De la gráfica anterior se observa que el error E es igual a 0 cuando $0.55 \leq V_D \leq 0.56$, por lo que hacuendo una nueva aproximación es posible encontrar un valor exacto. El siguiente paso es proponer que el intervalo $0.55 \leq V_D \leq 0.56$.

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()
```



Se observa que el error $E = 0$ se encuentra en el intervalo $0.556 \leq V_D \leq 0.557$, para encontrar un error con una buena presición se tiene que realizar este proceso más iteraciones, en este caso con ayuda de los **métodos númericos** se cuenta con algoritmos que hacen posible el cálculo de estos valores de forma automatica.

Usando uno de estos metodos (metodo de biseccion) se encuentra que el valor de V_D que produce un $E < 1^{-4}$ se tiene que:

In [2]:
```
import scipy.optimize as optimize

def func(Vd):
    Vi=5
    Is=0.05e-6
    R=2e3
    Vt=26e-3
    n=2
    return Is*(np.exp(Vd/(n*Vt))-1)*R+Vd-Vi

print("El valor estimado de Vd que soluciona el problema es:",optimize.bisect(func,0.5
```

```
El valor estimado de Vd que soluciona el problema es:
0.5564939428778598
```

el valor de la corriente del diodo utilizando (1) es $I_d = 0.0022217530285451636 \approx 2.22mA$

In [4]:
```
Is=0.05e-6
Vt=26e-3
n=2
Vd=0.5564939428778598
Is*(np.exp(Vd/(n*Vt))-1)
```

Out [4]:
```
0.0022217530285451636
```

3 Analisis de DC

En el caso de un circuito con diodos el sistema se puede analizar en dos partes, cuando el diodo esta polarizado en directa y cuando esta polariazado en inversa.

3.1 Polarización directa

En el caso ideal se tiene que un circuito con un diodo que esta polarizado en directa el diodo se puede considerar como un interruptor cerrado, tal como se muestra en la siguiente circuito

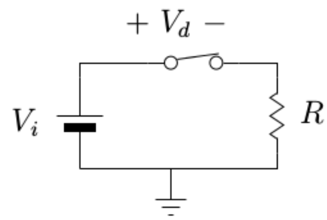


Figure 3: Circuito con un diodo polarizado en directa

donde $V_d = 0$ y por lo tanto la corriente se obtiene con la expresión

$$I_d = \frac{V_i}{R} \tag{7}$$

Sin embargo, la expresión anterior es incorrecta si consideramos que la corriente dada en el diodo es determinada por la expresión (1). En general del ejemplo anterior se observa que es dificil calcular el valor de la corriente con las ecuaciones exactas, por lo que se simplifica el problema que una vez establecida la corriente de DC, el voltaje del diodo $V_d = V_k$ donde $V_k \geq 0$, siendo el valor de V_k determinado de forma heuristica en función del tipo de diodo, por lo que una función aproximada para calcular la corriente sería:

$$I_d = \frac{V_i - V_k}{R} \tag{8}$$

Si consideramos la expresión (8) en el ejemplo 1, considerando que $V_k = 0.7$ se tiene

$$I_d = \frac{V_i - V_k}{R} = \frac{(5 - 0.7)V}{2k\Omega} = 2.15mA$$

lo cual solo tiene un error de $0.08mA$ con respecto al cálculo utilizando la expresión (1)

3.2 Polarización inversa

En el caso de la polarizción inversa se tiene que el diodo se puede sustituir con un interruptor abierto como se muestra en la figura que sigue:

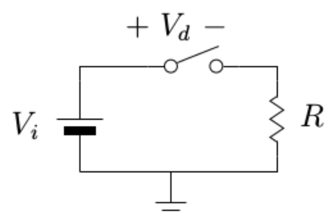


Figure 4: Circuito con un diodo polarizado en inversa

en el caso del circuito anterior se tiene que el voltaje en la resistencia $V_R = 0$, ya que no fluye corriente alguna, por lo que se tiene que $V_d = V_i$. Lo anterior será valido mientras no se alcance el voltaje de conducción en inversa (Voltaje de avalancha V_a o Voltaje Zenner V_z), ya que al alcanzar el voltaje de conducción en inversa el diodo se comporta de forma similar cuando esta polarizado en directa. Solo que ahora el voltaje utilizado para la expresión (8) no es V_k sino el voltaje de avalancha V_A

$$I_d = \frac{V_i - V_A}{R} \tag{9}$$

4 Circuito Rectificador de Onda Completa

El circuito rectificador de onda completa se muestra en la siguiente figura.

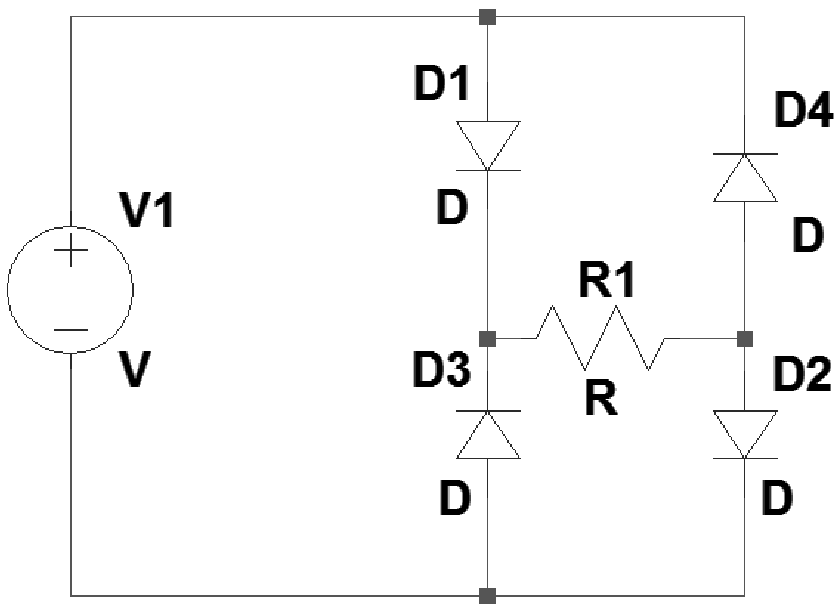


Figure 5: Rectificador de Onda Completa

4.1 Ciclo positivo

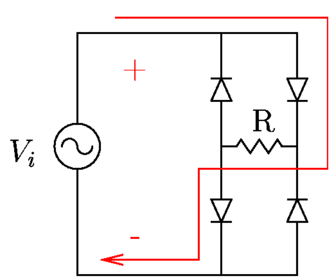


Figure 6: Rectificador de Onda Completa Ciclo Positivo

4.2 Ciclo negativo

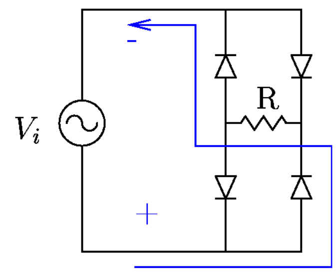


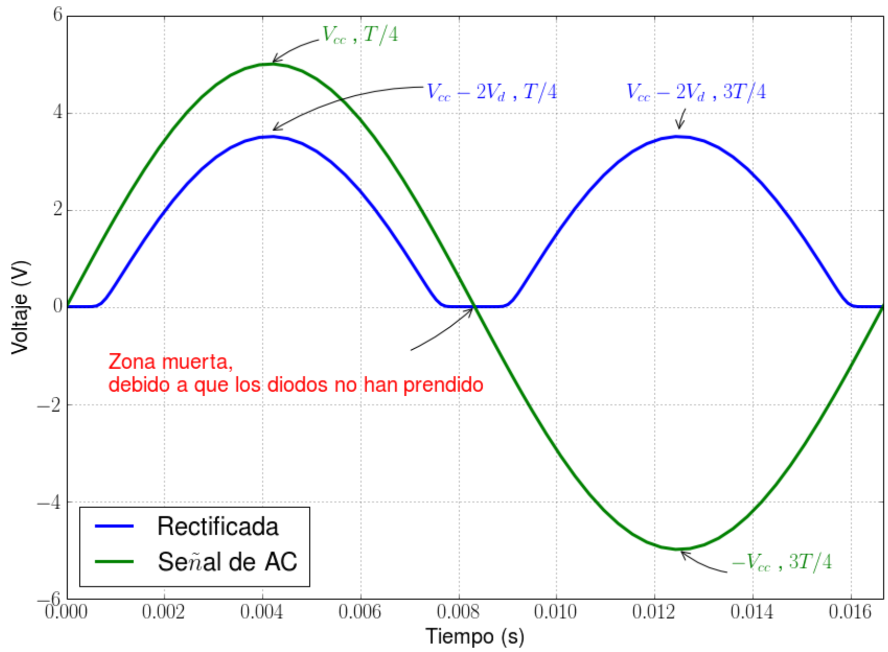
Figure 7: Rectificador de Onda Completa Ciclo Negativo

In [5]:

```
import numpy as np
import pylab
varia=np.loadtxt('puentedediodos2.txt', skiprows=1)
matplotlib.rcParams.update({'font.size': 20,'text.usetex': True})
figure(figsize=(14,10), dpi=150)

plot(varia[:,0],varia[:,1],linewidth=3,label='Rectificada')
plot(varia[:,0],varia[:,2],linewidth=3,label='AC')
xlabel('Tiempo (s)')
ylabel('Voltaje (V)')
xlim(0, 1/60)

legend(['Rectificada','Se$\~{n}$al de AC'],loc=3)
grid()
annotate('$V_{cc}$ , $ T/4$',
         xy=(1/240,5),
         xycoords='data',xytext=(50, 20), textcoords='offset points',
         arrowprops=dict(arrowstyle="->",connectionstyle="arc3,rad=.2"),
         color = 'green')
annotate('$-V_{cc}$ , $ 3T/4$',
         xy=(3/240,-5),
         xycoords='data',xytext=(50, -20), textcoords='offset points',
         arrowprops=dict(arrowstyle="->",connectionstyle="arc3,rad=-.3"),
         color = 'green')

annotate('$V_{cc}-2V_{d}$ , $ T/4$',
         xy=(1/240,5-1.4),
         xycoords='data',xytext=(150, 30), textcoords='offset points',
         arrowprops=dict(arrowstyle="->",connectionstyle="arc3,rad=.2"),
         color = 'blue')

annotate('$V_{cc}-2V_{d}$ , $ 3T/4$',
         xy=(3/240,5-1.4),
         xycoords='data',xytext=(-50, 30), textcoords='offset points',
         arrowprops=dict(arrowstyle="->",connectionstyle="arc3,rad=.2"),
         color = 'blue')

annotate('Zona muerta, \n debido a que los diodos no han prendido',
         xy=(2/240,0),
         xycoords='data',xytext=(-350, -80), textcoords='offset points',
         arrowprops=dict(arrowstyle="->",connectionstyle="arc3,rad=.2"),
         color = 'red')
```

Out [5]:

```
<matplotlib.text.Annotation at 0x69e2780>
```



5 Fuente de voltaje de CD con diodos

5.1 Transformador

Se denomina transformador a un dispositivo eléctrico que permite aumentar o disminuir la tensión en un circuito eléctrico de corriente alterna, manteniendo la potencia. La potencia que ingresa al equipo, en el caso de un transformador ideal (esto es, sin pérdidas), es igual a la que se obtiene a la salida. Las máquinas reales presentan un pequeño porcentaje de pérdidas, dependiendo de su diseño y tamaño, entre otros factores.

El transformador es un dispositivo que convierte la energía eléctrica alterna de un cierto nivel de tensión, en energía alterna de otro nivel de tensión, basándose en el fenómeno de la inducción electromagnética. Está constituido por dos o más bobinas de material conductor, devanadas sobre un núcleo cerrado de material ferromagnético, pero aisladas entre sí eléctricamente. La única conexión entre las bobinas la constituye el flujo magnético común que se establece en el núcleo. El núcleo, generalmente, es fabricado bien sea de hierro o de láminas apiladas de acero eléctrico, aleación apropiada para optimizar el flujo magnético. Las bobinas o devanados se denominan primario y secundario según correspondan a la entrada o salida del sistema en cuestión, respectivamente. También existen transformadores con más devanados; en este caso, puede existir un devanado “terciario”, de menor tensión que el secundario.

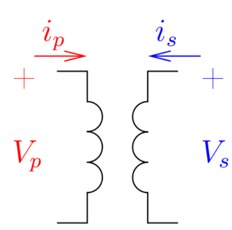


Figure 8: Transformador

En un transformador ideal, en el que el acoplamiento magnetico es perfecto se tiene que la energía de la bobina primaria es reflejada totalmente en la bobina secundaria, por lo que se tiene:

$$V_p i_p = V_s i_s \tag{10}$$

de la expresión (10) se puede relacionar el voltaje de las bobinas con el numero de vueltas de las mismas bobinas, a través de la expresión

$$\frac{V_p}{V_s} = \frac{N_p}{N_s} = \frac{i_s}{i_o} = \sqrt{\frac{L_p}{L_s}} = m \tag{11}$$

en la expresión anterios se tiene que la relación del primario y del secundario de un transformador, se puede expresar por el numero de vueltas de las bobinas N_x o la inductancia de cada uno de los transformadores L_x. Si conocemos el valor de la L_p entonces de (11) se tiene

$$L_s = \frac{L_p}{m^2} \tag{12}$$

5.2 Fuente de Voltaje de CD

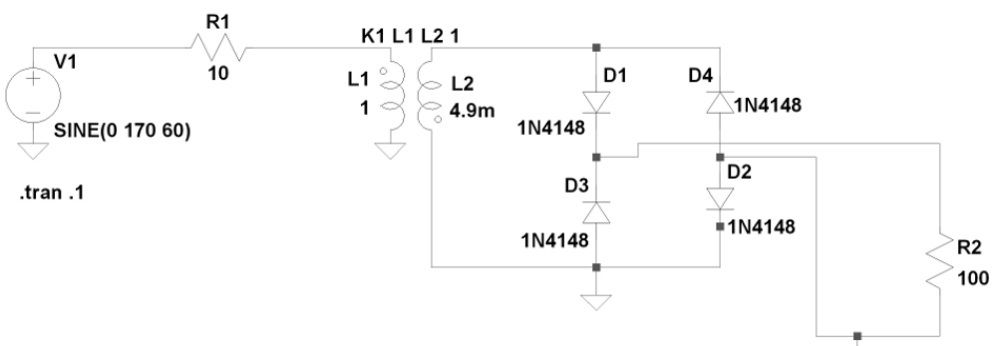


Figure 9: Fuente de DC simple

El circuito de la figura enterior solamente rectificara una señal de AC con voltaje V_p a la entrada del primario del transformador a $V_s - 2 * V_d$ y una frecuencia de $2f_p$, donde V_s es el voltaje en el secundario del transformador, V_d es el voltaje de disparo del diodo y f_p es la frecuencia de la señal en el primario del transformador. Que es el comportamiento de un puente de diodos.

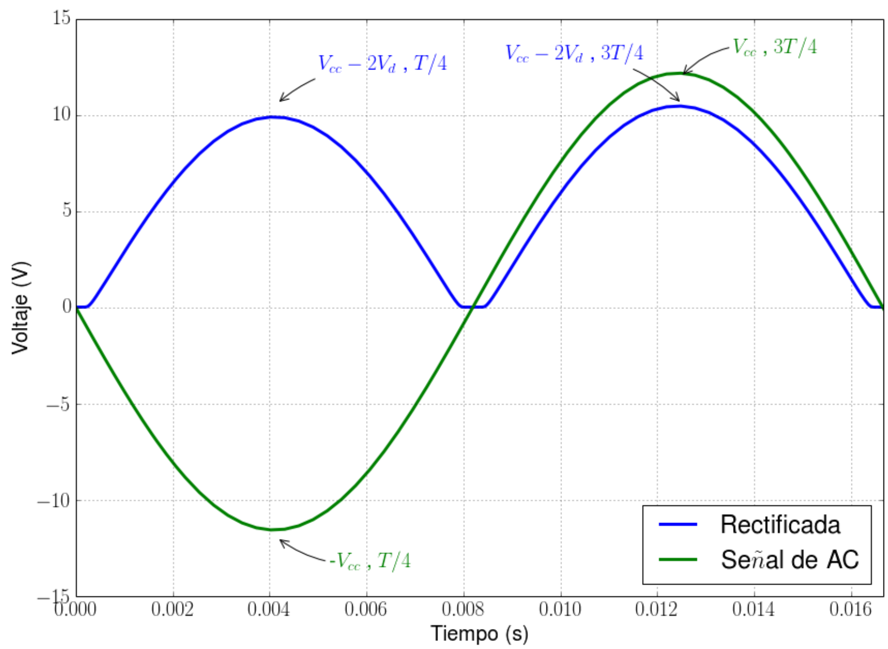
In [19]:

```
import numpy as np
import pylab
varia=np.loadtxt('fuentesimplesinfiltro.txt', skiprows=1)
matplotlib.rcParams.update({'font.size': 20,'text.usetex': True})
figure(figsize=(14,10), dpi=150)

plot(varia[:,0],varia[:,1],linewidth=3,label='Rectificada')
plot(varia[:,0],varia[:,2],linewidth=3,label='AC')
xlabel('Tiempo (s)')
ylabel('Voltaje (V)')
xlim(0, 1/60)

legend(['Rectificada','Se$\~{n}$al de AC'],loc=4)
grid()
annotate('-$V_{cc}$ , $ T/4$',
         xy=(1/240,-12),
         xycoords='data',xytext=(50, -30), textcoords='offset points',
         arrowprops=dict(arrowstyle="->",connectionstyle="arc3,rad=-.2"),
```

```
            color = 'green')
annotate('$V_{cc}$ , $ 3T/4$',
            xy=(3/240,12),
            xycoords='data',xytext=(50, 20), textcoords='offset points',
            arrowprops=dict(arrowstyle="->",connectionstyle="arc3,rad=.3"),
            color = 'green')

annotate('$V_{cc}-2V_{d}$ , $ T/4$',
            xy=(1/240,12-1.4),
            xycoords='data',xytext=(40, 30), textcoords='offset points',
            arrowprops=dict(arrowstyle="->",connectionstyle="arc3,rad=.2"),
            color = 'blue')

annotate('$V_{cc}-2V_{d}$ , $ 3T/4$',
            xy=(3/240,12-1.4),
            xycoords='data',xytext=(-170, 40), textcoords='offset points',
            arrowprops=dict(arrowstyle="->",connectionstyle="arc3,rad=-.2"),
            color = 'blue')
```

```
<matplotlib.text.Annotation at 0x9ee7f98>
```

Out [19]:



Para lograr una señal de DC, se agrega un capacitor C en paralelo a la Resistencia R_2, tal como se muestra en la siguiente figura

El comportamiento de este circuito se muestra en la siguiente figura:

El comportamiento mostrado en la figura anterios se puede considerar el voltaje de salida (grafica verde) V_c tiene un componente de DC (offset) más un componente de AC llamado voltaje de rizo.

Se tiene que el voltaje máximo del capacitor V_m es:

$$V_m = V_s - 2V_d \tag{13}$$

el voltaje mínimo V_l es determinado por la expresión:

$$V_l = V_m e^{\frac{-T_d}{RC}} \tag{14}$$

a partir de las expresiones (13) y (14) se de define el volateje de rizo como:

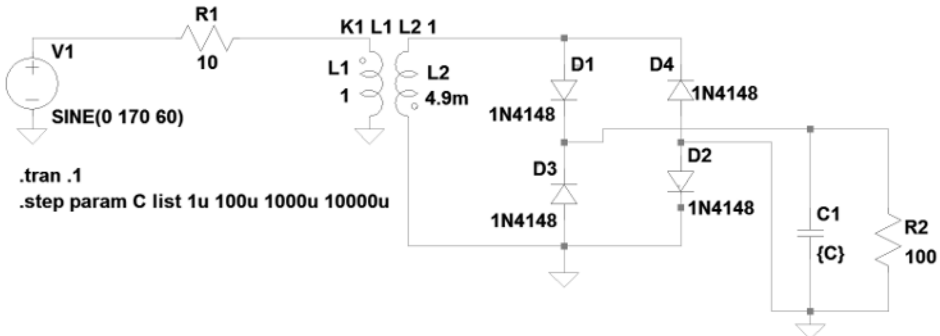


Figure 10: Fuente de DC simple

$$V_r = V_m - V_l = V_m - V_m e^{\frac{-T_d}{RC}} = V_m(1 - e^{\frac{-T_d}{RC}}) \tag{15}$$

la expresión (15) es una ecuación trasendente por lo que no tiene solución analitica, por lo que para obtener una solución aproximada se utiliza la expansión en series de Taylor de la siguiente manera:

$$e^{\frac{-T_d}{RC}} \approx 1 - \frac{T_d}{RC} \tag{16}$$

sustituyendo (16) en (15) se tiene

$$V_r \approx V_m \frac{T_d}{RC} \tag{17}$$

Se tiene que el tiempo de descarga del capacitor $T_d \leq T$, por lo que se considera que

$$V_r \leq V_m \frac{T}{RC} \tag{18}$$

In [6]:

```
filename = 'fuentesimple.txt'

# Using the newer with construct to close the file automatically.
with open(filename) as f:
    data2 = f.readlines()
k=0
inde=[]
for i in data2:
    if 'Step' in i:
        inde=append(inde,k)
    k +=1
inde=append(inde,len(data2))
k=len(inde)

leninde=[]
for i in range(0,k-1):
    leninde=append(leninde,inde[i+1]-inde[i])

print("Se encontraron {} pasos en la simulacion".format(k-1))
print(leninde)
```

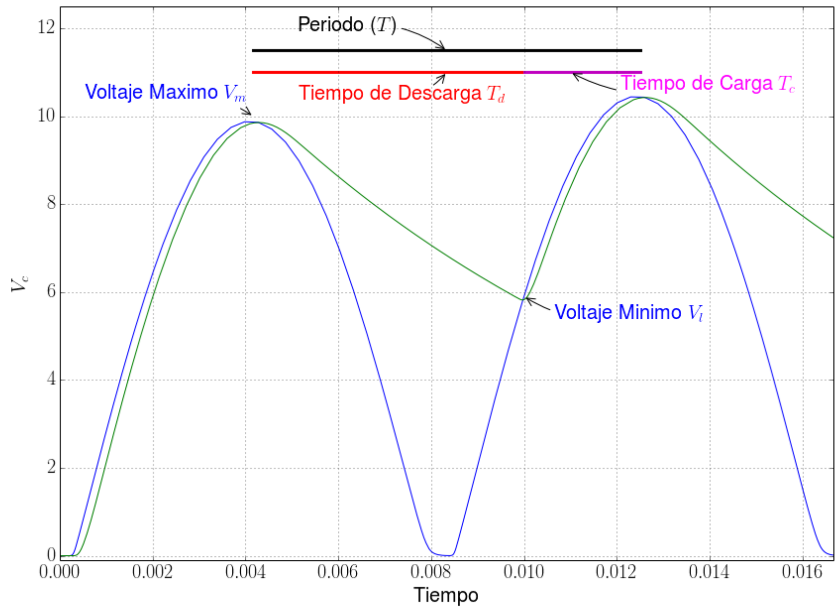
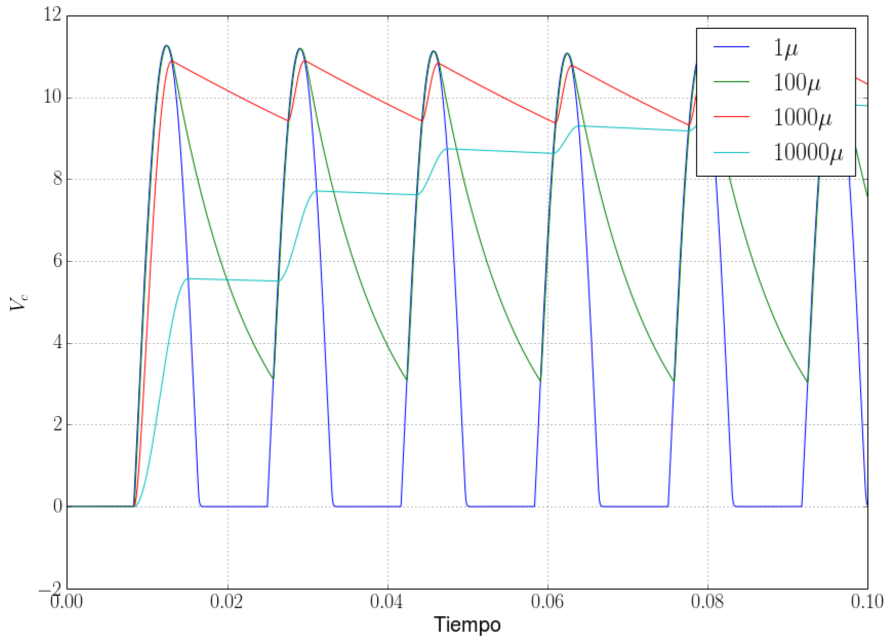


Figure 11: Salida Filtro

```
matplotlib.rcParams.update({'font.size': 20,'text.usetex': True})
figure(figsize=(14,10), dpi=150)

for inter in range(1,5):
    datainter=[]
    for i in range(int(inde[inter-1]+1),int(inde[inter]-1)) :
        datainter = append(datainter,data2[i].strip())

    tam1=len(datainter)
    tam2=len(datainter[1].split('\t'))

    datainter2=np.zeros((tam1,tam2))
    for i in range(tam1):
        s=datainter[i].split('\t')
        for k in range(tam2):
            datainter2[i,k]=np.asarray(s[k],dtype=np.float32)
    plot(datainter2[:,0],datainter2[:,2])
xlabel('Tiempo')
ylabel('$V_{c}$')
legend(['$1\mu$','$100\mu$','$1000\mu$','$10000\mu$'])
grid()
```

```
Se encontraron 4 pasos en la simulacion
[ 1434.  1235.  1240.  1273.]
```



6 Circuito Modulador de AM

Uno de los primeros circuitos usados en la electrónica fueron los relacionados con las comunicaciones, siendo la modulación en amplitud una de las primeras tecnicas utilizadas. Uno de los circuitos teoricos más faciles de usar es el que se muestra acontinuación.

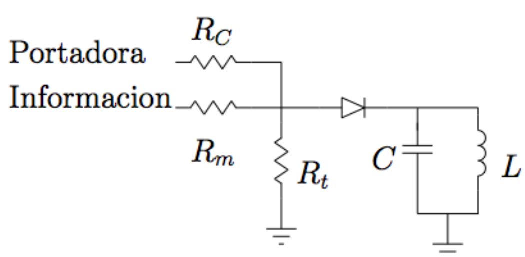


Figure 12: Modulador AM con un diodo

```
In [ ]:
```