

Ficha Prática N.º 6

Nesta folha de exercícios pretende-se que os alunos compreendam o princípio de funcionamento de um diodo retificador e, simultaneamente, consigam adquirir competências que lhes permitam analisar e projetar circuitos com diodos.

1. Análise de circuitos com um diodo

Recordamos que um diodo é um dispositivo não linear, que apenas conduz quando a queda de tensão aos seus terminais (V_{ak} - tensão do ânodo (A) para o cátodo (K)) for superior à tensão de arranque (V_a). Apresenta-se, em seguida, o modelo matemático deste dispositivo:

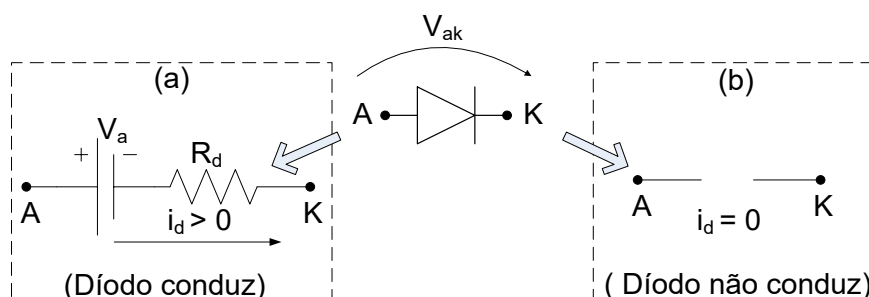


Fig. 1 – Modelo matemático do diodo retificador

- Se a tensão aos terminais do diodo for positiva e superior à tensão de arranque ($V_{ak} > V_a$), verifica-se um aumento significativo da corrente. Este aumento é limitado pela resistência interna do diodo. Deste modo, ambos os efeitos da resistência interna (R_d) e da tensão de arranque (V_a) deverão ser incluídos no modelo que permite representar o diodo quando diretamente polarizado. Deste modo, quando diretamente polarizado, se a tensão aplicada for superior à tensão de arranque ($V_{ak} > V_a$), a corrente no diodo cresce a um ritmo inversamente proporcional ao valor da sua resistência interna. Neste caso, o circuito equivalente deve ser composto por uma fonte de tensão independente (V_a) em série com uma resistência (R_d)^[1,2], como é possível observar na Fig. 1a.
- Caso o diodo se encontre inversamente polarizado, ignora-se o efeito da corrente inversa, e considera-se que o diodo se comporta como um circuito aberto^[1,2] (Fig. 1b).

^[1] Acácio Amaral, “Electrónica Analógica: Princípios, Análise e Projectos”, Edições Sílabo, 2017, Lisboa.

^[2] Acácio Amaral, “Electrónica Aplicada”, Edições Sílabo, 2021, Lisboa.

Nesta folha de exercícios serão apenas analisados circuitos compostos por um diodo. Para analisar circuitos compostos por um diodo deve-se aplicar o seguinte algoritmo^[2,3]:

1. Primeiro deve identificar-se o estado do diodo:
 - a. Como a identificação do estado do diodo nem sempre é fácil, deve-se partir do pressuposto que o diodo conduz, considerando que o sentido da corrente no circuito respeita o sentido de condução do diodo (do ânodo para o cátodo).
 - b. Seguidamente deve-se efetuar o cálculo da corrente, este valor permite concluir se o diodo se encontra em condução ou não:
 - i. Se o valor da corrente for positivo, então pode-se concluir que o sentido da corrente está correto, logo o diodo encontra-se em condução.
 - ii. Se pelo contrário, o valor da corrente for negativo, então pode-se concluir que o sentido da corrente está incorreto e o diodo encontra-se inversamente polarizado. Neste caso, deve-se substituir o símbolo do diodo pelo circuito equivalente quando este se encontra inversamente polarizado e refazer os cálculos.
2. Após identificar o estado do diodo, deve-se substituí-lo pelo seu circuito equivalente e recorrer a um qualquer método de análise de circuitos por forma a obter a informação pretendida:
 - a. Forma de onda da tensão de saída do circuito.
 - b. Forma de onda da corrente no diodo.
 - c. Função transferência do circuito.

A metodologia apresentada anteriormente será utilizada na análise do seguinte circuito (Fig. 2 - retificador de ½ onda).

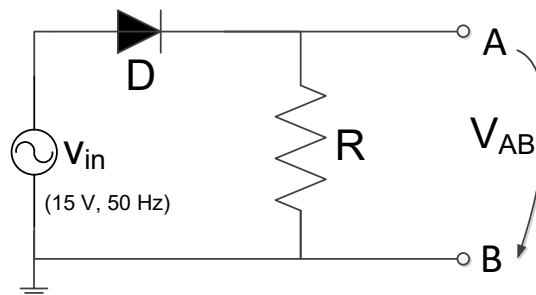


Fig. 2 – Retificador de ½ onda

De acordo com a informação apresentada na figura anterior pode-se concluir que a forma de onda da tensão de entrada (v_{in}) é sinusoidal com amplitude 15 V e frequência 50 Hz. Deste modo, como não é possível prever o valor da tensão de entrada, pois esta varia no tempo, iremos partir do pressuposto que o diodo conduz (Fig. 3).

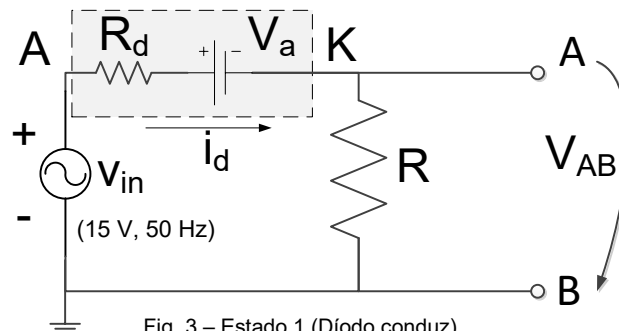


Fig. 3 – Estado 1 (Diodo conduz)

^[3] Acácio Amaral, “Análise de Circuitos e Dispositivos Eletrónicos”, Publindústria, 2013, Porto.

Aplicando o método das malhas obtém-se a seguinte equação:

$$-v_{in} + (R_d + R) \times i_d + V_a = 0$$

O diodo encontra-se em condução se $i_d > 0$, logo, utilizando a equação anterior é possível concluir que:

$$i_d = \frac{v_{in} - V_a}{(R_d + R)} > 0 \Rightarrow \boxed{v_{in} > V_a}$$

Desta forma, é possível concluir que o diodo representado no circuito da Fig. 2 conduz sempre que a tensão de entrada (v_{in}) for superior à tensão de arranque (V_a).

Seguidamente iremos determinar a relação analítica entre a tensão de saída (v_{AB}) e a tensão de entrada do circuito. Neste momento é possível obter a referida relação para o estado em que o diodo conduz (Fig. 3). Assim, utilizando a equação anterior e analisando o circuito da Fig. 3 é possível escrever:

$$\begin{cases} i_d = \frac{v_{in} - V_a}{(R_d + R)} \Rightarrow v_{AB} = \frac{R}{(R_d + R)} \times (v_{in} - V_a) \\ v_{AB} = R \times i \end{cases}$$

Relativamente ao estado 2, estado em que o diodo não conduz, é possível representar o seguinte circuito (Fig. 4):

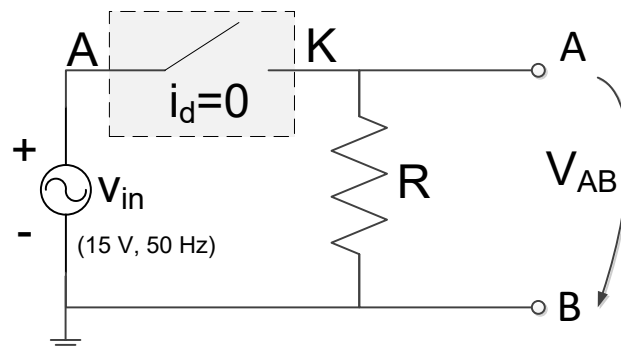


Fig. 4 – Estado 2 (Diodo não conduz)

A partir da análise do circuito da figura anterior é possível concluir que:

$$\begin{cases} i_d = 0 \\ v_{AB} = R \times i \end{cases} \Rightarrow v_{AB} = 0$$

Logo a função transferência analítica do circuito pode ser representada da seguinte forma:

$$v_{AB} = \begin{cases} \frac{R}{(R_d + R)} \times (v_{in} - V_a), & v_{in} \geq V_a \\ 0, & v_{in} \leq V_a \end{cases}$$

Para compreender o princípio de funcionamento do retificador de $\frac{1}{2}$ onda (Fig.2) é fundamental representar a forma de onda da tensão de saída, assim como, a função transferência. Para o efeito serão utilizadas duas Bibliotecas do *python*: o *numpy* e o *matplotlib*.

Relativamente à Biblioteca *numpy* esta já foi extensivamente abordada nas últimas

aulas, não sendo por isso abordada nesta ficha³.

2. Biblioteca *matplotlib*

Neste manual iremos apresentar apenas os comandos mais relevantes da Biblioteca *matplotlib* e que permitem realizar a representação gráfica de funções.

Recordamos que esta Biblioteca deve ser instalada. Para o efeito deve aceder à linha de comando do *Windows* na qualidade de administrador e executar o seguinte comando:

- `>> python -mpip install -U matplotlib`

A Biblioteca *matplotlib* é uma biblioteca composta por objetos e rotinas que permitem a criação de gráficos e a visualização de dados, sendo considerada uma extensão da Biblioteca *numpy*.

Antes de executar os comandos associados à Biblioteca *matplotlib*, na linha de comando do *IDLE*, é fundamental importar a referida Biblioteca. Deste modo, deve utilizar o seguinte comando:

- `>> from matplotlib import pyplot as plt4`

As funções *plot()* e *show()* permitem representar gráficos 2D.

- `>> plt.plot(x,y)` - em que *x* representa a função corresponde ao eixo dos X do gráfico (tipicamente o tempo) e *y* a função corresponde ao eixo dos Y do gráfico.
- `>> plt.show()` - permite a exibição do gráfico.

A função *subplot()* permite representar vários gráficos na mesma figura.

- `>> plt.subplot(x,y,pos)` - em que *x* representa o nº de gráficos relativos à posição horizontal, *y* representa o nº de gráficos relativos à posição vertical e *pos* a posição do gráfico.

As funções *xlabel()* e *ylabel()* permitem definir os títulos dos eixos do gráfico.

- `>> plt.xlabel("texto eixo do X")` – permite definir o título do eixo dos X.
- `>> plt.ylabel("texto eixo do Y")` – permite definir o título do eixo dos Y.

A função *title()* permite definir o título da figura.

- `>> plt.title("título da figura")` – permite definir o título da figura.

A função *grid()* cria uma grelha no gráfico.

- `>> plt.grid()` – cria uma grelha no gráfico.

Suponha que pretende representar a forma de onda da tensão de entrada do circuito da Fig. 2:

- Uma onda sinusoidal com amplitude 15 V e frequência (*f*) 50 Hz.

A referida forma de onda pode ser representa através da seguinte equação:

³ Se o aluno não se recordar dos seus principais comandos deve consultar a folha de exercícios nº 1.

⁴ Esta instrução permite criar um objeto do tipo *matplotlib*.

$$v_{in}(t) = 15 \times \sin(2 \times \pi \times t \times f) = 15 \times \sin(100 \times \pi \times t)$$

Como o período (P) é igual ao inverso da frequência (f) pode-se concluir que o período da referida onda é igual a:

$$P = \frac{1}{f} = \frac{1}{50} = 0.02 \text{ s}$$

Considerando que se pretende observar 100 amostras por período pode-se concluir que o período de amostragem (PA) é igual a:

$$PA = \frac{20 \text{ ms}}{100} = 0.2 \text{ ms}$$

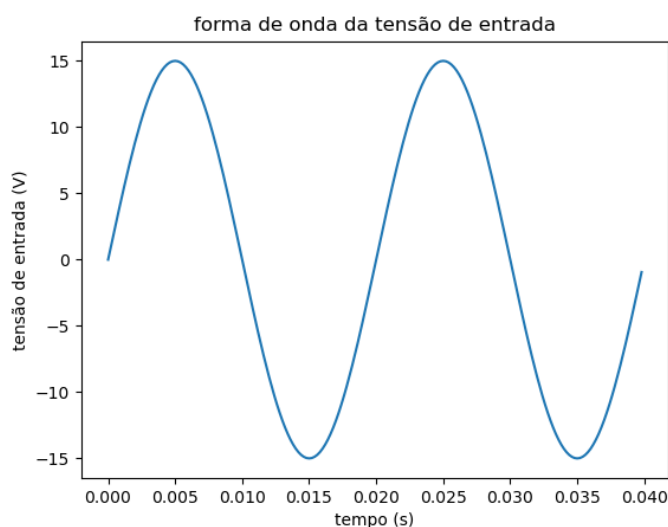
A informação anterior permite-nos construir o vetor tempo:

- Dois períodos do sinal, ou seja, com início a 0 s e fim 0.04 s.
- Período de amostragem igual a 0.0002 s.

Finalmente, utilizando a informação anterior, é possível representar a referida forma de onda com recurso ao *IDLE*. A figura seguinte mostra os comandos que devem ser utilizados para representar a forma de onda da tensão de entrada:

```
>>> import numpy as np
>>> f=50;P=1/f;
>>> t=np.arange(0,0.04,0.0002)
>>> from math import pi
>>> vin=15*np.sin(2*pi*f*t)
>>> from matplotlib import pyplot as plt
>>> plt.plot(t,vin)
[<matplotlib.lines.Line2D object at 0x000002010238FCC8>]
>>> plt.xlabel("tempo (s)")
Text(0.5, 0, 'tempo (s)')
>>> plt.ylabel("tensão de entrada (V)")
Text(0, 0.5, 'tensão de entrada (V)')
>>> plt.title("forma de onda da tensão de entrada")
Text(0.5, 1.0, 'forma de onda da tensão de entrada')
>>> plt.show()
```

Finalmente apresenta-se o resultado da execução dos comandos da Fig. 5.



3. Comandos básicos do *python*

A linguagem de programação *python* é simples, no entanto, é bastante poderosa, sendo utilizada no desenvolvimento de diferentes sistemas. Uma das suas principais características é o facto de não conter um elevado número de marcações ('.' ou ';'), marcadores ('{' e '}') ou palavras chave ('*begin*' e '*end*').

Os principais tipos de variáveis são:

- Tipo inteiro (números inteiros).
- Tipo real (número reais).
- Tipo *string* (conjunto de caracteres)

No *python* não é necessário declarar o tipo de variável, basta atribuir o valor à variável que esta automaticamente assume o tipo.

- `>> num_int=1234` – a variável *num_int* torna-se automaticamente do tipo inteiro.
- `>> num_rea=12.34` – a variável *num_rea* torna-se automaticamente do tipo real.
- `>> nome=input("indique o seu nome")` – a variável *nome* torna-se automaticamente do tipo *string*.

Os comandos *int* e *float* permitem converter os valores para os respetivos tipos.

- `>> num_int= int(input("indique a sua idade"))`
- `>> num_rea= float(input("indique a seu peso de Kg"))`

Os principais operadores aritméticos são:

- '+' – soma.
- '-' – subtração.
- '*' – multiplicação.
- '/' – divisão.
- '%' – resto da divisão.
- '**' – potência.

Os principais operadores utilizados na comparação entre valores numéricos são:

- '<' – menor.
- '<=' – menor ou igual.
- '>' – maior.
- '>=' – maior ou igual.
- '==' – igual.
- '!=' – diferente.

Os principais operadores lógicos são:

- 'not' – inversor lógico.
- 'And' – E lógico.
- 'Or' – Ou lógico.

As estruturas de decisão são:

- `if <condição>:`
 ...
• `if <condição>:`
 ...
 `else:`
 ...

- if <condição>:
 ...
elif <condição>:
 ...
else:
 ...

A figura seguinte mostra como pode utilizar as estruturas de decisão no âmbito de um programa em *python*.

```
>>> num1=1;num2=2;
>>> if num1 == num2:
    print(str(num1) + " é igual a " + str(num2))
elif num1 < num2:
    print(str(num1) + " é menor do que a " + str(num2))
else:
    print(str(num1) + " é maior do que a " + str(num2))

1 é menor do que a 2
>>> |
```

Fig. 7 – Exemplo de aplicação de estrutura de decisão em *Python*

As estruturas de repetição são:

- while <condição>:
 <bloco de comandos>
- for <variável> in range(início:fim:passo)
 <bloco de comandos>

A figura seguinte mostra como pode utilizar as estruturas de repetição *while* e *for* no âmbito de um programa em *python*.

```
>>> inicio=0;fim=4;conta=inicio;
>>> while conta < fim:
    print(conta)
    conta+=1

0
1
2
3
>>> for conta in range(0,4,1):
    print(conta)

0
1
2
3
>>>
```

Fig. 8 – Exemplo de aplicação de estrutura de repetição em *Python*

4. Simulação de um circuito com díodos em *python*

Nesta secção apresenta-se o código em *python* que permite simular o circuito da Fig. 2. Para o efeito foram utilizados os conceitos apresentados nas secções anteriores.

```
>>> import numpy as np
>>> f=50;P=1/f;
>>> PA=P/100;
>>> t=np.arange(0,P+PA,PA)
>>> from math import pi
>>> vin=15*np.sin(2*pi*f*t)
>>> vout=vin.copy()
>>> i=0;
>>> TAM=np.size(t)
>>> Va=0.7;Rd=100;R=1000;
>>> while (i<TAM):
>>>     if (vin[i]<Va):
>>>         vout[i]=0;
>>>         i+=1;
>>>     else:
>>>         vout[i]=R/(R+Rd)*(vin[i]-Va);
>>>         i+=1;

>>> from matplotlib import pyplot as plt
>>> plt.subplot(2,1,1)
<AxesSubplot:>
>>> plt.plot(t,vin)
[<matplotlib.lines.Line2D object at 0x000001C1CC8C3648>]
>>> plt.subplot(2,1,2)
<AxesSubplot:>
>>> plt.plot(t,vout)
[<matplotlib.lines.Line2D object at 0x000001C1CD616D48>]
>>> plt.show()
>>>
```

Referências Bibliográficas

- [1] Amaral, Acácio (2017), Electrónica Analógica: Princípios, Análise e Projectos, Edições Silabo, Lisboa, Portugal.
- [2] Amaral, Acácio (2021), Electrónica Aplicada, Edições Silabo, Lisboa, Portugal.
- [3] Amaral, Acácio (2015), Análise de Circuitos e Dispositivos Eletrónicos, Publindústria, Porto (2ª edição).

Exercícios

14. Considere o circuito da figura seguinte.

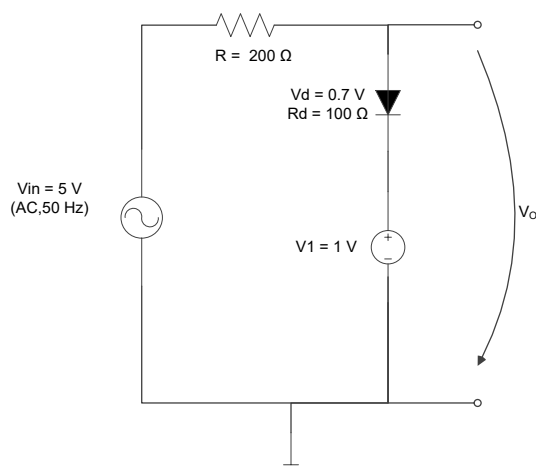


Fig. 10

- Determine para que valores de v_{in} o díodo conduz.
- Represente graficamente a característica de transferência do circuito.
- Represente a evolução temporal dos sinais v_{in} e v_o .
- Simule o circuito anterior em *Pspice*.
- Simule o circuito anterior em *python*.

15. Considere o circuito da figura seguinte ($V_d = 0.7\text{ V}$ e $R_d = 100\ \Omega$).

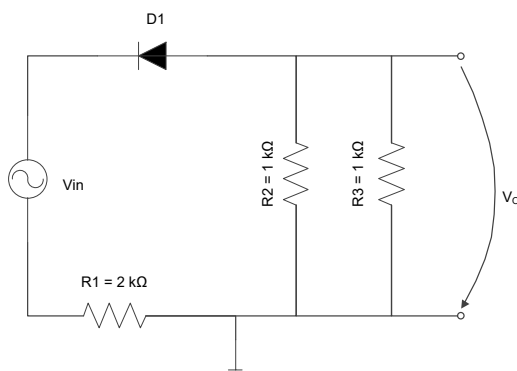
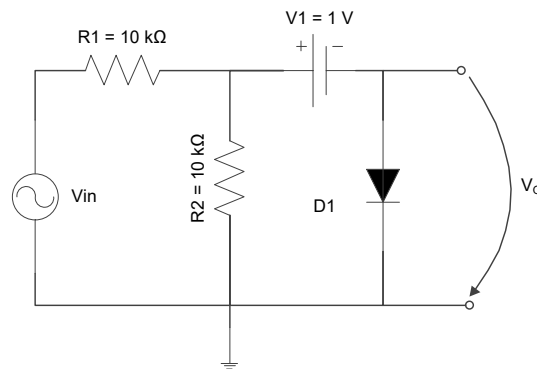


Fig. 11

- Determine os valores de v_{in} e v_o , para $i = 1\text{ mA}$.
- Represente graficamente a característica de transferência do circuito.
- Considerando que v_{in} é uma onda sinusoidal de amplitude 20 V e frequência 50 Hz , represente a evolução temporal de v_{in} e v_o .
- Simule o circuito anterior em *Pspice*.
- Simule o circuito anterior em *python*.

16. Considere o circuito da figura seguinte ($V_d=0.7\text{ V}$ e $R_d = 100\ \Omega$).



- Determine o valor de V_o , para $V_{in} = 0\text{ V}$.
- Represente graficamente a característica de transferência do circuito:
 - Utilize o teorema de *Thevenin* na análise do circuito.
 - Utilize o método das malhas na análise do circuito.
- Considerando que v_{in} é uma onda sinusoidal de amplitude 20 V e frequência 50 Hz , represente a evolução temporal de v_{in} e v_o .
- Simule o circuito anterior em *Pspice*.
- Simule o circuito anterior em *Python*.