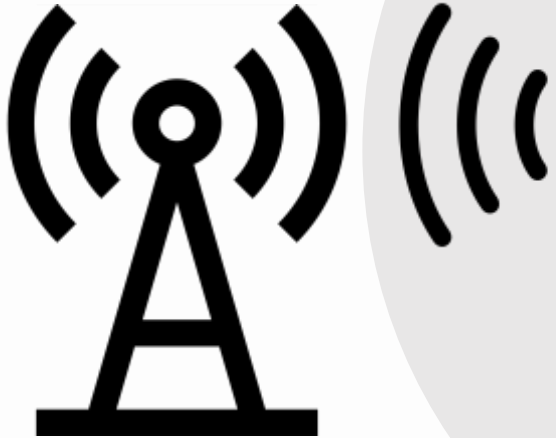
Capture d'énergie électromagnétique



Réalisé par : MOHAMMED SAFARI

Encadré par : MR FAKCHICH MIMOUN

Code SCEI: 15762 Code CNC: CA181M

Plan d'étude

01

Introduction

Présentation du contexte et des objectifs de l'étude

02

Capture d'énergie

Description du principe et du circuit utilisé

03

Choix des constituants

Sélection des composants adaptés

04

Simulation du circuit

Modélisation du circuit et analyse de son comportement

05

Polarisation et impédance

L'impact de l'orientation des antennes et de l'appariement des impédances

06

Conclusion

Synthese des résultats et perspectives sur l'exploitation de l'énergie

01

Introduction

• **Ondes RF :**
ondes électromagnétiques dont les fréquences sont comprises entre 3kHz et 300GHz

Maxwell-Faraday : $\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$

Maxwell-Ampère : $\nabla \times \vec{B} = \mu_0 \varepsilon_0 \frac{\partial \vec{E}}{\partial t}$

L'équation d'onde: $\nabla^2 \vec{E} = \mu\varepsilon \frac{\partial^2 \vec{E}}{\partial t^2}$

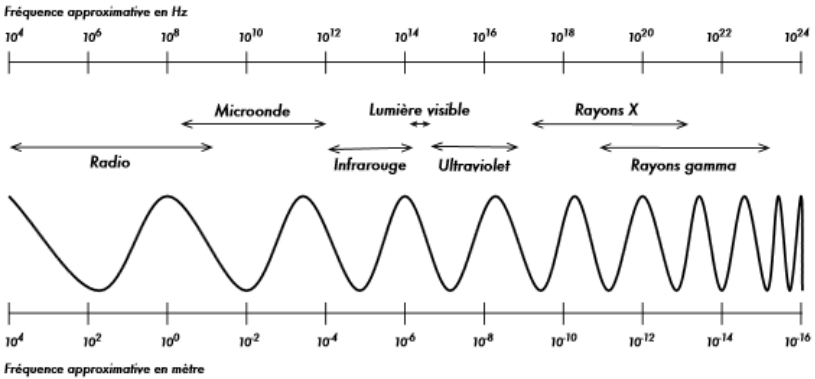


Figure 1: spectre des ondes électromagnétiques

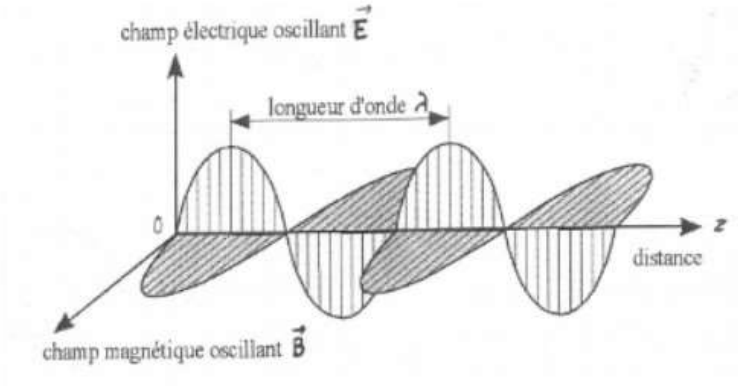
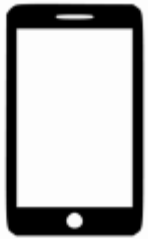


Figure 2: Représentation schématique d'une onde électromagnétique

Abondance énergitique des ondes électromagnéiques

Puissance reçue par une antenne réceptrice est donnée par la formule de Friis :

$$P_r = P_t \cdot G_t \cdot G_r \cdot \left(\frac{\lambda}{4\pi d}\right)^2$$



Ordres des grandeurs :

Constante	Réseau 4G	Réseau 5G
Puissance Pt	40 W	30 W
Gain Gt	40	100
Gain Gr	1	1
Fréquence	800MHz	3.5GHz

Figure 3: tableau des constantes prises pour la simulation

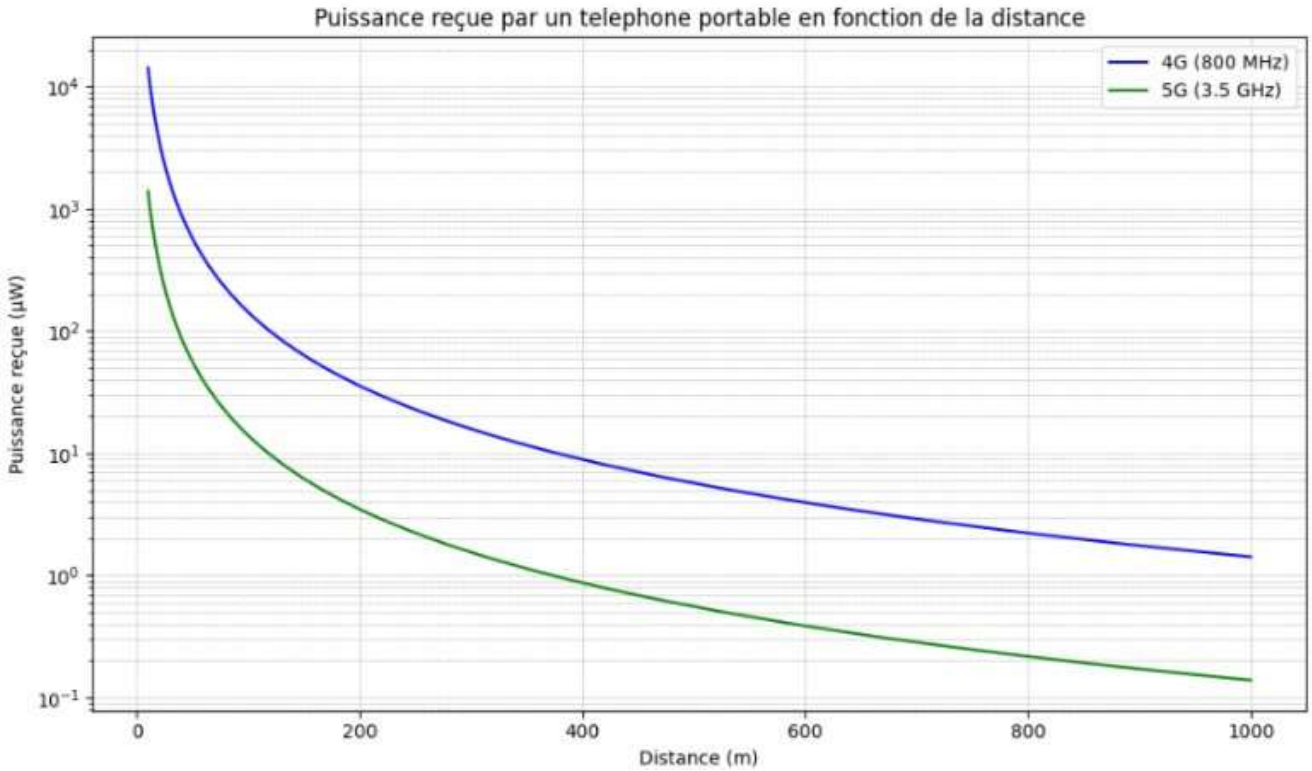


Figure 4: simulation de la puissance reçue par un téléphone portable

Technologies quasi-autonomes en énergie :

Certains appareils fonctionnent avec une très faible consommation comme:

- Capteur de qualité de l'air
- Etiquettes RFID passives
- Microcontrôleurs

Avantages de l'énergie ambiante RF

- **Energie gratuite et omnimprésente**

  L'environnement urbain est saturé des ondes électromagnétiques

  - **Autonomie énergétique**

    Fonctionnement sans sources d'énergie embarquées pendant de longues durées

    - **Maintenance réduite**

      Parfaits pour les environnements difficiles d'acces ou pour les capteurs déployés à long terme

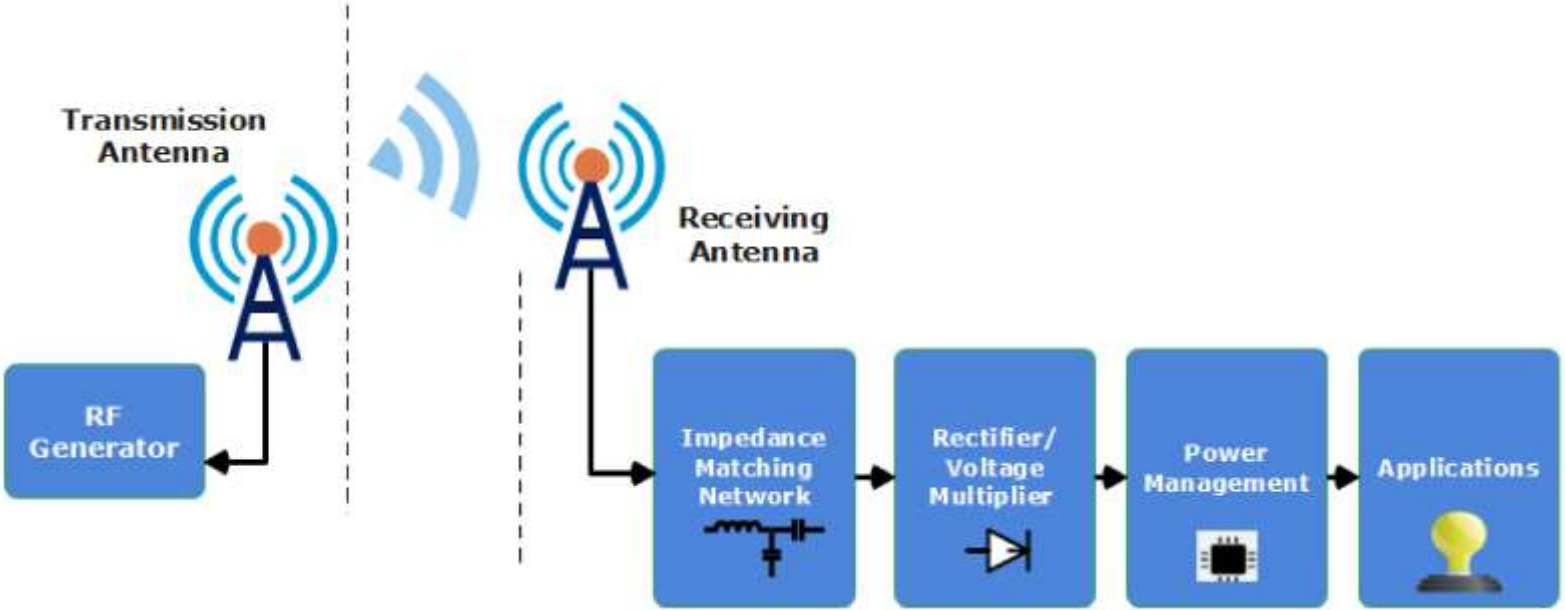
Problématique

Comment récupérer l'énergie des ondes électromagnétiques et comprendre le role des circuits pompes a charges ainsi que l'effet de l'impedance et la polarisation des antennes

02

Capture d'énergie

Le processus de capture de l'énergie électromagnétique



Facteurs influençant le capture de l'énergie électromagnétique par une antenne

Taille de l'antenne

La taille de l'antenne doit être multiple de la demi–longueur d'onde visée

Impédance de l'entrée

Elle doit être égale à celle du circuit de charge

Orientation de l'antenne

Elle doit être orientée dans la même direction que la polarisation de l'onde

Pompe à charge Dickson :

La pompe à charge Dickson permet de redresser la tension alternative reçue par l'antenne réceptrice en une tension quasi-continue, ainsi que de la multiplier en fonction de nombre des étages

Vout = N (Vin_max - Vd)

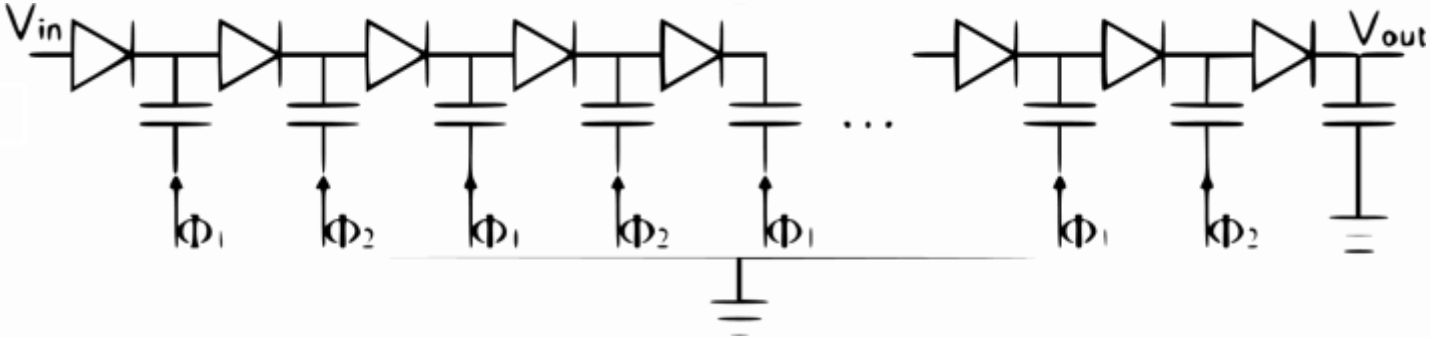
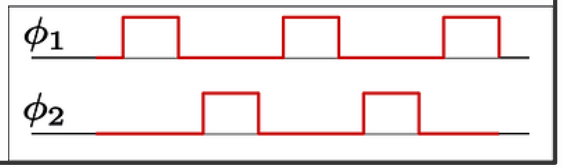


Figure 5: schéma d'une Pompe à charge Dickson à N étages



Pompe à charge Dickson

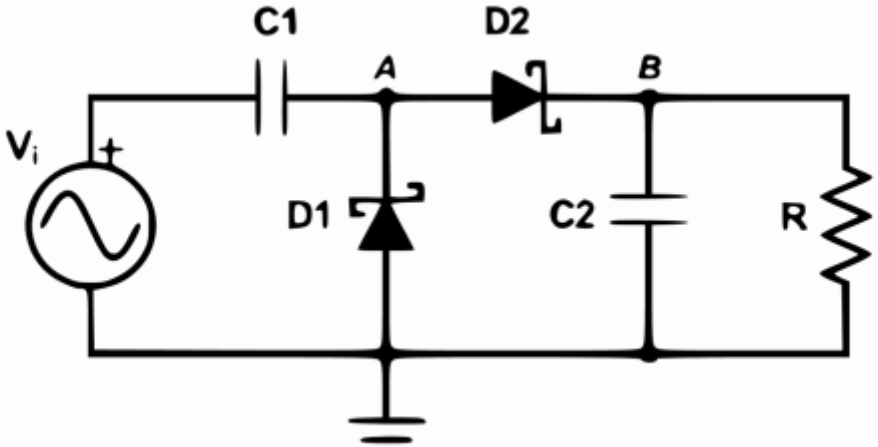


Figure 6: schéma d'une Pompe à charge Dickson simple (circuit de Greinacher)

Le circuit de Greinacher est une forme simplifie de pompe à charge Dickson qui transforme une tension alternative (AC) en une tension continue (DC) de valeur environ doublée par rapport à l'amplitude d'entrée.

Vout = 2 (Vin_max – Vd)

03

Choix des constituants

Comparaison entre les types des diodes

• Une diode est caracterisée par une tension de seuil et un temps de commutation

01	02	03
Diode ideale	**Diode de Schottky**	**Diode standard**
Temps de commutation nul	Temps de commutation faible (1-10ns)	Temps de commutation nul (2-5µs)
Vd = 0V Transmet tout le demi-signal positif sans pertes	**Vd = 0.2V** Pertes minimales, ne laisse passer que la partie du signal > 0.2V	**Vd = 0.7V** Pertes importantes, ne laisse passer que la partie du signal > 0.7V

Type de diode	Tension de seuil Vd
Diode idéale	0
Diode Schottky	0.2
Diode standard	0.7

Figure 6: tableau comparatif des types de diodes

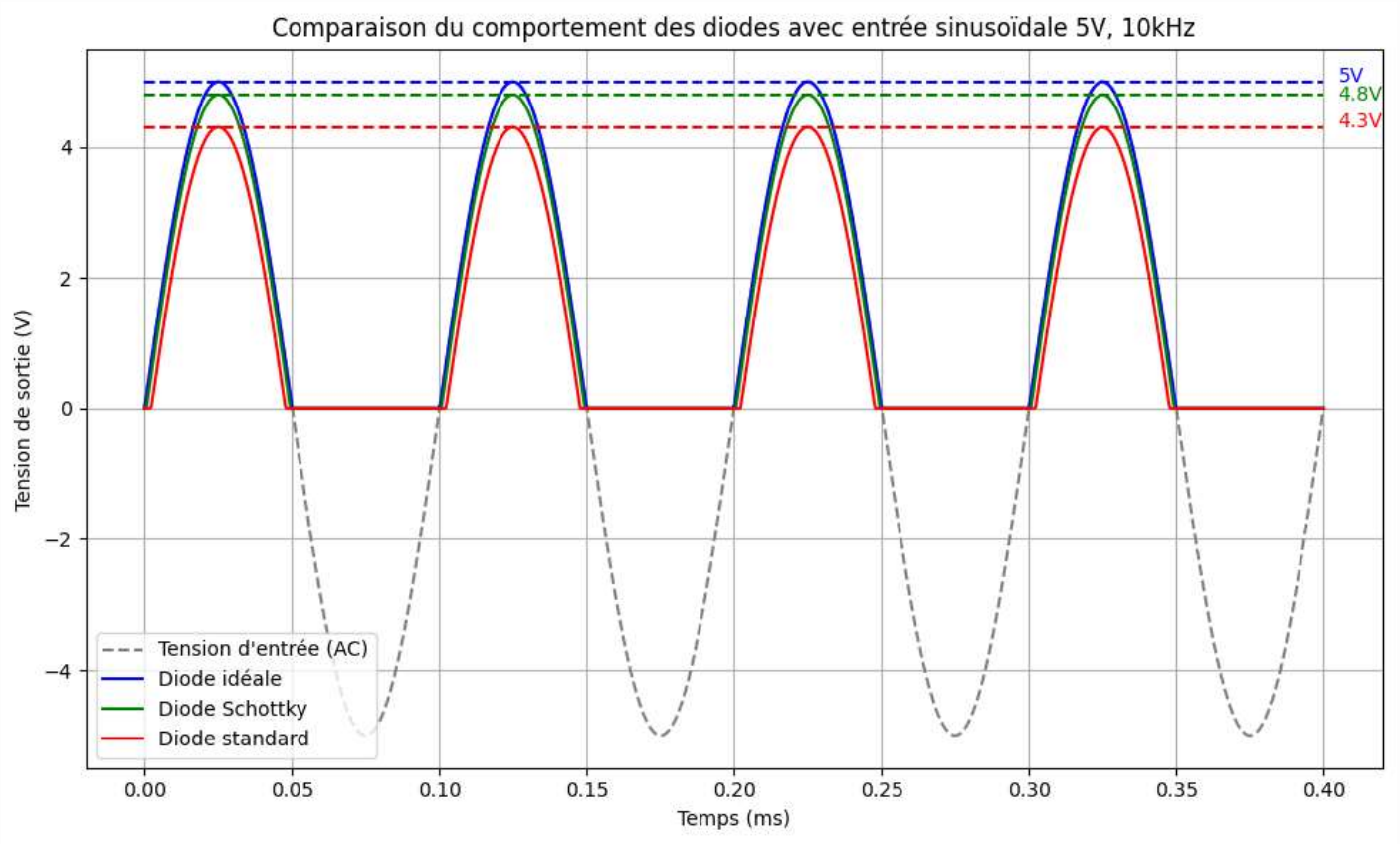


Figure 7: simulation de comportement des diodes

L'effet de la capacitance des condensateurs dans un circuit de garnacher

Capacité	Stabilité	Temps de montée
Faible (nF)	Peu stable	Rapide
Moyenne (µF)	Plus stable	Plus lent

Figure 8: tableau comparatif des capacités

04

Simulation du circuit

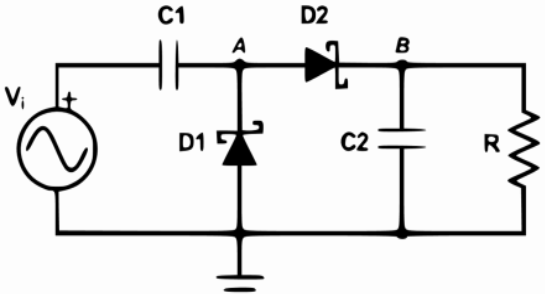


Figure 9: circuit de Greinacher

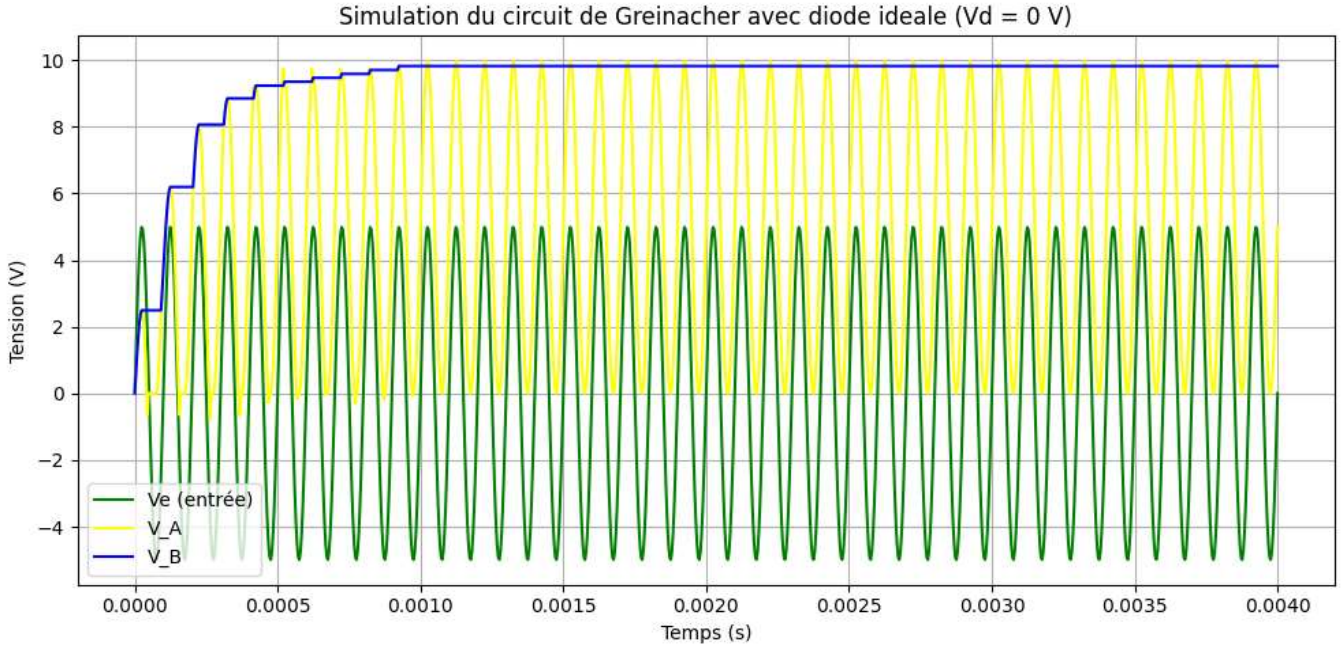


Figure 10 : la réponse du circuit de Greinacher a un signal d'entrée d'amplitude 5V et de frequence 10KHz pour des diodes ideales

La réponse du circuit de Greinacher a un signal d'entrée d'amplitude 5V et de frequence 10KHz

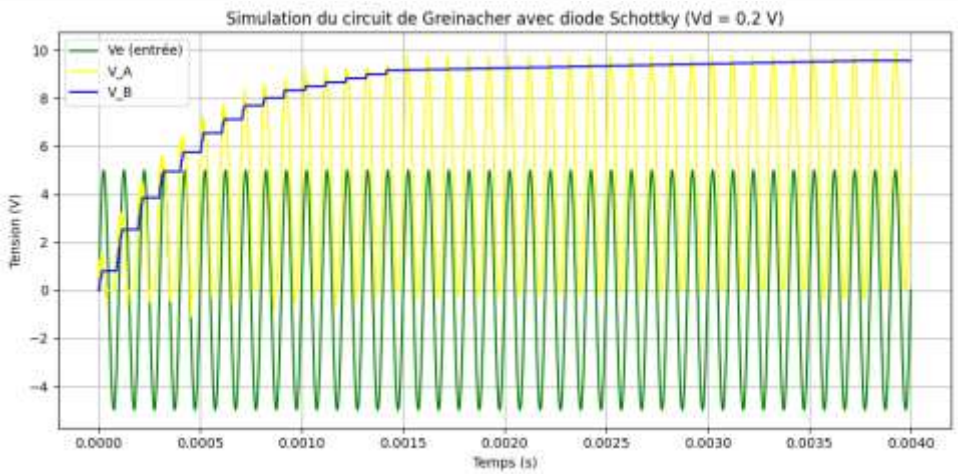


Figure 11: en utilisant des diodes Schottky
V_B = 9.6V

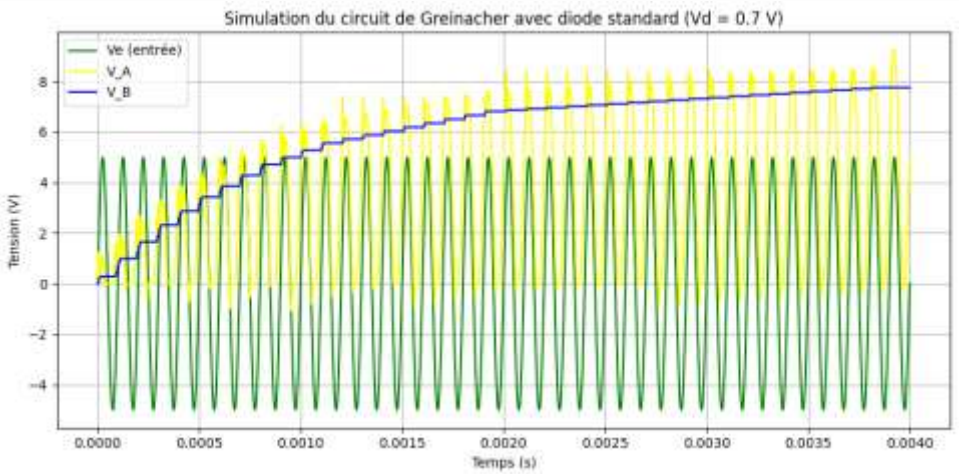
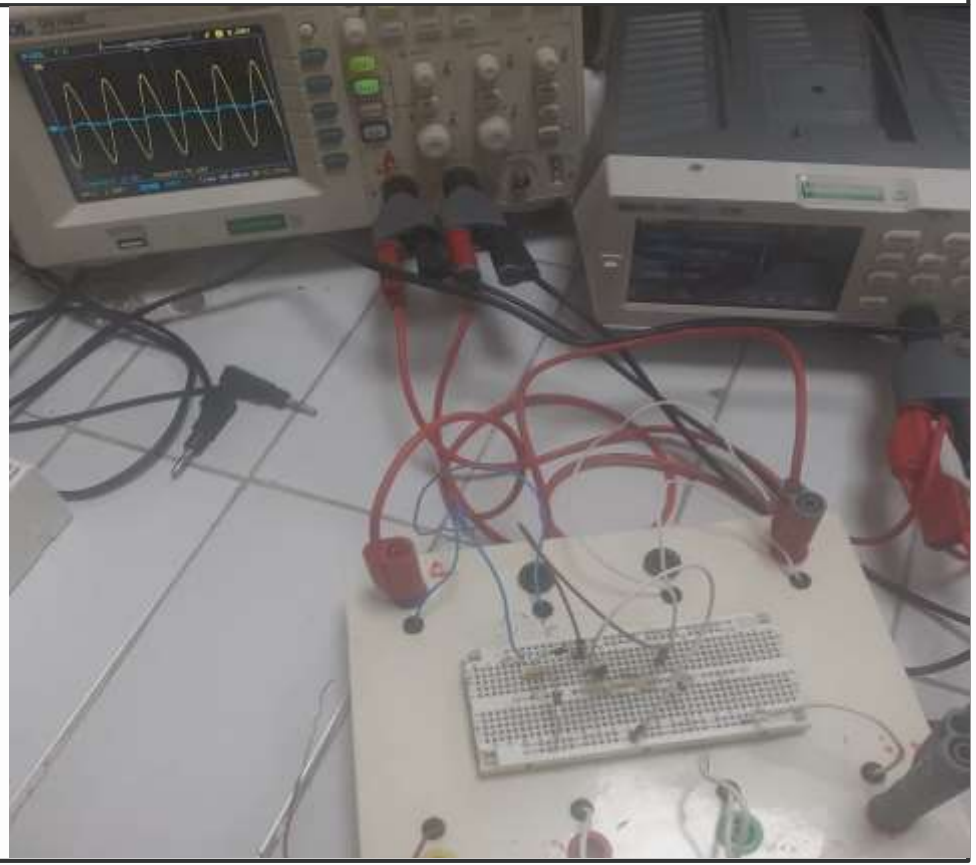


Figure 12: en utilisant des diodes standards
V_B = 8.6V

Resultat experimental de circuit de Greinacher soumis sous un signal AC d'amplitude 5V et de frequence 10KHz



Valeur mesurée de tension de sortie : 4.5V

Valeur théorique de tension de sortie : 8.6V

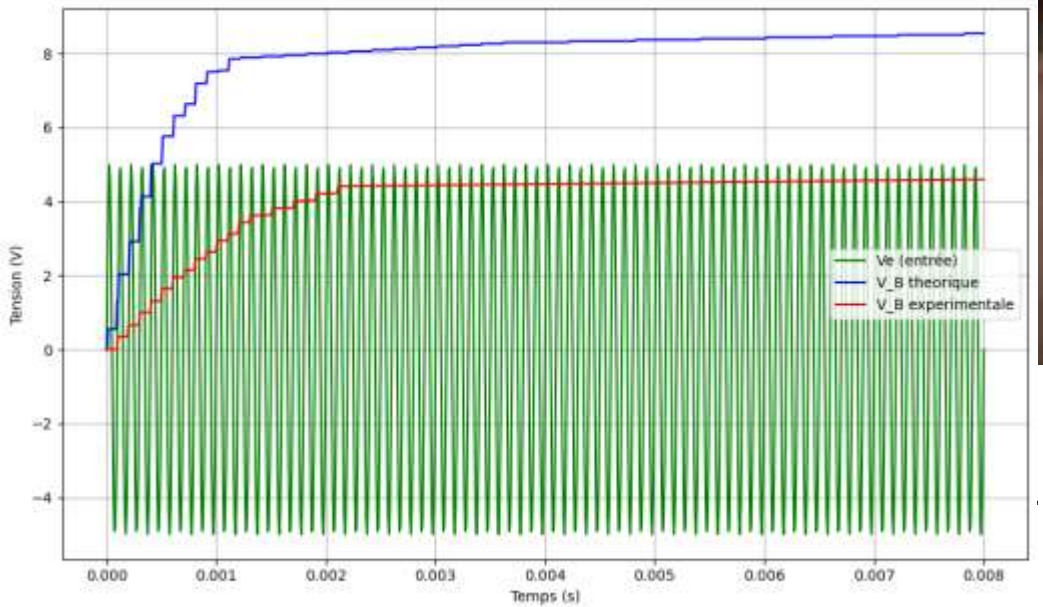


Figure 13: comparaison de la réponse du circuit de Greinacher expérimentale et théorique

05

Polarisation et impédance

Polarisation

La forme de l’antenne émettrice détermine la polarisation de l’onde EM : linéaire ou elliptique;

Son orientation impose la direction du champ électrique de l’onde qu’elle génère

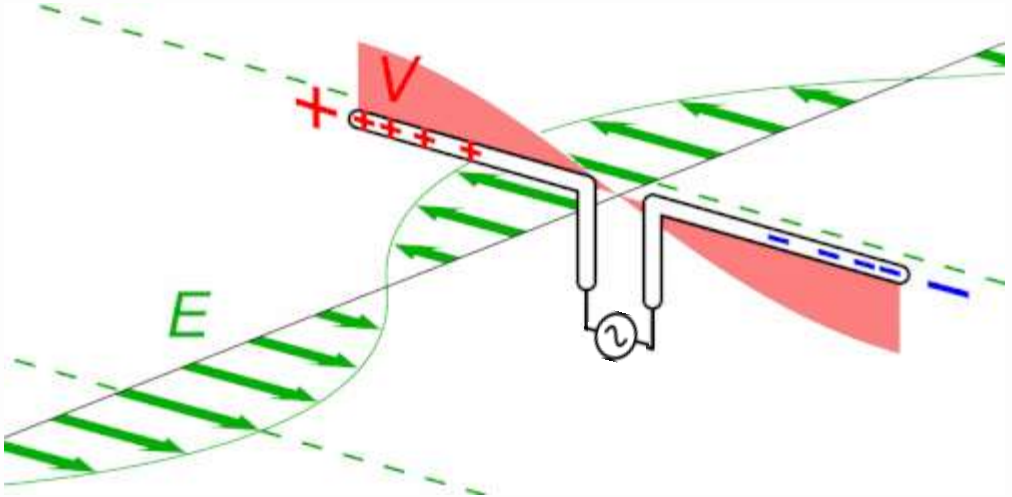
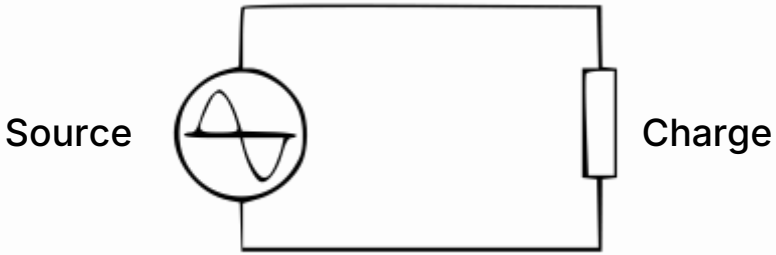


Figure 14: schéma d'une antenne dipôle demi-onde émettant une onde radio

	Réception maximale Vr = Ve
	Tension captée proportionnelle au cosinus de l'angle Vr ∝ cosα
	Rien n'est capté Vr = 0

Figure 15: tableau qui montre l'influence de l'orientation de l'antenne réceptrice par rapport à l'antenne émettrice sur la réception du signal

L'appariement d'impédance



Formule du coefficient de réflexion : $$\Gamma = \frac{Z_{\text{charge}} - Z_{\text{source}}}{Z_{\text{charge}} + Z_{\text{source}}}$$

Formule de la puissance transmise : $$P = P_{\text{max}} \cdot (1 - |\Gamma|^2)$$

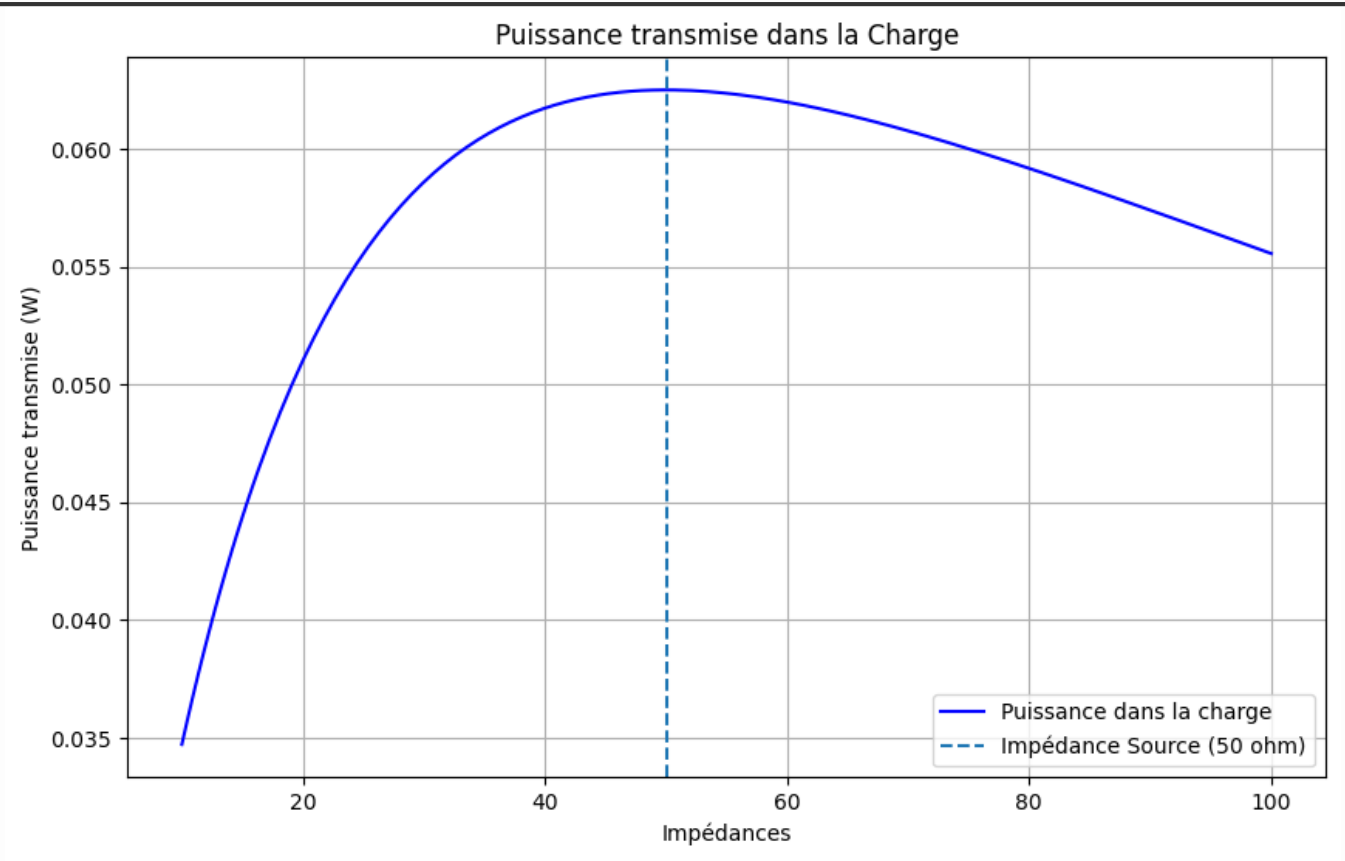


Figure 16: simulation de la puissance transmise à la charge en fonction de l'impédance de charge

06

Conclusion

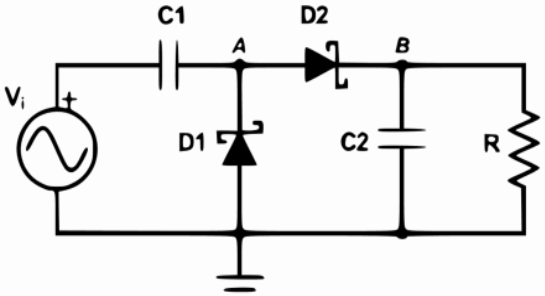
Annexes

Code python de la figure 4:

```python
import numpy as np
import matplotlib.pyplot as plt

Pt_4g = 40  # puissance emise pour 4G
Gt_4g = 40  # gain émetteur 4G
Gr_4g = 1   # gain récepteur 4G
lambda_4g = 0.375  # longueur d'onde 4G (800 MHz)

Pt_5g = 30  # puissance émise pour 5G
Gt_5g = 100  # gain émetteur 5G
Gr_5g = 1    # gain récepteur 5G
lambda_5g = 0.0857  # longueur d'onde 5G (3.5 GHz)

d = np.linspace(10, 1000, 1000)

def friis(Pt, Gt, Gr, lambd, d):
    return Pt * Gt * Gr * (lambd / (4 * np.pi * d))**2

Pr_4g = friis(Pt_4g, Gt_4g, Gr_4g, lambda_4g, d) * 1e6
Pr_5g = friis(Pt_5g, Gt_5g, Gr_5g, lambda_5g, d) * 1e6

plt.figure(figsize=(10, 6))
plt.plot(d, Pr_4g, label="4G (800 MHz)", color='blue')
plt.plot(d, Pr_5g, label="5G (3.5 GHz)", color='green')
plt.yscale("log")
plt.xlabel("Distance (m)")
plt.ylabel("Puissance reçue (µW)")
plt.title("Puissance reçue par un telephone portable en fonction de la distance")
plt.grid(True, which="both", ls="--", lw=0.5)
plt.legend()
plt.tight_layout()
plt.show()
```

Annexes

Fonctionnement de circuit greinacher figure 6:



Etat	Caractérisation	Condition
Etat 1: diode 1 passante et diode 2 bloquante	• $V_A = 0\,\mathrm{V}$ • $i_2 = C \cdot \frac{dV_B}{dt} = 0$ $\Rightarrow \quad V_B = \text{constante}$	• $i_1 > 0$ $\Rightarrow \quad C \cdot \frac{d(V_A - V_e)}{dt} < 0$ $\Rightarrow \quad \frac{dV_e}{dt} < 0$
Etat 2: diode 1 bloquante et diode 2 passante	• $V_A = V_B$ • $i_2 = C \cdot \frac{dV_B}{dt} = C \cdot \frac{d(V_e - V_A)}{dt}$ $\Rightarrow \quad \frac{dV_B}{dt} = \frac{1}{2} \cdot \frac{dV_e}{dt}$	• $i_2 > 0$ $\Rightarrow \quad \frac{dV_B}{dt} > 0$ $\Rightarrow \quad \frac{dV_e}{dt} > 0$
Etat 3: diode 1 bloquante et diode 2 bloquante	• $i_1 = i_2 = 0$ $\Rightarrow \quad \frac{d(V_e - V_A)}{dt} = \frac{dV_B}{dt} = 0$	• $V_A > 0 \quad \text{et} \quad V_B < V_A$

Annexes

Code python de la figure 7:

```python
import numpy as np
import matplotlib.pyplot as plt

f = 10e3  # fréquence en Hz
T = 1 / f
V0 = 5  # amplitude crête
N = 1000  # points
t = np.linspace(0, 4*T, N)
Ve = V0 * np.sin(2 * np.pi * f * t)

def redressement(Ve, seuil):
    Vs = []
    for v in Ve:
        if v >= seuil:
            Vs.append(v - seuil)
        else:
            Vs.append(0)
    return np.array(Vs)

Vs_ideale = redressement(Ve, seuil=0.0)
Vs_schottky = redressement(Ve, seuil=0.2)
Vs_standard = redressement(Ve, seuil=0.7)
```

```python
plt.figure(figsize=(10, 6))
plt.plot(t * 1000, Ve, label="Tension d'entrée (AC)", color='gray', linestyle='--')
plt.plot(t * 1000, Vs_ideale, label="Diode idéale", color='blue')
plt.plot(t * 1000, [V0 for i in range(1000)], color='blue', linestyle='--')
plt.plot(t * 1000, Vs_schottky, label="Diode Schottky", color='green')
plt.plot(t * 1000, [V0-0.2 for i in range(1000)], color='green', linestyle='--')
plt.plot(t * 1000, Vs_standard, label="Diode standard", color='red')
plt.plot(t * 1000, [V0-0.7 for i in range(1000)], color='red', linestyle='--')

plt.text(0.405,V0,'5V',color='blue')
plt.text(0.405,V0-0.3,'4.8V',color='green')
plt.text(0.405,V0-0.7,'4.3V',color='red')

plt.title("Comparaison du comportement des diodes avec entrée sinusoïdale 5V, 10kHz")
plt.xlabel("Temps (ms)")
plt.ylabel("Tension de sortie (V)")
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```

Annexes

Code python de la figure 10:

```python
import numpy as np
import matplotlib.pyplot as plt

# Paramètres du signal
T = 2*10e-4           # durée totale (s)
f = 10e3        # fréquence (Hz)
V0 = 5          # amplitude
N = 1000          # nombre d'échantillons
pas = T / N
Temps = np.linspace(0, T, N+1)

# Signal d'entrée
Ve = [V0 * np.sin(2 * np.pi * f * t) for t in Temps]

V_A = [0]
V_B = [0]

# Fonctions de comportement des diodes
def etat1(V_A, V_B, t):
    V_B.append(V_B[t])
    V_A.append(0)

def etat2(V_A, V_B, Ve, t):
    V_B.append(V_B[t] + 0.5 * (Ve[t+1] - Ve[t]))
    V_A.append(V_B[t+1])

def etat3(V_A, V_B, Ve, t):
    V_B.append(V_B[t])
    V_A.append(V_A[t] + (Ve[t+1] - Ve[t]))
```

```python
for t in range(len(Temps) - 1):
    if Ve[t+1] < Ve[t] and V_A[t] <= 0:
        etat1(V_A, V_B, t)
    elif Ve[t+1] > Ve[t] and V_A[t] >= V_B[t]:
        etat2(V_A, V_B, Ve, t)
    else:
        etat3(V_A, V_B, Ve, t)


plt.figure(figsize=(10, 5))
plt.plot(Temps, Ve, label='Ve (entrée)', color='green')
plt.plot(Temps, V_A, label='V_A', color='yellow')
plt.plot(Temps, V_B, label='V_B', color='blue')
plt.xlabel("Temps (s)")
plt.ylabel("Tension (V)")
plt.title("Simulation du circuit de Greinacher")
plt.legend()
plt.grid()
plt.show()
```

Annexes

Code python des figures 11 et 12:

```python
import numpy as np
import matplotlib.pyplot as plt

# paramètres du signal
T = 8 * 10e-4        # durée totale (s)
f = 10e3             # fréquence (Hz)
V0 = 5               # amplitude du signal
N = 1000             # nombre d'échantillons
pas = T / N
Temps = np.linspace(0, T, N+1)

# signal d'entrée sinusoïdal
Ve = [V0 * np.sin(2 * np.pi * f * t) for t in Temps]

# tension de seuil de la diode
V_seuil = 0.2  # pour diode shcottky et 0.7 pour diode standard

V_A = [0]
V_B = [0]

def etat1(V_A, V_B, t):
    V_B.append(V_B[t])
    V_A.append(0)

def etat2(V_A, V_B, Ve, t):
    dV = Ve[t+1] - Ve[t]
    if dV > V_seuil:
        V_B.append(V_B[t] + 0.5 * (dV - V_seuil))
        V_A.append(V_B[-1])
    else:
        V_B.append(V_B[t])
        V_A.append(V_A[t])

def etat3(V_A, V_B, Ve, t):
    dV = Ve[t+1] - Ve[t]
    V_B.append(V_B[t])
    V_A.append(V_A[t] + dV)

for t in range(len(Temps) - 1):
    if Ve[t+1] < Ve[t] and V_A[t] <= 0:
        etat1(V_A, V_B, t)
    elif Ve[t+1] > Ve[t] and V_A[t] >= V_B[t] + V_seuil:
        etat2(V_A, V_B, Ve, t)
    else:
        etat3(V_A, V_B, Ve, t)

plt.figure(figsize=(10, 5))
plt.plot(Temps, Ve, label='Ve (entrée)', color='green')
plt.plot(Temps, V_A, label='V_A', color='yellow')
plt.plot(Temps, V_B, label='V_B', color='blue')
plt.xlabel("Temps (s)")
plt.ylabel("Tension (V)")
plt.title("Simulation du circuit de Greinacher avec diode Schottky (Vd = 0.2 V)")
plt.legend()
plt.grid()
plt.tight_layout()
plt.show()
```

Annexes

Code python de la figure 13:

```python
import numpy as np
import matplotlib.pyplot as plt

# Paramètres du signal
T = 8 * 10e-4          # durée totale (s)
f = 10e3               # fréquence (Hz)
V0 = 5                 # amplitude du signal
N = 1000               # nombre d'échantillons
Temps = np.linspace(0, T, N+1)

# Signal d'entrée sinusoïdal
Ve = [V0 * np.sin(2 * np.pi * f * t) for t in Temps]

def simuler_greinacher(V_seuil):
    V_A = [0]
    V_B = [0]

    def etat1(t):
        V_B.append(V_B[t])
        V_A.append(0)

    def etat2(t):
        dV = Ve[t+1] - Ve[t]
        if dV > V_seuil:
            V_B.append(V_B[t] + 0.5 * (dV - V_seuil))
            V_A.append(V_B[-1])
        else:
            V_B.append(V_B[t])
            V_A.append(V_A[t])

    def etat3(t):
        dV = Ve[t+1] - Ve[t]
        V_B.append(V_B[t])
        V_A.append(V_A[t] + dV)

    for t in range(len(Temps) - 1):
        if Ve[t+1] < Ve[t] and V_A[t] <= 0:
            etat1(t)
        elif Ve[t+1] > Ve[t] and V_A[t] >= V_B[t] + V_seuil:
            etat2(t)
        else:
            etat3(t)

    return V_A, V_B

V_A1, V_B1 = simuler_greinacher(0.7)
V_A2, V_B2 = simuler_greinacher(1.8)

plt.figure(figsize=(10, 6))
plt.plot(Temps, Ve, label='Ve (entrée)', color='green')
plt.plot(Temps, V_B1, label='V_B theorique', color='blue')
plt.plot(Temps, V_B2, label='V_B experimentale', color='red')
plt.xlabel("Temps (s)")
plt.ylabel("Tension (V)")
plt.legend()
plt.grid()
plt.tight_layout()
plt.show()
```

Annexes

Code python de la figure 16:

```python
import numpy as np
import matplotlib.pyplot as plt

def calculer_puissance(V0, Z_source, Z_charge):

    V_eff = V0 / np.sqrt(2) # tension efficace à partir de la crête
    I_eff = V_eff / (Z_source + Z_charge)
    P_charge = (I_eff**2) * Z_charge # puissance dissipée dans la charge
    return P_charge


V0 = 5.0
Z_source = 50  # impédance de la source

Z_charge_values = np.linspace(10, 100, 200)  # impedances de charge à tester

puissance_values = []
for Z_charge in Z_charge_values:
    P = calculer_puissance(V0, Z_source, Z_charge)
    puissance_values.append(P)

max_p = max(puissance_values)
max_p_index = np.argmax(puissance_values)
Z_charge_max = Z_charge_values[max_p_index]
```

```python
plt.figure(figsize=(10, 6))
plt.plot(Z_charge_values, puissance_values, linestyle='-', color='b', label='Puissance dans la charge')

plt.axvline(x=Z_source, linestyle='--', color='r', label=f'Z_source = {Z_source} ohm')

plt.title('Puissance transmise à la charge en fonction de Z_charge')
plt.xlabel('Z_charge (ohm)')
plt.ylabel('Puissance dissipée (W)')
plt.grid()
plt.legend()
plt.tight_layout()
plt.show()
```