Haut voltage dans les procédés agroalimentaires

Taha BAANTAR

SCEI : 31653

Session : 2025

Plan

I. Motivation et Problématique

II. Exigences et proposition de générateur

III. Calcul numérique , Simulation Python

IV. Analyse expérimentale

I - Motivation

Figure 1: Machine PEF (Pulsed electric field)

Figure 2: Pomme de terres entrant un champ de haute tension

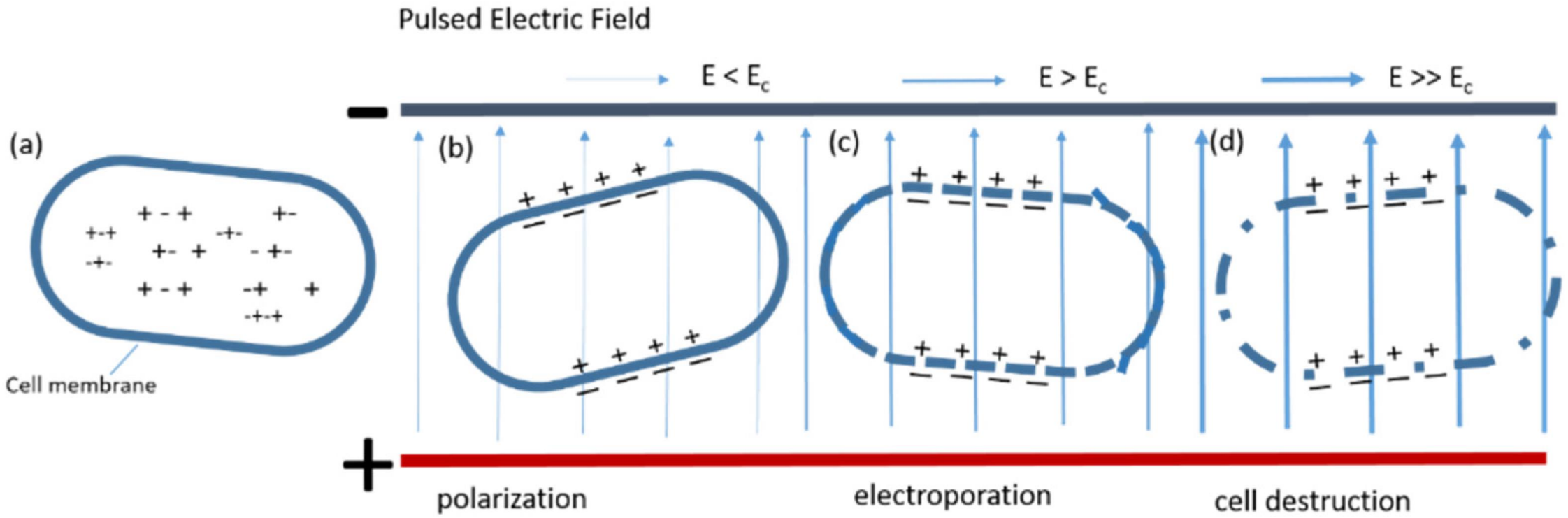


Figure 3 : Déactivation des cellules bacteries par un champ électrique

Ce procédure nécessite un champ d'intensité 20-80 kV/cm !

- **Problématique**

Comment pouvons-nous produire une haute tension avec un coût minimal et une efficacité élevée pour fabriquer une machine PEF ?

II – Exigences

- **Le générateur doit être flexible. (Selon le type d'aliment a traiter)**
- **Les composants doivent être abordables.**
- **Le générateur doit être capable de générer une haute tension sans panne. (Ordre de kV)**

Proposition de circuit

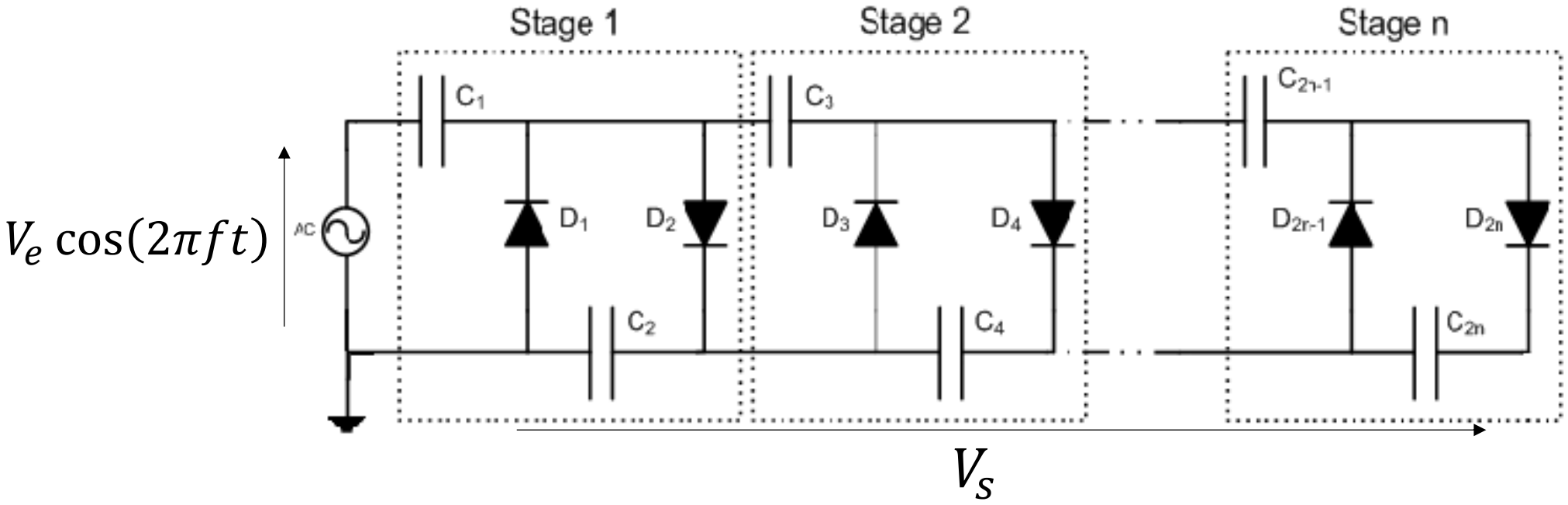


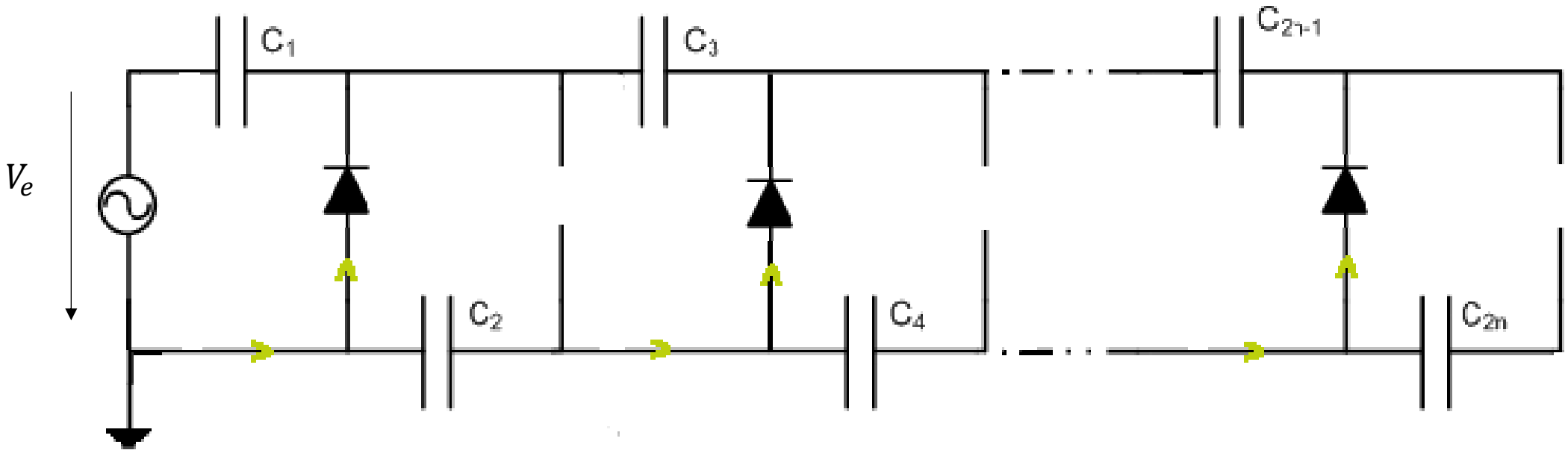
Figure 4 : Générateur Cockcroft-Walton a n étages

$$V_s = 2nV_e - 2nV_d$$

• V_d: *Tension de seuil de diode*

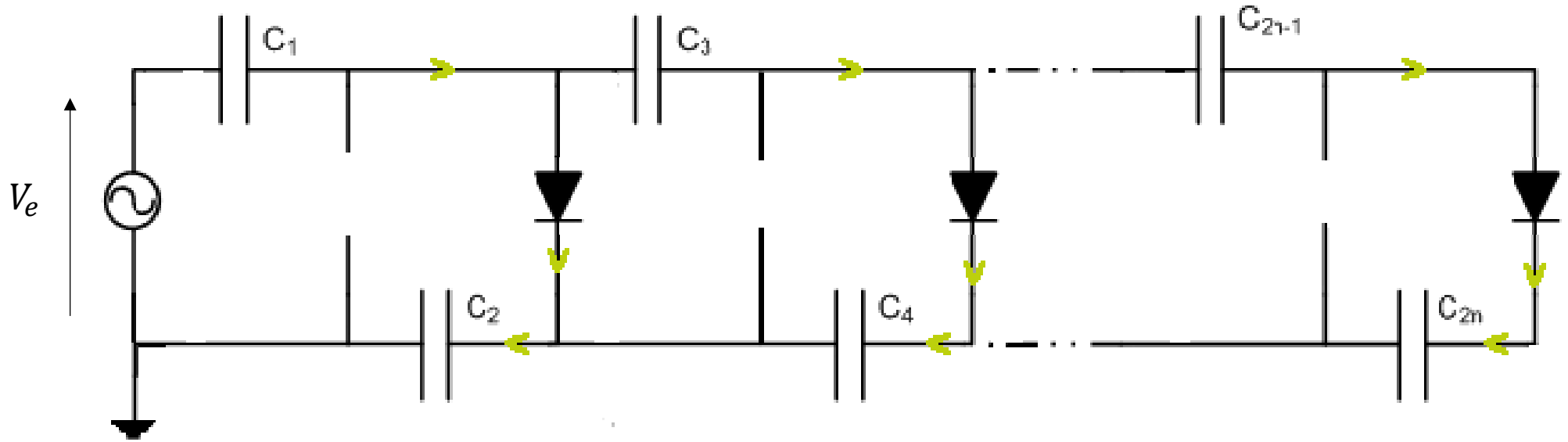
Principe de fonctionnement

Premiere demi-cycle:



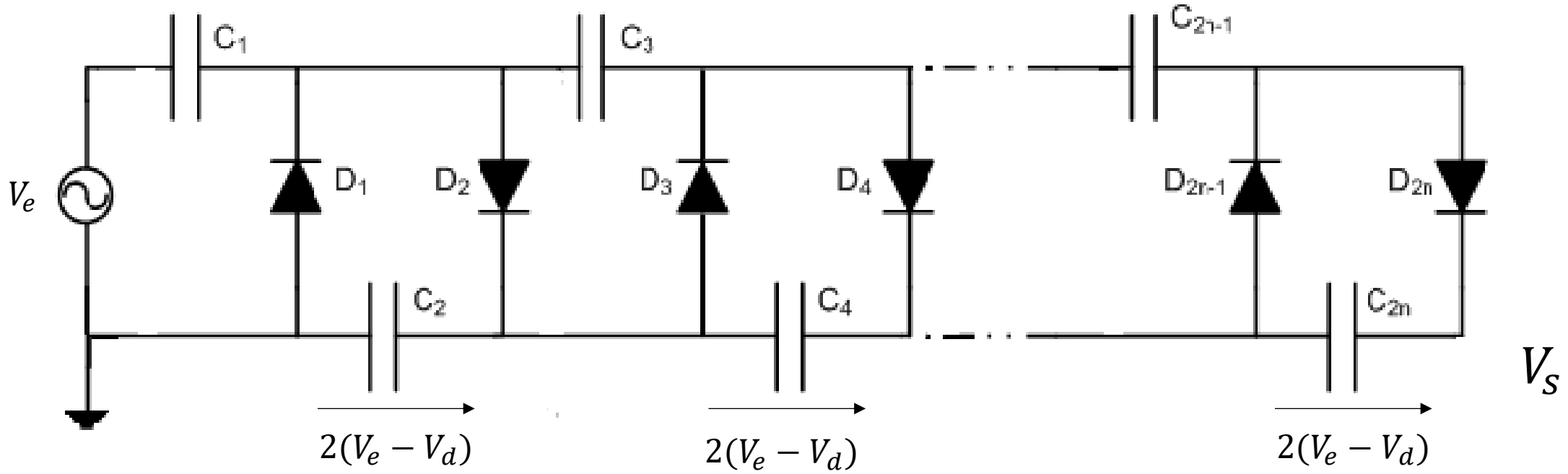
- **Le condensateur C1 se charge par le générateur et les autres se chargent par les condensateurs au dessous**

Deuxiéme demi-cycle:



- **Chaque condensateur en haut se décharge sur les condensateurs en bas**

- Ce cycle se répète jusqu'à ce que l'équilibre soit atteint dans cette situation



Les condensateurs en séries nous donnons:

$$V_s = 2nV_e - 2nV_d$$

- **Pour 1 étage**

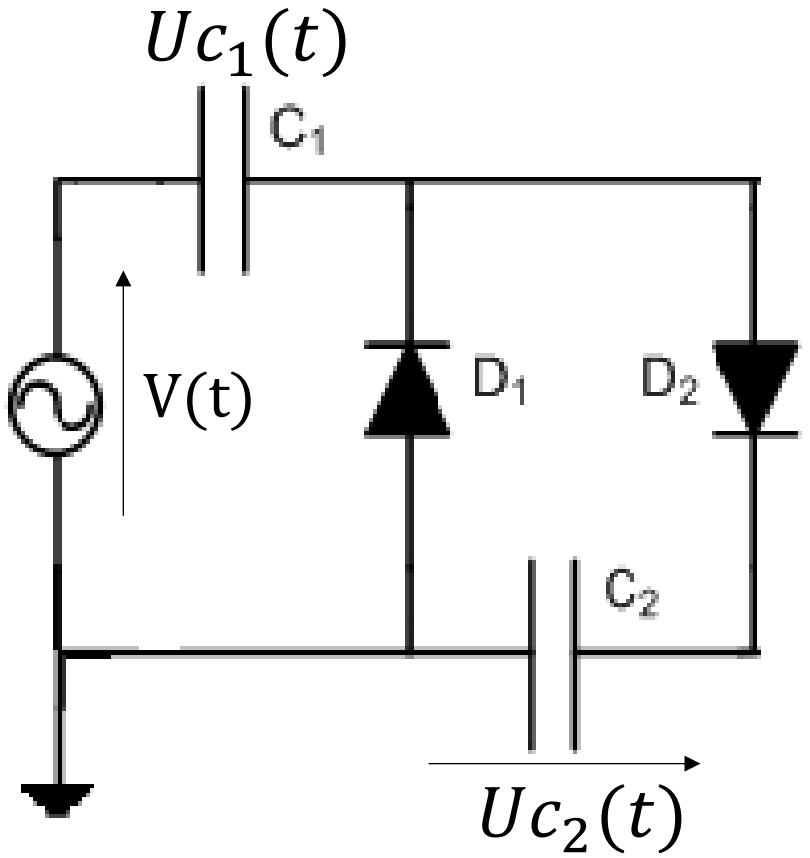
En approximant que V(t) = $\pm V_e$ qui varie a chaque dt

Avec la loi de Kirchhoff :

$$\mathrm{V}_e(t) = \begin{cases} Uc_1(t) + V_d & \text{Si } V_e(t) < 0 \\ Uc_1(t) - Uc_2(t) + V_d & \text{Sinon} \end{cases}$$

A chaque demi-cycle , on a les condensateurs sont en serie , donc :

$$\frac{dUc_1}{dt} = -\frac{dUc_2}{dt}$$



- V_d: *Tension de seuil de diode*

En ajustant les tensions en des suites récurrents pour chaque demi-cycle:

$$\begin{cases} Uc_1(t) = Uc_1(n) \\ Uc_2(t) = Uc_2(n) \end{cases}$$

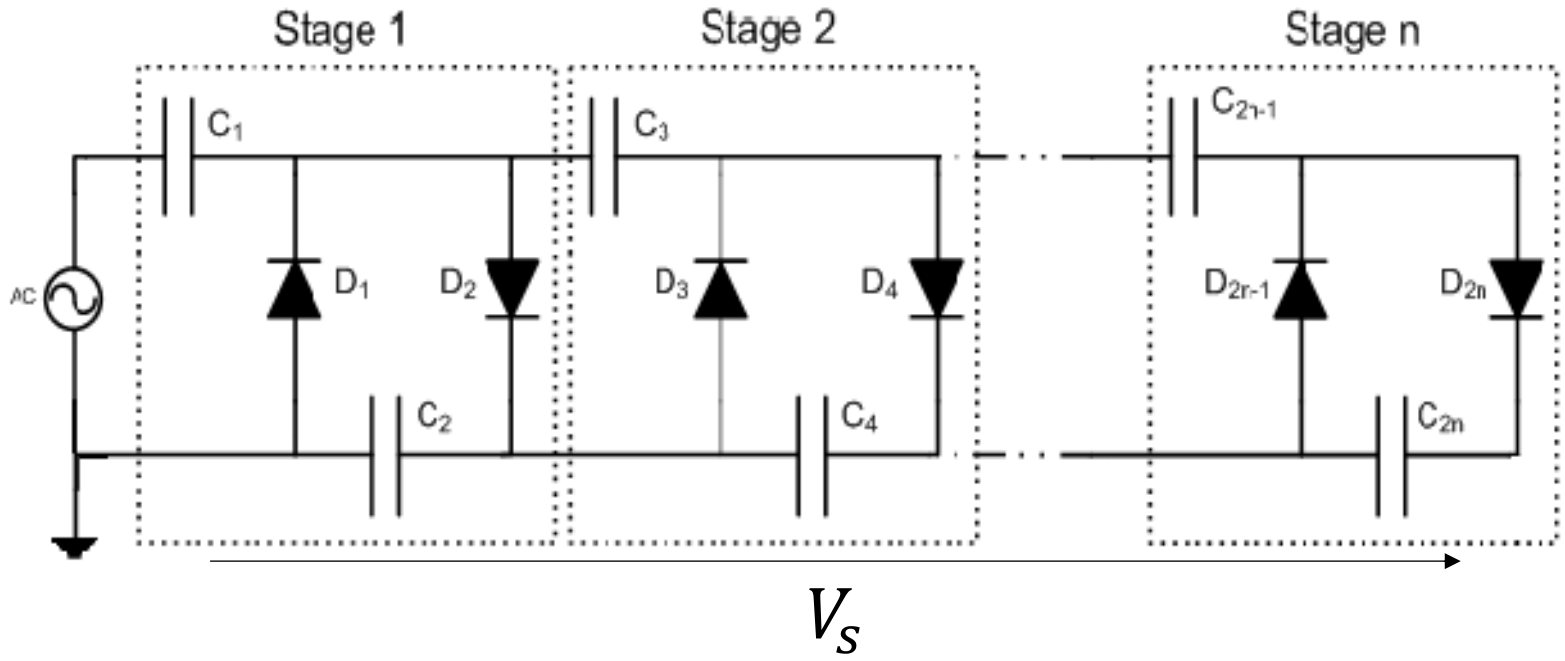
Nous trouvons:

$$Uc_2(n+2) = \frac{Uc_2(n)}{2} + V_e - V_d \qquad \textbf{Avec} \qquad Uc_2(0) = 0$$

Alors cette suite converge vers:

$$V_s = \lim_{n \to \infty} U_{c2}(n) = 2(V_e - V_d)$$

- **Pour n étages**



- **Le circuit est une simple cascade du circuit précédent**

- **En utilisant la loi de kirchhoff a chaque étage , Nous trouvons :**

$$\text{Si } V(t) < 0 : \begin{cases} V_e = Uc_1 + V_d \\ \forall i \in [2, n], \quad Uc_i = V_d + Uc_{i+1} \end{cases}$$

$$\text{Sinon} : \begin{cases} V_e + Uc_1 = Uc_2 + V_d \\ \forall i \in [3, n+1], \quad Uc_i = V_d + Uc_{i+1} \end{cases}$$

- **Difficile a calculer par main , Nous utilisons un calcul numérique !**

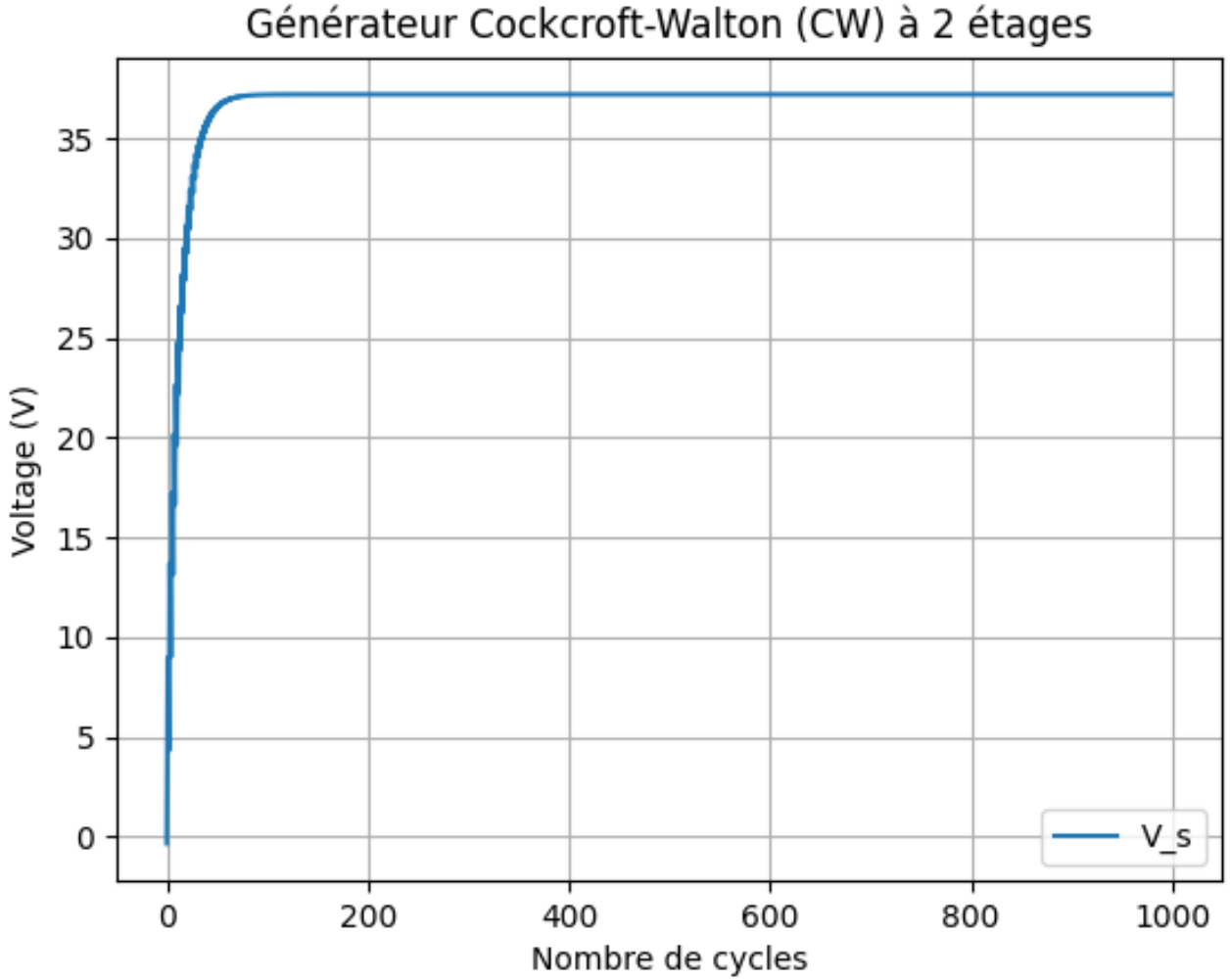
III – Calcul numérique

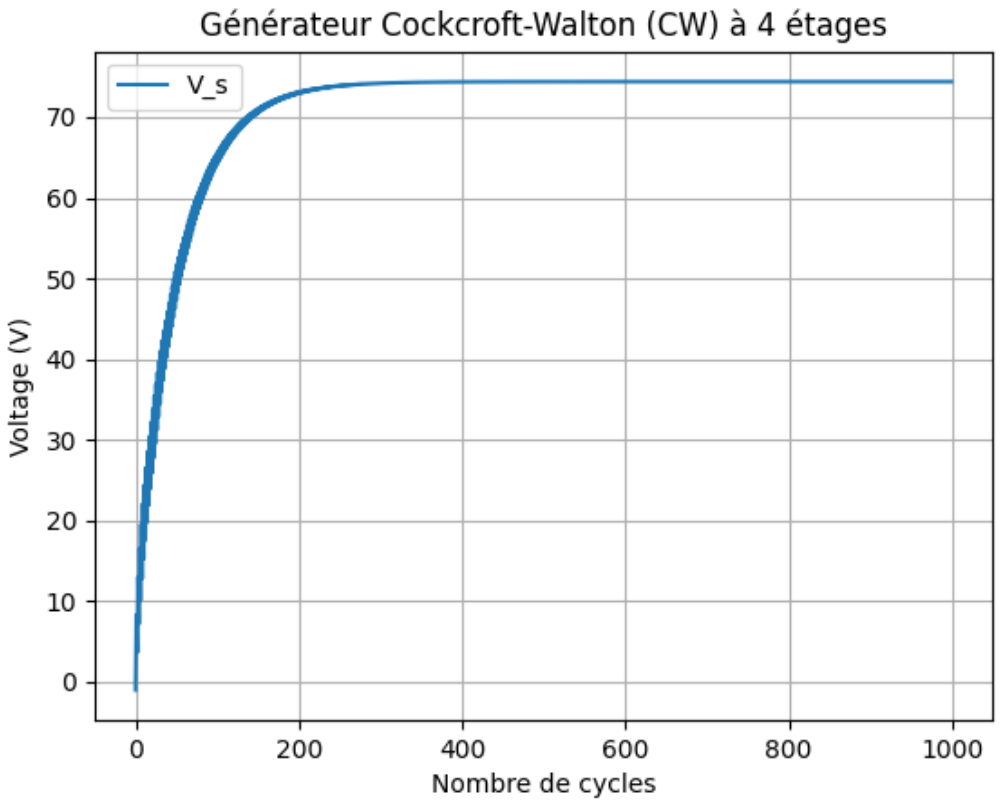
On prend pour valeurs :

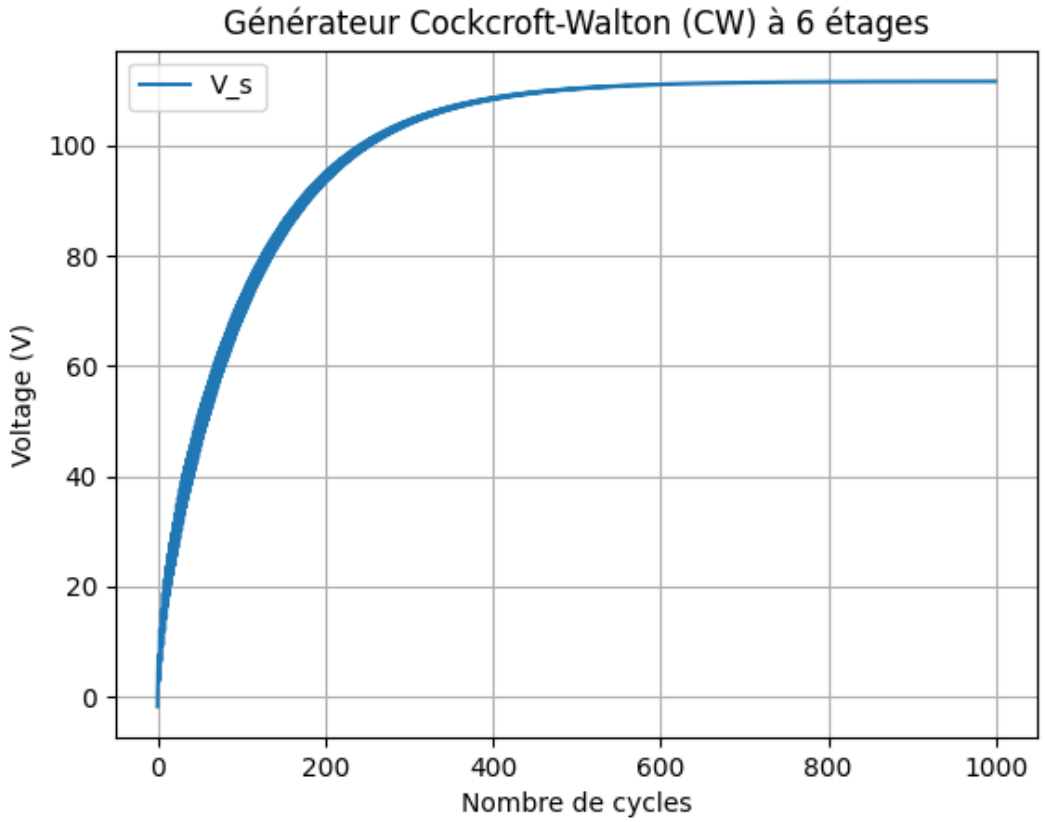
- $V_e = 10V$
- $V_d = 0{,}7\text{V}$ (Diode silicium)
- C independante d'étude

- Valeur finale : 37,2 V

Cela est conforme avec la formule







- **Valeur finale : 74,39 V**
- **Valeur finale : 111,2 V**

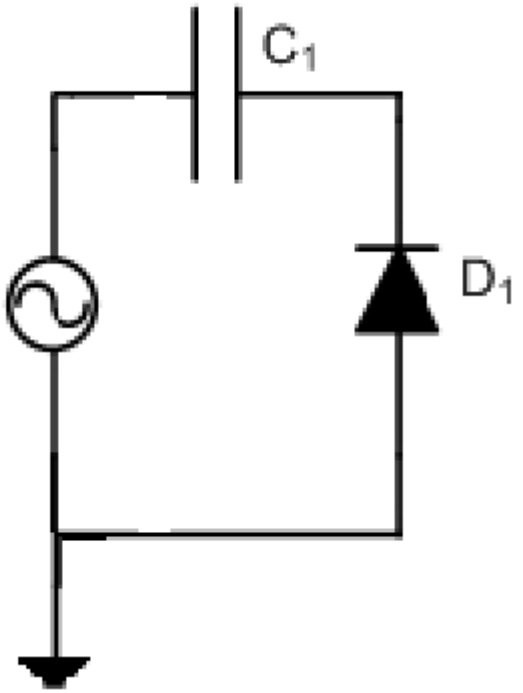
Cela est conforme avec la formule

- **Temps de réponse :**

$$Uc_1(t) = (V_e - V_d)(1 - exp(-t/RC))$$

Avec :

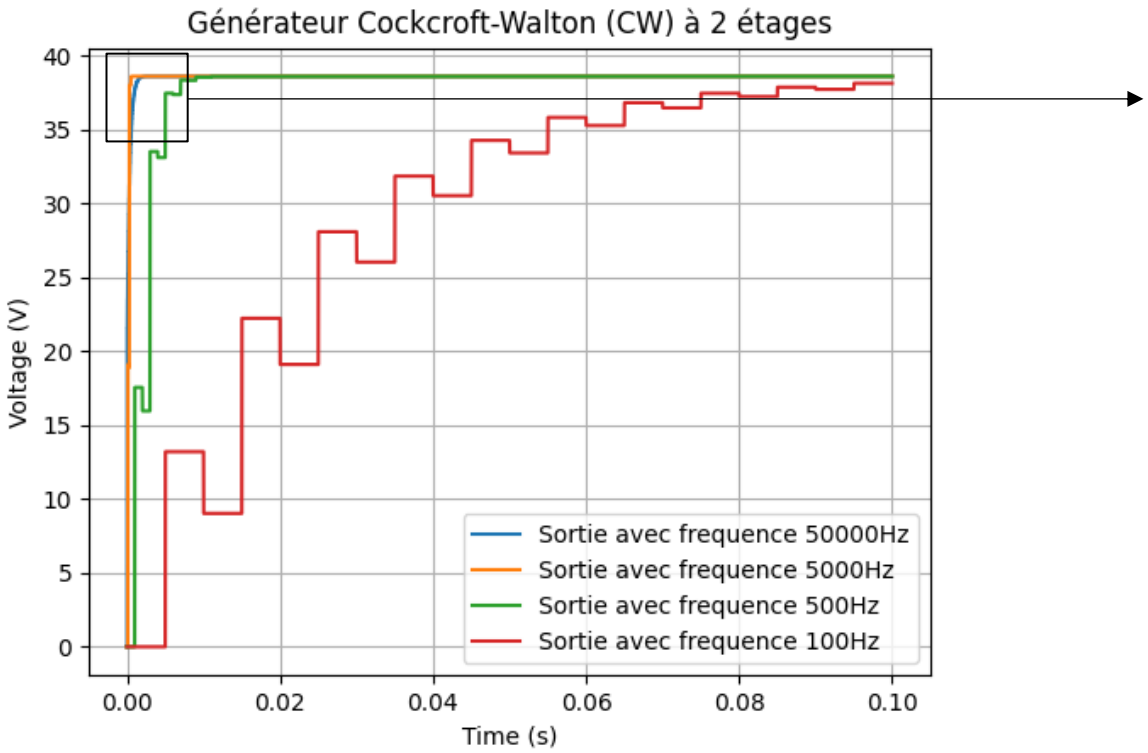
- **R : Resistance interne de condensateur**
- **C : Capacitance de condensateur**

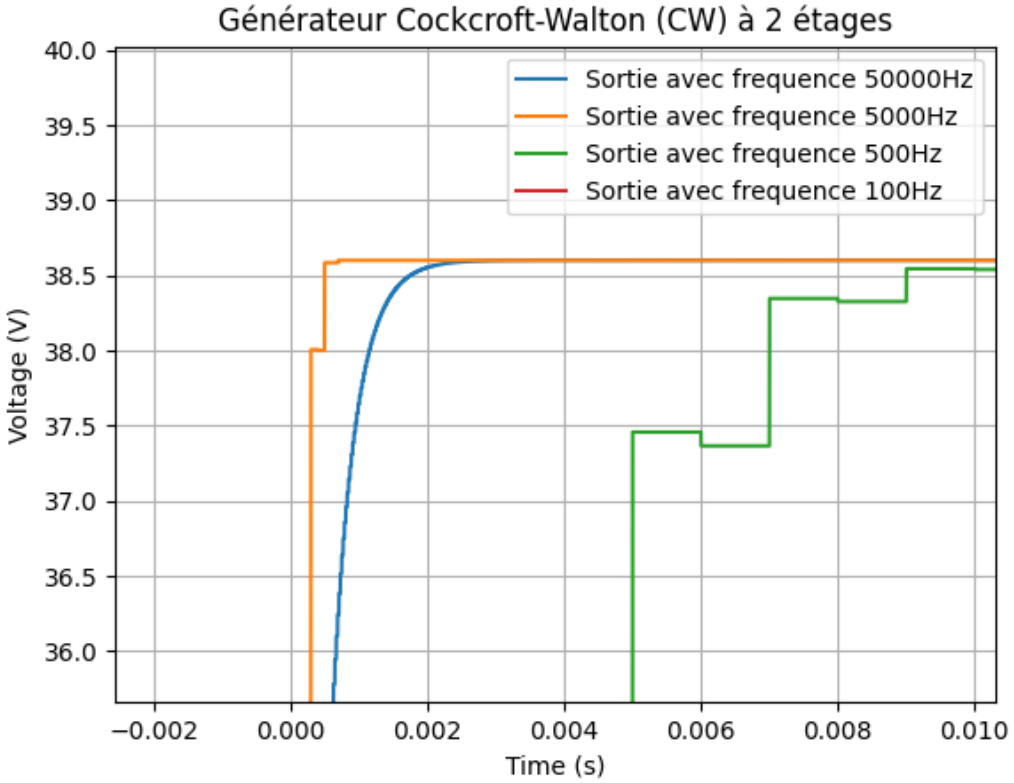


Une très haute fréquence et une très basse fréquence peut diminuer le temps de réponse !

Il existe une fréquence maximale

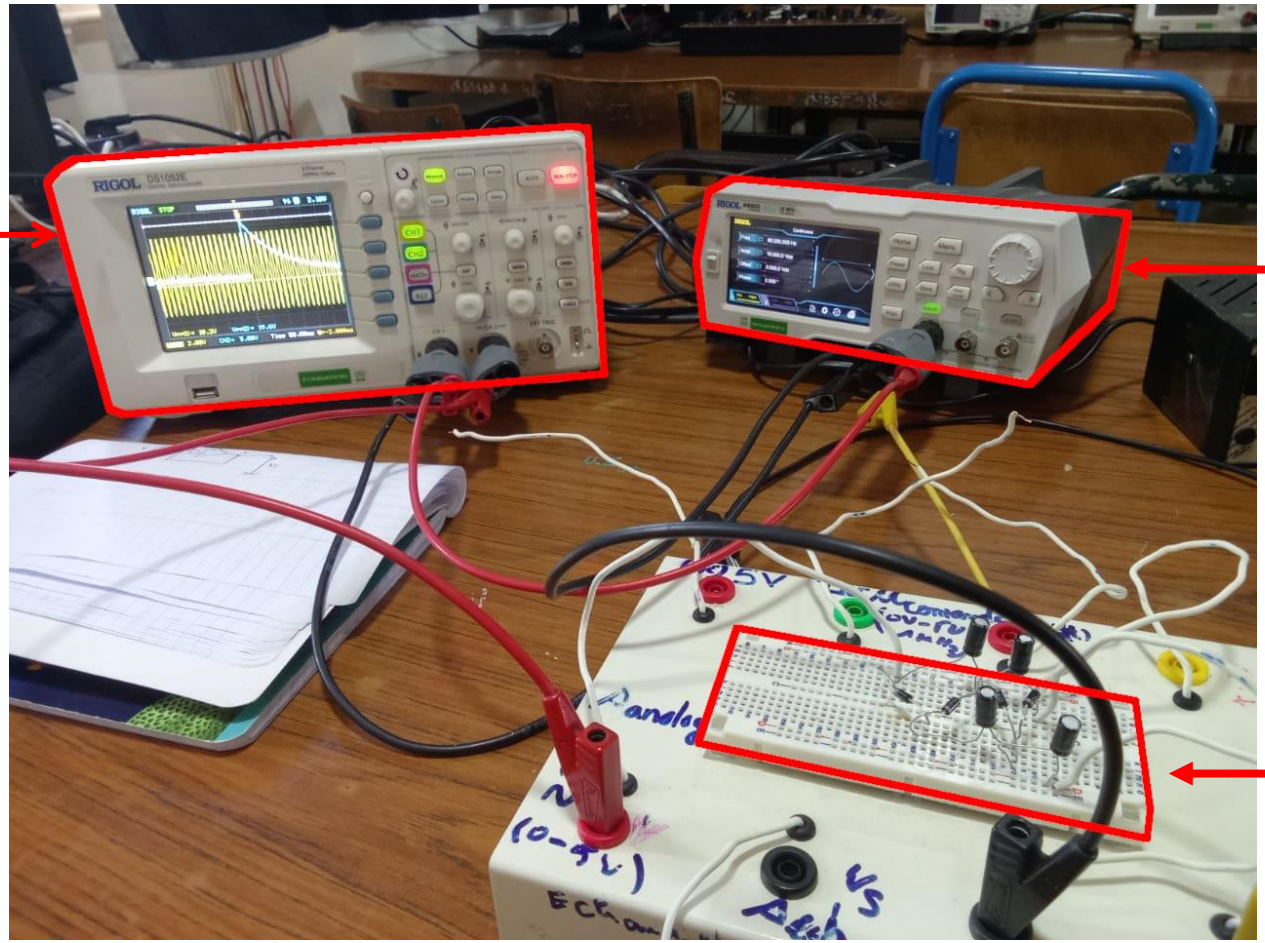
En tenant compte de frequence et valeurs de capacitance :



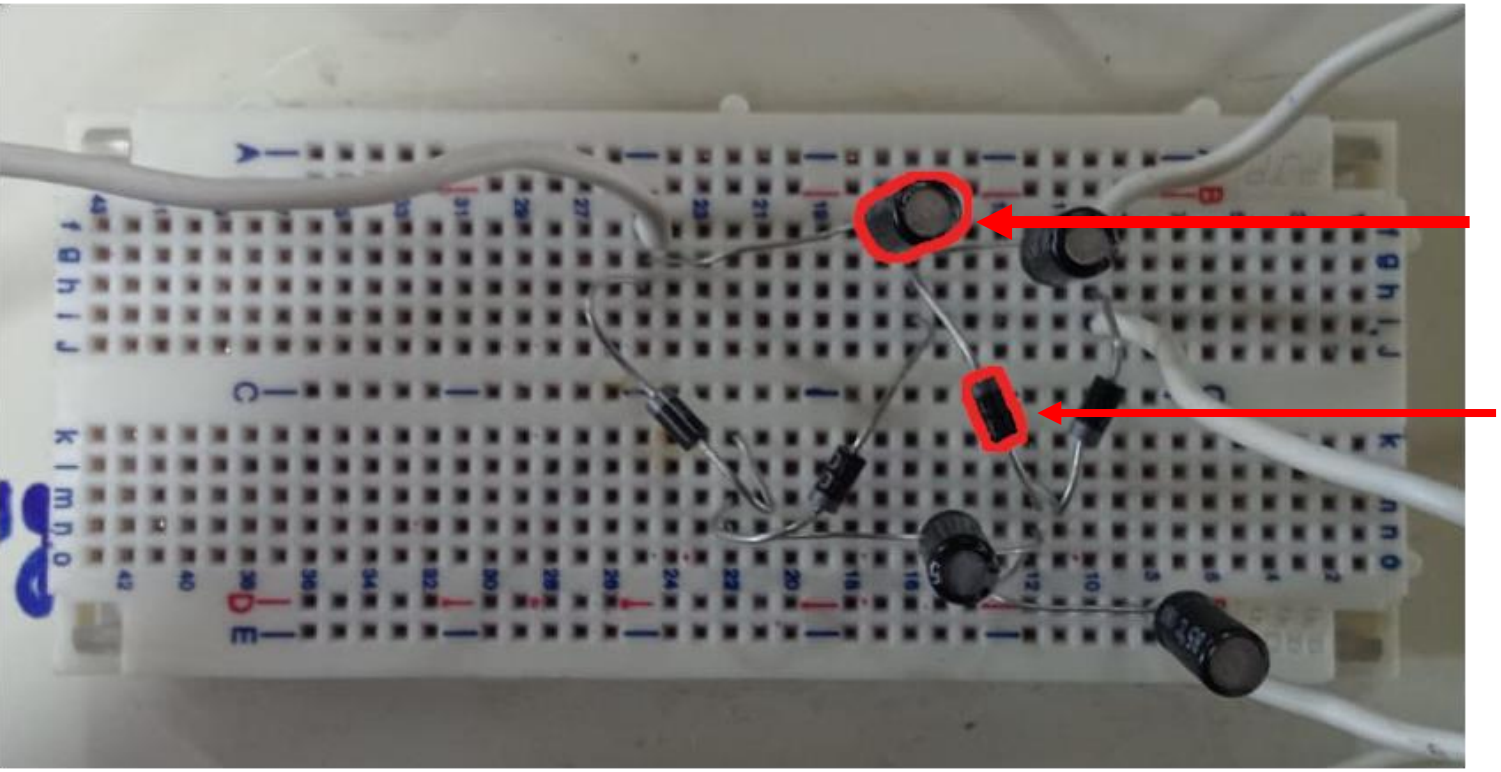


- $C = 10^{-3}$F
- R = 10 Ω
- V_e = 10 V

IV- Analyse expérimentale



- **Circuit du générateur a 2 étages :**



Condensteur électrolytique

Diode 1N4007 (Diode Silicon)

Valeurs utilisées :

Condensteur électrolytique 0,47 µF
Resistance interne : 0,1 – 10 Ω

Diode 1N4007 (Diode Silicon)
V_d = 0,7 V

Au pire des cas , où R = 10 Ω:

$$f_{max} = \frac{1}{RC} = \textbf{212 kHz}$$

- **On prend f = 60 Hz pour mieux simuler un signal d'entrée d'un prise électrique**

- **Lecture Oscilloscope**

Entrée : Vmax(1)= 7.80V

Sortie : Vmax(2)= 27.6V

Sortie attendu : 28.4 V

Entrée : Vmax(1)= 9.20V

Sortie : Vmax(2)= 32.8V

Sortie attendu : 34 V

- **Pertes possibles :**

  - Effet résistive du condensateur :

$$V_{\text{résistance}} = RI$$

  - Impedance du condensateur :

$$V_{\text{impédance}} = |Z_C| I = \frac{1}{2\pi f C} I$$

  - Effet inductive des câbles (Négligable car f est petite)

- **Mais des pertes deviennent négligeables devant une sortie de haute tension (kV)**

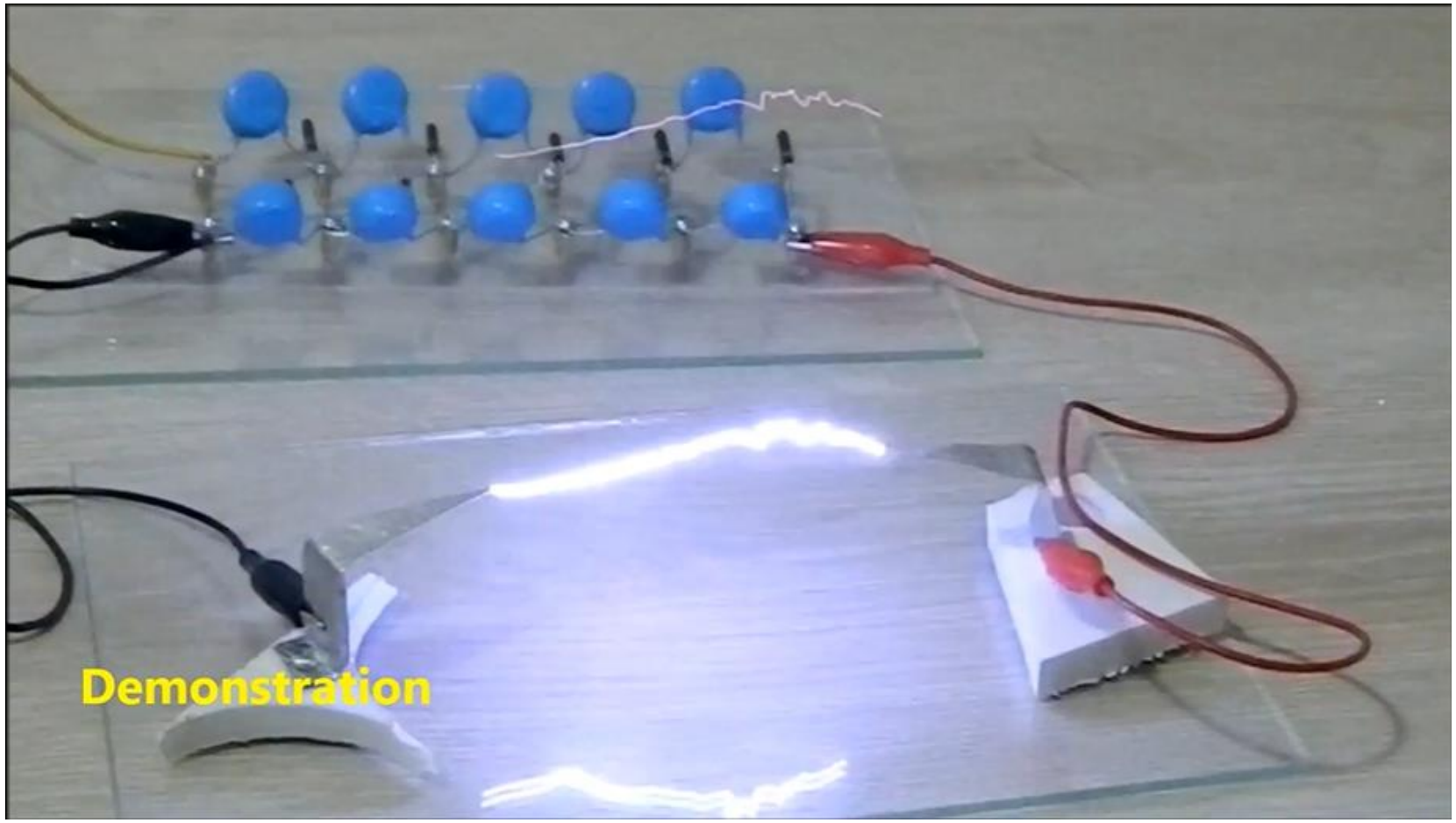


Figure 4 : Arc électrique de potentiel de 100 kV d'un entrée de 10 kV (x10 multiplication)
Source : Mirko Pavelski [https://www.youtube.com/watch?v=EFtNDtW804Q]

- **Verification des exigences :**

  - ✓ **Le générateur est flexible , On peut changer la multiplication du tension en modifiant le cascade du circuit.**
  - ✓ **Les composants (Condensateurs , Diodes) sont abordables.**
  - ✓ **Le générateur est capable de produire des hautes tensions.**

- ✓ **Donc ce circuit est capable d'étre integré dans une machine PEF !**

Merci pour votre attention !

- **Annexe : Code Python pour le tracé d'évolution de tension sortie**

```python
import numpy as np
import matplotlib.pyplot as plt
n = 4
E = 10
Values = 1000
diode_voltage = 0.7
Voltage_Capacitor = [[0 for i in range(Values)] for i in range(2*n)]
# Voltage_Capacitor[0] = U1 ...
# Voltage_Capacitor[1] = U2 ...
# Voltage_Capacitor[2] = U3 ...
for i in range(0,Values,2):
    Voltage_Capacitor[0][i] = E - diode_voltage
    for j in range(0,2*n-2,2):
        #Cycle 1 AC
        Voltage_Capacitor[j+1][i] = (Voltage_Capacitor[j+1][i-1]+Voltage_Capacitor[j+2][i-1] - diode_voltage)/2
        Voltage_Capacitor[j+2][i] = Voltage_Capacitor[j+1][i] + diode_voltage
        Voltage_Capacitor[j+3][i] = Voltage_Capacitor[j+3][i-1]

    for j in range(0,2*n,2):
        #Cycle 2 AC
        Voltage_Capacitor[j][i+1] = (Voltage_Capacitor[j][i]+Voltage_Capacitor[j+1][i] - E*(j == 0) + diode_voltage)/2
        Voltage_Capacitor[j+1][i+1] = Voltage_Capacitor[j][i+1] + E*(j == 0) - diode_voltage
array_sum = np.sum([Voltage_Capacitor[2*i+1] for i in range(0,n)],axis=0)
plt.plot(array_sum,label = "V_s")
plt.title("Générateur Cockcroft-Walton (CW) à "+str(n)+" étages")
plt.xlabel("Nombre de cycles")
plt.ylabel("Voltage (V)")
plt.legend()
plt.grid()

plt.show()

```

- **Annexe : Code Python pour la comparaison des fréquences**

```python
# Voltage_Capacitor[0] = U1 ...
# Voltage_Capacitor[1] = U2 ...
import numpy as np
import matplotlib.pyplot as plt

def update_voltage(old,target,t,resistor,capacitance):
        return target + (old - target)*np.exp(-t/(resistor*capacitance))
def sinFunction(E,t,frequency):
    return E*np.sin(t*frequency*2*np.pi)

cap_resistor = 10
cap_capacitance = 1e-3
diode_voltage = 0.7

Values = 300000
n = 2
timeStop = 0.1
time = np.linspace(0,timeStop,Values+1)
```

- **Annexe : Code Python pour la comparaison des fréquences**

```python
def CWGen(E,frequency):
    dt = time[1] - time[0]
    Voltage_Capacitor = [[0 for _ in range(Values+1)] for _ in range(2*n)]
    i = 1
    t = 0
    while t < timeStop and i <= Values:
        if sinFunction(E,t,frequency) < 0:
            for j in range(0,2*n,2):
                #Cycle 2 AC
                 Voltage_Capacitor[j][i] = update_voltage(Voltage_Capacitor[j][i-1],(Voltage_Capacitor[j][i-1]+Voltage_Capacitor[j+1]
                 [i-1]-E*(j == 0))/2,1/frequency,2*cap_resistor,cap_capacitance/2)
                 Voltage_Capacitor[j+1][i] = Voltage_Capacitor[j][i] + E*(j == 0)
        else:
            Voltage_Capacitor[0][i] = update_voltage(Voltage_Capacitor[0][i-1],E-diode_voltage,1/frequency,cap_resistor,
            cap_capacitance)
            for j in range(0,2*n-2,2):
            #Cycle 1 AC
                Voltage_Capacitor[j+1][i] = update_voltage(Voltage_Capacitor[j+1][i-1],(Voltage_Capacitor[j+1][i-1]+Voltage_Capacitor[j
                +2][i-1])/2,1/frequency,2*cap_resistor,cap_capacitance/2)
                Voltage_Capacitor[j+2][i] = Voltage_Capacitor[j+1][i]
                Voltage_Capacitor[j+3][i] = Voltage_Capacitor[j+3][i-1]
        t += dt
        i += 1
    return Voltage_Capacitor
```

- **Annexe : Code Python pour la comparaison des fréquences**

```
44  Voltage_Capacitor = CWGen(10,50000)
45  Voltage_Capacitor2 = CWGen(10,5000)
46  Voltage_Capacitor3 = CWGen(10,500)
47  Voltage_Capacitor4 = CWGen(10,100)
48
49  plt.xlabel("Time (s)")
50  plt.ylabel("Voltage (V)")
51  plt.title("Générateur Cockcroft-Walton (CW) à "+ str(n) +" étages")
52  Out = np.sum([Voltage_Capacitor[2*i+1] for i in range(0,n)],axis=0)
53  Out2 = np.sum([Voltage_Capacitor2[2*i+1] for i in range(0,n)],axis=0)
54  Out3 = np.sum([Voltage_Capacitor3[2*i+1] for i in range(0,n)],axis=0)
55  Out4 = np.sum([Voltage_Capacitor4[2*i+1] for i in range(0,n)],axis=0)
56  plt.plot(time,Out,label = "Sortie avec frequence 50000Hz")
57  plt.plot(time,Out2,label = "Sortie avec frequence 5000Hz")
58  plt.plot(time,Out3,label = "Sortie avec frequence 500Hz")
59  plt.plot(time,Out4,label = "Sortie avec frequence 100Hz")
60  plt.legend()
61  plt.grid()
62  plt.show()
63
64
```

- **Annexe : Preuve des tensions de générateur 1 étages.**

Par loi de Kirchhoff: $V_e + Uc_1(n+1) = V_d + Uc_2(n+1)(*)$
On pose: $\Delta V_n = \frac{dUc_2}{dt}$

$Uc_2(n+1) = Uc_2(n) + \Delta V_n$
$Uc_1(n+1) = Uc_1(n) - \Delta V_n$
$Uc_1(n) = V_e - V_d$

En utilisant (*) , On a :

$V_e + Uc_1(n) - \Delta V_n = V_d + Uc_2(n) + \Delta V_n$
$\rightarrow \Delta V_n = -\frac{Uc_2(n)}{2} + (V_e - V_d)$
Alors : $\boxed{Uc_2(n+1) = \frac{Uc_2(n)}{2} + V_e - V_d}$