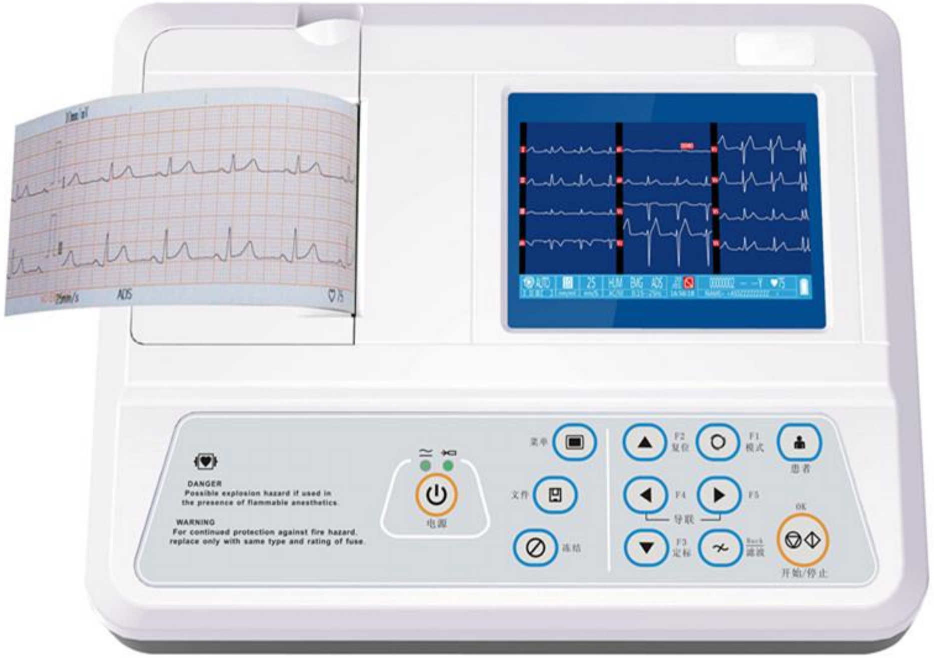
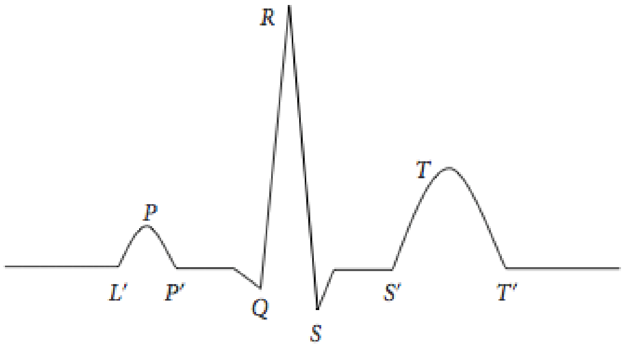
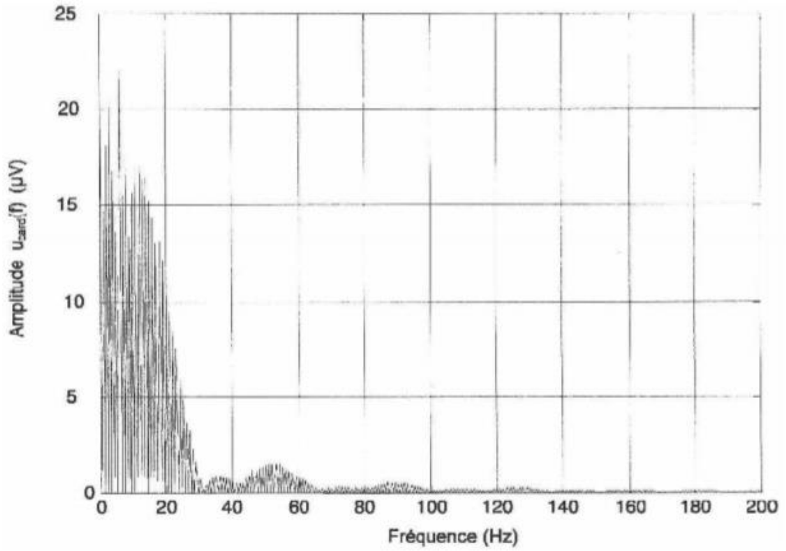


Atténuation du bruit en électrocardiographie

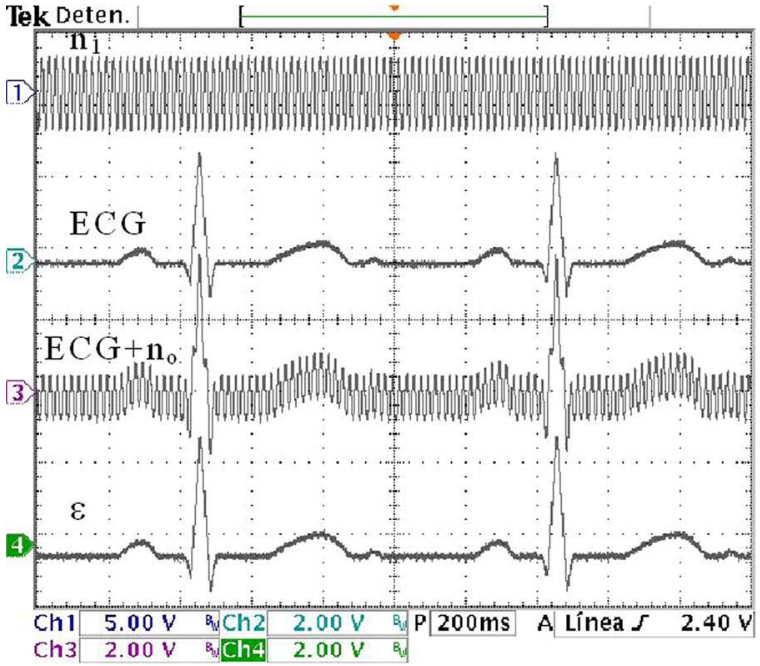
Problématique:



Composantes d'un signal cardiaque

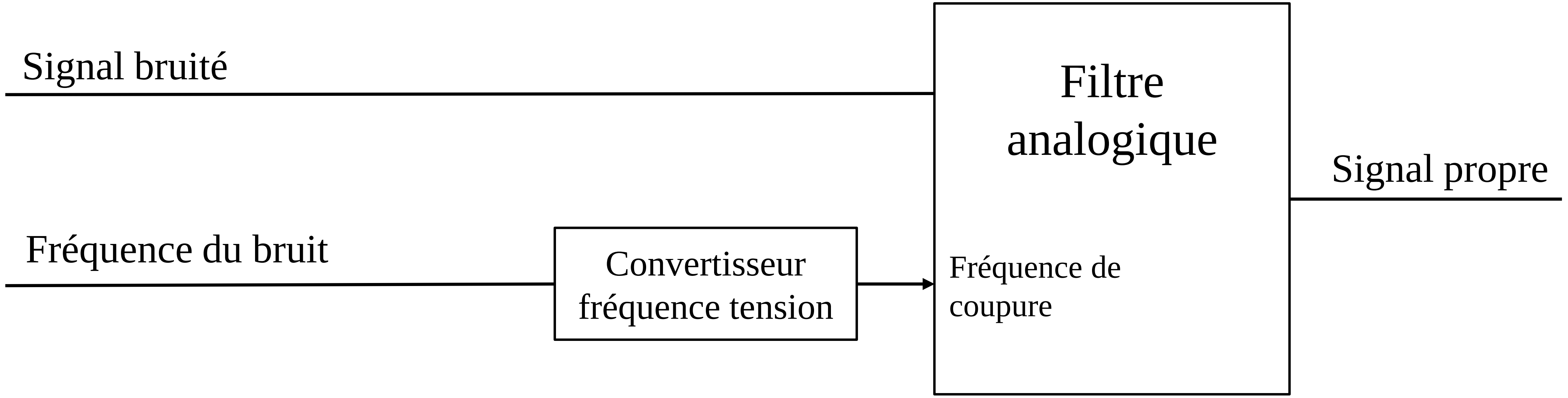


Spectre d'un signal ECG



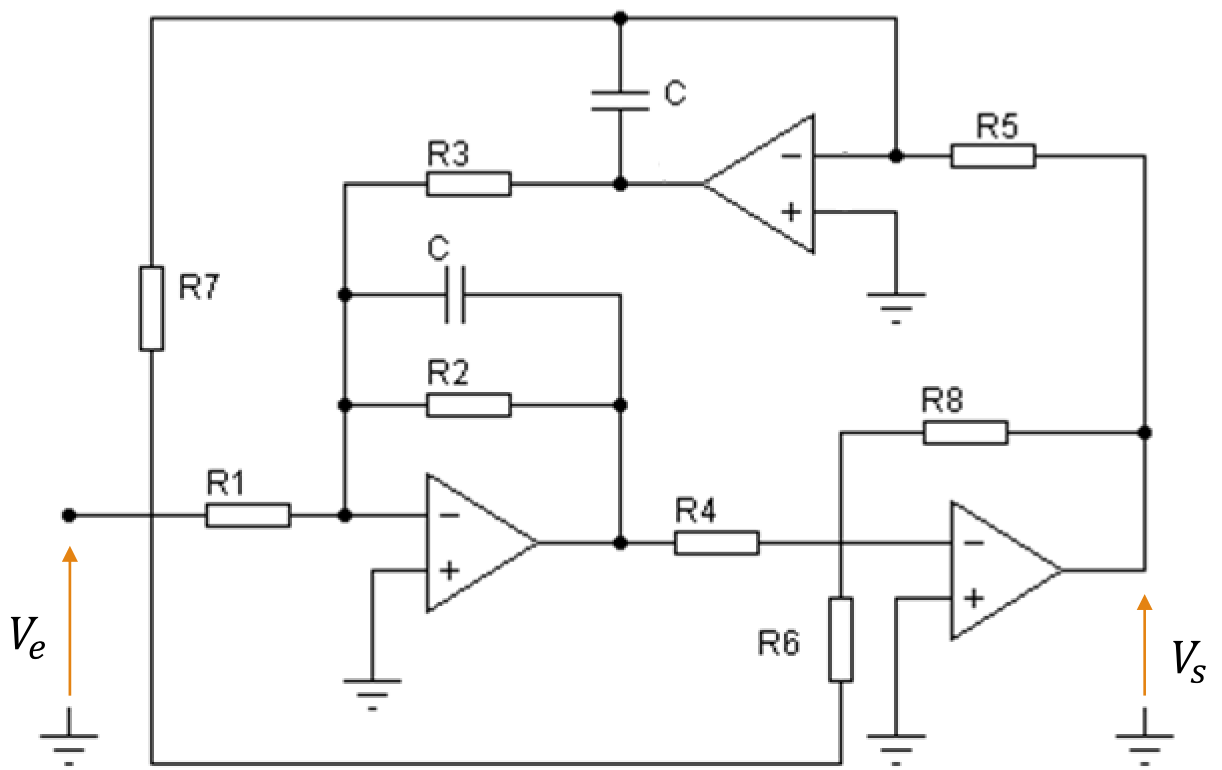
Atténuation par filtrage adaptatif

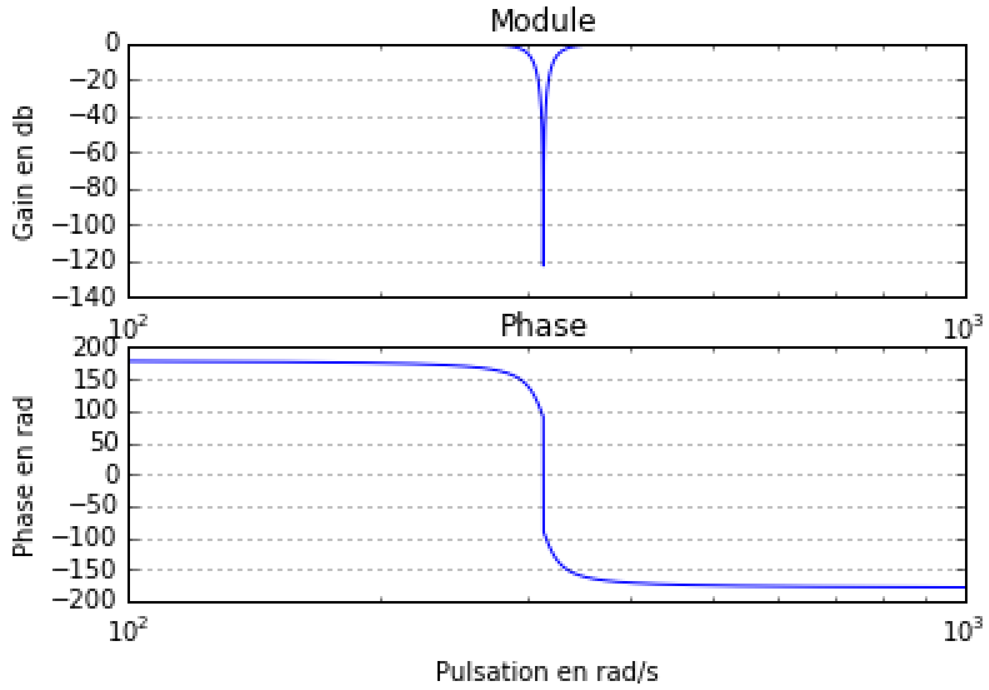
Modélisation du système:



Sommaire:

Filtre coupe-bande 4ème ordre: Caractéristiques

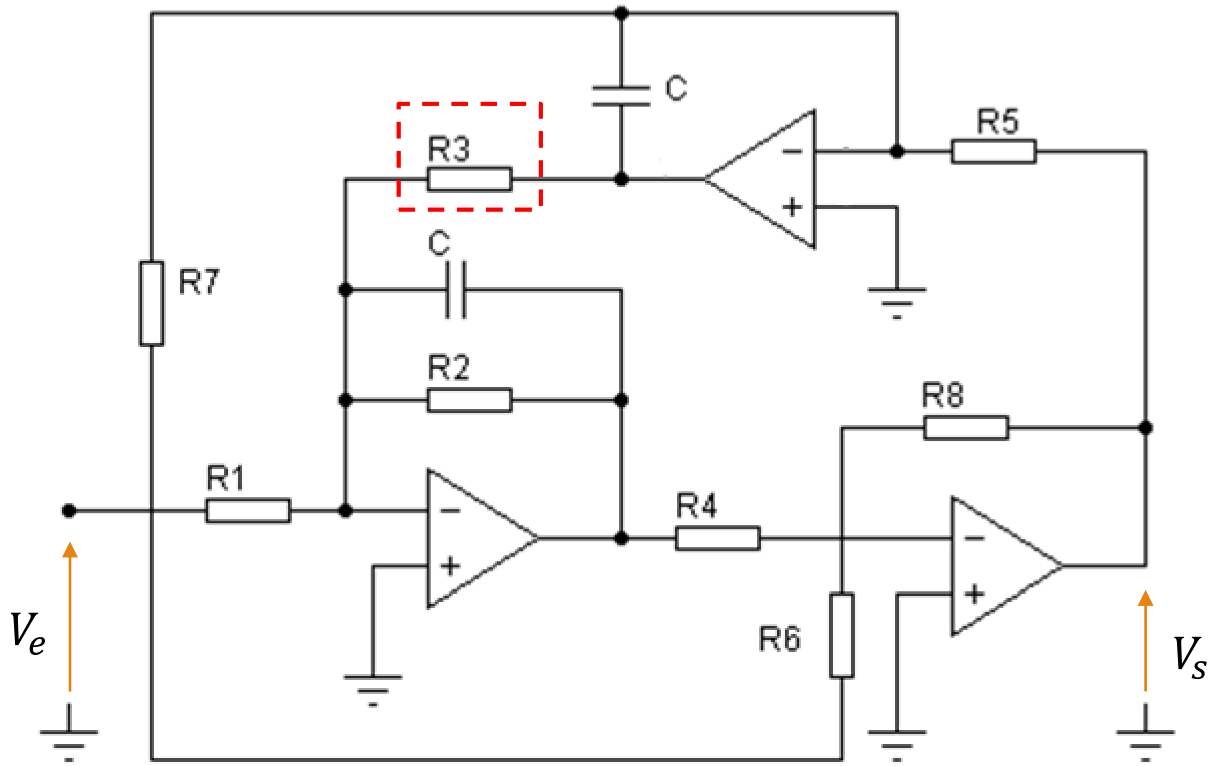


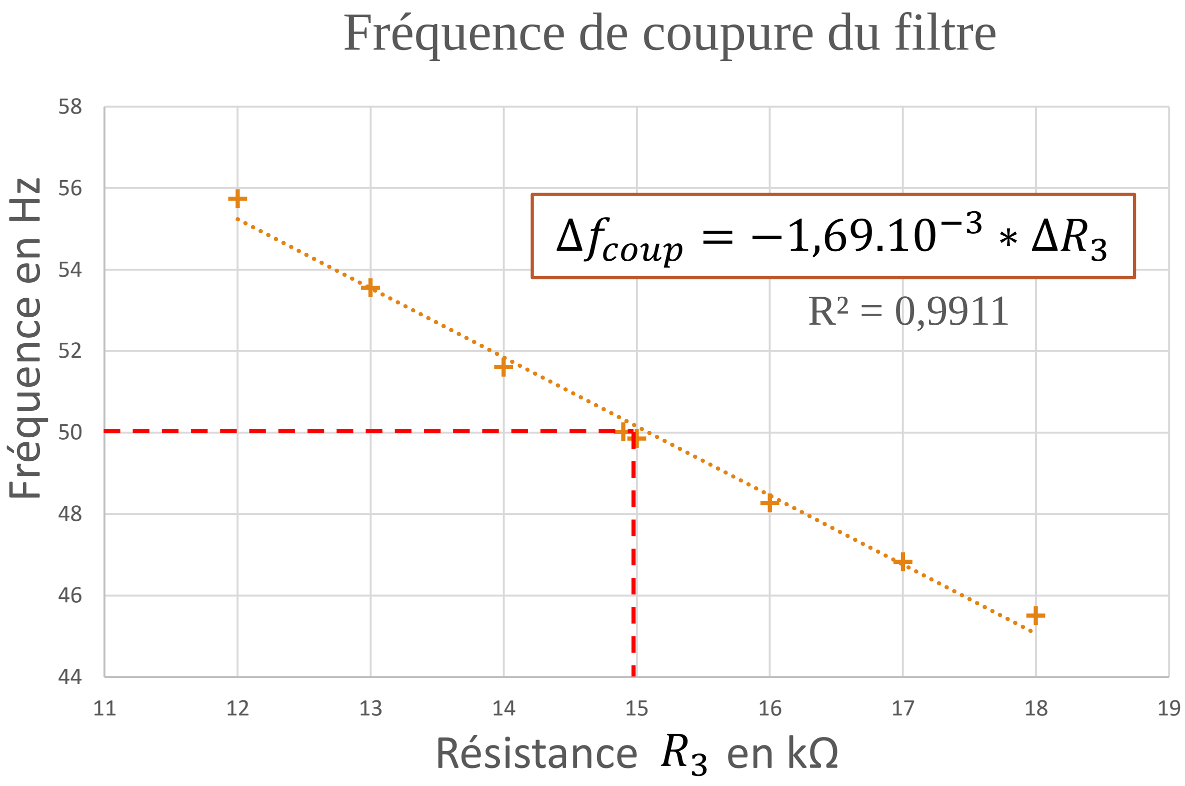


Q = 10

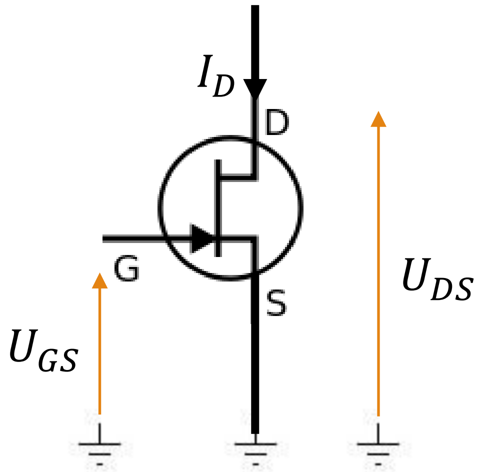
$\omega_0 = 2\pi.f = 314,2\ rad/s$

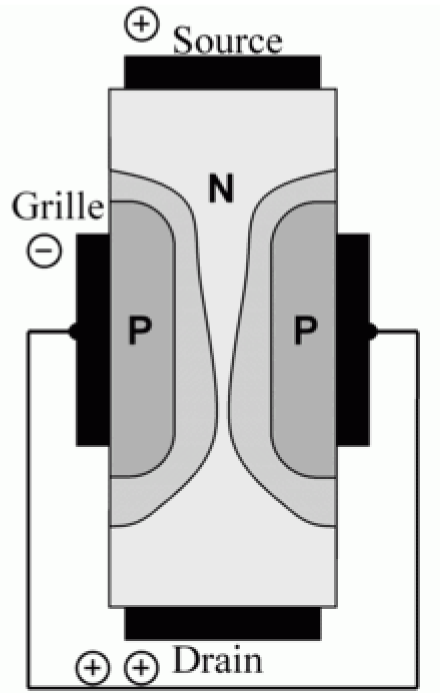
Filtre coupe-bande 4ème ordre: Résistance variable

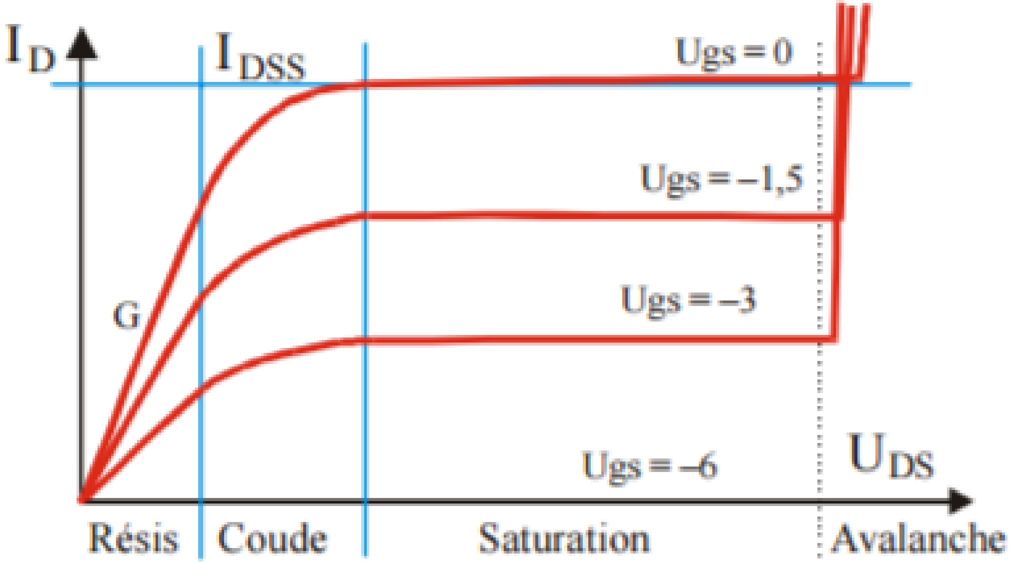




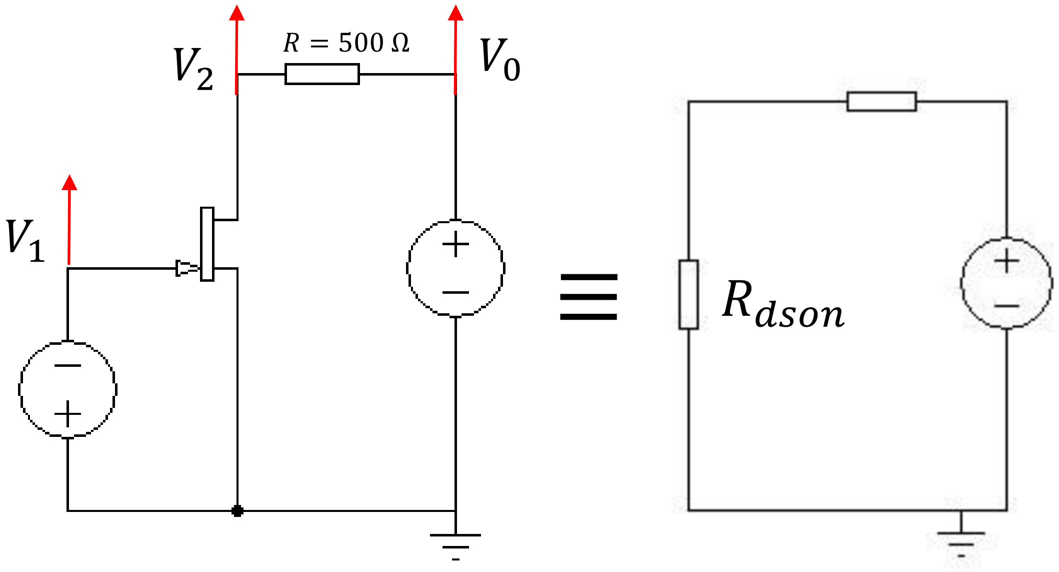
Transistor à effet de champ: Principe de fonctionnement





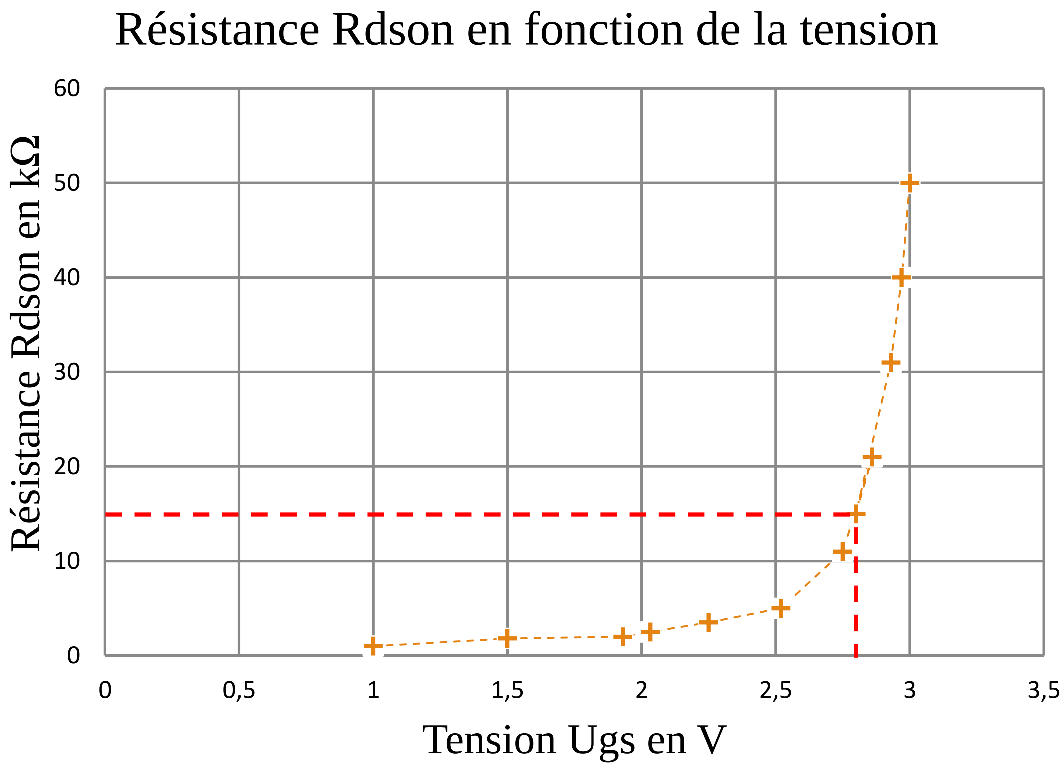


Transistor à effet de champ: Relevé de la caractéristique

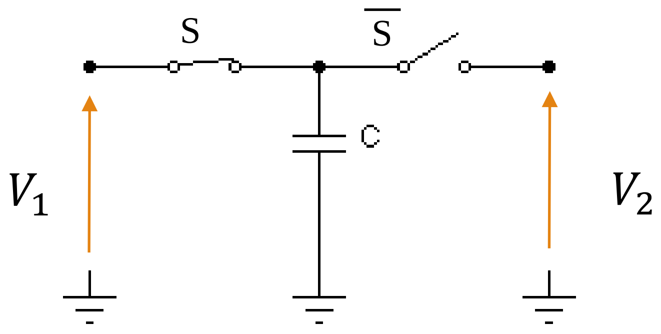


Courant dans le circuit: $I = \frac{V_2 - V_0}{R} = \frac{V_2}{R_{dson}}$

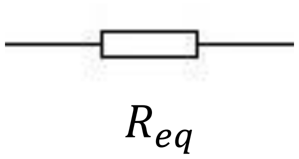
$$\boxed{R_{dson} = \frac{V_2}{V_2 - V_0} * R}$$

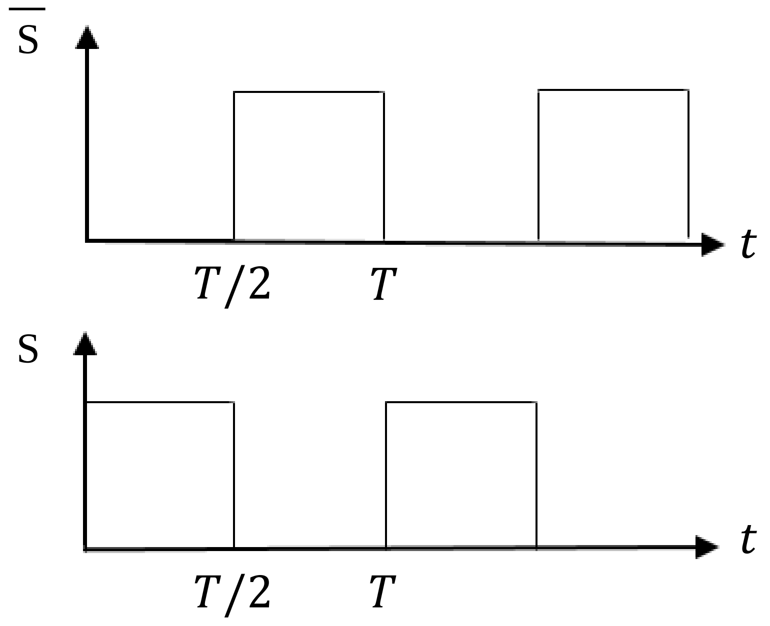


Circuit à capacités commutées: Principe de fonctionnement



$=$





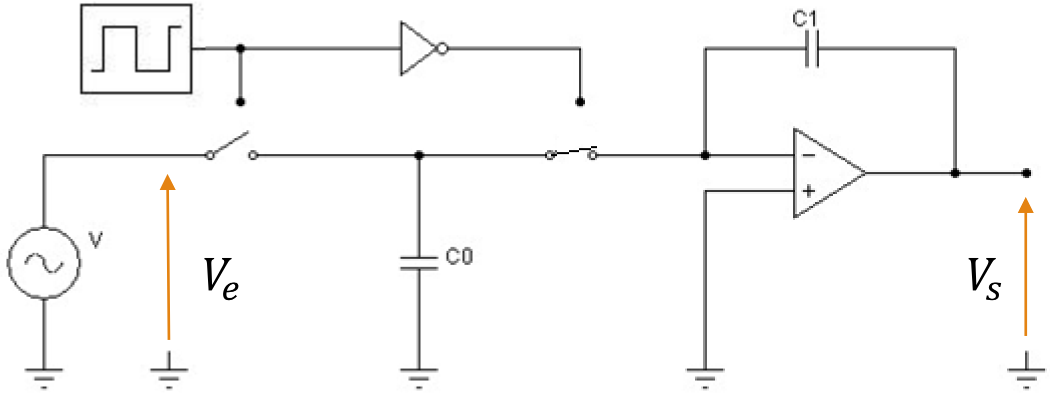
- Pendant [0,T/2]: S fermé, $V_1 = \frac{Q_1}{C}$
- Pendant [T/2,T]: S ouvert, $V_2 = \frac{Q_2}{C}$

$$\Delta Q = Q_2 - Q_1 = C(V_2 - V_1) = C.V_{21}$$

$I = \frac{\Delta Q}{T}$ donc $V_{21} = R_{eq}.I$ avec $\boxed{R_{eq} = \frac{T}{C} = \frac{1}{C.f_{clk}}}$

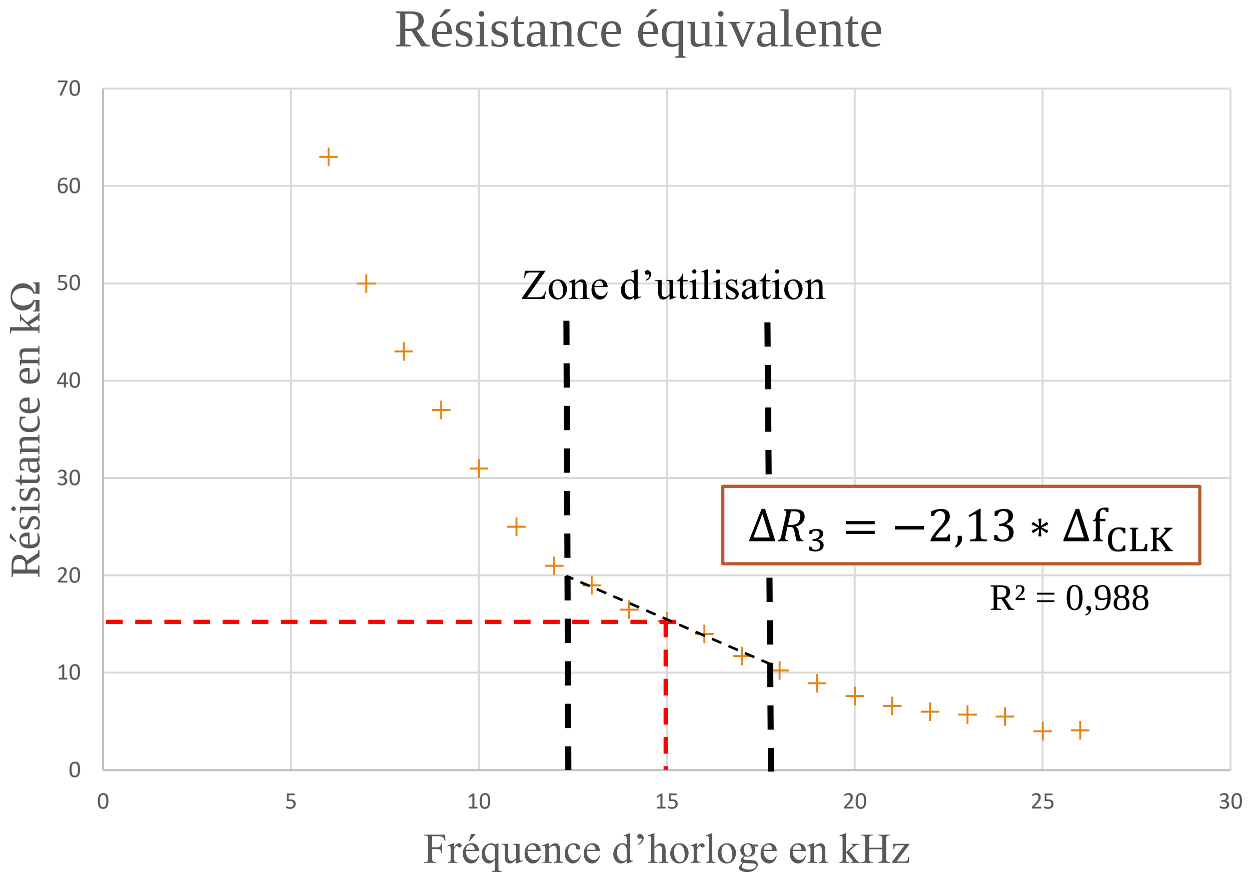
Circuit à capacités commutées: Relevé de la caractéristique

Fréquence d'horloge

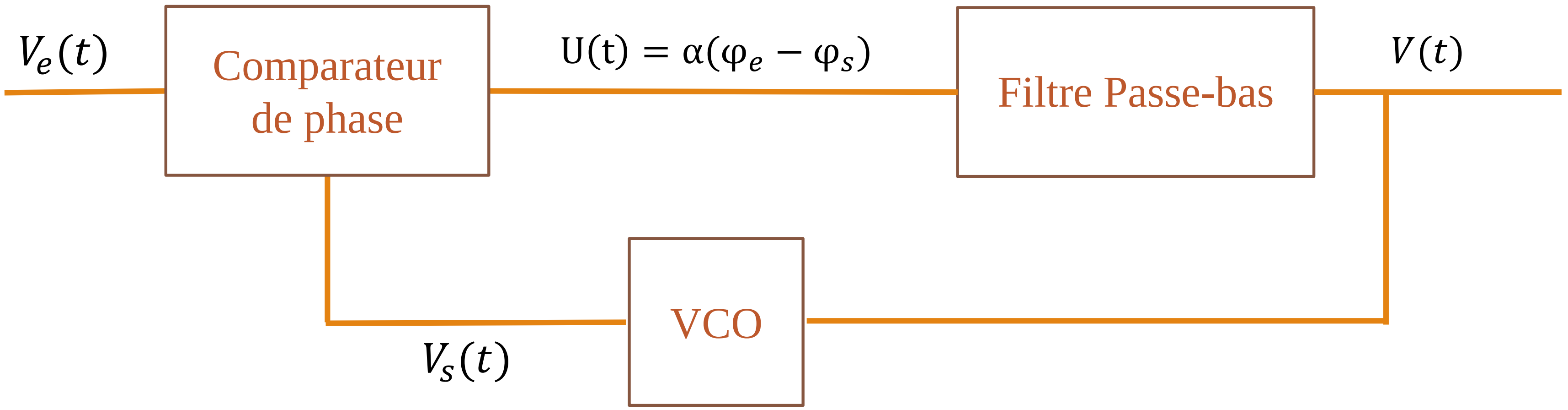


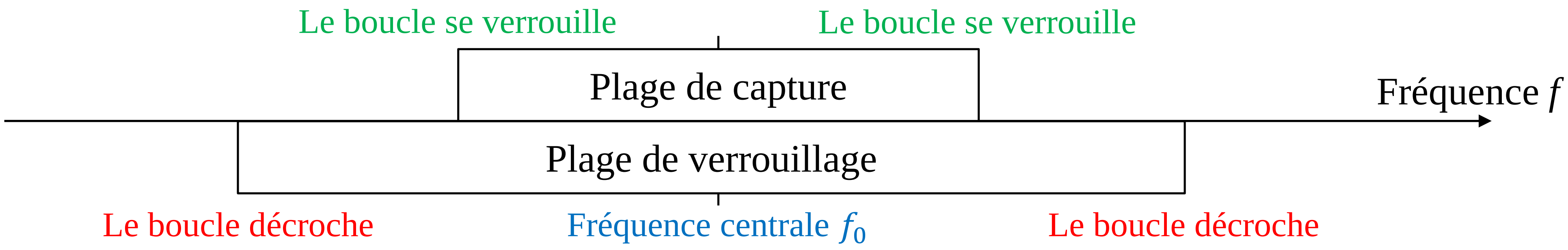
$$H(j\omega) = \frac{V_s}{V_e} = -\frac{1}{jR_{eq}C\omega}$$

$$\boxed{R_{eq} = \frac{1}{\frac{V_{sl}}{V_{el}}C\omega}}$$

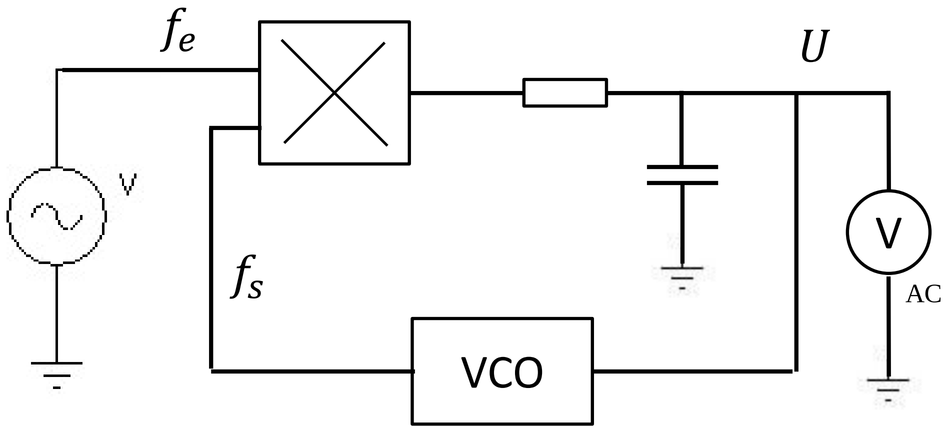


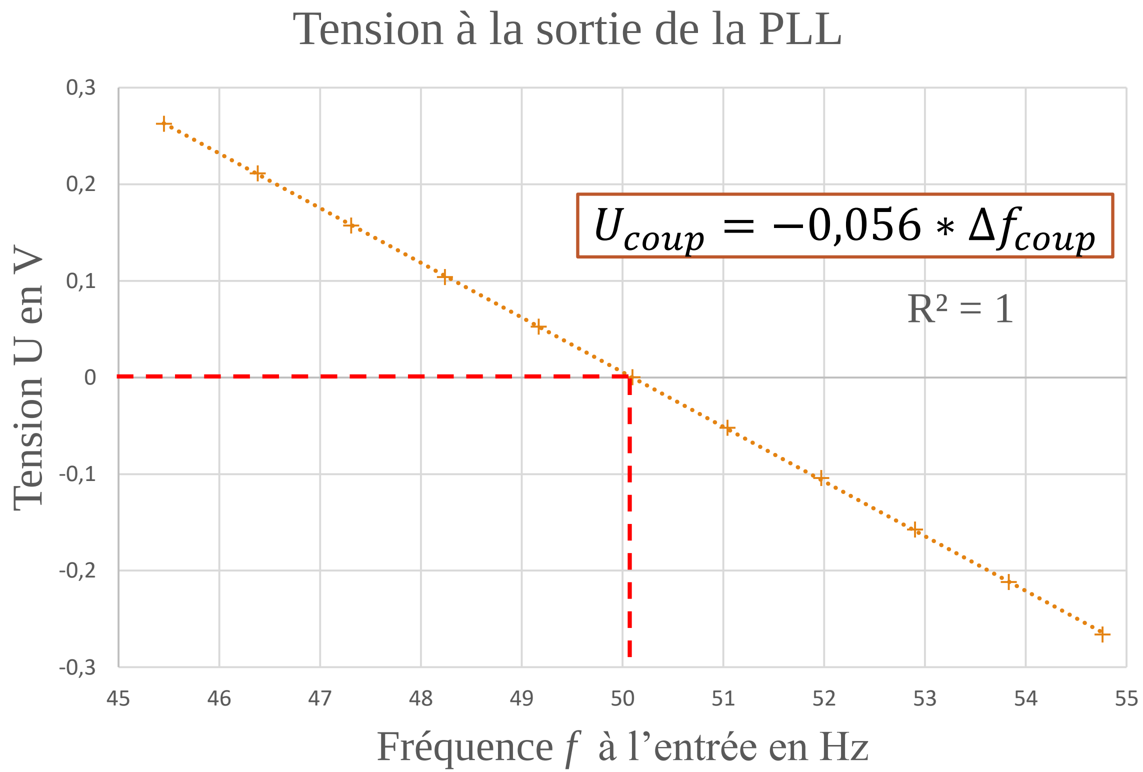
Boucle à verrouillage de phase: Principe de fonctionnement



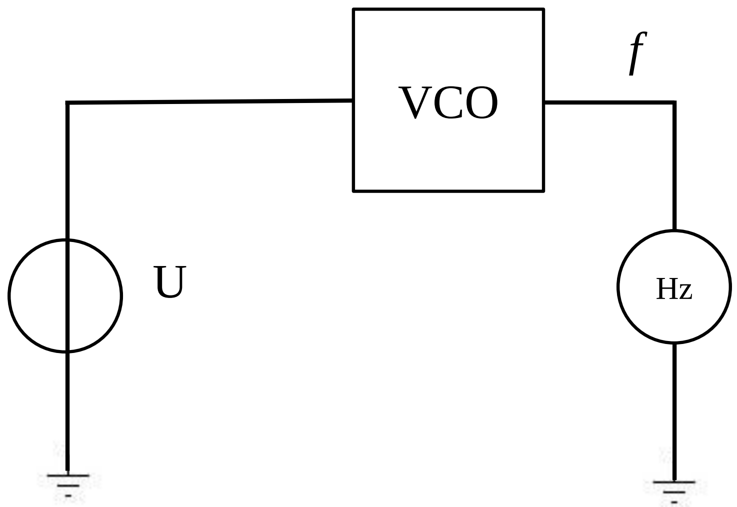


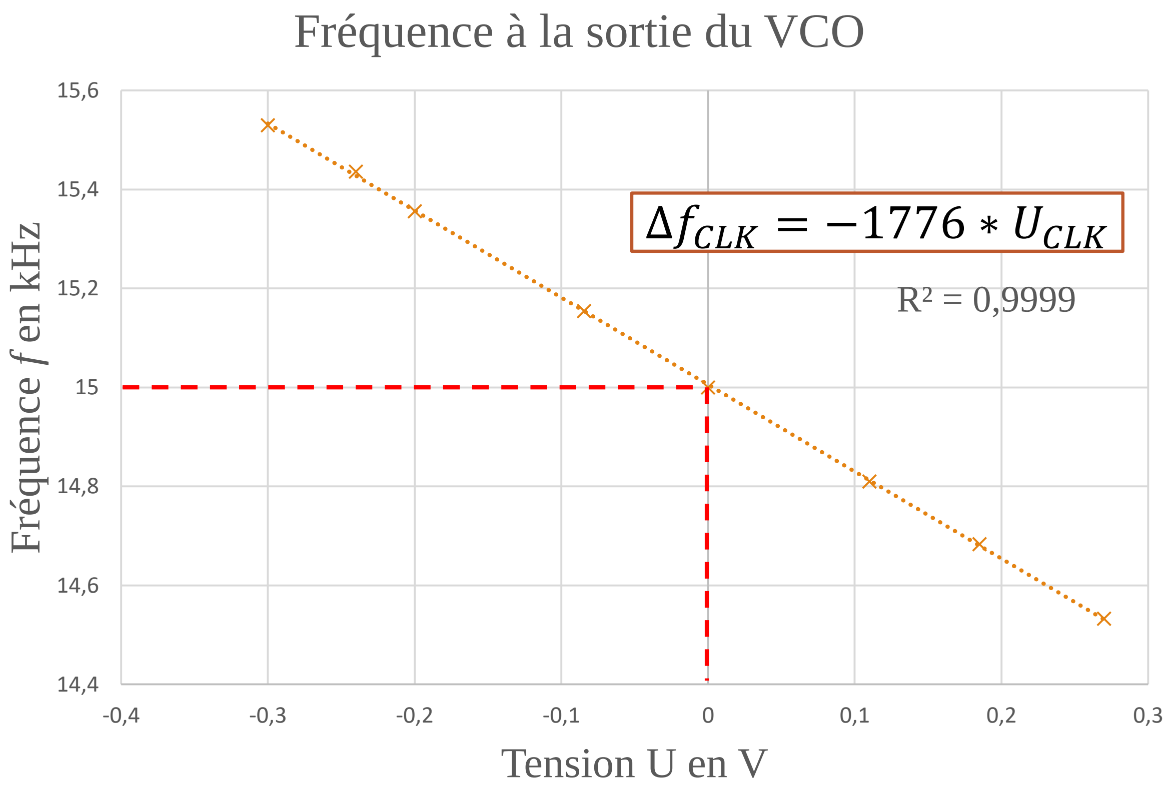
Boucle à verrouillage de phase:
Caractéristique en 50 Hz:



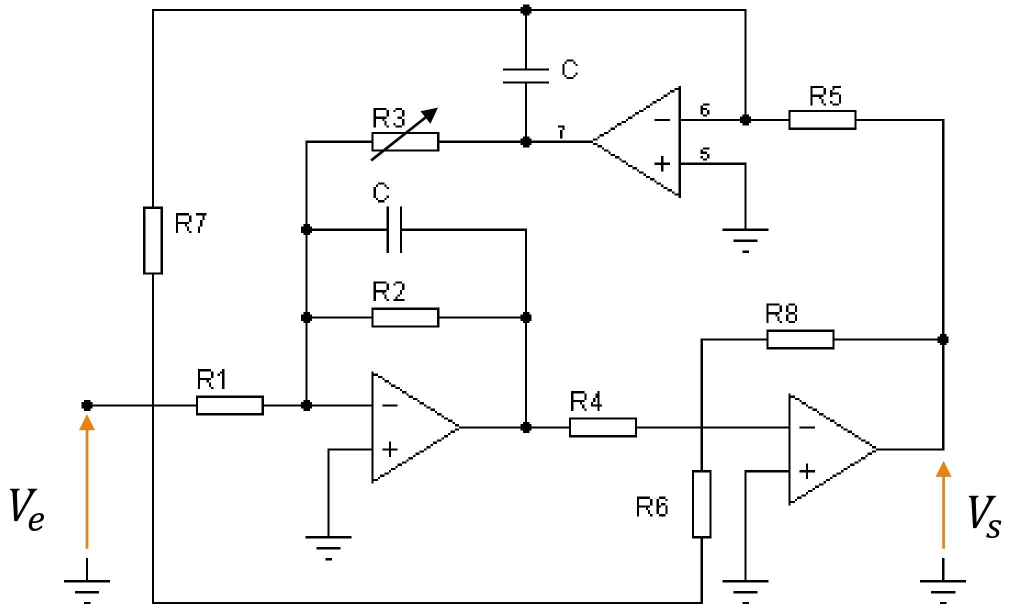


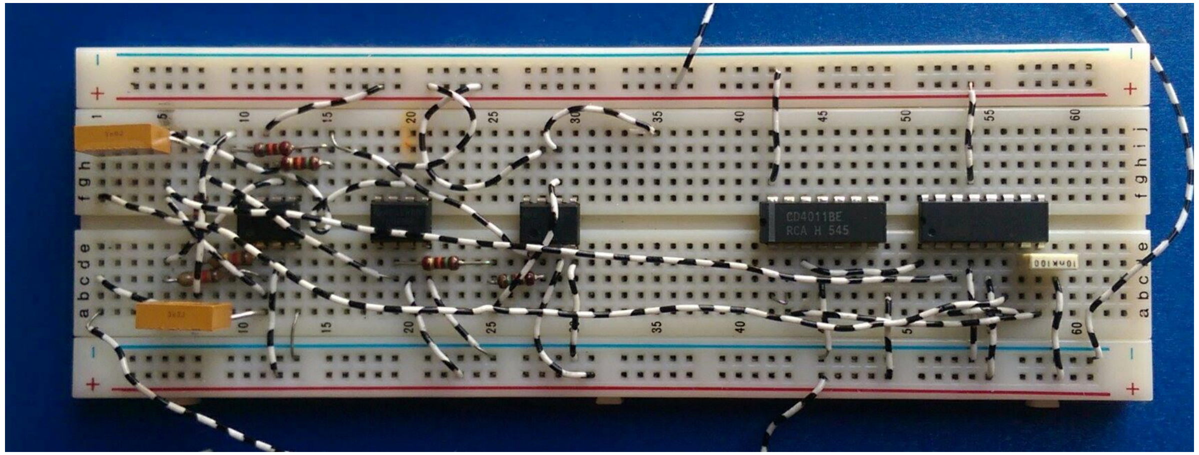
Oscillateur commandé en tension: Caractéristique en 15 kHz:





Système final:





PLL	Amplificateur	VCO	Circuit à CC
-0,056	2,79	-1,758	-2,13

Δf_{bruit} → PLL (-0,056) → U_{bruit} → Amplificateur (2,79) → U_{CLK} → VCO (-1,758) → Δf_{CLK} → Circuit à CC (-2,13) → ΔR_3

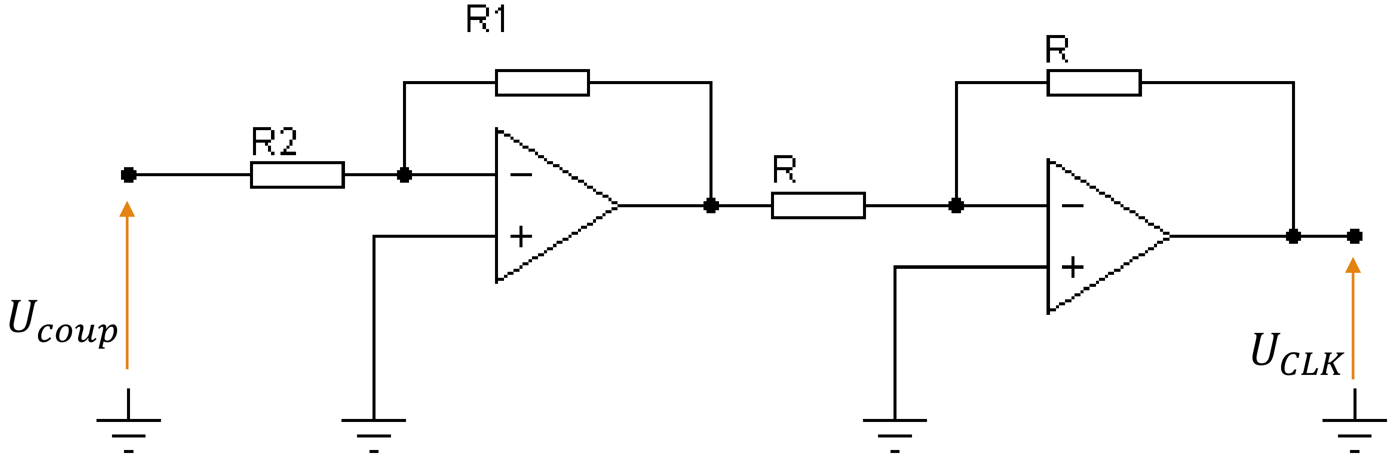
Calcul de l'amplificateur:

$\Delta f_{coup} = -0{,}0017 * \Delta R_3$ avec $\Delta f_{bruit} = \Delta f_{coup}$

$\Delta R_3 = -2{,}13 * \Delta f_{CLK}$

$\Delta f_{CLK} = -1776 * U_{CLK}$

$U_{coup} = -0{,}056 * \Delta f_{coup}$

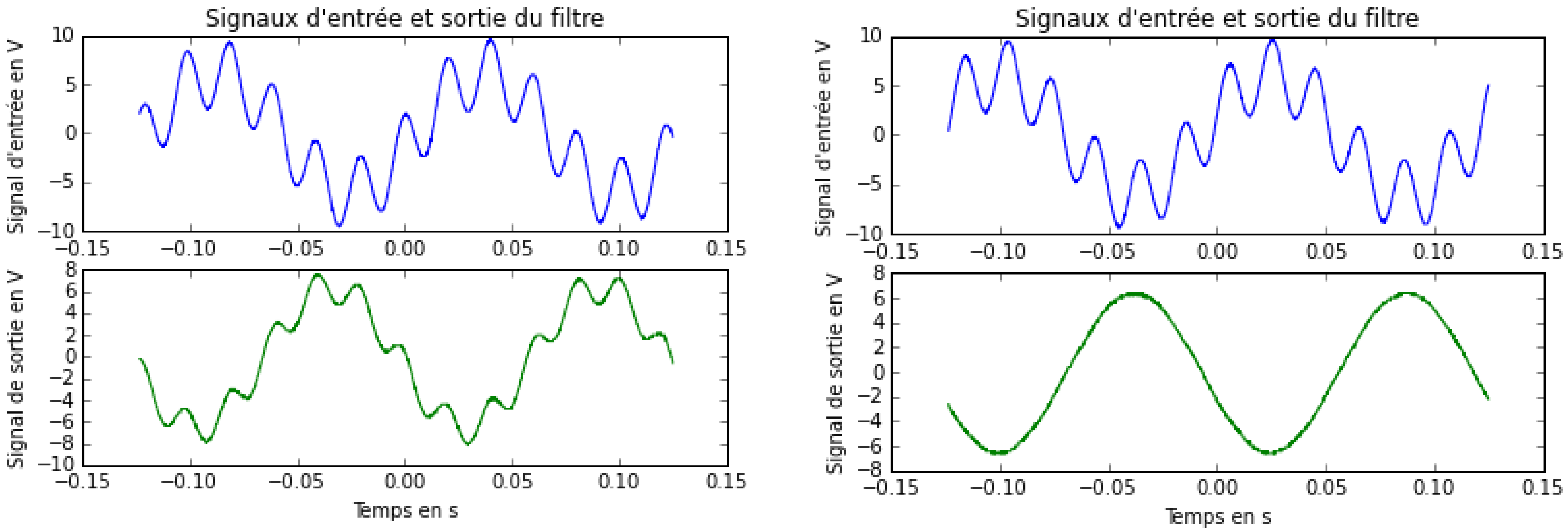


$$K = \frac{U_{CLK}}{U_{coup}} = \frac{1}{0{,}0036 * 0{,}056 * 1776}$$

$$\boxed{K = 2{,}79}$$

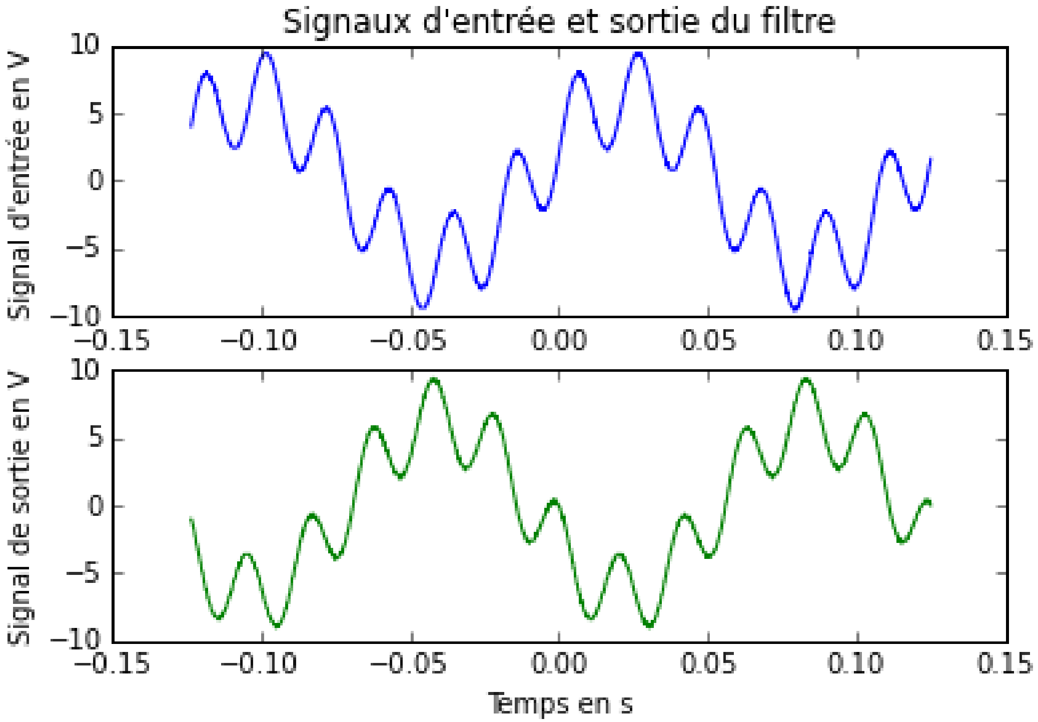
$$\boxed{K = \frac{U_{CLK}}{U_{coup}} = \frac{R_1}{R_2}}$$

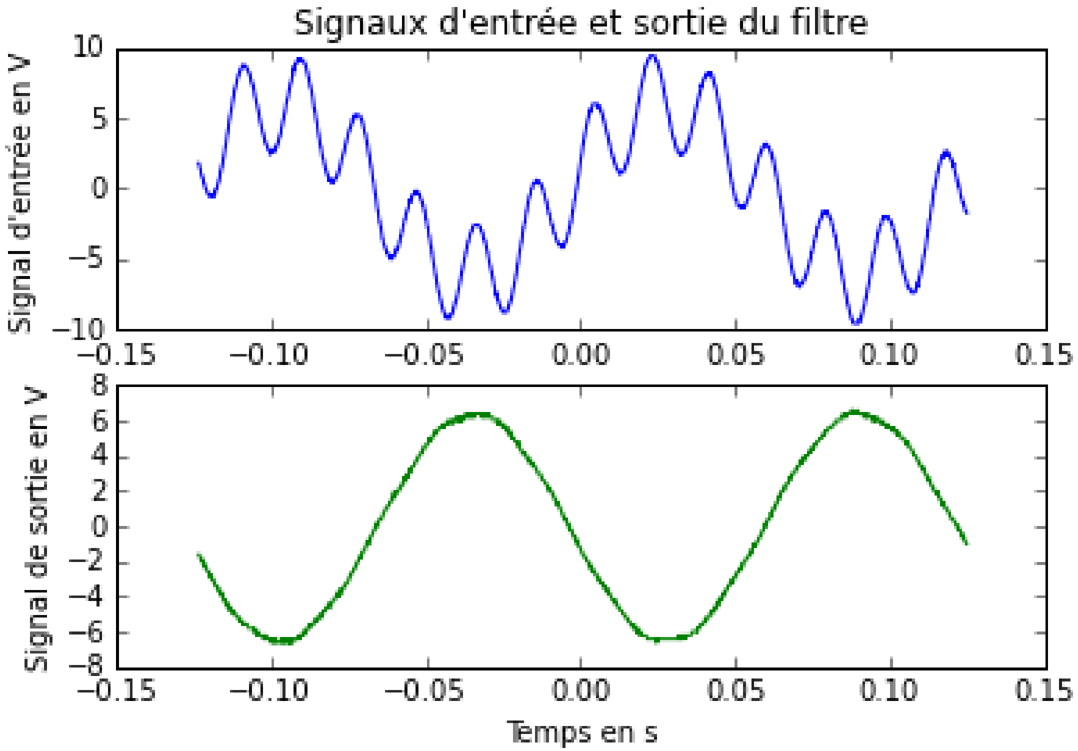
Résultats expérimentaux:



Fréquence du bruit: 49 Hz

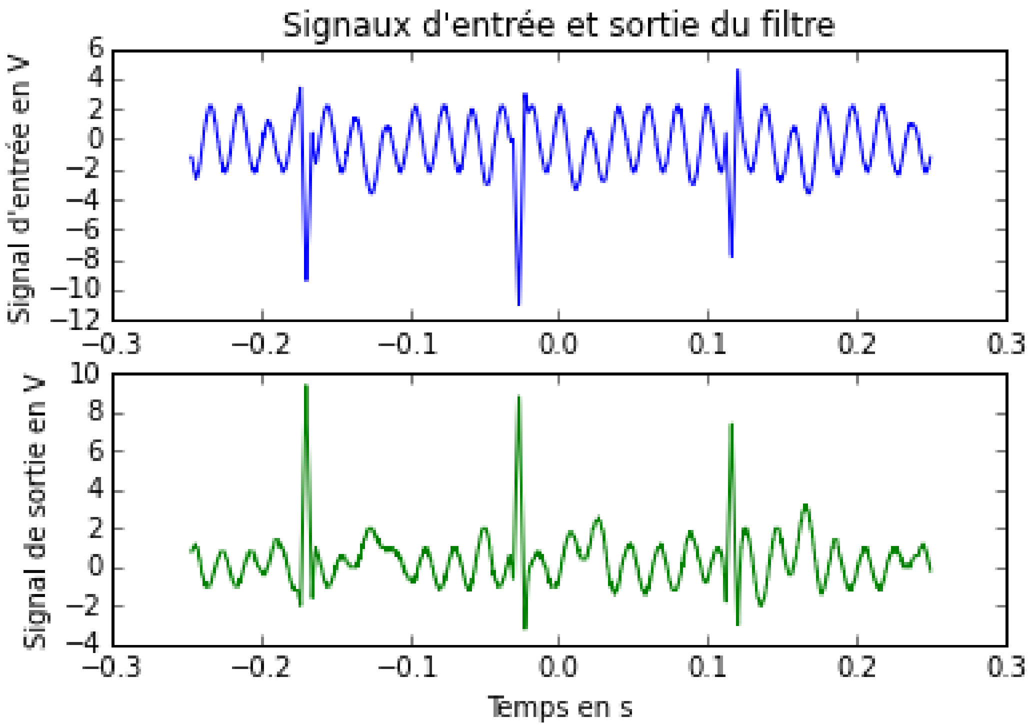
Résultats expérimentaux:

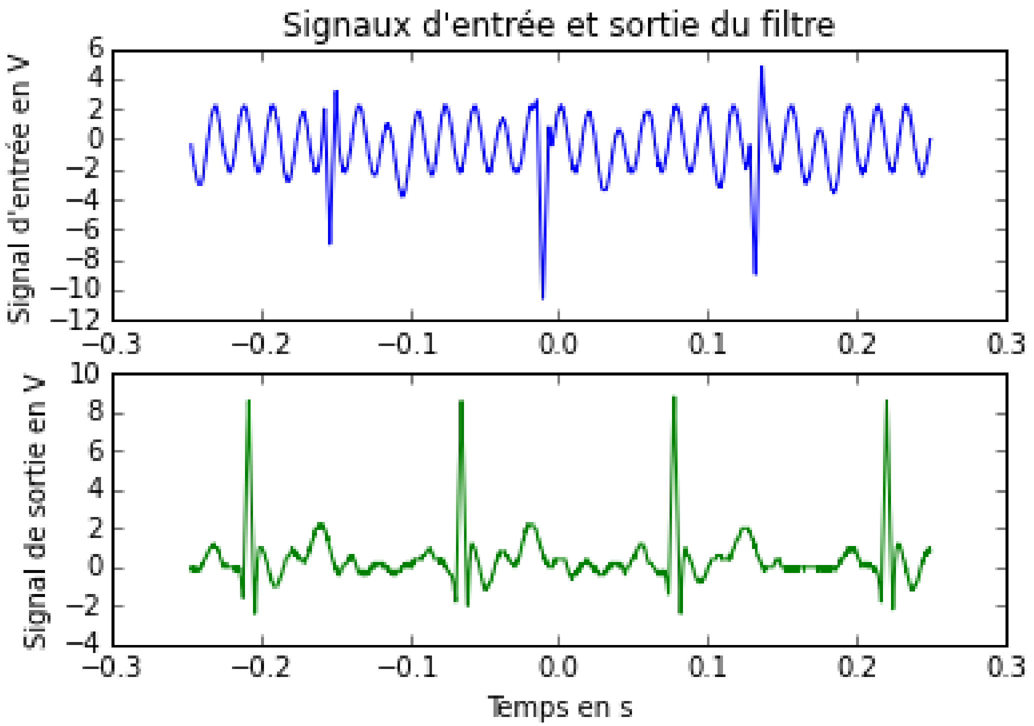




Fréquence du bruit: 53 Hz

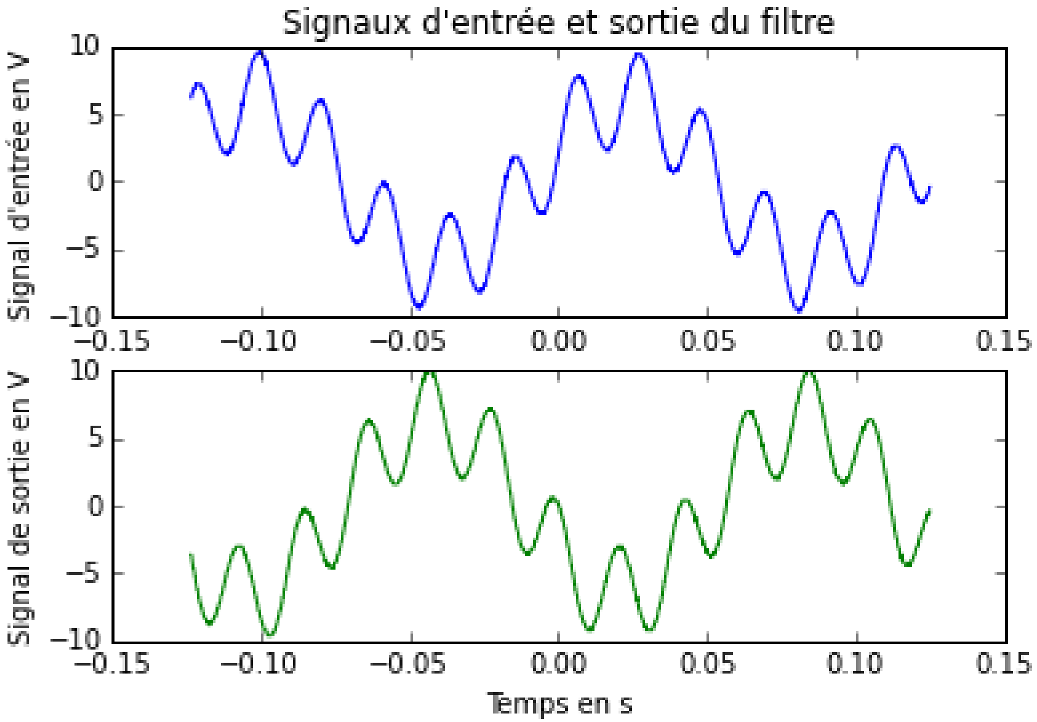
Résultats expérimentaux:

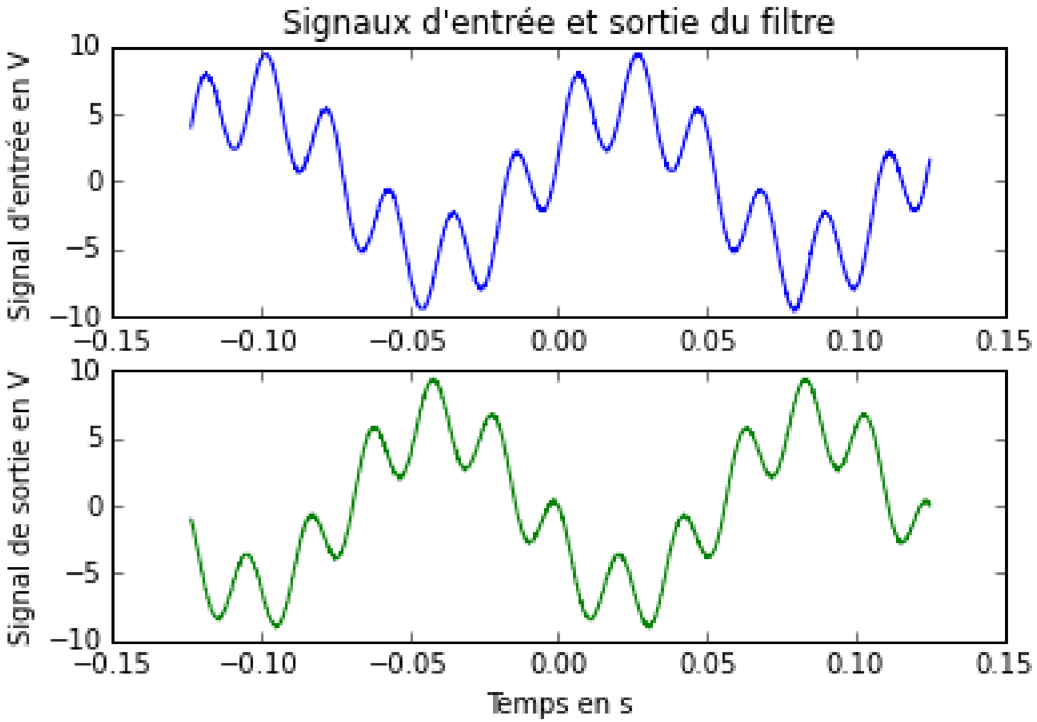




Fréquence du bruit: 52 Hz

Résultats expérimentaux:

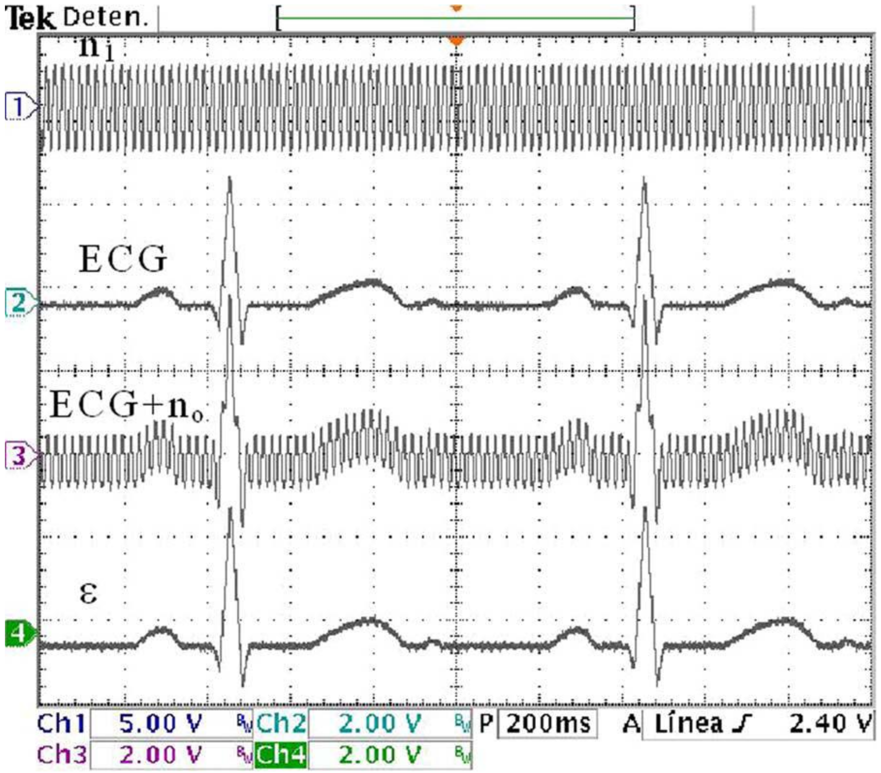




Fréquence du bruit: 55,5 Hz

Conclusion

- Réalisation de l'objectif
- Limites du modèle
- Contraintes retenues
- Pertinence du système



Annexe 1

```python
# -*- coding: utf-8 -*-
"""
Created on Mon Jun  5 11:51:46 2017

@author: youss
"""

from matplotlib.pyplot import *
from numpy import *
from pylab import *

R1=200e3
R2=200e3
#R3=153 #variable
R4=14.7e3
R5=14.7e3
R6=14.7e3
R7=14.7e3
R8=14.7e3
C=215e-9

def H1(omega):
    R3=14.9e3
    return ((1/R1)-R5/(R2*R6)-1j*(R4*C/R6)*omega+1j/(R7*R3*C*omega))/(R4/(R2*R8)+1j*(R4*C/R8)*omega-1j*(1/(R3*R5*C*omega)))

def H2(omega,f):
    R3=(75.56-f)/0.0017
    return ((1/R1)-R5/(R2*R6)-1j*(R4*C/R6)*omega+1j/(R7*R3*C*omega))/(R4/(R2*R8)+1j*(R4*C/R8)*omega-1j*(1/(R3*R5*C*omega)))

puissance_omega=arange(10,100,0.01)
omega=10*puissance_omega
phase=angle(H1(omega),'deg')
module=20*log(absolute(H1(omega)))

subplot(211)
semilogx(omega,module)
ylabel("Gain en db")
grid(True)
plt.title('Module')

subplot(212)
semilogx(omega,phase)
xlabel("Pulsation en rad/s")
ylabel("Phase en rad")
grid(True)
plt.title('Phase')

show()
```

Annexe 2

```python
# -*- coding: utf-8 -*-
"""
Created on Mon Jun  5 12:36:07 2017

@author: youss
"""

import matplotlib.pyplot as plt
from scipy.stats import linregress as reglin

fh=open("53hz2.txt","r")
donnees = fh.read()
donnees=donnees.splitlines()
x=[]
y1=[]
y2=[]

points=[i for i in range(len(donnees))]
for el in donnees:
    a=el.split(",")
    x.append(float(a[3]))
    y1.append(float(a[4]))
    y2.append(float(a[10]))

plt.subplot(211)
plt.plot(x,y1)
plt.xlabel("Temps en s")
plt.ylabel("Signal d'entrée en V")

plt.title("Signaux d'entrée et sortie du filtre ")

plt.subplot(212)
plt.plot(x,y2,'g')
plt.xlabel("Temps en s")
plt.ylabel("Signal de sortie en V")

plt.show()
```