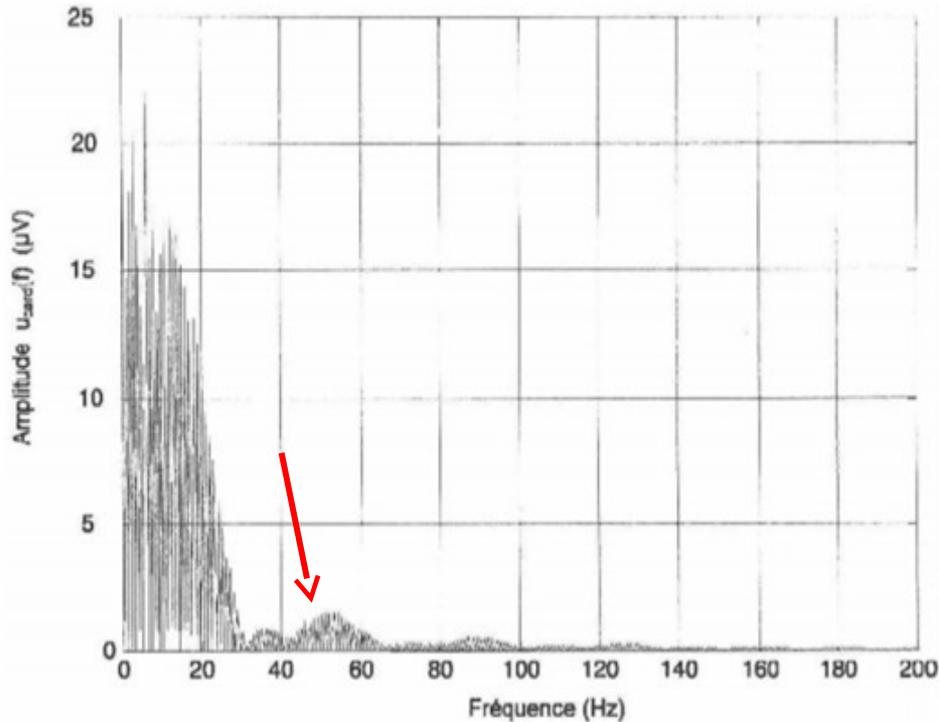


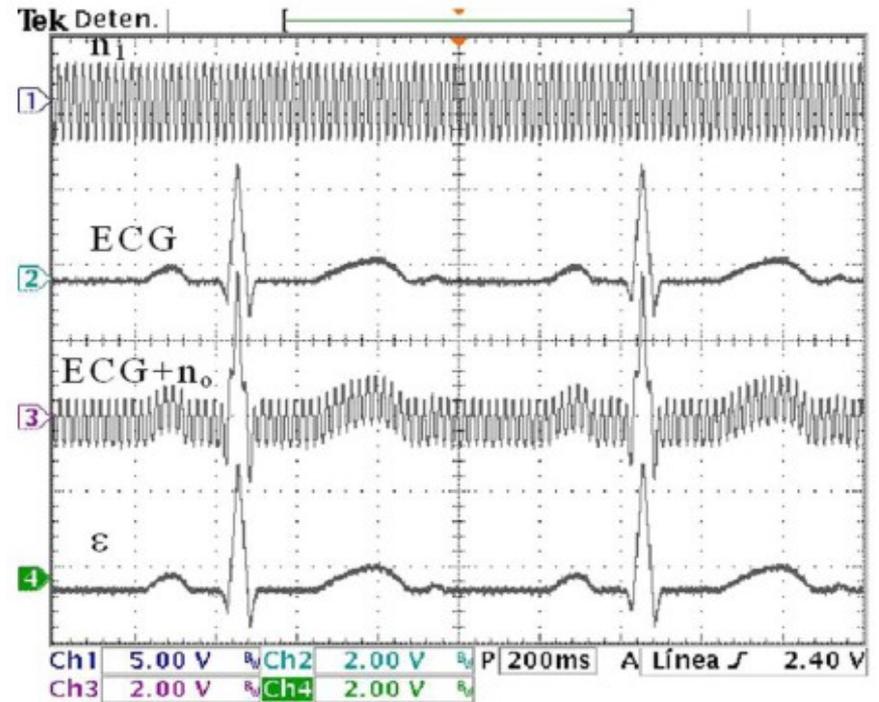
Atténuation du bruit en électrocardiographie



Problématique



Spectre du signal ECG



Signal ECG

Sommaire

- 1-Présentation du cadre de l'étude
- 2-Recherche de la résistance variable
- 3-Asservissement
- 4- Résultats expérimentaux et conclusion

1-Filtre « notch »

Topologie du circuit

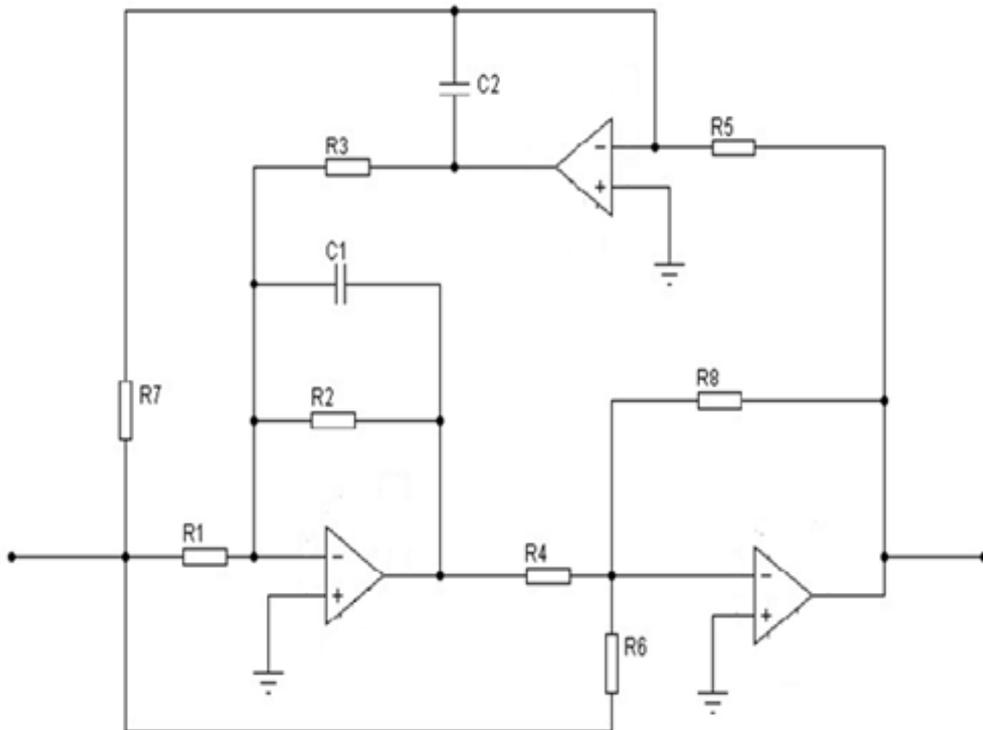
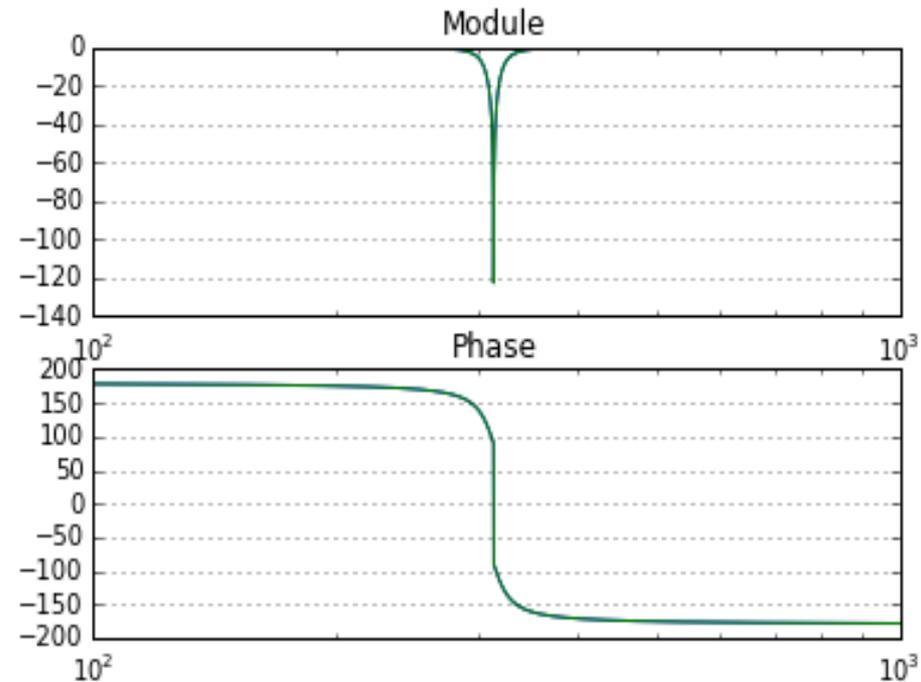


Diagramme de bode



Caractéristiques

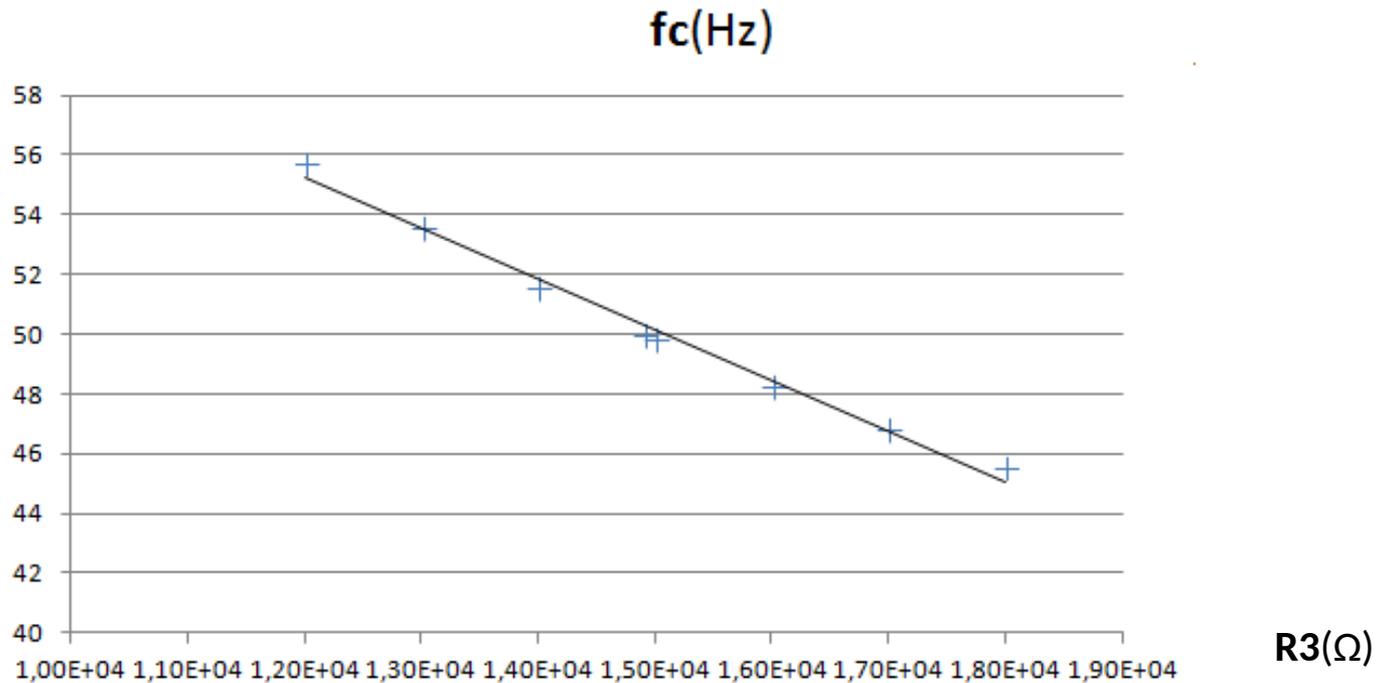
Filtre du 4^{ème} ordre

$\omega_0=314,15$ rad/s

Q=10

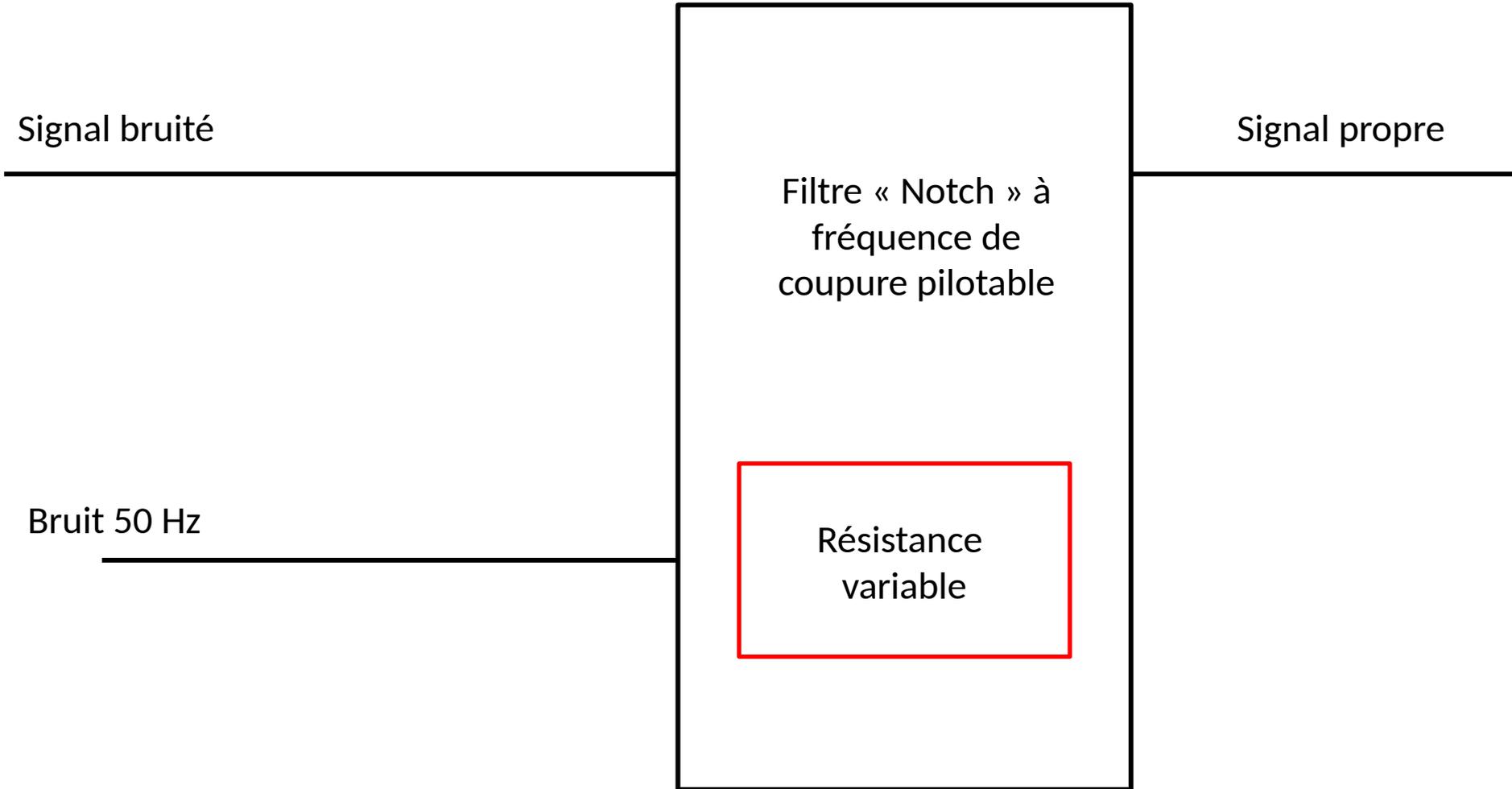
BP<5Hz

Courbe de R3 en fonction de la fréquence de coupure du filtre



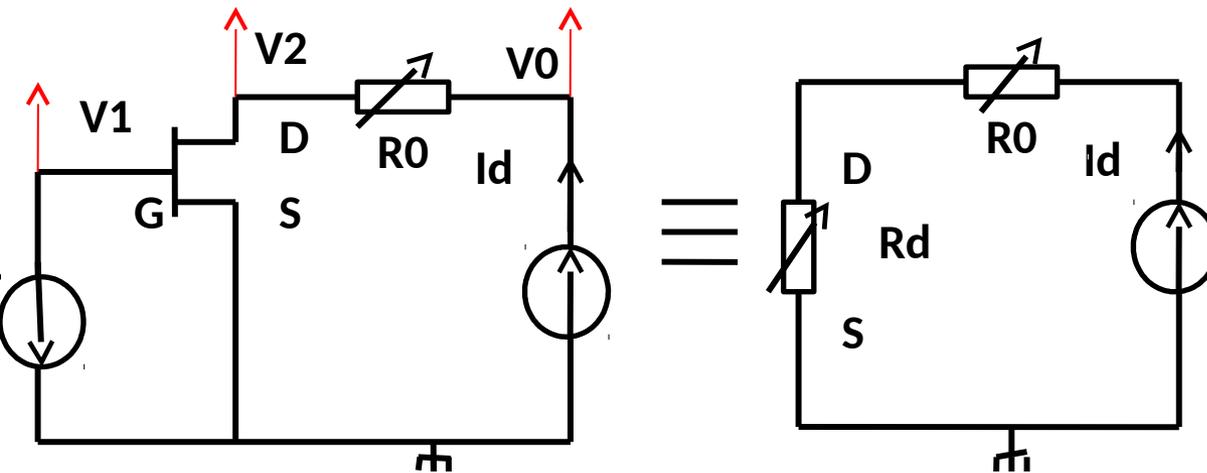
$$fc = -1,7 \cdot 10^{-3} \cdot R_3 + 75,56$$

1-Modélisation

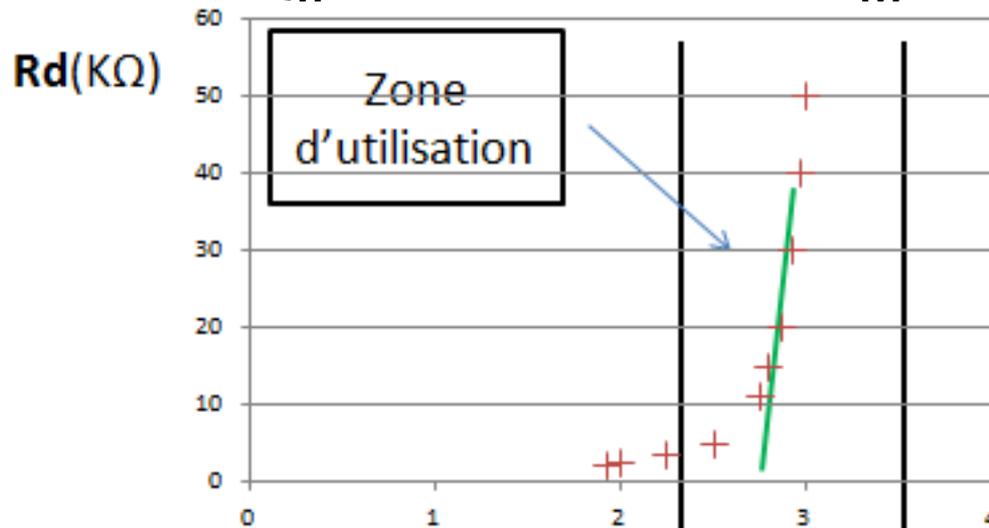


2-Transistors à effet de champ

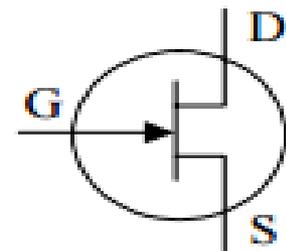
Protocole expérimental pour tracer la caractéristique



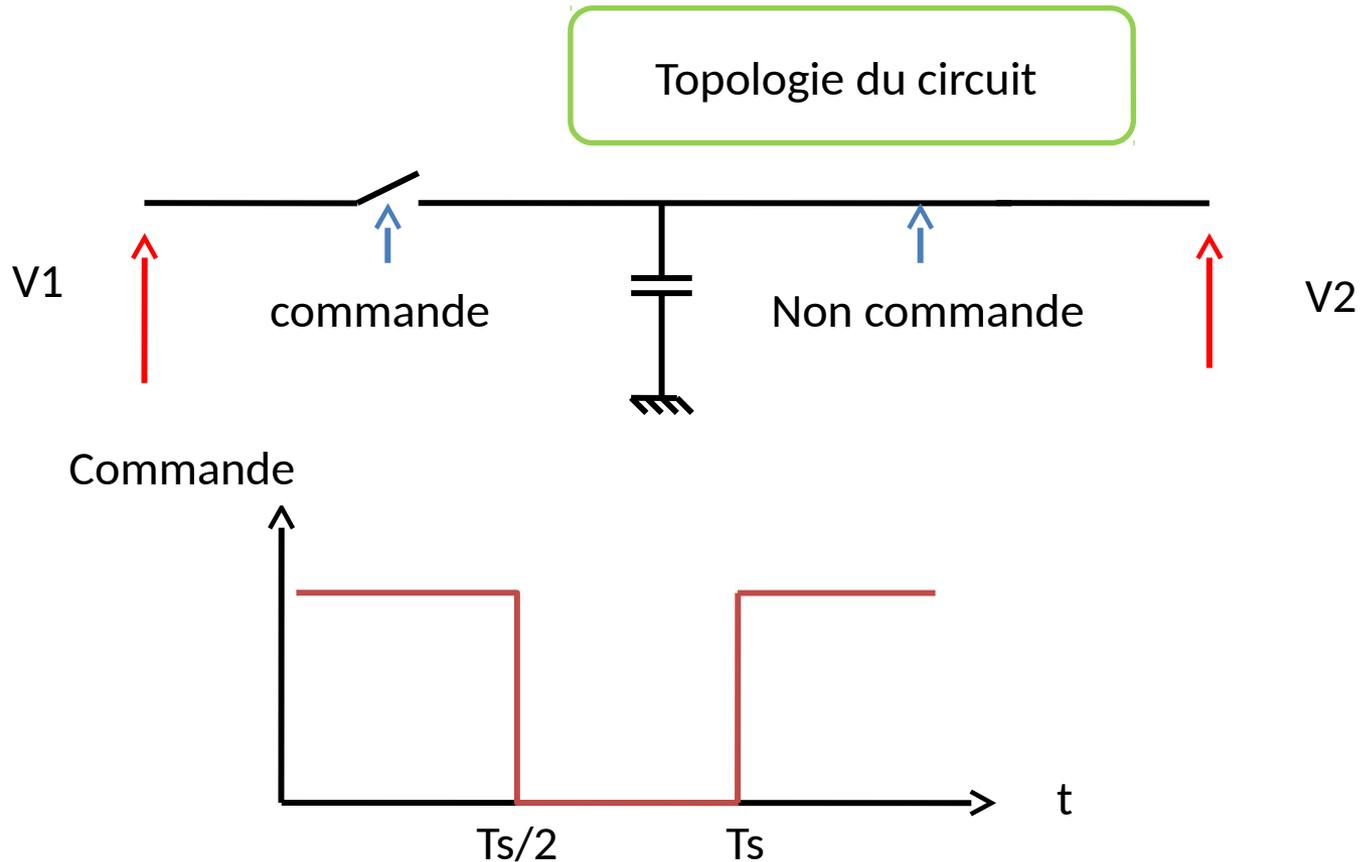
$$R_d = \frac{V_2}{V_2 - V_0} * R_0$$



+série2



2-Circuit à capacité commutée (Parallèle)



-Sur l'intervalle $[0, Ts/2]$ interrupteur 1 est fermé et interrupteur 2 est ouvert

-Sur l'intervalle $[Ts/2, Ts]$ interrupteur 2 est fermé et interrupteur 1 est ouvert

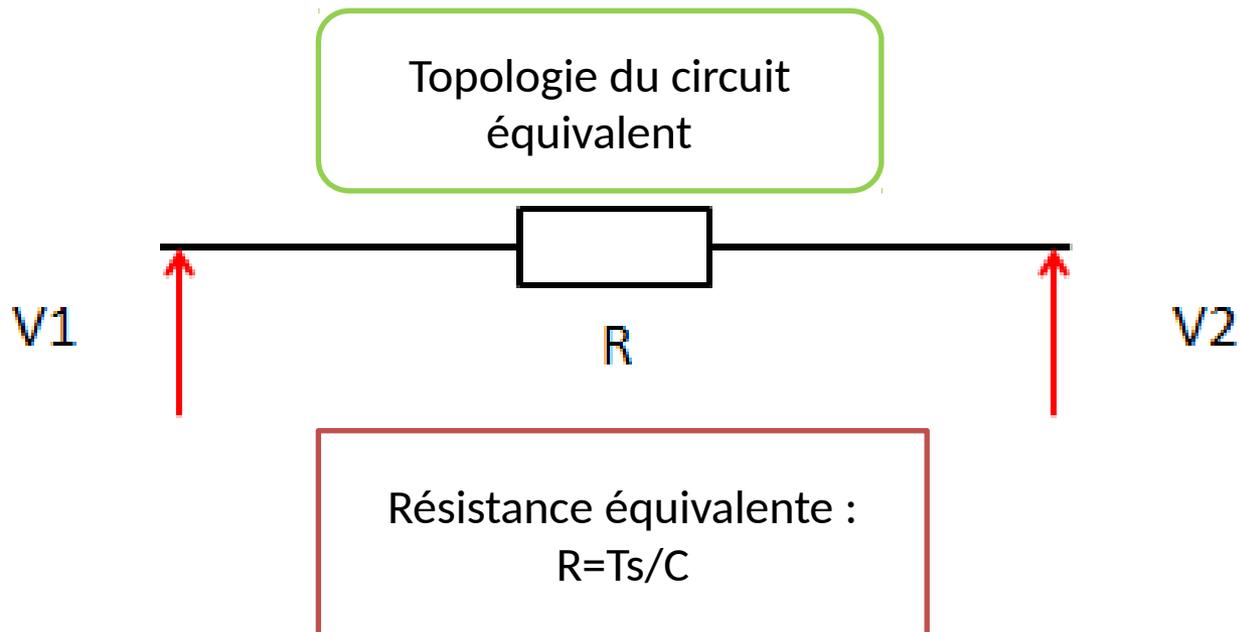
On a sur $[0, T_s/2]$: $V_1 = Q_1/C$
et sur $[T_s/2, T_s]$: $V_2 = Q_2/C$

Donc on a un transfert de charge: $\Delta Q = C.(V_2 - V_1) = C.V_e$

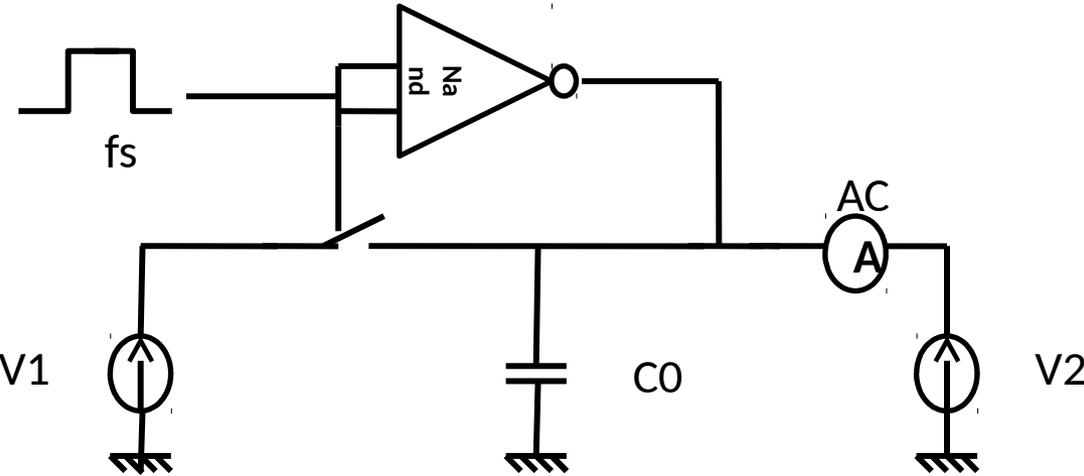
Le courant qui correspond a ce transfert de charge: $I = \Delta Q/T_s = (C.V_e)/C$

En moyenne , on a : $V_e = R.I$

Donc: $R = T_s/C$



Protocole expérimental pour tracer R en fonction de la fréquence des interrupteurs

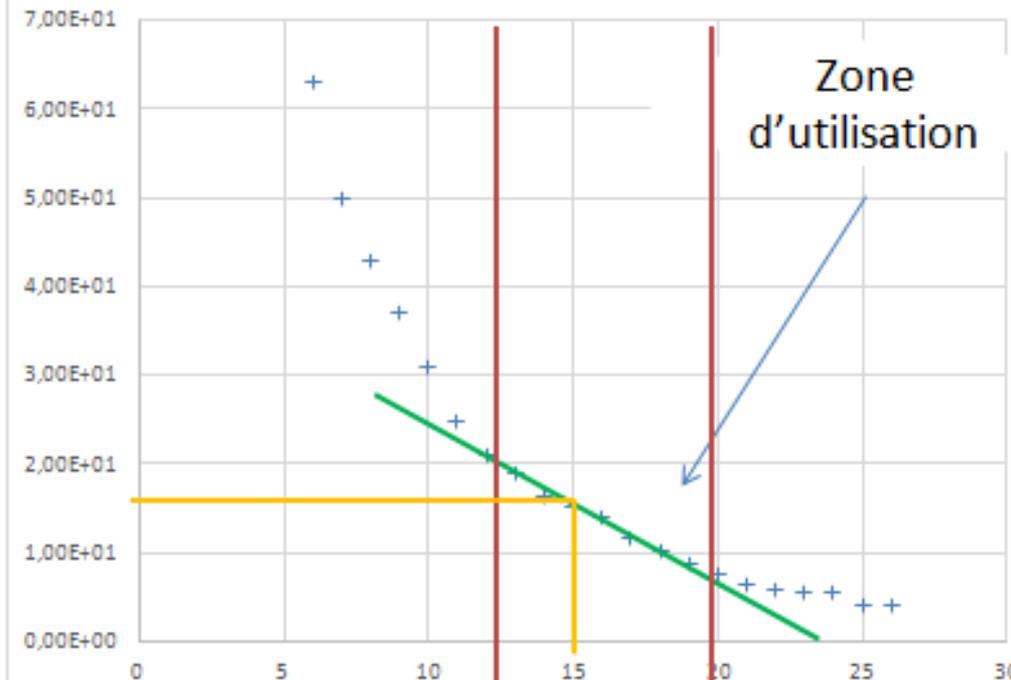


f_s : fréquence de l'horloge

Caractéristique du circuit à capacité commutée

$R_{exp}(K\Omega)$

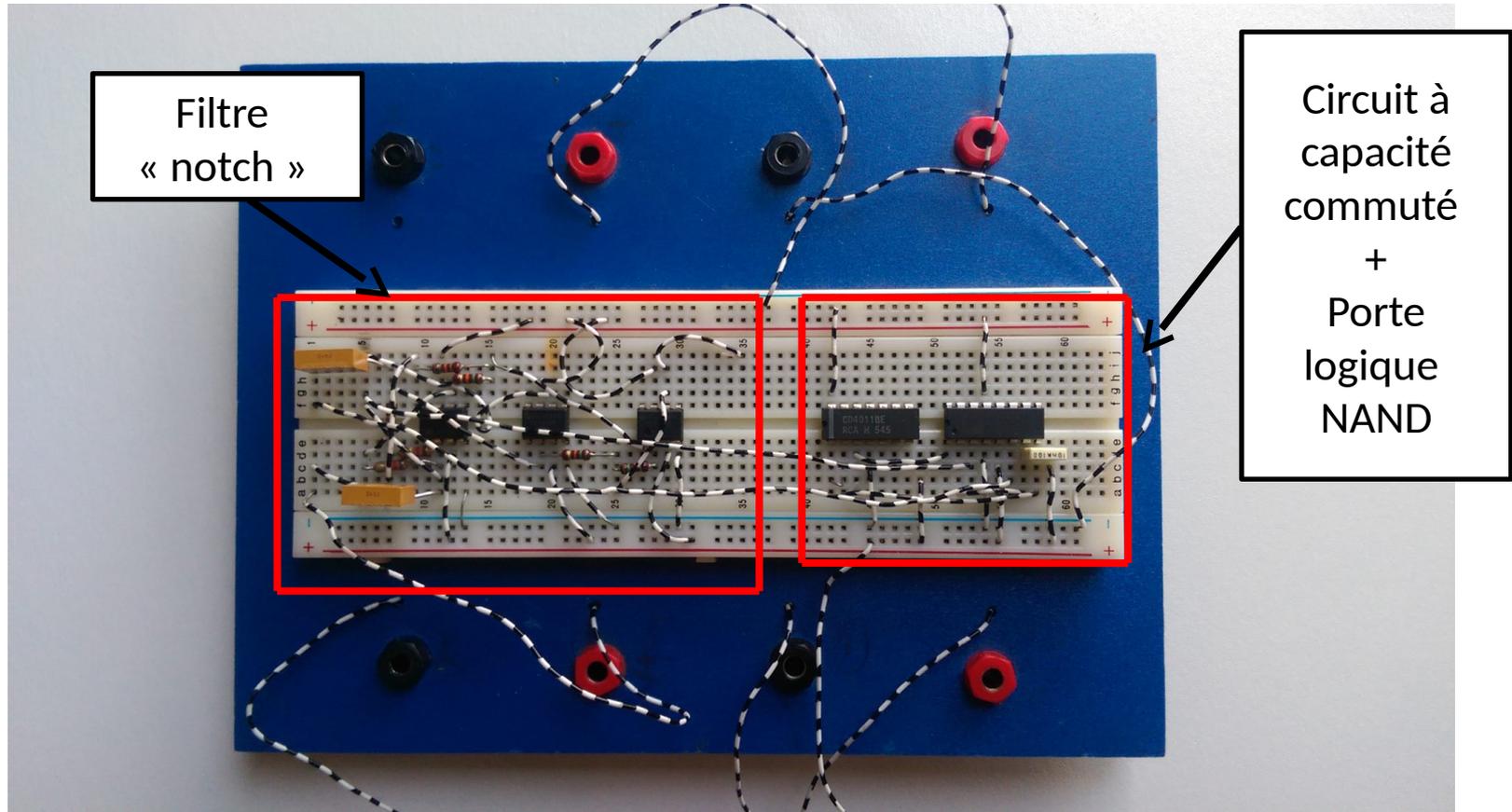
$$R_{exp} = \frac{V1 - V2}{I}$$



$$R3 = -0,307 * \Delta f_s$$

$f_s(\text{Hz})$

2-Filtre avec résistance variable

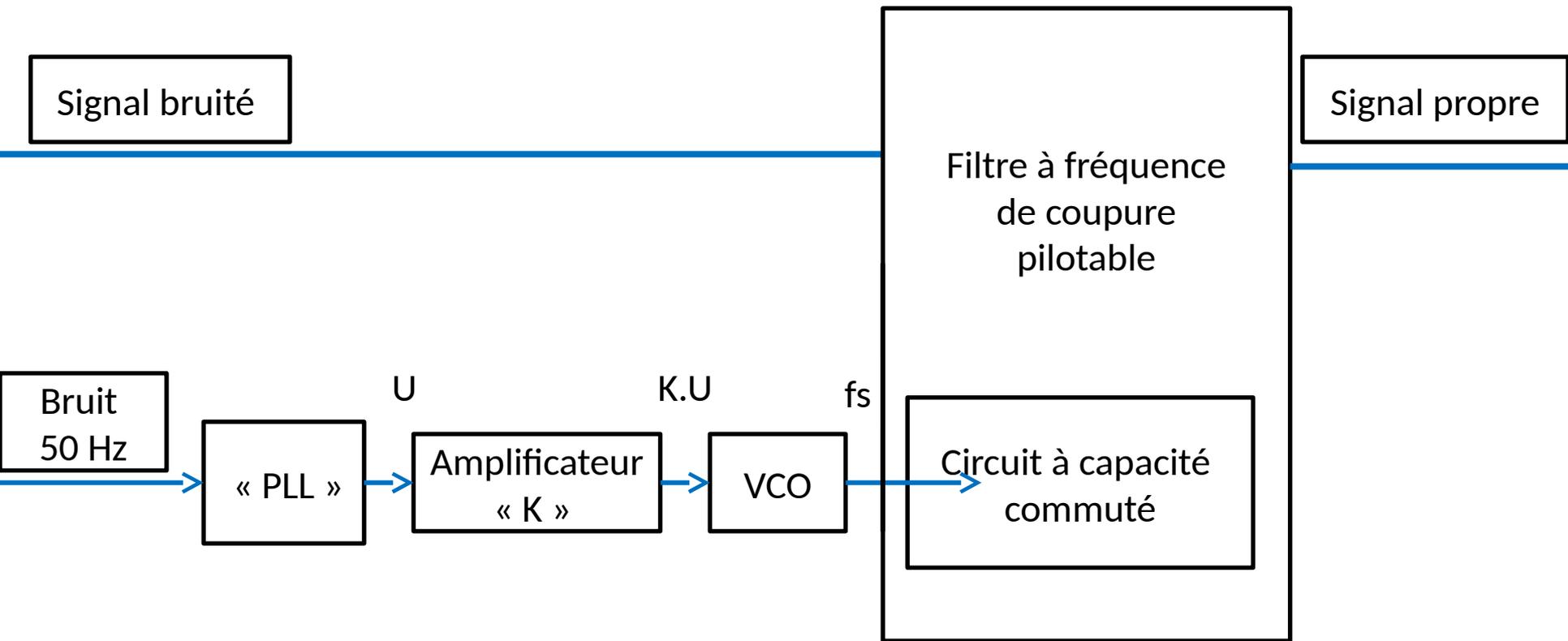


Signal
bruité

Filtre « notch »
Avec résistance variable

Signal
propre

3-Asservissement



Boucle à verrouillage de phase « PLL »

Schéma d'une boucle à verrouillage de Phase

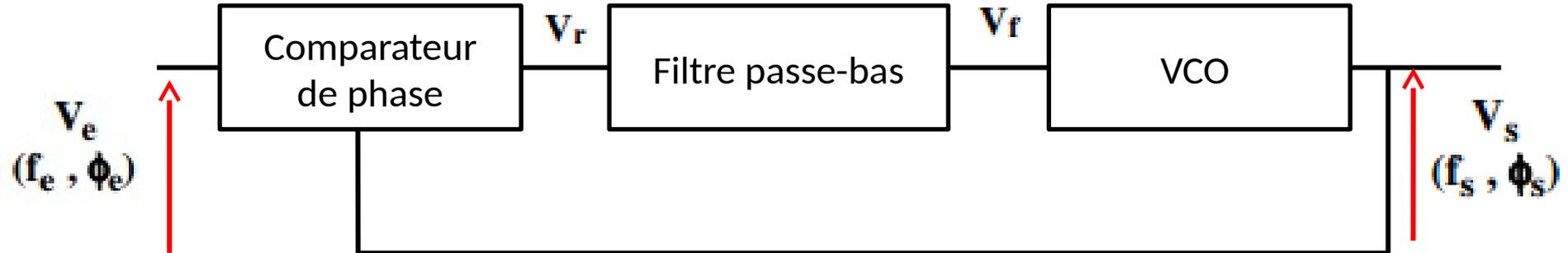
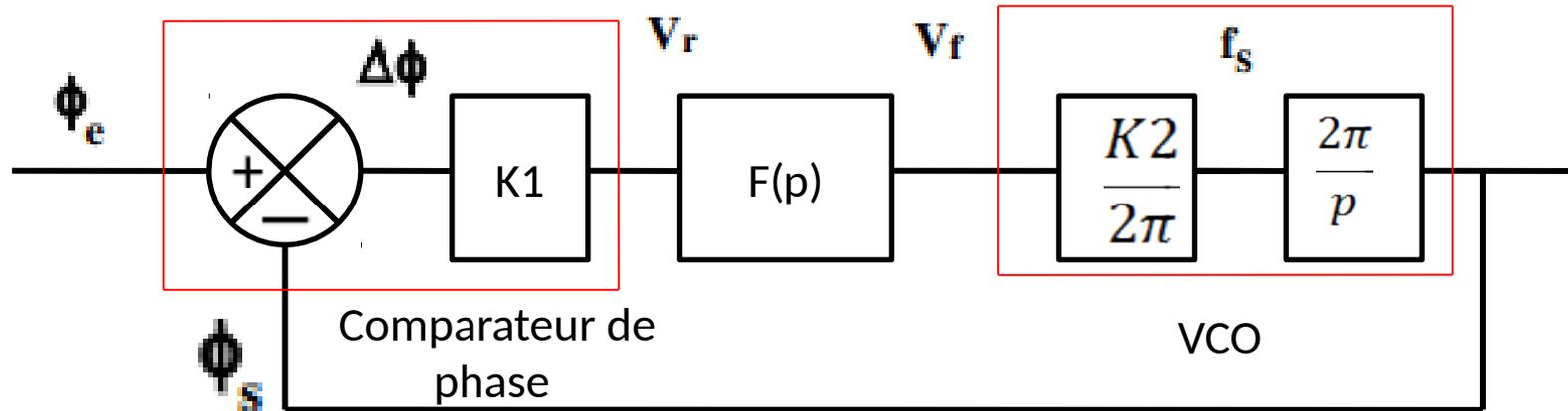
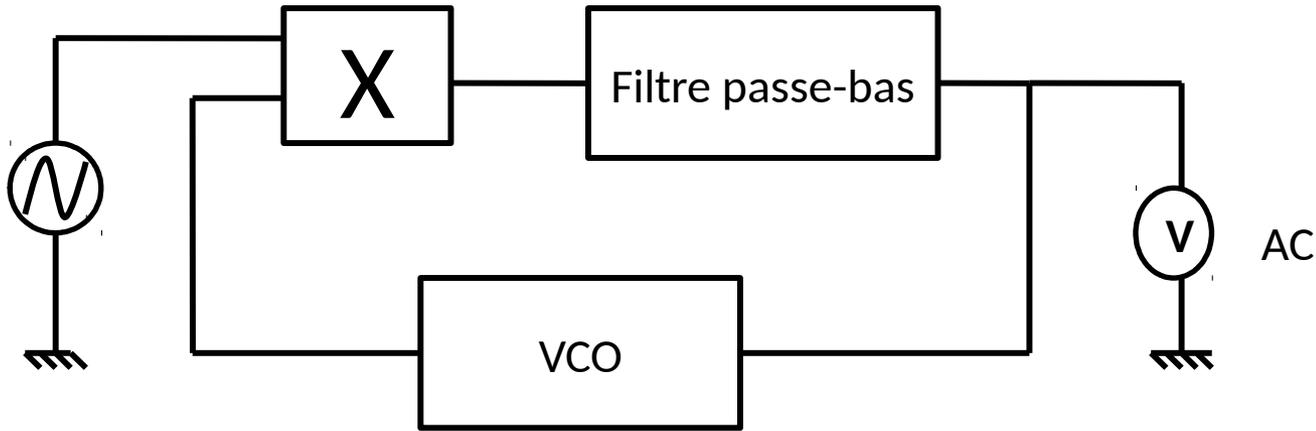


Schéma bloc de la PLL en grandeur phase

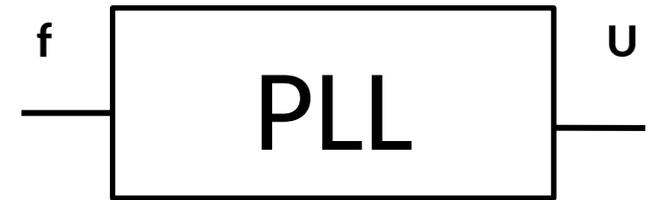
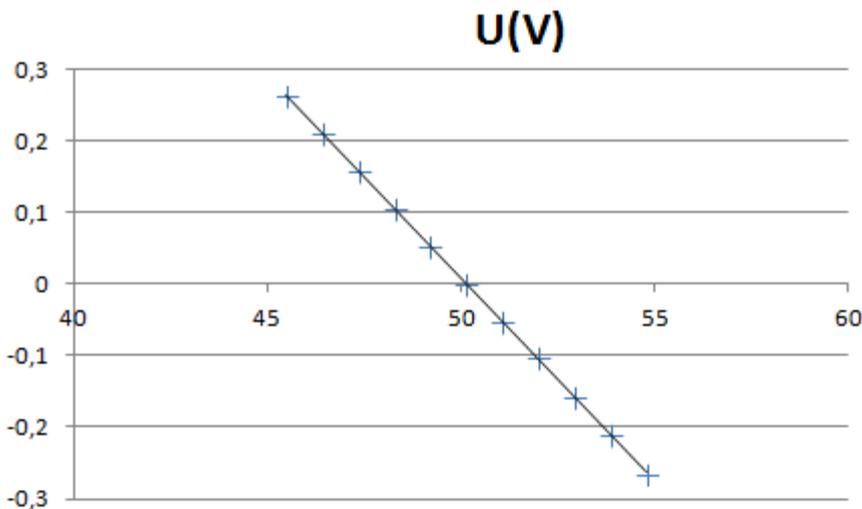


Protocole expérimentale pour tracer la caractéristique de la « PLL »



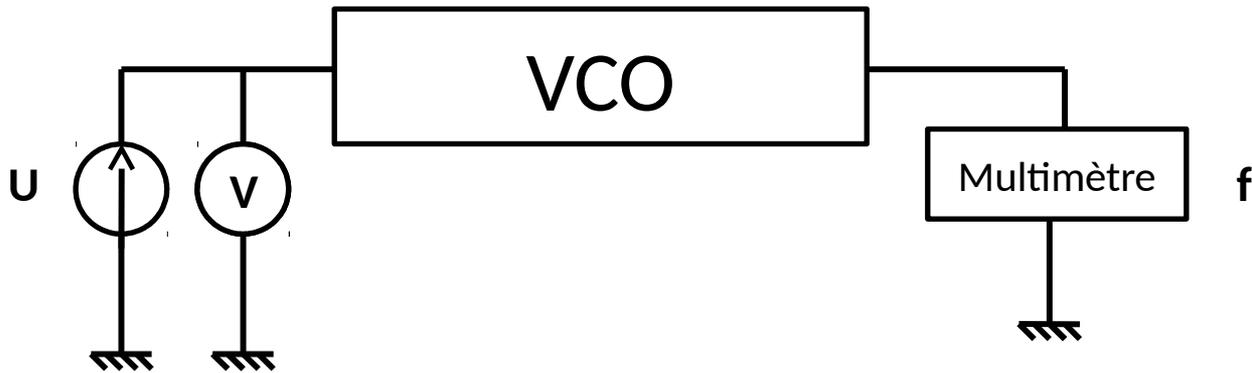
$f_c = 5\text{Hz}$
 Avec :
 f_c
 la fréquence
 de coupure
 du filtre
 passe-bas

Caractéristique de la « PLL »

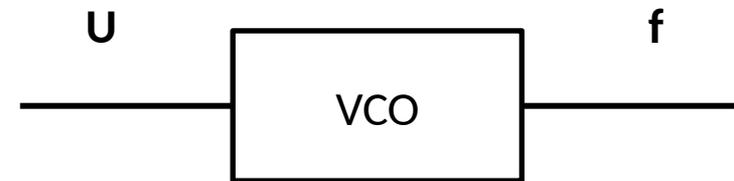
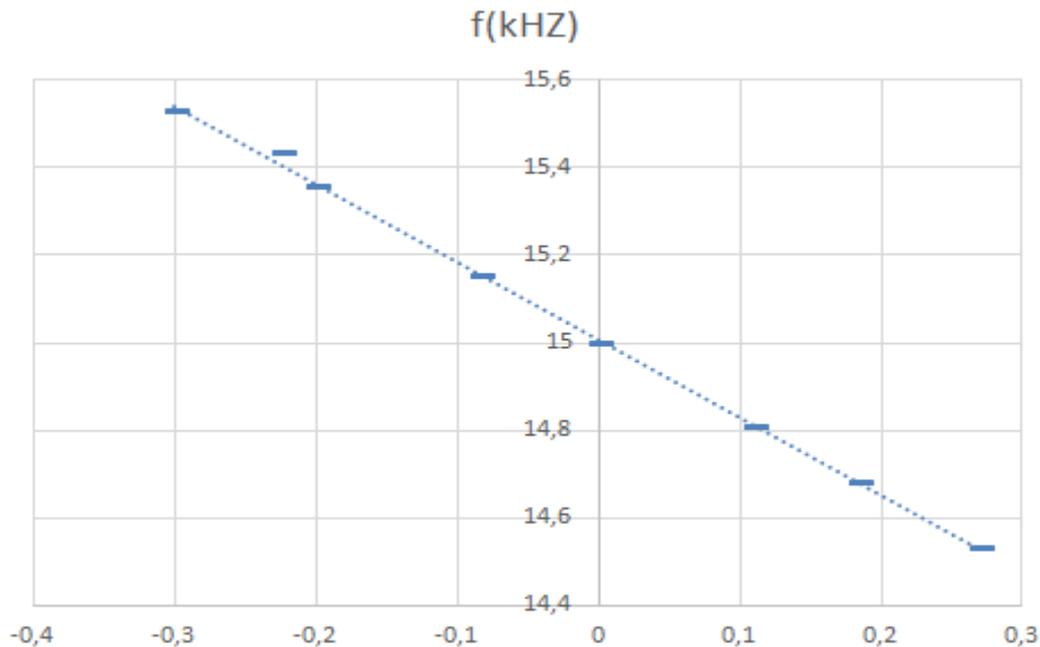


$$U = -0,056 * \Delta f$$

Protocole expérimentale pour tracer la caractéristique du VCO

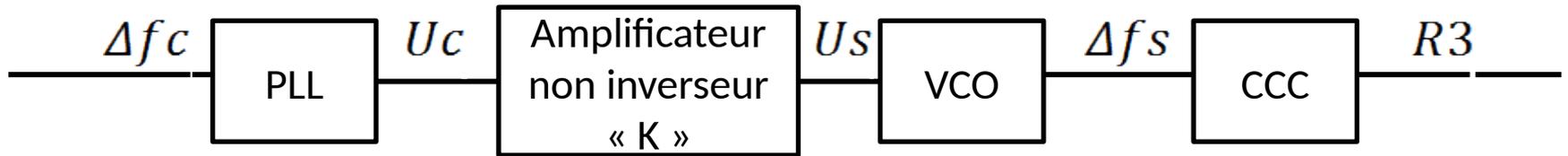


Caractéristique du VCO en 15 KHz



$$\Delta f = -1,776 * U$$

U(V)



$$K = \frac{U_s}{U_c}$$

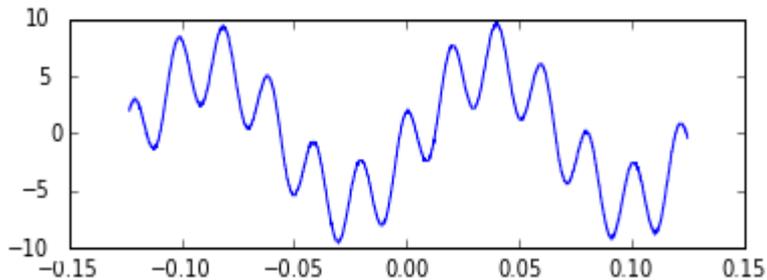
Afin d'ajuster la pente de la caractéristique de la « PLL » avec celle du « VCO » on a :

$$K=2,79$$

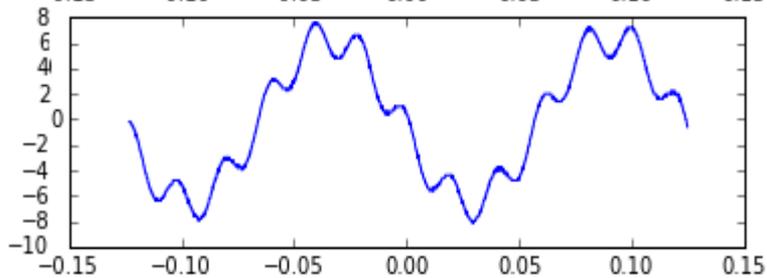
4-Résultats expérimentaux

Fréquence du bruit 49Hz

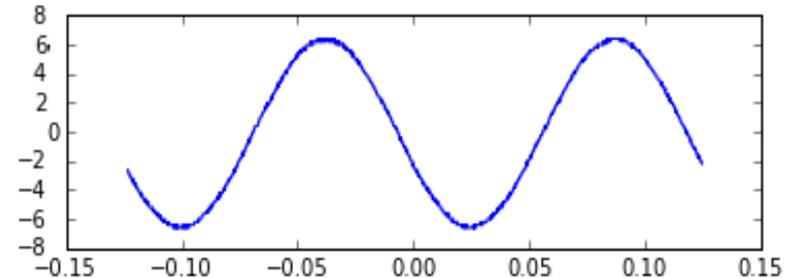
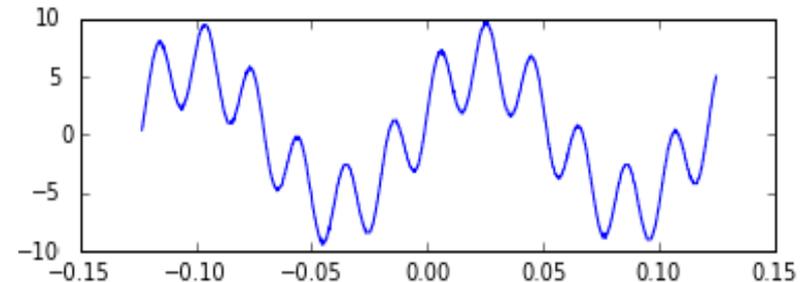
Amplitude en
entrée (V)



Amplitude en
sortie (V)

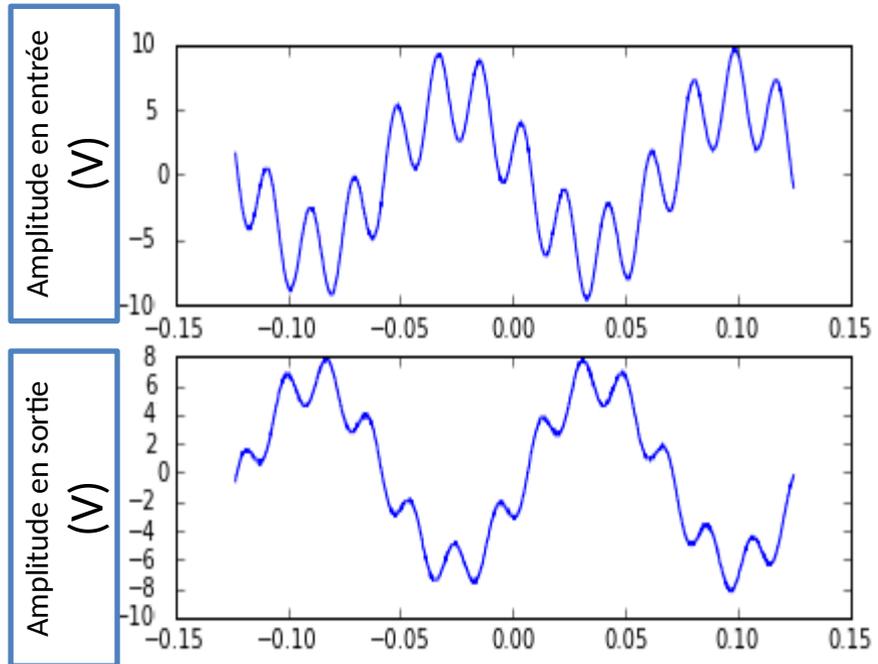


Filtre « notch »

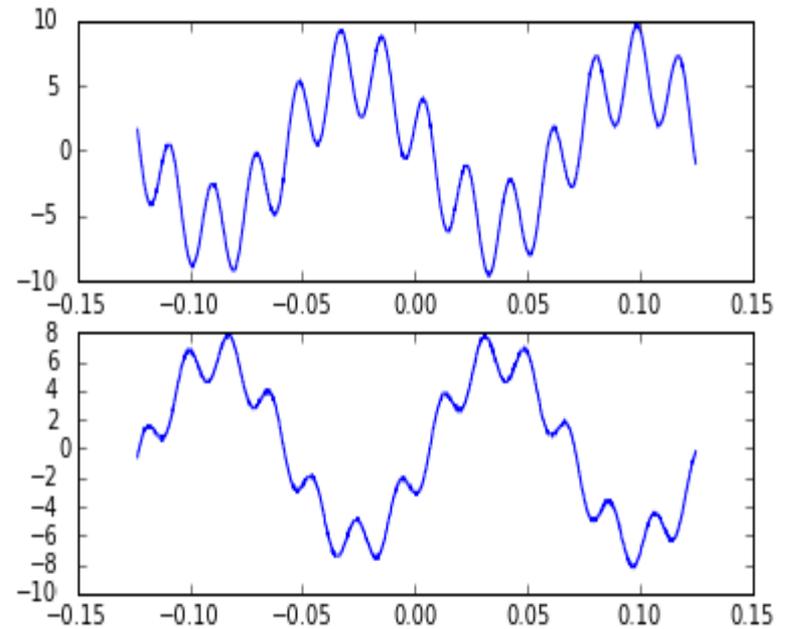


Filtre « notch » avec fréquence
de coupure variable

Fréquence du bruit 55 Hz

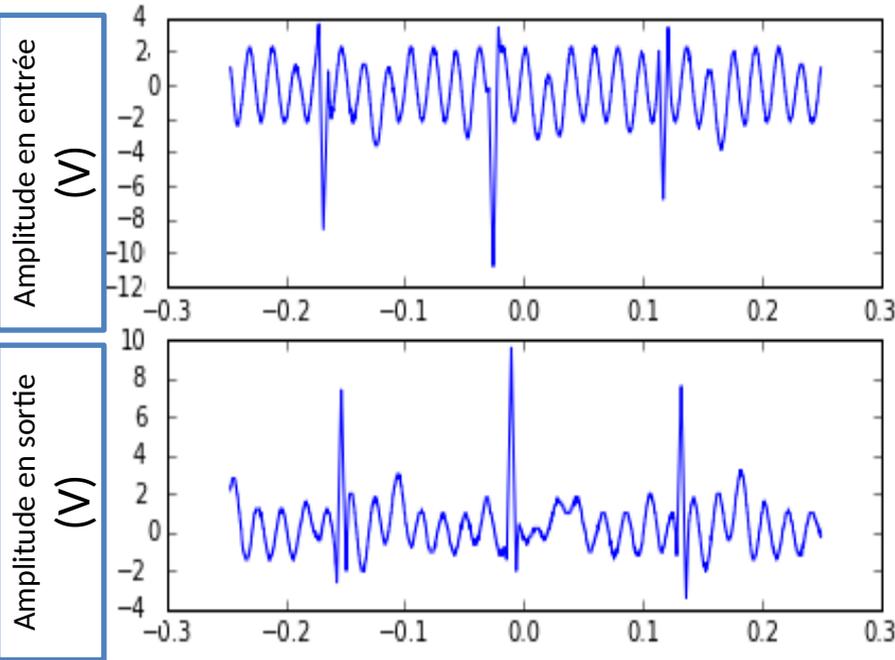


Filtre « notch »

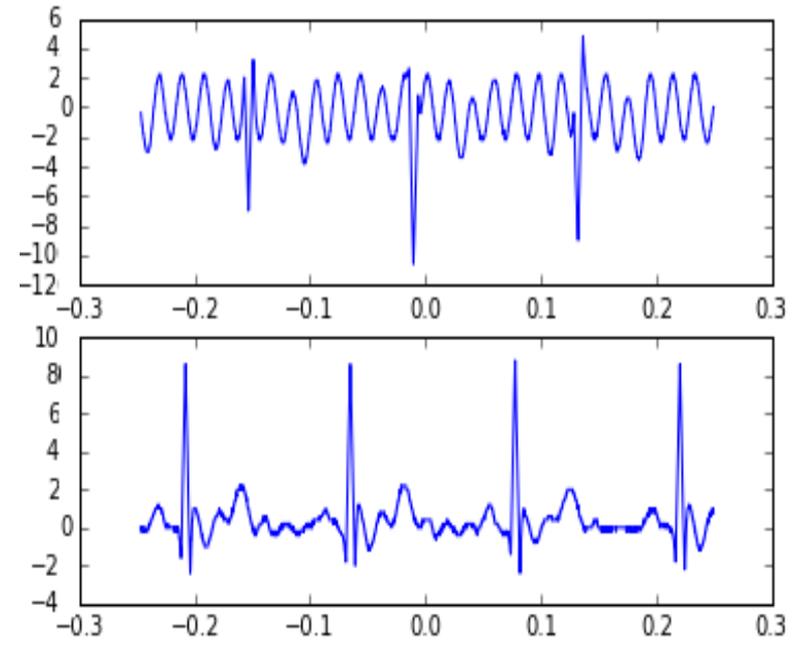


Filtre « notch » avec fréquence de coupure variable

Fréquence du bruit 51Hz



Filtre « notch »



Filtre « notch » avec fréquence de coupure variable

CONCLUSION

- Limites du modèle
- Relation avec le thème
- Solution dans le système réel

Annexe 1

```
8 from matplotlib.pyplot import *
9 from numpy import *
10 from pylab import *
11
12 R1=200e3
13 R2=200e3
14 R3=14.9e3 #variable
15 R4=14.7e3
16 R5=14.7e3
17 R6=14.7e3
18 R7=14.7e3
19 R8=14.7e3
20 C=215e-9
21 f1=58
22 f2=50
23 def fct_transfert(omega):
24     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)))
25 def H1(omega,f):
26     R3bis=(75.56-f)/0.0017
27     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)))
28 puissance_omega=arange(10,100,0.01)
29 omega=10*puissance_omega
30 phase1=angle(H1(omega,f1),'deg')
31 module1=20*log(absolute(H1(omega,f1)))
32 phase2=angle(H1(omega,f2),'deg')
33 module2=20*log(absolute(H1(omega,f2)))
34 subplot(211)
35 semilogx(omega,module1)
36 semilogx(omega,module2)
37 grid(True)
38 plt.title('Module')
39
40 subplot(212)
41 semilogx(omega,phase1)
42 semilogx(omega,phase2)
43 grid(True)
44 plt.title('Phase')
45 show()
```

Annexe 2



