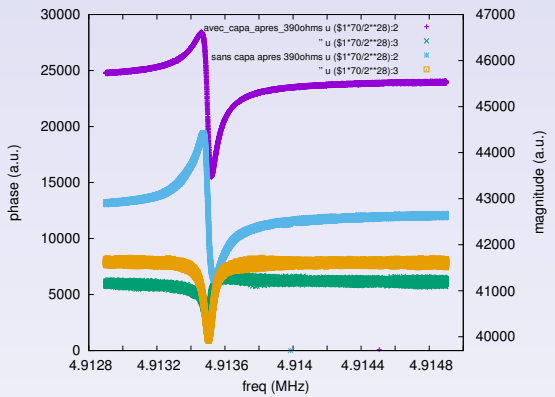
Analyseur de réseau RF embarqué

- Lecteur de capteurs passifs interrogeables sans fil SENSeOR
- STM32, synthèse numérique de fréquence (découpage pour impulsion RADAR), détecteur de puissance

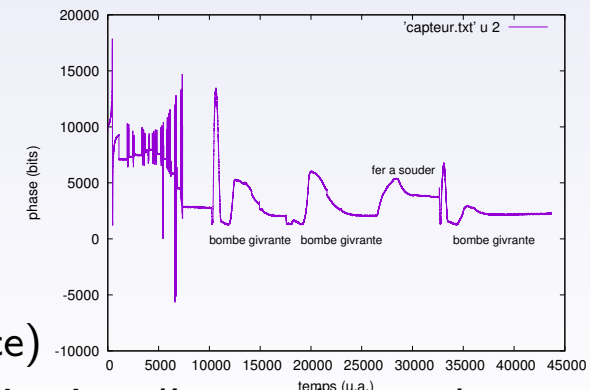
M1 :

- STM32 : microcontroleur 32 bits (application TPs microcontroleur 1er semestre)
- C ou FreeRTOS (environnements exécutifs – 2nd semestre)
- micro-acoustique : caractérisation de dispositifs radiofréquences propageant une onde élastique (résonateur à onde de volume – salle blanche)
- exploitation comme capteur : caractérisation de la fréquence de résonance en fonction de l'environnement (température, contrainte)



M2 :

- instrumentation (analyseur de réseau)
- micro-acoustique (TPs diapason & onde de surface)



Q. Macé, J.-M Friedt, *FreeRTOS : application à la réalisation d'un analyseur de réseau numérique sur STM32*, GNU/Linux Magazine France 207 (Sept. 2017)

Déroulement du projet

1. configuration et communication avec les périphériques (cf TPs 1er semestre)
2. balayage de fréquence et mesure d'amplitude et phase de la fonction de transfert d'un résonateur à quartz (affichage : gnuplot, octave ...)
3. interface de communication pour la configuration du microntrôleur (commandes, arguments : analyseur syntaxique)
4. interface Python sur PC : affichage (émission – cf instrumentation)
5. interface Python sur PC : commandes (réception – interruptions ?)
6. aspect capteur : asservissement de la fréquence sur un point de fonctionnement en phase
    1. identification en boucle ouverte de la dynamique du système
    2. identification des coefficients P et I de la boucle de rétroaction
    3. mise en œuvre de la PI(D) numérique
7. développement sous l'environnement exécutif Zephyr [1]

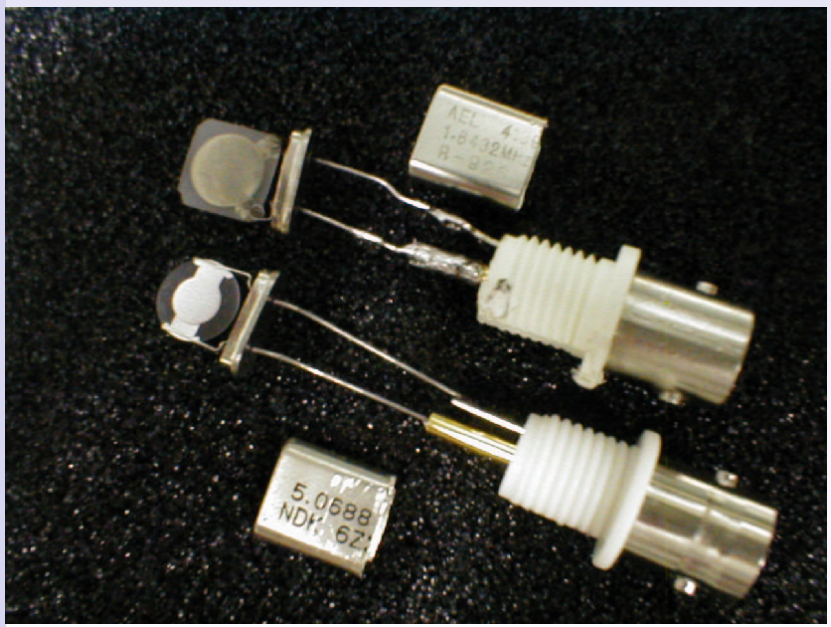
1. `https://www.zephyrproject.org/`

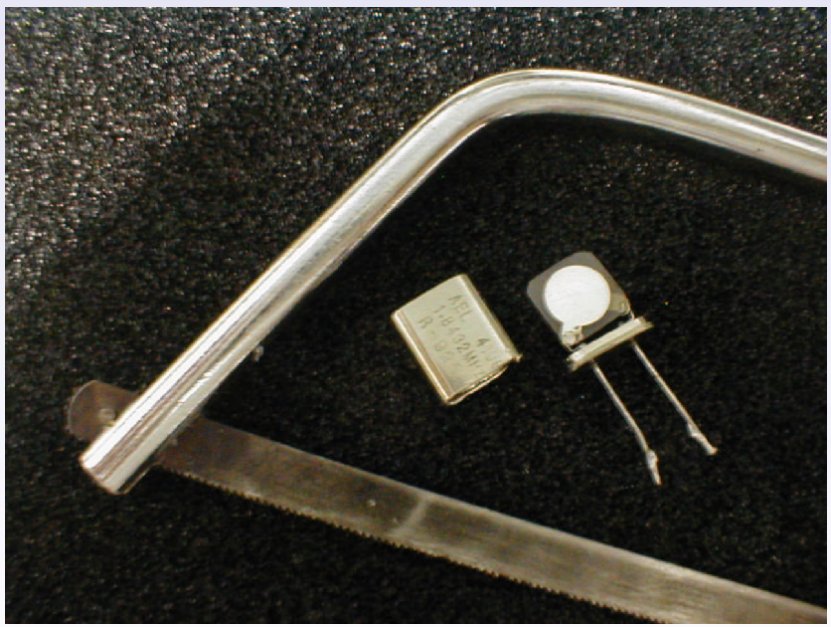
Objet de l'étude

Résonateur à quartz [2] : "micro" système électro-mécanique radiofréquence

- $f \in [4 - 20]$ MHz
- $Q \simeq 10^5$

$\Rightarrow$ pas de fréquence ?
$\Rightarrow$ attente entre deux mesures ?
$\Rightarrow$ nombre de points à acquérir ?
$\Rightarrow$ durée de la mesure

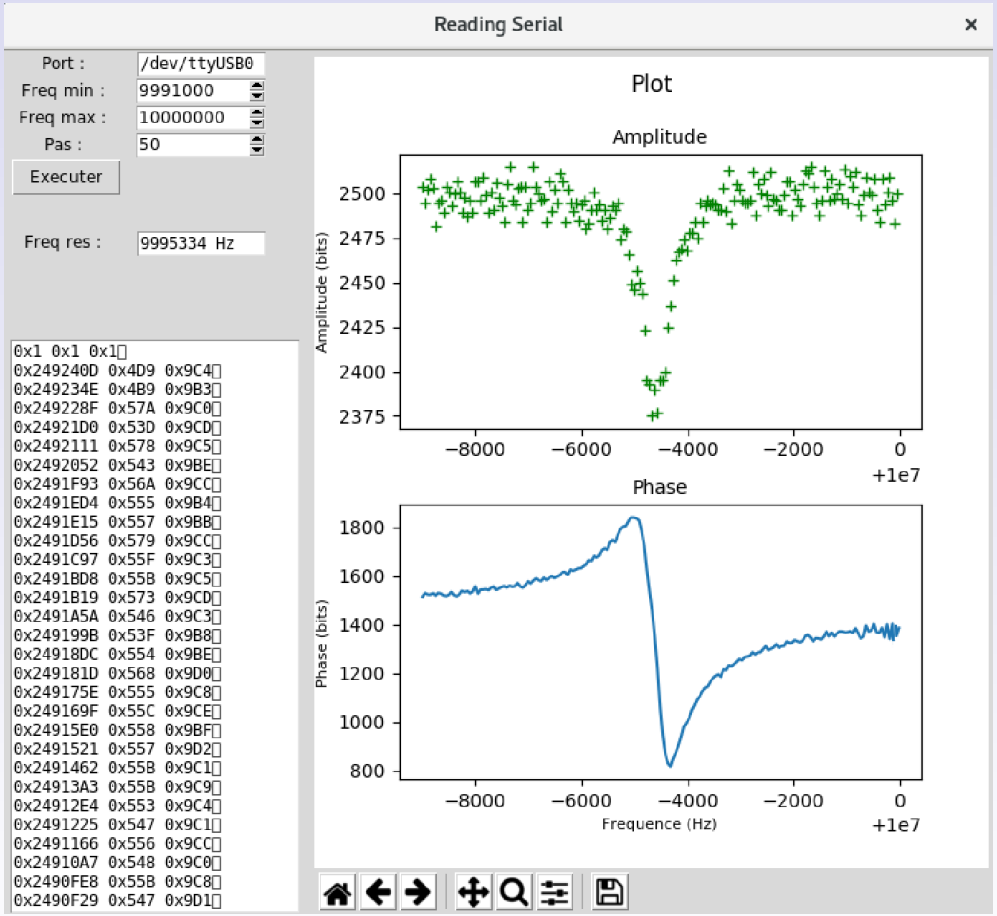




2. J.-M. Friedt, *Introduction à la microbalance à quartz : aspects théoriques et expérimentaux*, Bulletin de l'Union des Physiciens n.852 (Mars 2003), pp.429-440

Instrumentation

- Affichage au cours de la mesure des grandeurs observées (amplitude et phase v.s fréquence) – quel mode de communication efficace ?
- Interface utilisateur pour configurer les paramètres de mesure – lesquels ?



Rapport E. Altun & N. Texeira (2018)

Asservissement du point de fonctionnement

Boucle ouverte v.s boucle fermée (suivi d'une condition de phase en ajustant la fréquence)

1. ouverte :
   - suppose la bijection entre la grandeur observée (phase) et la grandeur mesurée (température) ...
   - et que cette loi reste constante au cours de la mesure.
   - Hypothèse localement valable, plage réduite de mesure.
2. fermée :
   - suivi d'une condition sur l'observable en agissant sur l'excitation ("oscillateur numérique")
   - identification de la relation phase-fréquence en boucle ouverte
   - asservissement de la fréquence pour maintenir phase consante
   - reporte l'hypothèse de bijection sur l'actuation et non sur l'observation
   - gamme de mesure étendue. [3]

3. D. Rabus, J.-M. Friedt, S. Ballandras, G. Martin, É. Carry, & V. Blondeau-Patissier, *A high sensitivity open loop electronics for gravimetric acoustic wave-based sensors*, IEEE Trans. UFFC **60** (6), 1219–1226 (2013)

Découpage du programme : asservir (1)

- Mise en œuvre d'une PID numérique en temps discret (quantification) : $\frac{dx}{dt} \rightarrow \frac{x_n - x_{n-1}}{T_e}$ et $X = \int x \cdot dt \rightarrow X += x_n \cdot T_e$
- $c(t) = K_p \cdot e(t) + K_i \int e(t) + K_d \frac{de(t)}{dt}$
- choix des paramètres : K_d et K_i nuls, K_p croît jusqu'à osciller.
- K_i pour rendre l'erreur statique nulle
- K_d pour augmenter la dynamique et réduire les dépassements à la consigne

```
erreur=0;
integrale=0;
erreur_avant=0;
while(1) {
  mesure(&freq_mesure);                                  // prend un temps dt
  erreur=consigne-freq_mesure;
  integrale+=erreur;                                     // *dt
  sature_integrale(&integrale);                          // anti-windup
  derivee=(erreur-erreur_avant);                         // /dt
  commande_pwm=Kp*erreur+Ki*integrale+Kd*derivee;        // PID
  sature_commande(commande_pwm);                         // rester dans la gamme
  erreur_avant=erreur;
}
```

Découpage du programme : asservir (2)

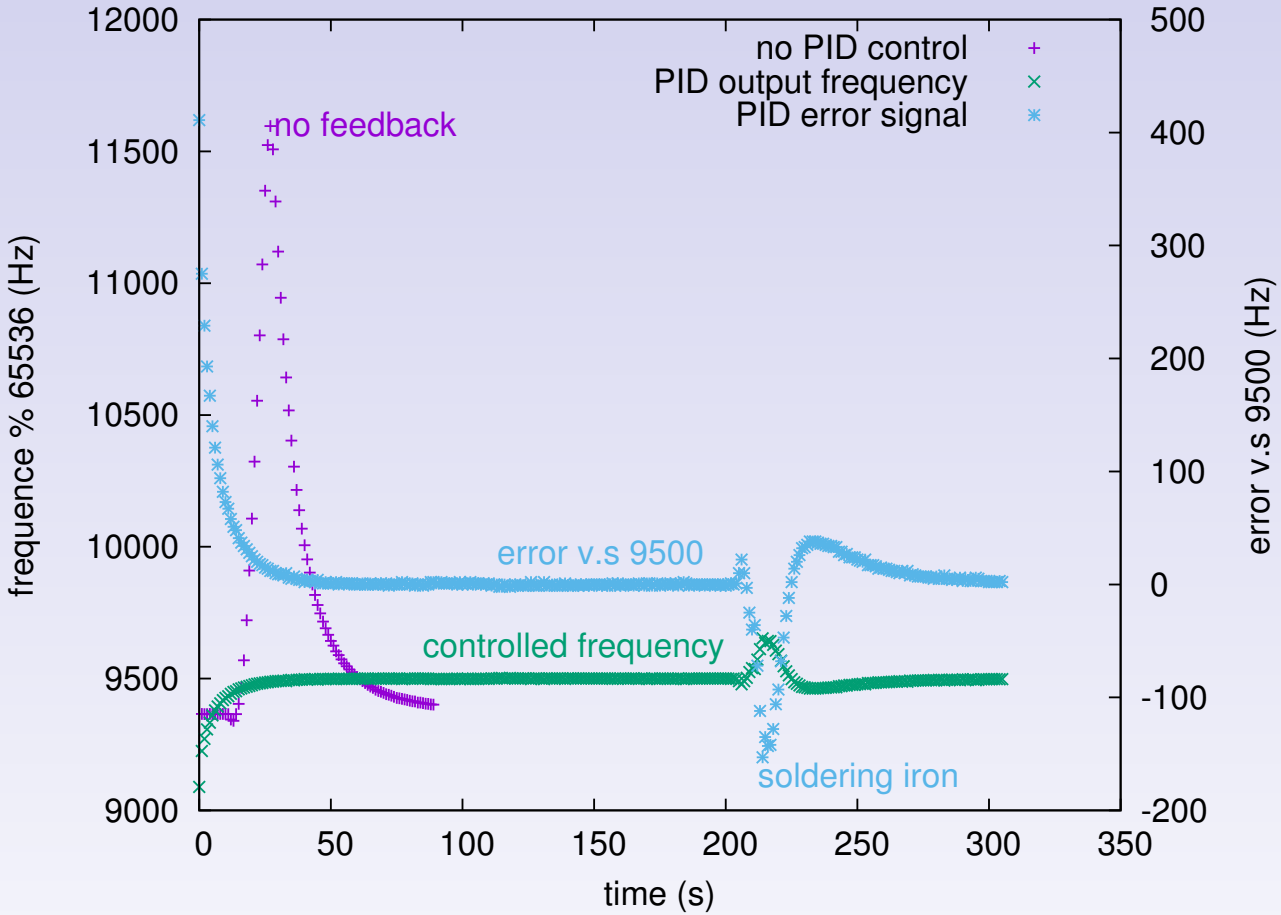
- évolution de $c(t)$ et en considérant (temps discret) la différence $c_{n+1} - c_n$
- loi récurrente de commande ...
- ... la somme infinie de l'intégrale se réduit à $K_i \cdot \varepsilon_{n+1}$
- les termes proportionnel et intégral sont $K_p \cdot (\varepsilon_{n+1} - \varepsilon_n) \cdot T_e$ et $K_d \cdot \frac{(\varepsilon_{n+1}-\varepsilon_n)-(\varepsilon_n-\varepsilon_{n-1})}{T_e} = K_d \cdot \frac{(\varepsilon_{n+1}-2\varepsilon_n+\varepsilon_{n-1})}{T_e}$.
- équation récurrente qui exprime la transformée en Z de la PID :

$$\underbrace{C(z)(z-1)}_{c_{n+1}-c_n} = E(z)\left(K_p \cdot (z-1) + K_i \cdot T_e + K_d \frac{z-2+z^{-1}}{T_e}\right)$$

$$\Leftrightarrow \frac{C(z)}{E(z)} = \frac{\alpha z + \beta + \gamma z^{-1}}{z-1} = \frac{\alpha + \beta z^{-1} + \gamma z^{-2}}{1 - z^{-1}}$$

```
err=0;
err_1=0;
err_2=0;
commande_pwm=0;
while(1) {
  mesure(&freq_mesure);                          // prend un temps dt
  err_2=err_1;
  err_1=err;
  err=consigne-freq_mesure;
  commande_pwm+=Kp*(err-err_1)+Ki*err+Kd*(err+err_2-2*err_1);
  sature_commande_et_integrale(&commande_pwm);
}
```

De la théorie à la pratique



Fréquence de l'oscillateur après chauffe puis refroidissement du quartz.

- Saturation de la commande
- Divergence de l'intégrale $\Rightarrow$ saturation de l'intégrateur ou stratégie d'*anti-windup*

Cas particulier de l'asservissement numérique

Caractérisation en boucle ouverte (gain a et retard L) pour en déduire les coefficients de la boucle d'asservissement [4]

Tableau 10 – Table de réglage de Takahashi

Régulateur	Coefficients de réglage	
	essai indiciel	essai fréquentiel
$u_k = K_p(r_k - y_k)$	$K_p = \frac{L}{a(L+T)}$	$K_p = 0{,}5k_0$
$u_k = u_{k-1} + K_p(y_{k-1} - y_k) + K_i T(r_k - y_k)$	$K_p = \frac{0{,}9L}{a(L+0{,}5T)} - 0{,}5K_iT$ $K_i = \frac{0{,}27L}{a(L+0{,}5T)^2}$	$K_p = 0{,}45k_0 - 0{,}5\ K_iT$ $K_i = 0{,}54\frac{k_0}{T_0}$
$u_k = u_{k-1} + K_p(y_{k-1} - y_k) + K_i T(r_k - y_k) + \frac{K_d}{T}(2y_{k-1} - y_k - y_{k-2})$	$K_p = \frac{1{,}2L}{a(L+0{,}5T)} - 0{,}5K_iT$ $K_i = \frac{0{,}6L}{a(L+0{,}5T)^2}$ $K_d = \frac{0{,}5L}{a}$ ou $\frac{0{,}6L}{a}$ si $\frac{L}{T}$ entier	$K_p = 0{,}6k_0 - 0{,}5\ K_iT$ $K_i = 1{,}2\frac{k_0}{T_0}$ $K_d = \frac{3}{40}k_0T_0$

4. A. Besançon-Voda & S. Gentil, *Régulateurs PID analogiques et numériques*, Tech. de l'ingénieur R7416 (1999), ou sans les fautes, le cours de Gonzalo Cabodevila disponible à `http://jmfriedt.free.fr/Gonzalo_cours1A.pdf`

Rapport

Le rapport a pour vocation de vous permettre de reproduire les résultats sans passer par toutes les étapes de recherche :

- Prendre des notes/illustrations au cours du projet
- Contextualiser l'étude pour un lecteur qui n'est pas au courant du sujet (=introduction)
- **PAS de plagiat** : référencer les sources/illustrations
- bibliographie ($\neq$ série de sites web)
- toute référence (figure, bibliographie) doit être citée dans le texte lorsque utile
- sections de code discutées dans le texte mais pas de pages de listing

Ressources : manuscrit décrivant le projet
`http://jmfriedt.free.fr/network_analyzer.pdf`

Organisation : 2 semaines à temps plein, diverses séances de 4 h au cours du semestre + travail chez soi