Éléments de correction

Sujet zéro de l'épreuve
informatique et modélisation de systèmes physiques

Étude d'un capteur de modification de fissure : fissuromètre

Q1. Déterminer sa masse linéique.

Masse linéique : $\mu = \rho\pi r^2$. AN : $\mu = 0{,}1\,\mathrm{kg\,m^{-1}}$.

Q2.a) Qualifier la nature de cette onde.

Onde progressive harmonique.

b) Donner en justifiant le sens de propagation de cette onde.

Sens des x croissants (cf $y = 0$, $y' < 0$ a progressé de 0,06 m pendant T_s).

c) Déterminer la valeur de la tension T_0 avec deux chiffres significatifs.

$T_s = T/4$ avec T période de vibration. $T = 2 \times 10^{-2}\,\mathrm{s}$.
$\lambda = 0{,}24\,\mathrm{m}$ $\quad V = \lambda/T$ $\quad V = 12\,\mathrm{m\,s^{-1}}$ $\quad T_0 = \mu V^2$ $\quad T_0 = 14\,\mathrm{N}$

d) Donner la forme de l'équation de propagation que vérifie l'élongation $y(x,t)$, puis la forme de $y(x,t)$.

$$\frac{\partial^2 y}{\partial x^2} - \frac{1}{V^2}\frac{\partial^2 y}{\partial t^2} = 0$$
$$y(x,t) = y_0 \sin\left[\omega\left(t - \frac{x}{V}\right)\right] \qquad y_0 = 0{,}01\,\mathrm{m} \qquad \omega = \frac{2\pi}{T} \qquad \omega = 3 \times 10^2\,\mathrm{rad\,s^{-1}}$$

Q3.a) Donner les valeurs des conditions limites de $y(x,t)$ en $x = 0$ et $x = L$ quel que soit t.

$y(0,t) = 0$ et $y(L,t) = 0$ $\qquad \forall t$

b) Déterminer l'équation vérifiée par $f(x)$ puis la forme générale de la solution.

$$\frac{\mathrm{d}^2 f(x)}{\mathrm{d}x^2} - \frac{\omega^2}{V^2} f(x) = 0$$
$$f(x) = A\sin k_0 x + B\cos k_0 x \qquad k_0 = \frac{\omega}{V}$$

c) En déduire l'expression des modes propres.

CL : $f(0) = 0 \implies B = 0$ $\qquad f(L) = 0 \implies A\sin k_0 L = 0$

donc $k = \dfrac{n\pi}{L}$ soit $\boxed{\omega_n = \dfrac{n\pi V}{L}}$ $n \in \mathbb{N}$

d) Justifier que la forme générale de l'onde peut s'écrire sous la forme $y(x,t) = \sum_{n=1}^{\infty} y_n(x,t)$ avec $y_n(x,t) = (a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t)) \sin\left(\dfrac{n\pi x}{L}\right)$.

Superposition car équation linéaire.

e) Donner les fréquences présentes dans les vibrations.

$f_n = \dfrac{nV}{2L}$

f) Donner la fréquence de la vibration la plus intense. Déterminer numériquement sa valeur avec deux chiffres significatifs.

$n = 1$ pour la fondamentale. $f_0 = \dfrac{V}{2L}$ $\quad T_0 = k\Delta x$
$T_0 = 15\,\mathrm{N}$ $\quad V = 12\,\mathrm{m\,s^{-1}}$ $\quad f_0 = 12\,\mathrm{Hz}$

g) En réalité le signal comporte pour chaque harmonique une décroissance exponentielle. Expliquer son origine.

Dissipation par frottement fluide.

h) Déterminer la variation de la fréquence propre fondamentale due à cette variation.

ΔL_0 induit une variation de tension de $\Delta T_0 = k\Delta L_0$ soit $2\dfrac{\Delta V}{V} = \dfrac{\Delta T_0}{T_0}$ ($L = $ cste) or $\dfrac{\Delta f_0}{f_0} = \dfrac{\Delta V}{V} \Longrightarrow \Delta f_0 = 2k\Delta L_0 \dfrac{f_0}{T_0}$

i) On veut détecter une variation de 0,1 mm. Donner la valeur de la précision du détecteur de fréquence.

$\Delta L_0 = 10^{-4}\,\mathrm{m}$ $\quad \Delta f_0 = 0{,}16\,\mathrm{Hz}$ $\quad$ précision à $0{,}1\,\mathrm{Hz}$

Q4.a) Donner l'équation locale de Maxwell Ampère en présence de charges et de courants.

Cours

b) En déduire le théorème d'Ampère en régime variable ou théorème d'Ampère généralisé.

Cours

c) Donner l'énoncé ce théorème en régime quasi-stationnaire. (On ne demande pas de justifications).

Cours

Q5.a) Discuter la légitimité de cette approximation.

rayon $r = \sqrt{\dfrac{S}{\pi}}$ $\quad r = 5 \times 10^{-3}\,\mathrm{m}$. On n'a pas d'approximation limite ! $r \ll h$

b) Précisez les composantes et les dépendances du champ magnétique $\vec{B}$ créé par ce solénoïde en tout point de l'espace. Donner la valeur du champ à l'extérieur en justifiant votre réponse.

Cours

c) Déterminer le champ magnétique à l'intérieur du solénoïde.

Cours

d) Evaluer l'ordre de grandeur de l'inductance de cette bobine.

$L_0 = \dfrac{N^2 S \mu_0}{h}$ $\quad L_0 \sim 4\,\mathrm{mH}$

e) Donner explicitement un protocole expérimental permettant de mesurer cette inductance.

Circuit RLC série...

f) Dire si la valeur réelle est plus grande ou plus faible que la valeur obtenue dans le cadre du solénoïde infini.

$|B_{\text{réel}}| < |B_\infty|$ donc $L_{0,\text{réel}} < L_{0,\infty}$

Q6.a) Donner le moment magnétique $\vec{m}$ d'une spire circulaire de rayon a et d'axe $(C, \vec{y})$ parcourue par un courant électrique d'intensité i.

Cours

b) Donner l'ordre de grandeur du moment magnétique d'un aimant.

Cours

Q7.a) En considérant un point de son axe, déterminer la direction du champ magnétique qu'elle crée.

Cours

b) Justifier que les lignes de champ sont comprises dans ce plan. Justifier les symétries observées d'axe $(C, \vec{y})$ et d'axe $(C, \vec{x})$.

Cours

c) En déduire l'expression du champ créé par le moment sur l'axe en fonction du moment m et de Y.

$B(Y) = \dfrac{\mu_0 m}{2\pi} \dfrac{1}{Y^3}$ car $a \to 0$.

Induction

d) On se place en circuit ouvert. Déterminer l'expression de cette tension dans le cas où la vibration du fil est sinusoïdale d'amplitude y_0 et que la bobine est située à une distance d sous l'axe $(O, \vec{x})$ (on a $d = 2\,\text{cm}$ et $y_0 = 2\,\text{mm}$). Déterminer l'amplitude de cette tension en prenant pour m une valeur que l'on justifiera.

$\phi = NSB(Y) \qquad Y = d + \sin(\omega t)$

$B(Y) = \dfrac{\mu_0 m}{2\pi d^3}\left(1 - 3\dfrac{y_0}{d}\sin(\omega t)\right) \qquad (y_0 \ll d)$

$e = \dfrac{3\mu_0 m y_0}{2\pi d^4}\omega \sin(\omega t) \qquad$ amplitude $E = \dfrac{3\mu_0 m y_0 f}{d^4}$

$m = 5 \times 2r \times 10^6 = 10^{-2}\,\text{A m}^2$ donc $E = 4\,\text{V}$.

Q8.a) Donner la valeur de la pulsation ω_0 et une contrainte sur la pulsation de coupure afin d'obtenir un signal constant en sortie.

Sortie de (X) : $kuu_0\,(\sin\omega t \cos\omega_0 t \cos\varphi - \sin\omega t \sin\omega_0 t \sin\varphi)$

$< u_X > \neq 0 \Longrightarrow \omega = \omega_0$

$\omega_c \ll 2\omega$

b) En pratique le déphasage entre les deux signaux est aléatoire. Expliquer quel est le problème pratique ainsi posé.

On ne contrôle as U donc pas l'amplitude de la déformation.

Q9.a) Définir en quoi consiste l'échantillonnage et proposer un montage pratique.

Cours

b) Donner la condition à respecter pour T_e afin de pouvoir retrouver le signal analogique.

Cours

c) Sachant que l'amplitude du signal appartient à $\pm 5\,\text{V}$, déterminé la résolution du signal numérisé (c'est-à-dire la variation de tension minimale détectable).

10 bits $\rightarrow$ 1024 valeurs soit $\Delta V = \dfrac{10}{1024}$ $\Delta V = 0{,}01\,\text{V}$.

Q10. Déterminer la quantité de mémoire nécessaire au traitement numérique du signal et en déduire la taille de mémoire interne nécessaire du micro-contrôleur.

Il faut $4 \times 100 \times 2 = 800$ octets de mémoire pour traiter le signal. On en déduit qu'il faut prendre un micro-contrôleur avec au moins 1024 octets de mémoire interne.

Q11. Ecrire une fonction `trans_fourier_freq(U,k)` retournant la partie réelle et la partie imaginaire de la transformée de Fourier pour la fréquence `k`.

Solution Python

```
def trans_fourier_freq(U,k):
    N = len(U)
    real = 0
    imag = 0
    for n in range(N):
        real += U[n]*cos(-2*pi*k*n/N)
        imag += U[n]*sin(-2*pi*k*n/N)
    return real,imag
```

Solution Scilab

```
function [real1,imag1]=trans_fourier_freq(U,k)
    N = length(U)
    real1 = 0
    imag1 = 0
    for n = [1:N]
        real1 = real1 + U(n)*cos(-2*%pi*k*(n-1)/N)
        imag1 = imag1 + U(n)*sin(-2*%pi*k*(n-1)/N)
    end
endfunction
```

Q12. Ecrire une fonction `module(a,b)` retournant le module du complexe $a + jb$.

Solution Python

```
def module(a,b):
    return sqrt(a**2 + b**2)
```

Solution Scilab

```
function y=module(a,b)
    y = sqrt(a**2 + b**2)
endfunction
```

Q13. Ecrire une fonction `trans_fourier(U)` retournant le module de la transformée de Fourier discrète sous la forme d'une liste de taille N contenant les modules des termes $TUf(k)$ en utilisant notamment des appels aux fonctions précédentes `trans_fourier_freq(U,k)` et `module(a,b)`.

Solution Python

```
def trans_fourier(U):
    TFU = zeros(len(U))
    for k in range(len(U)):
        ak,bk = trans_fourier_freq(U,k)
        TFU[k] = module(ak,bk)

    return TFU
```

Solution Scilab

```
function TFU = trans_fourier(U)
    TFU = zeros(length(U))
    for k = [0:length(U)-1]
        [ak,bk] = trans_fourier_freq(U,k)
        TFU(k+1) = module(ak,bk)
    end
endfunction
```

Q14. Déterminer la complexité du calcul de la transformée de Fourier par la fonction `trans_fourier(U)` en fonction de N et commenter en quelques phrases de l'efficacité de l'algorithme utilisé.

On a deux boucles de taille N imbriquées ce qui donne une complexité en ΘN^2. Cet algorithme n'est pas du tout optimal.

Q15. Déterminer la liste des fréquences pour :
- $N = 10$ et $Te = 1/10$,
- $N = 11$ et $Te = 1/11$.

Pour $N = 10$, $[0, 1, 2, 3, 4, -5, -4, -3, -2, -1]$
Pour $N = 11$, $[0, 1, 2, 3, 4, 5, -5, -4, -3, -2, -1]$

Q16. Ecrire une fonction `freq_fourier(U,Te)` qui renvoie la liste des fréquences `freq`.

```
def freq_fourier(U,Te):
    w = zeros(len(U))
    if (len(U)%2 == 1):
        for i in range(0,len(U)//2+1):
            w[i] = i / Te / len(U)
        for i in range(len(U)//2+1,len(U)):
            w[i] = (i-len(U)) / Te / len(U)
    else:
        for i in range(0,len(U)//2):
            w[i] = i / Te / len(U)
        for i in range(len(U)//2,len(U)):
            w[i] = (i-len(U)) / Te / len(U)
    return w
```

Solution Scilab

```
function w = freq_fourier(U,Te)
    w = zeros(length(U))
    if (modulo(length(U),2) == 1)
```

```
        for i = [1:int(length(U)/2)+1]
            w(i) = (i-1) / Te / length(U)
            end
        for i = [int(length(U)/2)+2:length(U)]
            w(i) = (i - 1 - length(U)) / Te / length(U)
        end
    else
        for i = [1:int(length(U)/2)]
            w(i) = (i-1) / Te / length(U)
            end
        for i = [int(length(U)/2)+1:length(U)]
            w(i) = (i - 1 - length(U)) / Te / length(U)
        end
     end
endfunction
```

Q17. Ecrire une fonction `freq_corde(U)` retournant la fréquence où le module de la transformée de Fourier est maximal; cette fréquence correspond à la fréquence de vibration de la corde. On supposera que le maximum est unique. On utilisera notamment les fonctions `freq_fourier(U,Te)` et `trans_fourier(U)` dans la fonction à écrire.

Solution Python

```
def freq_corde(U,Te):
    TF = trans_fourier(U)
    freq = freq_fourier(U,Te)
    max = 0
    imax = 0
    for i in range(len(U)):
        if TF[i] > max and freq[i] > 0:
            max = TF[i]
            imax = i
    return freq[imax]
```

Solution Scilab

```
function fmax = freq_corde(U,Te)
    TF = trans_fourier(U)
    freq = freq_fourier(U,Te)
    disp(TF)
    disp(freq)
    max1 = 0
    imax = 0
    for i = [1:length(U)]
        if TF(i) > max1 & freq(i) > 0
            max1 = TF(i)
            imax = i
        end
    end
    fmax = freq(imax)
endfunction
```

Q18. Expliciter la différence entre des variables globales et des variables locales.

Variable locale : existence uniquement dans la table de variables d'une fonction; création à l'appel et la fonction et destruction de l'ensemble à la fin de la création
Variable globale : appel dans une fonction à des variables définies dans la table global en dehors de la fonction.

Q19. Ecrire une fonction `moyenne_optimisee(var1, var2,...)`, dont les arguments sont à définir, qui détermine la nouvelle moyenne à partir de la nouvelle mesure et de la moyenne déterminée pour l'ensemble des mesures précédentes. Expliciter en terme de nombre d'opérations pourquoi votre fonction est optimale par rapport à un calcul classique de la moyenne.

Solution Python

```python
def moyenne_optimisee(U,M):
    return ((M * (len(U)-1) + U[-1])/len(U))
```

Solution Scilab

```
function moy = moyenne_optimisee(U,M)
    moy = ((M * (length(U)-1) + U($))/length(U))
endfunction
```

Ce calcul de moyenne nécessite 3 opérations quelque soit la taille de la liste (1 addition, 1 multiplication et 1 division).
Un calcul de moyenne classique nécessite N additions puis une division.

Q20. Ecrire une fonction `ecart_type_optimise(var1, var2, ...)`, dont les arguments sont à définir, qui sera optimisé : c'est-à-dire qu'elle utilisera la valeur de l'écart type déterminé pour l'ensemble des mesures précédentes (ou des variables intermédiaires permettant ce calcul que vous pourriez sauvegarder) pour déterminer le nouvel écart type. Expliciter en terme de nombre d'opérations pourquoi votre fonction est optimale par rapport à un calcul classique de l'écart type.

Solution Python

```python
def ecart_type_optimise(U,SUC,M):
    SUCn = SUC + U[-1]**2          #SUCn : somme U au carré
    if len(U) != 0:
        return sqrt(SUCn/len(U) - M**2),SUCn
    else:
        return 0
```

Solution Scilab

```
function [ec,SUCn] = ecart_type_optimise(U,SUC,M)
    SUCn = SUC + U($)**2          //SUCn : somme U au carré
    if length(U) <> 0
        ec = sqrt(SUCn/length(U) - M**2)
    else
        ec = 0
  end
endfunction
```

Cette solution ne nécessite que de 6 opérations (1 addition, 1 soustration, 2 passage au carré et une division et un calcul de racine).

La solution standard nécéssite $2n_1+1$ passage au carré, 1 division, 1 soustraction et 1 calcul de racine.

Q21. Ecrire une fonction `analyse()` qui permet de satisfaire à la description du début du paragraphe **??**. Vous pourrez faire appel aux fonctions :

- `ecart_type_optimise(var1, var2, ...)`,
- `moyenne_optimisee(var1, var2,...)`,
- `filtre_signal(Ue, Te, wc)` qui renvoie le signal filtré $u_f(t)$,

– la fonction `acquisition()` qui renvoie le signal brut $u_e(t)$.

Solution Python

```
def analyse():
    global mode_analyse, alarme
    liste_freq = []
    EC = 0
    Moy = 0
    SUC = 0
    while mode_analyse == True:
        Ue = acquisition()
        Uf = filtre_signal(Ue)
        liste_freq.append(freq_corde(Uf, Te))
        Moy = moyenne_optimisee(liste_freq , Moy)
        EC,SUC = ecart_type_optimise(liste_freq ,SUC,Moy)

        if liste_freq[len(liste_freq)-1] < Moy-5*EC or liste_freq[len(liste_freq
            )-1] > Moy+5*EC:
            mode_analyse= False
            alarme = True
        sleep(10)
```

Solution Scilab

```
function analyse()
    global mode_analyse, alarme
    liste_freq = []
    EC = 0
    Moy = 0
    SUC = 0
    while mode_analyse == True
        Ue = acquisition()
        Uf = filtre_signal(Ue,Te,2*%pi*20)
        liste_freq($+1)=freq_corde(Uf, Te)
        Moy = moyenne_optimisee(liste_freq , Moy)
        [EC,SUC] = ecart_type_optimise(liste_freq ,SUC,Moy)

        if liste_freq(length(liste_freq)) < Moy-5*EC | liste_freq(length(
            liste_freq)) > Moy+5*EC
            mode_analyse= False
            alarme = True
        end
        sleep(10000)
    end
endfunction
```

Q22. Observer le résultat de la proposition du candidat et justifier que cette solution ne répond pas aux attentes.

La liste n'est clairement pas aléatoire : en effet, c'est la même valeur du random qui est appliquée à chaque élément de la liste !
De plus, l'élément aléatoire ajouté est compris en 0 et a.

Q23. Proposer une fonction `y(t,a)` qui réponde à la définition souhaitée.

Solution Python

```
def y(t,a):
    yt = zeros(len(t))
```

```
    for i in range(len(t)):
        yt[i] = (2*rd.random() - 1)*a + t[i]
    return yt
```

Solution Scilab

```
function yt = y(t,a)
    yt = zeros(length(t))
    for i = [1:length(t)]
        yt(i) = (2*rand() - 1)*a + t(i)
    end
endfunction
```

L'implantation numérique du filtre passe par la détermination de la relation de récurrence entre la sortie à l'instant nTe avec la valeur de la sortie aux instants précédents ainsi que la valeur de l'entrée.
On choisit un filtre de Butterworth d'ordre 2 dont l'équation différentielle entre l'entrée u_e et la sortie u_f se met sous la forme :

$$\frac{1}{\omega_c^2}\ddot{u}_f(t) + \frac{\sqrt{2}}{\omega_c}\dot{u}_f(t) + u_f(t) = u_e(t)$$

Le signal d'entrée $u_e(t)$ est stocké dans la liste `Ue`, le signal filtré est stocké dans la liste `Uf`. Le pas de temps T_e est stocké dans la variable `Te` et la pulsation de coupure ω_c dans la variable `wc`.

Q24. Exprimer `Ufp_n`, la dérivée première de $u_f(t)$ à l'instant nTe, en fonction des valeurs de la liste `Uf` et de `Te` en appliquant la méthode d'Euler explicite.

$$Ufpn = \frac{Uf[n] - Uf[n-1]}{Te}$$

Q25. Exprimer `Ufpp_n`, la dérivée seconde de $u_f(t)$ à l'instant nTe, en fonction des valeurs de la liste `Uf` et de `Te` en appliquant la méthode d'Euler explicite.

$$Ufppn = \frac{Uf[n] - 2Uf[n-1] + Uf[n-2]}{Te^2}$$

Q26. En déduire la relation de récurrence du filtre en déterminant l'expression de `Uf[n]` en fonction de `Uf[n-1]`, `Uf[n-2]`, `Ue[n]`, `Te` et `wc`.

$$Uf[n] = \left(Ue[n] + \left(\frac{\sqrt{2}}{h\omega_c} + \frac{1}{h^2\omega_c^2}\right)Uf[n-1] - \frac{1}{h^2\omega_c^2}Uf[n-2]\right) \times \frac{1}{\frac{1}{h^2\omega_c^2} + \frac{\sqrt{2}}{h\omega_c} + 1}$$

Q27. Ecrire une fonction `filtre_signal(Ue, Te, wc)` renvoyant la liste `Uf` représentant le signal filtré. On supposera les conditions initiales nulles.

Solution Python

```
def filtre_signal(U,h,w):
    Us = zeros(len(U))
    for i in range(2,len(U)):
        Us[i] = (U[i] + (sqrt(2)/(h*w)+2/(h**2*w**2))*Us[i-1] - 1/(h**2*w**2)*Us
            [i-2]) / (1/(h**2*w**2)+sqrt(2)/(h*w)+1)
    return Us
```

Solution Scilab

```
function Us =  filtre_signal(U,h,w)
    Us = zeros(len(U))
    for i = [3:length(U)]
        Us(i) = (U(i) + (sqrt(2)/(h*w)+2/(h**2*w**2))*Us(i-1) - 1/(h**2*w**2)*Us
            (i-2)) / (1/(h**2*w**2)+sqrt(2)/(h*w)+1)
    end
endfunction
```

Q28. Donner la requête SQL à envoyer à la base de données permettant de récupérer la valeur des champs `time` et `depl` entre les deux instants `time1` et `time2`.

```
requete = SELECT time,depl FROM evol_fissure WHERE time > time1 AND time <
time2
```

Q29. Donner les commandes qui ont été nécessaires pour réaliser l'affichage de la FIGURE **??**, sachant que seule une valeur sur 10 est tracée.

Solution Python

```
import matplotlib.pyplot as plt
plt.plot(resultat_requete[0:-1:10,0], resultat_requete[0:-1:10,1],'r')
plt.xlabel('Durée (en jours)')
plt.ylabel('Déplacement de la fissure (en m)')
plt.show()
```

Solution Scilab

```
plot(resultat_requete(1:10:$,1), resultat_requete(1:10:$,2),'r')
xlabel('Durée (en jours)')
ylabel('Déplacement de la fissure (en m)')
```