

Devoir d'informatique numéro 2

Ne traiter que la partie III, donc les questions 18 à 35.

Étude d'un tensiomètre électronique

I Présentation

Dans le domaine médical, la pression artérielle est l'un des paramètres les plus surveillés dans le cadre des maladies cardio-vasculaires. En effet, quand la pression artérielle est trop élevée, pendant des durées très longues, le muscle cardiaque finit par se fatiguer jusqu'à atteindre des insuffisances cardiaques sévères ; on parle d'hypertension.

À l'inverse, une tension trop basse conduit à sous-alimenter les organes et peut conduire à des étourdissements ou évanouissements. L'hypotension n'est généralement pas considérée comme une maladie.

Communément, la mesure de la pression artérielle se fait par la détermination de deux valeurs classiques : la pression systolique (pression la plus élevée suite à la contraction du coeur) et la pression diastolique (pression la moins élevée après le relâchement du coeur quand celui-ci se remplit de sang).

Pour mesurer ces pressions, le médecin utilise classiquement la méthode du brassard associé à son stéthoscope. Le médecin gonfle le brassard jusqu'à couper la circulation sanguine dans le bras. Le médecin pose ensuite le stéthoscope sur l'artère et dégonfle progressivement le brassard. Dès qu'il écoute le sang passer à nouveau dans l'artère, il mesure la pression systolique sur le manomètre associé au brassard. Puis il continue à dégonfler le brassard jusqu'à ne plus écouter aucun bruit, là il lit la pression diastolique. On peut également mesurer ces pressions avec un tensiomètre électronique (voir figure 1). Il existe des modèles simples pour les particuliers et des modèles beaucoup plus complexes dans le cadre des milieux médicaux.



FIGURE 1 – Tensiomètre utilisé en milieu hospitalier.

Les applications numériques seront données avec 1 chiffre significatif, sauf contre-ordre.

II Étude du capteur de pression

II.1 Étude du capteur : modélisation de la déformation de la membrane

Le principe du capteur de pression repose sur l'effet piézorésistif (voir figure 2). La cellule sensible du capteur est constituée d'une membrane se déformant sous l'effet de la force appliquée par un micro-vérin. Le micro-vérin applique une force F proportionnelle à la pression du brassard. Sur la membrane sont fixées 4 jauges de déformation. Les résistances électriques de ces jauges varient sous l'effet d'une déformation. Puis cette variation de résistance est transformée en tension électrique V_p au moyen d'un circuit de conditionnement. L'objectif de cette partie est de relier la pression du brassard à la tension mesurée dans le circuit de conditionnement en fonction des différents paramètres caractéristiques du capteur.

On cherchera également à montrer que le capteur permet d'avoir une précision de moins d'un mmHg.

En première approximation, la membrane peut être modélisée par une poutre de longueur $2L$ encastree à chacune de ses extrémités soumise à un effort d'intensité F en son milieu (figure 3(a)).

Les caractéristiques de la poutre sont :

- E : le module d'Young
- S : la surface de la section droite de la poutre

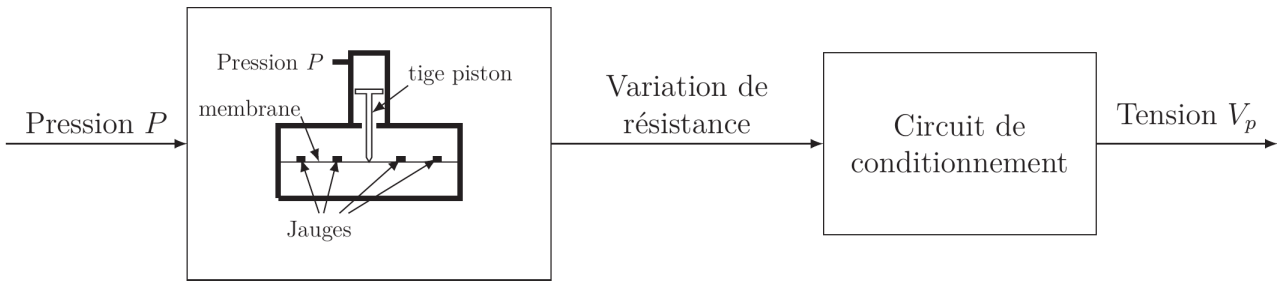


FIGURE 2 – Principe de fonctionnement du capteur et de la mesure.

- I : le moment d'inertie de la section autour de l'axe (G, \vec{z})

- e : l'épaisseur de la membrane

G est le centre d'inertie de la section droite de la poutre.

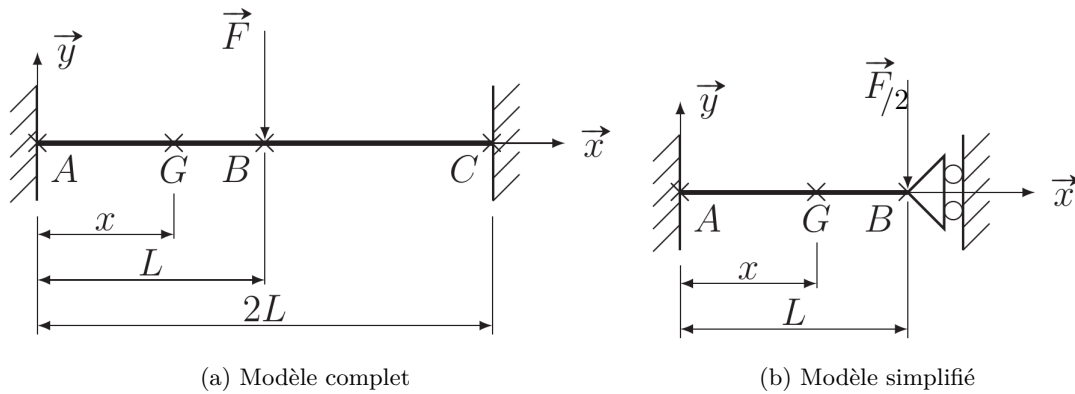
On note $\vec{AG} = x\vec{x}$.

On suppose que le déplacement de la section droite est $\vec{u}(G) = u_y(x)\vec{y}$.

La force appliquée par le vérin est notée $\vec{F} = -F\vec{y}$.

Les effets de la pesanteur sont négligés.

Le modèle est considéré comme plan.



(a) Modèle complet

(b) Modèle simplifié

FIGURE 3 – Modèle poutre de la membrane

Pour simplifier le calcul de la déformée de la membrane, on adopte le modèle donné sur la figure 3(b). On supposera que la liaison en B bloque la translation suivant \vec{x} et la rotation suivant (B, \vec{z}) .

Q1. Justifier pourquoi le modèle de la figure 3(b) est équivalent à celui de la figure 3(a). Justifier l'intérêt de cette simplification en déterminant le degré d'hyperstatisme des deux modèles.

On notera les torseurs des actions mécaniques au niveau des liaisons sous la forme :

$$\{\mathcal{T}(i \rightarrow j)\} = \left\{ \begin{array}{cc} X_M & - \\ Y_M & - \\ - & M_M \end{array} \right\}_{M,R}, \text{ le torseur de cohésion est noté } \{\mathcal{T}(coh)\} = \left\{ \begin{array}{cc} T_X & - \\ T_Y & - \\ - & M_Z \end{array} \right\}_{G,R}$$

(action de la partie droite sur la partie gauche).

Q2. Isoler le tronçon de poutre compris entre l'abscisse $[0, x]$. Écrire l'équation issue du principe fondamental de la statique permettant d'obtenir la relation entre le moment fléchissant M_Z , l'effort tranchant T_Y et une (ou des) inconnue(s) de la liaison en A .

Effectuer un second isolement et écrire l'équation nécessaire permettant d'exprimer l'effort tranchant T_Y en fonction de F .

En déduire la relation entre le moment fléchissant M_Z , l'effort F et une (ou des) inconnue(s) de la liaison en A .

On rappelle que la relation de comportement reliant le moment de flexion au déplacement u_y est $M_z = EI \frac{d^2 u_y}{dx^2}$. On se place dans les hypothèses de Navier-Bernoulli en petits déplacements, on a ainsi $\theta = \frac{du_y}{dx}$, θ la rotation de la section droite autour de l'axe (G, \vec{z}) .

Q3. Après intégration, montrer que le déplacement se met sous la forme $u_y(x) = A_1 x^3 + A_2 x^2$ où vous préciserez les expressions des constantes A_1 et A_2 en fonction des paramètres du modèle en éliminant les inconnues de la liaison en A .

On rappelle que le problème étant hyperstatique, le moment de flexion est exprimé en fonction d'inconnues de liaisons qui seront déterminées en appliquant les conditions limites.

On rappelle que la déformation dans une section droite de la poutre est donnée par $\varepsilon(x) = -y \frac{d\theta}{dx} = -y \frac{d^2 u_y}{dx^2}$

Q4. Montrer que la déformation sur la peau supérieure de la membrane ($y = e/2$) se met sous forme $\varepsilon = A_3 + A_4 x$, où vous préciserez les expressions des constantes A_3 et A_4 en fonction des paramètres du modèle.

Quatre jauges de déformation sont disposées sur la peau extérieure de la membrane (voir figure 4).

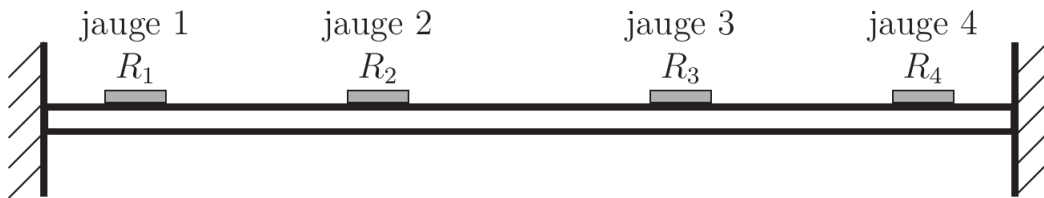


FIGURE 4 – Disposition des jauges de déformation sur la membrane.

Q5. Quelle serait la position idéale des jauges de déformation ? En pratique elles sont placées en $x = L/4$ et en $x = 3L/4$, donner la valeur des déformations de ces jauges et montrer que $\varepsilon_1 = -\varepsilon_2 = -\varepsilon_3 = \varepsilon_4 = kF$ où vous donnerez l'expression de k en fonction des paramètres du modèle.

II.2 Choix des jauges de déformations

Les jauges de déformation disposées sur la membrane (voir figure 4) subissent une déformation proportionnelle à l'intensité de la force F appliquée par le piston : $\varepsilon_1 = -\varepsilon_2 = -\varepsilon_3 = \varepsilon_4 = kF$.

On cherche maintenant à déterminer la relation entre la variation de la résistance de la jauge de déformation et l'effort presseur F .

Considérons une jauge de déformation de forme parallélépipédique constituée d'un matériau homogène de résistivité ρ (s'exprimant en $\Omega.m$). On note ℓ sa longueur, a et b ses dimensions transversales, s sa section droite (voir figure 5).

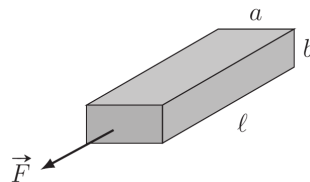


FIGURE 5 – Jauge de déformation.

La résistance électrique R de ce corps d'épreuve s'exprime au moyen de la relation $R = \rho \frac{\ell}{s}$

Q6. Lorsque la jauge d'épreuve est soumise à une force \vec{F} appliquée parallèlement à ℓ , elle se déforme : ℓ , s et ρ subissent de petites variations valant respectivement $\delta\ell$, δs et $\delta\rho$. Exprimer la variation relative de résistance électrique $\frac{\delta R}{R}$.

Les variations de ℓ et de a et b ne sont pas indépendantes : $\frac{\delta a}{a} = \frac{\delta b}{b} = -\nu \frac{\delta \ell}{\ell}$ avec ν une constante dépendant du matériau, appelée coefficient de Poisson.

Q7. Exprimer $\frac{\delta R}{R}$ en fonction de $\frac{\delta \ell}{\ell}$, ν et $\frac{\delta \rho}{\rho}$.

Dans le cas d'un métal, $\delta \rho$ est lié à la variation δV du volume V selon la loi $\frac{\delta \rho}{\rho} = c \cdot \frac{\delta V}{V}$, avec c une constante appelée constante de Bridgman.

Q8. Montrer que $\frac{\delta \rho}{\rho}$ peut s'écrire sous la forme :

$$\frac{\delta \rho}{\rho} = K \frac{\delta \ell}{\ell} \text{ avec } K \text{ une constante appelée facteur de jauge.}$$

Dans le cas d'un semi-conducteur, $\delta \rho$ est lié à la contrainte $\sigma = \frac{\|\vec{F}\|}{s}$ selon la loi $\frac{\delta \rho}{\rho} = \Pi \cdot \sigma$, avec Π une constante appelée constante de piézorésistivité ; de plus, le module de Young E du matériau lie la contrainte σ et la déformation $\frac{\delta \ell}{\ell} : \frac{\delta \ell}{\ell} = E \sigma$

Q9. Exprimer le facteur de jauge dans le cas d'un semi-conducteur.

On donne :

- Pour le cuivre : $\nu = 0,3$, $c = 1$;
- Pour le silicium : $\nu = 0,4$, $E = 1011 \text{ Pa}$, $\Pi = 10^{-9} \text{ Pa}^{-1}$.

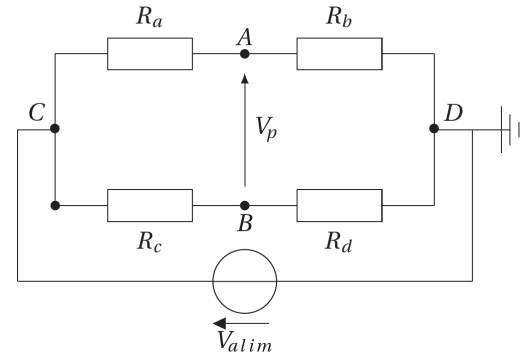
Q10. Évaluer le facteur de jauge dans le cas du cuivre et dans le cas du silicium. Commenter.

Q11. Quel phénomène peut induire une erreur sur la mesure de la contrainte ?

II.3 Circuit de conditionnement : pont de Wheatstone

Les 4 jauges semi-conductrices identiques, de résistance R_0 à l'équilibre, sont fixées sur la membrane (voir figure 4). La déformation $\frac{\delta \ell}{\ell}$ de la jauge placée en x vaut donc $\varepsilon(x)$.

La déformation des jauges conduit à une variation ΔR_i ($i = a, b, c$ ou d) de leur résistance. Les 4 jauges sont branchées en pont de Wheatstone, alimenté sous une tension $V_{alim} = 16 \text{ V}$. Les bornes A et B sont branchées à un amplificateur de résistance d'entrée très grande.



Q12. Déterminer l'expression de la tension V_p en fonction des paramètres du pont de Wheatstone. À l'équilibre le pont est réglé de manière à avoir $V_p = 0$.

Q13. Déterminer la variation de tension $\Delta V_p = V_p$ lorsque les résistances subissent des variations ΔR_i en fonction des variations $\Delta R_i, V_{alim}, R_0$.

Q14. La jauge 1 correspond à la résistance R_b . Déterminer la position qu'il convient de donner aux autres jauges dans le pont de Wheatstone (autrement dit faire le lien entre les jauges 2, 3 et 4 avec les résistances R_a, R_c , et R_d du pont).

Q15. Exprimer la tension V_p en fonction de V_{alim}, ε_1 et du facteur de jauge K .

On considère un capteur de pression constitué d'une membrane en silicium ayant les caractéristiques suivantes :

- surface $S_m = 4 \text{ mm}^2$;
- longueur $2L = 4 \text{ mm}$;
- épaisseur $e = 0,1 \text{ mm}$;
- module d'Young $E = 10^{11} \text{ Pa}$;
- moment d'inertie de la section $I = 2 \cdot 10^{-4} \text{ mm}^4$.

Q16. Etablir la relation entre la tension V_p et la pression $P = \frac{F}{S_m}$, puis définir et exprimer la sensibilité de ce capteur de pression en fonction de K, e, L, E, I, S_m et V_{alim} .

Q17. Quelle est la tension V_p obtenue pour une pression de 50 kPa ?

III Traitement numérique des informations

Le capteur fournit une tension $u(t)$ qui va subir un traitement informatique par une carte de commande. L'objectif de cette partie est de mettre en place les différents algorithmes permettant de déterminer les pressions diastolique et systolique ainsi que le pouls du patient après avoir filtré numériquement le signal. L'utilisation en milieu hospitalier nécessite la sauvegarde des différentes données dans une base de données afin de réaliser un meilleur suivi du patient.

Les algorithmes demandés au candidat peuvent être réalisés **dans le langage Python ou dans le langage Scilab ; toutes les questions seront traitées dans le même langage.**

Veillez à apporter les commentaires suffisants à la compréhension de votre programme et à utiliser des noms de variables explicites.

III.1 Numérisation du signal

Le signal $V_p(t)$ issu du conditionnement est amplifié pour donner la tension $u(t)$ fournie par le capteur ; cette tension $u(t)$ est numérisé par un convertisseur analogique-numérique.

La tension $u(t)$ mesurée appartient à $[0, 5 \text{ V}]$ pour une pression comprise entre $[0, 1350 \text{ HPa}]$.

La pression affichée est en mmHg, on rappelle que $1013 \text{ HPa} = 750 \text{ mmHg}$.

La résolution de l'affichage de la pression est de 1 mmHg , cependant la numérisation du signal nécessite une précision plus importante de l'ordre de $0,02 \text{ mmHg}$ maximum.

Le constructeur a, à sa disposition, des convertisseurs analogique-numérique (CAN) dotés d'une résolution de 10 bits, 12 bits et 16 bits.

Q18. Déterminer en justifiant votre réponse la résolution du (ou des) CAN permettant de respecter le critère de précision.

On cherche à déterminer la quantité de mémoire nécessaire pour stocker une mesure.

Une mesure peut durer jusqu'à 60 s. La fréquence d'échantillonnage est de 1000 Hz.

L'ensemble des valeurs de la mesure va être stocké dans un tableau.

Q19. Choisir un type de stockage pour les valeurs parmi les types suivants : entier relatif codé sur 8, 16, 32 ou 64 bits, entier non signé codé sur 8, 16, 32 ou 64 bits, réel codé sur 32 ou 64 bits. En déduire la quantité de mémoire nécessaire pour stocker le tableau, exprimée en octets (1 octet = 8 bits).

III.2 Filtrage numérique du signal

La fréquence d'échantillonnage est notée f_e , la période d'échantillonnage est notée T_e . Le signal numérisé est stocké dans une liste de taille N , notée \mathbf{Ue} , contenant l'ensemble des valeurs $U_e(nT_e)$ accessible par la commande $\mathbf{Ue}[n]$ avec $n \in \llbracket 0, N-1 \rrbracket$ en Python et par la commande $\mathbf{Ue}(n)$ avec $n \in \llbracket 1, N \rrbracket$ en Scilab.

Pour simplifier les notations, toute fonction du temps $U_g(t)$ évaluée au temps iT_e sera notée $U_{g,i}$ au lieu de $U_g(iT_e)$.

Le signal numérisé est filtré pour supprimer les bruits de mesure et permet d'obtenir le signal filtré numérique U_f sous forme d'une nouvelle liste \mathbf{Uf} .

Le filtre numérique retenu est un filtre du 2nd ordre modélisé par l'équation différentielle

$$\frac{1}{\omega^2} \ddot{U}_f(t) + \frac{2z}{\omega} \dot{U}_f(t) + U_f(t) = U_e(t)$$

On choisit $\omega = 20 \text{ rad/s}$ et $z = 0,7$.

Pour résoudre cette équation différentielle, on utilise le schéma numérique d'intégration de Newmark :

$$\begin{aligned} \dot{U}_{f,i+1} &= \dot{U}_{f,i} + (1 - \gamma)T_e \ddot{U}_{f,i} + \gamma T_e \ddot{U}_{f,i+1} \\ U_{f,i+1} &= U_{f,i} + T_e \dot{U}_{f,i} + T_e^2 (1/2 - \beta) \ddot{U}_{f,i} + T_e^2 \beta \ddot{U}_{f,i+1} \end{aligned}$$

où γ et β sont des paramètres de la méthode.

Q20. En introduisant les relations du schéma de Newmark dans l'équation différentielle exprimée au pas de temps $(i+1)T_e$, montrer que la relation de récurrence peut se mettre sous la forme suivante : $\ddot{U}_{f,i+1} = B_1 (U_{e,i+1} + B_2 \ddot{U}_{f,i} + B_3 \dot{U}_{f,i} + B_4 U_{f,i})$ avec $B_1 = \frac{1}{1/\omega^2 + \frac{2z\gamma}{\omega} T_e + T_e^2 \beta}$.

Vous exprimerez B_2 , B_3 et B_4 en fonction de z, ω, β, γ .

Afin d'analyser les propriétés de la méthode de Newmark, on résout l'équation différentielle pour un second membre simple avec différents pas de temps, pour un réglage des paramètres γ et β fixé. Les résultats sont synthétisés sur la figure 7.

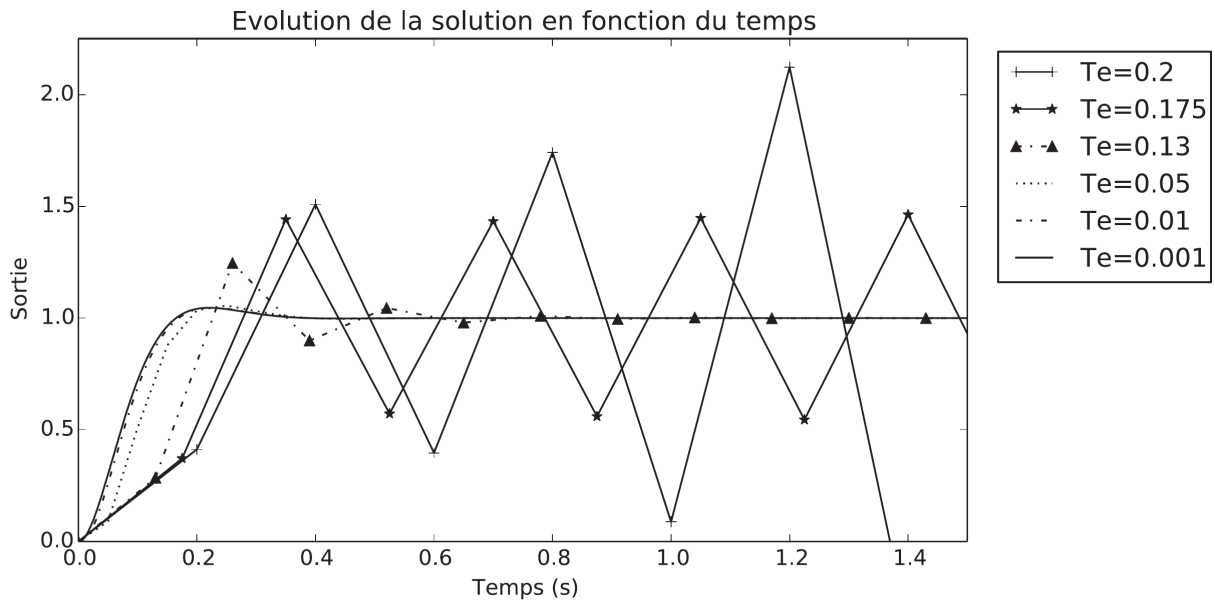


FIGURE 7 – Évolution de la solution en fonction de la valeur du pas de temps T_e .

Q21. Qualifier la stabilité du schéma d'intégration en fonction du pas de temps.

On définit l'erreur du schéma d'intégration par $\eta = \max_{i \in \llbracket 0, N \rrbracket} |U_{f,i} - U_{f,exacte,i}|$. Le tableau suivant donne l'évolution de l'erreur en fonction du pas de temps.

T_e	0,01	0,001	0,0001
η	0,05	0,005	0,0005

Q22. Étant donné l'évolution de l'erreur, proposer un ordre de convergence associé à la méthode de Newmark dans la configuration donnée.

Dans le cadre de l'acquisition numérique, la période d'échantillonnage est fixée, donc le pas de temps d'intégration aussi. On rappelle que la fréquence d'échantillonnage est de 1000 Hz.

Q23. Conclure quant à la qualité du calcul du signal filtré.

Q24. Écrire une fonction `newmark(gamma, beta, omega, z, e, Te)` qui prend en argument les paramètres γ et β de la méthode, les paramètres ω et z du filtre, le signal à filtrer e et la période d'échantillonnage T_e . Cette fonction retourne le signal filtré sous la forme d'une liste de même taille que la variable e . On supposera les conditions initiales nulles. On pourra utiliser directement les constantes B_1, B_2, B_3 et B_4 en les supposant définies comme variables globales.

Q25. Proposer une amélioration de la fonction `newmark` afin de minimiser la taille des variables intermédiaires, donner la nouvelle fonction. Si vous pensez avoir déjà un algorithme optimal, précisez-le sur votre copie en justifiant.

III.3 Détermination des pressions systolique, diastolique

Les courbes de la pression mesurée brute et filtrée sont données sur la figure 8. L'objectif de cette partie est d'analyser la courbe afin d'extraire les données recherchées. On dispose de deux tableaux : `data` contenant la pression mesurée filtrée et `t` contenant le temps associé à chaque mesure (les données sont rangées par temps croissant).

Pour déterminer la pression systolique, il faut parcourir la courbe ; celle-ci présente des successions de bosses et de creux qui seront respectivement nommés maximum local et minimum local. La pression systolique correspond à la valeur du maximum local où le critère

$\frac{\text{maximum_local} - \text{minimum_local}}{\text{minimum_local}} > 4 \times 10^{-5}$ est vérifié pour la première fois en parcourant les données par temps croissant.

Q26. Écrire une fonction `pression_systolique(data)` qui prend en argument les données mesurées filtrées et qui retourne la pression systolique. On supposera qu'il est toujours possible de trouver la pression systolique dans le tableau `data`.

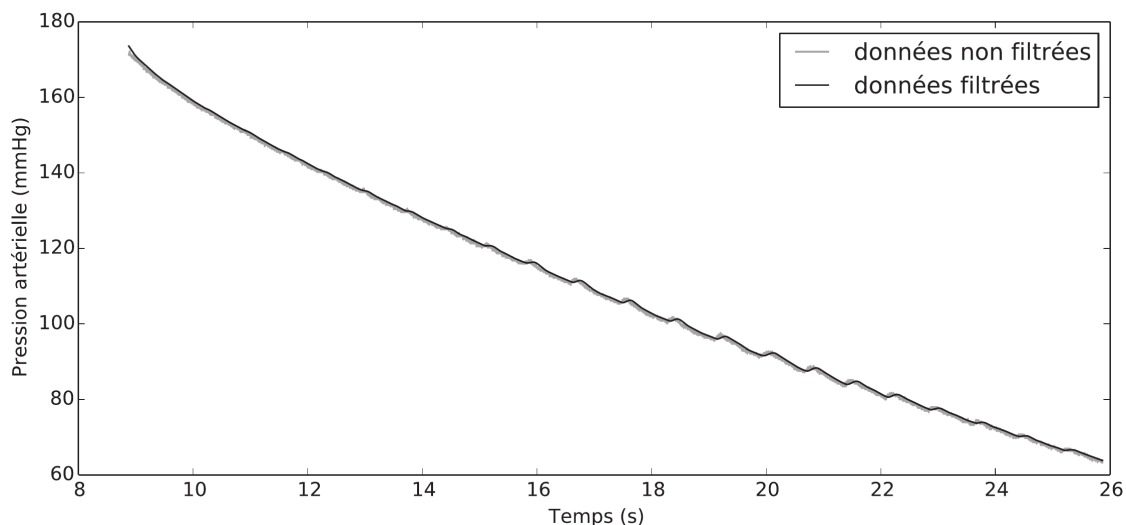


FIGURE 8 – Pression artérielle mesurée brute et filtrée.

La pression diastolique correspond à la valeur du maximum local où le critère $\frac{\text{maximum_local} - \text{minimum_local}}{\text{minimum_local}} > 4 \times 10^{-3}$ est vérifié pour la dernière fois en parcourant les données par temps croissant et que le maximum local reste supérieur à 40 mmHg pour éviter de détecter un bruit numérique.

Q27. Écrire une fonction `pression_diastolique(data)` qui prend en argument les données mesurées filtrées et qui retourne la pression diastolique. On supposera qu'il est toujours possible de trouver la pression diastolique à l'aide du tableau `data`.

IV Stockage et analyse des résultats

Le tensiomètre est utilisé dans le domaine hospitalier lors des visites régulières des infirmières dans les chambres des patients.

Le tensiomètre doit permettre d'enregistrer les différentes constantes pour chaque patient. Après utilisation, le tensiomètre est connecté au réseau de l'hôpital et synchronise ses mesures avec la base de données des patients.

La base de données est composée de plusieurs tables dont :

- patients avec les attributs :

- `id` : identifiant associé à un patient, clé primaire
- `numero_secu` : le numéro de sécurité sociale du patient
- `nom` : le nom du patient
- `prenom` : le prénom (ou les prénoms) du patient
- `adresse1` : adresse du patient
- `adresse2` : complément d'adresse du patient
- `code_postal` : code postal de la ville du patient
- `ville` : ville du patient
- `date_de_naissance` : date de naissance du patient
- `telephone` : numéro de téléphone
- ...

- mesures avec les attributs :

- `id` : identifiant associé à une mesure, clé primaire
- `type` : le type de mesure stockée : résultat d'analyse sanguine, pression artérielle, radio...
- `datetime` : la date et l'heure de l'examen sous la forme '2016-01-01 00 :00 :00'
- `pid` : identifiant du patient associé à la mesure
- `pdias` : pression diastolique en mmHg.
- `psyst` : pression systolique en mmHg.
- `pouls` : pouls en pulsations par minute.
- ...

Remarque : Les tables contiennent bien d'autres attributs, la base de données d'autres tables, seuls les éléments nécessaires à la suite ont été mis en place.

Pour le besoin du suivi médical des patients, les médecins doivent pouvoir avoir accès à l'évolution des constantes en fonction du temps lors d'une hospitalisation.

À l'aide d'une interface graphique non étudiée ici, le médecin peut sélectionner les données à extraire entre deux dates qui seront stockées dans les variables `time1` et `time2` stockées dans le format de `datetime`. Il souhaite tracer les évolutions des pressions systolique, diastolique et le pouls entre ces deux dates.

Q28. Donner la requête SQL à envoyer au serveur de gestion de la base de données permettant de récupérer la valeur des champs `datetime`, `pdias`, `psyst` et `pouls` entre les deux dates `time1` et `time2` (stockées dans le format de `datetime`).

La requête précédente renvoie toutes les mesures sans distinction de type de mesures et sans que ce soit lié à un unique patient.

Le médecin a sélectionné un patient avec son identifiant unique `id`, cette valeur est stockée dans la variable `id_patient`.

Q29. Donner la requête SQL à envoyer au serveur de gestion de la base de données permettant de récupérer la valeur des champs précédents entre les deux mêmes dates, pour le patient `id_patient` et pour un type de mesures 'tension'.

La requête est envoyée et traitée à l'aide d'une fonction `traitement_requete(requete)` qui retourne un tableau `resultat_requete` dont la première colonne est le champ `datetime`, la deuxième le champ `psyst`, la troisième le champ `pdias` et la dernière `pouls`.

On supposera pour simplifier que la valeur de la colonne `datetime` est comprise comme un entier par le langage de programmation retenu et qu'il peut être utilisé directement comme abscisse des graphiques à produire dans la question suivante.

Q30. Donner la séquence d'instructions permettant de tracer un graphique comportant l'évolution des pressions diastolique et systolique ainsi que le pouls en fonction du temps. Vous veillerez à mettre des titres et légendes sur le graphique.

Vous utiliserez des styles de lignes différents pour chaque ligne dont vous explicitez en commentaire le résultat visuel (couleur, type de trait : continu, pointillé...). Des éléments de documentation sont donnés en annexe.

À côté du graphique, le médecin souhaite avoir les valeurs minimale, maximale, moyenne et médiane de chacune des constantes.

Q31. Ecrire une fonction `analyse(valeurs)` qui prend en argument une liste de valeurs et qui retourne le minimum, le maximum et la moyenne.

Q32. Déterminer la complexité de votre fonction `analyse(valeurs)` en fonction de la taille de la liste de valeurs notée n dans le meilleur et le pire des cas. Vous calculerez pour cela le nombre de comparaisons et d'opérations (addition, soustraction, division, multiplication) réalisées.

Q33. A l'aide d'une méthode de tri de votre choix que vous explicitez, écrire une fonction `mediane(valeurs)` qui prend en argument une liste de valeurs et qui renvoie la valeur de la médiane.

Q34. Déterminer la complexité de votre fonction dans le meilleur et dans le pire des cas en comptant le nombre de comparaisons effectuées. Votre méthode est-elle efficace? Justifier.

Le médecin souhaite obtenir les nom, prénom et le numéro de téléphone des patients qui ont eu une pression systolique supérieure strictement à 160 mmHg, une pression diastolique supérieure strictement à 110 mmHg, avec un pouls compris strictement entre 100 et 150.

Q35. Écrire la requête SQL à envoyer au serveur de gestion de la base de données permettant de répondre à la demande du médecin.

V Annexe - documentation partielle

Les syntaxes entre le langage Python et Scilab sont tout à fait similaires pour les cas standards. La documentation officielle partielle suivante, issue de Python, peut être utilisée dans les deux langages. En Python, on supposera que les fonctions ont été importées et sont directement utilisables.

V.1 Tracer une courbe `plot(*args,**kwargs)`

Plot lines and/or markers to the Axes. `args` is a variable length argument, allowing for multiple `x, y` pairs with an optional format string (`fmt`). For example, each of the following is legal :

```
plot (x, y) # plot x and y using default linestyle and color
plot (x, y, 'bo') # plot x and y using blue circle markers
```

An arbitrary number of `x, y, fmt` groups can be specified, as in :

```
plot (x1, y1, 'g^', x2, y2, 'g-' )
```

Return value is a list of lines that were added.

By default, each line is assigned a different style specified by a 'style cycle'.

The following format string characters are accepted to control the line style or marker :

character	description	character	description
'-'	solid line style	'3'	tri_left marker
'--'	dashed line style	'4'	tri_right marker
'-.'	dash-dot line style	's'	square marker
'.'	dotted line style	'p'	pentagon marker
'.'	point marker	'*'	star marker
','	pixel marker	'h'	hexagon1 marker
'o'	circle marker	'H'	hexagon2 marker
'v'	triangle_down marker	'+'	plus marker
'^'	triangle_up marker	'x'	x marker
'<'	triangle_left marker	'D'	diamond marker
'>'	triangle_right marker	'd'	thin_diamond marker
'1'	tri_down marker	' '	vline marker
'2'	tri_up marker	'_'	hline marker

The following color abbreviations are supported :

character	color
b'	blue
g'	green
r'	red
c'	cyan
m'	magenta
y'	yellow
k'	black
w'	white

Line styles and colors are combined in a single format string, as in 'bo' for blue circles.

V.2 Titres et légende

`xlabel(s)` : Set the x axis label of the current axis with the string `s`.

`ylabel(s)` : Set the y axis label of the current axis with the string `s`.

`title(s)` : Set a title of the current axes with the string `s`.

`legend(*args)` : Places a legend on the axes. `args` must be a list of string.