Concours Commun INP (CCINP), France

Session 2024, filière PSI

Modélisation et ingénierie numérique

Durée : 4 heures

Ce que traite le sujet

- Partie I : redresseur à capacité en tête
- Partie II : correcteur de facteur de puissance
- Partie III : contrôle du correcteur de facteur de puissance par intelligence artificielle

Sujet reproduit sans retouche à partir de la page suivante.

Résumé des parties, chapitres de cours liés et autres années :
prepahelp.com/annales/ccinp/psi/2024/modelisation

Prepa Help ne revendique aucun droit sur le sujet, propriété de l'organisateur du concours.

ÉPREUVE SPÉCIFIQUE - FILIÈRE PSI

MODÉLISATION ET INGÉNIERIE NUMÉRIQUE

Durée : 4 heures

N.B. : le candidat attachera la plus grande importance à la clarté, à la précision et à la concision de la rédaction. Si un candidat est amené à repérer ce qui peut lui sembler être une erreur d'énoncé, il le signalera sur sa copie et devra poursuivre sa composition en expliquant les raisons des initiatives qu'il a été amené à prendre.

RAPPEL DES CONSIGNES

- *Utiliser uniquement un stylo noir ou bleu foncé non effaçable pour la rédaction de votre composition ; d'autres couleurs, excepté le vert, bleu clair ou turquoise, peuvent être utilisées, mais exclusivement pour les schémas et la mise en évidence des résultats.*
- *Ne pas utiliser de correcteur.*
- *Écrire le mot FIN à la fin de votre composition.*

Les calculatrices sont interdites.

Le sujet est composé de trois parties.

Sujet : 22 pages
Annexes : 2 pages

Étude d'un correcteur de facteur de puissance

Présentation générale

Le réseau électrique permet l'acheminement de l'énergie électrique des sites de production (centrales nucléaires, barrages hydroélectriques, parcs éoliens, etc.) vers les consommateurs. Cet acheminement engendre des pertes d'énergie dites *pertes en ligne* qu'il convient de minimiser.

Dans la **partie I** de ce sujet, nous étudions le redresseur dit à *capacité en tête* qui est mis en œuvre dans de nombreux appareils électriques alimentés par le réseau électrique, mais qui ne permet pas de minimiser les pertes en ligne. Dans la **partie II**, un convertisseur statique, appelé *correcteur de facteur de puissance*, est étudié. Lorsqu'il est intercalé entre le réseau et une installation électrique domestique, il minimise les pertes en ligne. Enfin, dans la **partie III**, une stratégie de commande du *correcteur de facteur de puissance* est élaborée par Intelligence Artificielle (IA) dans le but de trouver un compromis entre la minimisation des pertes en ligne et la minimisation des pertes au sein du convertisseur.

Un formulaire et un recueil de fonctions du langage Python sont fournis en annexes au sujet.

Partie I - Redresseur à capacité en tête

I.1 - Présentation du montage

Le schéma électrique du redresseur à capacité en tête est donné sur la **figure 1**. Les diodes D_1, D_2, D_3 et D_4 constituent le pont de Graetz et sont supposées idéales.

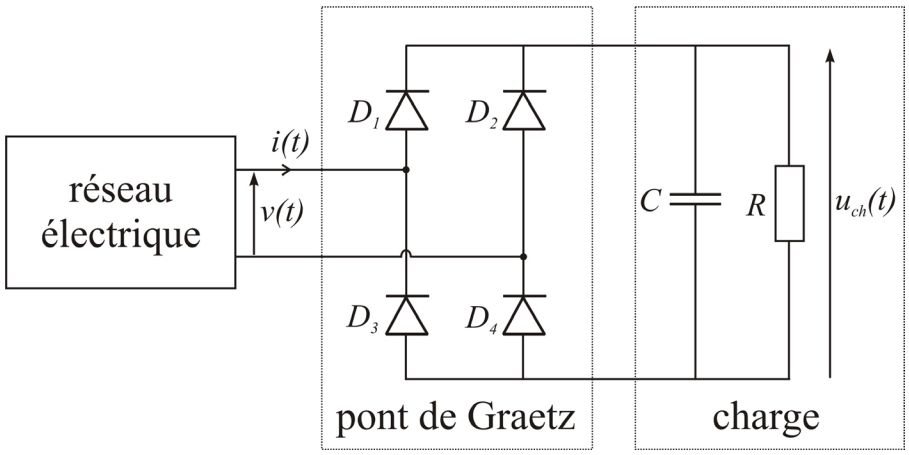


Figure 1 - Schéma électrique du redresseur à capacité en tête

Sauf mention explicite contraire, la tension du réseau sera modélisée dans l'ensemble du sujet par une sinusoïde $v(t) = \hat{V}\cos(2\pi f t)$ avec :

$$\hat{V} = \sqrt{2} \times 230 \simeq 325 \text{ V} \qquad \text{et} \qquad f = \frac{1}{T} = 50 \text{ Hz}. \tag{1}$$

Q1. On admet que si la charge ne comporte pas de condensateur alors :

- les diodes D_1 et D_4 sont passantes et les diodes D_2 et D_3 sont bloquées lorsque $v(t) > 0$;
- les diodes D_2 et D_3 sont passantes et les diodes D_1 et D_4 sont bloquées lorsque $v(t) < 0$.

Tracer alors l'allure de la tension $u_{ch}(t)$ si la charge n'est constituée que du résistor de résistance R.

Q2. Un condensateur de capacité C est placé en parallèle du résistor pour réduire l'ondulation de la tension déterminée à la question précédente. On admet que la tension aux bornes de la charge $u_{ch}(t)$ est quasiment constante lorsque le temps mis par le condensateur à se décharger dans le résistor est très long devant la période de $u_{ch}(t)$. Déterminer une inégalité entre R, C et T pour qu'il en soit ainsi.

La tension aux bornes de la charge sera supposée constante dans tout le reste du sujet.

I.2 - Spectre du courant de ligne

La **figure 2** présente les chronogrammes de la tension du réseau $v(t)$ et du courant de ligne $i(t)$ mesurés expérimentalement en présence du redresseur à capacité en tête.

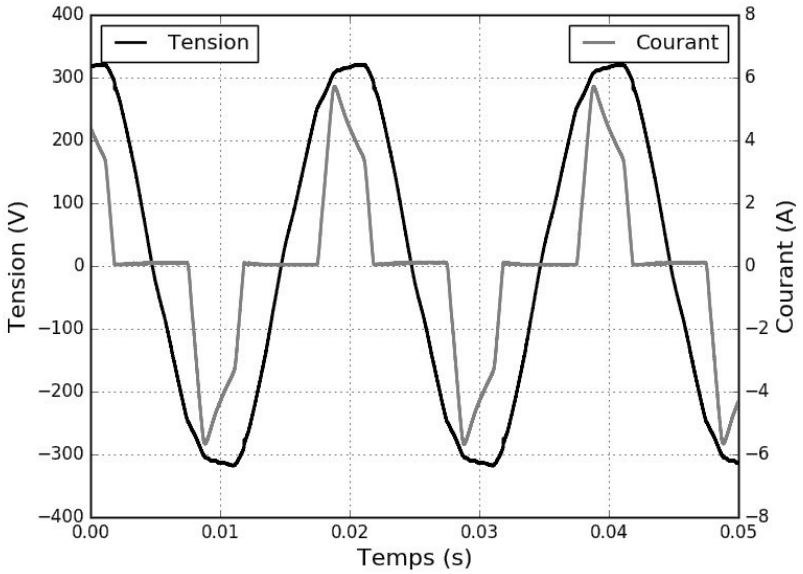


Figure 2 - Chronogrammes de la tension du réseau $v(t)$ et du courant de ligne $i(t)$

Un extrait du programme Python permettant de calculer le spectre du courant de ligne $i(t)$ est donné dans le **tableau 1**. Les bibliothèques numpy et pyplot sont supposées installées de la manière suivante : *import numpy as np* et *import matplotlib.pyplot as plt*. Le courant de ligne *i(t)*, échantillonné à une fréquence de 50 kHz par l'oscilloscope utilisé, est supposé chargé dans un tableau numpy 1D noté " courant ".

Q3. Compléter les **lignes 2**, **4**, **6**, **8**, **10**, **14**, **15**, **16**, **17** et **18** de l'extrait de programme donné dans le **tableau 1**. On rappelle qu'un recueil de fonctions Python utiles est donné en annexe à ce sujet.

N°	Programme Python
1	# Fréquence d'échantillonnage
2	Fe = ...
3	# Nombre d'échantillons
4	N = ...
5	# Calcul de la transformée de Fourier discrète
6	fft_courant = ...
7	# Module de la transformée de Fourier discrète
8	mod_fft_courant = ...
9	# Vecteur fréquence
10	freq = ...
11	
12	# Tracé du spectre
13	plt.figure(1,figsize=(8,6))
14	plt.plot(...,...,color='gray',linewidth=2)
15	plt.xlim(...,...)
16	plt.xlabel(...,fontsize=16)
17	plt.ylim(...,...)
18	plt.ylabel(...,fontsize=16)
19	plt.tick_params(axis='both',labelsize=16)
20	plt.grid()
21	plt.show()

Tableau 1 - Programme de calcul du spectre du courant de ligne

Le développement en série de Fourier (D.S.F.) de $i(t)$ est de la forme :

$$i(t) = \sum_{n=1}^{+\infty} \hat{I}_n \cos(2\pi n f t + \varphi_n). \tag{2}$$

Q4. Donner les valeurs des coefficients $\hat{I}_n$ qui peuvent être déduites graphiquement du spectre en amplitude de $i(t)$ de la **figure 3**, obtenu à partir du programme. Que peut-on dire des harmoniques pairs ?

Q5. Retrouver par le calcul la constatation faite à la **Q4** en remarquant que le courant d'intensité $i(t)$ vérifie la symétrie de glissement : $i(t) = -i(t - T/2)$.

I.3 - Puissance moyenne fournie par le réseau

On note P la puissance **moyenne** fournie par le réseau à l'entrée du pont de Graetz. Comme ce dernier est constitué de quatre diodes supposées idéales, P est également la puissance moyenne reçue par la charge.

Q6. Établir l'expression de P en fonction de $\hat{V}$ et de certains paramètres du D.S.F. de $i(t)$. Commenter. On rappelle qu'un formulaire est donné en annexe.

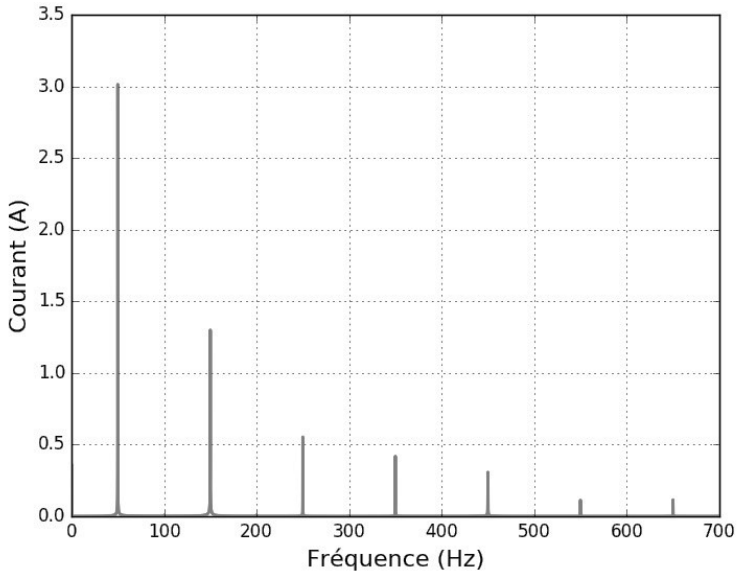


Figure 3 - Spectre en amplitude du courant

Q7. Calculer numériquement P en supposant que $\varphi_1 = 0$.

I.4 - Modélisation du réseau

On admet dans la suite que le réseau électrique peut être modélisé par l'association série d'une source de tension de force électromotrice $e(t)$ sinusoïdale de fréquence de 50 Hz, d'un résistor de résistance r et d'une bobine d'inductance λ (**figure 4**). Lorsque le réseau délivre un courant d'intensité $i(t)$ non sinusoïdal, alors la tension $v(t)$ se trouve légèrement déformée.

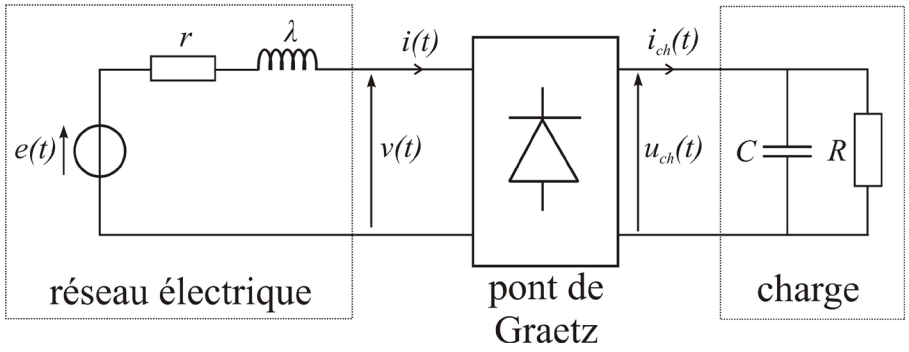


Figure 4 - Modélisation du réseau électrique

Q8. On souhaite enregistrer le chronogramme de la force électromotrice $e(t)$ à l'aide d'un oscilloscope numérique. Expliquer comment procéder.

Les chronogrammes de la différence $e(t) - v(t)$ et du courant d'intensité $i(t)$ sont donnés dans la **figure 5**.

Q9. Expliquer pourquoi les chronogrammes de la **figure 5** conduisent à considérer $\lambda = 0$ en première approximation. Évaluer ensuite le paramètre r.

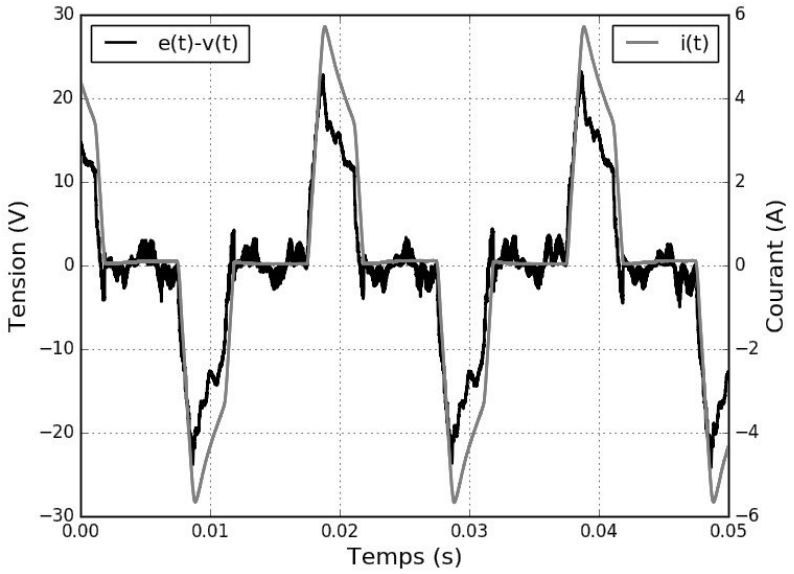


Figure 5 - Chronogrammes de la différence $e(t) - v(t)$ et du courant de ligne $i(t)$

I.5 - Pertes en ligne

On note P_{pertes} les pertes en ligne qui correspondent à la différence entre la puissance moyenne fournie par $e(t)$ et la puissance moyenne P reçue par la charge.

Q10. Donner l'expression littérale de P_{pertes}. Numériquement on trouve $P_{pertes} \simeq 21$ W. Expliquer comment est obtenue cette valeur numérique à partir des valeurs des coefficients $\hat{I}_n$ obtenues à la **Q4** (le calcul numérique complet n'est pas demandé).

Q11. Pour une même puissance moyenne fournie par le réseau électrique, expliquer pourquoi un courant de ligne $i(t)$ sinusoïdal en phase avec la tension du réseau $v(t)$ permet de réduire les pertes en ligne par rapport au cas d'un courant de ligne impulsionnel. Pour le redresseur à capacité en tête étudié, quelle réduction des pertes en ligne obtiendrait-on si le courant de ligne $i(t)$ était sinusoïdal en phase avec $v(t)$? On donnera le résultat sous forme d'un pourcentage.

Dans le but de réduire les pertes en ligne, on étudie dans la deuxième partie de ce sujet un dispositif permettant l'absorption d'un courant de ligne sinusoïdal.

Partie II - Correcteur de facteur de puissance

Un correcteur de facteur de puissance est un dispositif de l'électronique de puissance qu'on intercale entre le réseau et la charge qui permet au courant de ligne d'être sinusoïdal et en phase avec la tension délivrée par le réseau. Le correcteur de facteur de puissance étudié ici est conçu à partir d'un hacheur survolteur.

II.1 - Hacheur survolteur

Le hacheur survolteur est tout d'abord utilisé pour transférer de la puissance entre une source de tension constante et positive U_0 et la charge. Le montage correspondant est donné sur la **figure 6**. Le transistor T_r et la diode D sont supposés idéaux. La tension aux bornes de la charge est supposée constante : $u_{ch}(t) = U_{ch}$, avec $U_{ch} > U_0$. Avec les conventions d'orientation précisées sur la **figure 6**, on note :

- respectivement $u_{Tr}(t)$ et $u_D(t)$ les tensions aux bornes de T_r et de D ;
- $i_L(t)$, $i_{Tr}(t)$ et $i_D(t)$ les courants traversant respectivement la bobine, T_r et D.

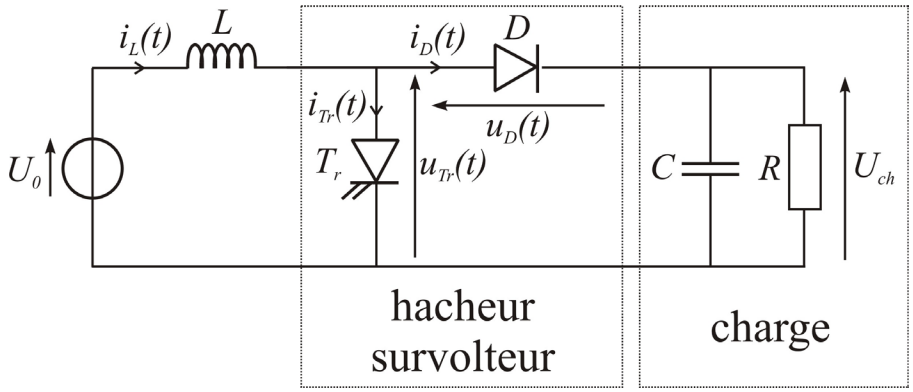


Figure 6 - Montage avec hacheur survolteur

Le signal de commande du transistor T_r est périodique de période T_H, il est tel que :

- pour $t \in [0, \alpha T_H[$, avec $\alpha \in]0, 1[$, le transistor T_r est fermé (on a donc $u_{Tr} = 0$) ;
- pour $t \in [\alpha T_H, T_H]$, le transistor T_r est ouvert (on a donc $i_{Tr} = 0$).

On étudie par la suite le régime établi de conduction continue pour lequel le courant d'intensité $i_L(t)$ dans la bobine ne s'annule pas. On notera I_{min} et I_{max} ses valeurs minimale et maximale respectivement.

Q12. Déterminer $u_{Tr}(t)$, $u_D(t)$, $i_L(t)$, $i_{Tr}(t)$ et $i_D(t)$ pour $t \in [0, \alpha T_H[$. On justifiera soigneusement l'état de la diode D.

Q13. Même question pour $t \in [\alpha T_H, T_H]$.

Q14. Tracer alors les allures des chronogrammes correspondants.

Q15. Établir la relation existant entre α, U_{ch} et U_0 dans ce régime de fonctionnement.

Q16. On a mesuré $\Delta I = I_{max} - I_{min} = 0,50$ A pour $\alpha = 0,50$, $U_0 = 80$ V et $T_H = 1,0 \cdot 10^{-3}$ s. Calculer numériquement la valeur de L.

II.2 - Principe du correcteur de facteur de puissance et commande du transistor

Le circuit de puissance du correcteur de facteur de puissance étudié est obtenu en remplaçant la source de tension constante U_0 par l'association du réseau et du pont de Graetz (**figure 7**).

On note $i_{L,cons}(t) = |\hat{I}\cos(2\pi f t)|$, le courant dans la bobine qui est associé à un courant de ligne de la forme $i(t) = \hat{I}\cos(2\pi f t)$ avec $\hat{I} > 0$, c'est-à-dire un courant de ligne sinusoïdal et en phase avec la tension du réseau $v(t) = \hat{V}\cos(2\pi f t)$.

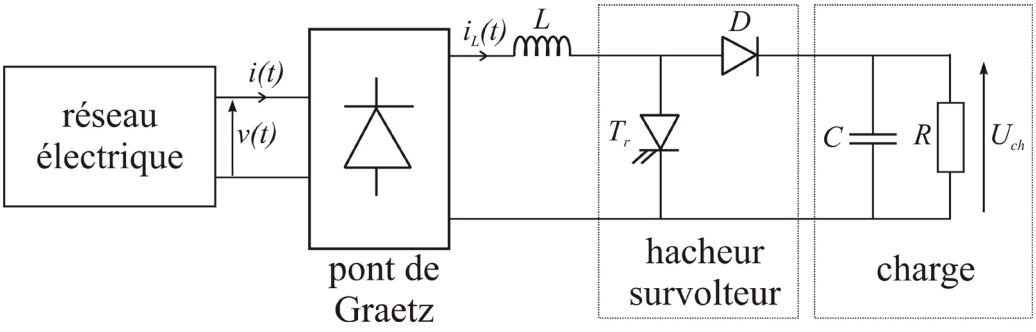


Figure 7 - Circuit de puissance du correcteur de facteur de puissance

Le principe de ce correcteur de facteur de puissance est de commander le transistor T_r pour que le courant d'intensité $i_L(t)$ soit le plus proche possible de $i_{L,cons}(t) = |\hat{I}\cos(2\pi f t)|$. Pour cela on dispose :

- d'une tension de consigne $v_{i_L,cons}$, telle que $v_{i_L,cons} = k\, i_{L,cons}$ avec $k = 1,0\ \text{V}\cdot\text{A}^{-1}$;
- d'une tension $v_{i_L,mes}$ image du courant d'intensité i_L, telle que $v_{i_L,mes} = k\, i_L$.

La loi commande adoptée est, à tout instant, construite à partir d'un paramètre $\Delta > 0$ comme suit :

- si $|v_{i_L,cons} - v_{i_L,mes}| < \Delta$ alors l'état du transistor est inchangé ;
- si $v_{i_L,mes} - v_{i_L,cons} > \Delta$ alors le transistor doit passer à l'état bloqué ;
- si $v_{i_L,mes} - v_{i_L,cons} < -\Delta$ alors le transistor doit passer à l'état passant.

On rappelle sur la **figure 8** trois montages à base d'Amplificateurs Linaires Intégrés (ALI) et les relations entrée(s)-sortie correspondantes obtenues avec le modèle de l'ALI idéal de gain infini. On note V_{sat} la tension de saturation supposée identique pour les trois ALI considérés.

Le signal de commande u_c du transistor T_r est tel que :

- si $u_c = -V_{sat}$ alors T_r est à l'état passant,
- si $u_c = +V_{sat}$ alors T_r est à l'état bloqué.

Q17. Proposer un circuit de commande du transistor T_r mettant en jeu deux des trois montages de la **figure 8**. Expliciter alors le paramètre Δ en fonction des résistances utilisées et de V_{sat}.

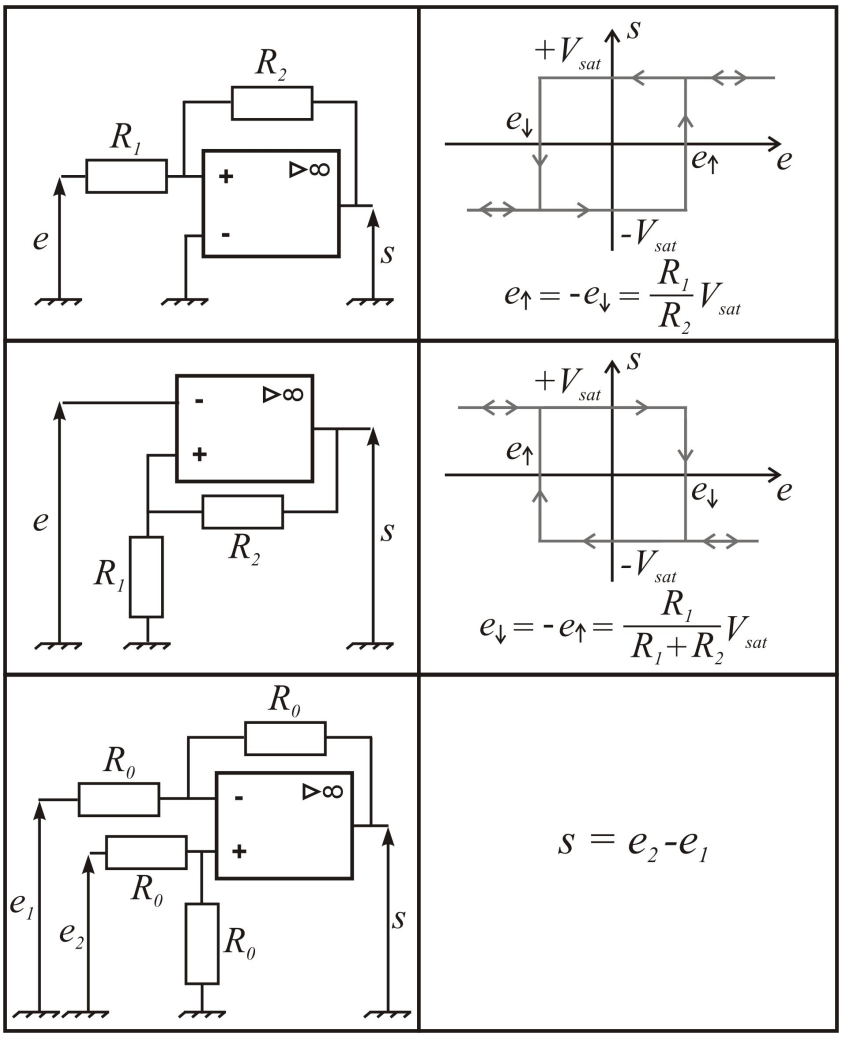


Figure 8 - Trois montages à base d'ALI.

II.3 - Mise en œuvre

Les **figures 9**, **10** et **11** présentent les relevés de la tension $v(t)$ et du courant d'intensité $i(t)$ pour trois valeurs du paramètre Δ.

Q18. Commenter l'allure des chronogrammes de la **figure 9**. Comment le paramètre Δ se manisfeste-t-il dans ces chronogrammes ? Évaluer la valeur de Δ_1 dans le cas de la **figure 9**. Comment évolue la valeur du paramètre Δ de la **figure 9** à la **figure 11** ?

Q19. Décrire l'allure du spectre en amplitude du courant de ligne $i(t)$. Comment l'allure du spectre est-elle modifiée lorsque le paramètre Δ passe de la valeur Δ_1 (**figure 9**) à la valeur Δ_3 (**figure 11**). On distinguera clairement les cas des parties " basses fréquences " et " hautes fréquences " du spectre.

Q20. Quelle caractéristique des ALI des montages de la **figure 8** constitue une limite technique à la diminution de Δ ?

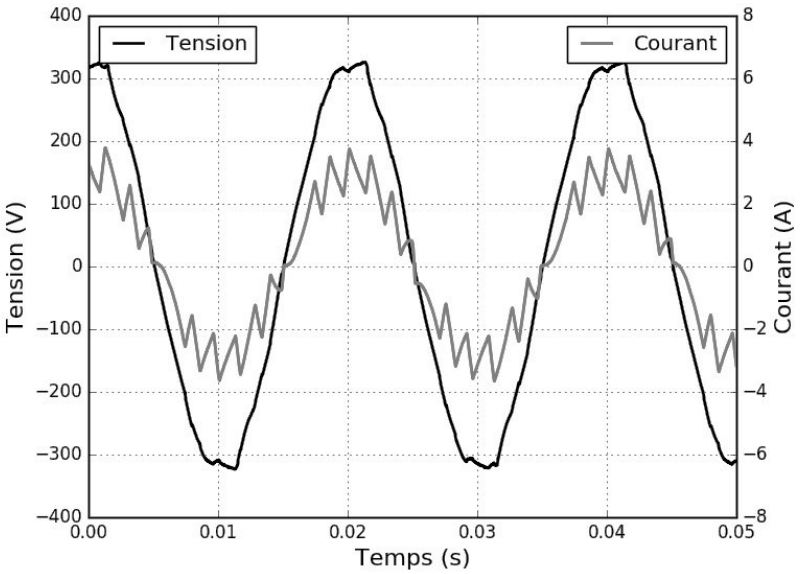


Figure 9 - Chronogrammes
de la tension $v(t)$ et du courant d'intensité $i(t)$ pour Δ_1

II.4 - Dissipation thermique dans le transistor

Lors du fonctionnement du correcteur de facteur de puissance, le transistor T_r commute K fois par période T de la tension $v(t)$ délivrée par le réseau électrique, ce qui correspond à une fréquence de commutation moyenne du transistor : $f_{cm} = \dfrac{K}{T}$.

Or K est d'autant plus grand que le paramètre Δ est petit (**figure 9** à **figure 11**), on en déduit que la fréquence de commutation moyenne du transistor est une fonction décroissante du paramètre Δ. Ceci est illustré par le relevé expérimental présenté sur la **figure 12**.

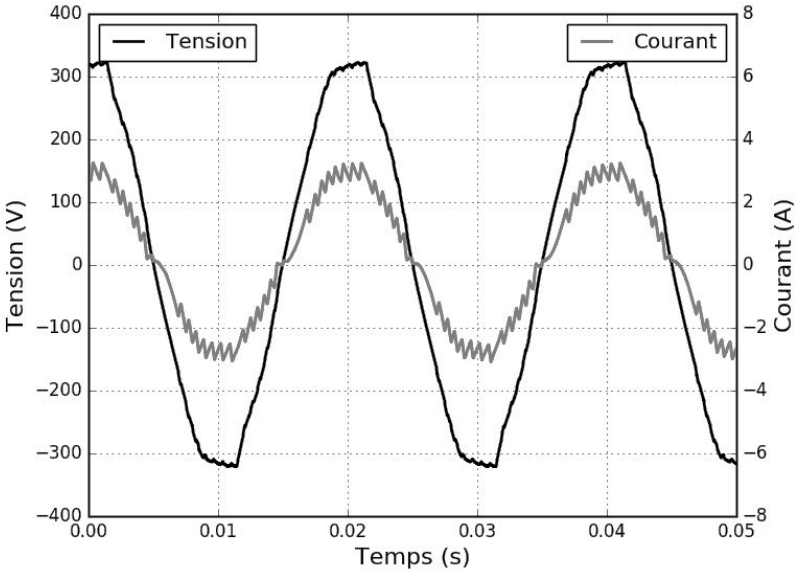


Figure 10 - Chronogrammes
de la tension $v(t)$ et du courant d'intensité $i(t)$ pour Δ_2

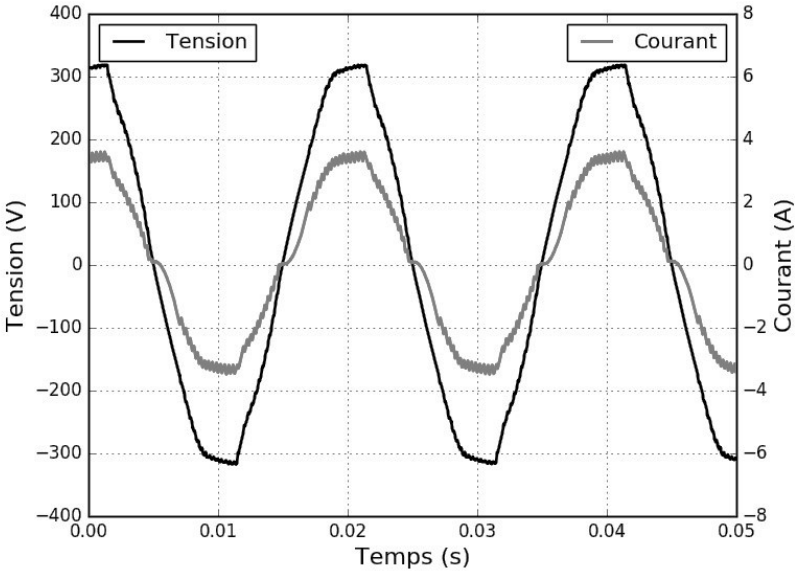


Figure 11 - Chronogrammes
de la tension $v(t)$ et du courant d'intensité $i(t)$ pour Δ_3

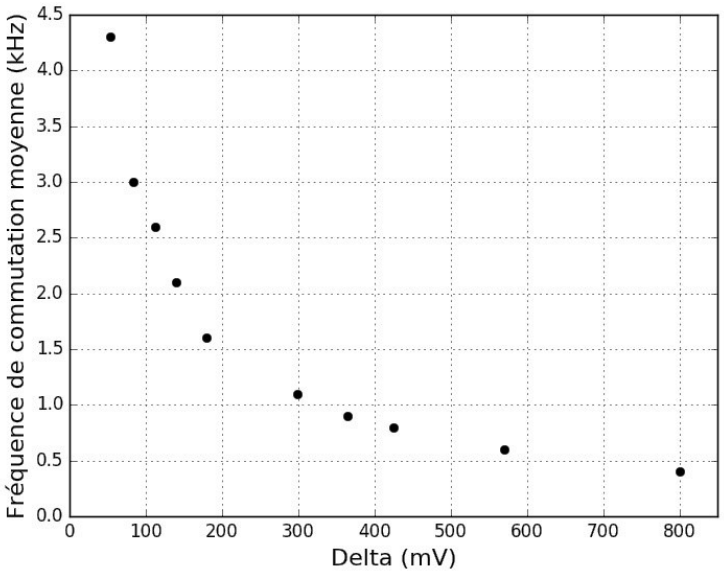


Figure 12 - Fréquence de commutation moyenne en fonction du paramètre Δ

Lors de son fonctionnement, le transistor dissipe sous forme thermique une partie de la puissance électrique qui transite dans le correcteur de facteur de puissance. Ces pertes sont principalement de deux types :

- les *pertes par conduction* P_{cond} qui dépendent uniquement de la valeur efficace du courant circulant dans le transistor ;
- les *pertes par commutation* P_{com} qui sont proportionnelles à la fréquence de commutation moyenne. À chaque commutation (blocage ou amorçage) du transistor une quantité d'énergie $E_{on/off}$ est dissipée sous forme thermique au sein du transistor.

Un dissipateur thermique est monté sur le transistor. On a relevé, en régime permanent, la température T_{dis} à la surface de ce dissipateur pour différentes valeurs du paramètre Δ. La précision des mesures de température est de $\pm$ 1 ^{o}C et les pertes par conduction sont supposées constantes lors cette expérience.

Sur la **figure 13**, les températures mesurées sont tracées en fonction de la fréquence de commutation moyenne correspondante.

Une régression affine donne : $T_{dis} = a\, f_{cm} + b$ avec a = 5,5 ^{o}C·kHz^{-1} et b = 40 ^{o}C.

Q21. Le modèle affine choisi est-il en cohérence avec les mesures effectuées ? Justifier.

On donne :

- la température de l'air ambiant loin du dissipateur : T_{air} = 20 ^{o}C ;
- la résistance thermique entre le transistor et la surface du dissipateur : $R_{th,\,Tr-dis}$ = 1,0 SI ;
- la résistance thermique entre la surface du dissipateur et l'air ambiant : $R_{th,\,dis-air}$ = 4,0 SI.

Q22. Proposer, sous la forme d'un schéma électrique équivalent, une modélisation du comportement thermique du transistor en régime permanent. En déduire les expressions des paramètres a et b.

Q23. Déterminer les valeurs numériques de P_{cond} et de $E_{on/off}$.

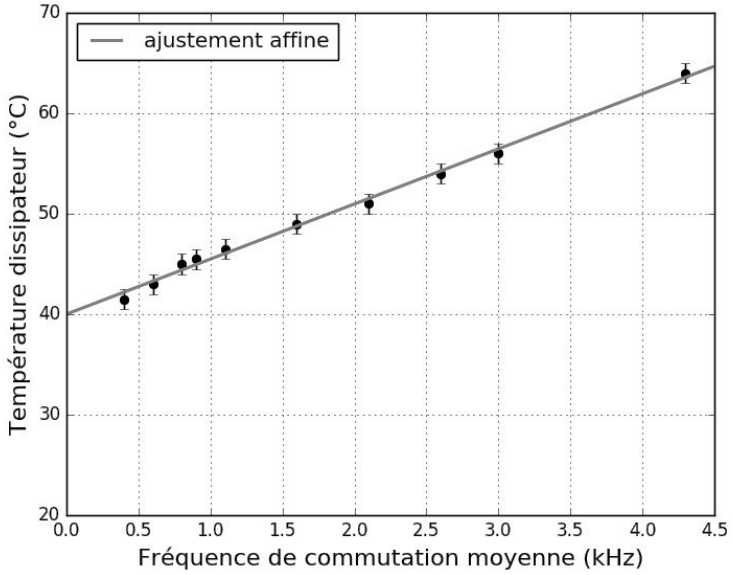


Figure 13 - Température du dissipateur en fonction de la fréquence moyenne de commutation

Partie III - Contrôle du correcteur de facteur de puissance par Intelligence Artificielle

III.1 - Présentation du problème

Dans cette dernière partie, le correcteur de facteur de puissance, étudié dans la **partie II**, est utilisé pour minimiser les pertes en lignes engendrées par l'installation électrique d'un petit studio.

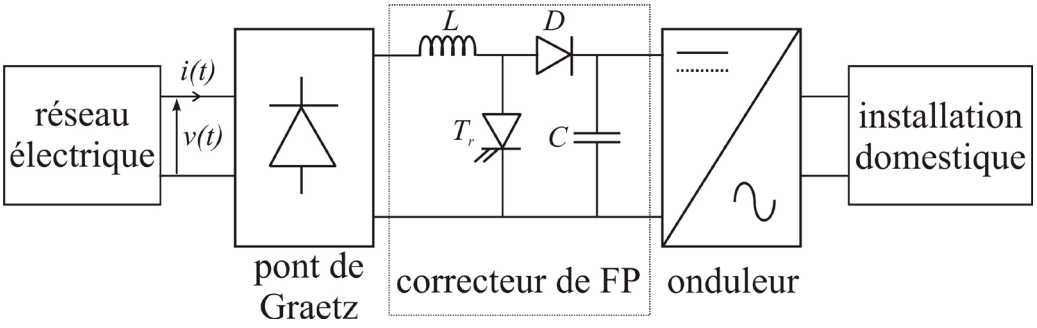


Figure 14 - Correcteur de facteur de puissance utilisé pour une installation domestique

La **figure 14** présente le schéma de principe du dispositif complet : la tension aux bornes du condensateur C est placée en entrée de l'association d'un onduleur de tension et de l'installation électrique du studio. Une boucle de régulation, non étudiée ici, permet de réguler cette tension à une valeur constante, de sorte que le fonctionnement du correcteur de facteur de puissance est identique à celui de la **partie II** (**figure 9** à **figure 11**).

L'installation électrique étudiée comporte :

- un ensemble box Internet-télévision,
- un système d'éclairage,
- un radiateur électrique,
- un chauffe-eau.

Q24. Les différents éléments de cette installation sont-ils câblés en série ou en dérivation ? Justifier ce choix.

L'ensemble box Internet-télévision et le système d'éclairage absorbent des courants impulsionnels semblables à celui de la **figure 2**. En revanche, leur fonctionnement étant basé sur l'utilisation d'une résistance chauffante, le radiateur électrique et le chauffe-eau ont un comportement purement résistif.

Q25. Quelle est l'allure des courants absorbés par le radiateur électrique et le chauffe-eau lorsqu'ils sont connectés directement au réseau électrique ?

Chacun des quatre éléments de l'installation électrique considérée présente trois modes de fonctionnement. Un niveau x_i valant 0, 1 ou 2 est associé à chacun de ces modes. Le **tableau 2** donne la correspondance entre le niveau x_i et le mode de fonctionnement de chaque élément. Certaines des puissances consommées par l'élément correspondant sont également données.

	box-TV x_1	éclairage x_2	radiateur x_3	chauffe-eau x_4
$x_i = 0$	éteint 0 W	éteint 0 W	éteint 0 W	éteint 0 W
$x_i = 1$	veille 100 W	puissance moitié 50 W	puissance moitié	maintient à température 50 W
$x_i = 2$	allumé 200 W	puissance maximale 100 W	puissance maximale	puissance maximale

Tableau 2 - Modes de fonctionnement

Q26. Donner un ordre de grandeur de la puissance électrique maximale consommée par un radiateur électrique et un chauffe-eau correspondant aux besoins d'un studio.

Q27. Combien y a-t-il de combinaisons possibles entre les différents modes de fonctionnement de chacun des quatre éléments ?

Comme cela a été vu dans la **partie II**, le dispositif de la **figure 14** permet de réduire les pertes en ligne engendrées par l'absorption d'un courant impulsionnel. Celles-ci tendent vers leur valeur minimale lorsque le paramètre Δ tend vers 0. Cependant, cette diminution se fait au prix d'une augmentation de la fréquence de commutation moyenne f_{cm} et donc des pertes au sein du transistor T_r (cf sous-partie II.4). Si la puissance dissipée dans T_r ne peut pas être exploitée, il est alors nécessaire de rechercher une valeur du paramètre Δ correspondant à une minimisation des pertes totales : pertes en ligne + pertes dans T_r.

La valeur du paramètre Δ permettant de minimiser les pertes totales, notée Δ_{opt}, a été évaluée empiriquement pour quatorze combinaisons entre les différents modes de fonctionnement des quatre éléments de l'installation électrique étudiée. Ces valeurs sont présentées dans le **tableau 3**. Le paramètre Δ s'exprime en volts et est positif. Lorsque la valeur -1 est indiquée pour Δ_{opt}, cela signifie que pour minimiser les pertes totales, il est préférable de désactiver le correcteur de facteur de puissance et de connecter l'installation directement au réseau électrique.

	box-TV x_1	éclairage x_2	radiateur x_3	chauffe-eau x_4	Δ_{opt}(V)
Données d'entraînement	0	0	0	1	-1
	0	0	1	1	-1
	0	0	2	0	-1
	0	0	0	2	-1
	0	0	1	2	-1
	0	1	0	0	0,18
	2	2	0	0	0,32
	2	2	2	0	0,49
	2	2	2	2	-1
Données test	0	0	1	0	-1
	0	0	2	1	-1
	0	0	2	2	-1
	1	0	0	0	0,22
	2	2	1	0	0,42

Tableau 3 - Valeurs optimales du paramètre Δ pour quatorze combinaisons différentes

Q28. Dans le **tableau 3**, expliquer pourquoi Δ_{opt} = -1 pour toutes les combinaisons telles que $x_1 = 0$ et $x_2 = 0$.

Q29. Commenter le choix Δ_{opt} = -1 pour la combinaison $(x_1, x_2, x_3, x_4) = (2, 2, 2, 2)$.

III.2 - Contrôle du paramètre Δ avec un réseau de neurones

Il faut désormais mettre en place un contrôleur capable de fournir la consigne Δ_{opt} en fonction des différentes combinaisons (x_1, x_2, x_3, x_4). Le contrôleur associé au correcteur de facteur de puissance doit permettre en temps réel :

- soit de désactiver le correcteur de facteur de puissance en connectant l'installation électrique du studio directement au réseau électrique (cas $\Delta = -1$) ;
- soit de fixer de la valeur du paramètre Δ lorsque le correcteur de facteur de puissance est activé.

Q30. Proposer un mode de communication pour que les différents appareils électriques de l'installation puissent transmettre leurs modes de fonctionnement au contrôleur. Aucun détail technique précis n'est attendu pour cette question.

Le **tableau 3** nous donne la correspondance entre les états des appareils x_i et le réglage Δ du correcteur. Les données d'entraînement sont celles qui vont servir à l'entraînement du réseau de neurones. Les données de tests sont celles qui vont permettre de valider si l'entraînement s'est bien effectué. D'un point de vue système, cela revient à chercher la relation entre des données d'entrées x_i et la sortie Δ (**figure 15**) qui illustre les entrées-sortie du système.

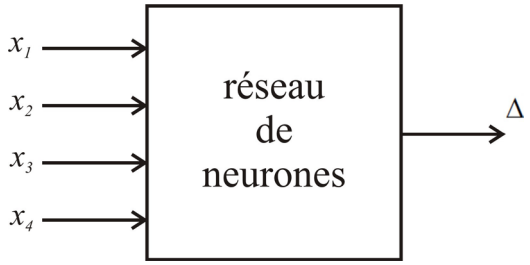


Figure 15 - Entrées-sortie du réseau de neurones.

L'objectif est de trouver un modèle de comportement qui permet, à partir du **tableau 3**, de :

- retrouver les valeurs d'entraînement,
- prédire les valeurs de test.

Pour y arriver, l'idée est d'entraîner un réseau de neurones de manière supervisée à partir des données du **tableau 3**.

Q31. Quels sont les avantages et les inconvénients du choix d'un réseau de neurones pour modéliser la commande ?

Q32. L'apprentissage supervisé est-il pertinent ? Justifiez votre réponse.

III.3 - Définition du perceptron

Le réseau de neurones choisi est un perceptron multicouche à une première couche cachée (couche d'entrée), une deuxième couche cachée et une couche de sortie. Le système prend en entrée un vecteur à quatre composantes (x_1, x_2, x_3, x_4). La couche d'entrée est composée de quatre neurones numérotés de 1 à 4. La deuxième couche cachée contient également

quatre neurones numérotés de 1' à 4'. La couche de sortie contient un seul neurone numéroté 1". La sortie de la couche de sortie est Δ. Chaque neurone prend en entrée les sorties de la couche précédente et renvoie une unique sortie.

La **figure 16** présente cette structure.

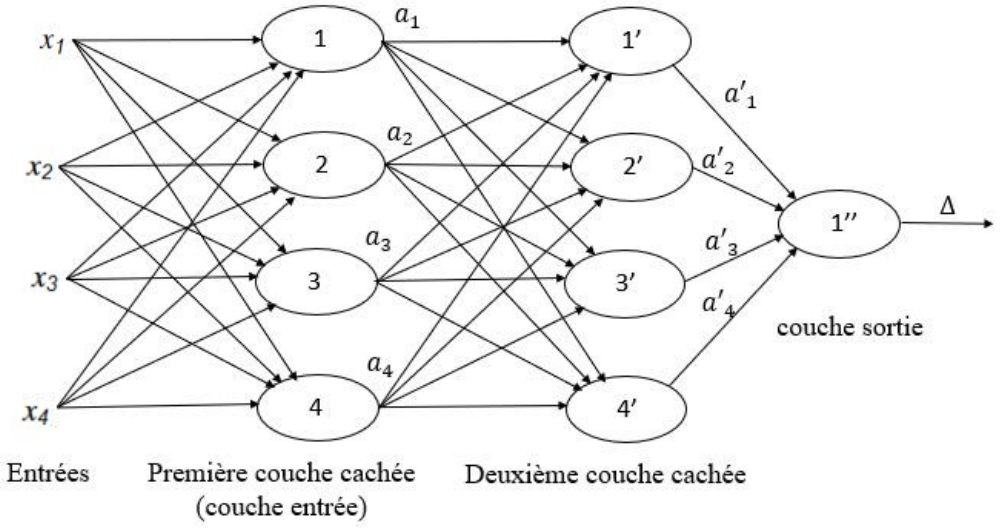


Figure 16 - Structure du réseau de neurones choisi

Le détail d'un neurone est défini par la **figure 17**.

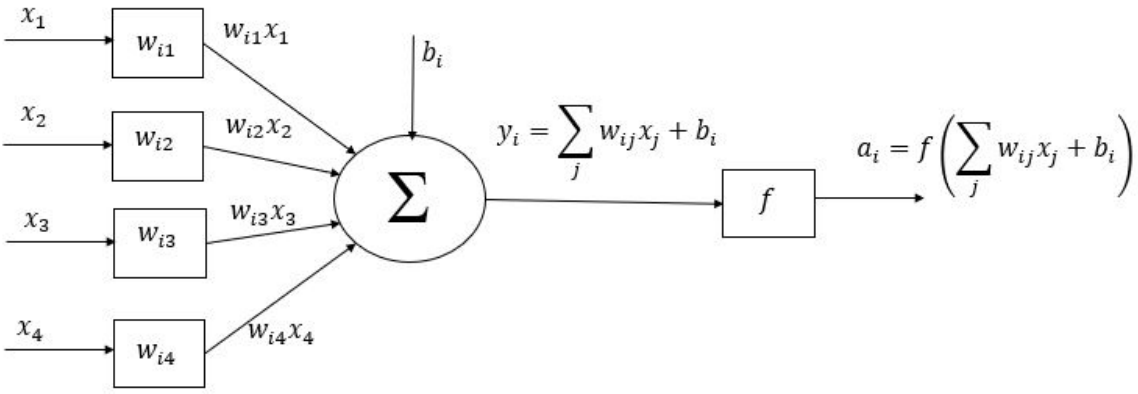


Figure 17 - Structure d'un neurone i de la première couche

La **figure 17** d'un neurone numéroté " i " de la première couche permet, à partir des valeurs d'entrée (x_1, x_2, x_3, x_4), de calculer la valeur de sortie a_i. Les sorties de la première couche deviennent les entrées de la couche suivante et ainsi de suite. Tous les neurones vont se comporter comme ceux de la première couche. Il est à noter que la sortie du dernier neurone correspond à Δ.

Nous noterons :

- w_{ij} : le poids entre l'entrée j et le neurone i, c'est un nombre réel ;
- b_i : le biais du neurone i, c'est un nombre réel ;
- f : la fonction d'activation du neurone i qui s'applique à $\sum_j w_{ij}x_j + b_i$.

Remarque : le processus qui permet de trouver les valeurs de b_i et de w_{ij} va être itératif et il faut donc des valeurs initiales. Celles-ci sont choisies aléatoirement.

La fonction d'activation choisie est une sigmoïde définie par $f : \mathbb{R} \longrightarrow \mathbb{R}, x \longrightarrow f(x) = \dfrac{1}{1+e^{-x}}$.
En Python, l'utilisation de la fonction exp de la librairie numpy permet de calculer, pour une entrée X de type tableau, la matrice $f(X)$, issue de l'application de la fonction f à tous les éléments de la matrice X. Cette fonction d'activation se traduit par :

N°	Programme Python
1	def f(x) :
2	return(1/(1+np.exp(-x)))

Tableau 4 - Fonction d'activation

Q33. Calculer la dérivée de la fonction f notée $f'(x)$ et écrire une fonction Python notée f_prime qui prend en argument x et qui renvoie la dérivée $f'(x)$. L'argument d'entrée x est de type tableau.

III.4 - Phase d'inférence

Notation :

- les grandeurs scalaires seront notées en minuscule, par exemple w ;
- les grandeurs matricielles seront notées en majuscule, par exemple W ;
- l'indice k fait référence à la couche, par exemple W_2 est pour la deuxième couche.

La phase d'inférence consiste à calculer la sortie à partir d'un vecteur d'entrée connu.

La sortie du dernier neurone du réseau de la **figure 16** est la valeur Δ recherchée et peut s'écrire sous la forme suivante :

$$\Delta = f(W_3(f(W_2 f(W_1 X + B_1) + B_2) + b_3)).$$

Q34. Donner les dimensions des matrices W_2, W_3, B_2 et b_3.

On cherche à modéliser la couche d'entrée du réseau de neurones de la **figure 16**.

Q35. Donner l'expression de la matrice W_1 qui vérifie :

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = W_1 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} - & - & - & - \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}.$$

Q36. Écrire en Python la fonction inference_couche qui va prendre en argument le vecteur d'entrées noté X, la matrice des poids notée W, le vecteur de biais noté B et qui renvoie le vecteur des sorties de la couche noté A.

La **figure 18** montre le tracé de la fonction d'activation sigmoïde.

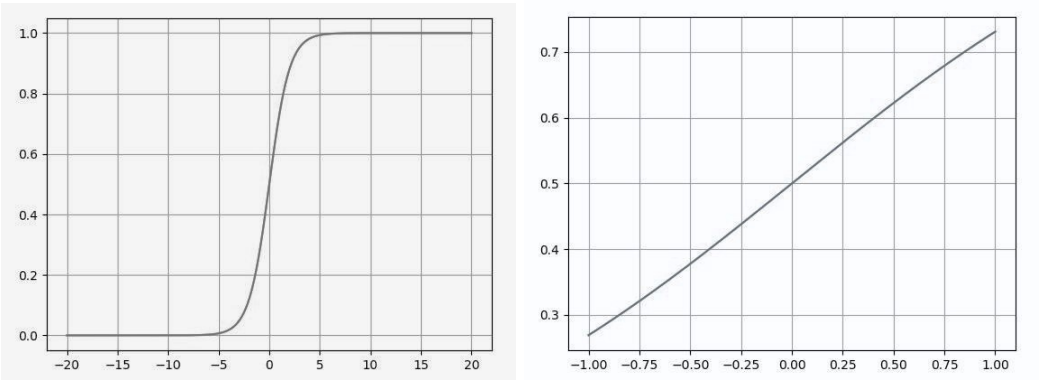


Figure 18 - Représentation graphique de la fonction sigmoïde f pour $x \in [-20, 20]$ (gauche) et $x \in [-1, 1]$ (droite)

Q37. Calculer $A = f(Y)$ dans le cas où :

$$A = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}; Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}; X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}; W = \begin{bmatrix} 0,3 & 0,2 & 0,25 & -0,04 \\ 0,4 & -0,3 & 0,6 & -0,3 \\ -0,4 & 0,3 & -0,6 & -0,4 \\ -1 & -0,75 & -0,1 & -0,5 \end{bmatrix}; B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Notons que, pour cette première phase d'apprentissage qu'on appelle souvent l'initialisation, les valeurs des poids ont été générées aléatoirement et les biais choisis nuls. En reproduisant le même calcul pour les couches suivantes, nous calculons que $\Delta = 0,59\ V$. Cela signifie que pour l'éclairage allumé et tous les autres appareils éteints, il faudrait un réglage $\Delta = 0,59\ V$.

Cependant, le **tableau 3** indique que pour cette combinaison d'états, le réglage optimal est $\Delta_{opt} = 0,18\ V$.

Il faut modifier les valeurs des poids et des biais pour que la sortie Δ du réseau de neurones se rapproche de Δ_{opt} et ce, pour toutes les combinaisons d'entrée possible. Autrement dit, pour toute combinaison d'entrée (x_1, x_2, x_3, x_4), on souhaite minimiser l'erreur (choisie quadratique ici) entre la sortie Δ calculée par le réseau de neurones, et le réglage optimal Δ_{opt}, déterminé à l'avance. Pour ne pas alourdir les notations on notera désormais t le réglage optimal Δ_{opt}. Le calcul de l'erreur $(t - \Delta)^2$ va être utilisé pour calculer des " meilleures " valeurs de poids et de biais : c'est la phase d'entraînement qui fait l'objet de la sous-partie suivante.

III.5 - Rétropropagation

Nous disposons de N = 9 données d'entraînement, c'est à dire toutes les paires $\{X_\ell, t_\ell\}$ où $X_\ell = (x_1, x_2, x_3, x_4)$ est le vecteur de la combinaison des entrées et t_ℓ est simplement la valeur optimale Δ_{opt} , évaluée empiriquement et qui constitue la valeur cible souhaitée en sortie du réseau de neurones.

Ces données d'entraînement sont recensées dans le **tableau 3**. Pour chaque donnée d'entraînement $\{X_\ell, t_\ell\}$, on notera Δ_ℓ le résultat réellement renvoyé par le réseau de neurones. Le but de la phase de rétropropagation est de calculer des valeurs de paramètres W et B qui " rapprochent " tous les Δ_ℓ calculés pour les X_ℓ des valeurs optimales t_ℓ. Une instance de rétropropagation s'effectue en cherchant les valeurs des paramètres W et B qui minimisent la fonction de coût quadratique entre une cible t_ℓ et la valeur calculée Δ_ℓ : $\mathcal{L}_\ell = (t_\ell - \Delta_\ell)^2$.

Pour des raisons de lisibilité, on notera $\mathcal{L} = \mathcal{L}_\ell$.

Pour ce faire, les poids W et les biais B sont mis à jour par la méthode de la descente du gradient, étudiée dans les questions ci-après. Afin de se fixer les idées, on s'intéresse à la première couche du réseau de neurones, ayant pour entrée le vecteur X et pour sortie le vecteur d'activations A : $A = f(W_1X + B_1)$. Le fonctionnement des autres couches sera identique. Pour des raisons de lisibilité, on posera désormais : $W = W_1$, $B = B_1$, $Y = WX + B$. On a donc :

$$A = f(WX + B) = f(Y).$$

Les valeurs des paramètres W et B de cette couche sont mis à jour grâce aux formules suivantes :

$$\begin{cases} W = W - \alpha \dfrac{\partial \mathcal{L}}{\partial W} \\ \\ B = B - \alpha \dfrac{\partial \mathcal{L}}{\partial B} \end{cases}$$

où α est le taux d'apprentissage, un paramètre à choisir manuellement. On adopte la notation suivante :

$$\frac{\partial \mathcal{L}}{\partial W} = \begin{bmatrix} \dfrac{\partial \mathcal{L}}{\partial w_{11}} & \cdots & \dfrac{\partial \mathcal{L}}{\partial w_{14}} \\ \vdots & \cdots & \vdots \\ \dfrac{\partial \mathcal{L}}{\partial w_{41}} & \cdots & \dfrac{\partial \mathcal{L}}{\partial w_{44}} \end{bmatrix}, \quad \frac{\partial \mathcal{L}}{\partial X} = \begin{bmatrix} \dfrac{\partial \mathcal{L}}{\partial x_1} \\ \vdots \\ \dfrac{\partial \mathcal{L}}{\partial x_4} \end{bmatrix}, \quad \frac{\partial \mathcal{L}}{\partial A} = \begin{bmatrix} \dfrac{\partial \mathcal{L}}{\partial a_1} \\ \vdots \\ \dfrac{\partial \mathcal{L}}{\partial a_4} \end{bmatrix}.$$

Nous avons $\dfrac{\partial \mathcal{L}}{\partial w_{ij}} = \dfrac{\partial \mathcal{L}}{\partial a_1}\dfrac{\partial a_1}{\partial w_{ij}} + ... + \dfrac{\partial \mathcal{L}}{\partial a_i}\dfrac{\partial a_i}{\partial w_{ij}} + ... + \dfrac{\partial \mathcal{L}}{\partial a_4}\dfrac{\partial a_4}{\partial w_{ij}}$ et $a_i = f\left(\sum_j w_{ij}x_j + b_i\right)$.

Q38. Montrer que $\dfrac{\partial \mathcal{L}}{\partial w_{ij}} = \dfrac{\partial \mathcal{L}}{\partial a_i} f'(y_i) x_j$.

Q39. En utilisant la relation précédente, donner la relation matricielle entre $\dfrac{\partial \mathcal{L}}{\partial W}, \dfrac{\partial \mathcal{L}}{\partial A}, f'(Y)$ et X, puis entre $\dfrac{\partial \mathcal{L}}{\partial W}, \dfrac{\partial \mathcal{L}}{\partial A}$,W, X et B.

En poursuivant l'analyse il est possible de trouver la relation suivante qui va permettre d'exprimer l'erreur en entrée en fonction de l'erreur en sortie :

$$\frac{\partial \mathcal{L}}{\partial X} = \frac{\partial \mathcal{L}}{\partial A}\left[W \cdot f'(WX + b)\right].$$

Il est à remarquer que dans la formule précédente la multiplication matricielle se fait terme à terme.

Q40. Écrire la fonction Python retropropagation-couche qui prend en argument :

- la valeur d'entrée d'une couche, notée X,
- la matrice de poids d'une couche, notée W,
- le vecteur de biais de la couche, noté B,
- le vecteur d'erreur en sortie de la couche, noté Ea,
- la valeur du coefficient d'apprentissage, notée alpha.

Cette fonction doit calculer :

- $E_W = \frac{\partial \mathcal{L}}{\partial W}$,
- $E_B = \frac{\partial \mathcal{L}}{\partial B}$,
- $E_X = \frac{\partial \mathcal{L}}{\partial X}$.

Cette fonction doit renvoyer :

- la matrice de poids W mise à jour ,
- le vecteur de biais B mis à jour,
- le vecteur gradient du coût $E_X = \frac{\partial \mathcal{L}}{\partial X}$.

Ainsi, après la phase d'initialisation, en appliquant une succession de phase d'inférence et de rétropropagation pour chacune des données d'entraînement, et ce, plusieurs fois, on peut faire diminuer la valeur de la fonction de coût total jusqu'à atteindre la précision demandée :

$$L_{tot} = 1/N \sum_{\ell=1}^{N} (t_\ell - \Delta_\ell)^2 = 1/N \sum_{\ell=1}^{N} L_\ell.$$

III.6 - Analyse des résultats

L'évolution de la fonction de coût total L_{tot} pendant la phase d'entraînement est donnée sur la **figure 19**. Elle comporte 100 itérations.

Le graphique montre que la phase d'entraînement, constituée d'une succession d'inférences et de rétropropagation, a permis de diminuer la fonction coût, c'est-à-dire l'erreur quadratique moyenne entre le résultat Δ du modèle et les valeurs cibles des données d'entraînement. Il y a désormais peu d'écart pour les données d'entraînement.

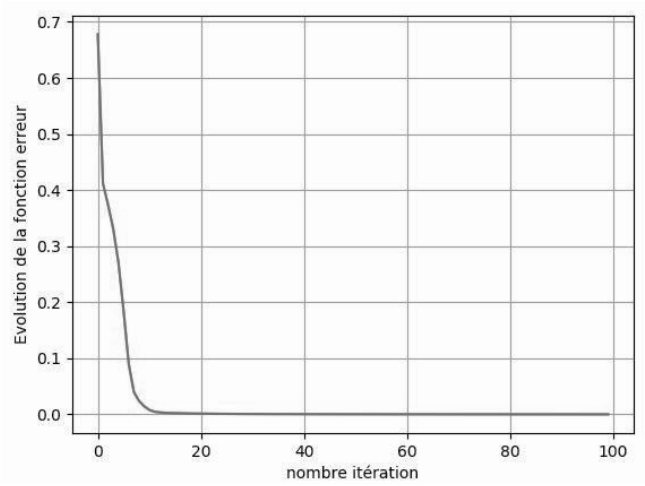


Figure 19 - Entraînement modèle

	box-TV x_1	éclairage x_2	radiateur x_3	chauffe-eau x_4	Δ_{opt} (V)	Δ (V)
Données d'entraînement	0	0	0	1	-1	- 0,981
	0	0	1	1	-1	- 0,979
	0	0	2	0	-1	- 0,961
	0	0	0	2	-1	- 0,983
	0	0	1	2	-1	- 0,980
	0	1	0	0	0,18	0,178
	2	2	0	0	0,32	0,341
	2	2	2	0	0,49	0,474
	2	2	2	2	-1	- 0,970
Données test	0	0	1	0	-1	- 0,919
	0	0	2	1	-1	- 0,977
	0	0	2	2	-1	- 0,978
	1	0	0	0	0,22	0,015
	2	2	1	0	0,42	0,412

Tableau 5 - Valeurs optimales du paramètre Δ pour quatorze combinaisons différentes

Le **tableau 5** présente les données d'entraînement et les données tests, et sa dernière colonne recense les valeurs Delta calculées par le réseau de neurones.

Sur les 14 données disponibles, N = 9 données sont utilisées lors de la phase d'entraînement et les 5 autres sont réservées comme données tests, afin de tester la robustesse du modèle en soumettant des entrées non utilisées pour la phase d'entraînement, afin de vérifier si la prédiction du modèle est bonne même sur des données nouvelles.

Q41. Analyser les résultats. Ce réseau de neurones est-il capable de bien prédire la valeur optimale de Δ ?

Q42. Pour améliorer le modèle est-il préférable d'augmenter le nombre d'itérations de l'entraînement ou d'augmenter le nombre de données de test ? Justifier votre réponse.

ANNEXE 1 - FORMULAIRE

Pour tous réels a et b :

$$\cos(a+b) = \cos a \cos b - \sin a \sin b$$
$$\cos(a-b) = \cos a \cos b + \sin a \sin b$$
$$\sin(a+b) = \sin a \cos b + \sin b \cos a$$
$$\sin(a-b) = \sin a \cos b - \sin b \cos a$$
$$\cos a \ \cos b = \frac{1}{2}\Big(\cos(a+b) + \cos(a-b)\Big)$$
$$\sin a \ \sin b = \frac{1}{2}\Big(\cos(a-b) - \cos(a+b)\Big)$$
$$\sin a \ \cos b = \frac{1}{2}\Big(\sin(a+b) + \sin(a-b)\Big)$$

Soit un signal $x(t)$, on notera <x(t)> la valeur moyenne temporelle de ce signal.

Si $x(t)$ est T-périodique alors :

$$< x(t) >= \frac{1}{T}\int_0^T x(t)\,dt.$$

Pour tous réels f_1, f_2, φ_1 et φ_2 :

$$< \cos(2\pi f_1 t + \varphi_1) >= 0 \qquad \text{si} \qquad f_1 \neq 0$$
$$< \sin(2\pi f_1 t + \varphi_1) >= 0 \qquad \text{si} \qquad f_1 \neq 0$$
$$< \cos^2(2\pi f_1 t + \varphi_1) >=< \sin^2(2\pi f_2 t + \varphi_2) >= \frac{1}{2}$$
$$< \cos(2\pi f_1 t + \varphi_1)\ \cos(2\pi f_2 t + \varphi_2) >= 0 \qquad \text{si} \qquad f_1 \neq f_2$$
$$< \sin(2\pi f_1 t + \varphi_1)\ \sin(2\pi f_2 t + \varphi_2) >= 0 \qquad \text{si} \qquad f_1 \neq f_2$$
$$< \cos(2\pi f_1 t + \varphi_1)\ \sin(2\pi f_2 t + \varphi_2) >= 0 \qquad \text{si} \qquad (f_1, f_2) \neq (0,0)$$

Sous des conditions de régularité qui seront supposées vérifiées dans ce sujet, un signal $x(t)$ T-périodique peut s'écrire :

$$x(t) =< x(t) > + \sum_{n=1}^{+\infty} \hat{X}_n \cos(2\pi\, n\, f\, t + \varphi_n).$$

Cette expression est appelée *développement en série de Fourier* (D.S.F.) du signal $x(t)$.

Les amplitudes $\hat{X}_n$ du développement en série de Fourier de $x(t)$ vérifient l'identité de Parseval :

$$< x(t)^2 > \quad = \quad < x(t) >^2 \quad + \quad \frac{1}{2}\sum_{n=1}^{+\infty} \hat{X}_n^2.$$

ANNEXE 2 - FONCTIONS PYTHON UTILES

La bibliothèque numpy est supposée installée de la manière classique - import numpy as np.

np.dot (A, B) : retourne le produit de deux matrices $[A]$ et $[B]$.

A.T : retourne la transposée d'une matrice $[A]$.

len(X) : retourne le nombre d'éléments de l'objet X. Ce dernier peut être une liste ou un tableau numpy.

np.abs(X) : retourne un tableau dont les éléments correspondent aux modules des éléments du tableau X.

np.arange(a, b, p) : retourne un tableau 1D dont les éléments sont répartis avec un pas p entre a et b, a est le premier élément du tableau, le dernier élément est strictement inférieur à b. Si l'argument p n'est pas renseigné, il est pris par défaut égal à 1.

np.fft.rfft(X) : retourne, sous forme d'un tableau 1D de nombres complexes, la transformée de Fourier discrète du signal échantillonné correspondant à X, un tableau 1D dont les éléments doivent être réels. Les éléments du tableau retourné correspondent aux fréquences $n\frac{F_e}{N}$ avec $n \in [\![0, \lfloor \frac{N}{2} \rfloor]\!]$, N le nombre d'éléments de X et F_e la fréquence d'échantillonnage. Enfin, le facteur multiplicatif $2/N$ doit être appliqué au tableau 1D retourné pour obtenir le spectre du signal continu correspondant au signal échantillonné contenu dans X.

np.linspace(a, b, N) : retourne un tableau 1D de N éléments répartis linéairement entre a et b, a est le premier élément du tableau, b le dernier.

plt.plot(X, Y) : permet de tracer le tableau 1D Y en fonction du tableau 1D X. Les deux tableaux doivent avoir le même nombre d'éléments. Des arguments optionnels permettent de choisir la couleur du tracé, l'épaisseur du trait, etc.

plt.xlim(x_min, x_max) : permet de limiter un tracé aux abscisses comprises entre x_min et x_max.

plt.xlabel('titre abscisses') : permet d'attribuer un titre à l'axe des abscisses. Des arguments optionnels permettent de choisir la taille de la police, la couleur du texte, etc.

plt.ylim(y_min, y_max) : permet de limiter un tracé aux ordonnées comprises entre y_min et y_max.

plt.ylabel('titre ordonnées') : permet d'attribuer un titre à l'axe des ordonnées. Des arguments optionnels permettent de choisir la taille de la police, la couleur du texte, etc.

FIN

IMPRIMERIE NATIONALE – 24 1105 – D'après documents fournis

Travailler les chapitres de ce sujet

Prepa Help suit le programme de PSI chapitre par chapitre : cours détaillé, méthodes, pièges, exercices et problèmes corrigés pas à pas, tirés des concours. L'essai gratuit donne accès aux chapitres offerts de ta filière, sans carte bancaire.

Chapitres liés à ce sujet

- Filtrage linéaire
- Électronique numérique
- Amplificateur linéaire intégré, stabilité et rétroaction
- Régimes transitoires
- Régime sinusoïdal forcé
- Mesures et incertitudes
- Calcul infinitésimal du physicien : dérivées et différentielles
- Bases de Python : expressions, types, instructions, fonctions

Ouvrir l'essai gratuit

Scanne le code ou clique sur le bouton.

prepahelp.com/#essai

Une question sur le concours ou sur ta filière : **écrire sur WhatsApp**

Les autres années et épreuves : prepahelp.com/annales/ccinp/psi