TP 05 : Synthèse de Fourier de signaux périodiques

Capacités exigibles :

- Énoncer qu'un signal périodique de fréquence f peut-être décomposé en une somme de signaux sinusoïdaux de fréquence multiple de f.
- Représenter et exploiter un spectre d'amplitude pour identifier le fondamental et les harmoniques.
- Calculer un rapport signal sur bruit.

Capacités expérimentales :

- Utiliser Python afin de tracer des signaux et d'obtenir leur spectre.
- Utilisation de la fonction réticule de Python.

Travail préparatoire (à faire à la maison) :

Afin de comprendre les notions abordées dans ce TP, visionner la vidéo du chapitre 03 :
« Savoir lire et exploiter un spectre en amplitude »

Puis, pour s'entrainer sur ces notions, compléter le QCM du chapitre 03 :
« De la représentation temporelle à la représentation fréquentielle et inversement »

I. Synthèse/simulation d'un signal « presque » triangulaire :

On souhaite simuler et synthétiser sur Python un signal nommé $s(t)$ variable, périodique, triangulaire, d'amplitude $U_m = 9{,}00\ V$, de valeur moyenne 3,00 V et de fréquence $f_1 = 100\ Hz$.

A. Étude théorique du signal :

Le signal ne contenant que les **six premiers harmoniques (jusqu'à $n = 6$)** du signal triangulaire $s(t)$ sera noté $u(t)$.

Lire l'encadré du *paragraphe II.A* du chapitre 03.

1. À l'aide du paragraphe lu, compléter le tableau situé en annexe 01. On explicitera sur sa copie, chaque calcul effectué.
2. Sur l'annexe 02, tracer au crayon à papier, la représentation fréquentielle théorique du signal $u(t)$, jusqu'à l'harmonique de rang $n = 6$.
3. Déterminer l'expression numérique de chacune des grandeurs suivantes : $u_1(t)$, $u_3(t)$, $u_5(t)$, correspondant respectivement au fondamental, l'harmonique de rang 3 et l'harmonique de rang 5.
4. En déduire l'expression littérale de l'harmonique de rang n, noté $u_n(t)$.
5. Déterminer l'expression numérique de $u(t)$, incluant l'ensemble des harmoniques jusqu'au rang $n = 6$.

APPEL 1 : Appeler le professeur afin qu'il valide et note votre travail.

B. Simulation sur PYTHON :

Dans PYZO, sous Python (version 3.6.7), ouvrir le fichier *TP05_representation_temporelle.py*

On souhaite tracer 4096 points du signal $u(t)$, sur une durée correspondant à deux périodes. Dans le code, la variable t_{max} correspond à la durée de deux périodes. Compléter les lignes 10 et 12 du fichier, sachant que l'unité de t_{max} est la seconde.

A l'aide votre travail dans le paragraphe précédent, sur la ligne 15 à 18, rédiger l'expression numérique de $\langle u \rangle$, $u_1(t)$, $u_3(t)$ *et* $u_5(t)$ sachant que la fonction cos se rédige $np.cos$

Avant de synthétiser le signal $u(t)$ ($somme\ de\ \langle u \rangle$, $u_1(t)$, $u_3(t)\ et\ u_5(t)$) , on souhaite tracer la représentation temporelle des harmoniques de rang 1, 3 puis 5.

Sur la ligne 20, indiquer que le signal $u(t)$ correspond uniquement à son harmonique de rang 1.
Lancer l'exécution du script.
Vérifier que la représentation temporelle obtenue est identique à celle donnée dans l'annexe 05 du TP pour $u(t)$ *: si ce n'est pas le cas, appeler l'enseignant.*

Modifier ensuite la ligne 20, afin de tracer l'harmonique de rang 3.
Lancer l'exécution du script.
Vérifier que la représentation temporelle obtenue est identique à celle donnée dans l'annexe 05 du TP pour $u(t)$ *: si ce n'est pas le cas, appeler l'enseignant.*

Modifier enfin la ligne 20, afin de tracer l'harmonique de rang 5.
Lancer l'exécution du script.
Vérifier que la représentation temporelle obtenue est identique à celle donnée dans l'annexe 05 du TP pour $u(t)$ *: si ce n'est pas le cas, appeler l'enseignant.*

APPEL 2 : Appeler le professeur afin qu'il valide et note votre travail.

Modifier la ligne 20 afin que le signal $u(t)$ soit la somme de sa composante continue et des six premiers harmoniques (jusqu'à $n = 6$) du signal triangulaire que l'on souhaite synthétiser.

Lancer l'exécution du script.

APPEL 3 : Appeler le professeur afin qu'il valide et note votre travail.

Mettre le graphe en plein écran, puis enregistrer-le au format PDF sur votre espace disque. Ouvrir le PDF et imprimer-le au format paysage.

6. Le signal $u(t)$ est-il variable, périodique, de motif triangulaire ?
7. A l'aide des réticules de PYZO, mesurer la période du signal $u(t)$ et en déduire la fréquence du signal $u(t)$.
8. La comparer à la fréquence attendue et conclure.
9. A l'aide des réticules de PYZO, mesurer l'amplitude expérimentale U_m du signal $u(t)$.
10. La comparer (sans calculer d'écart relatif) à l'amplitude attendue. Conclure.

C. Représentation fréquentielle du signal $u(t)$:

11. A l'aide du *paragraphe II.A* du chapitre 03, répondre à la question suivante : de combien d'harmoniques est composé un signal « parfaitement » triangulaire (on indiquera le rang maximal) ? Même question pour votre signal $u(t)$ précédent, tracé sur PYTHON.

Pour vérifier votre réponse à la question 11, ouvrir le fichier suivant :

TP05_representation_frequentielle_synthese_signal_triangulaire.py

Compléter les lignes 22 à 24 du fichier. La ligne 35 permet de simuler un signal « parfaitement » triangulaire à l'aide de la fonction *sawtooth* de la bibliothèque *scipy.signal.* Ce signal est nommé $s(t)$.

Compléter les lignes 39 à 43. Lancer l'exécution du script.

APPEL 4 : Appeler le professeur afin qu'il valide et note votre travail.

Mettre les deux graphes en plein écran, puis enregistrer-les au format PDF sur votre espace disque. Ouvrir les PDF et imprimer-les au format paysage.

12. A l'aide de vos impressions, répondre à la question suivante : de combien d'harmoniques est composé un signal « parfaitement » triangulaire (on indiquera le rang maximal) ? Même question pour votre signal $u(t)$ précédent, tracé sur PYTHON.

Vérifier que vos réponses 11 et 12 sont identiques : si ce n'est pas le cas, appelez le professeur.

13. A l'aide de l'outil réticule et de la fonction Zoom, compléter le tableau situé sur l'annexe 01 à l'aide de la représentation fréquentielle du signal $s(t)$. On ne complètera les cases que la lecture graphique permet de remplir.

Vérifier que les valeurs des amplitudes/valeurs et fréquences des deux tableaux de l'annexe 01 se correspondent : si ce n'est pas le cas, appelez le professeur.

14. En exploitant uniquement le spectre du signal de $s(t)$, quelle grandeur nous manque-t-il afin de pouvoir retrouver l'expression numérique des 6 premiers harmoniques (comme ceux de la question 4) ?

APPEL 5 : Appeler le professeur afin qu'il valide et note votre travail.

II. Synthèse d'un signal « parfaitement » triangulaire :

On souhaite synthétiser sur Python un signal variable, périodique, triangulaire, d'amplitude $U_m = 9{,}00\ V$, de valeur moyenne 3,00 V et de fréquence $f_1 = 100\ Hz$.

A. Simulation sur PYTHON pour obtenir la représentation temporelle :

Dans PYZO, sous Python (version 3.6.7), ouvrir le fichier suivant :

TP05_representation_temporelle_synthese_signal_triangulaire.py

On souhaite tracer 65536 points du signal $u(t)$, sur une durée correspondant à deux périodes.
Compléter les lignes 13 à 17 du fichier.

La ligne 26 permet de générer un signal « parfaitement » triangulaire à l'aide de la fonction *sawtooth* de la bibliothèque *scipy.signal.* Ce signal est nommé $s(t)$.

Compléter la ligne 33 : on appelle u_n, l'expression littérale de l'harmonique de rang n, déterminée à la question 4 de ce TP. La fonction puissance se rédige « ** » sur Python.

Python va calculer automatiquement les amplitudes, les fréquences des harmoniques de rang n . La fonction range (ligne 32) permet de lui indiquer quel harmonique calculer :

for n in range (a, b, c) :

a : rang du premier harmonique à calculer
b : rang du dernier harmonique à calculer

c : pas d'incrémentation

Exemple pour vous aider : vous souhaitez calculer les harmoniques de rang 1, 4, 7 et 10. Il faut alors rédiger ainsi :

for n in range (1, 11, 3) :

Compléter la ligne 32 afin que le signal $u(t)$ ne contiennent que l'harmonique de rang 1.

Compléter la ligne 37 afin que le signal d'erreur $e(t)$ soit la différence entre le signal parfaitement triangulaire et le signal ne contenant que certains harmoniques d'un signal triangulaire.

Lancer l'exécution du script.

APPEL 6 : Appeler le professeur afin qu'il valide et note votre travail.

Devant le professeur, modifier votre code pour que le signal $u(t)$ ne contiennent que les harmoniques jusqu'au rang 3, puis jusqu'au rang 5, et enfin jusqu'au rang 20. Faire un zoom sur un des points difficilement synthétisables.

Pour le rang 5, mettre le graphe en plein écran, puis enregistrer-le au format PDF sur votre espace disque. Ouvrir le PDF et imprimer-le au format paysage.

B. Signal d'erreur, rapport signal sur bruit :

Paramétrer à nouveau votre code PYTHON afin que le signal $u(t)$ ne contiennent que l'harmonique de rang 1. Lancer l'exécution du script.

15. Le signal $e(t)$ est–il périodique ? Le signal $e(t)$ est–il alternatif ?
16. Quels sont les points du signal triangulaire les plus difficile à synthétiser ? Justifier votre réponse en utilisant les extremums du signal d'erreur.
17. A l'aide vos connaissances, calculer la valeur efficace du signal $s(t)$, notée $U_{eff,signal}$.

Document 01 : qu'est-ce que le rapport signal sur bruit ?

Le rapport signal sur bruit (ou SNR_{dB}) indique la marge, en décibel (noté dB), qui sépare le niveau du signal utile de celui du bruit.
Le bruit peut-être est issu d'une erreur de quantification ou d'une erreur dans la synthèse du signal. Sa formule est alors :

$$SNR_{dB} = 20 \times \log \frac{U_{eff,signal}}{U_{eff,erreur}}$$

$U_{eff,signal}$: valeur efficace du signal « complet », en volt
$U_{eff,erreur}$: valeur efficace du signal d'erreur, en volt
SNR_{dB} : rapport signal sur bruit, en décibel (noté dB).

Si le signal d'erreur est nul, alors le rapport signal sur bruit est infini.
Si valeur efficace du signal d'erreur est égale à la valeur efficace du signal « complet » (ce qui sous-entend une forte erreur), alors le rapport signal sur bruit est nul.
Si valeur efficace du signal d'erreur est plus grande que la valeur efficace du signal « complet » (ce qui sous-entend une très forte erreur), alors le rapport signal sur bruit est négatif.

Dans le champ libre à côté des symboles >>>, Python permet de calculer rapidement la valeur efficace d'un signal, grâce à la fonction « $rms(\ )$ »

18. A l'aide de la fonction $rms(\ )$, déterminer et noter sur votre copie la valeur efficace du signal d'erreur, notée $U_{eff,erreur}$ (avec 3 chiffres significatifs)

On pourra vérifier à l'aide de la fonction $rms(\ )$, que la valeur obtenue à la question 16 est correcte.

19. En déduire, à l'aide du document 01, la valeur en décibel, du rapport signal sur bruit SNR_{dB}.

APPEL 7 : Appeler le professeur afin qu'il valide et note votre travail.

Paramétrer à nouveau votre code PYTHON afin que le signal $u(t)$ ne contiennent que les harmoniques jusqu'au rang 3. Lancer l'exécution du script.

20. A l'aide de la fonction $rms(\ )$, déterminer et noter sur votre copie la valeur efficace du signal d'erreur, notée $U_{eff,erreur}$. (avec 3 chiffres significatifs)

21. En déduire, à l'aide du document 01, la valeur en décibel, du rapport signal sur bruit SNR_{dB}.

Paramétrer à nouveau votre code PYTHON afin que le signal $u(t)$ ne contiennent que les harmoniques jusqu'au rang 5. Lancer l'exécution du script.

22. A l'aide de la fonction $rms(\ )$, déterminer et noter sur votre copie la valeur efficace du signal d'erreur, notée $U_{eff,erreur}$. (avec 3 chiffres significatifs)

23. En déduire, à l'aide du document 01, la valeur en décibel, du rapport signal sur bruit SNR_{dB}.

Paramétrer à nouveau votre code PYTHON afin que le signal $u(t)$ ne contiennent que les harmoniques jusqu'au rang 20. Lancer l'exécution du script.

24. A l'aide de la fonction $rms(\ )$, déterminer et noter sur votre copie la valeur efficace du signal d'erreur, notée $U_{eff,erreur}$. (avec 3 chiffres significatifs)

25. En déduire, à l'aide du document 01, la valeur en décibel, du rapport signal sur bruit SNR_{dB}.

26. Comment évolue le rapport signal sur bruit lorsqu'on prend en compte de plus en plus d'harmoniques du signal ?

27. Quel devrait-être la valeur du rang maximum pour que le signal $u(t)$ soit parfaitement « triangulaire » ? Que vaudrait alors le rapport signal sur bruit ?

APPEL 8 : Appeler le professeur afin qu'il valide et note votre travail.

III. Synthèse et simulation d'un signal carré :

On souhaite simuler et synthétiser sur Python un signal noté $s(t)$ variable, périodique, carré, d'amplitude $U_m = 9{,}00\ V$, de valeur moyenne $3{,}00\ V$ de fréquence $f_1 = 200\ Hz$.

A. Étude théorique du signal :

Le signal ne contenant que les **six premiers harmoniques (jusqu'à $n = 6$)** du signal carré $s(t)$ sera noté $u(t)$.

Lire l'encadré du *paragraphe II.B* du chapitre 03.

28. À l'aide du paragraphe lu, compléter le tableau situé en annexe 03. On explicitera sur sa copie, chaque calcul effectué.
29. Sur l'annexe 04, tracer au crayon à papier, le spectre théorique du signal $u(t)$, jusqu'à l'harmonique de rang $n = 6$.
30. Déterminer l'expression numérique de chacune des grandeurs suivantes : $u_1(t)$, $u_3(t)$, $u_5(t)$, correspondant respectivement au fondamental, l'harmonique de rang 3 et l'harmonique de rang 5.
31. En déduire l'expression littérale de l'harmonique de rang n, noté $u_n(t)$.
32. Déterminer l'expression numérique de $u(t)$, incluant l'ensemble des harmoniques jusqu'au rang $n = 6$.

B. Représentation fréquentielle du signal $u(t)$:

33. A l'aide du *paragraphe II.B* du chapitre 03, répondre à la question suivante : de combien d'harmoniques est composé un signal « parfaitement » carré (on indiquera le rang maximal) ? Même question pour votre signal $u(t)$ précédent, tracé sur PYTHON.

Pour vérifier votre réponse à la question précédente, ouvrir le fichier suivant :

TP05_representation_frequentielle_synthese_signal_carre.py

Compléter les lignes comportant des points d'interrogations. Lancer l'exécution du script.

APPEL 9 : Appeler le professeur afin qu'il valide votre travail.

34. A l'aide de vos impressions, répondre à la question suivante : de combien d'harmoniques est composé un signal « parfaitement » carré (on indiquera le rang maximal) ? Même question pour votre signal $u(t)$ précédent, tracé sur PYTHON.

35. A l'aide de l'outil réticule et de la fonction Zoom, compléter le tableau situé sur l'annexe 03 à l'aide de la représentation fréquentielle du signal $s(t)$. On ne complètera les cases que la lecture graphique permet de remplir.

Vérifier que les valeurs des amplitudes/valeurs et fréquences des deux tableaux de l'annexe 03 se correspondent : si ce n'est pas le cas, appelez le professeur.

36. A amplitude égale et valeur moyenne égale, quel signal (triangulaire ou carré) possède des harmoniques dont les amplitudes sont plus grandes ?

APPEL 10 : Appeler le professeur afin qu'il valide votre travail.

IV. Synthèse d'un signal « parfaitement » carré :

On souhaite synthétiser sur Python un signal variable, périodique, carré, d'amplitude $U_m = 9{,}00\,V$, de valeur moyenne 3,00 V et de fréquence $f_1 = 200\,Hz$.

A. Simulation sur PYTHON pour obtenir la représentation temporelle :

Dans PYZO, sous Python (version 3.6.7), ouvrir le fichier suivant :

TP05_representation_temporelle_synthese_signal_carre.py

Compléter les lignes comportant des points d'interrogations. Le premier signal $u(t)$ ne contient que les jusqu'au rang 10. Lancer l'exécution du script.

APPEL 11 : Appeler le professeur afin qu'il valide votre travail.

Devant le professeur, modifier votre code pour que le signal $u(t)$ ne contiennent que les harmoniques jusqu'au rang 40, puis jusqu'au rang 2000, et enfin jusqu'au rang 10000.

B. Signal d'erreur, rapport signal sur bruit :

Paramétrer à nouveau votre code PYTHON afin que le signal $u(t)$ ne contiennent que l'harmonique de rang 10. Lancer l'exécution du script.

37. Quels sont les points du signal carré les plus difficile à synthétiser ? Justifier votre réponse en utilisant les extremums du signal d'erreur.
38. A l'aide vos connaissances, calculer la valeur efficace du signal $s(t)$, notée $U_{eff,signal}$.

Dans le champ libre à côté des symboles >>>, Python permet de calculer rapidement la valeur efficace d'un signal, grâce à la fonction « $rms(\ )$ »

39. A l'aide de la fonction $rms(\ )$, déterminer et noter sur votre copie la valeur efficace du signal d'erreur, notée $U_{eff,erreur}$.

40. En déduire, à l'aide du document 01, la valeur en décibel, du rapport signal sur bruit SNR_{dB}.

Paramétrer à nouveau votre code PYTHON afin que le signal $u(t)$ ne contiennent que les harmoniques jusqu'au rang 40. Lancer l'exécution du script.

41. A l'aide de la fonction $rms(\ )$, déterminer et noter sur votre copie la valeur efficace du signal d'erreur, notée $U_{eff,erreur}$.

42. En déduire, à l'aide du document 01, calculer la valeur en décibel, du rapport signal sur bruit SNR_{dB}.

Paramétrer à nouveau votre code PYTHON afin que le signal $u(t)$ ne contiennent que les harmoniques jusqu'au rang 2000. Lancer l'exécution du script.

43. A l'aide de la fonction $rms(\ )$, déterminer et noter sur votre copie la valeur efficace du signal d'erreur, notée $U_{eff,erreur}$.

44. En déduire, à l'aide du document 01, calculer la valeur en décibel, du rapport signal sur bruit SNR_{dB}.

Paramétrer à nouveau votre code PYTHON afin que le signal $u(t)$ ne contiennent que les harmoniques jusqu'au rang $10\ 000$. Lancer l'exécution du script.

45. A l'aide de la fonction $rms(\ )$, déterminer et noter sur votre copie la valeur efficace du signal d'erreur, notée $U_{eff,erreur}$.

46. En déduire, à l'aide du document 01, calculer la valeur en décibel, du rapport signal sur bruit SNR_{dB}.

47. Comment évolue le rapport signal sur bruit lorsqu'on prend en compte de plus en plus d'harmoniques du signal ?

48. Quel devrait-être la valeur du rang maximum pour que le signal $u(t)$ soit parfaitement « carré » ? Que vaudrait alors le rapport signal sur bruit ?

APPEL 12 : Appeler le professeur afin qu'il valide votre travail.