

Électronique numérique

Le développement de l'informatique dont le fonctionnement repose sur le mode binaire 0 ou 1 a progressivement imposé de travailler avec des signaux numérisés par opposition aux signaux analogiques. Un signal analogique est un signal qui évolue de façon continue au cours du temps même si ses variations peuvent extrêmement rapides. Par opposition un signal numérique est un signal décrit par un ensemble de 0 et 1 dont la base de représentation est l'octet.

1 Généralités

À un ensemble de 0 et de 1 est associé une valeur fixée du signal, lorsque le signal évolue la composition de 0 et 1 évolue. Le signal numérique passe à une autre valeur. Imaginons que le signal soit représenté par 00111010. À cette combinaison correspond une valeur donnée parmi les valeurs disponibles. Si le signal évolue et se trouve représenté par 00111011, cela signifie qu'il aura une autre valeur différente de la première. Il y a discontinuité. Un signal numérique est donc constitué par une succession de valeurs de constantes. Le schéma de la figure 1 illustre la différence entre un signal analogique et un signal numérique.

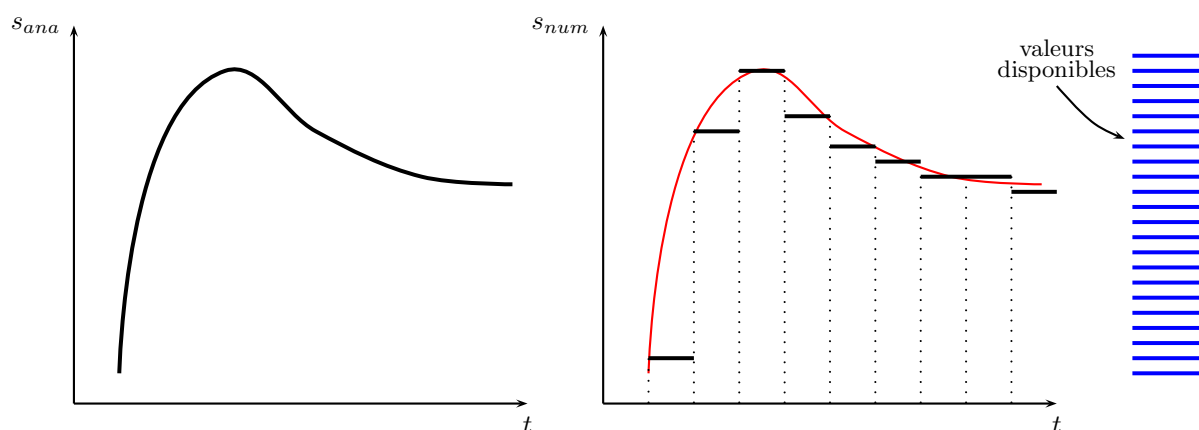


FIGURE 1 – Signal analogique et signal numérique

En général, la Physique n'apprécie guère les discontinuités. Les mesures effectuées sur un système sont analogiques par nature. Il est, par conséquent, évident au vu des graphiques de la figure 1 que le signal numérique associé au signal analogique est loin d'être idéal comme représentation ! La première question qui vient à l'esprit est le rythme auquel on va répéter les évaluations du signal analogique. Afin de pouvoir s'y retrouver, il est indispensable que celui-ci soit régulier : c'est la *période d'échantillonnage*. On la note en général T_e et on y associe la *fréquence d'échantillonnage* $F_e = 1/T_e$. Bien sûr, on pourrait trouver opportun de saisir plus de valeurs du signal pendant ses variations rapides que pendant ses variations lentes comme il est arrivé que vous le fassiez lors des dosages acido-basiques au niveau de l'équivalence. Mais pour l'utilisateur, ce serait vite ingérable, il faudrait en permanence l'informer du changement de rythme de l'échantillonnage. En fonction de la technologie disponible et de son coût, on recherchera la fréquence d'échantillonnage la plus élevée pour obtenir le maximum de précision dans la représentation du signal analogique.

Afin de bien représenter le signal analogique, il est aussi nécessaire de disposer d'un éventail important de valeurs disponibles. Travailler en 8 bits correspond à une représentation qui propose $2^8 = 256$ niveaux différents de tension dans un intervalle donné. Là encore, les sauts de valeurs d'un niveau au suivant sont constants sur tout l'intervalle de tension accepté. La précision de la représentation sera encore une fois augmentée si l'on augmente le nombre de bits. En passant à 16 bits, on aura $2^{16} = 65\,536$ niveaux. Encore une fois, tout sera une question de technologie disponible et de coût.

Quoi qu'il en soit, la représentation numérique d'un signal analogique ne sera jamais équivalente au signal analogique. Certains mélomanes regrettent le temps des disques vinyles où l'ensemble de la chaîne de traitement était analogique. Certes, la numérisation affecte le signal mais les progrès effectués depuis le début du numérique vers 1980 et surtout les traitements très évolués que l'on peut effectuer dans le domaine numérique doivent nous amener à ne pas voir que des inconvénients à cette technique de représentation d'un signal. De plus la transmission d'un signal numérique est beaucoup plus fiable que celle d'un signal analogique. Pour terminer, nous dirons que l'utilisation quasi-permanente d'ordinateurs nous l'impose.

L'objet de notre étude est de mettre en évidence les conséquences de la numérisation d'un signal, en particulier sur son spectre de fréquence. Nous étudierons aussi certains dispositifs de numérisation.

2 Échantillonnage

2.1 Principe et exemples d'échantillonnage

Le principe de l'échantillonnage consiste à multiplier le signal $e_0(t)$ par un signal ayant pour modèle un peigne de DIRAC $e_e(t)$ pour obtenir le signal numérique $e_{nu}(t) = s(t)$:

$$s(t) = e_{nu}(t) = e_0(t) \times e_e(t)$$

On peut voir sur les graphiques de la figure 2, deux exemples d'échantillonnage à des fréquences d'échantillonnage différentes. On peut se placer dans la situation de restitution du signal de départ après numérisation pour avoir une idée des conséquences de la numérisation. Ici, on a juste interpolé les points du signal numérique.

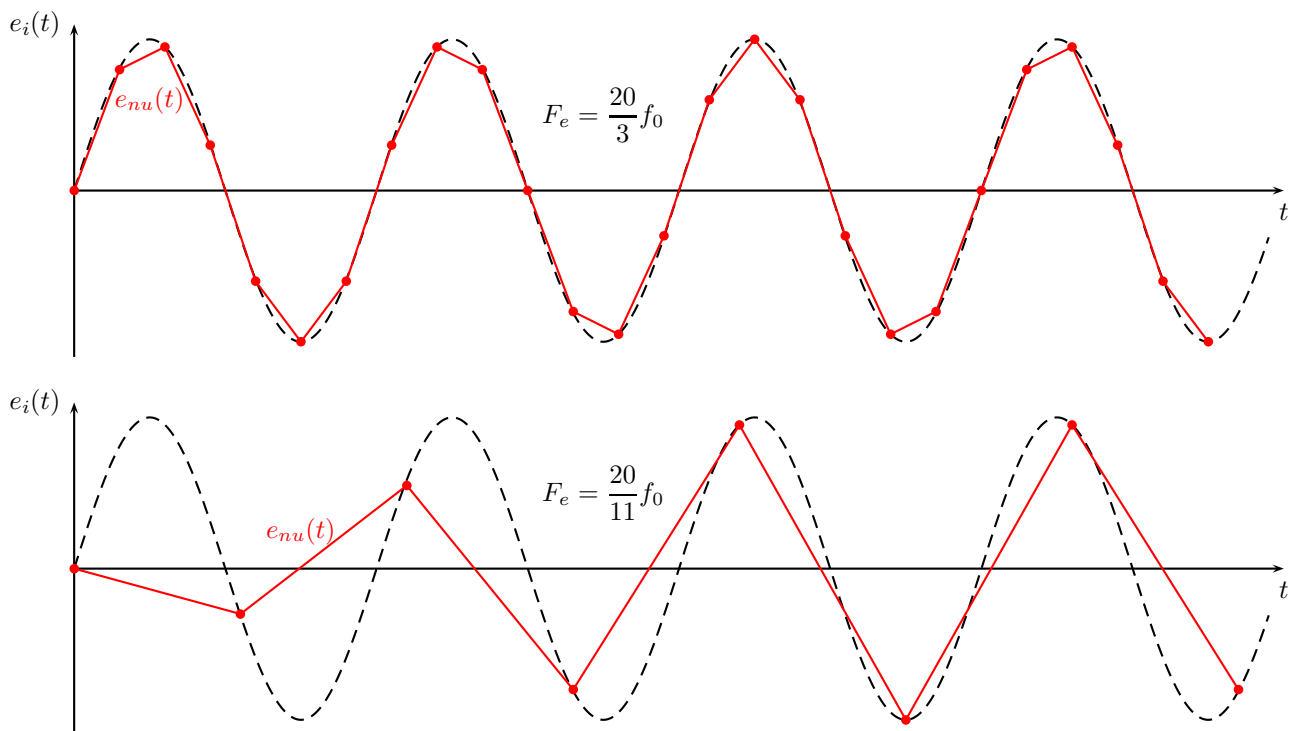


FIGURE 2 – Exemples de numérisation

On constate de façon visuelle évidente que la fréquence d'échantillonnage va être décisive pour que la numérisation soit la plus fidèle possible par rapport au signal analogique de départ. La solution est de la prendre la plus élevée possible mais cela n'a aucun intérêt si derrière les valeurs sélectionnées viennent se ranger dans un nombre de niveaux limité par le nombre de bits n qui propose 2^n niveaux. Bien sûr, cela dépend aussi de l'amplitude du signal car plus cette amplitude est élevée et plus on aura peut-être besoin de précision et donc d'un nombre de bits élevé. Travailler à une fréquence d'échantillonnage élevée, avec un nombre de bits élevé possède un coût en terme de composants électroniques rapides et de capacité mémoire. Il faudra trouver le bon compromis. Dans la suite, nous allons voir les conséquences qui peuvent s'avérer néfastes de la numérisation d'un signal. Il ne faut toutefois pas toutes les possibilités de traitement du signal offertes sur un signal numérisé.

2.2 Multiplication de deux signaux

On considère deux signaux sinusoïdaux, un signal de fréquence f_0 de moyenne non nulle et un second de fréquence F_e . Ils sont représentés par les fonctions réelles suivantes :

$$e_0(t) = (E_0 + E_1 \cos(2\pi f_0 t + \varphi_0)) \quad e_e(t) = E_e \cos(2\pi F_e t)$$

Ces deux signaux sont envoyés sur un circuit multiplieur dont la réponse est :

$$s(t) = \frac{e_1(t) \times e_e(t)}{V_0}$$

où V_0 est une tension référence fonction de la nature du circuit multiplieur. Comme nous l'avons déjà vu, le signal $s(t)$ présente un spectre différent de celui des signaux d'entrée :

Sur le schéma de la figure 3, on peut voir les différents signaux en mode temporel sans se préoccuper des valeurs relatives de leur amplitude.

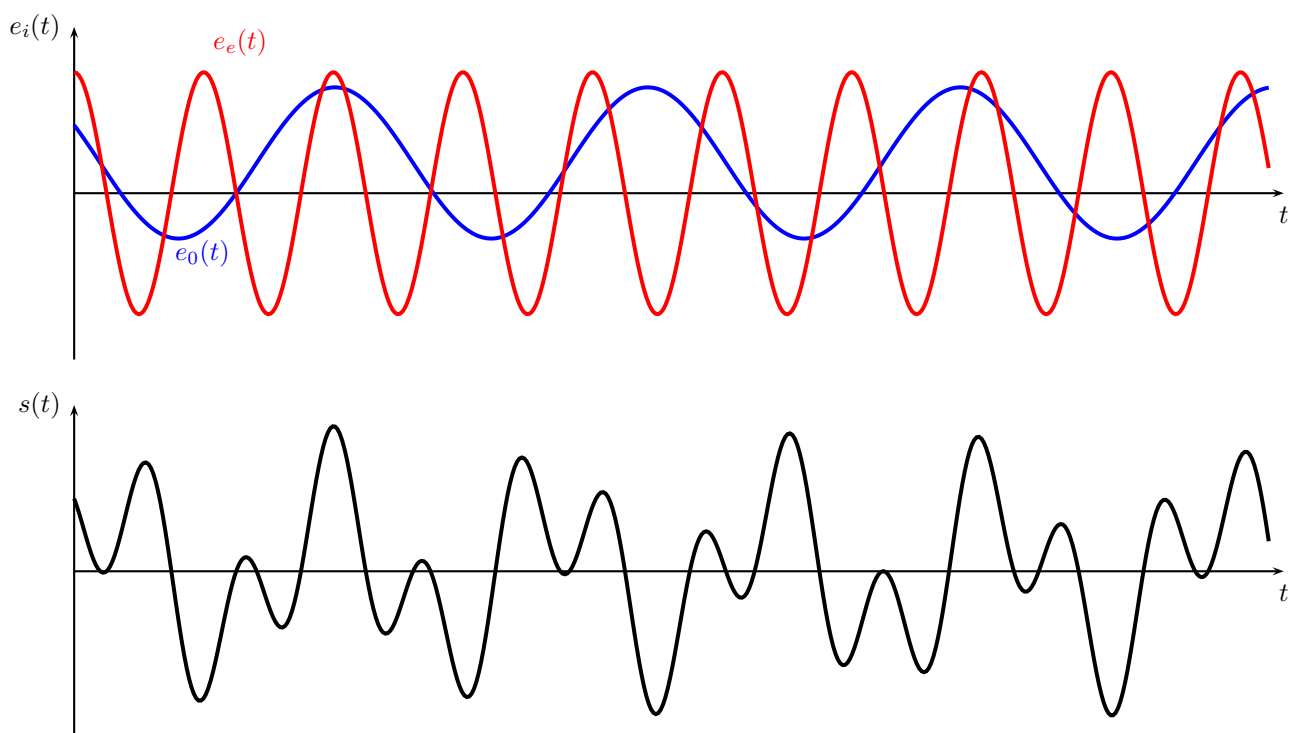


FIGURE 3 – Multiplication de deux signaux

2.3 Traitement par le peigne de Dirac

2.3.1 Mode temporel

On considère maintenant le peigne de DIRAC comme signal multiplicateur $e_e(t)$. Il agit sur un signal sinusoïdal $e_1(t)$ qui n'est pas nécessairement identique au précédent. Pour faciliter la perception des choses, le signal $e_1(t)$ présenté dans ce paragraphe est de fréquence plus faible que celui présenté dans le paragraphe précédent. On considère que ces deux signaux sont mis sur les entrées d'un circuit multiplieur comme celui qui précède. Les représentations temporelles sont assurées sur le schéma de la figure 4.

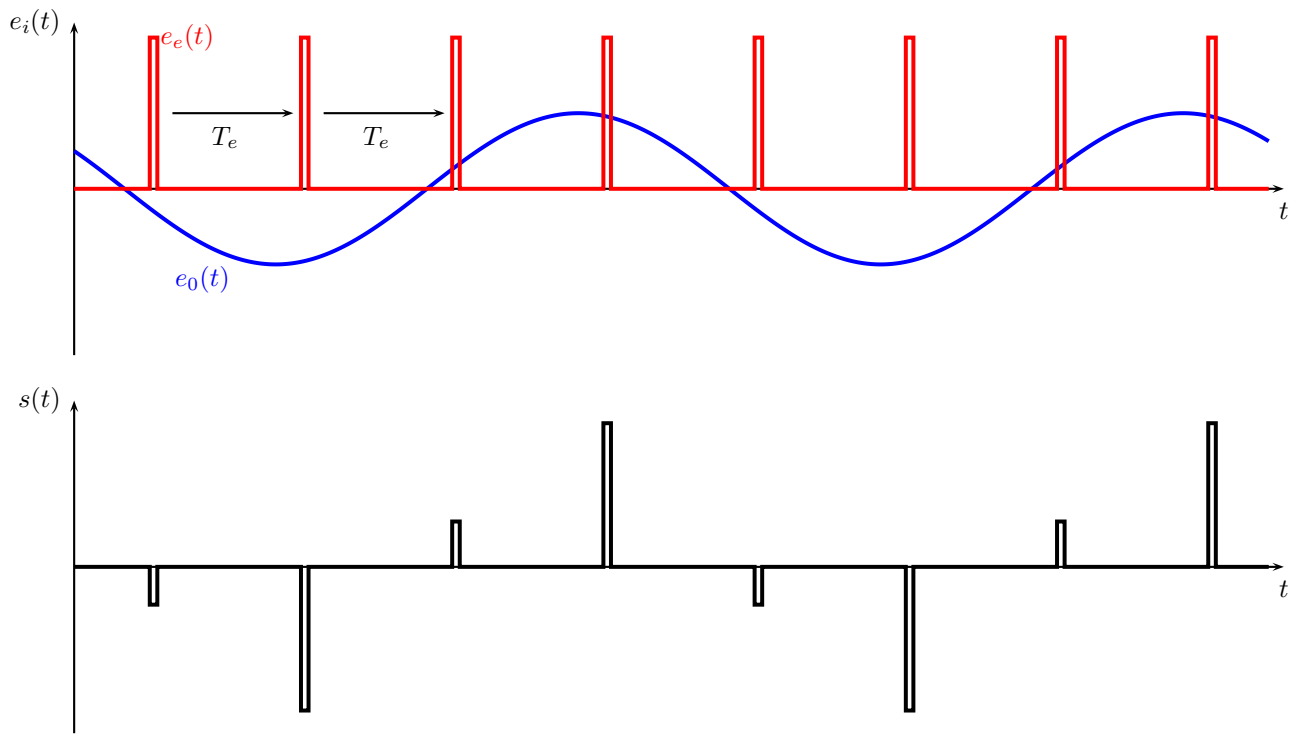


FIGURE 4 – Peigne de DIRAC et échantillonnage

2.3.2 Mode fréquentiel

2.3.3 Échantillonnage d'un signal triangulaire

Contrairement au cas précédent, le signal triangulaire est polychromatique. Nous avons vu que si sa fréquence était f_0 , il comportait les harmoniques $(2n + 1)f_0$ avec $n \in \mathbb{N}$. De plus, nous avons constaté qu'un très petit nombre d'harmoniques suffisaient pour représenter très correctement un signal triangulaire. Si par exemple, on considère le fondamental f_0 et les harmoniques $3f_0$, $5f_0$ alors le spectre du signal échantillonné se complique sérieusement :

2.3.4 Échantillonnage d'un signal quelconque

On considère désormais un signal quelconque non nécessairement périodique. Il est plus intéressant pour notre propos de le considérer dans le domaine fréquentiel plutôt que dans le domaine temporel, voir le schéma de la figure 5. Le spectre de ce signal est certes un peu caricatural mais, il permettra de bien comprendre une des problématiques fondamentales de l'échantillonnage. Ce signal est échantillonné comme les précédents par un signal de fréquence F_e .

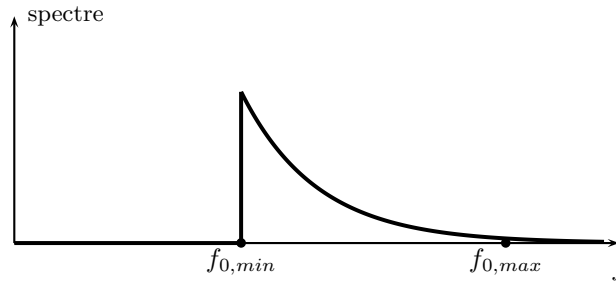


FIGURE 5 – Exemple de spectre d'un signal quelconque

2.3.5 Échantillonnage et blocage

En pratique, les dispositifs qui échantillonnent les signaux font de l'échantillonnage-blocage. Cela signifie que la valeur du signal saisie à une date $t = nT_e$ est maintenue jusqu'à la date $t' = (n+1)T_e$. Ceci est possible, par exemple, en utilisant un condensateur chargé à la date $t = nT_e$ et qui ne décharge pas (ou presque pas...) jusqu'à la date $t' = (n+1)T_e$ où il se verra imposé une tension correspondant à la valeur du signal à échantillonner à la date t' . Le signal échantillonné n'est pas en général une suite de petites impulsions séparées par la période T_e mais plutôt une succession de paliers de durée T_e , la valeur du palier correspondant, bien évidemment à la valeur de la tension à traiter à la date du début du palier. La photographie de la figure 6 illustre le phénomène d'échantillonnage-blocage appliqué à une tension analogique sinusoïdale. Le peigne de DIRAC agissant sur cette tension est aussi visible sur la photographie.

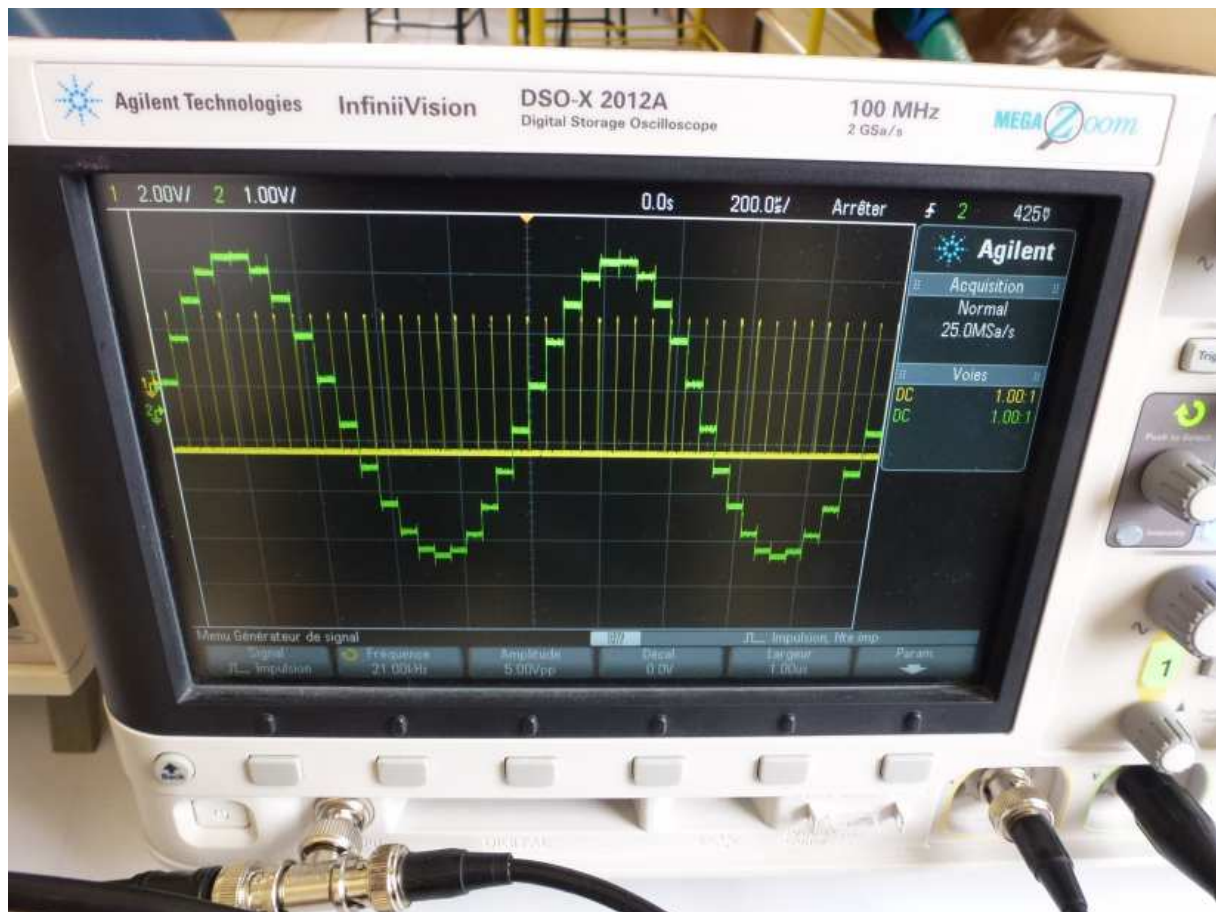


FIGURE 6 – Échantillonnage-blocage

3 Condition de Shannon

3.1 Nécessité

Nous venons de voir que l'échantillonnage affectait le signal analogique que l'on vient de numériser. Comme il est plus difficile d'en percevoir les effets en mode temporel, on privilégie la description des signaux par leur spectre. Le spectre du signal échantillonné est très différent de celui du signal analogique d'origine. La numérisation facilite le traitement et la transmission du signal mais à l'arrivée, nous devons être capable de retrouver le signal d'origine si possible tel qu'il était ou alors le plus voisin possible de ce qu'il était. Ceci n'est pas forcément simple ou tout simplement impossible comme nous allons le comprendre rapidement.

Revenons à la situation du signal sinusoïdal de fréquence f_0 échantillonné avec un peigne de DIRAC de fréquence F_e . En fonction de la valeur de la fréquence F_e par rapport à f_0 , le spectre obtenu sera totalement différent, voir les graphiques de la figure 7 sur lesquels on ne se préoccupera pas des amplitudes des différentes fréquences représentées.

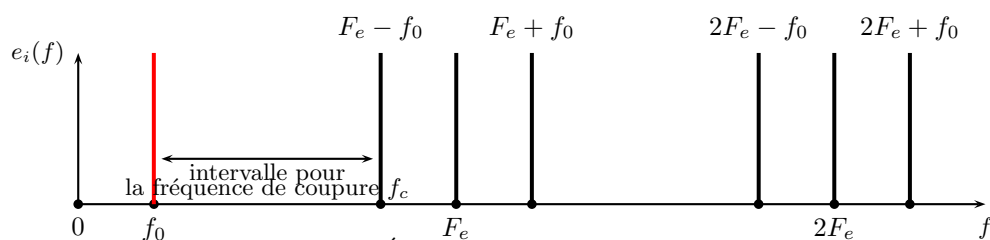


FIGURE 7 – Échantillonnage d'un signal sinusoïdal

Pour récupérer le signal analogique de départ de fréquence f_0 , on peut envisager d'utiliser un filtre passe-bas relativement atténuateur en dehors de sa bande-passante. Les filtres numériques peuvent aller jusqu'à un ordre 8 aisément, c'est-à-dire jusqu'à une atténuation de 160 dB par décade ! La solution est donc toute trouvée ci-dessus. Un bon filtre numérique passe-bas de fréquence de coupure $f_c \in]f_0, F_e - f_0[$ permet d'atteindre l'objectif.

Comme nous allons pouvoir le constater plus loin, la situation n'est pas aussi simple. Dans l'exemple de la figure 8, la fréquence d'échantillonnage F_e est beaucoup plus proche de la fréquence f_0 du signal sinusoïdal que cela n'était le cas précédemment. Le spectre est nettement différent dans l'ordre des fréquences qui y sont présentes, voir le graphique de la figure 8.

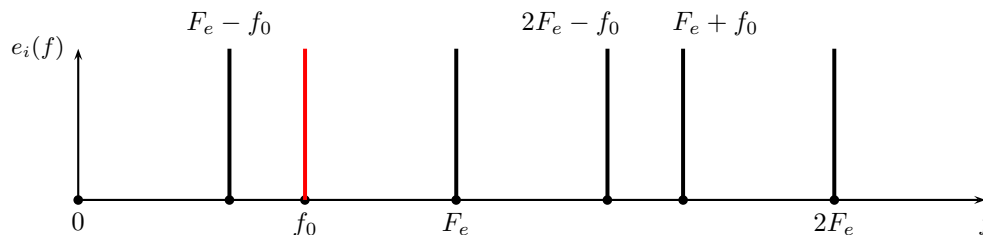


FIGURE 8 – Sous-échantillonnage d'un signal sinusoïdal

Dans le cas de la figure 8, il n'est plus possible d'utiliser un filtre passe-bas pour récupérer le signal de fréquence f_0 , on obtiendrait mélangé avec ce dernier un signal de fréquence $F_e - f_0$. Rien ne nous empêche d'envisager d'utiliser un filtre passe-bande très sélectif pour y arriver. Dans la pratique, on évite de se retrouver dans une telle situation avec la composante de fréquence $F_e - f_0$ située en-dessous de celle de fréquence f_0 . En effet, pour un signal plus quelconque que celui que nous venons d'envisager, généralement polychromatique à spectre continu, nous rencontrerions un très sérieux problème. Comme cela va être confirmé par l'exemple qui suit, il faut réaliser un échantillonnage qui assure que :

$$f_0 < F_e - f_0 \quad \text{ou ce qui revient au même} \quad F_e > 2f_0$$

Dans l'exemple qui suit, un signal présentant un spectre continu est sous-échantillonné, voir la figure 9.

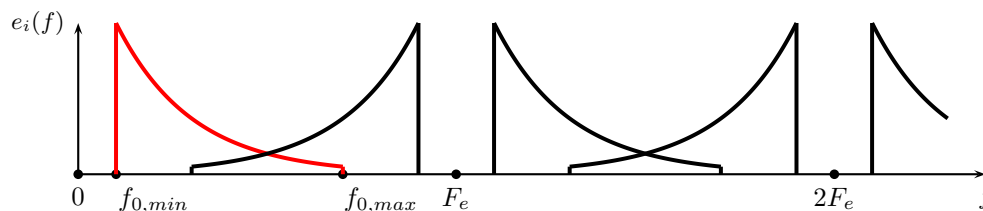


FIGURE 9 – Sous-échantillonnage d'un signal de spectre continu

3.2 Énoncé

Condition de Shannon :

Sur le schéma de la figure 10, le même signal que celui évoqué avant est échantillonné correctement. La fréquence utilisée est $F_e > 2f_{0,max}$.

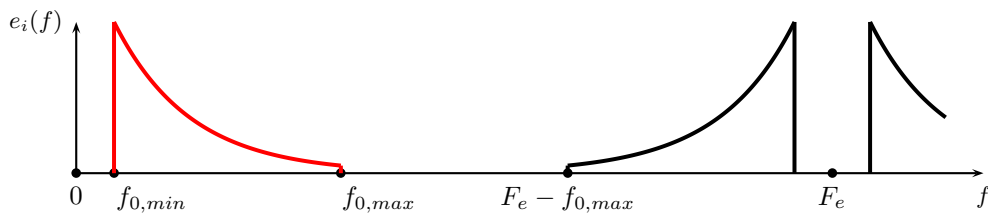


FIGURE 10 – Échantillonnage correct d'un signal de spectre continu

Dans le cas précédent, on peut avoir recours à un filtre passe-bas de fréquence de coupure comprise entre $f_{0,max}$ et $F_e - f_{0,max}$ pour récupérer le signal d'origine sans que de l'information ait été perdue. Échantillonner à la limite de la condition de SHANNON revient à prélever uniquement 2 valeurs du signal de fréquence maximale $f_{0,max}$ pendant une durée correspondant à la période T_e d'échantillonnage.

Nous terminerons en donnant deux exemples de fréquences d'échantillonnage. Pour le signal téléphonique, le fréquence d'échantillonnage est $F_e = 8$ kHz, le signal transmis comporte des fréquences nécessairement inférieures à 4 kHz. En fait, le signal est limité à 3,4 kHz ce qui permet de respecter la condition de SHANNON. Pour l'enregistrement des CD de musique, on utilise très fréquemment $F_e = 44,1$ kHz. Cela permet de respecter la condition d'échantillonnage correct puisque les fréquences maximales audibles sont de 20 kHz.

4 Numérisation

4.1 Quantification

Une fois l'opération d'échantillonnage réalisée, il faut transformer la valeur du signal en une combinaison de 0 et de 1. C'est l'étape de numérisation. Il faut tout d'abord établir une échelle de valeurs de référence. En général, on travaille dans le domaine de l'électronique avec un intervalle de tension de $\Delta U = 5$ V même si n'importe quel intervalle de tension raisonnable peut être envisagé.

Nous allons prendre un exemple de circuit électronique très simple qui réalise l'opération de conversion du signal analogique en signal numérique. Ce type de circuit porte le nom de CAN pour Convertisseur Analogique Numérique. Il existe de nombreuses formes de CAN qui sont plus ou moins précis, plus ou moins rapides. Nous ne rentrerons pas dans l'étude des différents types de circuits, seul le principe nous importe ici. Le schéma du circuit est réalisé à la figure 11.

Le circuit présenté comporte des résistances électriques R une source de tension constante $V_{ref} = +5$ V ainsi que 4 amplificateurs opérationnels utilisés en comparateur. Un seul montage comparateur a été représenté tel qu'il se présente. Les 4 amplificateurs opérationnels ne prélèvent aucun courant et délivrent en sortie une tension qui sera $u_{si} = \pm V_{sat}$ en fonction du signe de leur tension différentielle d'entrée $\varepsilon = V_+ - V_-$ avec la loi $u_{si} = \frac{\varepsilon}{|\varepsilon|} V_{sat}$. Les amplificateurs opérationnels sont alimentés par une source de tension symétrique par rapport à la masse qui n'est pas représentée sur le schéma. Sur le schéma, les connexions électriques sont matérialisées par un point. Lorsque deux fils se croisent sans point il n'y a pas de nœud à ce niveau du montage, les fils ne sont pas connectés. On suppose que le signal analogique est compris dans l'intervalle $[0 \text{ V}; V_{ref} = 5 \text{ V}]$, cela nécessite en général un circuit électronique classique de traitement que nous n'évoquerons pas.

Pour comprendre le fonctionnement de ce CAN, il faut tout d'abord voir que la succession de résistances R en série - puisqu'aucun prélèvement de courant ne s'effectue sur les AO - va imposer sur l'entrée inverseuse \ominus de chaque AO une tension qui sera $\frac{R}{5R} V_{ref} = 1$ V pour l'AO situé tout en bas du circuit, $\frac{2R}{5R} V_{ref} = 2$ V pour le suivant et ainsi de suite. La tension analogique u_{ana} échantillonnée est envoyée simultanément sur chacune des entrées non inverseuses \oplus des AO qui agissent en comparateur. Ainsi on compare simultanément la tension u_{ana} à 1 V sur le comparateur du bas, à 2 V sur le second, à 3 V sur le troisième et enfin à 4 V sur celui situé en haut.

Prenons un exemple : la tension d'entrée est $u_{ana} = 2,4$ V. Déterminons l'état des sorties des comparateurs. On convient de retenir le codage suivant pour les sorties des comparateurs : 0 lorsque $u_{si} = -V_{sat}$ et 1 lorsque $u_{si} = V_{sat}$. Pour convertir en notation binaire, il faut déterminer les valeurs 0 ou 1 des coefficients (α, β, γ) tels que la tension de l'échelle retenue puisse s'écrire $\alpha 2^2 + \beta 2^1 + \gamma 2^0$.

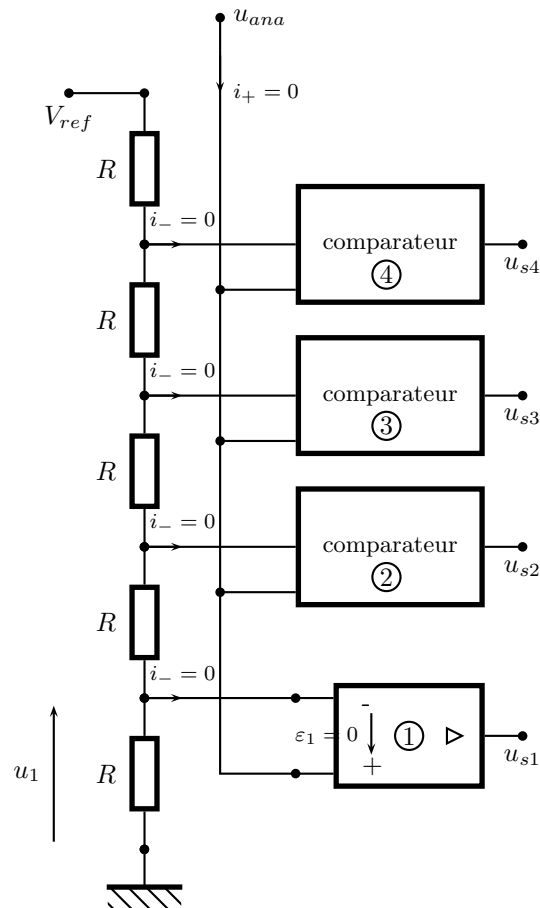


FIGURE 11 – Montage CAN de type flash

Le tableau qui résume le fonctionnement du circuit est le suivant :

u_a	codage des sorties	valeur référence de u_a	codage binaire
$0 < u_a < 1 \text{ V}$	0000	0 V	000
$1 \text{ V} < u_a < 2 \text{ V}$	0001	1 V	001
$2 \text{ V} < u_a < 3 \text{ V}$	0011	2 V	010
$3 \text{ V} < u_a < 4 \text{ V}$	0111	3 V	011
$4 \text{ V} < u_a < 5 \text{ V}$	1111	4 V	100

Considérons maintenant une succession de tensions échantillonnées données par le graphique de la figure 12. Nous allons déterminer la succession des codes binaires associés (voir la figure 12) et évaluer les erreurs commises pour la valeur du signal, (voir la figure 13).

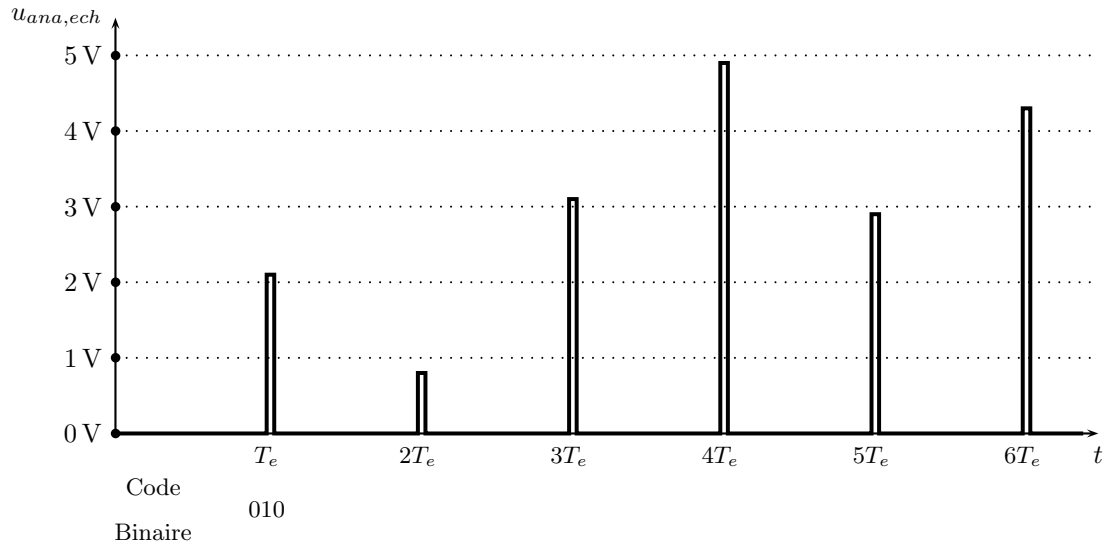
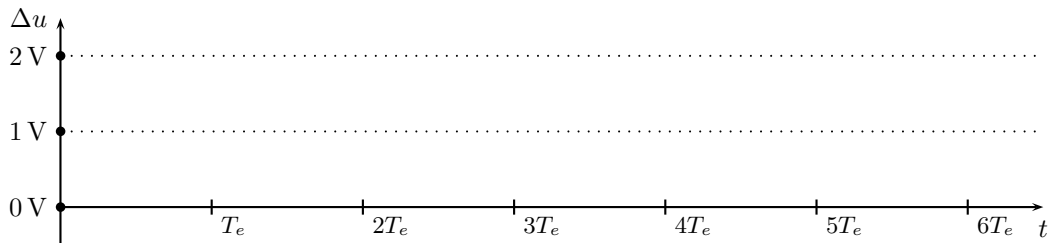
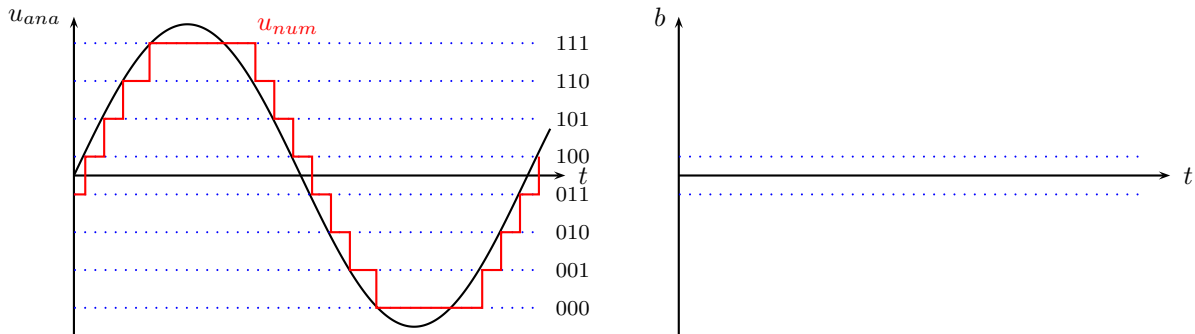


FIGURE 12 – Codage d’une succession de valeurs échantillonnées du signal analogique

FIGURE 13 – Erreurs sur les valeurs échantillonnées du signal analogique $\Delta u = u_{ana} - u_{num}$ FIGURE 14 – Bruit de quantification $b(t) = u_{num}(t) - u_{ana}(t)$ pour $2^3 = 8$ pas (3 bits) avec basculement en fin de pas de quantification

L'étude des figures 14 et 15 met en évidence le problème de l'imprécision associée à la conversion en binaire du signal analogique. Ce phénomène est inhérent à la quantification. Comme lorsque l'on grimpe à une échelle, il n'est possible à partir d'un barreau que de mettre le pied sur le suivant ou le précédent mais pas entre les deux. La précision sera aussi fonction du nombre de barreaux que comporte l'échelle pour l'intervalle de tension $\Delta u_{num} = [0, V_{ref}]$ - l'intervalle Δu_{num} s'appelle la dynamique du convertisseur -. Plus le nombre de barreaux sera élevé, meilleure sera la précision. Un codage sur n bits réserve 2^n possibilités. Pour une dynamique donnée Δu_{num} , le pas de quantification est donc :

$$q = \frac{\Delta u_{num}}{2^n - 1} \simeq \frac{\Delta u_{num}}{2^n}$$

Il est alors logique de préférer un convertisseur $n = 16$ bits à un convertisseur $n = 8$ bits. Mais il ne faut pas oublier que la moindre fluctuation peut entraîner alors des variations importantes en terme de bit. Le convertisseur le plus précis ne sera utile que s'il est très peu sensible aux fluctuations extérieures.

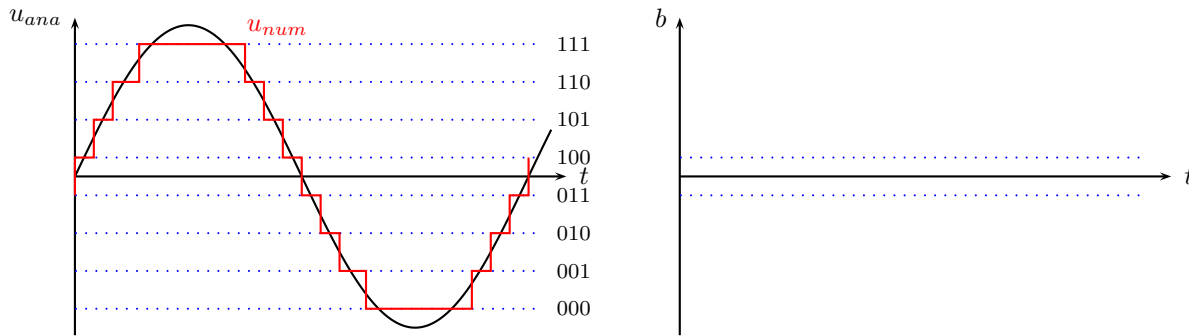


FIGURE 15 – Bruit de quantification $b(t) = u_{num}(t) - u_{ana}(t)$ pour $2^3 = 8$ pas (3 bits) avec basculement au milieu du pas de quantification

4.2 Chaîne de traitement

On peut résumer la situation par un schéma-bloc comme celui de la figure 16. Lorsqu'on emploie le terme de convertisseur analogique-numérique (CAN), il s'agit du dispositif complet qui intègre les fonctionnalités de conversion en numérique comme celle que nous venons de présenter mais aussi en amont le dispositif d'échantillonnage du signal analogique de départ. À la suite du convertisseur, on trouvera l'unité de traitement des données qui va permettre de les mettre en mémoire ou d'effectuer un certain nombre d'opérations dessus. Ces données numériques peuvent être ensuite utilisées sous cette forme mais on peut aussi reconstituer un signal analogique. L'opération va consister à faire une conversion du numérique vers l'analogique à l'aide de circuits électroniques adaptés à cette fonction : c'est le rôle du CNA ou convertisseur numérique-analogique.

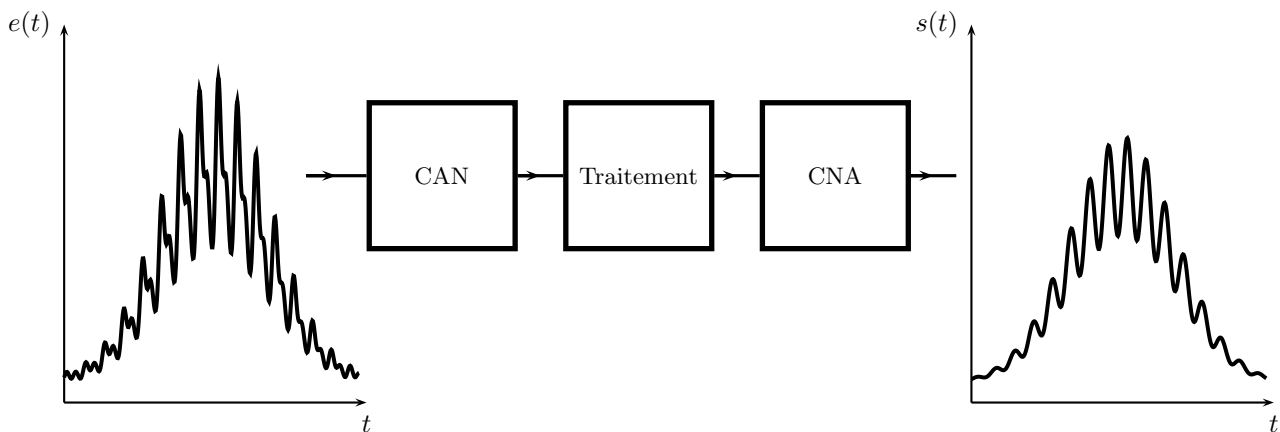


FIGURE 16 – Schéma-bloc d'une unité de traitement numérique du signal

La numérisation présente de nombreux avantages par rapport au traitement analogique des signaux. Toutefois, il ne faudrait pas penser que la méthode est miraculeuse. En effet, dès que l'on effectue l'échantillonnage du signal de départ $e(t)$, on le modifie même si l'on respecte la condition de SHANNON. De plus, comme nous l'avons vu au niveau de la quantification, deux valeurs différentes du signal analogique peuvent être converties en une même valeur en numérique car leur écart est inférieur au pas de quantification que nous avons noté q . L'erreur engendrée est appelée *bruit de quantification*. Certes, cette erreur est un inconvénient de la méthode mais elle a un intérêt : celui d'être parfaitement connue. On peut donc l'adapter en fonction des besoins. Les atouts fondamentaux de la numérisation sont liés aux possibilités de traitement et au stockage de l'information. Même si le signal est altéré, il est possible de le régénérer en utilisant des programmes informatiques qui détectent les erreurs et ensuite les corrigent (codes correcteurs d'erreurs). La transmission de l'information sous forme numérique est donc plus fiable que la transmission sous forme analogique car il y a une certaine réversibilité dans le domaine numérique.

5 Filtrage numérique

Parmi les opérations de traitement du signal numérique, une des plus fréquemment rencontrée est le filtrage. Nous allons voir dans ce qui suit deux exemples de filtres très classiques.

5.1 Filtre passe-bas du premier ordre

Le filtre passe-bas du premier ordre analogique est constitué de la mise en série d'une résistance R et d'une capacité C , voir le schéma de la figure 17.

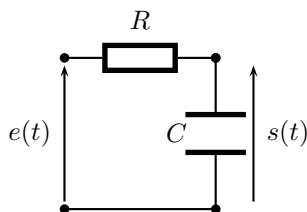


FIGURE 17 – Filtre analogique passe-bas du premier ordre

Sa fonction de transfert est très simple à établir grâce à un diviseur de tension :

$$\underline{H}(j\omega) = \frac{\underline{s}}{\underline{e}} = \frac{1}{1 + jRC\omega} = \frac{1}{1 + j\omega\tau}$$

On imagine qu'on envoie sur ce filtre un signal échelon de HEAVISIDE avec $e(t < 0) = 0$ et $e(t \geq 0) = E$. Pour obtenir la réponse de ce filtre, on peut établir la transformée de FOURIER du signal échelon $e(t)$ et pour chaque fréquence présente étudier l'effet de $\underline{H}(j\omega)$. Ceci serait particulièrement fastidieux. Par une autre forme du signal d'entrée, cette méthode serait tout à fait adaptée mais, ici, cela n'est pas le cas. Il faut, comme vous l'avez d'ailleurs fait en classe de Sup, revenir à la présentation de la situation en mode temporel où la multiplication par $j\omega$ traduit une opération de dérivation par rapport au temps. On obtient donc $(1 + j\omega\tau)\underline{s} = \underline{e}$. Cela revient à considérer pour des dates positives, l'équation différentielle :

$$\tau \frac{ds}{dt} + s = E$$

Cette équation différentielle possède une solution triviale à établir :

$$s(t) = E \left(1 - \exp - \frac{t}{\tau} \right)$$

Jusque-là, nous avons fait un traitement analogique du filtre. Pour traiter le signal échelon qui a été numérisé, on procède de la façon suivante : le filtre reçoit en entrée des valeurs successives e_n de $e(t)$ à raison d'une valeur toutes les périodes T_e (période d'échantillonnage). Ces valeurs sont utilisées pour déterminer les valeurs successives s_n de la sortie. Il faut établir le lien entre e_n et s_n en établissant la relation de récurrence qui découle de l'équation différentielle. Puisque le signal a été discrétisé par l'opération de numérisation, on assimilera la dérivée à un taux de variation en suivant la méthode d'EULER avec comme intervalle de temps, la période d'échantillonnage puisque tous les T_e , les valeurs des grandeurs changent. On a donc :

$$\tau \frac{ds}{dt} + s(t) = e(t) \quad \text{ou bien} \quad \tau \frac{s_n - s_{n-1}}{T_e} + s_n = e_n$$

On peut poser $\theta = \tau/T_e$ la constante de temps relative du filtre numérique. De l'équation précédente, on peut déduire que la valeur de la sortie s_n à une date nT_e (par exemple) est obtenue comme fonction de l'état de cette même sortie à la date $(n-1)T_e$ et de la valeur de l'entrée à la date nT_e :

$$s_n = \frac{\theta}{1 + \theta} s_{n-1} + \frac{1}{1 + \theta} e_n$$

En prenant en compte les valeurs initiales, on peut établir toutes les valeurs discrètes en sortie du filtre. On peut écrire un code *Python* pour réaliser ces calculs. Les valeurs discrètes obtenues sont très proches de celles obtenues en traitement analogique.

```

from pylab import *
CI=0.
# valeur de l'entrée (échelon)
E=2.
# nombre total de points du calcul
Ntot=200
# durée de l'intervalle de travail
Duree=20
# constante de temps et de temps réduite Ctemps
Tau = 2.
dt=Duree/Ntot
Ctemps=Tau/dt
# création d'une liste de valeur de la sortie initialisée à la valeur de départ
s=[CI]
# relation fonctionnelle du filtre passe-bas et initialise le premier terme (pour le temps aussi)
t=[0]
sn=CI
for n in range(Ntot) :
    sn = (Ctemps/(1+Ctemps))*sn + (1/(1+Ctemps))*E
    s.append(sn)
    t.append(n*dt)
# s.append ajoute à la liste s les termes sn successifs
x=linspace(0 ,Duree ,1000)
def y(x) :
    return E*(1-exp(-x/Tau))
clf() #nettoie d'une éventuelle figure précédente
axis([0,Duree,0,1.2*E]) # définit les axes
plot(t,s,marker='o',linestyle="",color='r')
# commande l'exécution du tracé, pas de ligne et des ronds rouges pour les points
plot(x,y(x)) # commande l'exécution du tracé de l'exponentielle
title("Réponses analogique et numérique sur 200 points")
show() # montre le tracé!

```

Sur les graphiques de la figure 18, on peut voir l'effet d'un traitement numérique d'une entrée en échelon dans le circuit RC passe-bas traditionnel. On peut constater que le choix du pas de temps de calcul est important. Cela peut se comprendre facilement lorsque la fonction d'entrée varie lentement, on peut avoir un rythme de calcul plus lent que lorsqu'elle varie rapidement. Lorsque la fonction présente à la fois des variations rapides - comme c'est le cas pour un échelon - et des variations lentes et même plus de variation du tout, le choix du pas de temps de calcul doit être en fait en fonction des évolutions rapides si l'on veut les représenter correctement. C'est ce que l'on constate en effectuant soit 20 itérations sur le graphique de gauche, soit 200 sur le même intervalle de temps total pour l'étude de la sortie du filtre RC .

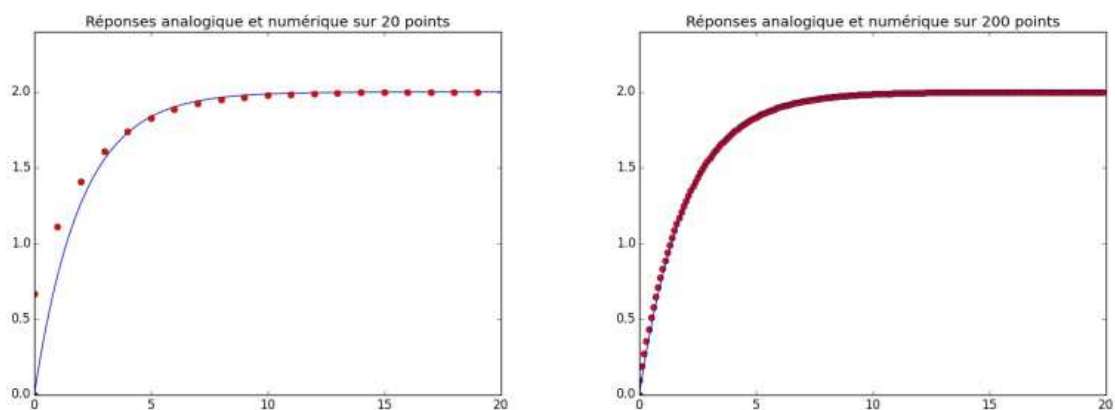


FIGURE 18 – Traitement numérique d'un filtre passe-bas du premier ordre

5.2 Un autre filtre passe-bas numérique : la moyenne glissante

La figure 19 montre l'effet d'une moyenne glissante sur un signal bruité par des hautes fréquences.

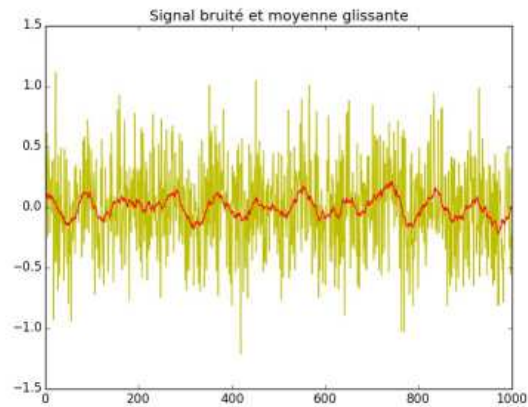


FIGURE 19 – Effet d'une moyenne glissante sur un signal bruité

5.3 Filtrage numérique par Transformée de Fourier

En utilisant la transformée de FOURIER rapide (FFT), on peut filtrer un signal bruité de façon efficace. Le principe repose sur la réalisation de la FFT du signal bruité. Ensuite, on récupère le tableau des fréquences et de leurs amplitudes respectives. On tronque la liste en éliminant les hautes fréquences qui nous paraissent indésirables¹. Ensuite, on refait une transformée de FOURIER pour, à partir du spectre tronqué, revenir au signal en fonction du temps, on parle alors de transformée de FOURIER inverse. Le résultat de l'opération est visible à la figure 20. C'est l'occasion de rappeler la définition couramment utilisée en Physique de la transformée de FOURIER d'une fonction $f(t)$, elle est notée $g(\omega)$. On dit que $g(\omega)$ est le spectre de $f(t)$:

$$TF(f(t)) = g(\omega) = \frac{1}{\sqrt{2\pi}} \int_{t \text{ de } -\infty}^{\infty} f(t) \exp -i\omega t dt$$

Ainsi, $f(t)$ est la transformée de FOURIER inverse de $g(\omega)$:

$$TF^{-1}(g(\omega)) = f(t) = \frac{1}{\sqrt{2\pi}} \int_{\omega \text{ de } -\infty}^{\infty} g(\omega) \exp i\omega t d\omega$$

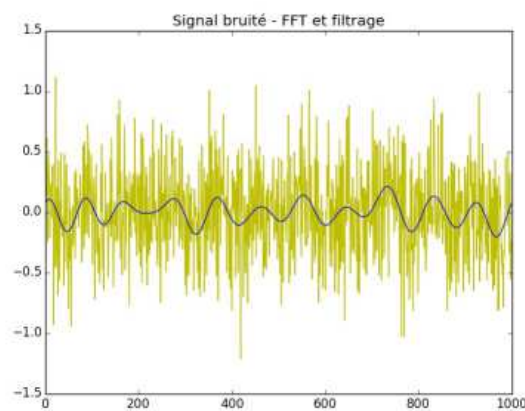


FIGURE 20 – Filtrage numérique par FFT sur un signal bruité

1. Encore faut-il avoir une idée de ce que l'on cherche. . .

```
import math, wave, sys, struct
import scipy.odr
import numpy as np
import scipy as sp
import matplotlib.pyplot as mpl
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from scipy.fftpack import fft, ifft
from scipy import interpolate
import csv

# Pour l'affichage interactif des graphiques avec Idle :
plt.ion()
# Chargement du csv dans un tableau numpy
bruit = open('SinBruit.csv','r')
br = csv.reader( bruit,delimiter=',')
donneesbruitees=[]
for row in br :
    # On remplace le séparateur décimal virgule par un point... Pb du csv...
    donneesbruitees.append(float(row[0].replace(',','.')))
donneesbruitees=np.array(donneesbruitees)
# Lissage par moyenne glissante :
# Taille de fenetre
p=30
fenetre=np.ones(p)
# Avec convolve, c'est plus direct...
donneesmoinsbruitees=np.convolve(fenetre,donneesbruitees,'same')/p
# Tracé
plt.figure(1)
plt.clf()
plt.plot(donneesbruitees,'y')
plt.plot(donneesmoinsbruitees,'r')

# Méthode fft
# Ne pas se soucier de l'éventuel ComplexWarning (normal)
fftdonneesbruitees=fft(donneesbruitees)
fftdonneesmoinsbruiteesBIS=fftdonneesbruitees
# On filtre les HF
fftdonneesmoinsbruiteesBIS[15 :-15]=0
donneesmoinsbruiteesBIS=ifft(fftdonneesmoinsbruiteesBIS)
plt.plot(donneesmoinsbruiteesBIS,'b')
# Pour l'affichage avec Idle des graphiques
plt.title("Signal bruité - Moyenne glissante - FFT et filtrage")
plt.show()
```