

# **CFMS**

## **Conversion numérique et formats asynchrones**

**Nicolas Sturmel, 14 Fevrier 2022**

14 Fevrier: A/N N/A 2

- Types de convertisseurs (//, sigma delta)
- Noise shaping
- Conversion de fréquence d'échantillonnage sync/async
- Le word clock, la synchronisation des equipment's
- Bonus: filtres numériques, correction d'erreur

A la fin du cours je sais:

- décrire et énumérer les différents types de convertisseurs
- expliquer le fonctionnement du sigma delta
- expliquer le noise shaping et le dithering
- déterminer quand je fais une conversion synchrone, asynchrone
- expliquer grossièrement d'où vient la latence en conversion et en traitements.

# Conversions analogique - numérique et numérique - analogique II

## I. Architectures des convertisseurs analogique – numérique.

On distingue deux grandes familles de CAN basées sur deux approches différentes de l'échantillonnage : les CAN classiques dont la fréquence d'échantillonnage est telle que le spectre du signal converti occupe quasiment toute la bande de Nyquist (*Nyquist Rate ADC*) et les CAN à sur échantillonnage (*Oversampling ADC*) dont seule une partie réduite du bruit de quantification affecte le signal converti.

### CAN classiques.

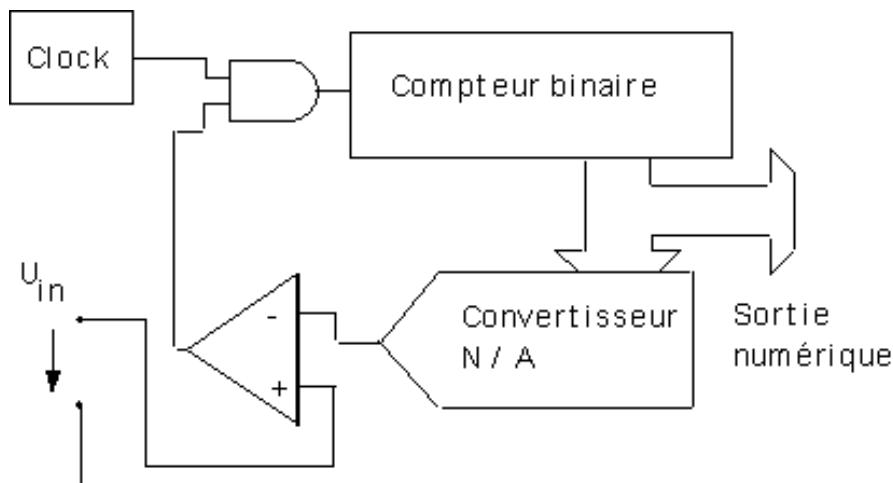
Ils sont basés sur deux principes de conversion, série ou parallèle ; et se subdivisent en trois sous-familles, les CAN série, les CAN parallèle et les CAN série - parallèle.

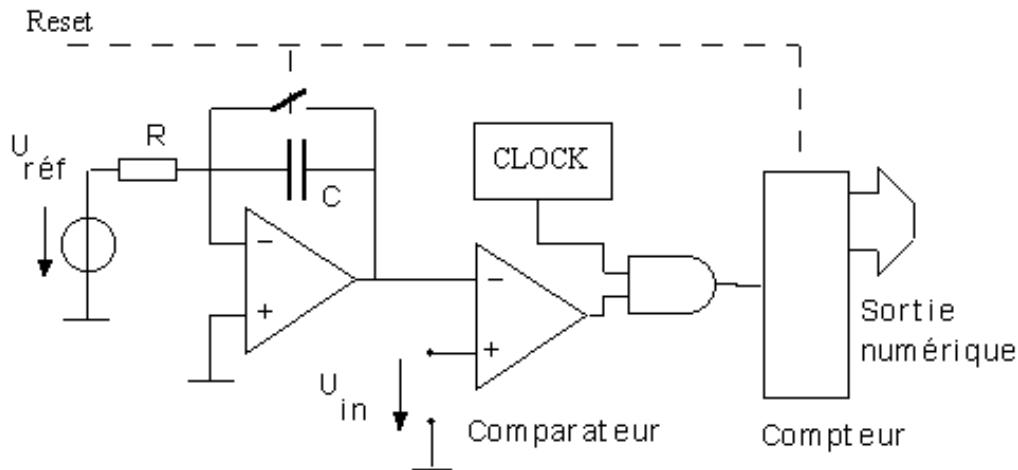
La conversion dans un CAN série est effectuée pas à pas, il en est ainsi des CAN à intégration, à approximations successives et à redistribution de charges.

La conversion parallèle consiste à comparer simultanément la valeur à convertir à tous les seuils, le nom donné à ces convertisseurs est CAN Flash.

Les CAN série – parallèle combinent les deux approches afin de tirer partie de leurs avantages respectifs tout en limitant les effets de leurs défauts.

### Convertisseur par intégration.





Il existe plusieurs sortes de convertisseurs à intégration basés sur un principe similaire, le décompte du temps écoulé lors de la charge d'une capacité.

### Convertisseur à approximations successives (SAR converter).

La figure II.18 donne la topologie d'un convertisseur à approximations successives (SAR converter, pour *Successive Approximation Register*).

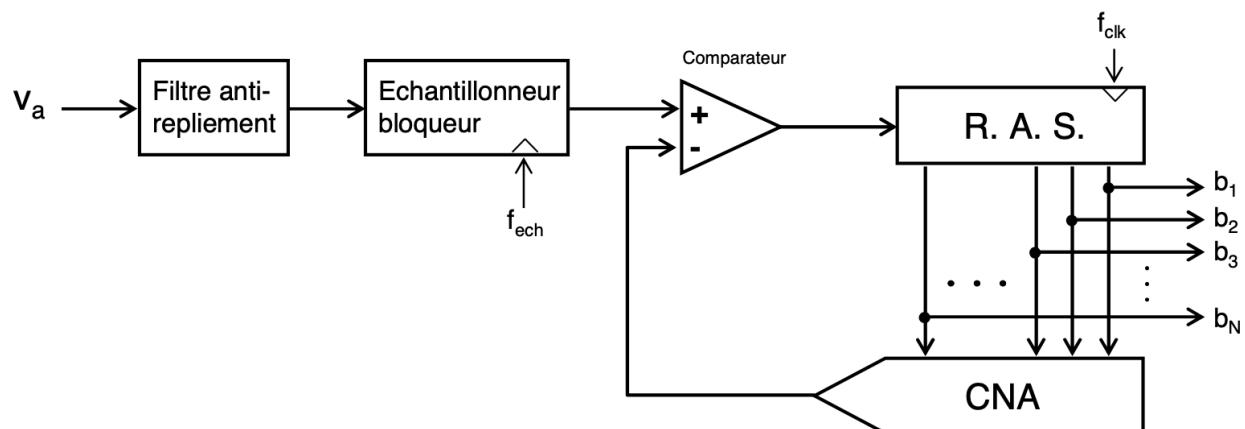


Fig. II.18 – Convertisseur à approximations successives.

Le signal analogique à convertir passe par un filtre anti-repliement puis est échantillonné et bloqué pendant toute la phase de conversion.

Le CAN possède une boucle de rétroaction, constituée d'un CNA de même résolution et d'un comparateur qui commande un Registre à Approximation Successive (RAS, qui donne son nom à cette architecture).

Le principe de conversion est basé sur une recherche du code de sortie par dichotomie (cf. Fig. II.19), à chaque coup d'horloge l'intervalle de recherche est divisé par 2. En début de conversion tous les bits de sortie

(RAS et CAN) sont positionnés à zéro à l'exception du MSB,  $b_1$ , qui est fixé à un. Le mot binaire correspondant (100...0) est présenté au CNA qui délivre en sortie une tension  $V_{ref}/2$ . Cette dernière est comparée à  $v_a$ . Si  $v_a$  est inférieur à  $V_{ref}/2$  alors  $b_1$  passe à zéro, dans le cas contraire il reste à un ; dans les deux cas il s'agit de la valeur finale de conversion du bit considéré.

Tous les bits de sortie jusqu'au LSB sont testé successivement sur le même principe. C'est le RAS, commandé par la sortie du comparateur, qui gère les valeurs données aux bits.

La figure II.19 présente l'évolution de la sortie numérique d'un CAN 3 bits pour une tension analogique  $v_a$  correspondant au code de sortie 101.

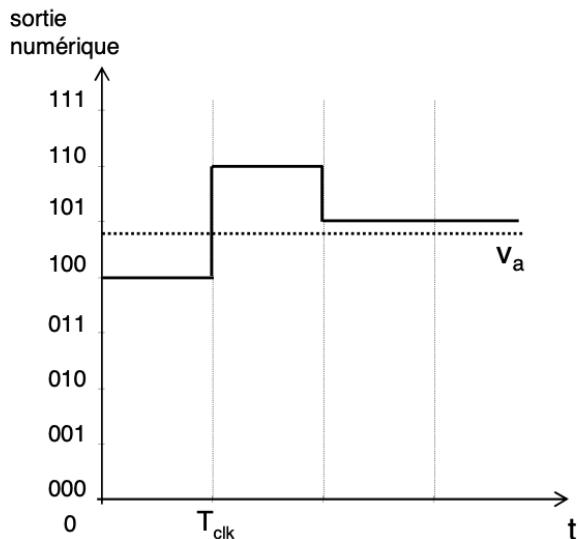


Fig. II.19 – Illustration conversion.

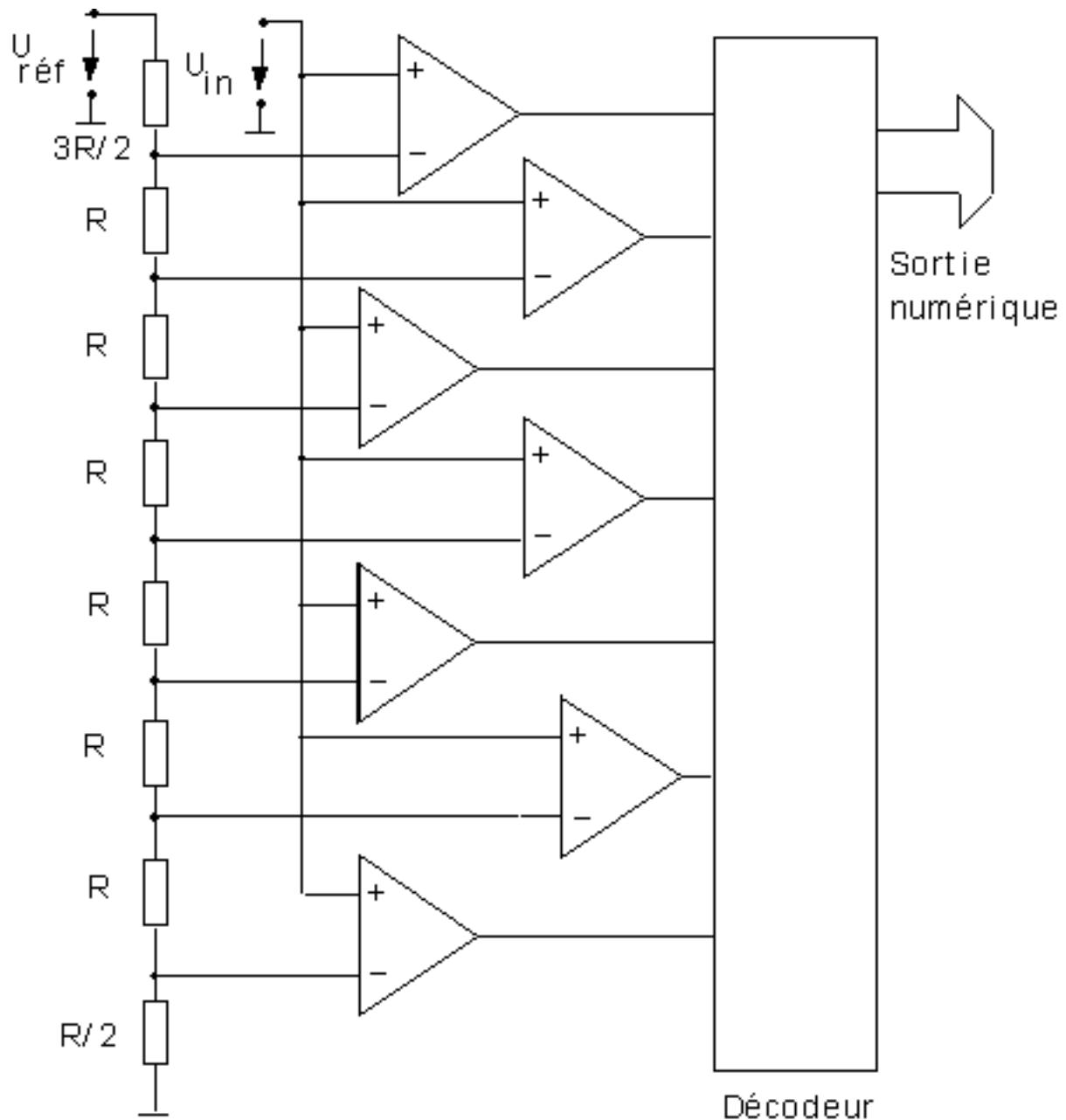
La précision du comparateur doit être supérieure au LSB.

En première approximation, le temps de conversion est  $N \cdot T_{clk}$  (à noter cependant que la durée des phases élémentaires s'accroît pour les bits de poids le plus faible afin de laisse le temps au signal issu du CNA de se stabiliser avant la comparaison).

C'est une architecture de conception ancienne, mais encore très répandue. On trouve des CAN SAR jusqu'à 18 bits et quelques MHz.

## Convertisseur Flash.

Le convertisseur flash est un convertisseur parallèle, l'entrée analogique à convertir est comparée simultanément aux  $2^N - 1$  tensions de seuils (pour un CAN N bit).



Ces tensions de seuil sont obtenues par un pont diviseur comportant 2 en série entre  $V_{ref}$  et la masse. Si toutes les résistances sont identiques on obtient des tensions de seuil correspondant à une quantification linéaire par défaut. Pour obtenir une quantification linéaire centrée, la résistance

connectée à la masse est prise égale à  $R/2$  et celle connectée à  $V_{ref}$  égale à  $3R/2$  (c'est le cas de l'exemple présenté sur la figure).

Un CAN flash à  $N$  bits comporte  $N^2 - 1$  comparateurs (un pour chaque seuil à comparer),  $N^2 - 1$  bascules d'échantillonnage et une logique de conversion.

Chacun des comparateurs délivre en sortie le résultat de la comparaison entre la tension de seuil correspondante et le signal analogique  $v_a$ , le résultat est stocké dans une bascule d'échantillonnage.

En considérant une valeur de  $v_a$  supérieure à la tension de seuil  $V_k$  de moins d'un LSB, les comparateurs de  $C_k$  à  $C_{2N-1}$  dont les tensions de seuils associées sont inférieures à  $v_a$  délivrent en sortie un 1 logique ; les comparateurs de  $C_{k-1}$  à  $C_1$  dont les tensions de seuils associées sont supérieures à  $v_a$  délivrent en sortie un 0 logique, d'où la nécessité d'inclure une logique de conversion vers un code binaire classique.

La conversion est réalisée en un seul cycle d'horloge, ce type de convertisseur est donc par essence extrêmement rapide.

Cependant leur complexité croît exponentiellement avec  $N$  le nombre de bits (en 2). Le coût résultant en terme de surface ( $2 - 1$  comparateurs,  $2 - 1$  bascules), pour une résolution élevée limite leur emploi à une douzaine de bits (on gardera à l'esprit  $2 = 4096$ ).

### **Convertisseur pas à pas (multi-step).**

Les convertisseurs série – parallèles marient et tempèrent les avantages et les inconvénients des approches précédentes.

Un exemple de convertisseur pas à pas 8 bits est présenté figure II.22.

Un premier convertisseur parallèle (CAN 4bits MSB) détermine les bits de poids fort correspondant au signal analogique  $v_a$ . Le résultat de cette conversion partielle est stocké dans un registre de bascule en attendant la fin de la conversion. Un CNA, sur 4 bits également, reconvertis le mot obtenu en un signal analogique qui est soustrait à  $v_a$ .

Un second CAN parallèle sur 4 bits (CAN 4 bits LSB) permet de déterminer les 4 bits de

poids faible. Avant cette dernière conversion, le signal analogique est amplifié avec un gain  $N^2 = 16$ , à fin d'être recalé en pleine échelle ; cela

permet d'utiliser deux CAN identiques ayant la même tension de référence.

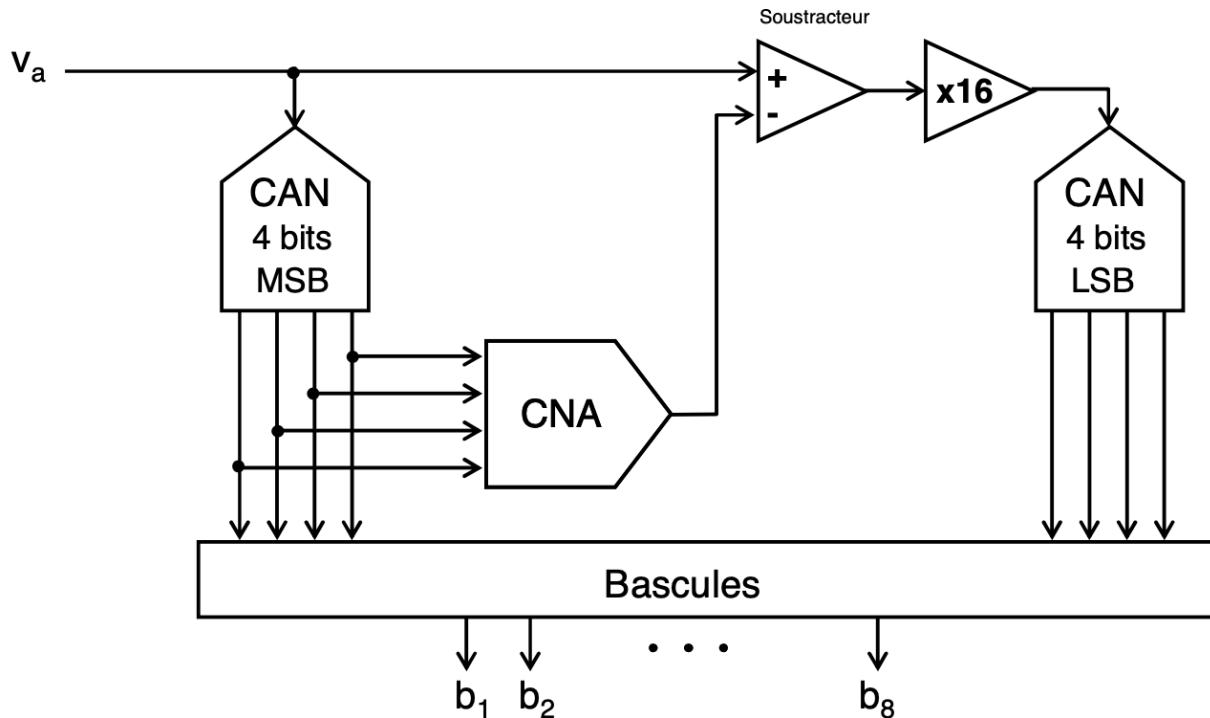


Fig. II.22 – Convertisseur multi-step.

Les deux CAN utilisés sont des CAN Flash, ainsi une conversion est réalisée en deux coups d'horloge. Cette durée,  $2 T_{clk}$ , correspond à un retard entre l'instant où le signal analogique est présenté en entrée du CAN et l'instant où la conversion est terminée.

Cependant, lors du calcul des LSB le premier CAN est de nouveau disponible. Il peut donc être utilisé pour calculer les MSB d'une nouvelle valeur échantillonnée de  $v_a$ . On obtient alors un résultat de conversion tous les coups d'horloge, avec un retard de deux coups d'horloge sur le signal analogique.

La dégradation du temps de conversion s'accompagne d'un gain en surface important.

**Application:** quel est le gain en composants pour un convertisseur 8 bits.

## Convertisseur Pipeline.

Les convertisseurs Pipeline sont l'application du principe précédent poussé à l'extrême : un CAN N bits étant réalisé avec N étages de 1 bit.

## Convertisseur à sur échantillonnage, convertisseur Sigma – Delta.

### Principe du sur échantillonnage.

Les convertisseurs A/N dont nous venons d'étudier le principe dans ce qui précède sont dits à échantillonnage de Nyquist (Nyquist Rate Converters), leur fréquence d'échantillonnage est choisie du même ordre de grandeur que le double de la fréquence maximale contenue dans le signal échantillonné (et bien évidemment supérieure à  $2f_{\max}$ ). La représentation spectrale associée correspond à la partie gauche de la figure II.25. Le recours à une fréquence d'échantillonnage supérieure de plusieurs ordres de grandeur à  $2f_{\max}$  permet d'accroître le rapport sur bruit d'un convertisseur (cf. partie droite de la figure II.25).

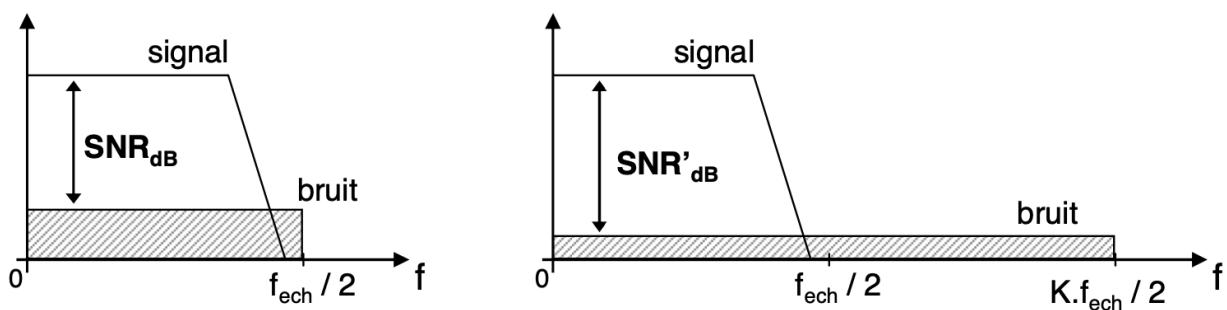


Fig. II.25 – Effet du sur échantillonnage sur le bruit de quantification.

On obtient alors un rapport signal sur bruit amélioré exprimé par :  $\text{SNR}'_{\text{dB}} = 6,02.N + 1,76 + 10.\log K$

Ainsi, le fait de multiplier la fréquence d'échantillonnage par 4 permet d'augmenter la rapport signal sur bruit de  $10\log(4) = 6,02 \text{ dB}$  (ce qui correspond également au gain du  $\text{SNR}_{\text{dB}}$  lorsque l'on augmente le nombre de bits de 1).

On notera également, que le sur échantillonnage, permet de relâcher les contraintes sur le design du filtre anti-repliement (coupure à  $Kf_{\text{ech}}/2$ ).

Les convertisseurs qui utilisent ce principe sont appelés convertisseurs à sur échantillonnage (*oversampling converters*). C'est le cas du convertisseur A/N sigma – delta présenté au paragraphe suivant.

## Convertisseur Sigma – Delta ( $\Sigma\Delta$ ).

La figure II.26 présente l'architecture d'un convertisseur A/N Sigma – Delta.

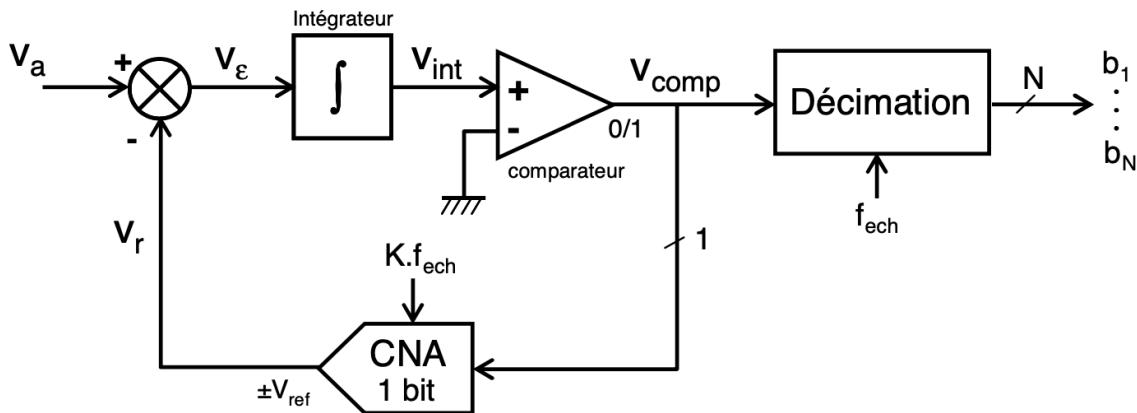


Fig. II.26 – Topologie d'un convertisseur sigma – delta.

Le signal analogique d'entrée  $v_a$  est converti en mots binaires de  $N$  bits à la fréquence  $f_{\text{ech}}$  (fréquence de Nyquist). Le sur échantillonnage est localisé au niveau du système bouclé constitué par un soustracteur, un intégrateur, un comparateur et un convertisseur N/A sur 1 bit (sa fréquence de conversion est  $Kf_{\text{ech}}$ ,  $K$  est appelé facteur de sur échantillonnage).

Le signal analogique  $v_e$  en sortie du soustracteur correspond à la soustraction du signal issu de la boucle de retour  $v_r$  par le signal d'entrée  $v_a$ .  $v_e$  est intégré pour donner en sortie de

l'intégrateur le signal  $v_{\text{int}}$ . Ce dernier est connecté à l'entrée + d'un comparateur dont l'entrée – est reliée à la masse. En sortie du comparateur, le signal  $v_{\text{comp}}$  sur un bit, peut prendre les valeurs logiques 1 ou 0 (pour  $v_{\text{int}} \geq 0$  on a  $v_{\text{comp}} = 1$  et pour  $v_{\text{int}} < 0$  on a  $v_{\text{comp}} = 0$ ).

$V_{\text{comp}}$  commande le CNA 1 bit de la boucle de retour ; sa sortie  $v_r$  peut prendre les valeurs  $V_{\text{ref}}$  ou  $-V_{\text{ref}}$  (pour  $v_{\text{comp}} = 1$  on a  $v_r = V_{\text{ref}}$  et pour  $v_{\text{comp}} = 0$  on a  $v_r = -V_{\text{ref}}$ ). La valeur de  $v_{\text{comp}}$  est prise en compte par le CNA sur les fronts montants d'une l'horloge à la fréquence  $Kf_{\text{ech}}$ . Intuitivement, en

assimilant l'intégration d'un signal au calcul de sa valeur moyenne et l'action du comparateur à l'annulation de  $v_{int}$  (en moyenne), on conçoit que ce système assure des valeurs moyennes identiques pour  $v_a$  et  $v_r$ . Pour une fréquence d'échantillonnage largement supérieure à celle de  $v_a$ , la valeur moyenne de  $v_r$  correspond à la valeur instantanée de  $v_a$  (qui se confond avec sa valeur moyenne). De plus  $v_r$  est une image du flot de bits de  $v_{comp}$ , ce dernier donne donc une représentation de  $v_a$ .

On peut modéliser le  $\Sigma\Delta$  par le schéma de la figure II.27 en considérant que  $v_r$  est l'image bruité de  $v_{int}$ , soit  $v_r = v_{int} + b$ , en notant  $b$  le bruit de quantification ajouté par le comparateur (CAN 1 bit).

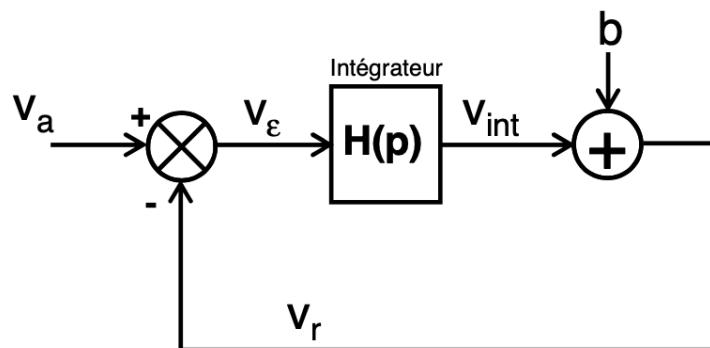
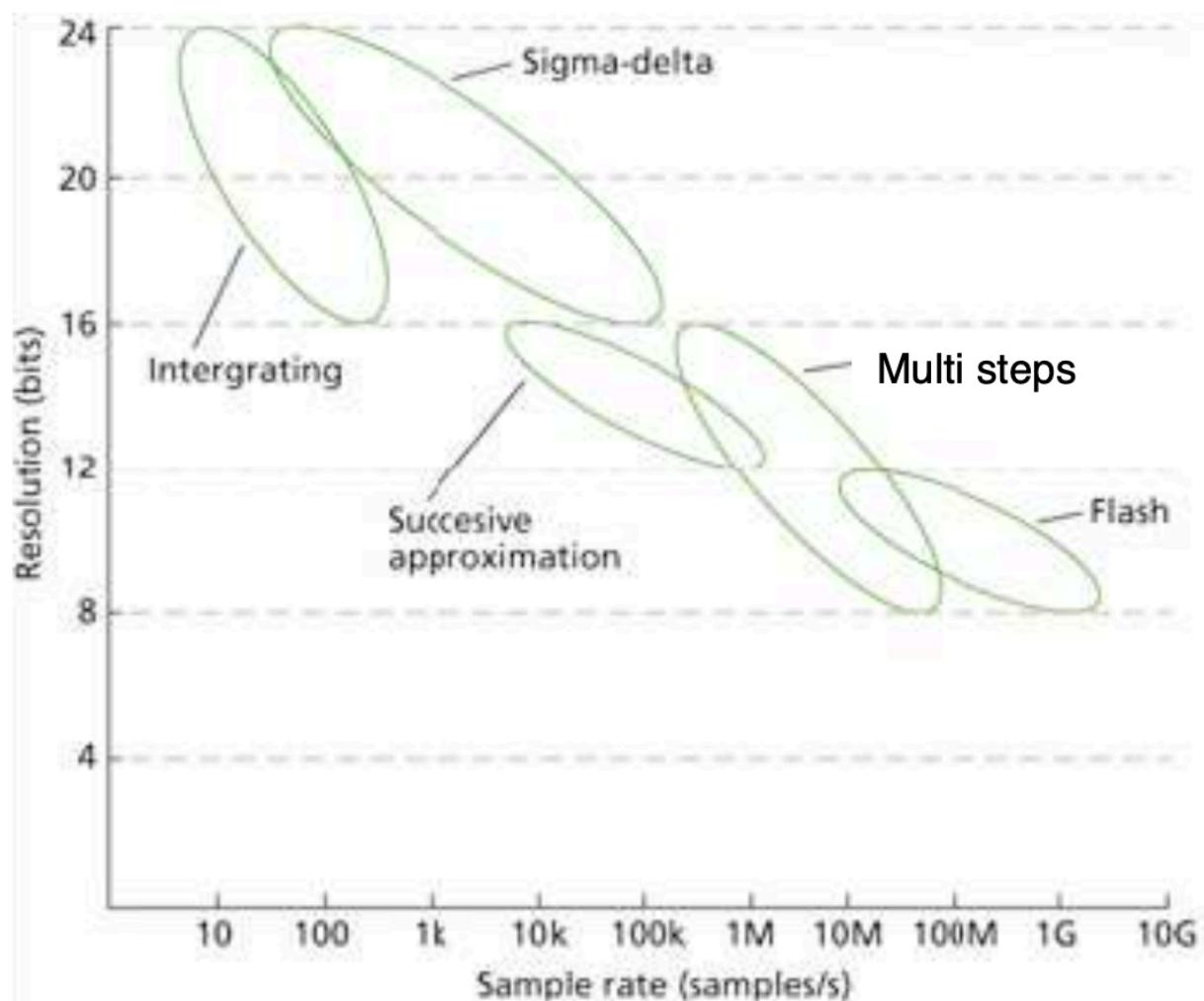
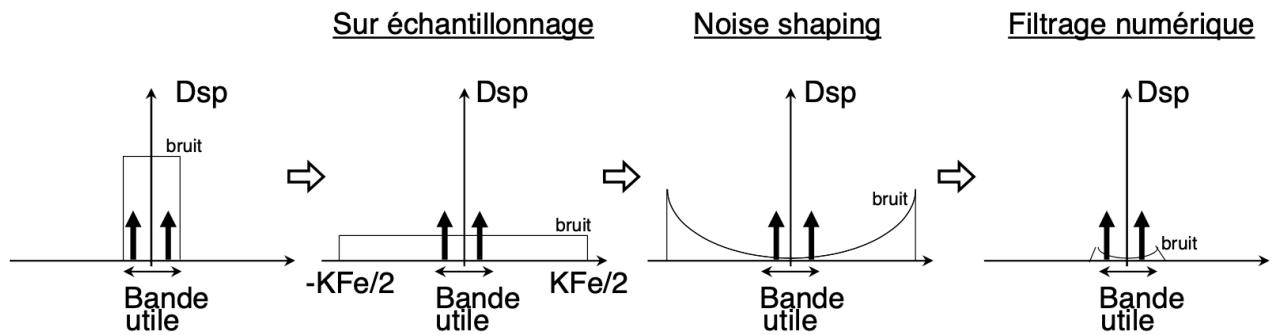


Fig. II.27 – Mise en équation du sigma – delta.

## Conclusion



## Noise shaping



Le noise shaping est l'action de repartir de manière non uniforme le bruit sur l'ensemble de la plage de fréquence du signal. Ceci est particulièrement utilisé quand la bande passante du signal numérisé est très faible par rapport à la bande disponible. Ce noise shopping peut être utilisé pour sous échantillonner un signal en augmentant son SNR.

Il s'agit ici d'estimer et filtrer l'erreur de quantification, dont on n'expliciteras pas ici les détails.

Le noise shaping est utilisé en conversion numérique/numérique:

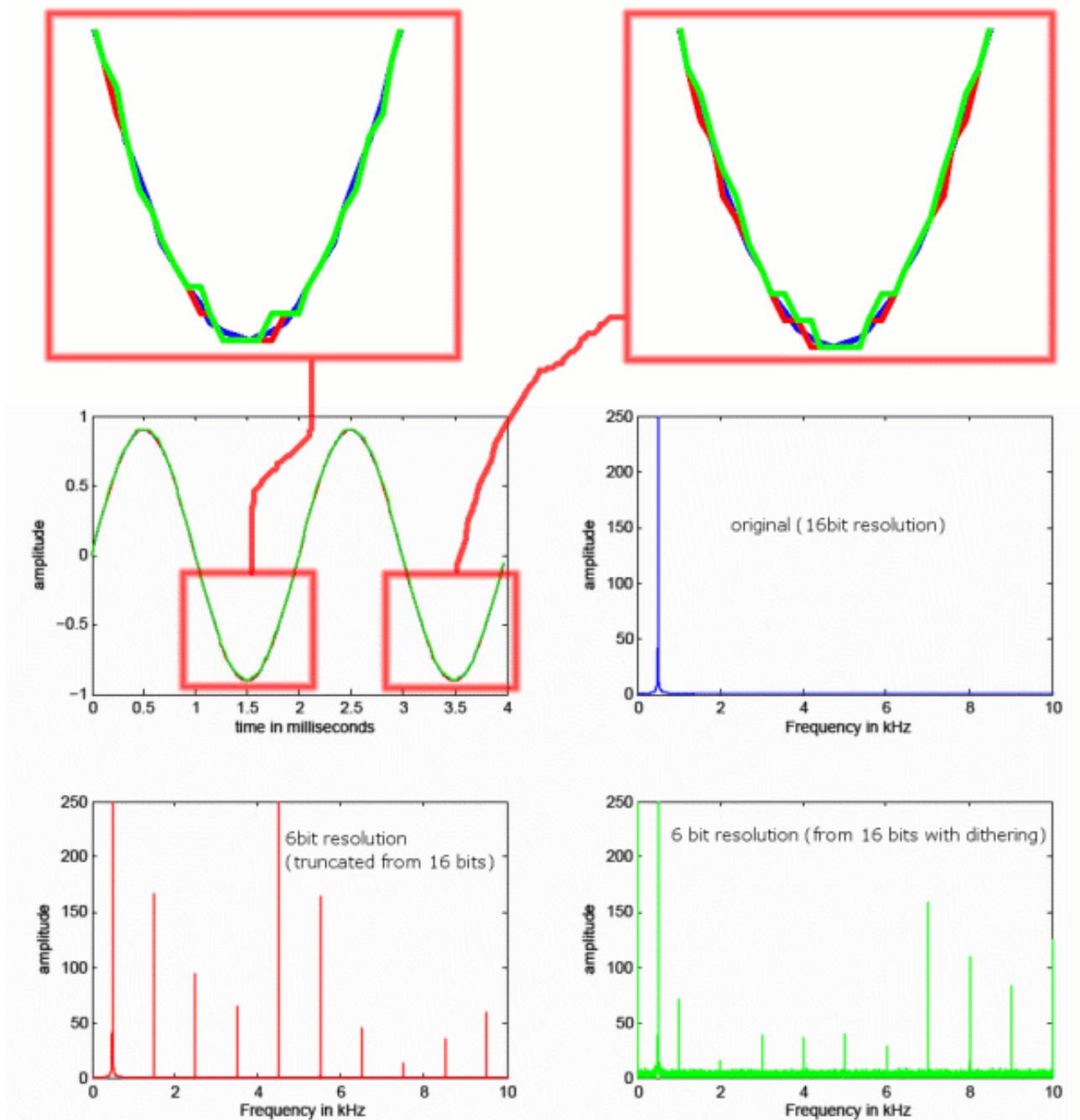
- pour baisser la résolution N en distribuant le bruit sur des fréquences où l'oreille humaine est moins sensible.
- pour baisser la fréquence d'échantillonnage en cachant le bruit dans la partie de spectre qui sera éliminée.

## Demo

Demo avec audacity est un logiciel de traitement du son pour entendre la différence.

## Dithering

Le dithering est l'action de *rajouter* du bruit pour que le résultat soit plus plaisant. En effet, lors d'opération numérique le bruit rajouté peut être carrelé au signal et donc ressenti alors que sa puissance est faible. En rajoutant du bruit on *étaie* la puissance du bruit et, contre intuitivement, on réduit le bruit perçu.



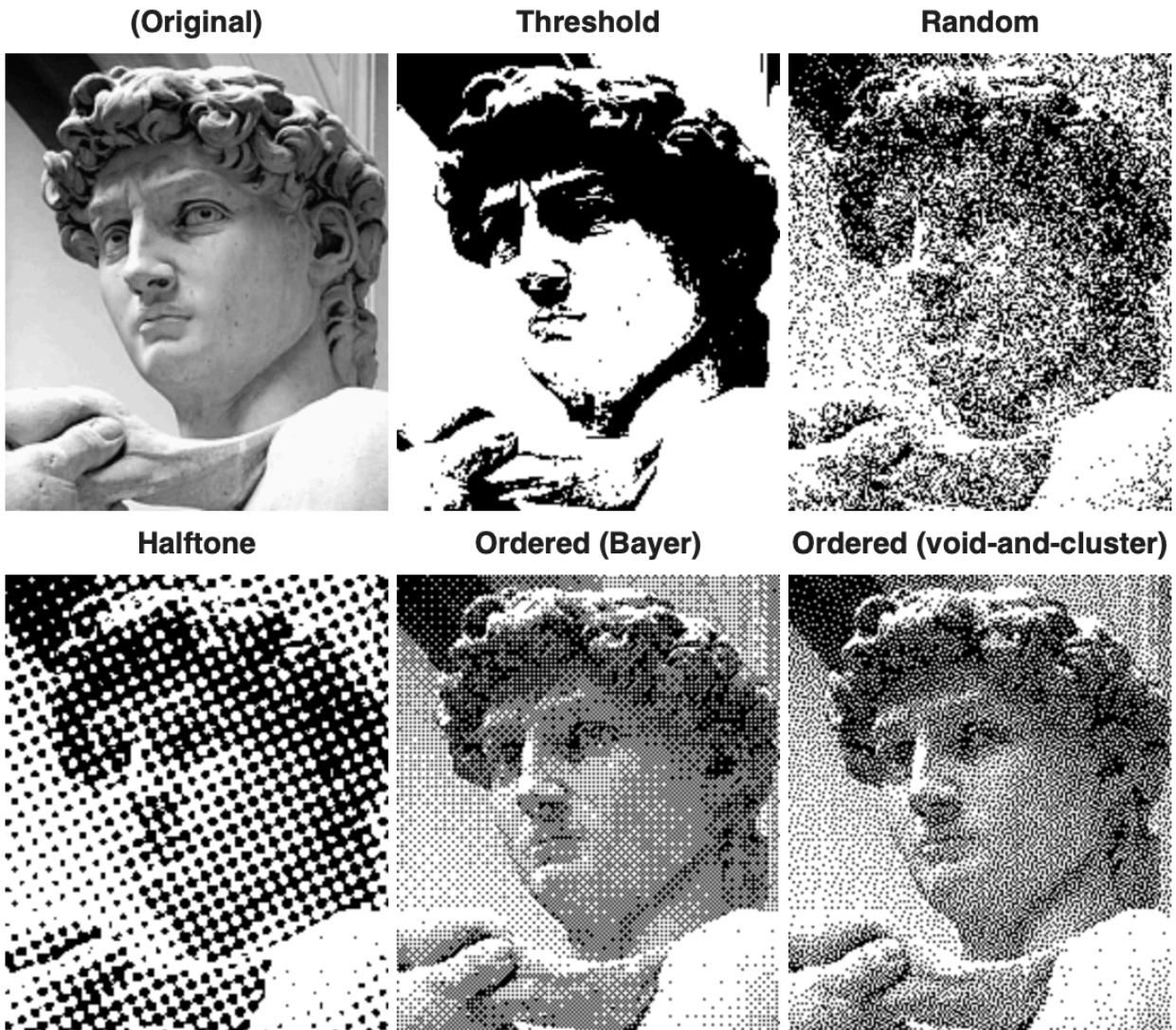
**Reducing amplitude resolution of a 500Hz sine wave from 16 to 6 bits:**

The Blue spectrum shows the original sine at 500Hz.

Truncating to 6 bits introduces harmonics/distortion (multiples of 500Hz) - red spectrum

Dithering reduces the amplitude of these distortions, but introduces background noise - green spectrum (please note that the spectral plots above have been clipped at 250. The amplitude at 500Hz has thus been clipped and is actually much larger than can be seen here)

The sine wave at the top shows that truncation (red) always rounds values the same way while dithering randomizes the choice of rounding up or down (green)



Demo sur wikipedia

## Un petit mot sur les filtres numériques

L'avantage d'une conversion analogique vers numérique reside dans l'aspect très pratique de realiser des traitement, notamment des traitements linéaire

### Definition

**Traitement linéaire:** soit  $H$  une fonction de traitement d'un signal,  $H$  est linéaire si  $H(a+b) = H(a) + H(b)$ .

Comme chaque signal peut être décrit comme une somme infinies de sinusoïdes de fréquence simple, on en déduit qu'un filtrage linéaire ne modifie pas le nombre d'harmoniques d'un signal, mais juste leur phase et leur amplitude, c'est un filtrage.

### Filtrage == traitement linéaire

$$S(n) = \sum_i \sin(2\pi f_i n)$$

$$H(S) = \sum_i H(\sin(2\pi f_i n)) = \sum_i h_i \sin(2\pi f_i n + \phi_i)$$

C'est pour ca qu'un diagramme de bode représente si bien les caractéristique d'un système en premiere approximation.

### Demo

Jouons un peu avec ceci

### Dans une table de mixage

Dans une table de mixage, des processeurs, parfois spécialisé (DSP, FPGA ou VLSI) s'occupent de realiser ces filtrages.

## **Conversion Numerique/Numerique**

Lorsque l'on travaille avec des fichiers ou flux numérique, il peut être nécessaire d'adapter la fréquence d'échantillonnage. Soit pour en changer (de 48k à 44,1k) ou pour s'adapter à une nouvelle horloge (de 48k à 48k mais désynchronisés).

Similaire à ce que nous avons vu précédemment il s'agit avant tout d'éviter les effets de repliement spectraux qui peuvent mener à des artefacts audio disgracieux.

### **Conversion synchrone**

Si on souhaite changer de fréquence d'échantillonnage sans changer de base de temps (le rapport de  $F_i$  et  $F_o$  est constant), on parle de conversion synchrone. C'est typiquement ce cas qui se présente quand vous convertissez un fichier.

On parle souvent de SRC, pour Sample Rate Converter

### **Fréquence sous multiple, sous échantillonnage**

Si on réduit  $F_e$  (de 48k à 24k) il faut limiter la bande passante *avant* de supprimer un échantillon sur 2. Ceci afin d'éviter le repliement des fréquences.

Lorsque  $F_{e1}$  (fréquence initiale) est un multiple de  $F_{e2}$  (fréquence finale), tel que  $F_{e1} = Q.F_{e2}$ , alors sous échantillonner revient à:

- limiter la bande passante du signal à  $F_{e2}/2$
- prendre 1 échantillon tous les  $Q$  échantillons

### **Augmentation de $F_e$ , sur échantillonnage**

Si on augmente  $F_e$  (de 48k à 96k), il faut lisser le signal pour limiter sa bande passante à 24kHz après avoir doublé les échantillons.

Lorsque  $F_{e2}$  (fréquence finale) est un multiple de  $F_{e1}$  (fréquence initiale), tel que  $F_{e2} = P.F_{e1}$ , alors sous échantillonner revient à:

- ajouter  $P-1$  échantillons pour chaque échantillons

- limiter la bande passante du signal à  $F_e/2$ , afin de lisser le signal.

## Cas General

Lorsque  $F_e2 = P/Q F_e1$ , on réalise successivement les deux étapes précédentes.

### Questions:

Dans quel ordre faut-il faire les opérations ?

Quel rapport permet de faire une conversion de 48k à 44,1k ?

## Conversion asynchrone

Lorsque le rapport des fréquences ne peut pas être exprimé comme un quotient, on parle de conversion asynchrone. Dans ce cas le rapport  $F_e1/F_e2$  peut même être variable au cours du temps.

On parle de ASRC (Asynchronous Sample Rate Conversion)

Il s'agit ici de recréer un signal à bande limitée idéal et de le ré-échantillonner. Pour cela on utilise le sinus cardinal qui est un signal à bande limitée idéal.

$$\text{sinc}(x) = \frac{\sin(x)}{x}$$

## Demo

Demo sur scilab, juste pour illustrer

## **Importance du WordClock**

On comprend bien que les calculs d'un ASRC sont beaucoup plus complexes, ce qui nécessite des ressources et qui modifie chaque échantillons du signal.

L'importance de la synchronisation de Fe entre équipements est ici primordiale, ceci évite l'emploi d'ASRC:

- on gagne le temps de traitement (moins de latence)
- on conserve les échantillons dans leur état d'origine

## Ce qu'il faut retenir

- Il y a plusieurs technologies de convertisseurs en fonction de la fréquence d'échantillonnage et de la resolution
- dans la gamme audio (16 a 32bits, 32 a 384kHz) c'est le convertisseur de type sigma delta qui est utilisé
- Les convertisseurs types intégrateurs, parallèles et à approximation successive doivent aussi pouvoir être décrits.
- Aujourd'hui ce n'est plus un convertisseur 1 bit, mais 2 ou 4 pour augmenter la stabilité du noise shaping
- Le noise shaping est le fait de contrôler la distribution du bruit sur le spectre, notamment pour augmenter dynamique et/ou resolution.
- Le dithering est l'action de rajouter du bruit afin d'améliorer la conversion N/N ou un traitement.
- La conversion N/N peut se faire en synchrone (SRC) ou asynchrone (ASRC) en fonction de l'horloge partagée ou pas.
- Bien que les traitements numériques soient aisés, ils coutent chers en latency, il n'est pas rare d'avoir des ASRC qui ont plus d'une milliseconde de latency, ou des sigma delta de latency 10 à 20 échantillons