

Un filtre en cosinus surélevé avec Pspice

J'invite le lecteur à consulter le site pour des informations complémentaires.

Page d'accueil du site Internet : [page d'accueil](#)

d'autres pdf, sur différents sujets : [liste des PDF](#)

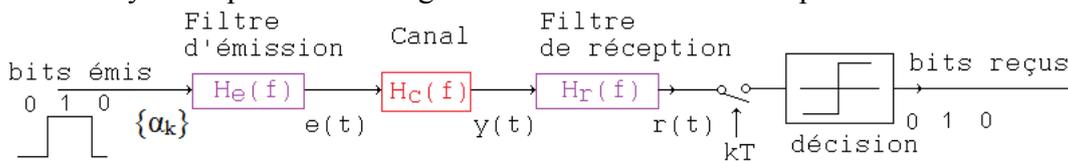
Pour cet article, on suppose connues la théorie des transmissions numériques, et la notion d'interférences entre symboles (IES). Pour une première approche, le lecteur peut consulter mon [cours sur l'électronique pour les communications numériques](#), pages 25 à 32.

L'article présenté ici montre comment, avec Pspice d'évaluation, simuler facilement un filtre à cosinus surélevé. La transmission est en bande de base.

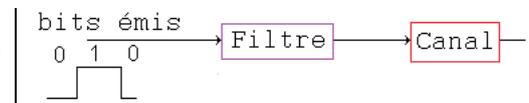
1) CQFS dans une chaîne de transmission numérique

1.a) Représentation de la chaîne de transmission en bande de base

On peut représenter symboliquement les étages entre l'émission et la réception des bits à transmettre :



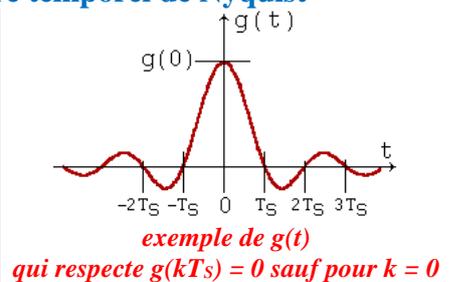
Pour cet article, pour simplifier, on fusionne les 2 filtres et on n'étudie pas le récepteur :



1.b) La réponse impulsionnelle de la chaîne doit respecter le critère temporel de Nyquist

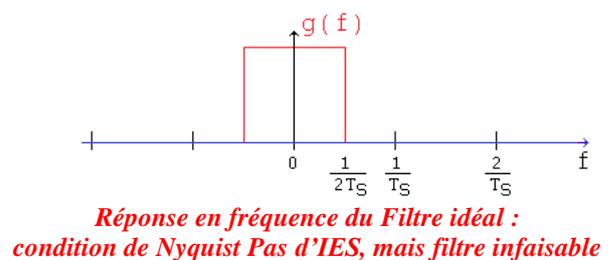
Le **critère temporel de Nyquist** stipule que la réponse impulsionnelle de la chaîne s'annule aux instants d'échantillonnage des symboles au récepteur.

Cela s'exprime par : $g(kT_s) = g(0)$ pour $k = 0$
avec $T_s = T_{\text{symbole}}$. $= 0$ pour $k \neq 0$.



1.c) Dans une transmission numérique, on ne dispose que d'une largeur de bande allouée.

Si un canal dispose d'une bande passante limitée B, il sera pleinement exploité en choisissant $B = 1/2T_s$. Cette limite est la **relation de Nyquist** qui s'écrit également : $R_{\text{max}} = 2 B$ où R est la rapidité de modulation, qui s'exprime en baud ou nombre de symboles / seconde. Cette relation ne sous-entend aucun bruit additif.



Remarque : dans le cas d'une transmission binaire, un symbole = un bit.

1.d) Une solution industrielle : placer un filtre en cosinus surélevé en amont de la transmission

Réponse en fréquence

Il existe un filtre passe bas, réalisable aisément, appelé filtre en **cosinus surélevé**.

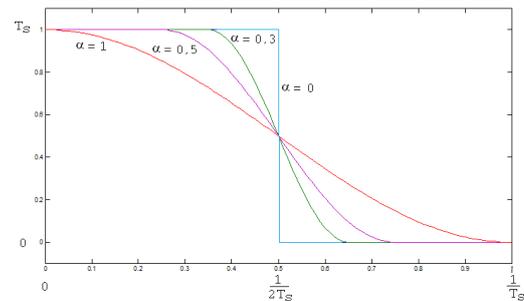
α est le coefficient de roll-off, paramètre de réglage.

$$H_{eq}(f) = \begin{cases} T_s, & |f| < \frac{1-\alpha}{2T_s} \\ \frac{T_s}{2} \left(1 + \cos \left(\frac{\pi T_s}{\alpha} \left(f - \frac{1-\alpha}{2T_s} \right) \right) \right), & \frac{1-\alpha}{2T_s} \leq |f| \leq \frac{1+\alpha}{2T_s} \\ 0, & |f| > \frac{1+\alpha}{2T_s} \end{cases}$$

La réponse harmonique (en échelle linéaire) présente un point de symétrie en $1/2T_s = R_{max}/2$. On se situe à mi puissance, donc à -3 dB. Ce point est un point de symétrie.

- Pour $\alpha = 0$, la réponse est idéale, la bande est $1/2T_s$: c'est le filtre de Nyquist.
- Pour α s'approchant de 1, la bande passante se rétrécit.

L'atténuation maximale est atteinte à $f = \frac{1+\alpha}{2T_s}$



Réponse harmonique, représentée ici en mono latéral

Réponse temporelle du filtre à cosinus surélevé

Sa **réponse impulsionnelle** s'écrit :

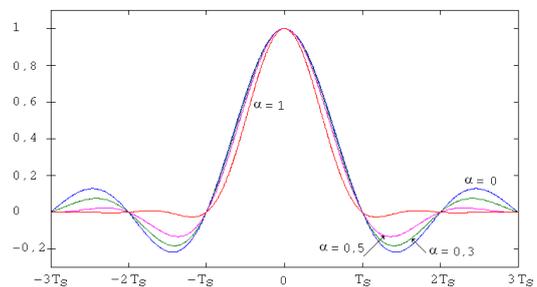
$$h(t) = \sin c \left(\frac{t}{T_s} \right) \frac{\cos \alpha \pi \frac{t}{T_s}}{1 - (2\alpha \frac{t}{T_s})^2}$$

Cette réponse impulsionnelle, paramétrée en α , est donnée ci-contre :

- à $t = 0$, $h(t)$ passe par son maximum = 1.
- à $t = kT_s$, $h(t)$ passe par 0 (conséquence du $\sin c$).

Pour α proche de 1, $h(t)$ est voisin de 0 dès que $t > T_s$. Les ondulations sont négligeables.

Pour α proche de 0, $h(t)$ présente des oscillations amorties pour $t > T_s$.



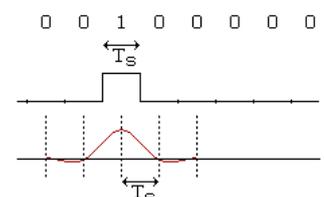
Réponse impulsionnelle

Il faut donc un compromis : α élevé est préférable sur le plan temporel (pas d'étalement de la réponse impulsionnelle), mais demande de la bande passante. Les valeurs usuelles de α sont de 0,2 à 0,6.

2) Première réalisation pratique

On doit intercaler, en début de la chaîne de transmission, un filtre à cosinus surélevé.

Raisonnons tout d'abord sur une transmission d'un seul bit à 1, les autres étant à 0. Le filtre doit donc remplacer le symbole « rectangle » de largeur T_s par le motif en forme de cloche dont l'intervalle entre les 2 zéros fait $2 T_s$.

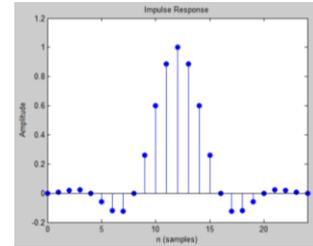


Pour une meilleure visibilité, on ne montre pas le décalage temporel entre l'entrée et la sortie

2.a) La réponse échantillonnée

La réponse temporelle montrée en 1.d) est à temps continu. Mais on peut générer cette forme d'onde par un signal à temps discret en injectant des échantillons dont les valeurs sont $h(kT_e)$, où T_e est une période de sur échantillonnage.

Illustrons ceci par un graphe (donné par la commande `impz(rcosine(1,4),1)` de Matlab) :



Une succession de valeurs analogiques

On peut donc remplacer un symbole (de valeur 1), par $h(t)$ construit par morceaux, et tronqué pour retirer les lobes secondaires de faible niveau.

Il faut donc présenter, tous les T_e , les valeurs analogiques suivantes (on se limite ici au lobe principal + les 2 premiers lobes négatifs, ce qui fait 15 échantillons), et ce, associé à un bloqueur d'ordre 0 pour maintenir les valeurs entre chaque échantillon :

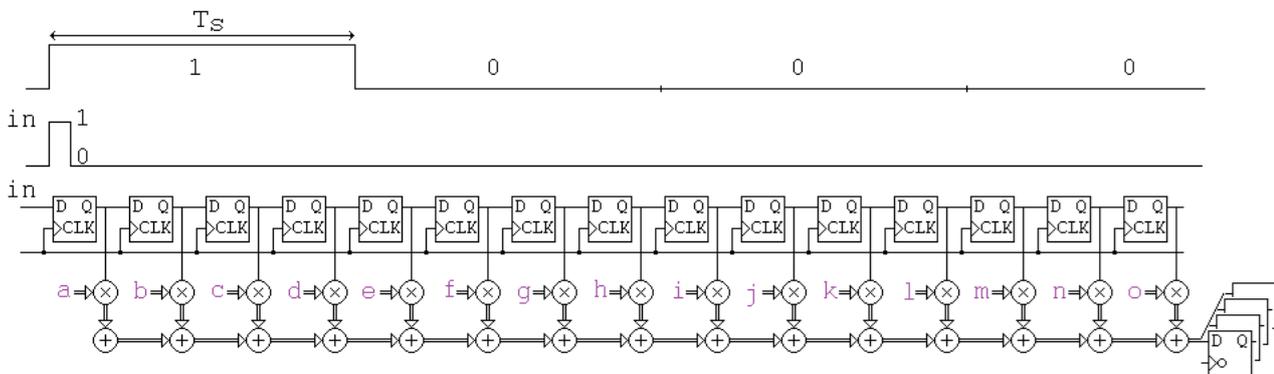
durée du symbole : 4 échantillons

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
-0,057	-0,12	-0,122	0	0,262	0,6	0,887	1	0,887	0,6	0,262	0	-0,122	-0,12	-0,057

- Remarques :**
- Par simplicité d'écriture, ces valeurs sont repérées par une lettre.
 - On repère la valeur de T_s par le passage à 0, soit ici 4 échantillons de T_e .

2.b) Schéma de principe

C'est le schéma ci-dessous :



Un registre à décalage (une succession de 15 bascules D) permet de faire circuler un unique niveau logique haut, tel un chenillard, de gauche à droite. L'horloge est à la période T_e . Ce qui signifie que toutes les sorties Q valent 0, sauf une, qui est le niveau haut se déplaçant, sur les fronts montants de CLK.

Chaque sortie Q de ces bascules D est connectée à une entrée d'un multiplieur numérique qui reçoit sur l'autre entrée le coefficient du tableau précédent (a, b, c, d...) : toutes les sorties de ces multiplieurs affichent 0, sauf celle du multiplieur recevant le niveau 1 logique qui délivre alors la valeur du coefficient qui lui est associé.

Un sommateur logique fait la somme de toutes les sorties. Comme une seule est à une valeur donnée, et les autres nulles, le sommateur ne fait que reporter cette valeur en sa sortie finale.

Remarques :

- Cette topologie est celle d'un filtre RIF.
- Les bascules D font office de bloqueur d'ordre 0 : la valeur de Q est conservée durant une période T_e .

- En général, les multiplieurs binaires manipulent des opérands sur N bits, et le résultat est sur 2 N bits. Ici, un opérande est un booléen (1 bit) et l'autre la valeur analogique convertie en numérique. Il faut donc voir dans ce schéma 15 convertisseurs A/N qui délivrent les valeurs converties en binaire de a, b, c etc. (Dans notre exemple, on pourrait simplifier le schéma pour tenir compte que d = 0, h = 1 et l = 0).
- En général, les additionneurs binaires manipulent des opérands sur N bits, et le résultat est sur N+1 bits.
- Ici, un seul terme est actif, et est codé sur N bits. **D'où tous les bus de ce schéma sont formatés N bits.**
- L'information binaire issue du sommateur présente des valeurs intermédiaires avant d'afficher la valeur exacte : ceci est dû au temps de propagation des opérations fait en interne dans les additionneurs. Il faut donc ajouter un registre de mémorisation qui capture et verrouille le résultat quand ce dernier est stable. C'est un ensemble de N bascules D qui réalise cela, commandé par le même signal CLK, mais retardé. Une solution peut être l'exploitation de N bascules D sur front descendant.

2.c) Schéma de simulation Pspice (fichier cosinus_15.cir)

On réalise un schéma fonctionnel : les bascules D sont remplacées par des cellules retard de durée T_e (par une source de type E associée à une fonction de Laplace), et, pour représenter les multiplieurs et additionneurs, on place une source de tension commandée par 15 tensions avec pondération (source E). C'est donc une simulation comportementale que l'on exécute. On n'a pas une représentation binaire des bus de N fils, mais une information analogique en volt.

Il faut une horloge délivrant le signal de période T_e : on choisit une source PWL.

La largeur de T_e doit être égale à $T_s/4$: après 4 décalages (ici, 4 retards), le décalage est de T_s .

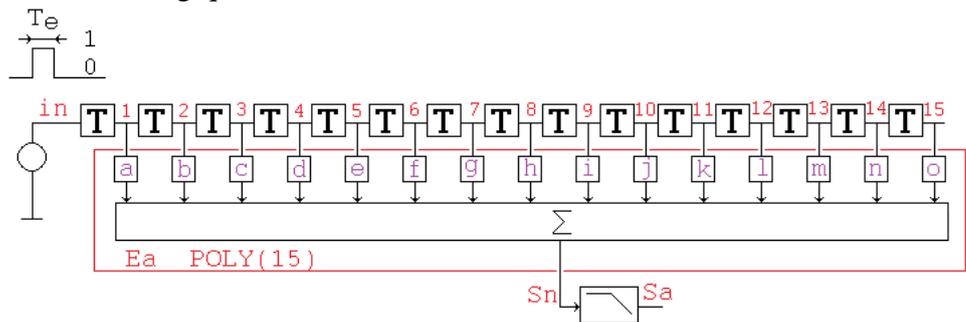
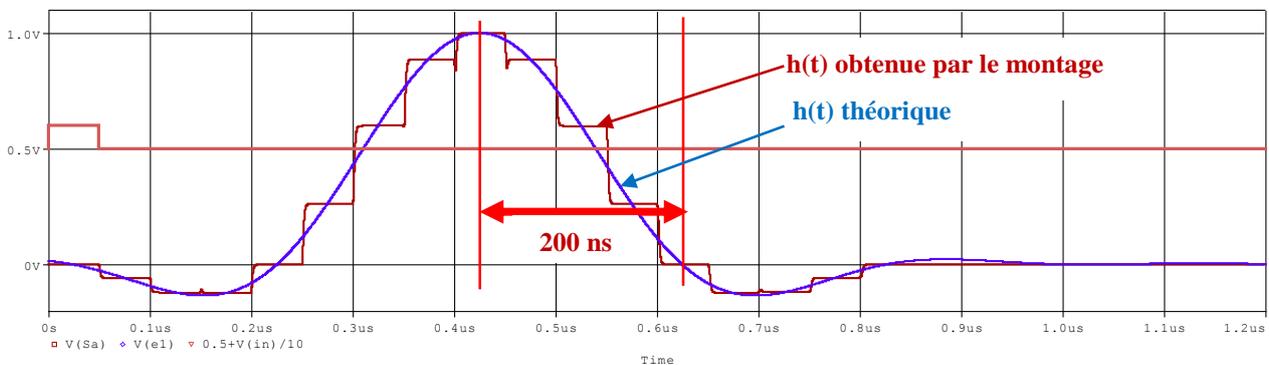


Schéma simulé. En rouge, les numéros ou noms de nœuds

Application numérique : 1 symbole $T_s = 200$ ns d'où $T_e = 50$ ns. (La transmission est à 5 Mbits/s). Un filtre passe bas du premier ordre avec une constante de temps de 1 ns permet de lisser la sortie du sommateur qui présente des valeurs intermédiaires lors de l'établissement de la valeur finale. La sortie est le potentiel S_a .

2.d) Simulation



Signal en sortie du circuit : cosinus surélevé en 15 échantillons (Pour une meilleure visibilité, le signal d'entrée est réduit et décalé verticalement)

Interprétation

A $t = 0$, les sorties des blocs retard sont à 0, et, par conséquence, la sortie globale $V(Sa)$ affiche 0 V.

Une entrée logique au niveau haut (valeur 1 V) est placée durant 50 ns.

Ce signal se propage dans la chaîne de retard, et on lit clairement, de 50 ns à 100 ns, la valeur -57 mV. Puis, par pas de 50 ns, les autres valeurs programmées : -0,12 V, -0,122 V, 0 V etc.

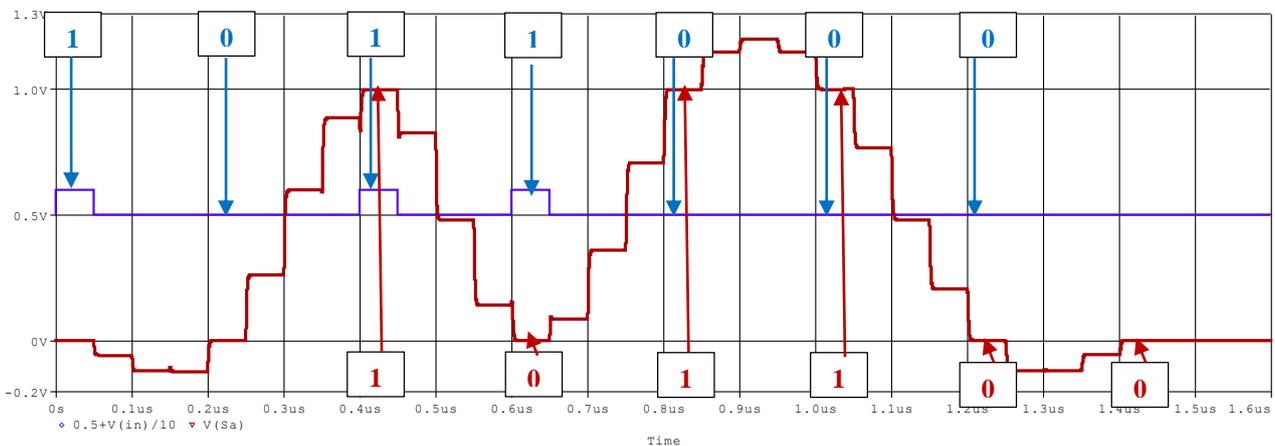
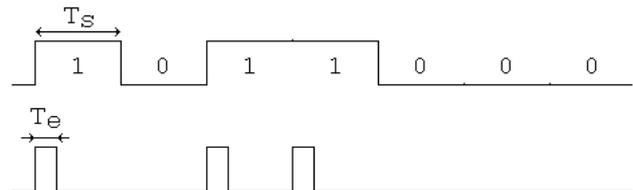
Le meilleur instant pour lire la valeur du symbole est à $t = 425$ ns après le front montant du signal d'entrée. Le temps de traversée du filtre est de 400 ns.

Il a été ajouté, par la simulation, la « vraie » équation de $h(t)$, centrée sur 425 ns. On peut ainsi apprécier le juste positionnement des 15 échantillons.

Raisonnons maintenant sur une transmission de plusieurs bits consécutifs.

Soit par exemple 1 0 1 1 0 0 0 :

Il faut donc injecter en $V(in)$ des signaux de largeur 50 ns tels le chronogramme ci-contre :



Signal en sortie du filtre en cosinus surélevé recevant 1 0 1 1 0 0 0

Interprétation

Le départ des traces est le même que précédemment (transmission de 1 puis 0).

A $t = 400$ ns, le niveau haut se présente en entrée. Comme vu dans l'essai précédent, la valeur du symbole doit être lue 425 ns après, donc à $t = 825$ ns. Effectivement la sortie est exactement à 1 V à cet instant.

A $t = 600$ ns, un nouveau niveau haut se présente en entrée. La sortie affiche à 1 V à $t = 1,025$ μ s.

On remarque très nettement qu'entre ces 2 instants, la sortie passe par 1,149 V et 1,2 V car le sommateur reçoit $(0,887 \text{ V} + 0,262 \text{ V})$ et $(0,6 \text{ V} + 0,6 \text{ V})$. Ces valeurs intermédiaires seront ignorées par le récepteur si l'échantillonnage de l'information est aux multiples de T_s .

De même, les valeurs « 0 » sont bien à 0 V aux temps 625 ns, 1,225 μ s, 1,425 μ s.

Remarque : la présence de plusieurs « 1 » dans le registre à décalage de 15 bascules a pour conséquence une valeur supérieure à 1 V (mais inférieure à 2 V) délivrée par le sommateur. Ramené sur le schéma logique, cela signifie que **le bus doit avoir une taille de N+1 bits**.

3) Deuxième réalisation pratique

La simulation précédente montrait une nette discrétisation, dû au faible nombre d'échantillons pour définir $h(t)$. Rapprochons nous d'une forme d'onde plus réaliste, en fixant 65 échantillons, au lieu de 15.

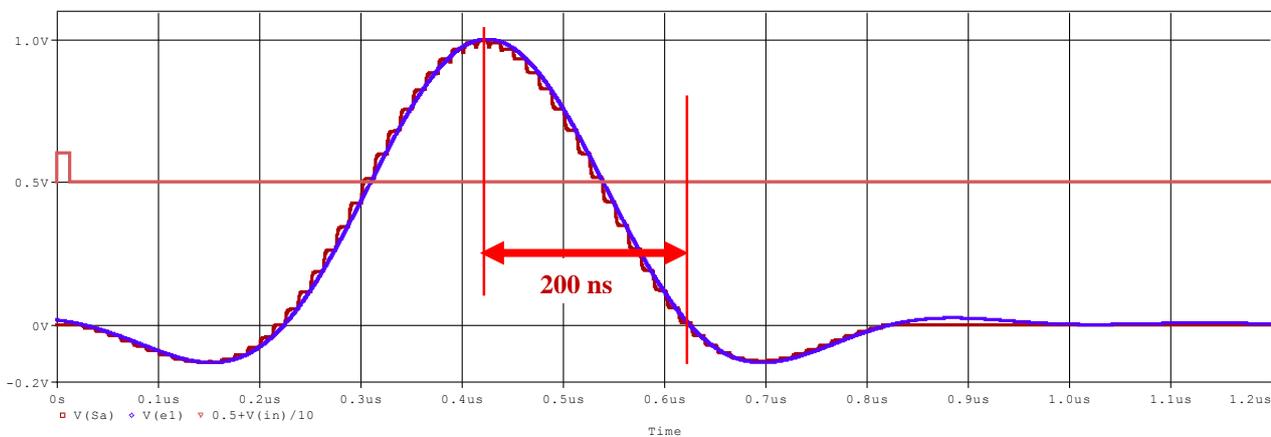
Remarque : un nombre plus élevé n'est pas possible avec la version d'évaluation de Pspice qui est limitée à 75 noeuds dans le schéma simulé.

La structure est donc absolument la même, mais avec 65 blocs retard, et un sommateur avec 65 entrées, coefficienté avec 65 valeurs.

La durée symbole est inchangée : 200 ns, mais le signal d'entrée dure $T_e = 12,5$ ns. Il faut donc 16 échantillons pour un temps symbole.

3.a) Essai préalable : un seul symbole transmis (fichier `cosinus_65.cir`)

On place en $V(in)$ un signal de 1 V qui dure 12,5 ns.



Interprétation

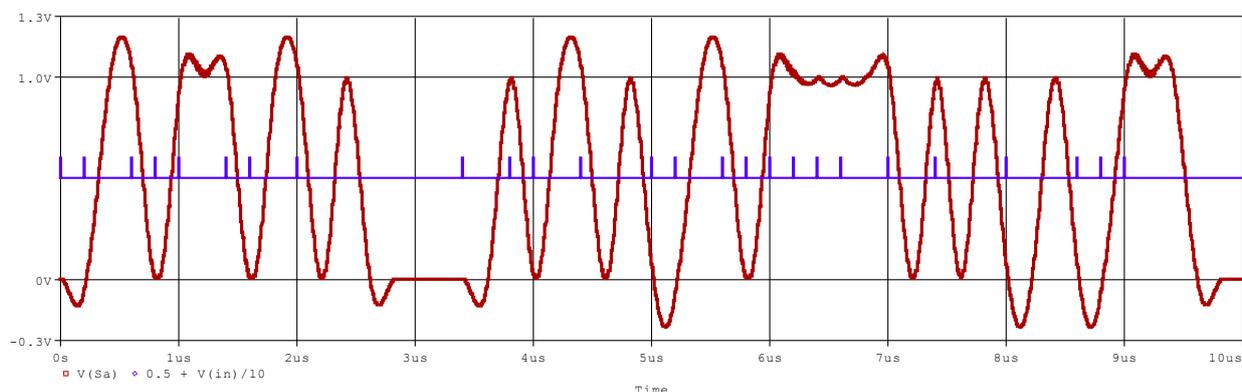
A $t = 0$, $V(Sa)$ affiche 0 V. L'entrée logique au niveau haut (valeur 1 V) est placée durant 12,5 ns.

Ce signal se propage dans la chaîne de retard, et les valeurs données en $V(Sa)$ sont 4 fois plus rapprochées. Par exemple, -0,12 V ; -0,122 V, de l'essai précédent est remplacé par : -0,12 V ; -0,129 V ; -0,133 V ; -0,132 V ; -0,122 V, et ce durant le même temps.

Les 65 échantillons suivent de façon fidèle la « vraie » équation de $h(t)$, centrée sur 425 ns.

3.b) Essai complet : un flot binaire transmis

Observons maintenant l'action de ce filtre attaquée par un flot binaire aléatoire : à chaque symbole de niveau haut, une impulsion de 12,5 ns est placée en $V(in)$. On simule ici durant 10 µs, soit 50 symboles.



*Des impulsions de 12,5 ns se présentent aléatoirement toutes les 200 ns.
La sortie du filtre en cosinus surélevé.*

Interprétation

On reconnaît nettement la succession de motifs, avec pour chacun d'entre eux, l'influence du motif précédent. Mais l'important n'est pas dans les régimes transitoires (les ondulations), mais dans la valeur aux instant multiples de T_s , soit, ici, toutes les 200 ns.

A cette échelle, il est difficile de vérifier ces valeurs. Aussi, il existe un procédé qui consiste à présenter cette courbe avec en ordonnée, une durée symbole : le diagramme de l'oeil (cf article 69 : [document PDF](#)).

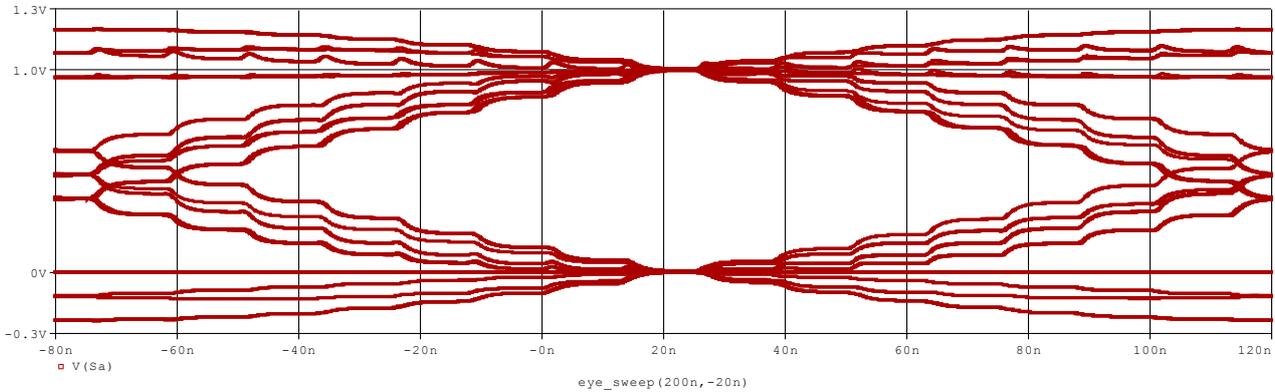


Diagramme de l'oeil

Interprétation

Avec cette échelle des temps, le run est ramené au cumul des 50 traces.

On reconnaît nettement les 16 échantillons que dure un symbole. Les différentes traces se croisent exactement au même instant pour 1 V et pour 0 V. Cela montre que l'instant de décision (pour déterminer si on lit un niveau haut ou niveau bas) doit être placé à ce point judicieux. On comprend aisément qu'un décalage temporel fait prendre le risque au récepteur d'interpréter un niveau haut reçu pour un niveau bas émis et réciproquement, surtout en milieu bruité.

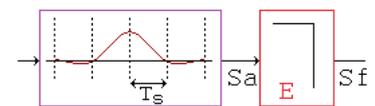
3.c) Prise en compte du canal de transmission.

Le canal est à bande limitée : au-delà d'une certaine fréquence, une atténuation apparaît.

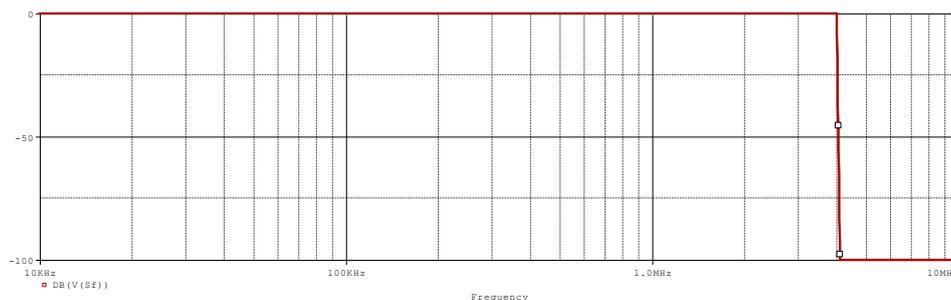
Or, le filtre en cosinus surélevé, à $\alpha = 0,5$, a une bande passante $= \frac{1 + \alpha}{2T_s} = \frac{1 + 0,5}{0,4 \mu s} = 3,75 \text{ MHz}$.

Il faut donc que le canal laisse passer au moins la bande $[0 ; 3,75 \text{ MHz}]$, pour ne pas dégrader la transmission.

Pour illustrer ceci, on ajoute un filtre « parfait » en sortie du filtre en cosinus surélevé pour modéliser le canal. Ainsi on simule la chaîne émission - canal. Pour ce faire, on exploite une source E de Pspice.

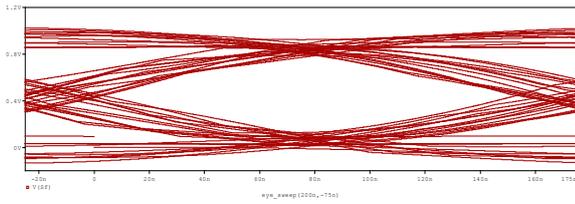


filtre montrant la limitation de bande

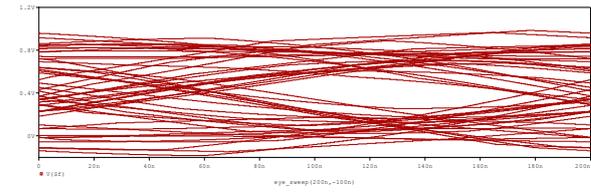


Réponse harmonique du filtre modélisant le canal : Bande passante : 4 MHz bande coupée 4,1 MHz à -100 dB
E_LP Sf 0 FREQ {V(Sa)}=(0,0,0) (4Meg,0,0) (4.1Meg,-100,0)

On essaie 2 réponses harmoniques du canal, de part et d'autre de 3,75 MHz :



Bande passante : 4 MHz
bande coupée : 4,1 MHz à - 100 dB



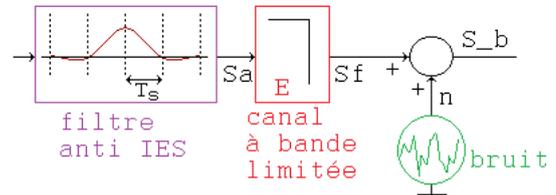
Bande passante : 2 MHz
bande coupée : 2,1 MHz à - 100 dB

Interprétation

- L'aspect discrétisé (les 16 niveaux par symbole) est très atténué par le comportement passe bas du canal.
- Comme pressenti, l'œil est plus fermé à faible bande passante. Une bande de 4 MHz est acceptable, mais à 2 MHz c'est trop critique, le taux d'erreur par bit augmentant.

3.d) Prise en compte du bruit dans le canal de transmission.

Le bruit est inévitable dans une communication. Bien que tous les éléments en soient affectés, c'est essentiellement sur le canal de transmission que le bruit intervient. Par la simulation Pspice, il est facile d'ajouter une source de bruit externe qui vient polluer la transmission.

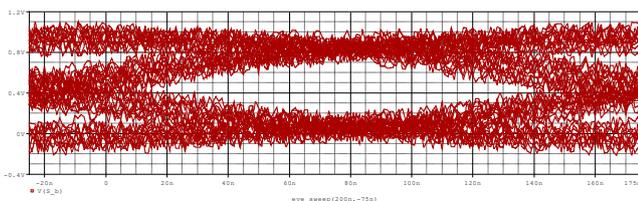


Ajout d'un bruit en fin de transmission

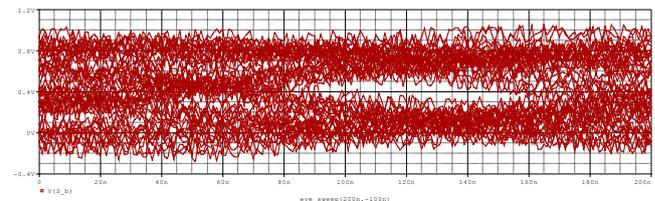
On choisit pour cela une source PWL, qui reproduit un fichier dont les coordonnées sont issues d'un tableau Excel exploitant la fonction ALEA(). Une valeur aléatoire, comprise entre -0,1 V et 0,1 V apparaît toutes les ns. Le spectre en tension passe par 0 à 1 GHz, ce qui donne un bruit blanc dans la bande [0 ; 50 MHz]. Ce signal, extérieur, est sommé avec celui présent en sortie du canal à bande limitée.

Bien entendu, le signal qui arrive au récepteur en est affecté, comme le montre le digramme de l'œil :

On reprend les 2 réponses harmoniques précédentes du canal (même échelle) :



Bande passante : 4 MHz



Bande passante : 2 MHz

Interprétation

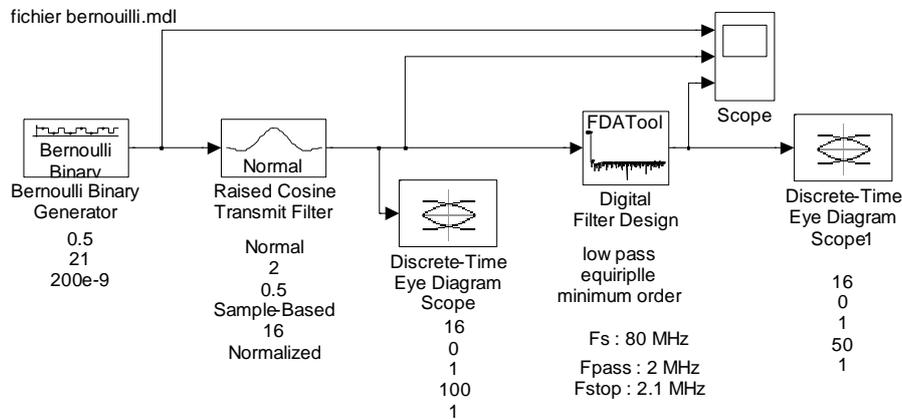
Ces runs illustrent bien l'importance de l'instant de la prise de décision, car la présence de bruit dégrade le discernement des niveaux. Malgré le filtre anti IES, une bande passante de 2 MHz pour un canal de transmission est manifestement déconseillée pour un débit de 5 Mbits/seconde.

Conclusion

Dans la littérature des communications numériques, lorsqu'on aborde le filtre en cosinus surélevé, l'accent est fréquemment porté sur l'aspect mathématique. L'angle de vue « électronique » faite dans ce document permet de comprendre le rôle des différents constituants d'un schéma réel. Par une simulation comportementale, cet article a montré, au travers un cas numérique concret, le rôle et l'intérêt d'un filtre anti IES.

Annexe 1 : Simulation Simulink

Il est proposé ici d'exploiter le logiciel Matlab/Simulink pour simuler une chaîne de transmission numérique. Afin de mieux comparer les simulations Pspice données dans cet article, on reproduit des conditions identiques : flot binaire aléatoire rythmé à $T_s = 200$ ns, même paramétrage du filtre en cosinus surélevé, et visualisation à l'oscilloscope ainsi que le diagramme de l'oeil.



Explication :

Par des commentaires placés sur la page, il a été ajouté le paramétrage des blocs.

Une source aléatoire de 0 et 1 est réalisée par le « Bernoulli Binary Generator ». Le signal est envoyé en voie 1 de l'oscilloscope.

Le bloc « Raised Cosine Transmit Filter » est paramétré en « Normal », ce qui permet de ne pas activer le filtre en racine carrée de cosinus surélevé. Sa sortie est sur la voie 2 de l'oscilloscope.

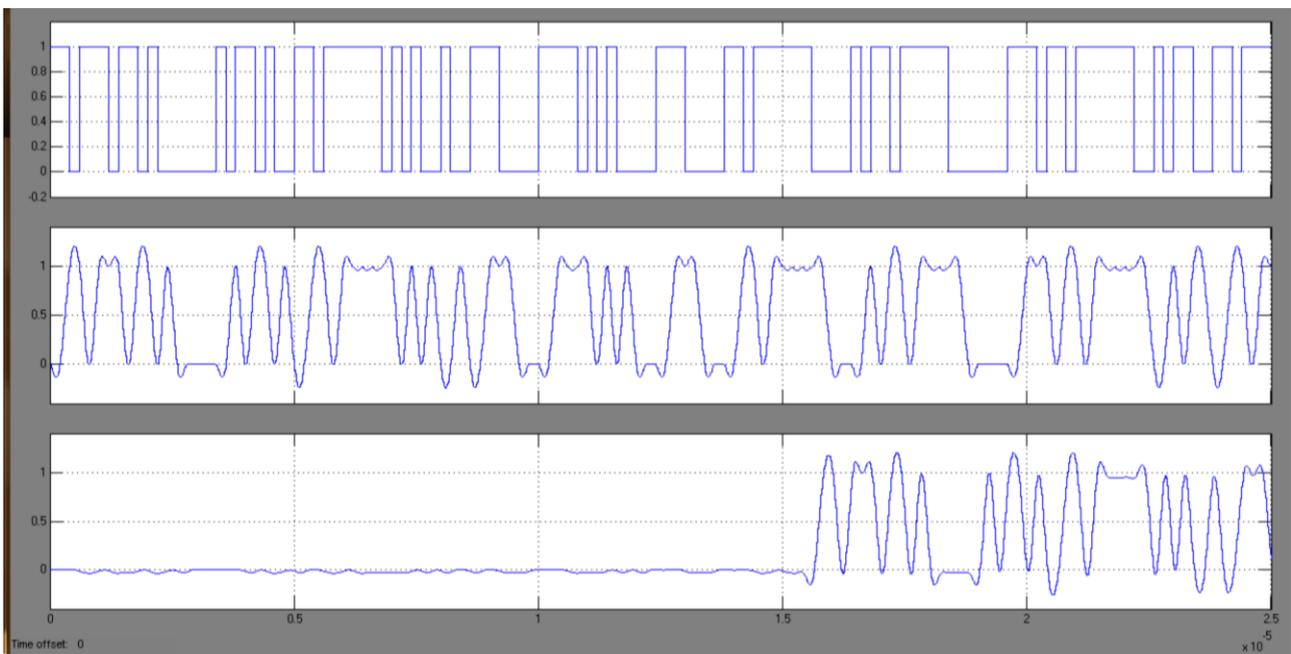
Le canal est modélisé par le bloc « Digital Filter Design ». Ce n'est pas un filtre analogique, mais un filtre numérique.

Son gabarit en fréquence est paramétrable par 2 points : (2 MHz ; 0 dB) et (2,1 MHz ; - 100 dB).

On refait un run avec un autre jeu de valeurs : (4 MHz ; 0 dB) et (4,1 MHz ; - 100 dB).

On simule ici sur 25 μ s, ce qui permet de comparer avec les 10 μ s données en 3.b) :

Chronogrammes :



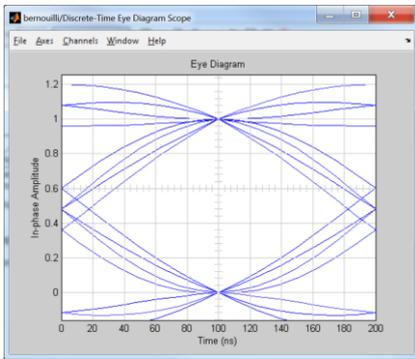
Flot binaire, sortie filtre en cosinus sur élevé, sortie canal coupant à 4 MHz.

Interprétation :

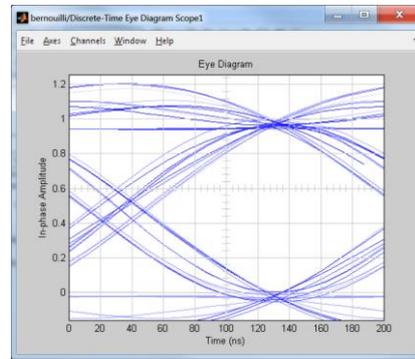
C'est exactement la même séquence binaire. La comparaison en est facilitée. La sortie du filtre en cosinus surélevé est parfaitement identique à celle donnée par Pspice.

La sortie du canal présente un régime transitoire d'environ 15 μ s : c'est lié à l'ordre 2469 (!) de ce filtre numérique. La durée de simulation a été fixée à 25 μ s, de façon à reconnaître les symboles durant 10 μ s.

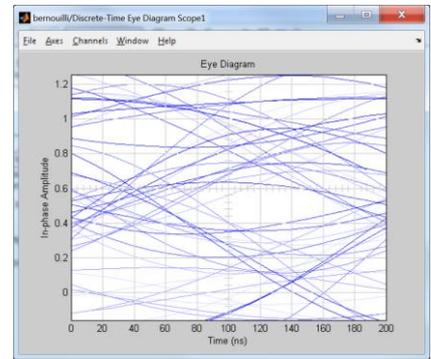
Diagrammes de l'œil :



en sortie du filtre en cosinus surélevé



en sortie du canal, coupant à 4 MHz

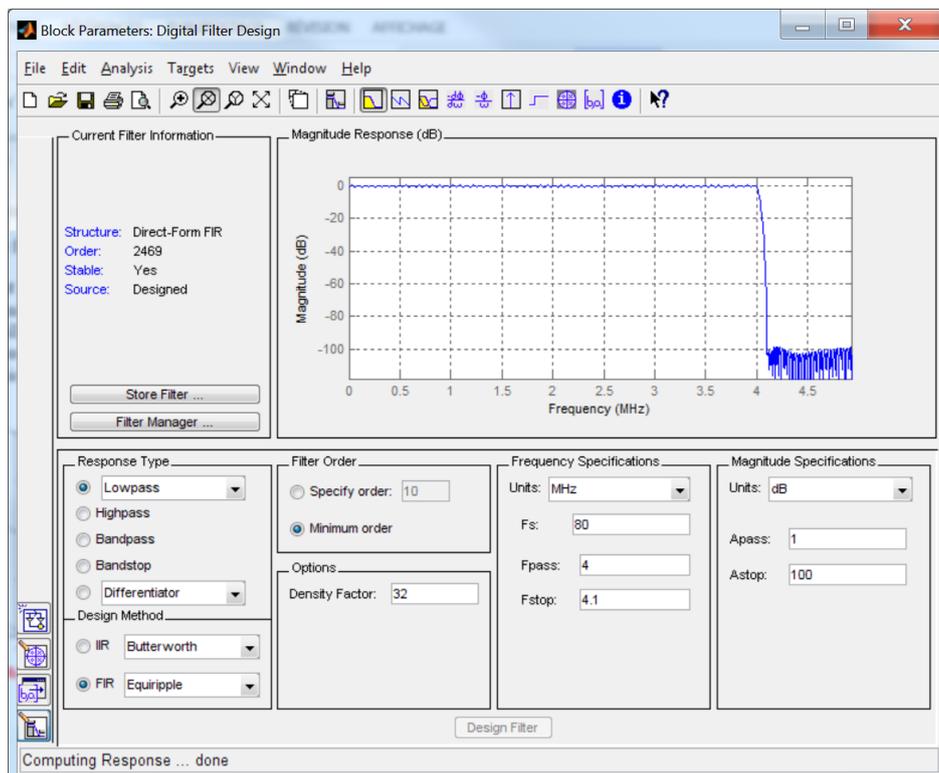


en sortie du canal, coupant à 2 MHz

Interprétation :

En sortie du filtre en cosinus sur élevé, l'œil est bien ouvert, et les trajectoires se croisent en un point. Bien entendu, on ne voit pas le côté discrétisation comme sur les résultats Pspice car le bloc est défini par l'équation et non par morceaux de 12,5 ns.

En sortie du canal, le diagramme de l'œil confirme une dégradation si la bande passante se réduit. Les trajectoires ne sont pas - et ne peuvent pas - être superposables avec celles données par Pspice, car le filtre numérique de Simulink n'a pas la même équation temporelle que celle de Pspice. Pour information, on donne ci-dessous ses caractéristiques :



Annexe 2 : fichiers cosinus_15.cir

```
motif cosinus sureleve en 15 echantillons
* fichier cosinus_15.cir

* bit symbole à transmettre 200 ns
Vinb eb 0 pwl (0,0 200n,0 201n,1 600n,1 601n,0)

* source obtenue par equation sous excel
Vin1 e1 0 PWL (FILE=cos_sur.txt)

* entrée du montage FIR
Vin2 in 0 PWL
+ (0,0 100p,1 49.9n,1 50n,0
+ 400n,0 400.1n,1 450n,1 450.1n,0
+ 600n,0 600.1n,1 650n,1 650.1n,0
)

.param T = 50n
.subckt RETARD in out
Ez out 0 LAPLACE {V(in)} {exp(-s*{T})}
.ends

.param a = 1
.param b = 0.887
.param c = 0.6
.param d = 0.262
.param e = 0
.param f = -0.122
.param g = -0.12
.param h = -0.057

Xdelay1 in 1 RETARD
Xdelay2 1 2 RETARD
Xdelay3 2 3 RETARD
Xdelay4 3 4 RETARD
Xdelay5 4 5 RETARD
Xdelay6 5 6 RETARD
Xdelay7 6 7 RETARD
Xdelay8 7 8 RETARD
Xdelay9 8 9 RETARD
Xdelay10 9 10 RETARD
Xdelay11 10 11 RETARD
Xdelay12 11 12 RETARD
Xdelay13 12 13 RETARD
Xdelay14 13 14 RETARD
Xdelay15 14 15 RETARD

Ea Sn 0 POLY(15) (1,0) (2,0) (3,0) (4,0) (5,0) (6,0) (7,0) (8,0)
+ (9,0) (10,0) (11,0) (12,0) (13,0) (14,0) (15,0)
+ 0 {h} {g} {f} {e} {d} {c} {b} {a} {b} {c} {d} {e} {f} {g} {h}

ESa Sa 0 LAPLACE {V(Sn)} { 1 / (1+1n*s)} ; légère mise en forme par tau=1ns

.TRAN 1n 2u 0u 1n
.PROBE
.END
```

Netliste, prête à simuler

Annexe 3 : fichiers cosinus_65.cir

```
motif cosinus sureleve en 65 echantillons
* fichier cosinus_65.cir

* bit symbole à transmettre 200 ns
*Vinb eb 0 pwl (0,0 200n,0 201n,1 600n,1 601n,0)

* source obtenue par equation sous excel
*Vin1 e1 0 PWL (FILE=cos_sur.txt)

* 50 bits à transmettre
* commenter / décommenter pour choisir les impulsions
Vin2 in 0 PWL
+ (0,0 100p,1 12.5n,1 12.6n,0
+ 200n,0 200.1n,1 212.5n,1 212.6n,0
*+ 400n,0 400.1n,1 412.5n,1 412.6n,0
+ 600n,0 600.1n,1 612.6n,1 612.6n,0
+ 800n,0 800.1n,1 812.6n,1 812.6n,0
+ 1000n,0 1000.1n,1 1012.5n,1 1012.6n,0
*+ 1200n,0 1200.1n,1 1212.5n,1 1212.6n,0
+ 1400n,0 1400.1n,1 1412.5n,1 1412.6n,0
+ 1600n,0 1600.1n,1 1612.5n,1 1612.6n,0
*+ 1800n,0 1800.1n,1 1812.5n,1 1812.6n,0
+ 2000n,0 2000.1n,1 2012.5n,1 2012.6n,0
*+ 2200n,0 2200.1n,1 2212.5n,1 2212.6n,0
*+ 2400n,0 2400.1n,1 2412.5n,1 2412.6n,0
*+ 2600n,0 2600.1n,1 2612.5n,1 2612.6n,0
*+ 3000n,0 3000.1n,1 3012.5n,1 3012.6n,0
*+ 3200n,0 3200.1n,1 3212.5n,1 3212.6n,0
+ 3400n,0 3400.1n,1 3412.5n,1 3412.6n,0
*+ 3600n,0 3600.1n,1 3612.5n,1 3612.6n,0
+ 3800n,0 3800.1n,1 3812.5n,1 3812.6n,0
+ 4000n,0 4000.1n,1 4012.5n,1 4012.6n,0
* 4200n,0 4200.1n,1 4212.5n,1 4212.6n,0
+ 4400n,0 4400.1n,1 4412.5n,1 4412.6n,0
*+ 4600n,0 4600.1n,1 4612.5n,1 4612.6n,0
*+ 4800n,0 4800.1n,1 4812.5n,1 4812.6n,0
+ 5000n,0 5000.1n,1 5012.5n,1 5012.6n,0
+ 5200n,0 5200.1n,1 5212.5n,1 5212.6n,0
*+ 5400n,0 5400.1n,1 5412.5n,1 5412.6n,0
+ 5600n,0 5600.1n,1 5612.6n,1 5612.6n,0
+ 5800n,0 5800.1n,1 5812.6n,1 5812.6n,0
+ 6000n,0 6000.1n,1 6012.5n,1 6012.6n,0
+ 6200n,0 6200.1n,1 6212.5n,1 6212.6n,0
+ 6400n,0 6400.1n,1 6412.5n,1 6412.6n,0
+ 6600n,0 6600.1n,1 6612.6n,1 6612.6n,0
*+ 6800n,0 6800.1n,1 6812.6n,1 6812.6n,0
+ 7000n,0 7000.1n,1 7012.5n,1 7012.6n,0
*+ 7200n,0 7200.1n,1 7212.5n,1 7212.6n,0
+ 7400n,0 7400.1n,1 7412.5n,1 7412.6n,0
*+ 7600n,0 7600.1n,1 7612.6n,1 7612.6n,0
*+ 7800n,0 7800.1n,1 7812.6n,1 7812.6n,0
+ 8000n,0 8000.1n,1 8012.5n,1 8012.6n,0
*+ 8200n,0 8200.1n,1 8212.5n,1 8212.6n,0
*+ 8400n,0 8400.1n,1 8412.5n,1 8412.6n,0
+ 8600n,0 8600.1n,1 8612.6n,1 8612.6n,0
+ 8800n,0 8800.1n,1 8812.6n,1 8812.6n,0
+ 9000n,0 9000.1n,1 9012.5n,1 9012.6n,0
*+ 9200n,0 9200.1n,1 9212.5n,1 9212.6n,0
*+ 9400n,0 9400.1n,1 9412.5n,1 9412.6n,0
*+ 9600n,0 9600.1n,1 9612.6n,1 9612.6n,0
*+ 9800n,0 9800.1n,1 9812.6n,1 9812.6n,0
+ 10000n,0 10000.1n,1 10012.5n,1 10012.6n,0
```

```
.param T = 12.5n

.subckt RETARD in out
Ez out 0 LAPLACE {V(in)} {exp(-s*{T})}
.ends
```

```
.param c1 = 0
.param c2 = -0.0116
.param c3 = -0.0253
.param c4 = -0.0409
.param c5 = -0.0576
.param c6 = -0.0749
.param c7 = -0.0917
.param c8 = -0.1072
.param c9 = -0.1200
.param c10 = -0.1292
.param c11 = -0.1334
.param c12 = -0.1315
.param c13 = -0.1225
.param c14 = -0.1054
.param c15 = -0.0795
.param c16 = -0.0444
.param c17 = 0
```

```
.param c18 = 0.0536
.param c19 = 0.1159
.param c20 = 0.1859
.param c21 = 0.2625
.param c22 = 0.3441
.param c23 = 0.4290
.param c24 = 0.5151
.param c25 = 0.6002
.param c26 = 0.6822
.param c27 = 0.7587
.param c28 = 0.8278
.param c29 = 0.8872
.param c30 = 0.9354
.param c31 = 0.9709
.param c32 = 0.9927
```

```
.param c33 = 1
```

```
Xdelay1 in 1 RETARD
Xdelay2 1 2 RETARD
Xdelay3 2 3 RETARD
Xdelay4 3 4 RETARD
Xdelay5 4 5 RETARD
Xdelay6 5 6 RETARD
Xdelay7 6 7 RETARD
Xdelay8 7 8 RETARD
Xdelay9 8 9 RETARD
Xdelay10 9 10 RETARD
Xdelay11 10 11 RETARD
Xdelay12 11 12 RETARD
Xdelay13 12 13 RETARD
Xdelay14 13 14 RETARD
Xdelay15 14 15 RETARD
Xdelay16 15 16 RETARD
Xdelay17 16 17 RETARD
Xdelay18 17 18 RETARD
Xdelay19 18 19 RETARD
Xdelay20 19 20 RETARD
Xdelay21 20 21 RETARD
Xdelay22 21 22 RETARD
Xdelay23 22 23 RETARD
Xdelay24 23 24 RETARD
Xdelay25 24 25 RETARD
```

```
.param c34 = 0.9927
.param c35 = 0.9709
.param c36 = 0.9354
.param c37 = 0.8872
.param c38 = 0.8278
.param c39 = 0.7587
.param c40 = 0.6822
.param c41 = 0.6002
.param c42 = 0.5151
.param c43 = 0.4290
.param c44 = 0.3441
.param c45 = 0.2625
.param c46 = 0.1859
.param c47 = 0.1159
.param c48 = 0.0536
```

```
.param c49 = 0
```

```
.param c50 = -0.0444
.param c51 = -0.0795
.param c52 = -0.1054
.param c53 = -0.1225
.param c54 = -0.1315
.param c55 = -0.1334
.param c56 = -0.1292
.param c57 = -0.1200
.param c58 = -0.1072
.param c59 = -0.0917
.param c60 = -0.0749
.param c61 = -0.0576
.param c62 = -0.0409
.param c63 = -0.0253
.param c64 = -0.0116
.param c65 = 0
```

```
Xdelay33 32 33 RETARD
Xdelay34 33 34 RETARD
Xdelay35 34 35 RETARD
Xdelay36 35 36 RETARD
Xdelay37 36 37 RETARD
Xdelay38 37 38 RETARD
Xdelay39 38 39 RETARD
Xdelay40 39 40 RETARD
Xdelay41 40 41 RETARD
Xdelay42 41 42 RETARD
Xdelay43 42 43 RETARD
Xdelay44 43 44 RETARD
Xdelay45 44 45 RETARD
Xdelay46 45 46 RETARD
Xdelay47 46 47 RETARD
Xdelay48 47 48 RETARD
Xdelay49 48 49 RETARD
Xdelay50 49 50 RETARD
Xdelay51 50 51 RETARD
Xdelay52 51 52 RETARD
Xdelay53 52 53 RETARD
Xdelay54 53 54 RETARD
Xdelay55 54 55 RETARD
Xdelay56 55 56 RETARD
Xdelay57 56 57 RETARD
```

Xdelay26 25 26 RETARD	Xdelay58 57 58 RETARD
Xdelay27 26 27 RETARD	Xdelay59 58 59 RETARD
Xdelay28 27 28 RETARD	Xdelay60 59 60 RETARD
Xdelay29 28 29 RETARD	Xdelay61 60 61 RETARD
Xdelay30 29 30 RETARD	Xdelay62 61 62 RETARD
Xdelay31 30 31 RETARD	Xdelay63 62 63 RETARD
Xdelay32 31 32 RETARD	Xdelay64 63 64 RETARD
	Xdelay65 64 65 RETARD

```

Ea2 Si1 0 POLY(32)
+ (1,0) (2,0) (3,0) (4,0) (5,0) (6,0) (7,0) (8,0) (9,0) (10,0) (11,0) (12,0) (13,0)
(14,0) (15,0) (16,0)
+ (17,0) (18,0) (19,0) (20,0) (21,0) (22,0) (23,0) (24,0) (25,0) (26,0) (27,0)
(28,0) (29,0) (30,0) (31,0) (32,0)
+ 0
+ {c1} {c2} {c3} {c4} {c5} {c6} {c7} {c8} {c9} {c10} {c11} {c12} {c13} {c14} {c15}
{c16}
+ {c17} {c18} {c19} {c20} {c21} {c22} {c23} {c24} {c25} {c26} {c27} {c28} {c29}
{c30} {c31} {c32}

Ea3 Si2 0 POLY(33)
+ (33,0) (34,0) (35,0) (36,0) (37,0) (38,0) (39,0) (40,0) (41,0) (42,0) (43,0)
(44,0) (45,0) (46,0) (47,0) (48,0)
+ (49,0) (50,0) (51,0) (52,0) (53,0) (54,0) (55,0) (56,0) (57,0) (58,0) (59,0)
(60,0) (61,0) (62,0) (63,0) (64,0) (65,0)
+ 0
+ {c33} {c34} {c35} {c36} {c37} {c38} {c39} {c40} {c41} {c42} {c43} {c44} {c45}
{c46} {c47} {c48}
+ {c49} {c50} {c51} {c52} {c53} {c54} {c55} {c56} {c57} {c58} {c59} {c60} {c61}
{c62} {c63} {c64} {c65}

Ea4 Sn 0 POLY(2) (Si1,0) (Si2,0) 0 1 1
Ea Sa 0 LAPLACE {V(Sn)} { 1 / (1+2n*s)} ; légère mise en forme par tau=2ns

E_LP Sf 0 FREQ {V(Sa)}=(0,0,0) (2Meg,0,0) (2.1Meg,-100,0) ; canal

*Vbruit n 0 PWL ( FILE=source_bruit.txt )
*Esom S_b 0 poly(2) (Sf,0) (n,0) 0 1 0.1 ; sortie canal avec bruit

*Vtestac Sa 0 AC=1
*.AC DEC 1000 10k 10Meg

.TRAN 10n 10u 0 10n ;

.PROBE
.END

```

Netliste, prête à simuler

articles 1 à 43 : sur le livre

Tableau récapitulatif des articles PDF disponibles sur ce site

n°	titre	lien présentation	lien direct article
	Guide d'installation et d'emploi simplifié	présentation	document PDF
44	Exemples basiques et des exercices...	présentation	document PDF
45	Un exemple de circuit passif	présentation	document PDF
46	Un oscillateur Colpitts	présentation	document PDF
47	Compensation en fréquence des amplificateurs opérationnels	présentation	document PDF
48	Un amplificateur à transistors bipolaires	présentation	document PDF
49	Une bascule D Flip Flop CMOS	présentation	document PDF
50	Une porte XOR à transistors MOS	présentation	document PDF
51	Un VCO à 12 transistors MOS	présentation	document PDF
52	Une PLL à moins de 20 transistors MOS	présentation	document PDF
53	Un oscillateur à résistance négative	présentation	document PDF
54	Une charge électronique	présentation	document PDF
55	Un amplificateur en classe C	présentation	document PDF
56	Le monostable 74 123	présentation	document PDF
57	Un amplificateur en classe D	présentation	document PDF
58	Le transformateur en linéaire	présentation	document PDF
59	La loi d'ohm thermique	présentation	document PDF
60	Le transformateur en non linéaire	présentation	document PDF
61	Robustesse d'un oscillateur en anneau	présentation	document PDF
62	Une alimentation stabilisée	présentation	document PDF
63	Modélisation d'un haut-parleur	présentation	document PDF
64	Un synthétiseur de fréquence	présentation	document PDF
65	Un ampli audio de Sparkfun	présentation	document PDF
66	Simulation logique et analogique	présentation	document PDF
67	Un oscillateur à relaxation	présentation	document PDF
68	Lecteur de TAG RFID 125 kHz	présentation	document PDF
69	Diagramme de l'œil avec Pspice	présentation	document PDF
70	Un amplificateur hautes fréquences	présentation	document PDF
71	Une bizarrerie enfin expliquée...	présentation	document PDF
72	Comprendre le paramétrage de la FFT	présentation	document PDF
73	La relation de Bennett	présentation	document PDF
74	Simuler un circuit à plus de 20 transistors avec PSpice Eval	présentation	document PDF
75	Une horloge biphase sans recouvrement	présentation	document PDF
76	Quelques simulations sur la diode	présentation	document PDF
77	Un ampli classe A, avec transformateur de sortie	présentation	document PDF
78	Des stimuli pour PSpice	présentation	document PDF
79	Simuler le TL431 : zener ajustable	présentation	document PDF
80	Un ADC flash	présentation	document PDF
81	Une chaine d'acquisition : S&H, ADC, DAC	présentation	document PDF
82	Un amplificateur 50 MHz	présentation	document PDF
83	Un dérivateur non inverseur	présentation	document PDF
84	Un amplificateur bipolaire avec push pull CMOS	présentation	document PDF
85	Rôle des répéteurs logiques dans un circuit intégré	présentation	document PDF
86	Un driver logique CMOS pour charge 50 ohms	présentation	document PDF
87	Des triggers de Schmitt et des applications	présentation	document PDF
88	Un filtre gaussien analogique	présentation	document PDF
89	Un générateur de bruit rose	présentation	document PDF

90	Un anémomètre à fil chaud : simulation comportementale	présentation	document PDF
91	Un oscillateur à pont de Wien stabilisé par CTN	présentation	document PDF
92	L'emballement thermique d'une diode	présentation	document PDF
93	Les puissances dans un amplificateur	présentation	document PDF
94	Asservissement de puissance dans une résistance	présentation	document PDF
95	Asservissement de la puissance émise par une antenne radio	présentation	document PDF
96	Un driver de LED de puissance	présentation	document PDF
97	Exploiter Pspice pour simuler des filtres numériques	présentation	document PDF
98	Un filtre en cosinus surélevé avec Pspice	présentation	document PDF
99	Effet de la température sur un amplificateur en classe A	présentation	document PDF
100	Un amplificateur à transistors JFET et bipolaires	présentation	document PDF
Supplément, hors article :			
mon cours « Electronique pour les communications numériques », polycopié couleur 201 pages en pdf			

[retour à l'écran d'accueil de ce site](#)