

*FFT*

*Transformée de Fourier Rapide*

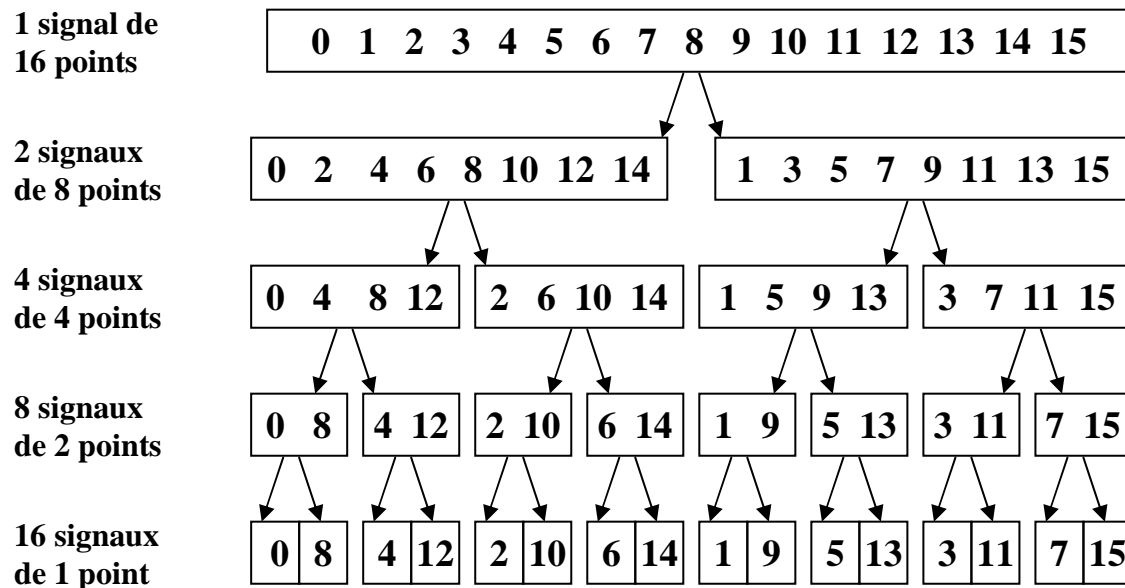
**Cours DSP**

# Principe de la FFT

**La FFT utilise le formalisme de la TFD complexe.**


**Méthode de J.W.Cooley et J.W.Tuckey (1965)**

**1<sup>ère</sup> étape :** Décompositions par alternance du signal de  $N = 2^m$  points dans le domaine temporel en  $N$  signaux de 1 point. A chaque fois la suite des échantillons pairs forment un nouveau signal et la suite des échantillons impairs forme l'autre nouveau signal. Il y a  $\log_2 N$  décompositions successives en tout (16→4, 512→7, 4096→12 etc...)



# Principe de la FFT

**1<sup>ère</sup> étape (suite) : Une autre façon de voir la décomposition qui revient à réordonner les échantillons en fonction d'un index dont les bits sont inversé par rapport à l'index de départ.**

Sample numbers in normal order			Sample numbers after bit reversal	
<i>Decimal</i>	<i>Binary</i>		<i>Decimal</i>	<i>Binary</i>
0	0000		0	0000
1	0001		8	1000
2	0010		4	0100
3	0011		12	1100
4	0100		2	0010
5	0101		10	1010
6	0110		6	0100
7	0111		14	1110
8	1000		1	0001
9	1001		9	1001
10	1010		5	0101
11	1011		13	1101
12	1100		3	0011
13	1101		11	1011
14	1110		7	0111
15	1111		15	1111

# Principe de la FFT

**2<sup>ème</sup> étape : Trouver le spectre TF de chaque signal de 1 point.**

**Cette étape est triviale: le spectre d'un signal de 1 point est égal à lui-même.**

**Démonstration tout aussi triviale :**

$$X[k] = \sum_{i=0}^{N-1} x[i] e^{-2\pi ki/N}$$

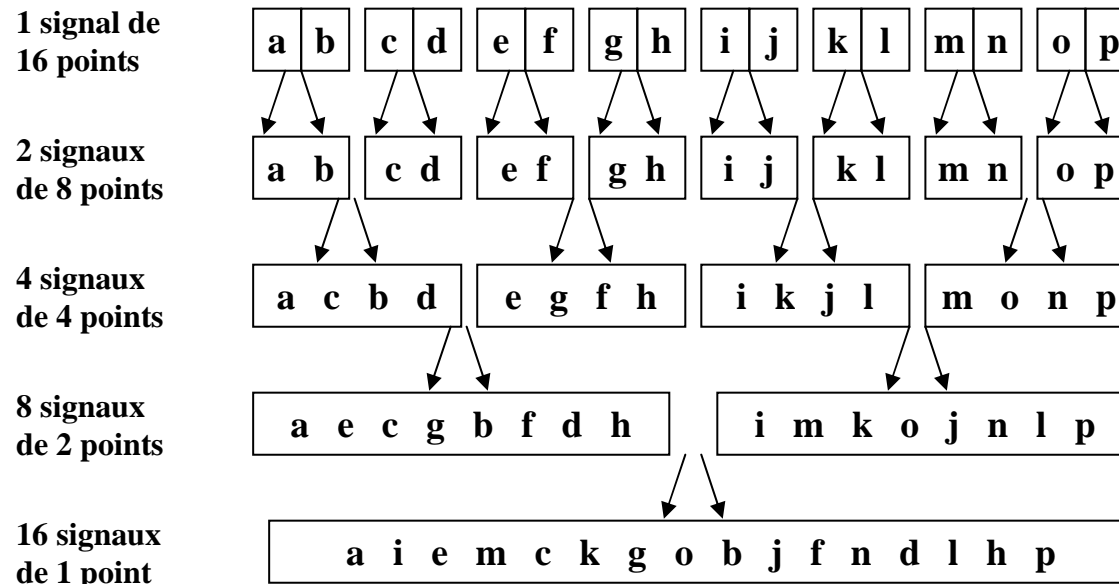
$$\text{pour } N = 1, X[0] = x[0]$$

**Cette étape se réduit donc à ne rien faire !**

# Principe de la FFT

**3<sup>ème</sup> et dernière étape :** Combiner les N spectres de fréquence à 1 point chacun de façon à « défaire » la décomposition dans le domaine temporel de la première étape.

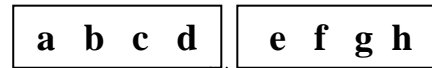
**Il faut donc faire dans le domaine fréquentiel la manipulation correspondant à la suivante dans le domaine temporel:**



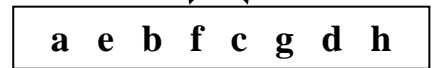
# Principe de la FFT

**3<sup>ème</sup> et dernière étape (suite) : Prenons l'un des étages de cette recombinaison :**

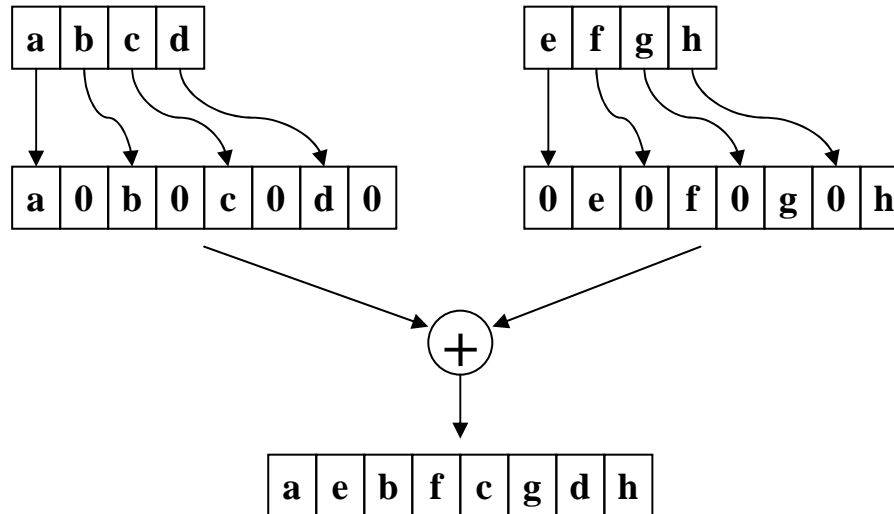
4 signaux  
de 4 points



8 signaux  
de 2 points

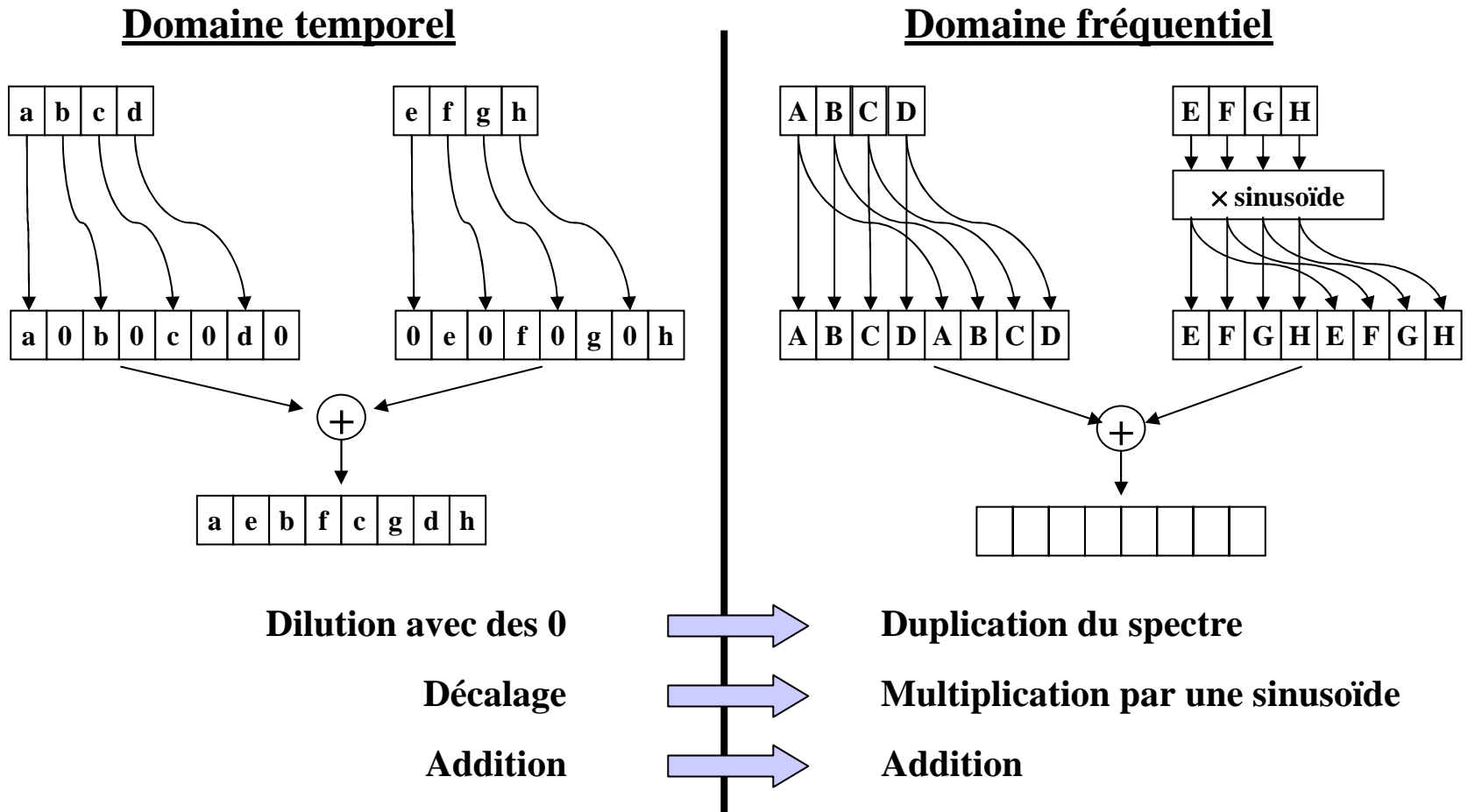


**Cette recombinaison dans le domaine temporel peut être faite de la façon suivante, par dilution des deux signaux à combiner avec des 0, décalage et addition :**



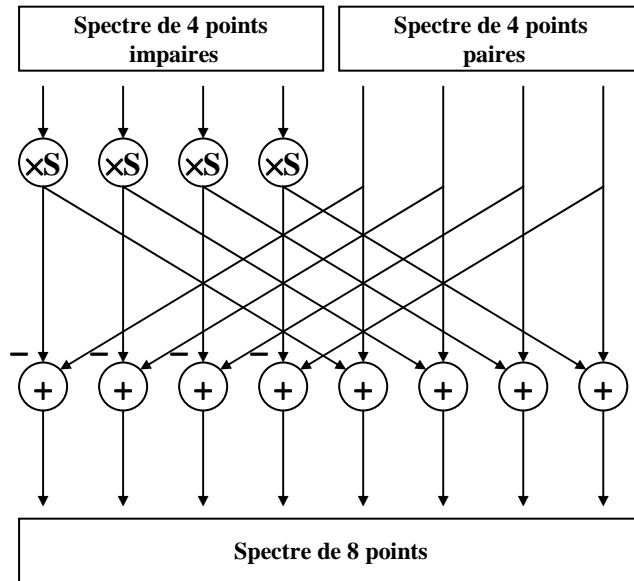
# Principe de la FFT

3<sup>ème</sup> et dernière étape (suite) : La correspondance temporel-fréquentiel de cette suite d'opération est la suivante :

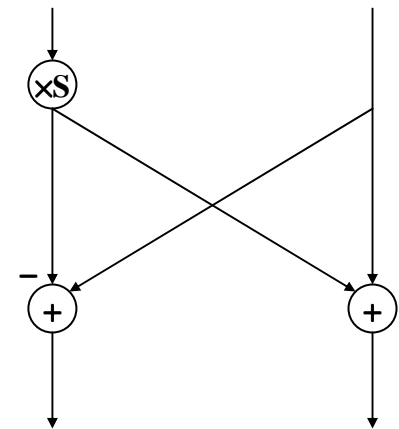


# Principe de la FFT

**3<sup>ème</sup> et dernière étape (suite) :** Le diagramme en bloc de la dernière étape ou étape de synthèse est donc:



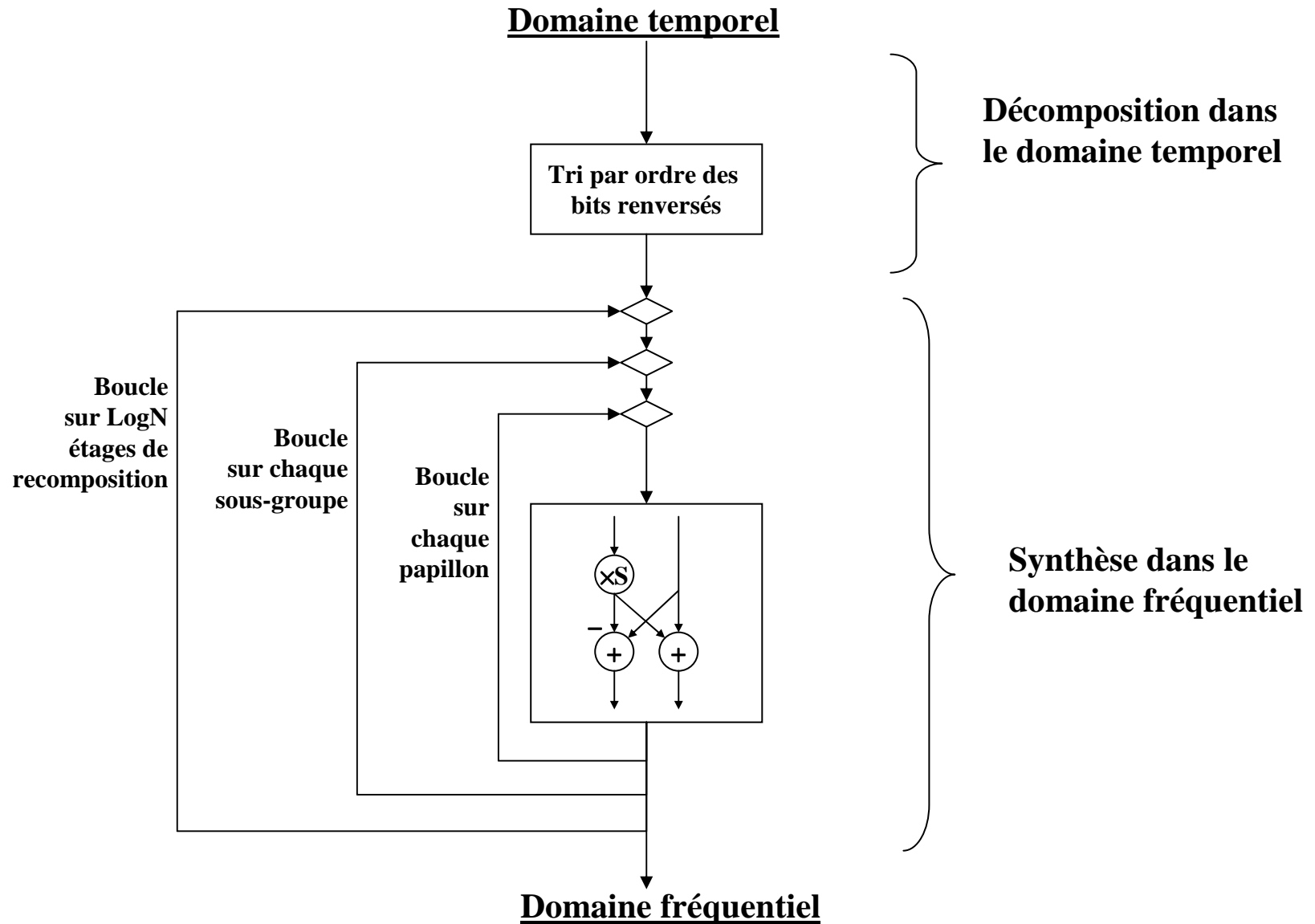
**L'opérateur de base se réduisant à:**





# Principe de la FFT

Diagramme entier de l'algorithme FFT :



# Programmes FFT

## Programme TFD complexe par corrélation :

```
5000 'COMPLEX DFT BY CORRELATION
5010 'Upon entry, N% contains the number of points in the DFT, and
5020 'XR[ ] and XI[ ] contain the real and imaginary parts of the time domain.
5030 'Upon return, REX[ ] and IMX[ ] contain the frequency domain data.
5040 'All signals run from 0 to N%-1.
5050 '
5060 PI = 3.14159265                                'Set constants
5070 '
5080 FOR K% = 0 TO N%-1                              'Zero REX[ ] and IMX[ ], so they can be used
5090 REX[K%] = 0                                     'as accumulators during the correlation
5100 IMX[K%] = 0
5110 NEXT K%
5120 '
5130 FOR K% = 0 TO N%-1                              'Loop for each value in frequency domain
5140 FOR I% = 0 TO N%-1                              'Correlate with the complex sinusoid, SR & SI
5150 '
5160 SR = COS(2*PI*K%*I%/N%)                          'Calculate complex sinusoid
5170 SI = -SIN(2*PI*K%*I%/N%)
5180 REX[K%] = REX[K%] + XR[I%]*SR - XI[I%]*SI
5190 IMX[K%] = IMX[K%] + XR[I%]*SI + XI[I%]*SR
5200 '
5210 NEXT I%
5220 NEXT K%
5230 '
5240 RETURN
```

# Programmes FFT

## Programme FFT en FORTRAN :

```
      SUBROUTINE FFT(X,M)
      COMPLEX X(4096),U,S,T
      PI=3.14159265
      N=2**M
      DO 20 L=1,M
      LE=2**(M+1-L)
      LE2=LE/2
      U=(1.0,0.0)
      S=CMPLX(COS(PI/FLOAT(LE2)),SIN(PI/FLOAT(LE2)))
      DO 20 J=1,LE2
      DO 10 I=J,N,LE
      IP=I+LE2
      T=X(I)+X(IP)
      X(IP)=(X(I)-X(IP))*U
10     X(I)=T
20     U=U*S
      ND2=N/2
      NM1=N-1
      J=1
      DO 50 I=1,NM1
      IF(I.GE.J) GO TO 30
      T=X(J)
      X(J)=X(I)
      X(I)=T
30     K=ND2
40     IF(K.GE.J) GO TO 50
      J=J-K
      K=K/2
      GO TO 40
50     J=J+K
      RETURN
      END
```

# Programmes FFT

## Programme FFT en BASIC :

```
1000 'THE FAST FOURIER TRANSFORM
1010 'Upon entry, N% contains the number of points in the DFT,
1020 'REX[ ] and IMX[ ] contain the real and imaginary parts of the input.
1030 'Upon return, REX[ ] and IMX[ ] contain the DFT output.
1040 'All signals run from 0 to N%-1.
1050 PI = 3.14159265                'Set constants
1060 NM1% = N%-1
1070 ND2% = N%/2
1080 M% = CINT(LOG(N%)/LOG(2))
1090 J% = ND2%
1100 '
1110 FOR I% = 1 TO N%-2              'Bit reversal sorting
1120 IF I% >= J% THEN GOTO 1190
1130 TR = REX[J%]
1140 TI = IMX[J%]
1150 REX[J%] = REX[I%]
1160 IMX[J%] = IMX[I%]
1170 REX[I%] = TR
1180 IMX[I%] = TI
1190 K% = ND2%
1200 IF K% > J% THEN GOTO 1240
1210 J% = J%-K%
1220 K% = K%/2
1230 GOTO 1200
1240 J% = J%+K%
1250 NEXT I%
1260 '
1270 FOR L% = 1 TO M%                'Loop for each stage
1280 LE% = CINT(2^L%)
1290 LE2% = LE%/2
1300 UR = 1
1310 UI = 0
1320 SR = COS(PI/LE2%)              'Calculate sine & cosine values
1330 SI = -SIN(PI/LE2%)
1340 FOR J% = 1 TO LE2%            'Loop for each sub DFT
1350 JM1% = J%-1
1360 FOR I% = JM1% TO NM1% STEP LE% 'Loop for each butterfly
1370 IP% = I%+LE2%
1380 TR = REX[IP%]*UR - IMX[IP%]*UI 'Butterfly calculation
1390 TI = REX[IP%]*UI + IMX[IP%]*UR
1400 REX[IP%] = REX[I%]-TR
1410 IMX[IP%] = IMX[I%]-TI
1420 REX[I%] = REX[I%]+TR
1430 IMX[I%] = IMX[I%]+TI
1440 NEXT I%
1450 TR = UR
1460 UR = TR*SR - UI*SI
1470 UI = TR*SI + UI*SR
1480 NEXT J%
1490 NEXT L%
1500 '
1510 RETURN
```

# Programmes FFT

## Programme FFT inverse utilisant la FFT précédente :

```
2000 'INVERSE FAST FOURIER TRANSFORM SUBROUTINE
2010 'Upon entry, N% contains the number of points in the IDFT, REX[ ] and
2020 'IMX[ ] contain the real and imaginary parts of the complex frequency domain.
2030 'Upon return, REX[ ] and IMX[ ] contain the complex time domain.
2040 'All signals run from 0 to N%-1.
2050 '
2060 FOR K% = 0 TO N%-1                                'Change the sign of IMX[ ]
2070 IMX[K%] = -IMX[K%]
2080 NEXT K%
2090 '
2100 GOSUB 1000                                        'Calculate forward FFT
2110 '
2120 FOR I% = 0 TO N%-1                                'Divide the time domain by N% and
2130 REX[I%] = REX[I%]/N%                               'change the sign of IMX[ ]
2140 IMX[I%] = -IMX[I%]/N%
2150 NEXT I%
2160 '
2170 RETURN
```

# Rapidité

Temps d'exécution pour une TFD complexe  
par la méthode de corrélation :

$$T = k_{TFD} N^2$$

	$k_{TFD}$
sin et cos calculés dans la boucle	25 $\mu$ s
sin et cos précalculés	7 $\mu$ s

(temps pour un  
Pentium 100MHz)

soit plus de 25 (ou 7) secondes pour une TFD sur 1024 points,  
ou près de 25 (ou 7) ms par points, c'est lent !

Temps d'exécution pour une FFT :

$$T = k_{FFT} N \log_2 N$$

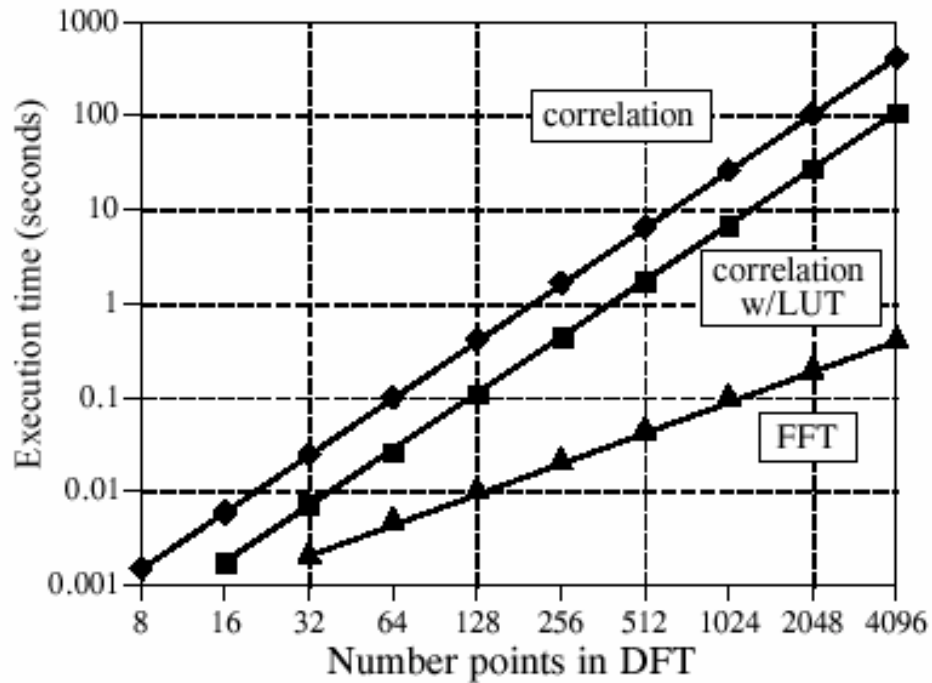
$$k_{TFD} \approx 7\mu\text{s} \quad (\text{sur un Pentium 100MHz})$$

soit environ 70 ms pour une TFD sur 1024 points,  
ou près de 300 fois plus vite que la TFD classique.

Nombre de points	16	32	64	128	256	512	1024	...	4096	...	$2^{20} = 1048576$
$\frac{T_{TFD}}{T_{FFT}} \approx \frac{N}{\log_2 N}$	4	6,4	10,7	18,3	32	56,9	102,4	...	341,3	...	52428

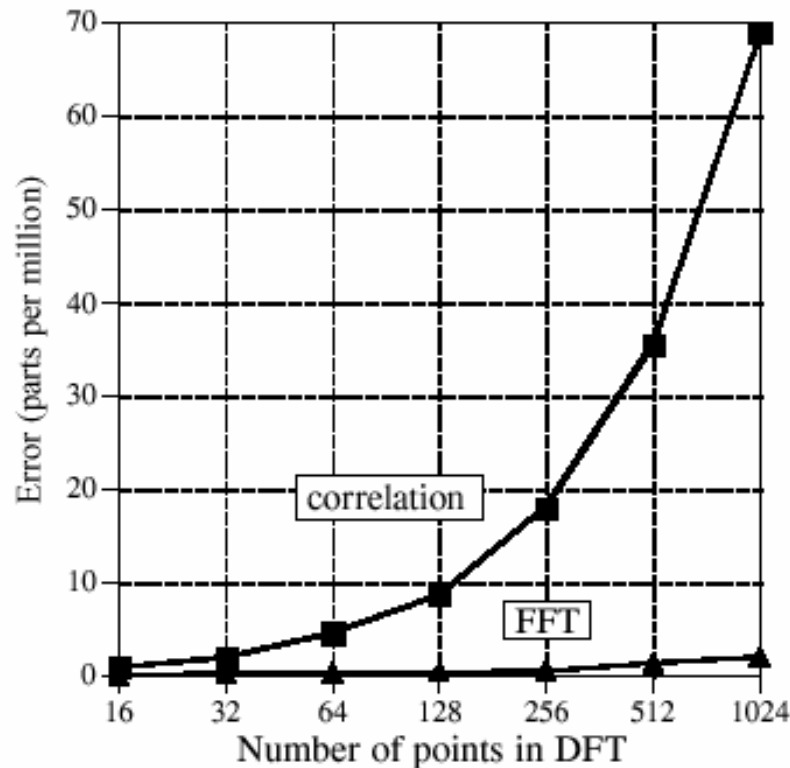
# Rapidité

Temps d'exécution pour une FFT  
comparé à une TFD complexe par la méthode de corrélation :



# Précision

La précision est en général bien meilleure pour la FFT que pour la TFD par corrélation, car toutes choses égales par ailleurs, la FFT requiert beaucoup moins d'opérations que la TFD et souffre beaucoup moins des erreurs d'arrondi et de précision du calcul en virgule flottante.

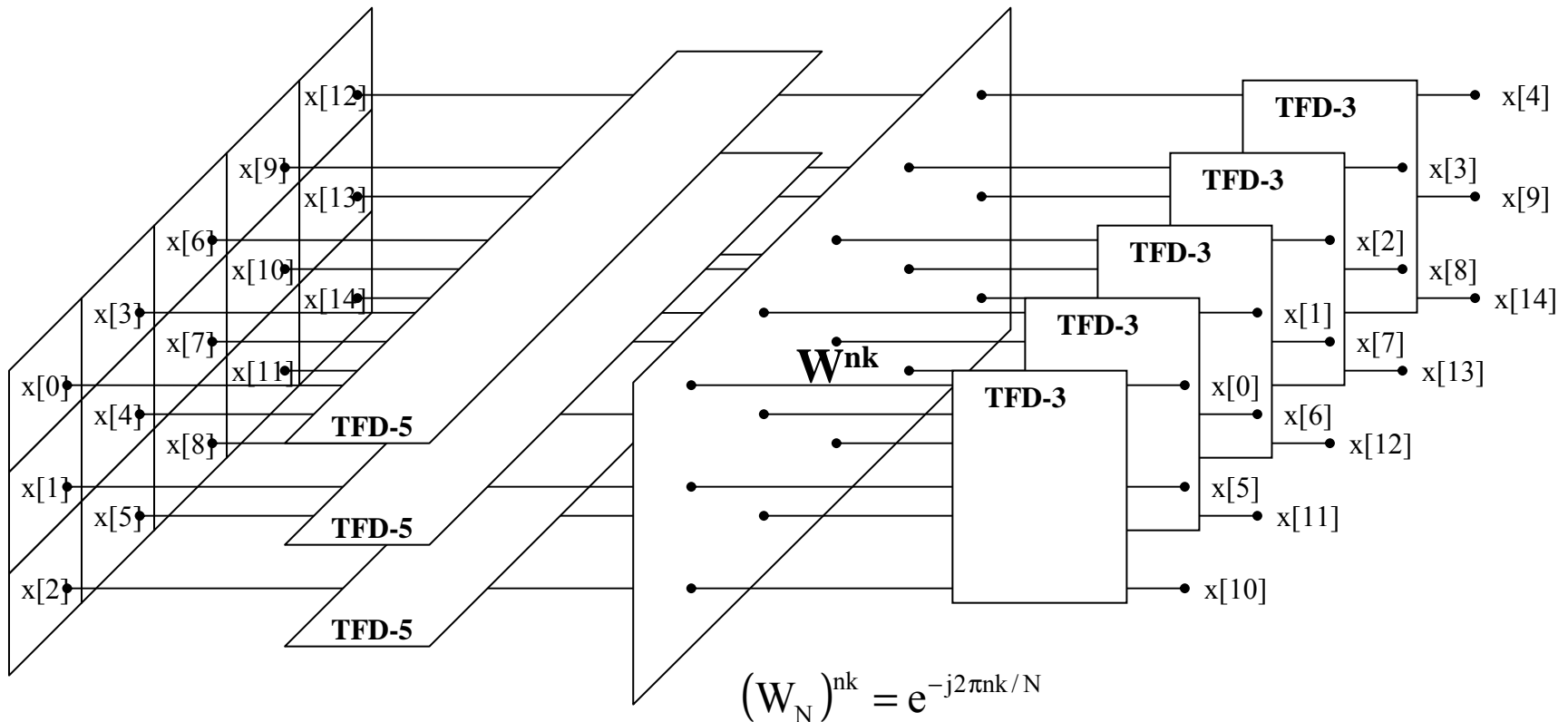




# Toujours plus vite

Il existe de nombreuses variantes autour de l'algorithme de Cooley-Tukey. Elles consistent toujours à réduire une TFD sur un grand nombre de points en plusieurs TFD plus courtes, intercalées avec une multiplication par des matrices de phases correspondants aux décalages temporels.

Exemple: une FFT Cooley-Tukey sur un signal de longueur  $15 = 3 \times 5$



# Toujours plus vite

Autre exemple: une FFT Cooley-Tuckey de base  $N \times 4$ , idem l'exemple précédent, qui se réduit sous forme matricielle à :

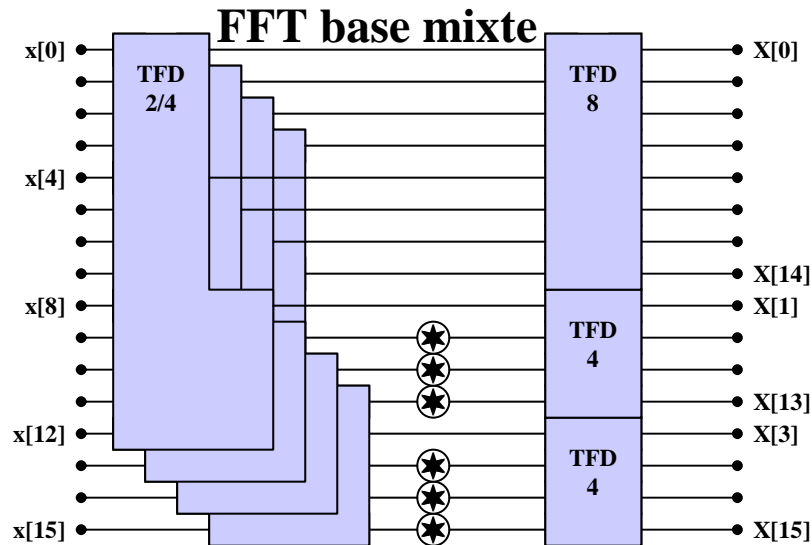
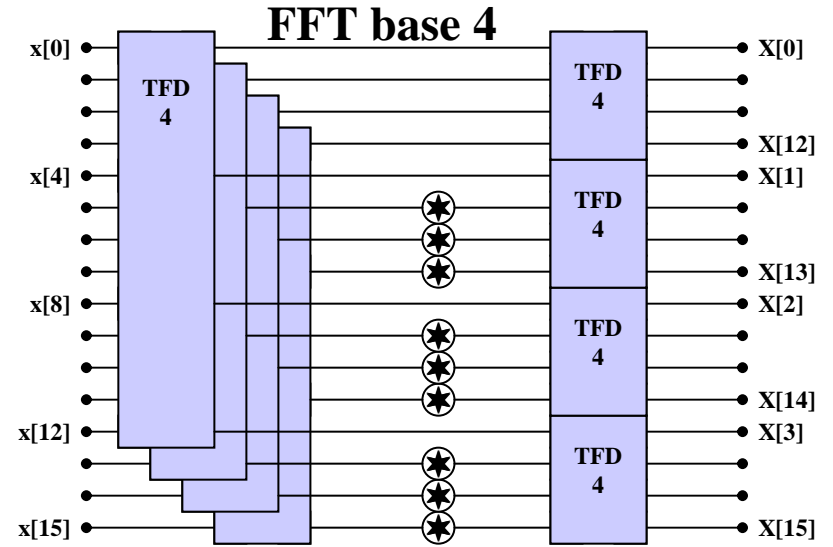
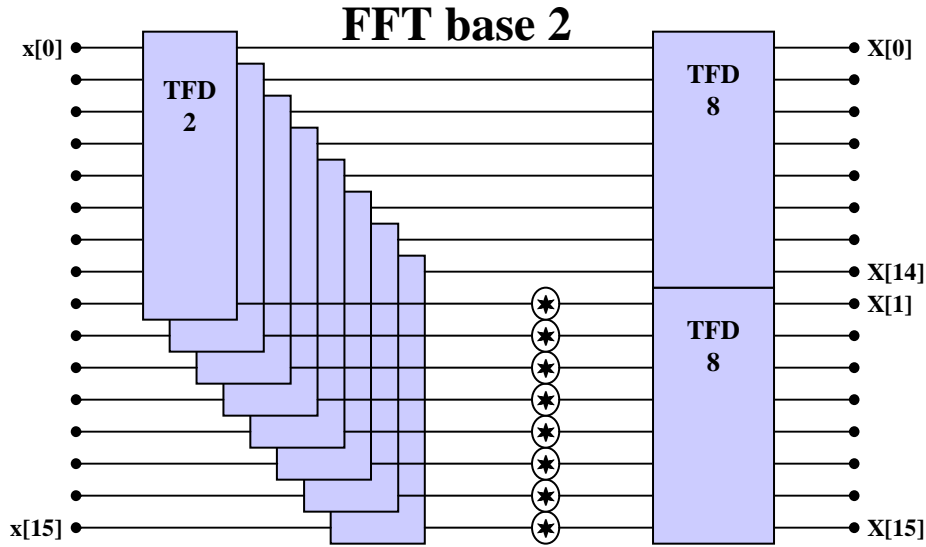
$$\begin{bmatrix} X[n] \\ X[n + N/4] \\ X[n + 2N/4] \\ X[n + 3N/4] \end{bmatrix} = M \begin{bmatrix} \sum_{k=0}^{(N/4)-1} W_{N/4}^{kn} x[4k] \\ \sum_{k=0}^{(N/4)-1} W_{N/4}^{kn} x[4k + 1] \\ \sum_{k=0}^{(N/4)-1} W_{N/4}^{kn} x[4k + 2] \\ \sum_{k=0}^{(N/4)-1} W_{N/4}^{kn} x[4k + 3] \end{bmatrix}$$

$$\text{ou } M = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ W_N^n \\ W_N^{2n} \\ W_N^{3n} \end{bmatrix}$$

et  $W_N = e^{-j2\pi/N}$  est la  $N$  - ième racine de l'unité

# Toujours plus vite

Autre exemples: une FFT de base 2, de base 4 et de base mixte (split radix)



# Toujours plus vite

## Comparaisons de rapidité :

### Nombres de multiplications

N	base 2	base 4	mixte
16	24	20	20
32	88		68
64	264	208	196
128	712		516
256	1800	1392	1284
512	4360		3076
1024	10248	7856	7172
2048	23560		16388

### Nombres d'additions

N	base 2	base 4	mixte
16	152	148	148
32	408		388
64	1032	976	964
128	2504		2308
256	5896	5488	5380
512	13566		12292
1024	30728	28336	27652
2048	68616		61444

Les algorithmes de base 2 et de base 4 requièrent respectivement 50% et 12,5% plus de multiplications que l'algorithmes de base mixte.

**Remarque :** les algorithmes basés sur la méthode de Cooley-Tuckey sont adaptés à des signaux dont la longueur est une puissance de 2, ou dont la longueur est composite  $N = N_1 \times N_2$ . Il existe des algorithmes adaptés aux cas non-factorisable ou la longueur du signal est un nombre premier. **L'algorithme de Rader**, la **permutation de Good**, **l'algorithme des facteurs premiers** et **l'algorithmes de Winograd** en sont des exemples.

# *Transformation de Laplace*

**Cours DSP**

# Transformation de Laplace

**La définition mathématique rigoureuse  
de la transformée de Laplace est la suivante:**

**Soit  $f(t)$  fonction du temps possédant les 3 propriétés suivantes:**

- $\forall t < 0, f(t) = 0$  (signal causal, ceci correspond à une réalité physique)
- $f$  est continue par morceaux sur  $]0, \infty[$
- $\forall S < \infty, f(t)$  est sommable sur  $[0, S]$
- $\exists$  trois nombres  $M, c, T$  tels que:  $|f(t)| \leq Me^{ct}, \forall t \geq T$

**Alors la transformée de Laplace de la fonction  $f(t)$  existe et est définie par :**

$$f(t) \xrightarrow{L} L[f(t)] = F(s)$$

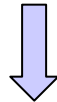
**avec**

$$F(s) = \int_0^{\infty} e^{-st} f(t) dt$$

# Transformation de Laplace

Analyse non plus seulement en terme de composantes fréquentielles,  
mais en termes de sinusoides et d'exponentielles

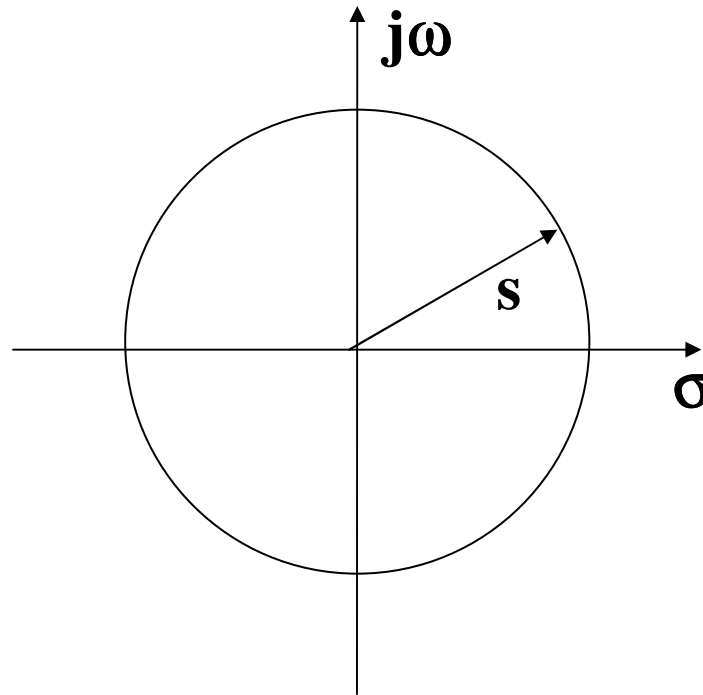
**TF:** 
$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$$



**TL:** 
$$X(\sigma, \omega) = \int_{-\infty}^{\infty} x(t) e^{-\sigma t} e^{-j\omega t} dt$$

avec  $\sigma + j\omega = s$

# Le plan complexe ou domaine s



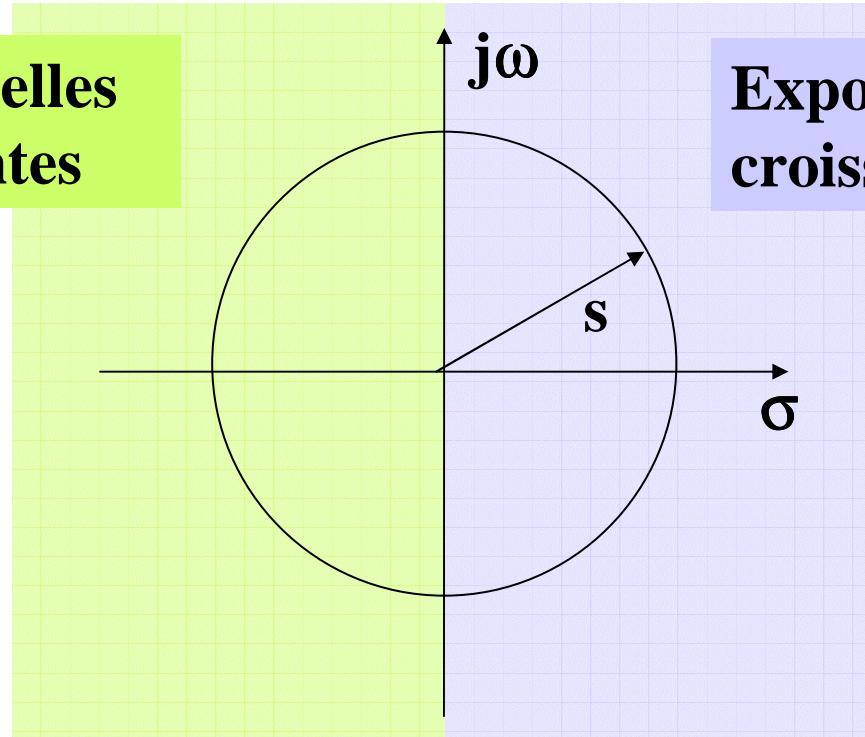
$$s = \sigma + j\omega$$

**NB: on trouve souvent la variable complexe notée  $p$  au lieu de  $s$  dans les ouvrages francophones**



# Le plan complexe ou domaine s

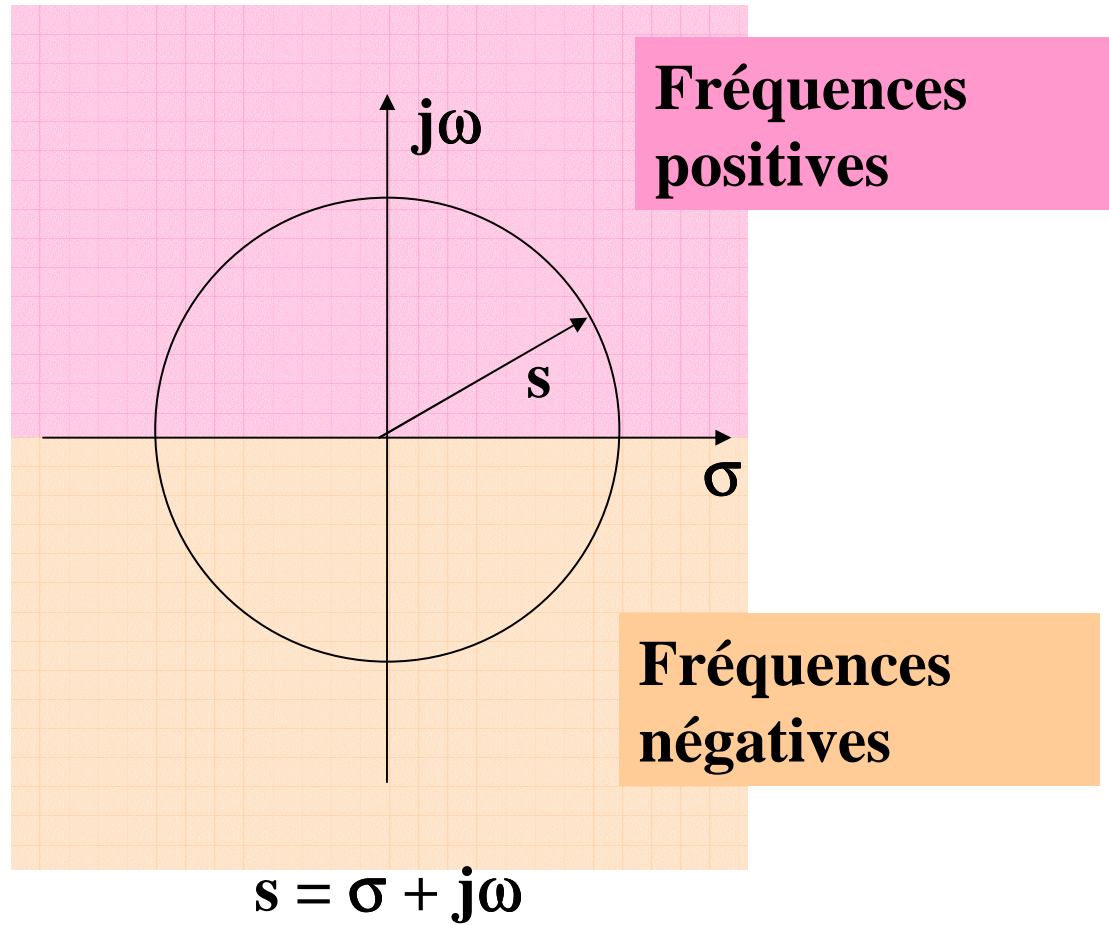
**Exponentielles  
décroissantes**



**Exponentielles  
croissantes**

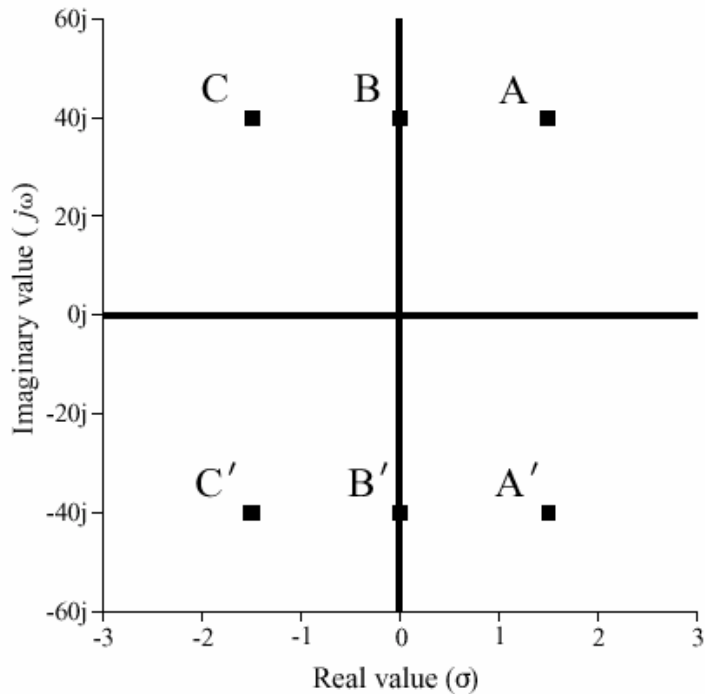
$$s = \sigma + j\omega$$

# Le plan complexe ou domaine s

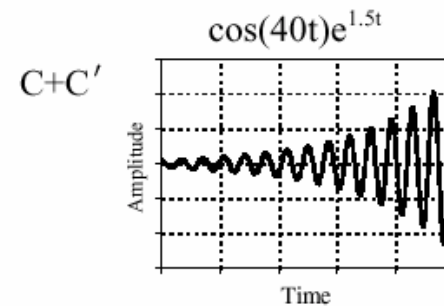
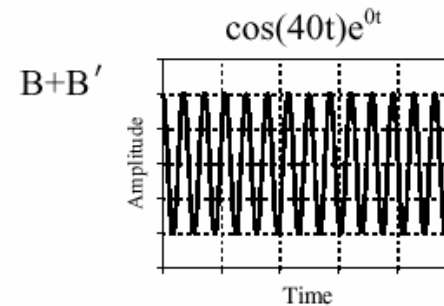
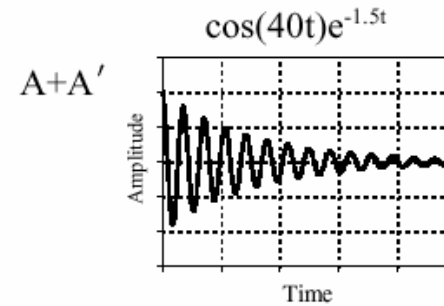


# Le plan complexe ou domaine s

## Domaine s



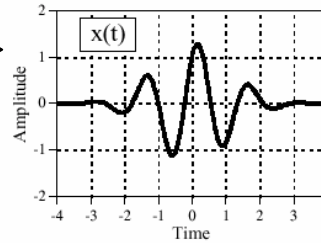
## Signaux associés



# Analyse de Laplace

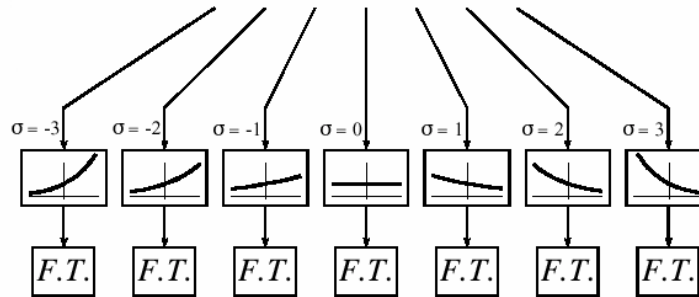
1<sup>ère</sup> étape:

Signal dans le domaine temporel  $x(t)$



2<sup>ème</sup> étape:

Multiplication par  $e^{-\sigma t}$   
pour tout  $\sigma \in ]-\infty, \infty[$



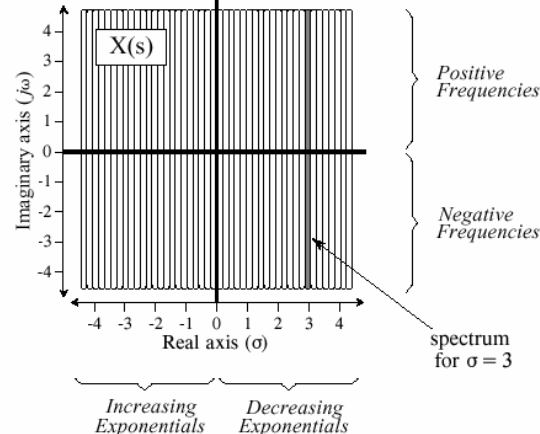
3<sup>ème</sup> étape:

Calculer la TF de  
chaque signal pondéré  
par une exponentielle



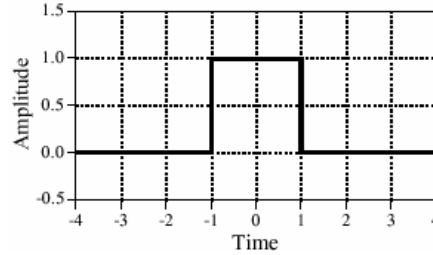
4<sup>ème</sup> étape:

Construire la TL dans le plan  $s$   
constituée d'une infinité de  
spectres en fréquence juxtaposés  
le long de l'axe  $\sigma$

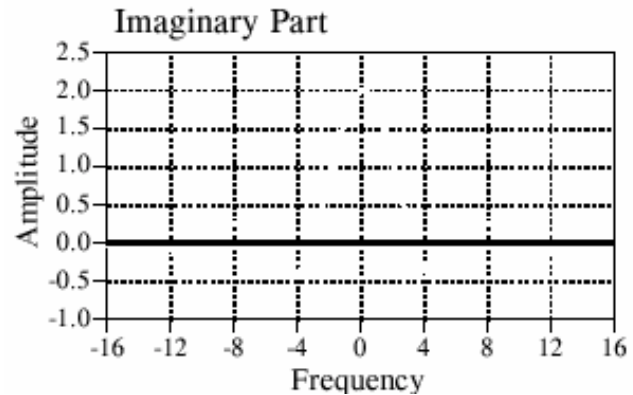
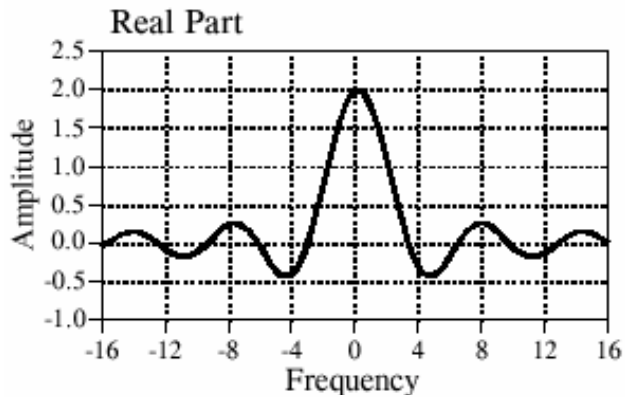


# Transformation de Laplace

## Domaine temporel



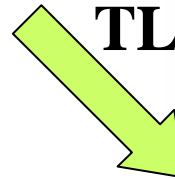
## Domaine Fréquentiel



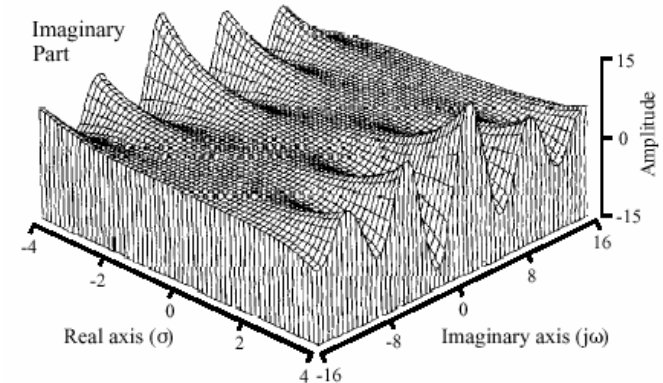
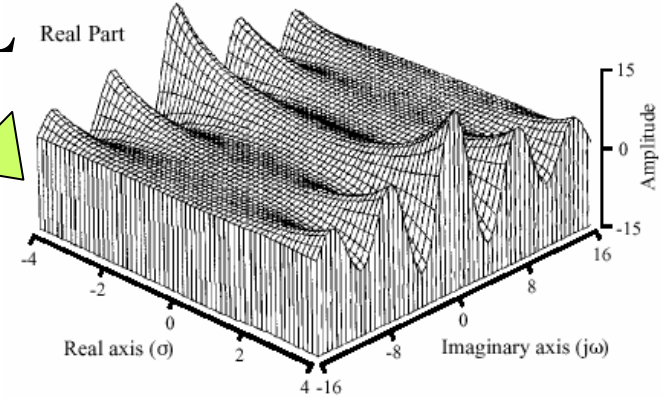
TF



TL

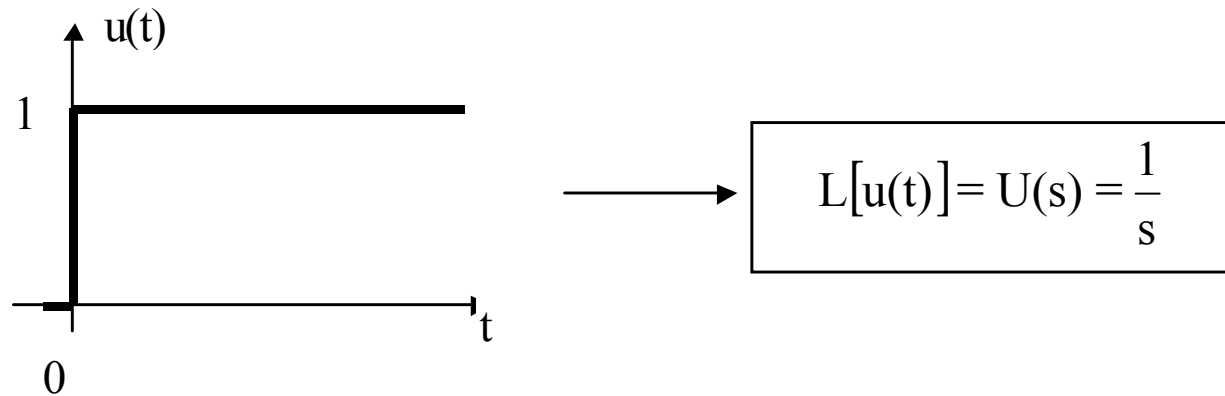


## Domaine s



# Signaux Tests

## 1) Échelon unitaire :



**C'est également la fonction de transfert  
d'un intégrateur pur.**

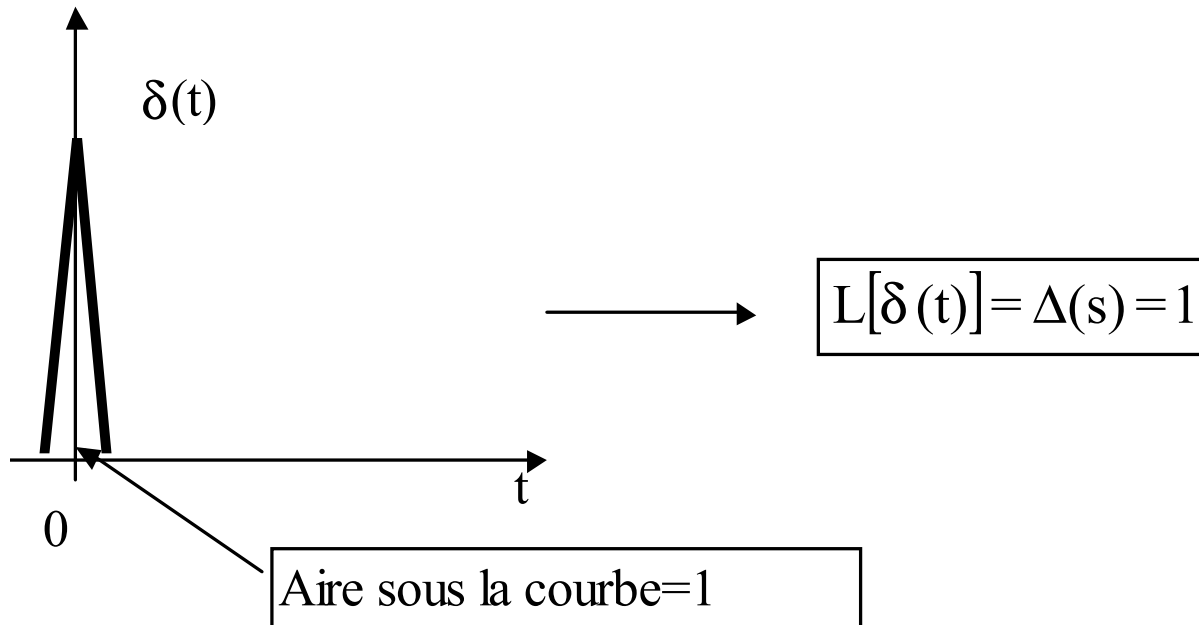
# Signaux Tests

## 2) Impulsion de Dirac

Définition de l'impulsion de Dirac  $\delta(t)$ :

-  $\forall t \neq 0, \delta(t) = 0$

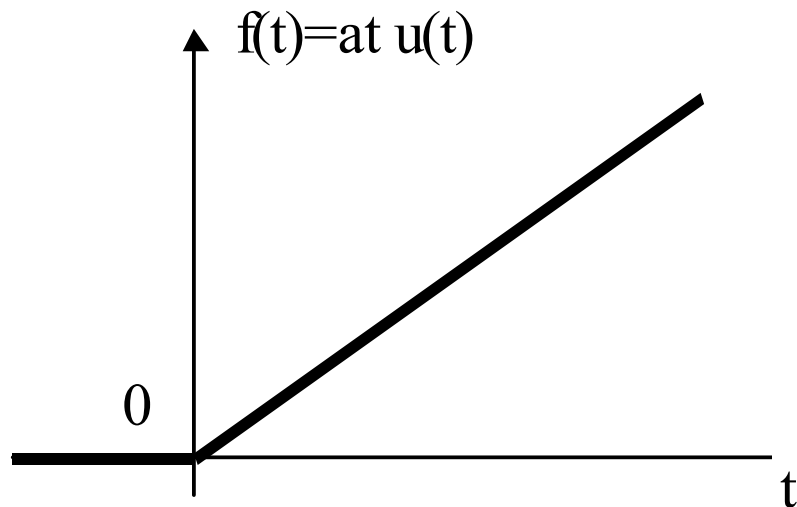
- pour  $t=0, \delta(0) = \infty$ , avec aire sous la courbe de  $\delta(t) = 1$



# Signaux Tests

## 3) Rampe at

$f(t)=0$  pour  $t < 0$  ,  $f(t) =at \forall t \geq 0$  . On peut écrire  $f(t)=at u(t)$

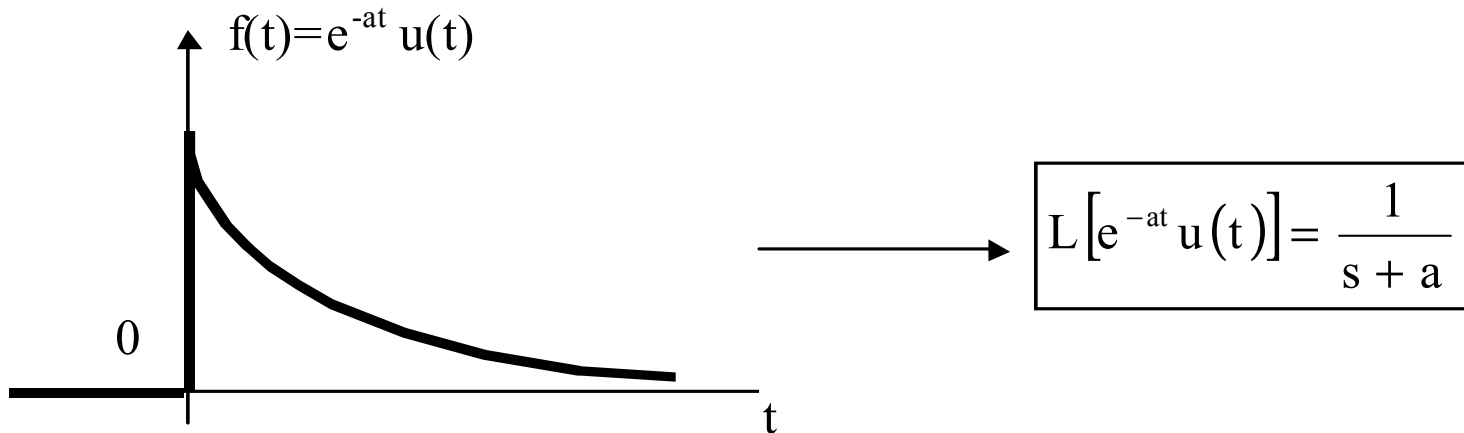


$$L[at] = \frac{a}{s^2}$$



# Signaux Tests

## 4) Exponentielle décroissante



# Propriétés générales

Propriétés	Originales	Transformées
Linéarité	$af(t) + bg(t)$	$aF(s) + bG(s)$
Facteur d'échelle	$f(at)$	$\frac{1}{ a } F\left(\frac{s}{ a }\right)$
Théorème du retard	$f(t - \tau)$	$e^{-\tau s} F(s)$
Théorème du décalage	$e^{-\omega t} f(t)$	$F(s + \omega)$
Théorème de la dérivée	$\frac{df(t)}{dt}$	$sF(s) - f(0^+)$
Dérivée seconde	$\frac{d^2f(t)}{dt^2}$	$s^2F(s) - sf(0^+) - f'(0^+)$
Intégration	$\int_0^t f(u)du$	$\frac{1}{s} F(s)$
Convolution	$\int_0^t f(u - \tau)g(\tau)d\tau$	$F(s)G(s)$

# Transformées Classiques

$$\mathcal{L}[u(t)] = \frac{1}{s}$$

$$\mathcal{L}[\delta(t)] = 1$$

$$\mathcal{L}[at] = \frac{a}{s^2}$$

$$\mathcal{L}[e^{-at}] = \frac{1}{s+a}$$

$$\mathcal{L}[te^{-at}] = \frac{1}{(s+a)^2}$$

$$\mathcal{L}[\sin \omega t] = \frac{\omega}{s^2 + \omega^2}$$

$$\mathcal{L}[\cos \omega t] = \frac{s}{s^2 + \omega^2}$$

# Théorèmes de la valeur initiale et de la valeur finale

## Théorème de la valeur finale

$$f(\infty) = \lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} pF(s) \text{ si cette limite existe}$$

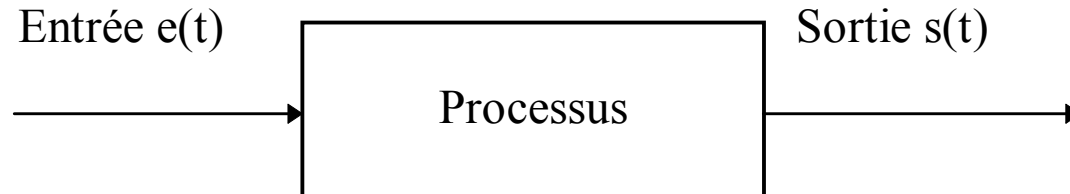
## Théorème de la valeur initiale

$$f(0^+) = \lim_{t \rightarrow 0} f(t) = \lim_{s \rightarrow \infty} pF(s) \text{ si cette limite existe}$$

# Applications aux fonctions de transfert

## Equations différentielles linéaires

Considérons le système :



dont le processus est régi par les équations différentielles suivantes:

$$\frac{d^2 y}{dt^2} + 5 \frac{dy}{dt} + 6y = e(t)$$

$$\text{Conditions initiales : } \frac{dy}{dt}(0) = 2 \text{ et } y(0) = 2$$

A  $t = 0$ , on applique une entrée constante  $e(t) = 6$ . Que vaut  $y(t)$  ?

# Application aux fonctions de transfert

Appliquons les transformées de Laplace à :

$$\frac{d^2y}{dt^2} + 5 \frac{dy}{dt} + 6y = e(t)$$

On obtient :

$$s^2Y(s) - 2s - 2 + 5(sY(s) - 2) + 6Y(s) = E(s)$$

Conditions initiales :

$$e(t) = 6 \Rightarrow E(s) = 6/s$$

On en tire  $Y(s)$ :

$$Y(s) = \frac{2s^2 + 12s + 6}{s(s^2 + 5s + 6)} = \frac{2s^2 + 12s + 6}{s(s+3)(s+2)}$$

# Application aux fonctions de transfert

$$Y(s) = \frac{2s^2 + 12s + 6}{s(s^2 + 5s + 6)} = \frac{2s^2 + 12s + 6}{s(s+3)(s+2)}$$

Décomposition en éléments simples :

$$Y(s) = \frac{2s^2 + 12s + 6}{s(s+3)(s+2)} = \frac{a}{s} + \frac{b}{s+3} + \frac{c}{s+2}$$

Pour trouver a, on multiplie les 2 membres de l'égalité par s et on fait s = 0

Pour trouver b, on multiplie les 2 membres de l'égalité par s + 3 et on fait s = -3

Pour trouver c, on multiplie les 2 membres de l'égalité par s + 2 et on fait s = -2

On trouve alors :

$$Y(s) = \frac{1}{s} - \frac{4}{s+3} + \frac{5}{s+2}$$

soit :

$$y(t) = 1 - e^{-3t} + 5e^{-2t}$$

# Applications aux systèmes électriques

## Rappels:

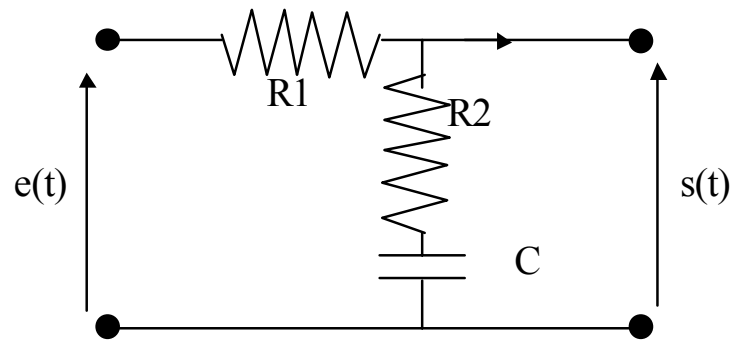
Soit  $u(t)$  la tension aux bornes d'un composant ( $U(s)$  sa transformée de Laplace) et  $i(t)$  le courant qui le traverse ( $I(s)$  sa T.L.):

- Pour une résistance:  $U(s) = R I(s)$
- Pour une capacité:  $U(s) = I(s) / C s$
- Pour une inductance:  $U(s) = L s I(s)$



# Applications aux systèmes électriques

Exemple :



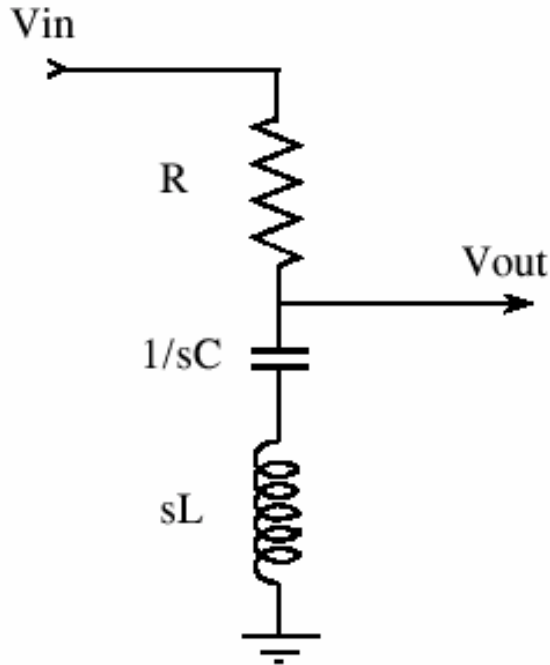
I supposé nul ( impédance d'entrée du circuit suivant infinie)

---

On a donc :

$$\frac{S(s)}{E(s)} = \frac{R_2 + \frac{1}{Cs}}{R_1 + R_2 + \frac{1}{Cs}} = \frac{1 + R_2Cs}{1 + (R_1 + R_2)Cs}$$

# Zéros et Pôles



$$H(s) = \frac{Ls^2 + 1/C}{Ls^2 + Rs + 1/C}$$

$$H(s) = \frac{(s - z_1)(s - z_2)}{(s - p_1)(s - p_2)}$$

avec :

$$z_1 = j/\sqrt{LC} \quad p_1 = \frac{-R + \sqrt{R^2 - 4L/C}}{2L}$$

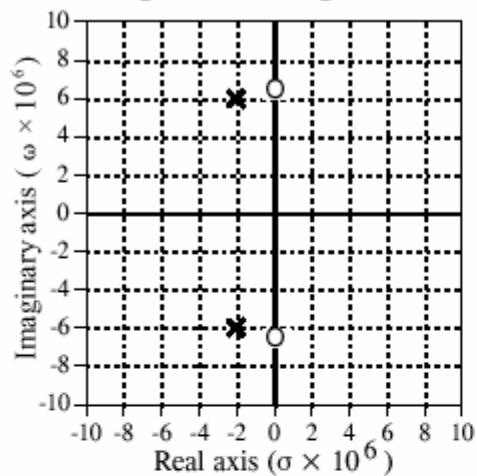
$$z_2 = -j/\sqrt{LC} \quad p_2 = \frac{-R - \sqrt{R^2 - 4L/C}}{2L}$$

**Zéros**

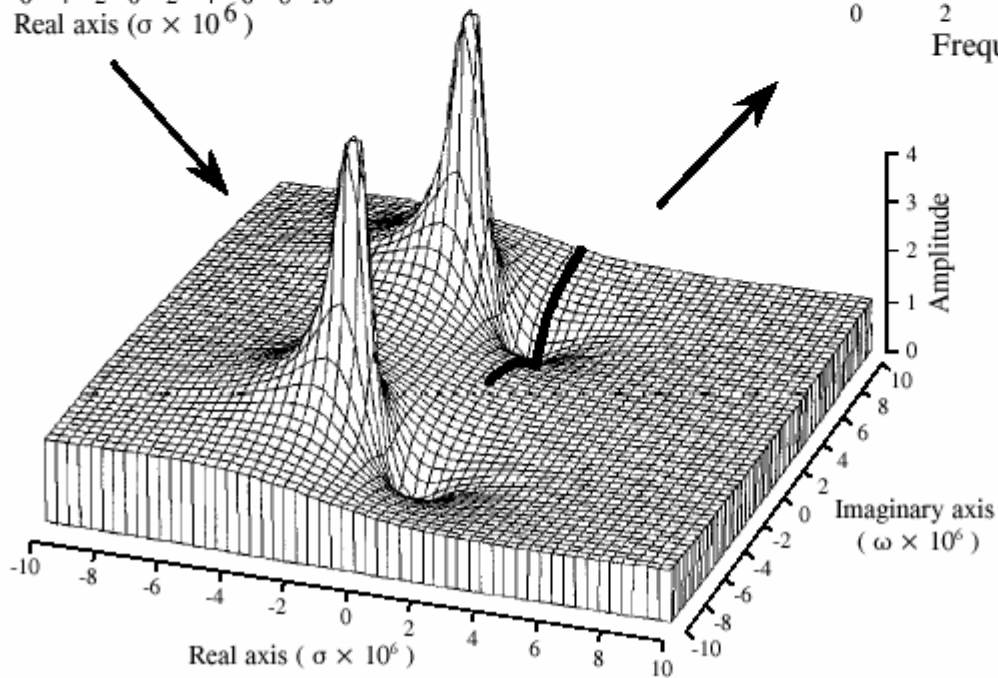
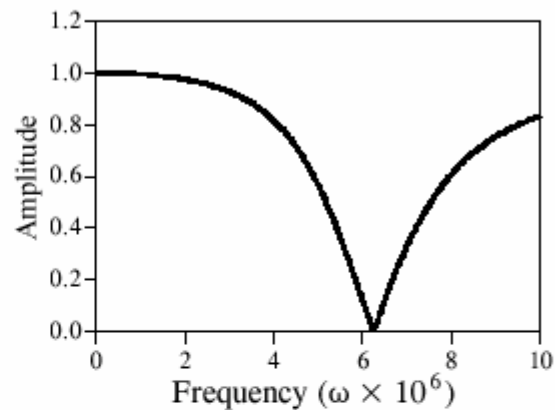
**Pôles**

# Zéros et Pôles

pole-zero plot

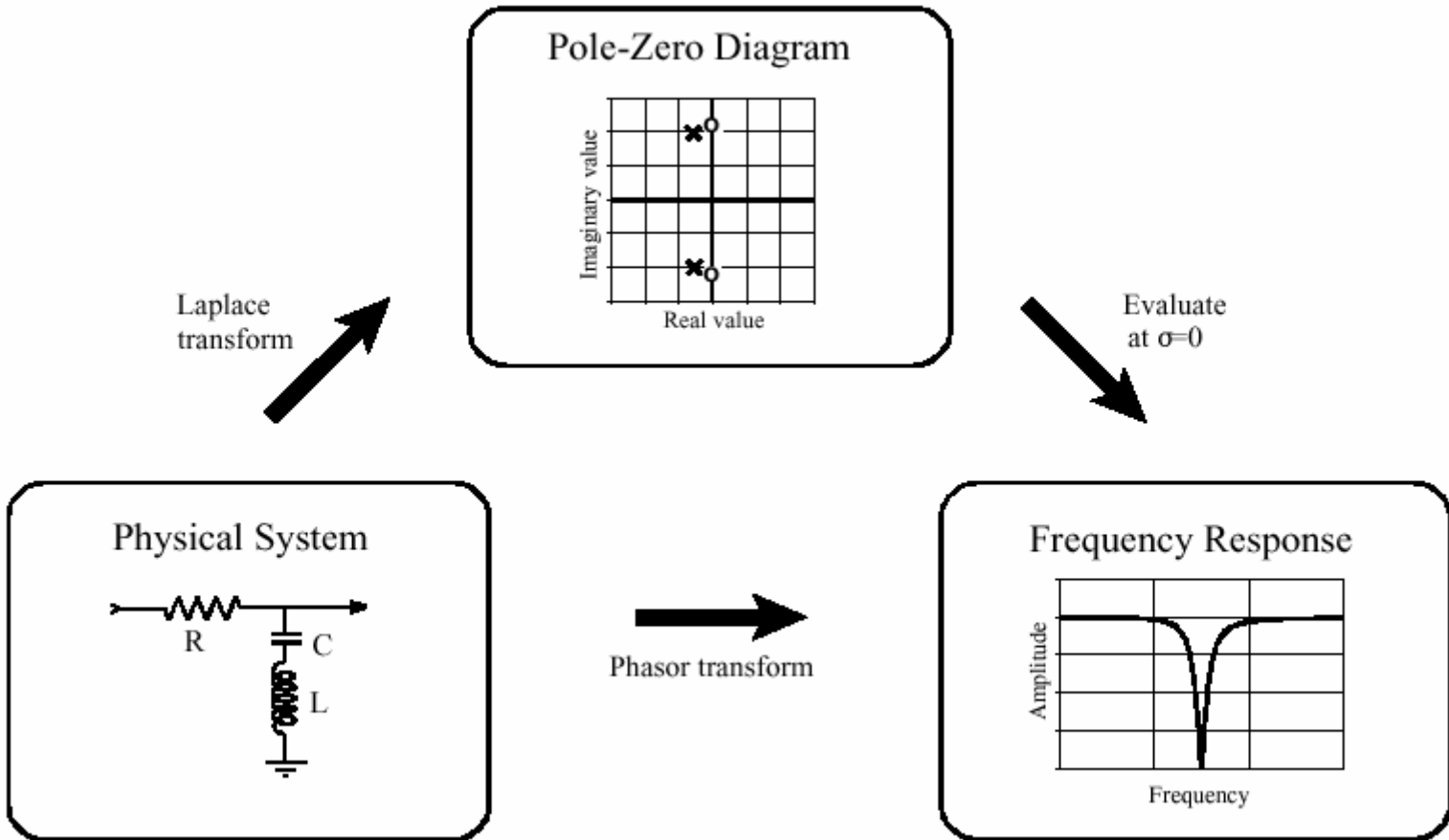


frequency response



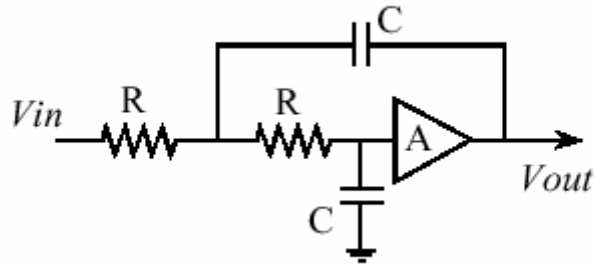
s-domain

# Zéros et Pôles



# Analyse de filtres dans le domaine s

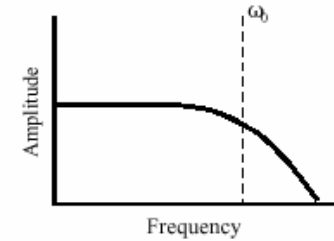
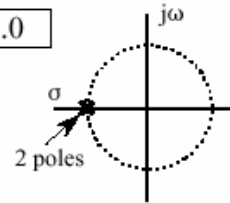
## Circuit Sallen-Key



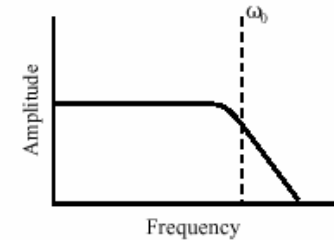
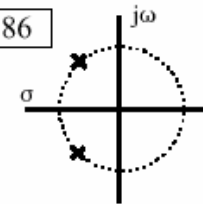
## Pôles en $\sigma + j\omega$

$$\sigma = \frac{A-3}{2RC} \quad \omega = \frac{\pm \sqrt{-A^2 + 6A - 5}}{2RC}$$

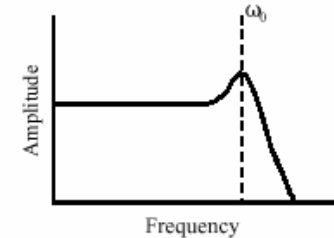
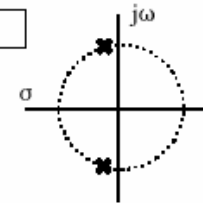
a.  $A = 1.0$



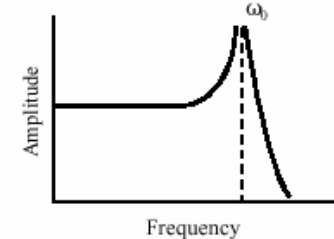
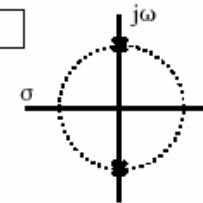
b.  $A = 1.586$



c.  $A = 2.5$

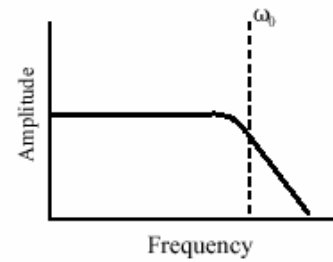
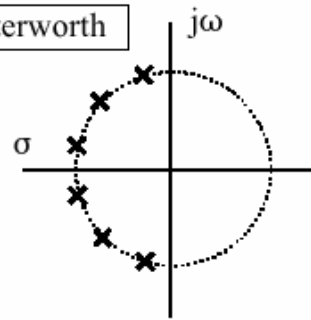


d.  $A = 3.0$

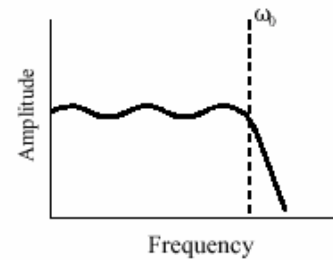
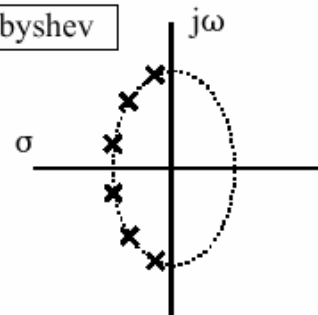


# Analyse de filtres dans le domaine s

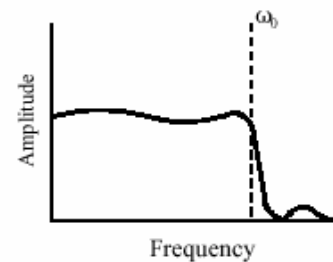
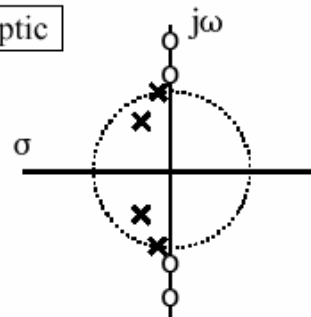
Butterworth



Chebyshev



Elliptic



# *Transformation en $z$*

**Cours DSP**

# Transformation en Z

La transformation en z reprend le formalisme et les propriétés de la transformation de Laplace et les applique aux signaux discrets.

Transformée en Z = Transformée de Laplace  
de la fonction échantillonnée avec  $z = e^{Ts}$  ou T est la période d'échantillonnage.

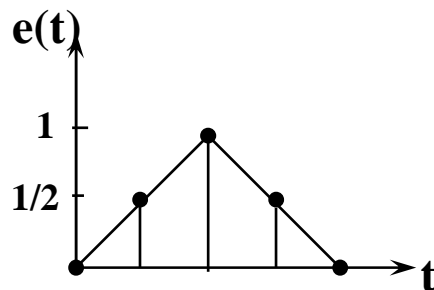
$$\text{TZ}\{f(t)\} = F(z) = \mathcal{L}[f^*(t)]_{z=e^{Ts}}$$

Transformation de Laplace

Transformation en Z

$$x(t) \xrightarrow{\mathcal{L}} X(p) = \int_0^{\infty} x(t)e^{-st} dt \quad \left| \quad x[k] = x(kt) \xrightarrow{\mathcal{Z}} S(z) = \sum_{k=0}^{\infty} x[k]z^{-k}$$

Exemple :

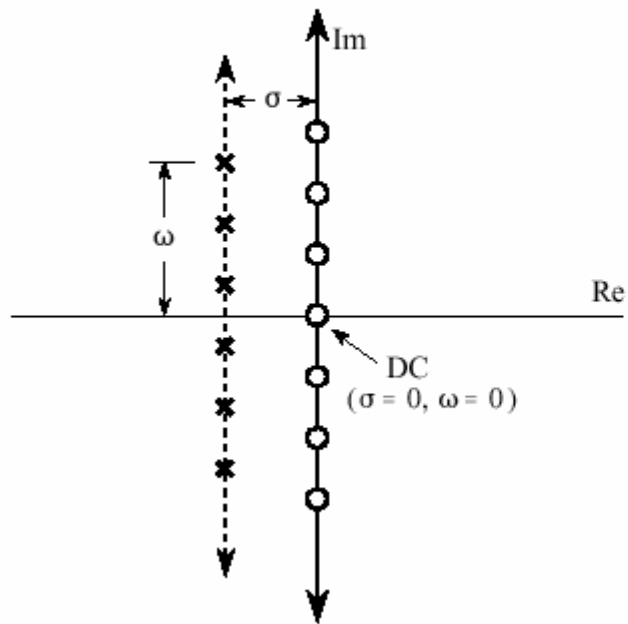


$$E^*(s) = 0 + \frac{1}{2}e^{-Ts} + e^{-2Ts} + \frac{1}{2}e^{-3Ts}$$
$$E(z) = \text{TZ}[e(t)] = \frac{1}{2}z^{-1} + z^{-2} + \frac{1}{2}z^{-3}$$

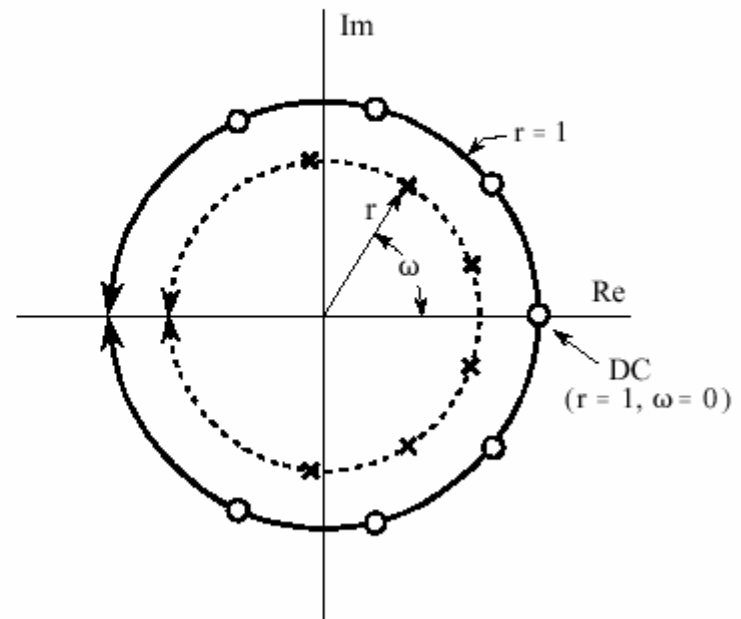


# Le plan ou domaine z

s - Plane



z - Plane



# Propriétés de la transformées en Z

Linéarité:  $\text{TZ}\{\alpha f(t) + \beta g(t)\} = \alpha F(z) + \beta G(z)$

Retard:  $\text{TZ}\{x[n - k]\} = F(z)z^{-k}$

**Cette propriété est particulièrement importante. Elle donne tout son intérêt à la transformée en Z. En effet, la multiplication par  $z^{-1}$  correspond à ma translation en temps d'un échantillon au suivant :**

$$\text{TZ}\{x[n - 1]\} = X(z)z^{-1}$$

Avance:  $\text{TZ}\{x[n + k]\} = \left[ X(z) - \sum_{m=0}^{m=k-1} x[m]z^{-m} \right] z^k$

# Propriétés de la transformées en $Z$

## Conditions initiales et finales :

$$\underline{\text{Initiales}}: \lim_{z \rightarrow 0} X(z) = \lim_{n \rightarrow 0} x[n]$$

$$\underline{\text{Finales}}: \lim_{n \rightarrow \infty} x[n] = \lim_{z \rightarrow 1} [(z-1)X(z)] \quad \text{si cette limite existe}$$

**Convolution** :    Soit  $y[n] = \sum_{k=0}^n h[n-k] \cdot x[k]$   
alors  $Y(z) = H(z) \cdot X(z)$

# Transformées en Z

**Échelon unitaire :**

$$\boxed{\text{TZ}[u(t)] = \frac{z}{z-1}}$$

$$U^*(s) = 1 + e^{-Ts} + e^{-2Ts} + e^{-3Ts} + \dots$$

comme  $|e^{-T}| < 1$ , la série converge

$$\text{TZ}[u(t)] = \left( \frac{1}{1 - e^{-Ts}} \right)_{z=e^{Ts}} = \frac{1}{1 - z^{-1}} = \frac{z}{z-1}$$

**Rampe unitaire :**

à démontrer

$$\boxed{\text{TZ}[tu(t)] = \frac{Tz}{(z-1)^2}}$$

# Transformées en Z

**Exponentielle** :

$$\text{TZ}[e^{-at}u(t)] = \frac{z}{z - e^{-aT}}$$

**Sinus et Cosinus** :

$$\text{TZ}[\cos(at)u(t)] = \frac{z(z - \cos(aT))}{z^2 - 2\cos(aT)z + 1}$$

$$\text{TZ}[\sin(at)u(t)] = \frac{z\sin(aT)}{z^2 - 2\cos(aT)z + 1}$$

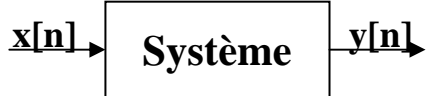
**Signaux test élémentaires**  
(échelon unitaire, rampe, paraboles, etc ...) :

$f(t)$	$L[f(t)]$	$Z[f(t)]$
$u(t)$	$\frac{1}{s}$	$\frac{z}{z-1}$
$tu(t)$	$\frac{1}{s^2}$	$\frac{Tz}{(z-1)^2}$
$\frac{1}{2!}t^2u(t)$	$\frac{1}{s^3}$	$\frac{N_2(z)}{(z-1)^2}$
$\vdots$	$\vdots$	$\vdots$
$\frac{1}{n!}t^nu(t)$	$\frac{1}{s^{n+1}}$	$\frac{N_n(z)}{(z-1)^n}$

avec  $N_2(z), N_3(z), \dots, N_n(z)$   
polynômes en z

# Fonction de transfert et Pôles dans le domaine Z

**La sortie  $y[n]$  et  $x[n]$  d'un système physique linéaire échantillonné sont reliés par une équation aux différences :**

$$y[n] = \sum_{k=0}^M a_k x[n-k] + \sum_{k=1}^P b_k y[n-k]$$


la sortie au temps  $nT$  dépend des valeurs des échantillons connus à ce moment c.a.d  $x[n]$ ,  $x[n-1]$ ,  $x[n-2]$ ... et  $y[n-1]$ ,  $y[n-2]$ , ... et des caractéristiques du système définies par les  $a_k$  et  $b_k$  (système linéaire causal)

**En prenant la TZ et en utilisant les propriétés de linéarité et de translation en temps, on obtient :**

$$Y(z) \left( 1 - \sum_{k=1}^P b_k z^{-k} \right) = X(z) \left( \sum_{k=0}^M a_k z^{-k} \right)$$

**On définit la fonction de transfert  $H(z)$  qui ne dépend que des  $a_k$  et  $b_k$  et qui résume les propriétés du système :**  $Y(z) = H(z) X(z)$

**Donc :**

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M a_k z^{-k}}{1 - \sum_{k=1}^P b_k z^{-k}}$$

# Fonction de transfert et Pôles dans le domaine Z

**Fonction de transfert en z, un rapport de polynômes en z avec  $M \leq P$  pour un système physique.**

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M a_k z^{-k}}{1 - \sum_{k=1}^P b_k z^{-k}}$$

**qui peut se mettre également sous la forme :**

$$H(z) = a_0 z^{P-M} \frac{\prod_{i=1}^M (z - z_i)}{\prod_{j=1}^P (z - p_j)}$$

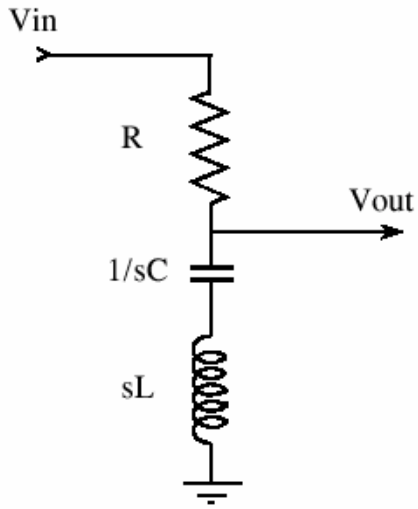
**Les  $z_i$  sont les zéros et les  $p_j$  sont les pôles du système.**

**Les zéros et les pôles décrivent le système à un facteur d'amplitude près  $a_0$  (gain statique) et à un retard près  $z^{P-M}$  avec  $P-M \leq 0$ .**

**En particulier, on appellera**

- **filtres MA (à Moyenne Adaptée) les filtres n'ayant que des zéros (et des pôles en  $z = 0$ )**
- **filtres AR (Auto Régressifs) les filtres n'ayant que des pôles (et des zéros en  $z = 0$ )**

# Pôles dans le domaine Z



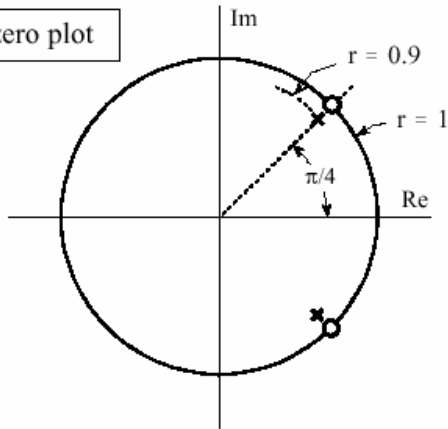
$$H(z) = \frac{(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)}$$

avec :

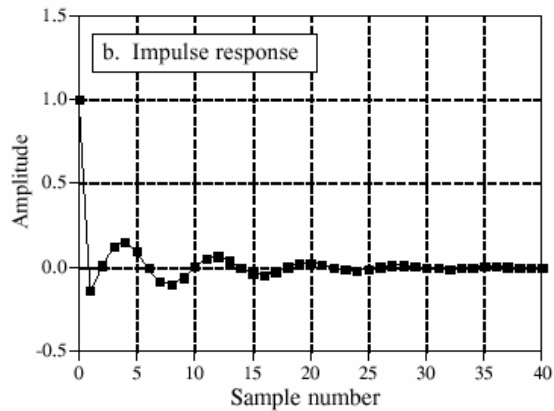
$$z_1 = j/\sqrt{LC} \quad p_1 = \frac{-R + \sqrt{R^2 - 4L/C}}{2L}$$

$$z_2 = -j/\sqrt{LC} \quad p_2 = \frac{-R - \sqrt{R^2 - 4L/C}}{2L}$$

a. Pole-zero plot



b. Impulse response



c. Frequency response

