Electronique pour la conception des systèmes embarqués

Filtrage Numérique

Chadi Jabbour

Année scolaire 2024-2025

Plan

Outline

Caractéristiques et propriétés des filtres numériques

Types de Filtres

Synthèse des filtres

Décimation et interpolation

Implémentation

Conclusion

Systèmes Discrets

- Caractérisé par une relation entrée sortie $y[n] = f(x[n])$
- Les valeurs en entrée et en sortie ne sont connues qu'à des instants discrets
- L'intervalle de temps qui sépare deux de ces instants est la période d'échantillonnage Te noté aussi T_s en anglais

Linéarité et invariance

Système linéaire

$$f(ax1[n] + bx2[n]) = af(x1[n]) + bf(x2[n])$$

La réponse à une somme d'excitations est la somme des réponses partielles aux excitations

Système invariant

$$y[n] = f(x[n]) \Longrightarrow y[n-k] = f(x[n-k])$$

Si la séquence d'entrée est retardée de kTe, la séquence de sortie sera également retardée de kTe

Causalité

- Signal causal : $x[n] = 0$ si $n < 0$
- Système causal : $x[n]$ est causal $\Longrightarrow$ y[n] est causal
- La réponse ne précède jamais l'entrée (temps de propagation positif)
- Exemples de signaux causaux discrets :
  - Impulsion unité ($\delta[n]$=1 pour n=0, $\delta[n]$=0 sinon)
  - Echelon unitaire ($u[n]$=1 pour n$\geq$ 0, $u[n]$=0 sinon)

Relation entrée sortie : produit de convolution discret

Définition :

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Propriétés :

Commutativité $h[n] * x[n] = x[n] * h[n]$

Distributivité $h[n] * (e[n] + x[n]) = h[n] * e[n] + h[n] * x[n]$

Elément neutre $h[n] * \delta[n] = h[n]$

Décalage $h[n] * \delta[n-k] = h[n-k]$

Transformée en $\mathcal{Z}$

Définition pour une entrée $x[n]$ causale :

$$T\mathcal{Z}\left\{x[n]\right\} = X(\mathcal{Z}) = \sum_{n=0}^{\infty} x[n]\mathcal{Z}^{-n}$$

Transformée en $\mathcal{Z}$ de la sortie du filtre

$$Y(\mathcal{Z}) = \sum_{n}\sum_{k} x[k]h[n-k]\mathcal{Z}^{-n} = \sum_{n}\sum_{k} x[k]h[n-k]\mathcal{Z}^{-[n-k]-k}$$

$$Y(\mathcal{Z}) = \sum_{k} x[k]\mathcal{Z}^{-k}\sum_{n-k} h[n-k]\mathcal{Z}^{-[n-k]} = X(\mathcal{Z})\cdot H(\mathcal{Z})$$

La convolution dans le domaine temporel se traduit par une multiplication dans le domaine des $\mathcal{Z}$

Systèmes continus à systèmes discrets

Temps continu	Temps Discret
$h(t)$ réponse impulsionnelle	$h[n]$ réponse impulsionnelle discrète
Réponse temporelle $y(t) = x(t) * h(t)$	Réponse temporelle $y[n] = x[n] * h[n]$
Fonction de transfert en p $H(p)$	Fonction de transfert en $\mathcal{Z}$ $H(\mathcal{Z})_{\mathcal{Z}=e^{p \cdot Te}}$
Domaine complexe $Y(p) = X(p) \cdot H(p)$	Domaine complexe $Y(\mathcal{Z}) = X(\mathcal{Z}) \cdot H(\mathcal{Z})$
Transformée de Fourier $H(p)_{p=j\omega}$	Domaine fréquentiel $H(\mathcal{Z})_{\mathcal{Z}=e^{j\omega \cdot Te}}$

Stabilité

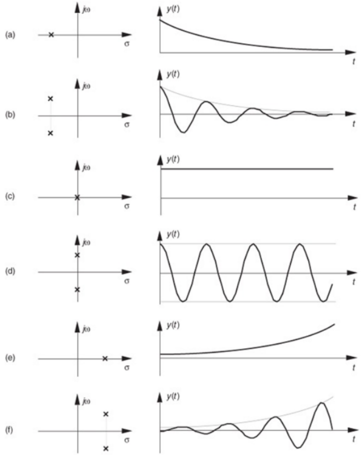
Stablilité dans le domaine temporel

Un filtre $h[n]$ est stable si et seulement si $\sum_{n=-\infty}^{\infty} \mid h[n] \mid < \infty$

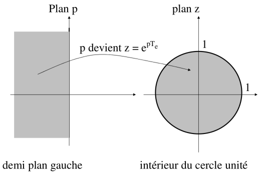
Stablilité dans le domaine des $\mathcal{Z}$

Un filtre $H(\mathcal{Z})$ est stable si et seulement tous ses poles sont à l'intérieur du cercle unité

Stabilité temps continu



Stabilité équivalence temps continu- discret

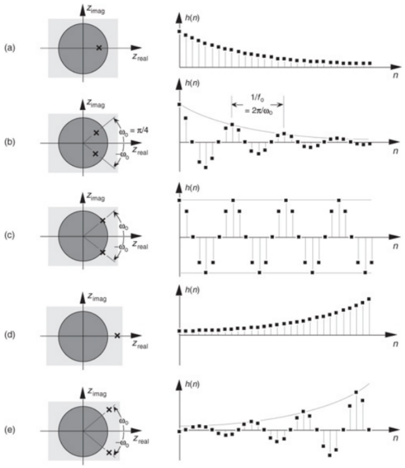


Un système de fonction de transfert H(p) est stable si les pôles sont à partie réelle négative

$$p = \sigma + j\omega \Longrightarrow \mathcal{Z} = e^{(\sigma+j\omega)Te} = e^{\sigma Te} \cdot e^{j\omega Te}$$

$$\sigma < 0 \Longrightarrow \mid e^{pTe} \mid < 1$$

Stabilité temps discret



Stabilité exemples

a) Calculer les réponses impulsionnelle et indicielle pour le filtre 1 sur 10 cycles. Déduisez-en si le filtre est stable ou instable
b) Etudier la stabilité des 4 filtres en analysant leurs poles.
c) Simuler les 4 filtres sous Octave/Matlab et comparer vos résultats à ceux de la question précédente

1. $H(z) = \frac{1}{1-2\mathcal{Z}^{-1}}$
2. $H(z) = \frac{1+0.2\mathcal{Z}^{-1}}{1-0.8\mathcal{Z}^{-1}}$
3. $H(z) = \frac{1}{1+2\mathcal{Z}^{-1}}$
4. $H(z) = \frac{1+7\cdot\mathcal{Z}^{-1}}{1-0.3\mathcal{Z}^{-1}+0.5\mathcal{Z}^{-2}}$

Outline

Relation Entrée-sortie

Il existe trois approches principales pour exprimer la relation entrée-sortie d'un filtre :

- Equation aux différences
- Convolution avec la réponse impulsionnelle
- Produit avec la transformée en $\mathcal{Z}$

1ere approche : Equation aux différences

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{r=0}^{M} b_r x[n-r]$$

en normalisant a_0 à 1, ça donne

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{r=0}^{M} b_r x[n-r]$$

$N > 1$	$N = 0$
Filtre à réponse récursive	Filtre à réponse non-récursive
$y[n]$ dépend de $y[n-k]$	$y[n]$ ne dépend pas de $y[n-k]$
Infinité de termes $x[n-k]$	Nombre fini de terme $x[n-k]$
La réponse impulsionnelle est infinie (RII, ou IIR en anglais)	La réponse impulsionnelle est finie (RIF, ou FIR en anglais)

2eme approche : Réponse impulsionnelle

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

S'il existe un N et n_0, tel que

$$h[n] = \begin{cases} \neq 0, & \text{pour} n_0 \leq n < n_0 + N - 1 \\ = 0, & \text{sinon} \end{cases}$$

le filtre a une réponse impulsionnelle de longueur N, il s'agit donc d'un FIR

si $h[n] \neq 0$ pour $n > n_0$
$\Longrightarrow$ le filtre a une réponse impulsionnelle infinie, il s'agit donc d'un IIR

3eme approche : Fonction en $\mathcal{Z}$

A partir de l'équation aux différences

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{r=0}^{M} b_r x(n-r)$$

on peut calculer la transformée en $\mathcal{Z}$

$$H(\mathcal{Z}) = \frac{\sum_{r=0}^{M} b_r \mathcal{Z}^{-r}}{\sum_{k=0}^{N} a_k \mathcal{Z}^{-k}}$$

3eme approche : Fonction en $\mathcal{Z}$

3 types de filtres peuvent être distingués

- Filtre non récursif (FIR)

$$H(\mathcal{Z}) = \sum_{r=0}^{M} b_r \mathcal{Z}^{-r}$$

- Filtre purement récursif (IIR-cas particulier)

$$H(\mathcal{Z}) = \frac{b_0}{\sum_{k=0}^{N} a_k \mathcal{Z}^{-k}}$$

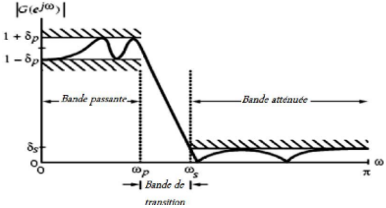
- Filtre récursif à moyenne ajustée (IIR-cas général)

$$M > 0 \quad \& \quad N > 1$$

Outline

Synthèse des filtres

Synthétiser un filtre revient à déterminer les coefficients a_k et b_r pour lesquels la réponse fréquentielle du filtre satisfait le gabarit donné.



Approche

Partant d'un filtre analogique (Butterworth, Bessel, elliptic,...), le problème revient à établir un passage entre la transformée de Laplace et la transformée en $\mathcal{Z}$

La relation exacte $\mathcal{Z} = e^{pTe}$ donne $pTe = ln(\mathcal{Z})$

Problématique et solution

Problématique

En effectuant cette transformation, il est impossible d'obtenir une équation aux différences à partir de la fonction de transfert $H(p)$, puisque $H(\mathcal{Z})$ ne sera pas un rapport de polynômes en $\mathcal{Z}^{-1}$ mais une fonction de $ln(\mathcal{Z})$

Solution

Partir d'un filtre analogique continu, utiliser une approximation sur la fonction de conversion p à $\mathcal{Z}$ pour avoir un résultat similaire sur une plage de fréquence

1ere méthode : Invariance de la dérivée ou méthode de Euler

Dérivée en temporel

$$\frac{dx}{dt} \simeq \frac{x(t) - x(t - Te)}{Te} \simeq \frac{x[n] - x[n-1]}{Te}$$

Ce qui donne dans le domaine des $\mathcal{Z}$, $X(\mathcal{Z})\frac{1-\mathcal{Z}^{-1}}{Te}$

Par ailleurs, la transformée en p d'un dérivateur vaut p

Donc le passage entre p et $\mathcal{Z}$ est donné par :

$$p \longrightarrow \frac{1 - \mathcal{Z}^{-1}}{Te}$$

Exemple

Filtre passe bas du premier ordre

$$H(p) = \frac{1}{1 + \tau p}$$

2eme méthode : Invariance de l'intégrale ou transformation bilinéaire

Soit $y(t)$ l'intégrale de la fonction $x(t)$
On cherche à obtenir une approximation de cette intégrale en utilisant les échantillons de x(t).

On l'approxime par l'aire des trapèzes formés en joignant les échantillons entre eux.

$$y[n] = y(n-1) + \frac{Te}{2}(x[n] + x[n-1])$$

$$Y(\mathcal{Z})(1 - \mathcal{Z}^{-1}) = \frac{Te}{2} X(\mathcal{Z})(1 + \mathcal{Z}^{-1})$$

$$\frac{Y(\mathcal{Z})}{X(\mathcal{Z})} = \frac{Te}{2} \frac{1 + \mathcal{Z}^{-1}}{1 - \mathcal{Z}^{-1}}$$

2eme méthode : Invariance de l’intégrale ou transformation bilinéaire

Par ailleurs, la transformée en p d’un intégrateur vaut $\frac{1}{p}$

Donc le passage entre p et $\mathcal{Z}$ est donné par :

$$p \longrightarrow \frac{2}{Te}\frac{1-\mathcal{Z}^{-1}}{1+\mathcal{Z}^{-1}}$$

Cette transformation est nommée transformation bilinéaire

3eme méthode : Transformée de Fourier Inverse

Partons de la réponse fréquentielle idéale $H(j\omega)$ du filtre voulu et calculons sa transformée de Fourier inverse pour déterminer la réponse impulsionnelle

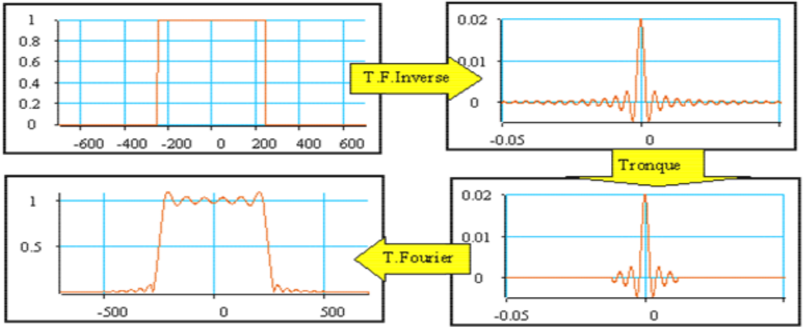
Problématique

La transformée de Fourier inverse d'une fonction bornée en fréquence donne une fonction non limitée dans le temps.On obtient donc une fonction $h[n]$ avec une infinité de termes

Solution “simple”

Le plus simple est de tronquer cette réponse idéale et de ne garder que la partie centrale qui comprend les valeurs les plus grandes.

Problème de la solution “simple”



On constate alors l’apparition d’ondulations importantes autour de la réponse en fréquence idéale. En effet la transformée de Fourier d’une fonction limitée dans le temps est une fonction non limitée en fréquence

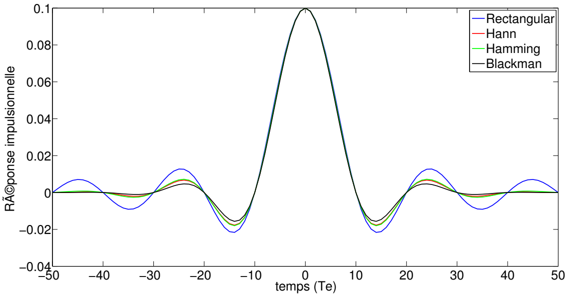
Apodisation

Pour réduire ces ondulations, il faut "arrondir les angles" de la fenêtre temporelle qui effectue la troncature en utilisant une fenêtre d'apodisation

Fenêtre (valeurs pour 0 ≤ n < N - 1)	Demi-largeur de bande du lobe principal à la base	Atténuation $20\log\left\|\frac{H(f_{lobe1})}{H(0)}\right\|$
Rectangulaire : $w = 1$	$\frac{1}{N}$	-13,5 dB
Bartlett : $w = \frac{2n}{N-1}$ si $0 \le n \le \frac{N}{2} - 1$ (N pair) $w = \frac{2(N-n-1)}{N-1}$ si $\frac{N}{2} \le n \le N-1$	$\frac{2}{N}$	-26,5 dB
Hanning : $w = \frac{1}{2}\left(1 - \cos(2\pi \frac{n}{N-1})\right)$	$\frac{2}{N}$	-31,5 dB
Hamming : $w = 0{,}54 - 0{,}46\cos(2\pi \frac{n}{N-1})$	$\frac{2}{N}$	-44,0 dB

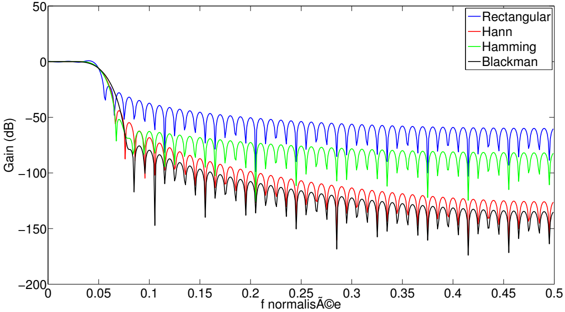
Apodisation

Exemple pour filtre passe bas avec un fréquence de coupure à 0.1 Fe



Apodisation

Exemple pour filtre passe bas avec un fréquence de coupure à 0.1 Fe



Méthodes de génération numérique

Approches de génération numérique

Ces méthodes sont globalement basées sur une minimisation, à l'aide d'approches itératives, de la différence entre le filtre “voulu“ et le filtre obtenu

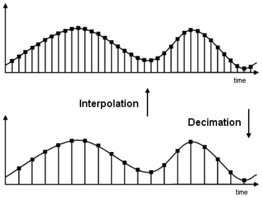
Exemples dans Matlab :

- `fircls` uses an iterative least-squares algorithm to obtain an equiripple response. The algorithm is a multiple exchange algorithm that uses Lagrange multipliers and Kuhn-Tucker conditions on each iteration.
- `firpm` designs a linear-phase FIR filter using the Parks-McClellan algorithm. The Parks-McClellan algorithm uses the Remez exchange algorithm and Chebyshev approximation theory to design filters with an optimal fit between the desired and actual frequency responses.

Outline

Décimation

- La décimation est la compression d'un signal
- Elle correspond à réduire son taux d'échantillonnage
- Le calcul du spectre après décimation est équivalent à un échantillonnage à la nouvelle fréquence d'échantillonnage. Par exemple : décimer par 2 un signal échantillonné à 20 kHz, revient à l'échantillonner à 10 kHz.

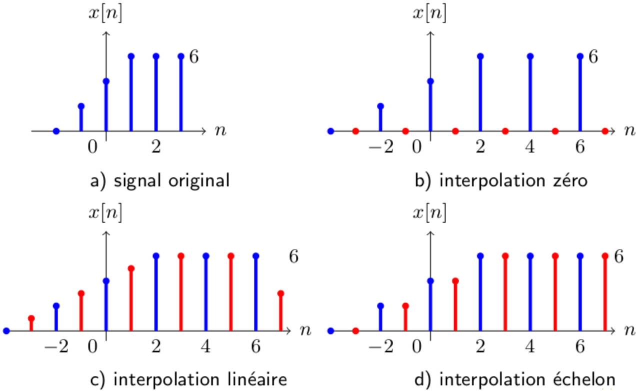


Interpolation

- L'interpolation est l'étirement d'un signal.
- Elle correspond à augmenter son taux d'échantillonnage
- Exemple : soit un signal $x[n] = y[2n]$
  - $y[n] = x(n/2)$ est une version allongée de $x[n]$
  - Elle correspond à rajouter un échantillon entre tout échantillon de $x[n]$
- Le signal aura plus d'échantillons que le signal d'origine

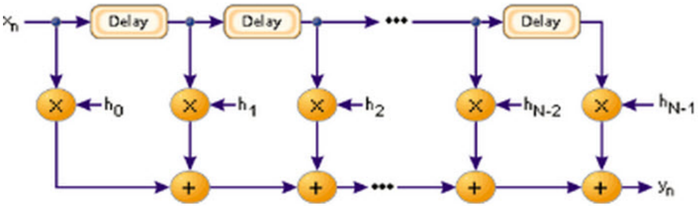
Interpolation

- Il existe plusieurs approches pour faire l’interpolation
- Le choix est un compromis entre la performance et la complexité



Outline

Implémentation

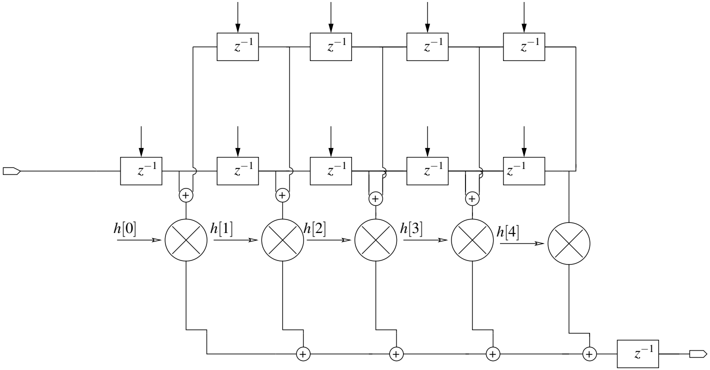


L'implémentation des filtres nécessitent :

- Des retards
- Des multiplications avec les coefficients h_i
- Des sommateurs

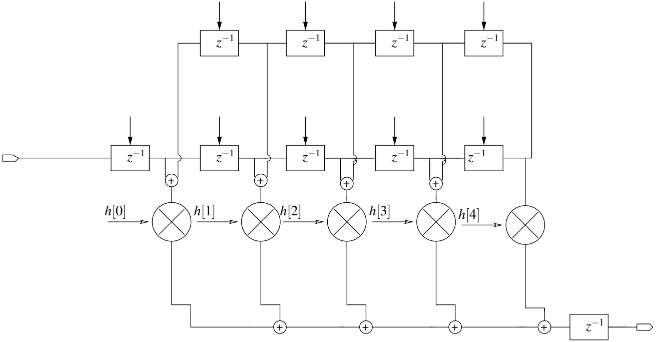
Architecture symmétrique

if $h[0] = h[8]$, $h[1] = h[7]$, $h[2] = h[6]$, $h[3] = h[5]$



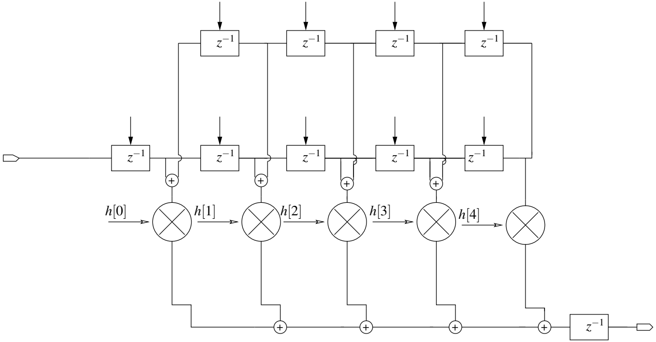
N/2 instead of N multipliers

Filtre sans multiplieur



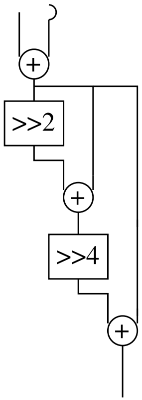
Filtre sans multiplieur

$$h[0] = 81 = 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$



Filtre sans multiplieur

$$h[0] = 81 = 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$



Canonical Signed Digit (CSD)

number	2's complement	CSD
3	011	$10\bar{1}$
2	011	010
1	001	001
0	000	000
-1	111	$00\bar{1}$
-2	110	$0\bar{1}0$
-3	101	$\bar{1}01$
-4	100	$\bar{1}00$

33% de bits # 0 pour CSD contre 50% pour les encodages classiques

Exemple

classique $79 = 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

CSD $79 = 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 - 1 \times 2^0$

Outline

Conclusion

- Analyse d'un filtre numérique consiste à extraire les caractéristiques du filtre :
  - La fonction de transfert H(z)
  - La réponse impulsionnelle et indicielle
  - Stabilité
  - Réjection, l'ondulation de la réponse fréquentielle
- La construction des filtres peut se faire à l'aide de méthodes analytiques (Euler, Bilatéral, IFFT)
- La construction des filtres peut se faire aussi à l'aide de méthodes numériques (Least square ou Parks–McClellan)
- Pour éviter le recouvrement spectral pendant la décimation, les composantes supérieures à $\frac{1}{2 \cdot M}$ doivent être filtrées au préalable

Conclusion FIR vs IIR

	Filtre IIR	Filtre FIR
Stabilité	Risque d'instabilité	Pas de rique
Méthodes	Transformation d'Euler Transformation bilinéaire	IFFT
Nombre de Coefficients	Faible	élevé
Complexité	Faible	Faible
Phase	Difficile à contrôler	Plutôt Facile à contrôler