

Introduction à la Synthèse sonore

Cours 1 Vibrations dans le temps

Matthias Puech

`matthias.puech@lecnam.net`

STMN — Cnam ENJMIN, Angoulême

21-22 novembre 2017

Synthèse sonore

Definition

Ensemble de techniques permettant la génération de signaux sonores.

Synthèse sonore

Definition

Ensemble de techniques permettant la génération de signaux sonores.



Synthèse sonore

Definition

Ensemble de techniques permettant la génération de signaux sonores.



Synthèse sonore

Definition

Ensemble de techniques permettant la génération de signaux sonores.



Synthèse sonore

Notre définition

ensemble de **traitements numériques** et **algorithmes** permettant la génération de signaux sonores par un **microprocesseur**.

Synthèse sonore

Notre définition

ensemble de **traitements numériques** et **algorithmes** permettant la génération de signaux sonores par un **microprocesseur**.

Applications

- musicale
 - ▶ jouer n'importe quel instrument sur n'importe quel instrument
 - ▶ sons inédits
- illustration sonore
 - ▶ effets (cinéma)
 - ▶ design sonore (ex : indicatif SNCF, voiture électrique)
 - ▶ sonification
- scientifique
 - ▶ simulation d'instruments physique
 - ▶ analyse acoustique

Synthèse sonore

Les outils

No silver bullet

- des outils mathématiques
(théorie de Fourier, filtrage)
- des algorithmes et des principes
(synthèse additive, granulaire, FFT...)
- l'oreille comme seule guide

Synthèse sonore

Les outils

No silver bullet

- des outils mathématiques
(théorie de Fourier, filtrage)
- des algorithmes et des principes
(synthèse additive, granulaire, FFT...)
- l'oreille comme seule guide

But de ce cours

- comprendre les bases de l'audio numérique
- connaître les grandes méthodes de synthèse sonore
- générer un flux sonore en temps réel, en fonction de signaux

Aspects pratiques

`http://cedric.cnam.fr/~puechm/ens/synthese`

Aspects pratiques

<http://cedric.cnam.fr/~puechm/ens/synthese>

Cours

En 3 séances :

Cours 1 (4h) Vibrations dans le temps

Cours 2 (3h) Signaux : modulation et filtrage

Cours 3 (3h) Techniques et algorithmes

Aspects pratiques

<http://cedric.cnam.fr/~puechm/ens/synthese>

Cours

En 3 séances :

Cours 1 (4h) Vibrations dans le temps

Cours 2 (3h) Signaux : modulation et filtrage

Cours 3 (3h) Techniques et algorithmes

Travaux Pratiques

- deuxième moitié de chaque séance pour pratiquer
- feuille de TP sur la page du cours
- dernier TP noté
(monome ou binôme, à rendre à matthias.puech@lecnam.net)

Synthèse sonore

Introduction

Amplitude, fréquence, timbre

Le son et sa perception

Signaux périodiques

La théorie de Fourier

Introduction à Pure Data

Boîtes et objets

Sinusoides

Enveloppes

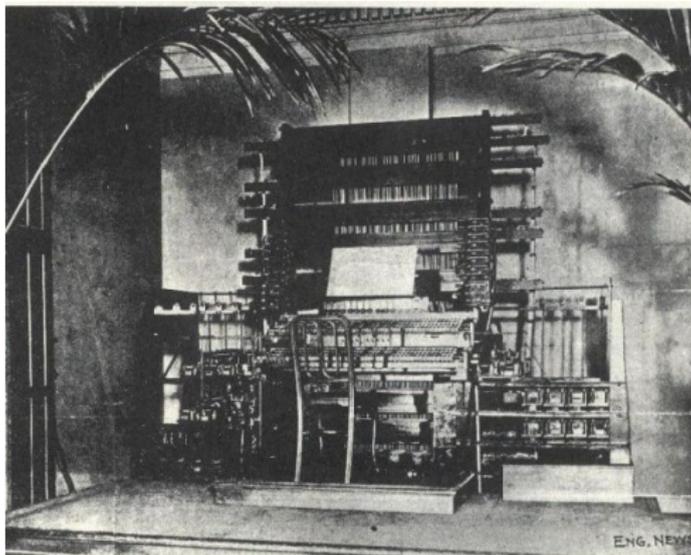
Tables d'ondes

Bibliographie

- *The Theory and Technique of Electronic Music*, Miller Puckette
<http://msp.ucsd.edu/techniques.htm>
- Tutoriel Pure Data FLOSS Manuals
<https://fr.flossmanuals.net/puredata/introduction/>
- *The Computer Music Tutorial*, Curtis Roads
- *Les Fous du Son*, Laurent de Wilde

Histoire sélective

1897 Le Telharmonium



Thaddeus Cahill

- électromécanique : principe de la roue phonique
- polyphonique et clavier dynamique

Histoire sélective

1919 Le Thérémin



Lev Sergeïevitch Termen

- <https://www.youtube.com/watch?v=K6KbEnGnymk>

Histoire sélective

1928 Les Ondes Martenot

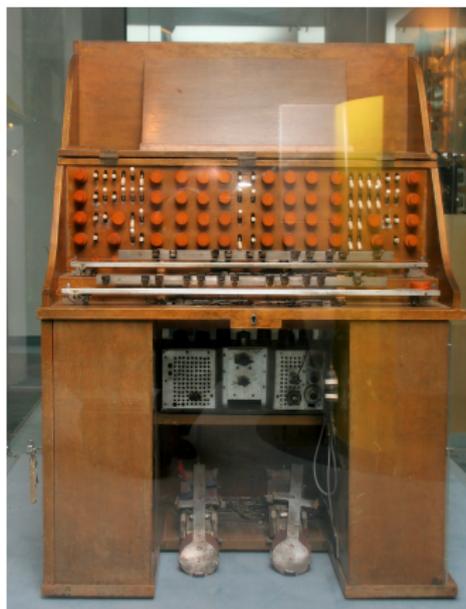


Maurice Martenot

- monophonique, contrôlé par “bague”, clavier et bouton
- <https://www.youtube.com/watch?v=1nckmibkVgI>

Histoire sélective

1930 Le Trautonium

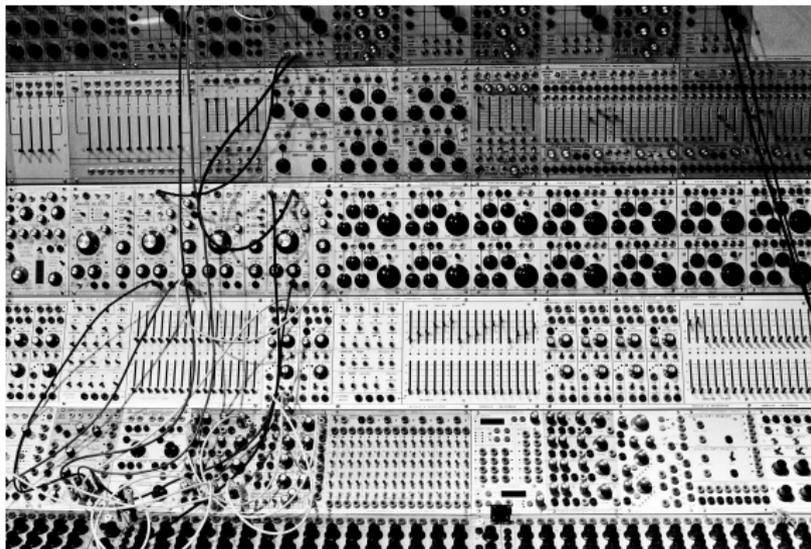


Friedrich Trautwein

- timbre modulable de façon continue, grande palette
- <https://www.youtube.com/watch?v=-tQQEChMq1A>

Histoire sélective

1962 Le synthétiseur modulaire



Don Buchla & Bob Moog

- configuration et chemin du signal non fixé
- <https://www.youtube.com/watch?v=2II0dxgQurM>

Histoire sélective

1970 Minimoog



- monophonique, clavier intégré
- configuration fixée (peu de paramètres) : VCO→VCF→VCA
- <https://www.youtube.com/watch?v=-YO8GrOaAcw>

Histoire sélective

1979 Fairlight CMI



- premier échantillonneur (sampler)
- entièrement numérique et programmé (OS dédié)
- <https://www.youtube.com/watch?v=i01PCpSmhRM>

Histoire sélective

1983 Synthèse FM et DX7



John Chowning & Yamaha

- nouvelle méthode de génération de timbres
- production numérique du son
- <https://www.youtube.com/watch?v=F3rrjQtQe5A>

Histoire sélective

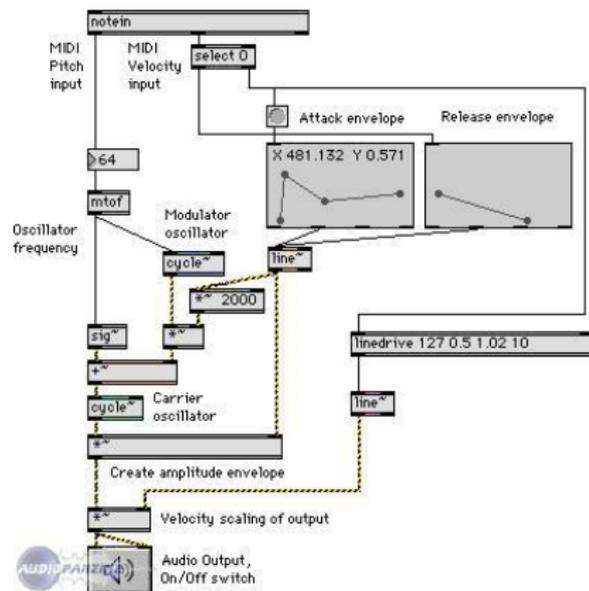
1984 Stations de calcul de l'Ircam (4x, ISPW)



- microprocesseurs dédiés à la synthèse sonore
- temps différé

Histoire sélective

1990 Max/MSP



- langage de programmation graphique
- temps réel, tourne sur MCU généraliste
- encore développé aujourd'hui par *Cycling'74*

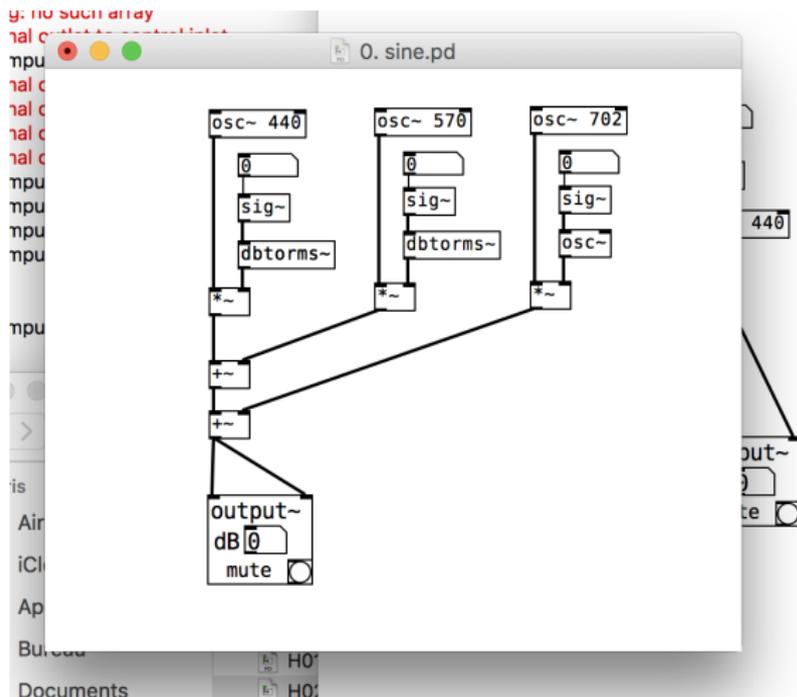
Outils de synthèse numérique

On utilise aujourd'hui des logiciels de :

- séquençage et montage audio
(Pro Tools, Ableton Live, Logic Pro...)
- sampling
(Kontakt, Ableton Sampler...)
- reproduction virtuelle d'instruments hardware
(ReBirth, plugins Native Instruments,...)
- effets : réverbération, compression, filtrage, distortion
(plugins UAD, Waves, ...)
- programmation audio
(Pure Data, Max/MSP, Reaktor...)

Pure Data

C'est l'outil que l'on va utiliser dans ce cours



Miller Puckette (1996)

Pure Data

- langage de programmation graphique / data flow
(programme = réseau de boîtes)
- multi-plateforme
(GNU/Linux, Mac OS X, iOS, Android, Windows)
- cousin de Max/MSP
- libre et gratuit
- temps réel et interactif
(pas de cycle édition→compilation→exécution)
- manipule des flux audio et de contrôle
- interaction hardware et software
(UI, OSC, MIDI...)
- possibilité de manipuler des flux vidéo
(bibliothèque GEM)

Synthèse sonore

Introduction

Amplitude, fréquence, timbre

Le son et sa perception

Signaux périodiques

La théorie de Fourier

Introduction à Pure Data

Boîtes et objets

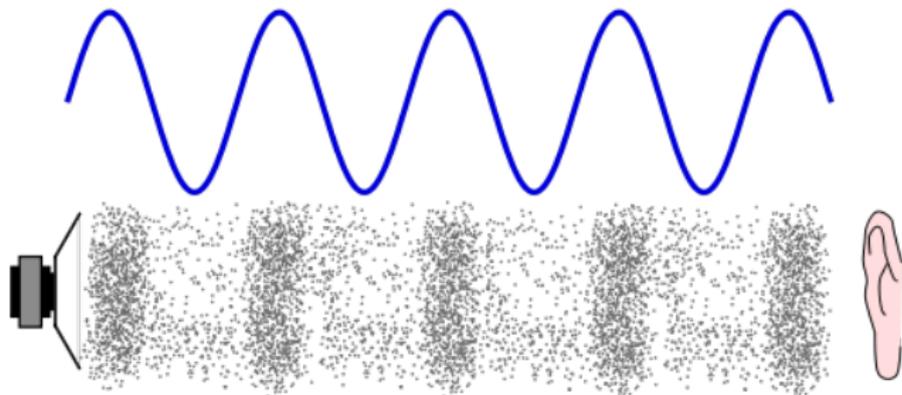
Sinusoïdes

Enveloppes

Tables d'ondes

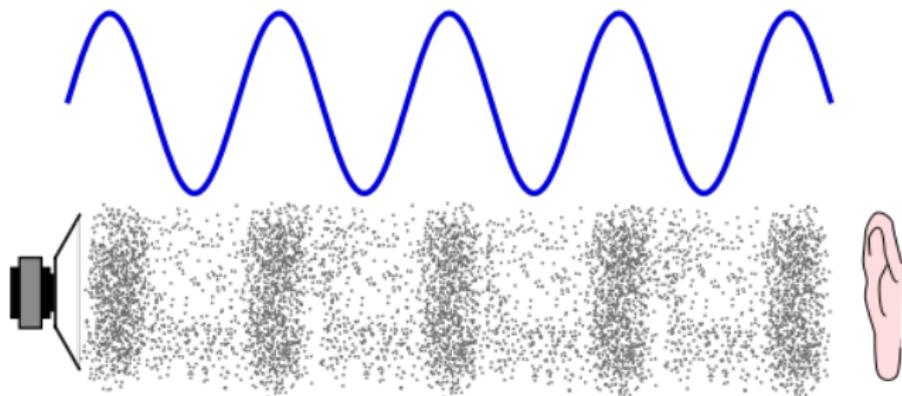
Qu'est-ce qu'un son ?

Un son est la propagation une onde *audible* de pression dans l'air



Qu'est-ce qu'un son ?

Un son est la propagation une onde *audible* de pression dans l'air



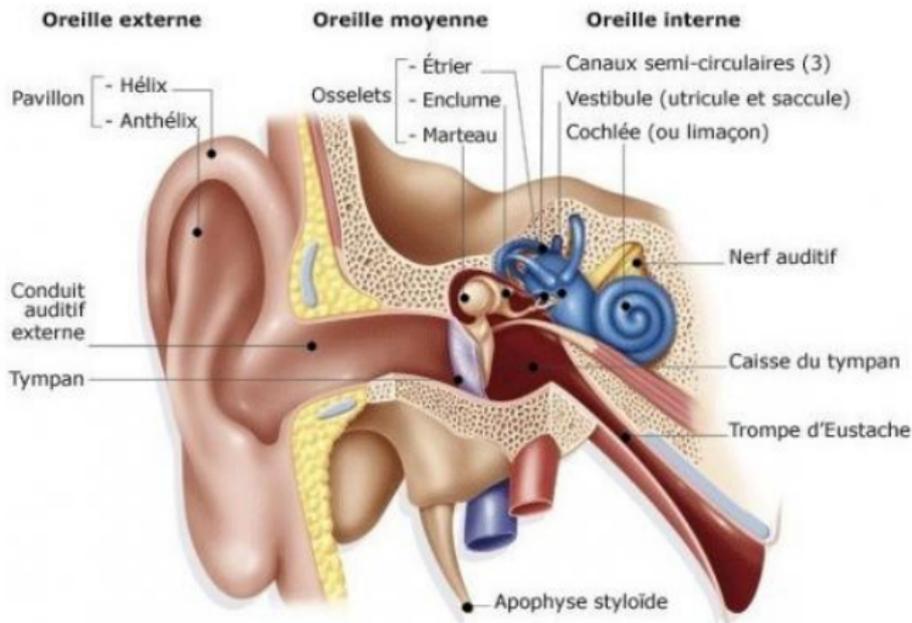
Remarques

- *audible* = 20Hz – 20kHz
- ...ou dans un autre milieu (liquide, gaz, solide)

Anatomie

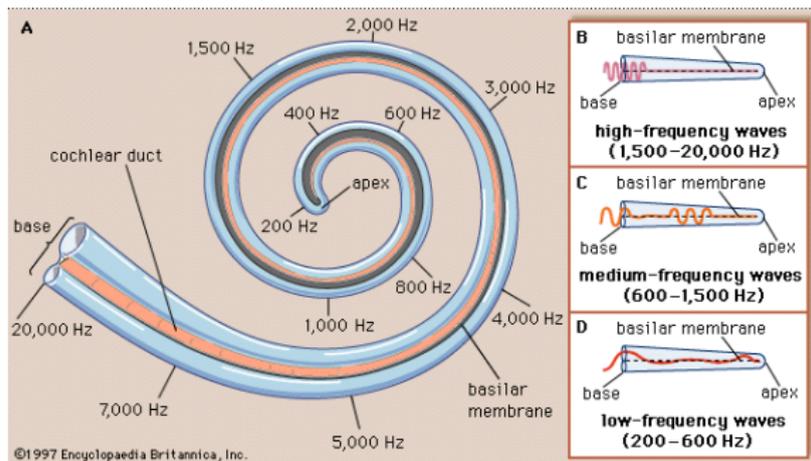
L'oreille humaine

Convertit l'onde de pression en une information électrique transmise au cerveau (nerf auditif)



Anatomie

La cochlée



- différentes fréquences résonnent à différentes “profondeurs” (base=20kHz, apex=20Hz)
- tout son long, des *cils vibratils* créent du courant électrique
- le cerveau reçoit l’intensité pour chaque profondeur (à une fréquence $\approx 50\text{Hz}$)

Anatomie

Quiz

Quelle onde sonore excite exactement 1 endroit de la cochlée ?

Anatomie

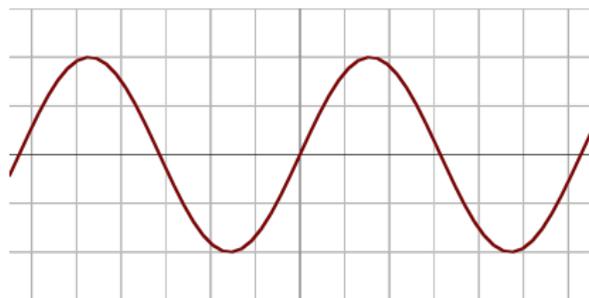
Quiz

Quelle onde sonore excite exactement 1 endroit de la cochlée ?

Réponse

La sinusoïde :

$$s(t) = A \cos(2\pi ft + \phi)$$

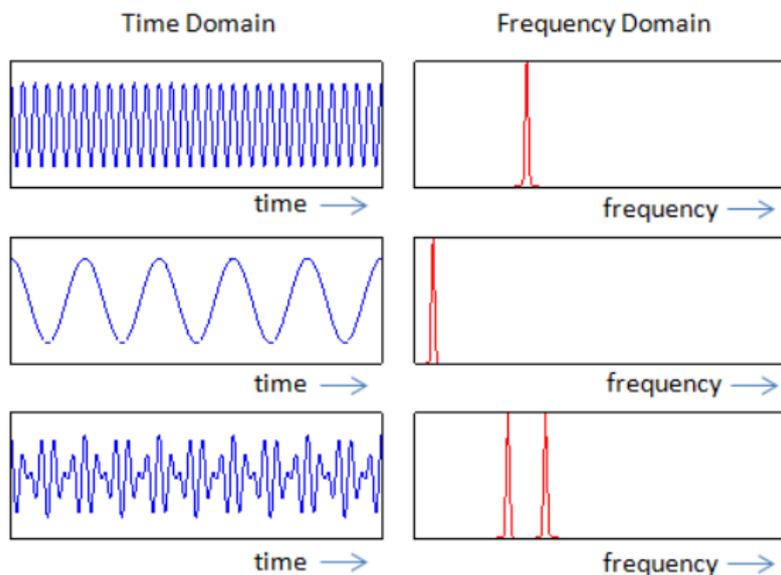


<https://www.desmos.com/calculator/tlnztjjhjm>

- A amplitude (Pa)
- f fréquence (Hz)
- ϕ phase (rad)

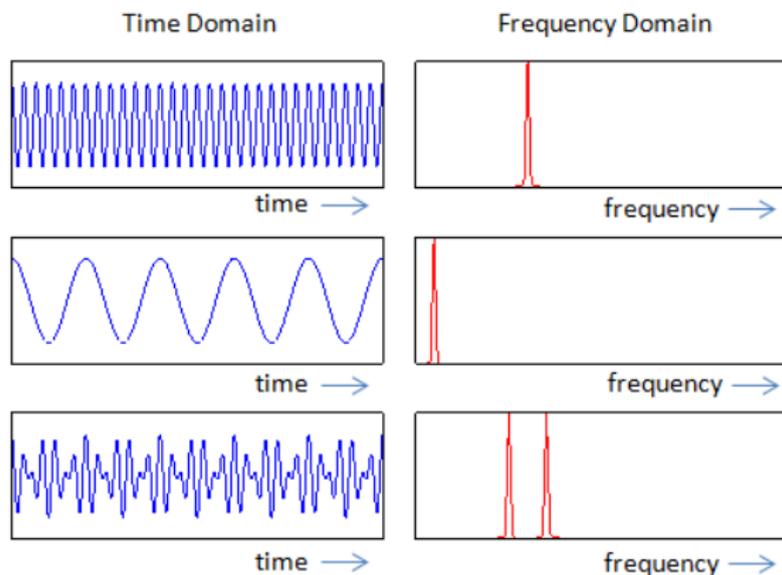
00.domains.pd Domaines fréquentiel et temporel

Il y a donc deux façons de représenter les sons :



00.domains.pd Domaines fréquentiel et temporel

Il y a donc deux façons de représenter les sons :



“on entend dans le domaine fréquentiel”

- les événements $f < 50\text{Hz}$ sont distinguables dans le temps
- les événements $20\text{Hz} < f < 20\text{kHz}$ sont continus et ont une hauteur

Réponse en fréquence & amplitude

Réponse en fréquence

La cochlée capte plus de fréquences à son apex qu'à sa base

Exemple à la base, 1mm = 1000 Hz, à l'apex, 1mm = 10 Hz¹

1. les quantités sont arbitraires, pour illustration

Réponse en fréquence & amplitude

Réponse en fréquence

La cochlée capte plus de fréquences à son apex qu'à sa base

Exemple à la base, 1mm = 1000 Hz, à l'apex, 1mm = 10 Hz¹

↪ *réponse en fréquence exponentielle*

Unité relative 1 octave = rapport de fréquence de 2

1 demi-ton = rapport de fréquence de $\sqrt[12]{2}$

Unité absolue la *note midi* : nombre de demi-tons au dessus de 8.1757989156Hz (“do 0”)

Exemple la note midi 69 a la fréquence $8.1757989156 * (\sqrt[12]{2})^{69} = 440\text{Hz}$

1. les quantités sont arbitraires, pour illustration

Réponse en fréquence & amplitude

Réponse en amplitude

Les cils vibratiles produisent proportionnellement plus de courant avec des petites amplitudes qu'avec des grandes

Exemple pour $1\text{nPa}\cdot\text{s}^{-1}$ de différence de pression, $1\mu\text{V}$;
pour $10\text{nPa}\cdot\text{s}^{-1}$, seulement $2\mu\text{V}$

1. les quantités sont arbitraires, pour illustration

Réponse en fréquence & amplitude

Réponse en amplitude

Les cils vibratiles produisent proportionnellement plus de courant avec des petites amplitudes qu'avec des grandes

Exemple pour $1\text{nPa}\cdot\text{s}^{-1}$ de différence de pression, $1\mu\text{V}$;
pour $10\text{nPa}\cdot\text{s}^{-1}$, seulement $2\mu\text{V}$

↪ *réponse en amplitude exponentielle*

Unité relative 1 dB = rapport d'amplitude de 1,26
3 dB = rapport d'amplitude de 2

Unité absolue n dB SPL = n dB au dessus de la pression la plus faible perceptible par l'oreille humaine : $20\mu\text{Pa}$

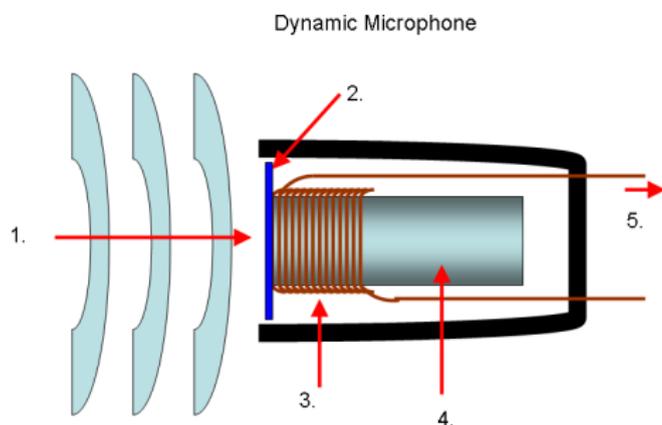
Exemple 50 dB SPL = voix humaine ;
120 dB SPL = seuil de douleur

1. les quantités sont arbitraires, pour illustration

Les transducteurs artificiels

Microphone

Transforme le signal acoustique (Pa) en signal électrique (V).

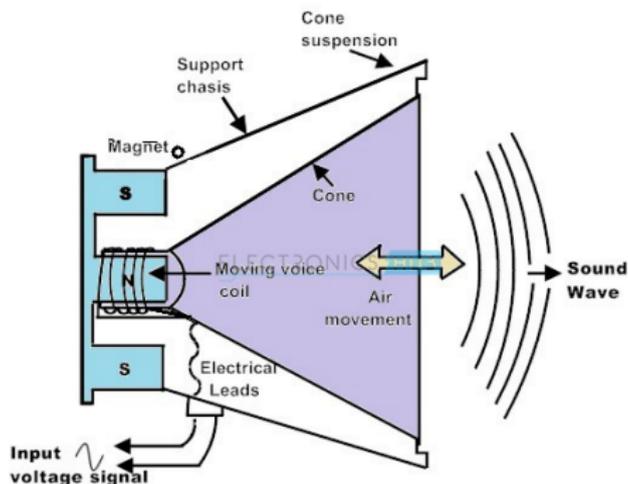


- l'air fait vibrer une membrane reliée à une bobine
- la bobine crée un courant électrique induit
- transformation *analogique*
(variation de pression proportionnelle au courant)

Les transducteurs artificiels

Haut-parleur

Transforme le signal **électrique** (V) en signal **acoustique** (Pa).



- le courant crée un champ magnétique dans la bobine
- la bobine se déplace, et déplace le cône
- le cône déplace l'air, qui crée une différence de pression

Amplification/atténuation d'un signal sonores

Quiz

Comment modifier le signal $s(t)$ pour qu'il nous parvienne

- plus fort ?
- moins fort ?

Amplification/atténuation d'un signal sonores

Quiz

Comment modifier le signal $s(t)$ pour qu'il nous parvienne

- plus fort ?
- moins fort ?

Réponse

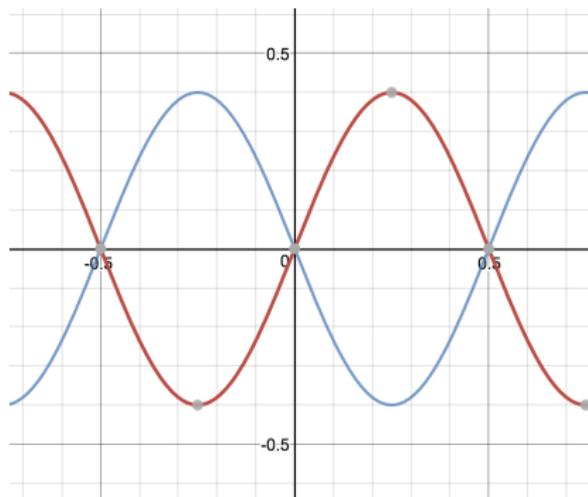
Il faut le *multiplier* par un signal de valeur constante :

- $\nu > 1$ pour l'amplifier
- $0 < \nu < 1$ pour l'atténuer

Amplification/atténuation d'un signal sonores

Quiz

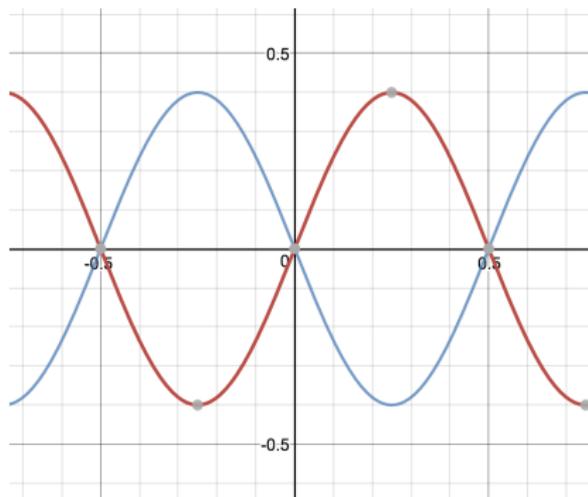
Qu'entend-on si on multiplie un signal sonore par -1 ?



Amplification/atténuation d'un signal sonores

Quiz

Qu'entend-on si on multiplie un signal sonore par -1 ?



Réponse

La même chose !

L'oreille n'est pas sensible aux différences de phase.

Mixage de signaux sonores

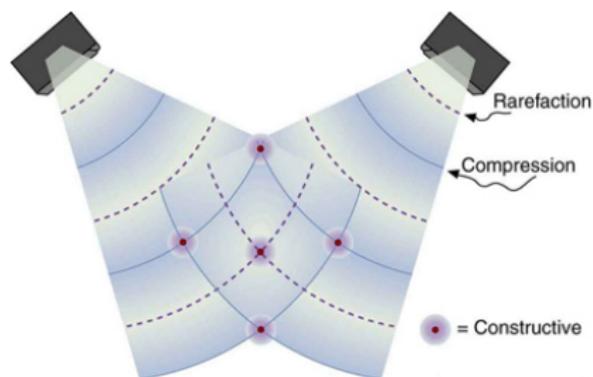
Quiz

Qu'entend-on quand on est à la même distance de deux sources sonores $s_1(t)$ et $s_2(t)$?

Mixage de signaux sonores

Quiz

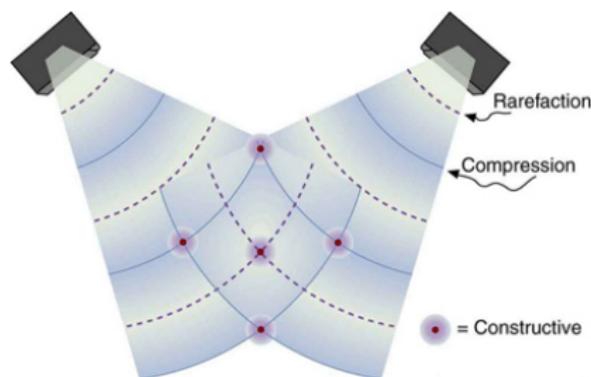
Qu'entend-on quand on est à la même distance de deux sources sonores $s_1(t)$ et $s_2(t)$?



Mixage de signaux sonores

Quiz

Qu'entend-on quand on est à la même distance de deux sources sonores $s_1(t)$ et $s_2(t)$?



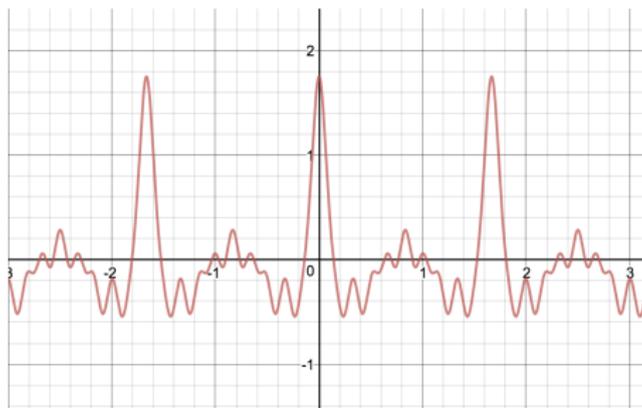
Réponse

La *somme* des deux signaux $s_1(t) + s_2(t)$

↪ “mixer” deux source, c’est faire leur somme

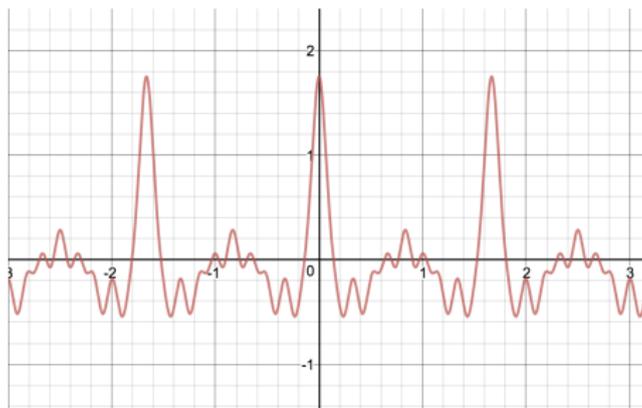
Signal (a)périodique

Un signal *périodique* se répète infiniment avec une période T , et une fréquence $\frac{1}{T}$:



Signal (a)périodique

Un signal *périodique* se répète infiniment avec une période T , et une fréquence $\frac{1}{T}$:



Remarques

- aucun son n'est périodique (de $-\infty$ à $+\infty$)
(mais *pseudo-périodique*)
- si sa fréquence est $< 50\text{Hz}$, il est perçu comme ayant une *hauteur*

Signal (a)périodique

Un signal *apériodique* ne se répète jamais (bruit)

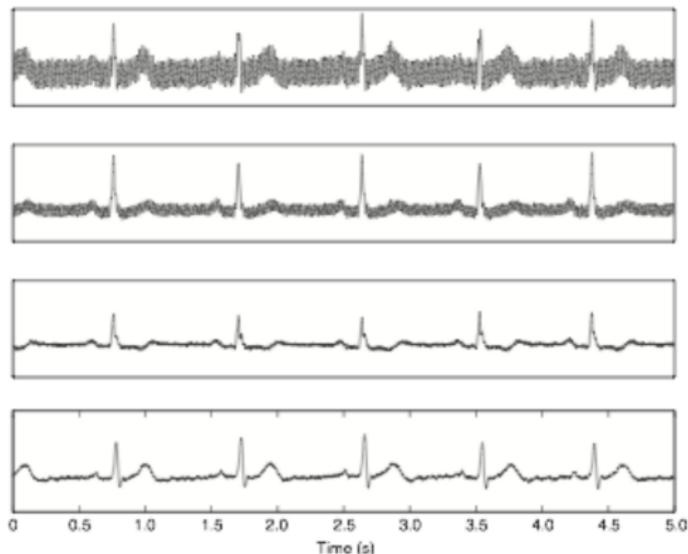


FIGURE 1.1 Three-lead electrocardiograph (ECG) signals. Note the significant amount of interference (noise) in the raw signals (top three traces).

La théorie de Fourier



Joseph Fourier (1768–1830)

“Toute fonction périodique $s(t)$ de fréquence f_0 , peut être décomposée comme une somme infinie de sinusoides de fréquences multiples de f_0 .”

$$s(t) = \sum_{k=0}^{+\infty} A_k \cos(2\pi k f_0 t + \phi_k)$$

La théorie de Fourier

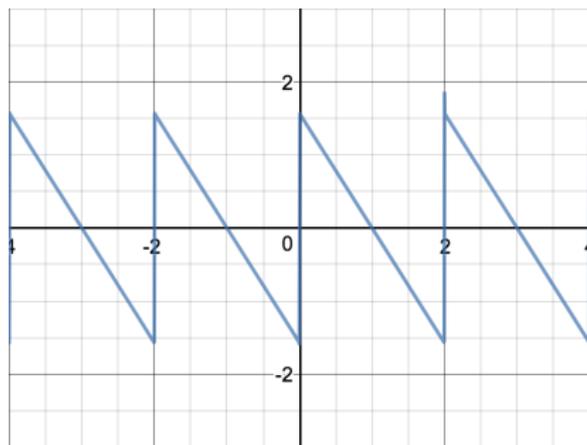
Exemple

La fonction *dent de scie* :

$$s(t) = (-t \bmod 1) \cdot 2 - 1$$

peut s'écrire :

$$s(t) = \sum_{k=0}^{+\infty} \frac{1}{k} \cos\left(2\pi kt + \frac{\pi}{2}\right)$$



La théorie de Fourier

$$s(t) = \sum_{k=0}^{+\infty} A_k \cos(2\pi k f_0 t + \phi_k)$$

Conclusion

Un son périodique n'est caractérisé que par les amplitudes A_k de ses **harmoniques**.

(les phases ϕ_k ne comptent pas pour l'oreille)

Des noms pour les harmoniques

$k = 0$ la *composante continue*

$k = 1$ la *fréquence fondamentale*

$k > 1$ le $(k - 1)$ ème *partiel*

Notion de timbre

Le *timbre* est la “couleur” d’un son périodique

- différencie le son d’une flûte de celui d’un violon (à fréquence égale)
- déterminé par l’amplitude A_k des harmoniques
- on visualise son *spectre* à un instant donné (*spectrogramme*)

Exemples

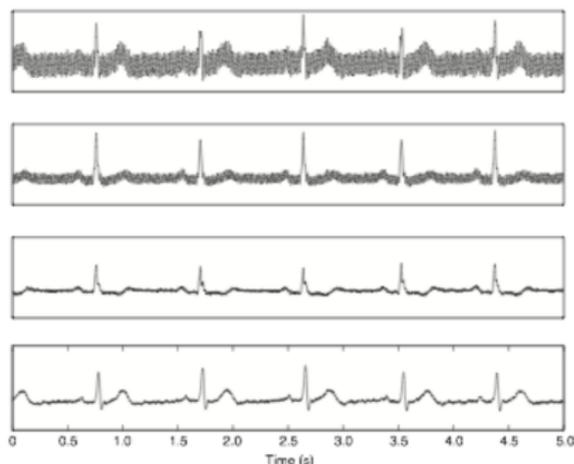
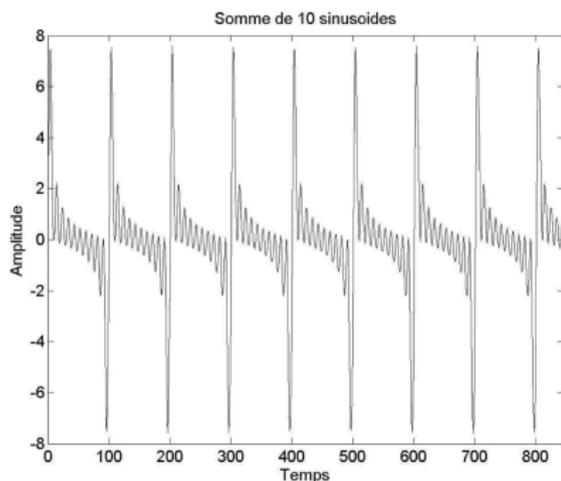


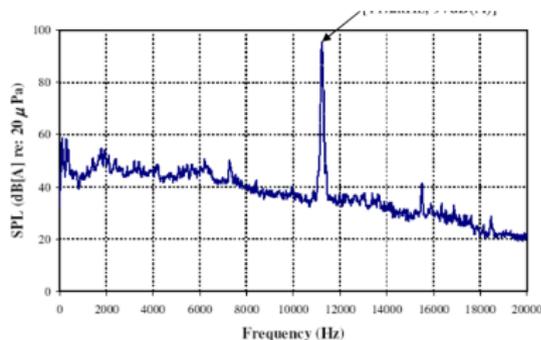
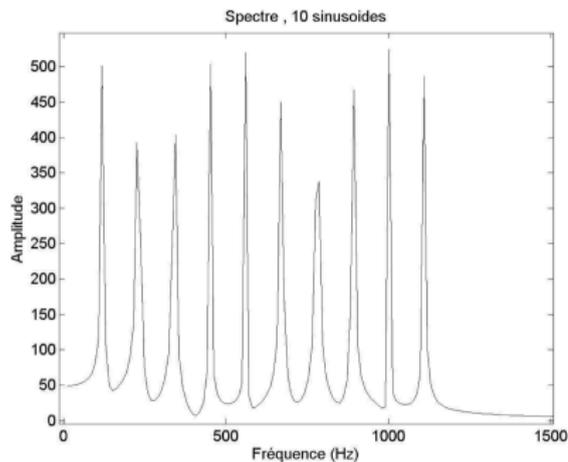
FIGURE 1.1 Three lead electrocardiograph (ECG) signals. Note the significant amount of interference (noise) in the raw signals (top three traces).

Notion de timbre

Le *timbre* est la “couleur” d’un son périodique

- différencie le son d’une flûte de celui d’un violon (à fréquence égale)
- déterminé par l’amplitude A_k des harmoniques
- on visualise son *spectre* à un instant donné (*spectrogramme*)

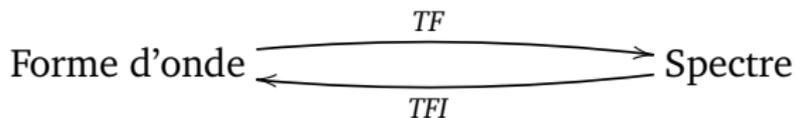
Exemples



La transformée de Fourier (inverse)

TF Transforme une forme d'onde $x(t)$ en son spectre $X(f)$

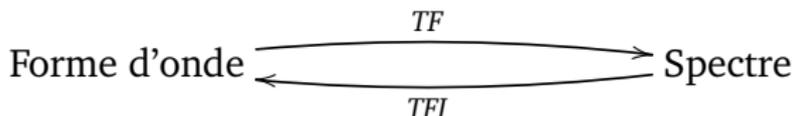
TFI fait l'inverse



La transformée de Fourier (inverse)

TF Transforme une forme d'onde $x(t)$ en son spectre $X(f)$

TFI fait l'inverse



Expression mathématique

Pour les mathématiciens :

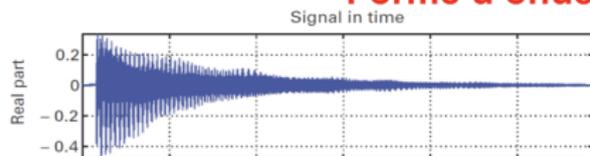
$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-i2\pi ft} dt$$

(I)FFT : (Inverse) Fast Fourier Transform

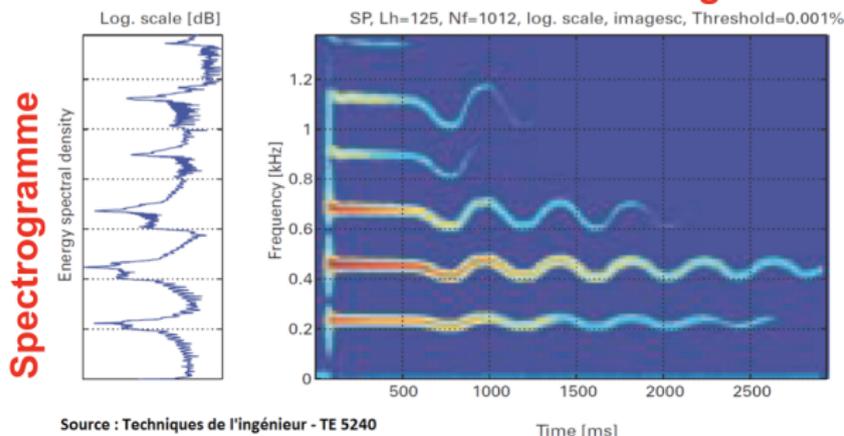
Son algorithme le plus connu

Le sonagramme

Forme d'onde



Sonagramme



Combine 2 représentations en une en 3D :

- forme d'onde (amplitude/temps)
- spectrogramme (amplitude/fréquence à un temps donné)

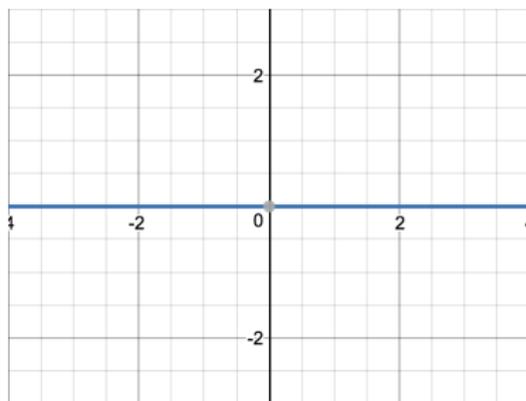
Synthèse additive

Votre premier algorithme de synthèse

Additionnez des sinusoides de fréquences $f_0, 2f_0, 3f_0$ etc. et d'amplitude différentes pour obtenir un timbre.

Exemple (la dent de scie)

https://en.wikipedia.org/wiki/File:Additive_220Hz_Sawtooth_Wave.wav



$$s(t) = \sum_{k=0}^{\infty} \frac{1}{k} \cos\left(2\pi kt + \frac{\pi}{2}\right)$$

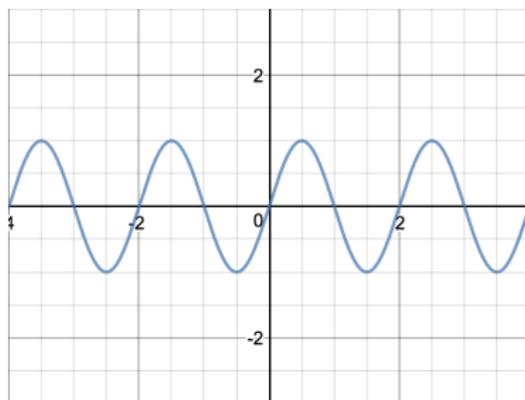
Synthèse additive

Votre premier algorithme de synthèse

Additionnez des sinusoïdes de fréquences $f_0, 2f_0, 3f_0$ etc. et d'amplitude différentes pour obtenir un timbre.

Exemple (la dent de scie)

https://en.wikipedia.org/wiki/File:Additive_220Hz_Sawtooth_Wave.wav



$$s(t) = \sum_{k=0}^{\infty} \frac{1}{k} \cos\left(2\pi kt + \frac{\pi}{2}\right)$$

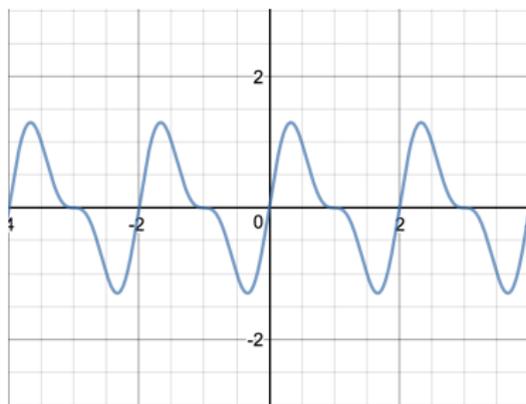
Synthèse additive

Votre premier algorithme de synthèse

Additionnez des sinusoïdes de fréquences $f_0, 2f_0, 3f_0$ etc. et d'amplitude différentes pour obtenir un timbre.

Exemple (la dent de scie)

https://en.wikipedia.org/wiki/File:Additive_220Hz_Sawtooth_Wave.wav



$$s(t) = \sum_{k=0}^2 \frac{1}{k} \cos\left(2\pi kt + \frac{\pi}{2}\right)$$

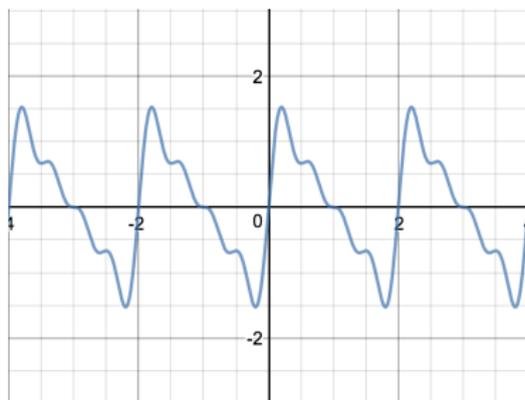
Synthèse additive

Votre premier algorithme de synthèse

Additionnez des sinusoïdes de fréquences $f_0, 2f_0, 3f_0$ etc. et d'amplitude différentes pour obtenir un timbre.

Exemple (la dent de scie)

https://en.wikipedia.org/wiki/File:Additive_220Hz_Sawtooth_Wave.wav



$$s(t) = \sum_{k=0}^4 \frac{1}{k} \cos\left(2\pi kt + \frac{\pi}{2}\right)$$

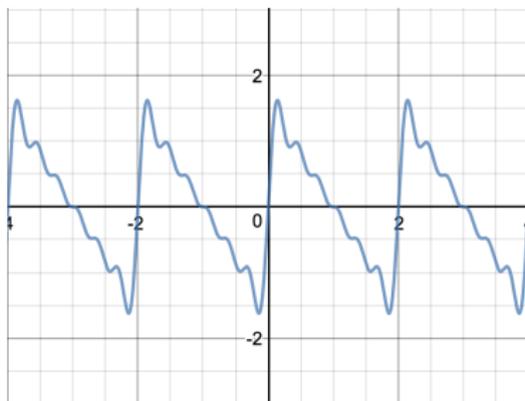
Synthèse additive

Votre premier algorithme de synthèse

Additionnez des sinusoides de fréquences $f_0, 2f_0, 3f_0$ etc. et d'amplitude différentes pour obtenir un timbre.

Exemple (la dent de scie)

https://en.wikipedia.org/wiki/File:Additive_220Hz_Sawtooth_Wave.wav



$$s(t) = \sum_{k=0}^6 \frac{1}{k} \cos\left(2\pi kt + \frac{\pi}{2}\right)$$

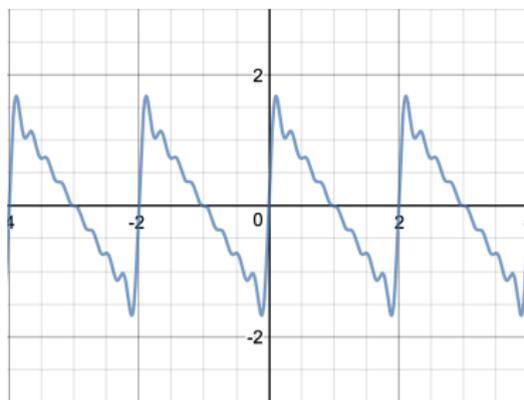
Synthèse additive

Votre premier algorithme de synthèse

Additionnez des sinusoïdes de fréquences $f_0, 2f_0, 3f_0$ etc. et d'amplitude différentes pour obtenir un timbre.

Exemple (la dent de scie)

https://en.wikipedia.org/wiki/File:Additive_220Hz_Sawtooth_Wave.wav



$$s(t) = \sum_{k=0}^8 \frac{1}{k} \cos\left(2\pi kt + \frac{\pi}{2}\right)$$

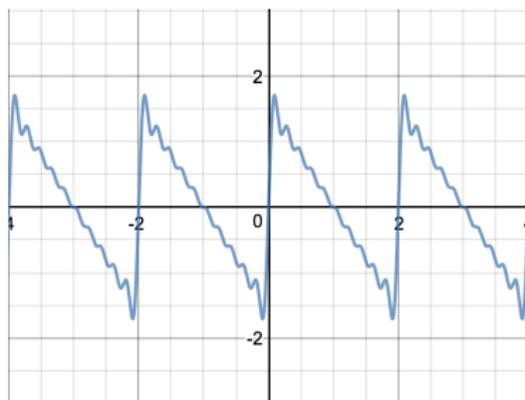
Synthèse additive

Votre premier algorithme de synthèse

Additionnez des sinusoïdes de fréquences $f_0, 2f_0, 3f_0$ etc. et d'amplitude différentes pour obtenir un timbre.

Exemple (la dent de scie)

https://en.wikipedia.org/wiki/File:Additive_220Hz_Sawtooth_Wave.wav



$$s(t) = \sum_{k=0}^{10} \frac{1}{k} \cos\left(2\pi kt + \frac{\pi}{2}\right)$$

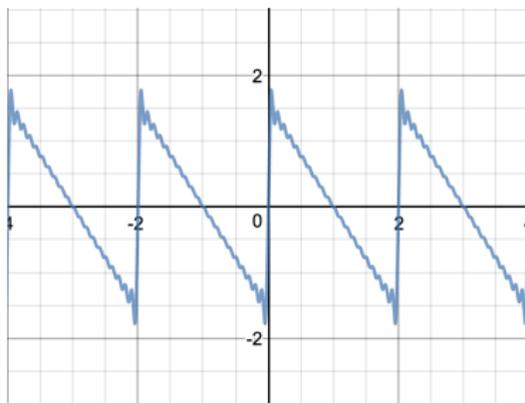
Synthèse additive

Votre premier algorithme de synthèse

Additionnez des sinusoïdes de fréquences $f_0, 2f_0, 3f_0$ etc. et d'amplitude différentes pour obtenir un timbre.

Exemple (la dent de scie)

https://en.wikipedia.org/wiki/File:Additive_220Hz_Sawtooth_Wave.wav



$$s(t) = \sum_{k=0}^{20} \frac{1}{k} \cos\left(2\pi kt + \frac{\pi}{2}\right)$$

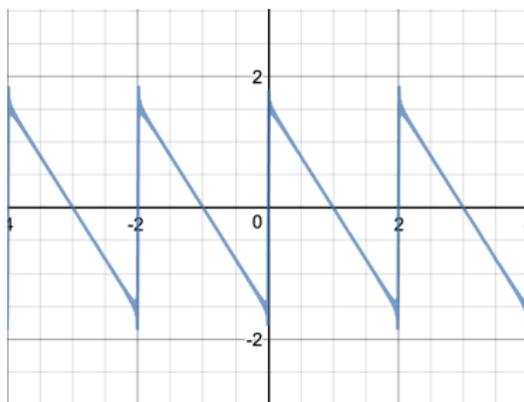
Synthèse additive

Votre premier algorithme de synthèse

Additionnez des sinusoides de fréquences $f_0, 2f_0, 3f_0$ etc. et d'amplitude différentes pour obtenir un timbre.

Exemple (la dent de scie)

https://en.wikipedia.org/wiki/File:Additive_220Hz_Sawtooth_Wave.wav



$$s(t) = \sum_{k=0}^{50} \frac{1}{k} \cos\left(2\pi kt + \frac{\pi}{2}\right)$$

Synthèse sonore

Introduction

Amplitude, fréquence, timbre

Le son et sa perception

Signaux périodiques

La théorie de Fourier

Introduction à Pure Data

Boîtes et objets

Sinusoïdes

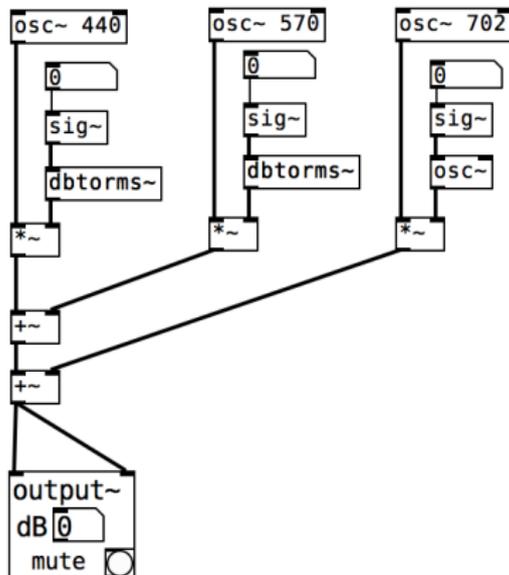
Enveloppes

Tables d'ondes

Le patch

Un *patch* est un programme réactif
(tourne en boucle et réagit aux événements)

- des *boîtes* reliées par des *fils*
- certaines boîtes sont interactives
- dans les fils circulent :
 - ▶ *signaux* (fils épais), ou
 - ▶ *messages* (fils fins)

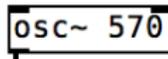


Signaux vs messages

- un signal est continu
(toujours de l'information,
nombres flottants)
- un message est un discret
(événement qui contient une
chaîne de caractère)

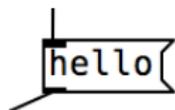
Types de boîtes

objet traite message ou signal selon sa fonction



- nom : `osc~`, argument : 570 (sinusoïde à 570 Hz)
- 2 entrées, 1 sortie
- ~100 objets dans pd, + librairies

message envoie message sur sa sortie



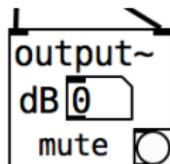
- activé par clic
- ... ou réception du message "bang"

nombre envoie un nombre sur sa sortie



- activé par drag-and-drop, entrée manuelle,
- ... ou réception du message "bang"

objet complexe contient d'autres boîtes (UI)



Interface utilisateur

Le patch reflète le programme et son exécution
(en modifiant le patch on observe instantanément les changements)

Mode édition

- off** on interagit avec le patch
(clic sur message, change nombres...)
- on** on édite le patch à la souris
(*astuce* : en maintenant Ctrl on peut interagir avec)

Raccourcis clavier

- Ctrl-E** (dés-)active le mode édition
- Ctrl-1** nouvel objet
- Ctrl-2** nouveau message
- Ctrl-3** nouveau nombre
- Ctrl-D** duplique les éléments sélectionnés

01.sine.pd Addition et multiplication de sinusoides

Objets utilisés

Par convention, les objets finissant par `~` émettent des **signaux**

`osc~` génère une sinusoïde

- entrée 1 ou argument = sa fréquence (Hz)

`+~` additionne deux signaux

- argument = signal constant en entrée 2

`*~` multiplie deux signaux

- argument = signal constant en entrée 2

`sig~` convertit message nombre en signal constant

- le signal sortie prend la valeur du dernier message reçu.

`output~` objet complexe de sortie stéréo

`dbtorms` convertit de dB (0–100) à ratio d'amplitude (0–1)

- il existe l'inverse `rmstodb`, et les variantes `~`

02.freq.pd Modulation de fréquence

(*moduler* = faire varier un paramètre dans le temps)

Objets utilisés

`mtof` convertit de note MIDI (0-127) à fréquence (Hz)

- il existe l'inverse `ftom`, et les variantes `~`

`key` à chaque touche du clavier pressée,
émet son code ASCII

Où trouver de l'aide

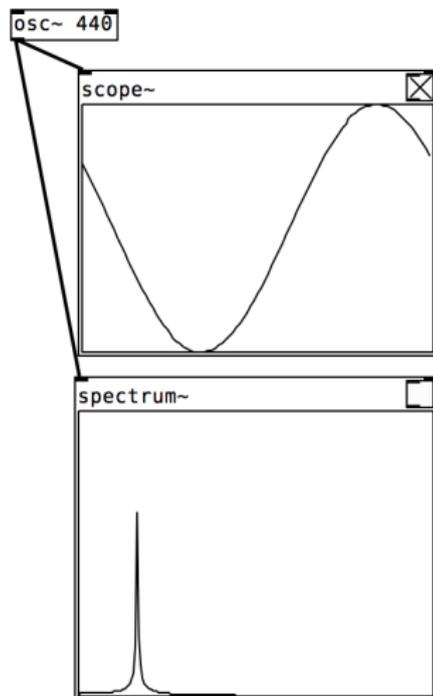
- menu "Aide" (manuel)
- menu contextuel "Aide" de chaque boîte
- "Aide" → Liste des objets

Objets de visualisation fournis

Dans le dossier fourni, vous trouverez des objets complexes :

`scope~` visualise un signal
(rafraîchi toutes les 100ms)

`spectrum~` visualise son spectre



Qu'est-ce qu'une enveloppe ?

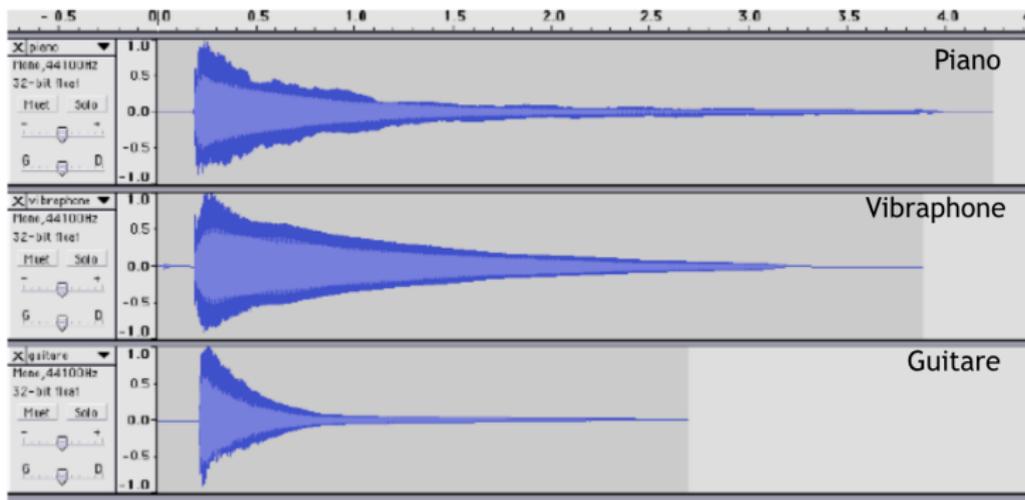
C'est un signal lent qui module un paramètre de la synthèse

- vitesse sub-audio
- forme d'onde libre
- pas nécessairement périodique

Exemples

- tremolo/vibrato
(sine de $\sim 1\text{Hz}$ modulant l'amplitude/la fréquence)
- enveloppe d'amplitude
(zéro au repos, >0 quand on appuie sur une touche)
- enveloppe de timbre
(timbre modulé au cours du temps)

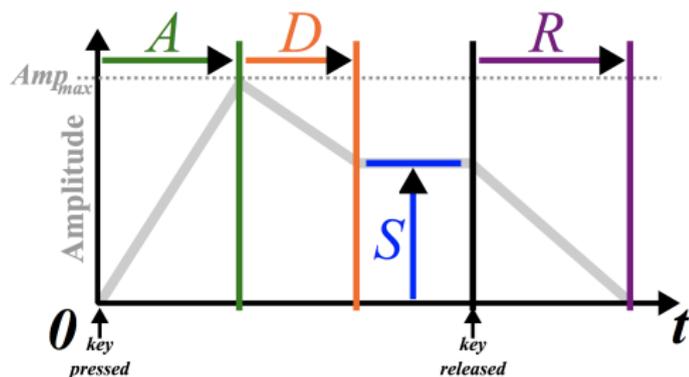
Exemple physique : enveloppe d'amplitude



On distingue plusieurs “phases” dans l’amplitude

- *attaque* : temps de montée quand on fournit de l’énergie
- *déclin* : temps de descente quand on arrête d’en fournir (en général exponentiel)

Exemple historique : l'enveloppe ADSR



- signal par segments : Attack, Decay, Sustain, Release
- 0 au repos
- selon les paramètres A, D, S, R on peut approximer l'enveloppe d'amplitude d'instruments naturels

03.envelope.pd L'objet line~

Génère un segment de droite :

valeur courante $\xrightarrow{\text{temps donné}}$ valeur cible

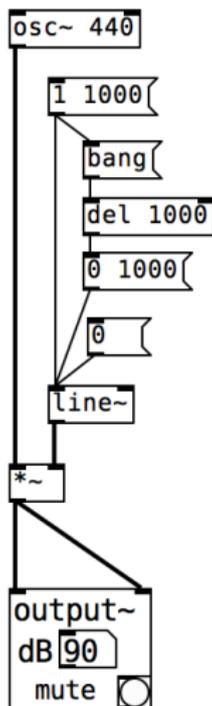
entrée un message de la forme :

- un nombre v :
saute à la valeur v
- deux nombres v et t séparés
par des espaces (liste) :
va de la valeur courante à v
en un temps t

sortie le signal segment de droite

Objet ami : del/delay

- argument : temps t de délai (ms)
- quand reçoit bang, émet bang après
temps t



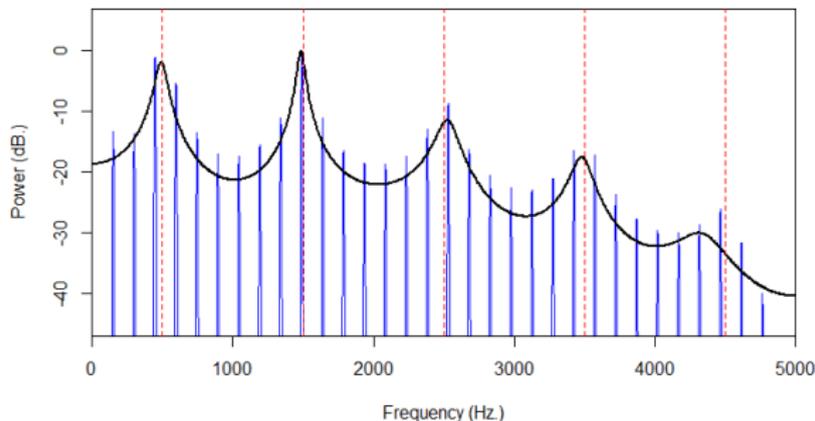
Ne pas confondre

Enveloppe temporelle

Évolution d'un paramètre de synthèse dans le temps
(ce dont on vient de parler)

Enveloppe spectrale

Évolution de l'amplitude des harmoniques d'un son périodique
dans les fréquences = le "contour" que forment les harmoniques
(à un instant donné)



04.objects.pd Quelques objets complémentaires

Éléments d'interface graphique

bang émet un message “bang” quand cliqué



interrupteur émet messages 0 ou 1 quand cliqué
(suivant état courant)



slider émet un nombre 0..127
quand on change sa position



04. objects .pd Quelques objets complémentaires

Éléments d'interface graphique

bang émet un message “bang” quand cliqué



interrupteur émet messages 0 ou 1 quand cliqué
(suivant état courant)



slider émet un nombre 0..127
quand on change sa position



Objets utiles

+, **-**, *****, **/** opérations sur des messages nombre (attention !)

pack concatène des messages pour former des listes

metro émet un “bang” à intervalle régulier

send/receive transmet des messages sans utiliser de fils

random tire un nombre au hasard

print affiche le message qu'il reçoit dans la console

Tables d'ondes

Motivations

- stocker des formes d'ondes arbitraires
- générer des signaux autres que sinusoïdes et segments

Qu'est-ce qu'une table d'onde ?

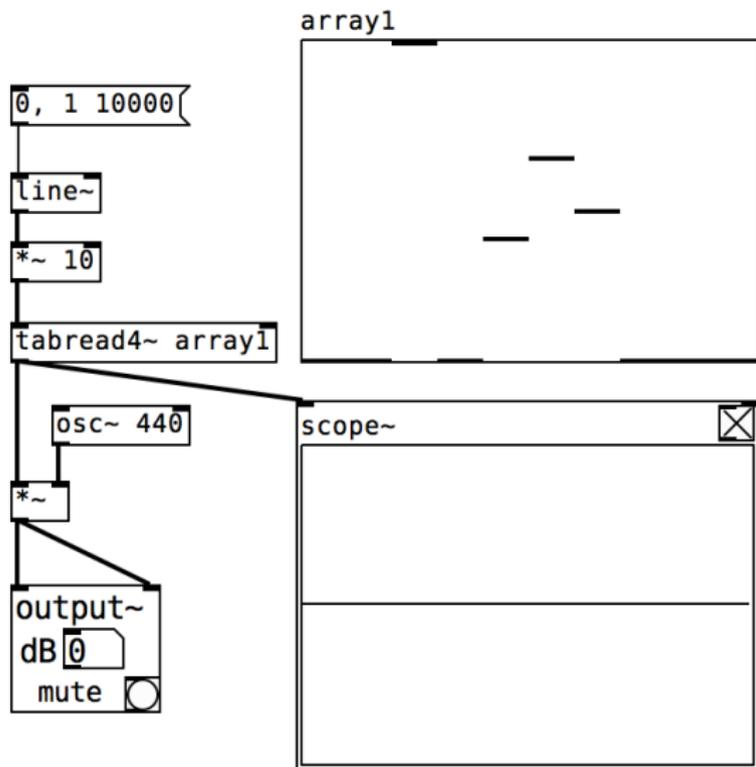
C'est un tableau de valeurs flottantes, qu'on lit pour créer un signal (signal audio ou enveloppe)

Création Ajouter → Tableau → donner nom et taille

Lecture objet `tabread~` et `tabread4~` :

- argument : nom du tableau
- entrée : signal des indices à lire
- sortie : signal des valeurs lues

05.table.pd Génération d'enveloppe arbitraire



Interpolation

- la valeur d'entrée (l'index) n'est pas forcément un entier
- `tabread~` arrondit l'entrée à l'entier le plus proche, et renvoie sa valeur dans le tableau
↳ effet d'escalier / clics
- `tabread4~` fait de l'*interpolation* entre les valeurs du tableau
↳ “arrondit les angles”

Interpolation

Le fait de rendre continu une fonction qui n'est définie qu'à des valeurs discrètes, en “inventant” des valeurs intermédiaires.

Oscillateur à table d'onde

On peut aussi lire une table en boucle à vitesse audio

`phasor~`

permet de scanner une table d'onde sur toute sa longueur :

entrée fréquence f en Hz

sortie signal dent de scie
($0 \rightarrow 1, 0 \rightarrow 1, \dots$)

Oscillateur à table d'onde

On peut aussi lire une table en boucle à vitesse audio

`phasor~`

permet de scanner une table d'onde sur toute sa longueur :

entrée fréquence f en Hz

sortie signal dent de scie
($0 \rightarrow 1, 0 \rightarrow 1, \dots$)

06.wavetable.pd

- quelle différence entre `tabread4~` et `tabread~` ?
- que se passe-t-il si on augmente la valeur 255 ?

