

# Mini-Projet d'AMEA

## Synthèse de Karplus Strong

Benjamin Lévy

### *Indications*

*Ce mini-projet d'AMEA est probablement long. Le but n'est donc pas d'arriver au bout à tout prix mais plutôt de comprendre ce que l'on fait, comprendre les éventuelles erreurs, bugs ou observations. Les questions donnent des éléments qui doivent apparaître dans le compte-rendu (si possible numérique, en pdf) mais ne sont pas exhaustives. Toutes les remarques, commentaires ou explications supplémentaires que vous jugerez utiles sont les bienvenus. Les scripts Matlab sont bien entendu à joindre au compte-rendu. Il n'est pas nécessaire de faire apparaître chaque "état" des scripts si vous montrez dans les réponses et commentaires que les étapes antérieures ont bien été effectuées. Pensez également à commenter vos scripts.*

*Vous trouverez en fin de sujet une petite bibliographie contenant quelques articles et sites qui m'ont permis de construire ce sujet. N'hésitez pas à vous en servir et chercher d'autres documents si nécessaire. N'hésitez pas également à me contacter : [benjamin.levy@ensea.fr](mailto:benjamin.levy@ensea.fr)*

## 1 Introduction

Dans ce sujet il s'agit d'étudier et de programmer dans Matlab l'algorithme de synthèse de Karplus-Strong. Trois méthodes de programmation sont proposées successivement qui permettent d'éclairer ce type de synthèse de manières différentes. La synthèse de Karplus-Strong peut être utilisée pour synthétiser des sons de type corde pincée et de type percussif. Une fois ces deux types de sons obtenus, on essayera de raffiner la seconde méthode pour obtenir un meilleur rendu sonore.

## 2 Fonctionnement

L'algorithme de synthèse de Karplus-Strong, inventé autour de 1983 par Alex Strong et étudié en détail par Kevin Karplus est représenté sur le schéma suivant (FIG 1).

**Q1** : Calculez la fonction de transfert et l'équation de la boucle de retour. Quelle est l'opération effectuée, de quel type de filtre s'agit-il ?

**Q2** : Calculez la fonction de transfert globale et l'équation aux différences correspondante.

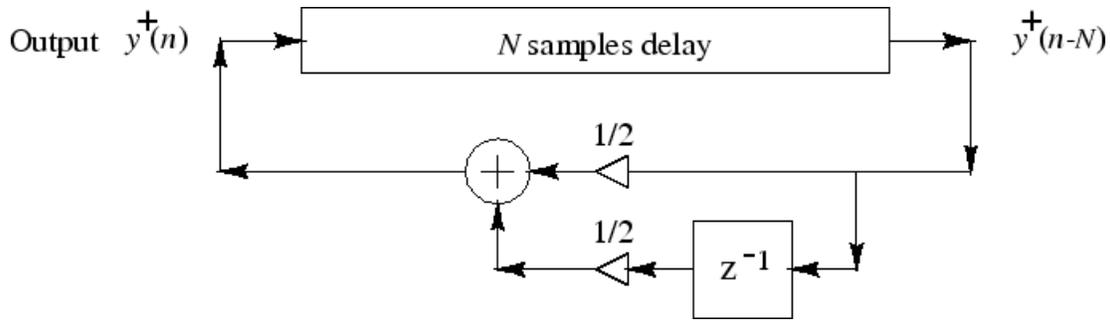


FIGURE 1 – Schéma de l’algorithme

La fréquence fondamentale correspondante est  $f_0 = \frac{F_s}{N+1/2}$  où  $F_s$  est la fréquence d’échantillonnage.

**Q3 :** Pourquoi pouvait-on prévoir l’apparition d’une fréquence fondamentale ? Quels problèmes posent l’expression de  $f_0$  ?

### 3 Programmation par tableau

La première méthode pour programmer cette synthèse est la suivante : partant d’un tableau d’échantillons de bruit blanc de longueur  $N$  (utiliser la fonction `rand`), on effectue la moyenne des échantillons consécutifs (décalage circulaire du tableau puis demi-somme). Le résultat est accolé au premier tableau et l’on itère ce processus jusqu’à ce que l’amplitude du son soit faible.

**Q4 :** De quel type de synthèse cette méthode se rapproche-t-elle ?

**Q5 :** Programmez la synthèse selon l’algorithme décrit. Écoutez le son obtenu et observez son spectre. Vérifiez la valeur de la fréquence fondamentale pour différentes valeurs de  $N$ .

**Q6 :** Observez et commentez l’effet de  $N$  sur le résultat (hormis le changement de hauteur).

### 4 Programmation par échantillons

La seconde méthode proposée reprend exactement le même principe que la première mais en effectuant les calculs échantillon par échantillon (dans une boucle `for`) à partir du tableau de bruit blanc. Chaque échantillon est donc calculé suivant l’équation aux différences trouvée au 2.

**Q7 :** Programmez la synthèse selon cette méthode. Écoutez le son obtenu et observez son spectre.

**Q8 :** Comparez le signal obtenu avec celui obtenu par la méthode précédente. Expliquez les éventuelles différences.

## 5 Programmation par filtrage

La synthèse de Karplus peut également être vue comme un filtrage récursif global de fonction de transfert  $H(z)$  calculée précédemment. L'implémentation dans Matlab est alors beaucoup plus efficace en utilisant la fonction `filter`.

**Q9** : A quel type de synthèse cette méthode se rattache-t-elle ?

**Q10** : Programmez la synthèse avec la fonction `filter` de Matlab. Écoutez le son obtenu et observez son spectre.

**Q11** : Comparez le signal obtenu avec celui obtenu par les autres méthodes. Expliquez les éventuelles différences.

## 6 Son de type percussion

Une méthode très simple pour obtenir un son de type percussif a été trouvée en 1979 par Kevin Karplus. Il s'agit d'introduire un signe aléatoire sur la moyenne selon le schéma suivant (FIG 2) :

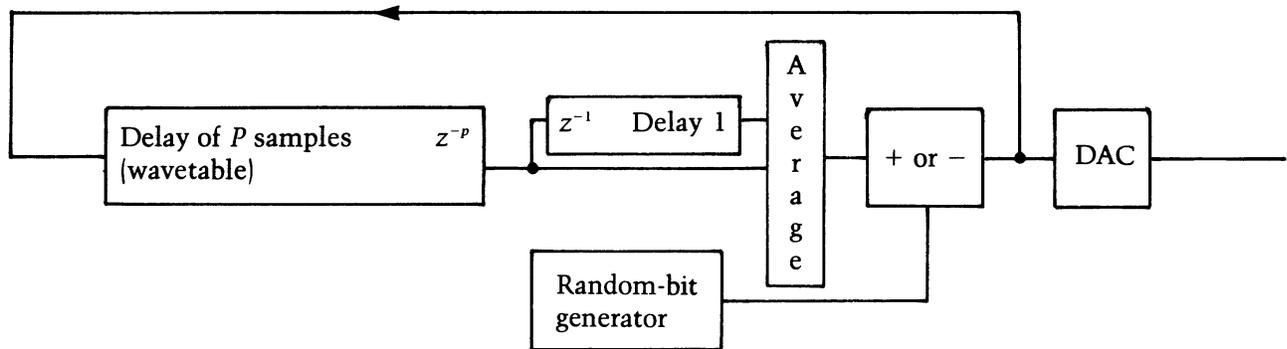


FIGURE 2 – Schéma de l'algorithme pour la synthèse de son de percussion

**Q12** : En utilisant la seconde méthode (boucle `for`), programmez la synthèse d'un son de type percussif. On introduira un tirage aléatoire (probabilité  $p$  et  $1 - p$ ) du signe à chaque tour (utilisez la fonction `rand` et une comparaison).

**Q13** : Observez et décrivez l'effet de la longueur  $N$  du tableau de départ pour différentes valeurs de  $p$ .

## 7 Quelques raffinements

### 7.1 Décroissance

**Q14** : Proposez (et implémentez) une méthode simple pour accélérer la décroissance du son, c'est à dire raccourcir la résonance. Commentez.

Pour allonger le temps de décroissance, on introduit une nouvelle variable  $S \in [1, +\infty]$  (stretch factor). La nouvelle récurrence s'écrit (dans le cas du son de corde pincée) :

$$y[n] = \begin{cases} y[n - N] & \text{probabilité } 1 - 1/S \\ \frac{1}{2}(y[n - N] + y[n - N - 1]) & \text{probabilité } 1/S \end{cases}$$

**Q15** : Programmez ce réglage de la décroissance dans l'algorithme par échantillons (boucle `for`) et testez. Commentez.

**Q16** : Adaptez cet allongement dans le cas du son percussif (toujours dans l'implémentation par échantillons). Comparez avec le cas de la corde pincée.

## 7.2 Source

S'il reste du temps, on pourra essayer d'utiliser comme échantillons de départ, non plus un bruit blanc mais d'autres types de bruit ou les échantillons d'un extrait de son instrumental et écouter les résultats obtenus. Commentez alors.

## Références

- [1] K.Karplus, A. Strong, *Digital Synthesis of Plucked-String and Drum Timbres*, Computer Music Journal, Vol. 7, No. 2, 1983.
- [2] D.A. Jaffe, J.O. Smith, *Extensions of the Karplus-Strong Plucked-String Algorithm*, Computer Music, Journal, Vol. 7, No. 2, 1983.
- [3] J.O. Smith *Virtual Acoustic Musical Instruments : Review of Models and Selected Research*, <http://ccrma.stanford.edu/~jos/Mohonk05/Mohonk05.html>.
- [4] A-M. Burns, *Karplus-Strong Plucked-String Synthesis Algorithm or How To Create String Instruments Out of Noise*, <http://www.music.mcgill.ca/~amburns/physique/index.html>