# Synthèse rapide de signaux aléatoires à temps continu échantillonnés non uniformément

**Livia Nita et Jacques Oksman**

École Supérieure d'Électricité - Service des Mesures
Plateau de Moulon, 91192 Gif-sur-Yvette, France
nita@supelec.fr, oksman@supelec.fr

RÉSUMÉ

Cet article propose une méthode de synthèse de signaux aléatoires stationnaires gaussiens à temps continu échantillonnés non uniformément. Il s'agit d'une méthode numérique fondée sur la résolution analytique exacte des équations différentielles stochastiques linéaires à coefficients constants. Cela permet de surmonter les difficultés numériques liées à des intervalles d'échantillonnage très petits. Ainsi, cette méthode est applicable quel que soit le schéma d'échantillonnage non uniforme. L'algorithme correspondant à cette méthode, ses performances et aussi ses limitations sont décrits et analysés dans l'article.

ABSTRACT

This paper proposes a synthesis method of nonuniformly sampled continuous-time Gaussian stationary signals. We present a numerical method based on the exact analytical resolution of linear stochastic differential equations with constant coefficients. It allows us to address numerical difficulties caused by very small sampling intervals. So, an accurate solution is insured for any nonuniform sampling scheme. The algorithm corresponding to this method, its performances and its limits, too, are described and analysed in the paper.

## 1 Introduction

Des signaux échantillonnés non uniformément se rencontrent dans de nombreuses applications (odométrie Doppler, vélocimétrie laser, méthodes de comptage de particules, transfert de signaux sur des réseaux asynchrones, etc. [1]). La conception de méthodes de traitement efficace est donc utile, en particulier dans le cas de l'analyse spectrale. Afin de pouvoir valider ces traitements, il est nécessaire de disposer de méthodes et d'outils permettant de générer des signaux synthétiques correctement caractérisés.

Dans la littérature, il existe des méthodes permettant de générer des échantillons de signaux aléatoires à temps continu, fondées sur différents schémas d'intégration numérique des équations différentielles stochastiques (approximation par différences finies, décomposition en série de Stratonovitch-Taylor, schémas de type Runge-Kutta, Euler, à pas multiples etc. [2][3][4]).

Nous proposons dans cet article un principe de synthèse de signaux à temps continu échantillonnés non uniformément fondé sur la résolution analytique exacte des équations différentielles stochastiques linéaires à coefficients constants. Cela permet de minimiser les erreurs systématiques numériques tout en surmontant les difficultés numériques liées à des intervalles de temps trop petits (tels ceux rencontrés dans le cas de processus de Poisson).

La section 2 présente les préliminaires et la version « rapide », adaptée à l'échantillonnage non uniforme, de l'algorithme proposé dans [5]. La section 3 analyse les limitations de ce principe et décrit les améliorations que nous y avons apportées dans le cas des intervalles d'échantillonnage très petits ou des pôles multiples. La section 4 présente les performances et les résultats obtenus par l'algorithme de synthèse développé.

## 2 Principe de synthèse

Sous la forme la plus générale, on considère qu'un signal à temps continu échantillonné non uniformément est constitué d'un processus aléatoire à temps continu $x(t)$ et d'un processus ponctuel d'échantillonnage $\pi=\{t_k\}$ [6].

Le processus aléatoire $x(t)$ est classiquement modélisé par la sortie d'un filtre linéaire analogique RII dont l'entrée est un bruit blanc gaussien à temps continu $w(t)$ (plus précisément des incréments d'un mouvement brownien) :

$$A(\mathrm{d})x(t) = B(\mathrm{d})w(t) \quad , \quad \mathrm{E}(w(t)w(t+s)) = \delta(s) \qquad (2.1)$$

$$\begin{aligned} A(\mathrm{d}) &= \mathrm{d}^n + a_{n-1}\mathrm{d}^{n-1} + \ldots + a_1\mathrm{d} + a_0 \quad (a_n = 1) \\ B(\mathrm{d}) &= b_n\mathrm{d}^n + b_{n-1}\mathrm{d}^{n-1} + \ldots + b_1\mathrm{d} + b_0 \end{aligned} \qquad (2.2)$$

$A(\mathrm{d})$ et $B(\mathrm{d})$ sont des polynômes (en l'opérateur différentiel d) irréductibles, les zéros $\lambda_k$ de $A$ sont à partie réelle négative et la condition $\deg(A) > \deg(B)$ (i.e. $b_n = 0$) garantit que $x(t)$ a une variance finie.

Le but de la synthèse est de calculer les échantillons non uniformes $\{x(t_k)\}$ connaissant la densité spectrale de puissance $S_x(\omega)$ de $x(t)$ (donnée par $A$ et $B$) et également le processus d'échantillonnage $\pi=\{t_k\}$.

Le processus $x(t)$ est décrit dans l'espace d'état par :

$$\begin{cases}\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{g}w \\ x = \mathbf{b}^{\mathrm{T}}\mathbf{z}\end{cases} \tag{2.3}$$

$$\mathbf{A}=\begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{bmatrix},\ \mathbf{g}=\begin{bmatrix}0\\0\\\vdots\\0\\1\end{bmatrix},\ \mathbf{b}=\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-2} \\ b_{n-1}\end{bmatrix} \tag{2.4}$$

La valeur initiale $\mathbf{z}(0)$ du vecteur d'état $\mathbf{z}(t)$ se calcule en utilisant sa matrice de covariance :

$$\mathbf{M}_{\mathbf{z}} = \mathrm{E}(\mathbf{z}(0)\mathbf{z}^{*}(0)) = [m_{ij}] \tag{2.5}$$

Il existe deux possibilités pour calculer les $m_{ij}$ .

1) En les exprimant en fonction des $a_k$ [5] :

$$m_{ij} = \begin{cases} 0, & i+j \ \text{impair} \\ (-1)^{(j-i)/2} m_{(j+i)/2}, & i+j \ \text{pair} \end{cases} \tag{2.6}$$

où $m_0, m_1, \dots, m_{n-1}$ se calculent en résolvant les $n$ équations linéaires suivantes :

$$\sum_{q=\lceil k/2\rceil}^{\lfloor (n+k)/2\rfloor} (-1)^q a_{2q-k} m_q = \begin{cases} 0, & k = 0,\dots,n-2 \\ (-1)^{n-1}/2, & k = n-1 \end{cases} \tag{2.7}$$

2) En les exprimant en fonction de $\lambda_k$ [7] :

$$m_{ij} = \frac{1}{2}\sum_{k=1}^{n} \frac{(-1)^{j+1}\lambda_k^{i+j}}{\mathrm{Re}(\lambda_k)\prod_{l\neq k}(\lambda_l - \lambda_k)(\bar{\lambda}_l + \lambda_k)} \tag{2.8}$$

Cette deuxième méthode est plus rapide mais elle exige que les valeurs propres de $\mathbf{A}$ soient toutes distinctes.

$\mathbf{M}_{\mathbf{z}}$ étant définie positive, elle admet une décomposition de Cholesky : $\mathbf{M}_{\mathbf{z}} = \mathbf{T}_{\mathbf{z}}\mathbf{T}_{\mathbf{z}}^{*}$ ($\mathbf{T}_{\mathbf{z}}$, matrice triangulaire inférieure réelle). Cela permet le calcul du vecteur d'état initial :

$\mathbf{z}(0) = \mathbf{T}_{\mathbf{z}}\mathbf{w}^{(0)}$, où $\mathbf{w}^{(0)}$ est un vecteur gaussien obtenu par tirage aléatoire (de même pour tout vecteur $\mathbf{w}^{(k)}$).

En intégrant l'équation d'état du système (2.3), on obtient la relation de récurrence entre les vecteurs d'état successifs :

$$\begin{aligned}&\mathbf{z}(t_{k+1}) = \mathrm{e}^{\mathbf{A}\Delta t}\mathbf{z}(t_k) + \mathbf{r}(\Delta t) \\ &= \mathrm{e}^{\mathbf{A}\Delta t}\mathbf{z}(t_k) + \int_0^{\Delta t} \mathrm{e}^{\mathbf{A}(\Delta t - s)}\mathbf{g}w(t_k + s)\mathrm{d}s \quad , \quad \Delta t = t_{k+1} - t_k\end{aligned} \tag{2.9}$$

où $\mathbf{r}(\Delta t)$ est la contribution du bruit blanc (de l'entrée du système) entre $t_k$ et $t_{k+1}$. Le vecteur $\mathbf{r}(\Delta t)$ n'étant pas corrélé avec $\mathbf{z}(t_k)$ [5], on peut le calculer à l'aide de sa matrice de covariance :

$$\begin{aligned}&\mathbf{M}_{\mathbf{r}} = \mathrm{E}(\mathbf{r}(\Delta t)\mathbf{r}^{*}(\Delta t)) \\ &\mathrm{E}\Big(\int_0^{\Delta t}\int_0^{\Delta t} \mathrm{e}^{(\Delta t - s_1)\mathbf{A}}\mathbf{g}w(t_k + s_1)w^{*}(t_k + s_2)\mathbf{g}^{*}\mathrm{e}^{(\Delta t - s_2)\mathbf{A}^{*}}\mathrm{d}s_1\mathrm{d}s_2\Big) \\ &= \int_0^{\Delta t} \mathrm{e}^{s\mathbf{A}}\mathbf{C}\mathrm{e}^{s\mathbf{A}^{*}}\mathrm{d}s \quad , \quad \mathbf{C} = \mathbf{g}\mathbf{g}^{*}\end{aligned} \tag{2.10}$$

Une ***première*** version de l'algorithme utilise l'intégration par parties de cette dernière équation :

$$\mathrm{e}^{\mathbf{A}\Delta t}\mathbf{C}\mathrm{e}^{\mathbf{A}^{*}\Delta t} - \mathbf{C} = \mathbf{A}\mathbf{M}_{\mathbf{r}} + \mathbf{M}_{\mathbf{r}}\mathbf{A}^{*} \tag{2.11}$$

Cela demande la résolution d'un système linéaire de taille $n(n+1)/2$ à chaque instant $t_k$ (en utilisant le fait que $\mathbf{M}_{\mathbf{r}}$ est symétrique) :

$$\begin{aligned}&\sum_{k=1}^{j}\mathbf{A}(i,k)\mathbf{M}_{\mathbf{r}}(j,k) + \sum_{k=j+1}^{n}\mathbf{A}(i,k)\mathbf{M}_{\mathbf{r}}(k,j) \\ &+\sum_{k=1}^{i}\mathbf{A}(j,k)\mathbf{M}_{\mathbf{r}}(i,k) + \sum_{k=i+1}^{n}\mathbf{A}(j,k)\mathbf{M}_{\mathbf{r}}(k,i),\ 1\le j\le i\le n \\ &= \theta_{in}\theta_{jn} - \delta(i,n)\delta(j,n) \quad , \quad \mathrm{e}^{\mathbf{A}\Delta t} = [\theta_{ij}]\end{aligned} \tag{2.12}$$

*N.B.* $\mathbf{A}$ étant connue, on résoudra les systèmes à partir d'une seule inversion de la matrice.

La ***deuxième*** version, qui améliore encore plus la rapidité de l'algorithme, utilise la décomposition diagonale de la matrice $\mathbf{A}$ [8], applicable uniquement dans le cas où les valeurs propres de $\mathbf{A}$ sont toutes distinctes :

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1} \tag{2.13}$$

$$\text{où}\quad \mathbf{\Lambda} = \begin{bmatrix}\lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n\end{bmatrix},\ \mathbf{U} = \begin{bmatrix} 1 & \dots & 1 \\ \lambda_1 & \dots & \lambda_n \\ \lambda_1^2 & \dots & \lambda_n^2 \\ \vdots & & \vdots \\ \lambda_1^{n-1} & \dots & \lambda_n^{n-1}\end{bmatrix} \tag{2.14}$$

On obtient alors une expression analytique pour $\mathbf{M}_{\mathbf{r}}$ :

$$\begin{aligned}&\mathbf{M}_{\mathbf{r}} = \mathbf{U}\mathbf{\Phi}\mathbf{E}\mathbf{\Phi}^{*}\mathbf{U}^{*}\ ,\ \mathbf{\Phi} = \mathrm{diag}[\mathbf{U}^{-1}(i,n)]\ ,\ i=1,\dots,n \\ &\mathbf{E} = [e_{ij}] = \left[\frac{\mathrm{e}^{\Delta t(\lambda_i + \bar{\lambda}_j)} - 1}{\lambda_i + \bar{\lambda}_j}\right],\quad i,j = 1,\dots,n\end{aligned} \tag{2.15}$$

La décomposition de Cholesky appliquée à $\mathbf{M}_{\mathbf{r}}$, $\mathbf{M}_{\mathbf{r}} = \mathbf{T}_{\mathbf{r}}\mathbf{T}_{\mathbf{r}}^{*}$, permet alors d'obtenir $\mathbf{r}(\Delta t)$ :

$$\mathbf{r}(\Delta t) = \mathbf{T}_{\mathbf{r}}(\Delta t)\mathbf{w}^{(k+1)} \tag{2.16}$$

et, par conséquent, $\mathbf{z}(t_{k+1})$.

# 3 Limitations et améliorations

Les deux versions présentent certaines erreurs numériques pour des valeurs $\Delta t$ très petites, telles celles susceptibles d'apparaître dans un processus de Poisson. L'expression analytique de $\mathbf{M}_{\mathbf{r}}$ nous permet de comprendre cela : quand $\Delta t$ tend vers zéro, le calcul des $e_{ij}$ implique la différence entre deux quantités très proches. Les erreurs numériques se manifestent pour des intervalles d'échantillonnage $\Delta t < \Delta t_{\text{critique}} = \sqrt{\varepsilon}/\max|\lambda_i|$, où $\varepsilon$ est la précision relative utilisée dans les calculs (typiquement, $\varepsilon = 10^{-15}$ pour les réels double précision, sur 8 octets).

Ces erreurs rendent impossible la décomposition de $\mathbf{M}_{\mathbf{r}}$, car, dans ce cas, la plupart de ses valeurs singulières tendent aussi vers zéro et, à cause des erreurs d'arrondi, peuvent même devenir négatives (quand $\Delta t$ tend vers zéro les

valeurs singulières de **E** sont [$n\Delta t$ 0 … 0] et $\mathbf{M_r}$ les hérite).

Une amélioration supplémentaire consiste à calculer **E** par un développement en série de Taylor :

$$e_{ij|\Delta t\to 0} = \Delta t \sum_{k\geq 0} \frac{[\Delta t(\lambda_i + \overline{\lambda}_j)]^k}{(k+1)!} \qquad (3.1)$$

Contrairement à ce que l'on pourrait croire, pour $\Delta t < \Delta t_{\text{critique}}$, ce calcul s'avère expérimentalement plus précis que le calcul employant l'exponentielle. On montre que, dans ces conditions, un développement d'ordre 2 est suffisant.

Une troisième amélioration consiste à effectuer une rotation orthogonale du vecteur d'état **z**(*t*) afin d'obtenir une équation d'état caractérisée par une matrice de transition diagonale $\mathbf{\Lambda}$ [7]. Ainsi, les équations (2.3), (2.9) et (2.10) deviennent :

$$\dot{\boldsymbol{\xi}} = \mathbf{\Lambda}\boldsymbol{\xi} + \mathbf{U}^{-1}\mathbf{g}w \quad , \quad \mathbf{z} = \mathbf{U}\boldsymbol{\xi} \quad , \quad x = \mathbf{b}^{\mathrm{T}}\mathbf{U}\boldsymbol{\xi} \qquad (3.2)$$

$$\begin{aligned}\boldsymbol{\xi}(t_{k+1}) &= \mathrm{e}^{\Lambda\Delta t}\boldsymbol{\xi}(t_k) + \mathbf{q}(\Delta t) \\ &= \mathrm{e}^{\Lambda\Delta t}\boldsymbol{\xi}(t_k) + \int_0^{\Delta t} \mathrm{e}^{\Lambda(\Delta t - s)}\mathbf{U}^{-1}\mathbf{g}w(t_k + s)\mathrm{d}s\end{aligned} \qquad (3.3)$$

$$\begin{aligned}\mathbf{M_q} &= \mathrm{E}(\mathbf{q}(\Delta t)\mathbf{q}^*(\Delta t)) \\ &= \int_0^{\Delta t} \mathrm{e}^{s\mathbf{\Lambda}}\mathbf{U}^{-1}\mathbf{g}\mathbf{g}^*(\mathbf{U}^{-1})^*\mathrm{e}^{s\mathbf{\Lambda}^*}\mathrm{d}s = \mathbf{\Phi}\mathbf{E}\mathbf{\Phi}^*\end{aligned} \qquad (3.4)$$

Ceci rend l'algorithme plus rapide mais, en revanche, il utilise des matrices à valeurs complexes.

Ainsi, la décomposition $\mathbf{M_q} = \mathbf{T_q}\mathbf{T_q}^*$, où $\mathbf{T_q} \in \mathbf{C}^{n\times n}$, n'est pas unique et elle se fait sous la contrainte que $\mathbf{UT_q}$ soit réel (car $x(t_k)$ doit être réel). Soit $\mathbf{T_E}$ la matrice complexe obtenue analytiquement par la décomposition de Cholesky, $\mathbf{E} = \mathbf{T_E}\mathbf{T_E}^*$ et **Q**, la matrice unitaire qui rend $\mathbf{U\Phi T_E}$ réelle (i.e. $\mathbf{U\Phi T_E Q}$ est réelle). Alors : $\mathbf{T_q} = \mathbf{\Phi T_E Q}$.

Pour le calcul de $\mathbf{T_E}$ (quand $\Delta t < \Delta t_{\text{critique}}$), nous proposons une méthode de Cholesky modifiée qui s'arrête dès que les valeurs diagonales de $\mathbf{T_E}$ deviennent négatives ou trop proches de 0. Cette amélioration permet de minimiser au maximum les erreurs systématiques numériques.

Cependant, toutes ces procédures s'appliquent uniquement dans le cas où les valeurs propres $\lambda_i$ de la matrice **A** sont toutes distinctes (**U** doit être inversible).

Une ***dernière*** version que nous proposons concerne le cas où **A** a des valeurs propres multiples. Dans ce cas, $\mathbf{A} = \mathbf{UJU}^{-1}$ où **J** est une matrice de type Jordan et **U** est composée cette fois-ci de vecteurs propres et de vecteurs principaux [8]. Ainsi :

$$\mathbf{J} = \begin{bmatrix} \mathbf{J_1} & & 0 \\ & \mathbf{J_2} & \\ 0 & & \mathbf{J_k} \end{bmatrix}, \ \mathbf{J_i} = \begin{bmatrix} \lambda_i & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda_i \end{bmatrix}, \ \mathbf{J_i} \in \mathbf{C}^{\sigma_i \times \sigma_i} \qquad (3.5)$$

où $\sigma_i$ représente l'ordre de multiplicité algébrique de $\lambda_i$.

Cette forme particulière de la matrice **J** permet d'écrire :

$$\mathrm{e}^{s\mathbf{J}} = \begin{bmatrix} \mathbf{B_1} & & 0 \\ & \mathbf{B_2} & \\ 0 & & \mathbf{B_k} \end{bmatrix}, \ \mathbf{B_i} = \mathrm{e}^{s\lambda_i} \begin{bmatrix} 1 & \frac{s}{1!} & \frac{s^2}{2!} & \cdots & \frac{s^{\sigma_i - 1}}{\sigma_i!} \\ & 1 & \frac{s}{1!} & \ddots & \vdots \\ & & \ddots & \ddots & \frac{s^2}{2!} \\ & & & \ddots & \frac{s}{1!} \\ 0 & & & & 1 \end{bmatrix} \qquad (3.6)$$

Ainsi, on obtient des équations similaires aux équations (3.2) - (3.4) :

$$\begin{aligned}\dot{\boldsymbol{\xi}} &= \mathbf{J}\boldsymbol{\xi} + \mathbf{U}^{-1}\mathbf{g}w, \ \boldsymbol{\xi}(t_{k+1}) = \mathrm{e}^{\mathbf{J}\Delta t}\boldsymbol{\xi}(t_k) + \mathbf{q}(\Delta t) \\ &= \mathrm{e}^{\mathbf{J}\Delta t}\boldsymbol{\xi}(t_k) + \int_0^{\Delta t} \mathrm{e}^{\mathbf{J}(\Delta t - s)}\mathbf{U}^{-1}\mathbf{g}w(t_k + s)\mathrm{d}s\end{aligned} \qquad (3.7)$$

$$\begin{aligned}\mathbf{M_q} &= \mathrm{E}(\mathbf{q}(\Delta t)\mathbf{q}^*(\Delta t)) = \int_0^{\Delta t} \mathrm{e}^{s\mathbf{J}} \underbrace{\mathbf{U}^{-1}\mathbf{g}\mathbf{g}^*(\mathbf{U}^{-1})^*}_{\mathbf{V}(s)} \mathrm{e}^{s\mathbf{J}^*}\mathrm{d}s \\ &= \int_0^{\Delta t} \mathbf{V}(s)\mathbf{V}^*(s)\mathrm{d}s = \mathbf{\Phi}\mathbf{E}_{\mathbf{mod}}\mathbf{\Phi}^*\end{aligned} \qquad (3.8)$$

où le vecteur **V**(*s*) a la forme suivante :

$$\mathbf{V}(s) = \begin{bmatrix} \mathbf{B_1}\mathbf{U}^{-1}(1{:}\sigma_1, n) \\ \mathbf{B_2}\mathbf{U}^{-1}(\sigma_1 + 1{:}\sigma_1 + \sigma_2, n) \\ \vdots \\ \mathbf{B_k}\mathbf{U}^{-1}(\sigma_1 + \sigma_2 + \ldots + \sigma_{k-1} + 1{:}n, n) \end{bmatrix} \qquad (3.9)$$

En adoptant la notation $\mathbf{U}^{-1} = [ui_{ij}]$, l'expression des éléments $v_i$ de **V**(*s*) est donnée par :

$$v_i = \mathrm{e}^{s\lambda_{idx\_i}} ui_{in} \underbrace{\sum_{k=0}^{\sigma_1 + \ldots \sigma_{idx\_i} - i} \frac{ui_{i+k,n}}{ui_{in}} \cdot \frac{s^k}{k!}}_{P_i(s)} = \mathrm{e}^{s\lambda_{idx\_i}} ui_{in} P_i(s) \qquad (3.10)$$

où *idx_i* désigne l'indice du bloc $\mathbf{B_{idx\_i}}$ incluant la ligne *i*. Donc :

$$\begin{aligned}\mathbf{M_q} &= \int_0^{\Delta t} [v_i(s)v_j^*(s)]_{ij}\mathrm{d}s \quad , \quad i, j = 1, \ldots, n \\ &= \int_0^{\Delta t} [\mathrm{e}^{(\lambda_{idx\_i} + \overline{\lambda}_{idx\_j})s} ui_{in}\overline{ui}_{jn} P_i(s)\overline{P}_j(s)]_{ij}\mathrm{d}s \\ &= \left[ ui_{in} \frac{\mathrm{e}^{(\lambda_{idx\_i} + \overline{\lambda}_{idx\_j})\Delta t} R_{ij}(\Delta t)}{\lambda_{idx\_i} + \overline{\lambda}_{idx\_j}} \overline{ui}_{jn} \right]_{ij} \\ &= \mathbf{\Phi} \underbrace{\left[ \frac{\mathrm{e}^{(\lambda_{idx\_i} + \overline{\lambda}_{idx\_j})\Delta t} R_{ij}(\Delta t)}{\lambda_{idx\_i} + \overline{\lambda}_{idx\_j}} \right]_{ij}}_{\mathbf{E}_{\mathbf{mod}}} \mathbf{\Phi}^* = \mathbf{\Phi}\mathbf{E}_{\mathbf{mod}}\mathbf{\Phi}^*\end{aligned} \qquad (3.11)$$

On adopte les notations suivantes :

$$\lambda_{ij} = \lambda_i + \overline{\lambda}_j \qquad (3.12)$$

$$R_{ij}(s) = r_0 + r_1 s + r_2 s^2 + \ldots + r_m s^m \qquad (3.13)$$

$$Q_{ij}(s) = P_i(s)\overline{P}_j(s)$$
$$= \left( \sum_{k=0}^{\sigma_1+\ldots+\sigma_{idx\_i}-i} \frac{ui_{i+k,n}}{ui_{in}} \cdot \frac{s^k}{k!} \right) \left( \sum_{k=0}^{\sigma_1+\ldots+\sigma_{idx\_j}-j} \frac{\overline{ui}_{j+k,n}}{\overline{ui}_{jn}} \cdot \frac{s^k}{k!} \right) \quad (3.14)$$
$$= q_0 + q_1 s + q_2 s^2 + \ldots + q_m s^m$$

Ainsi, les coefficients de $R_{ij}(s)$ sont calculés en fonction des coefficients de $Q_{ij}(s)$ selon :

$$r_0 = q_0 + \frac{1!}{(-\lambda_{ij})^1} q_1 + \frac{2!}{(-\lambda_{ij})^2} q_2 + \ldots + \frac{m!}{(-\lambda_{ij})^m} q_m \quad (3.15)$$
$$r_{k-1} = q_{k-1} - k r_k / \lambda_{ij}, \quad k = 1, \ldots, m \quad \text{et} \quad r_m = q_m$$

La suite de cette dernière version de l'algorithme a la même forme que les deux versions précédentes.

On note que la rapidité des méthodes proposées est aussi améliorée car le calcul des matrices **E**, **Φ**, $\mathbf{U}^{-1}$, $\mathbf{b}^T\mathbf{U}$ utilisées à chaque instant $t_k$, est effectué une seule fois au début de l'algorithme.

# 4 Résultats et conclusions

L'algorithme présenté (incluant des optimisations spécifiques pour chaque cas de figure) a été testé pour la synthèse de signaux à enveloppe spectrale quelconque. Nous présentons ci-après les résultats obtenus pour deux filtres dont les fonctions de transfert $H_1(p)$ et $H_2(p)$ sont :

$$H(p) = \frac{p^3 + 3p^2 + p}{(p^2 + 0.04p + 0.01)(p^2 + 4p + 100)} \quad (4.1)$$

$$H(p) = \frac{2p^2 + 6p + 2}{(p+1)(p+2)(p+3)(p+4)(p+5)} \quad (4.2)$$

Les densités spectrales de puissance $S_x(\omega) = |H(j\omega)|^2$ reconstituées à partir de $128 \cdot 10^3$ échantillons, sont représentées dans les figures 1 et 2 respectivement. Les tableaux 1 et 2 présentent le nombre d'opérations en virgule flottante effectuées $N_{op\_filtre}$ et respectivement le temps de calcul $T_{calcul\_filtre}$ spécifiques à l'algorithme optimisé en fonction du nombre d'échantillons $N_{ech}$ synthétisés. La simulation Matlab a été exécutée sur Pentium® 200 MMX.

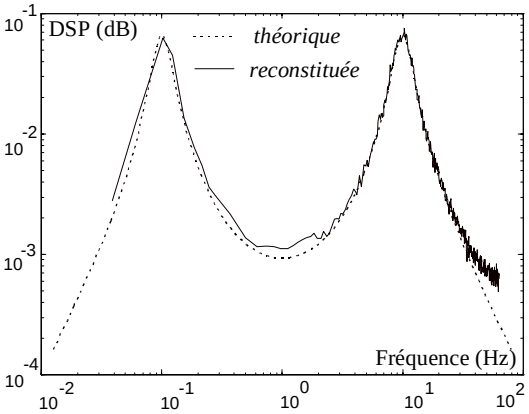


*Figure 1. DSP du signal synthétisé*

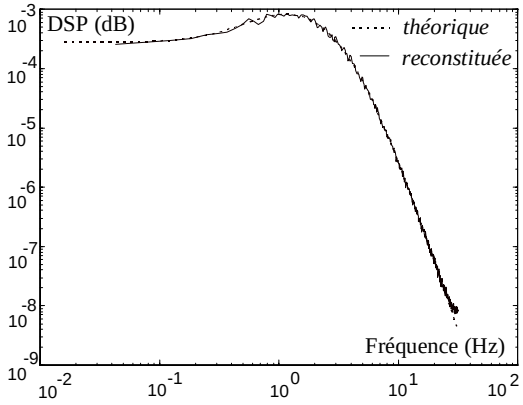


*Figure 2. DSP du signal synthétisé*

*Tableau 1.*

| $N_{ech}$ | $10^3$ | $2 \cdot 10^3$ | $4 \cdot 10^3$ | $8 \cdot 10^3$ |
|---|---|---|---|---|
| $N_{op\_filtre1}$ | $2.2 \cdot 10^6$ | $4.4 \cdot 10^6$ | $8.7 \cdot 10^6$ | $17.4 \cdot 10^6$ |
| $N_{op\_filtre2}$ | $10^6$ | $2 \cdot 10^6$ | $4.1 \cdot 10^6$ | $8.1 \cdot 10^6$ |

*Tableau 2.*

| $N_{ech}$ | $10^3$ | $2 \cdot 10^3$ | $4 \cdot 10^3$ | $8 \cdot 10^3$ |
|---|---|---|---|---|
| $T_{calcul\_filtre1}$ (s) | 0.82 | 1.63 | 3.28 | 6.57 |
| $T_{calcul\_filtre2}$ (s) | 0.68 | 1.34 | 2.69 | 5.37 |

La méthode de synthèse précédente permet de synthétiser à la fois précisément et rapidement des signaux échantillonnés de façon quelconque. Ces signaux serviront à alimenter différentes méthodes de traitement, en particulier d'analyse spectrale.

# 5 Références

[1] Oksman J., « *Modèle général des signaux à échantillonnage non périodique* », Rapport interne, *Supélec*, 1995.

[2] Kloeden P.E. and Platen E., « *Numerical solution of stochastic differential equations* », Springer-Verlag, Berlin, 1992.

[3] Sobczyk K., « *Stochastic differential equations* », Kluwer Academic Publishers, Dordrecht, 1991.

[4] Aldebert P., « *Simulation en régime transitoire des bruits dans les circuits électroniques* », Thèse Université Paris XI Orsay, 1994.

[5] Franklin J.N., « *Numerical simulation of stationary and non-stationary gaussian random processes* », SIAM Review, 7 (1), pp. 68-80, January 1965.

[6] Leneman O.A.Z., « *Random sampling of random processes : Impulse processes* », IEEE-IC 9, pp. 347-363, 1966.

[7] Jones R.H., « *Fitting a continuous time autoregression to discrete data* », in « *Applied Time Series Analysis* », Vol. II, pp. 651-682, Academic Press, New York, 1981.

[8] Stoer J. and Bulirsch R., « *Introduction to numerical analysis* », Springer-Verlag, New York, 1980.