International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:4, No:2, 2010

Simulation of sample paths of nonGaussian stationary random fields

Fabrice Poirion, and Bénédicte Puig

Abstract—Mathematical justifications are given for a simulation technique of multivariate nonGaussian random processes and fields based on Rosenblatt's transformation of Gaussian processes. Different types of convergences are given for the approaching sequence. Moreover an original numerical method is proposed in order to solve the functional equation yielding the underlying Gaussian process autocorrelation function.

Keywords—simulation, nonGaussian, random field, multivariate, stochastic process.

I. Introduction

THE use of nonGaussian models in order to mimic the natural world uncertainty has gained some popularity among the civil engineering community (but not only) for several reasons: firstly, there exist today large amounts of experimental measures which show that many physical phenomena are not Gaussian. It is indeed the case for the sea state, where the statistical distribution of the largest wave height cannot be deduced from a Gaussian assumption [1], [2], [3], for seisms [4], [5], winds in the atmospheric boundary layer [6], [7], and also in astrophysics [8]. Secondly, the formidable progress of computer technology allows the use of Monte Carlo simulation (MCS) methods for real life industrial problems. Lastly, manufacturers are looking for cost reducing technologies and, taking into account more realistic models for environment, is a step towards reducing security margins.

Various methods have been proposed for generating simulated paths of non-Gaussian real valued processes [9], [10], [11], [12], [13], [14], [15], [16]. As it is not realistic to construct a numerical model for a non-Gaussian process $X(t)$ based on its entire family of joint distributions $\{\mathcal{L}(X(t_1),\ldots,X(t_n)), n \geq 1, t_i \in \mathbb{R}\}$, all the proposed methods focus on the following reduced objective: construct a model which has the same one-dimension marginal probability distribution and the same correlation function. The numerical methods which are proposed in the litterature are all related , except [2], to real valued processes. This is because they explicitely use the inverse of the prescribed marginal cumulative distribution function, but this approach cannot be used in the context of multivariate processes since such an inverse does not exist.

The goal of this paper is to propose an extension of the general method given by the authors in [17] for generating simulated paths of non-Gaussian homogeneous random scalar fields to the vector case. The extension is based on Rosenblatt's transformation, in order to generalize the use of the inverse cumulative function, and on its projection on the multivariate Hermite polynomial basis. Different types of convergence will be given for the approximating sequence. As in the scalar case, the autocorrelation function of the underlying Gaussian process will be approached by an optimization problem. This approach is well adapted to the case where the first order marginal probability distribution is described through copulas. Several numerical examples will be given in order to illustrate this approach and its generality. Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. All the random variables or stochastic processes appearing in this paper will be constructed on this abstract probability space.

F. Poirion is with the French Aerospace Lab, ONERA e-mail: poirion@onera.fr.
B. Puig is with the Pau University, France, e-mail: benedicte.puig@univ-pau.fr .

II. Data and assumptions

Let $X(t) = (X_1(t), ..., X_n(t))$; $t \in \mathbb{R}^d$ a n-dimensional random field, which is weakly stationary:$\forall t, s \in \mathbb{R}^d$

$$E(X(t)) = E(X(s)) \ ; \ E(X(t+s)X(t)^T) = R_X(s) \quad (1)$$

The spectral measure M_X of the random field is related to the autocorrelation function R_X through:

$$R_X(s) = \int_{\mathbb{R}^d} e^{i<s,\lambda>} dM_X(\lambda) = \int_{\mathbb{R}^d} e^{i<s,\lambda>} S_X(\lambda) d\lambda \quad (2)$$

$\forall \lambda, s \in \mathbb{R}^d$ the last integral being meaningful when the spectral measure has a density with respect to the Lebesgue measure: $dM_X(\lambda) = S_X(\lambda)d\lambda$.

As it is explained in the introduction, the goal is to construct a simulation method of a vector valued nonGaussian random field $X(t)$ which is described only by its autocorrelation function R_X, or equivalently by its spectral measure M_X, and by its first order marginal distribution $f(x_1, x_2, ..., x_n)$ which is independant of parameter t since the random field is assumed to be stationary. Therefore let Y be an n-dimensional random variable with probability distribution $P_Y = f(x_1, x_2, ..., x_n)dx_1...dx_n$ and $R_X : \mathbb{R}^d \longrightarrow \mathrm{Mat}_{\mathbb{R}}(n,n)$ a $L^2(\mathbb{R}^d)$ function such that $R_X(0) = I$, the unit matrix, and such that the trace $\mathrm{tr}(R_X)$ is a nonnegative definite function.

III. Memoryless transformation construction for multivariate processes

Starting from the fact that generation of Gaussian sample paths is a classical problem : there exist various methods in the literature which can be used [18], the proposed method is

a 2 step procedure. The first step is to find a functional representation of the nonGaussian process in terms of stationary Gaussian process. This representation is constructed from the marginal distribution data, and the second step is to identify the autocorrelation function of the underlying Gaussian processes using the nonGaussian autocorrelation function $R_X(t)$. In order to find the functional representation, a natural way to proceed is to use what is done for the simulation of multivariate random variables. The simplest but not the most frequent case is when the multivariate distribution is directly expressed as a function of independant real-valued random variables for instance the Dirichlet distribution which can be expressed in terms of independant gamma distributions [19]. Apart this very particular situation, there exist two main methods in the literature for simulating random vectors: the rejection method or the conditional distribution method. Clearly the first method is based on a purely algorithmic construction and cannot be represented as a function of given scalar random variables. The second method however can be described functionnaly using the Rosenblatt's transformation [20]. It is also the case when copulas are introduced for describing the dependency of the random vector components.

A. Rosenblatt's transformation

Let $Y = (Y_1, Y_2, \ldots, Y_n)$ a random vector with distribution $f(x_1, x_2, ..., x_n)$. The random variable can be simulated using the following algorithm [21]:

$$\begin{cases} F_1(Y_1) = U_1 \\ F_2(Y_2|Y_1) = U_2 \\ \vdots \\ F_n(Y_n|Y_1, \ldots, Y_{n-1}) = U_n \end{cases}$$

where $U_1, ..., U_n$ are independant random variables with uniform distribution on $[0,1]$, F_1 is the cumulative function of random variable Y_1 and $F_j(Y_j|Y_1, \ldots, Y_{j-1})$, $j = 1, n$, is the conditionnal cumulative function of random variable Y_j knowing $Y_1, \ldots, Y_{j-1}$. Those last relations can be rewritten as :

$$\begin{cases} Y_1 = F_1^{-1}(U_1) = \mathcal{F}_1(U_1) \\ Y_2 = F_2^{-1}(Y_2|Y_1 = F_1^{-1}(U_1))(U_2) = \mathcal{F}_2(U_1, U_2) \\ \vdots \\ Y_n = F_n^{-1}(Y_n|Y_1, \ldots, Y_{n-1})(U_n) = \mathcal{F}_n(U_1, U_2, ..., U_n) \end{cases} \quad (3)$$

hence we have constructed a functional relation between the random vector $U = (U_1, U_2, ..., U_n)$ and the random vector Y:

$$\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^n : U \mapsto Y = \mathcal{F}(U) \; ; \; \mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_n)$$

Now, each random variable U_j can be written in terms of independant Gaussian random variable G_j with distribution $\mathcal{N}(0,1)$: $U_j = F_G(G_j)$. Which yields a functional representation of process X in terms of independant Gaussian processes:

$$X(t) = f(G(t)) \; ; \; f = (f_1, ..., f_n) \; ; \; G(t) = (G_1(t), ..., G_n(t))$$

$$f_1(x) = \mathcal{F}_1 \circ F_G(x_1) \; ; \; ... \; ;$$

$$f_n(x) = \mathcal{F}_n(F_G(x_1), F_G(x_2), ..., F_G(x_n))$$

The first order marginal distribution of the random process $X(t)$ is P_Y by construction. In order to solve the problem, the autocorrelation function of each Gaussian process G_j has to be defined.

B. Copulas

As it is written in the introduction, simulation methods are used to reproduce real life phenomena and as such, have to be constructed from experimental in situ measurements. Although it is relatively simple to obtain some statistic information through measures for each component of a random vector valued process, often, dependancy information between each component is lacking. What is done in general (especially in financial mathematics [22] or geophysics [23], [24], [25]) is to introduce such a dependancy through the use of copula [26]. One advantage of copulas is that a wide range of them are described through one or two parameter models. The statistical problem of fitting those parameters is much simpler than the problem of estimating the entire distribution of the original random vector. Without going into the details (see [26] for a complete overview on the subject), a copula can be defined as:

Definition 3.1: Let $U = (U_1, U_2, ..., U_n)$ a n-variate random vectors such that each component U_j has a $[0,1]$ uniform distribution. Its cumulative distribution function is defined by:

$$F_U(x_1, x_2, \ldots, x_n) = P\{U_1 \leq x_1, \ldots, U_n \leq x_n\} \; ; \; x_i \in \mathbb{R} \quad (4)$$

The restriction of function F_U to the hypercube $[0,1]^n$ is called a copula and is denoted $C : [0,1]^n \to [0,1]$:

$$(u_1, ..., u_n) \mapsto C(u_1, ..., u_n) = F_U(u_1, ..., u_n) \quad (5)$$

The utility of copulas comes from the Sklar's theorem:

Theorem 3.2: Let F be the cumulative distribution function of a n-variate random vector $Y = (Y_1, ..., Y_n)$ and let F_j denote the cumulative distribution function of component Y_j. Then there exists a copula C such that:

$$F(x_1, ..., x_n) = C(F_1(x_1), ...F_n(x_n)). \quad (6)$$

Moreover if the functions F_i are continuous, C is unique and:

$$C(u_1, ..., u_n) = F(F_1^{-1}(u_1), ..., F_n^{-1}(u_n))$$

This theorem shows how to build the distribution F of a random vector given its marginal distributions F_i and a copula C: different copulas yield different distributions having the same marginal distributions. Therefore the modelling of a multivariate distribution is decomposed into two steps, the first one being the identification of the marginal distributions, the second one being the identification of the copula.

Simulation of such distributions which are described through marginal distribution and a copula is straightforward: simulate a n-variate random vector $U = (U_1, ..., U_n)$ such that each component U_j has a $[0,1]$ uniform distribution according to the n-variate distribution defined by the copula C and use relation (6) to generate vector Y:

$$Y = (F_1^{-1}(U_1), ..., F_n^{-1}(U_n))$$

The simulation of random vector U is done through the conditional distribution method (3) starting from n independant $[0,1]$ uniform random variables V_i and using the following property of copulas(when the following derivatives exist):

$$\begin{aligned} F_{U,2}(U_2|U_1) &= P\{U_2 \leq u, U_3 \leq 1, ...|U_1\} \\ &= \frac{\partial C}{\partial u_1}(U_1, u, 1, ..., 1) \end{aligned} \quad (7)$$

$$\begin{aligned} F_{U,3}(U_3|U_1,U_2) &= P\{U_3 \leq u, |U_1, U_2\} \\ &= \frac{\partial^2 C}{\partial u_1 \partial u_2}(U_1, U_2, u, 1, ..., 1), \end{aligned} \quad (8)$$

and so on. The simulation of vector U is done by solving:

$$\begin{cases} U_1 = V_1 = \mathcal{F}_1(V_1) \\ U_2 = \frac{\partial C}{\partial u_1}^{-1}(V_1, V_2, 1, ..., 1) = \mathcal{F}_2(V_1, V_2) \\ \vdots \\ U_n = \frac{\partial^{n-1} C}{\partial u_1 ... \partial u_{n-1}}^{-1}(V_1, \ldots, V_n) = \mathcal{F}_n(V_1, V_2, ..., Vn) \end{cases}$$

A functional relation $\mathcal{F}$ between the independant random variables V_j and Y can therefore be constructed:

$$\begin{cases} Y_1 = F_1^{-1} \circ \mathcal{F}_1(V_1) \\ Y_2 = F_2^{-1} \circ \mathcal{F}_2(V_1, V_2)) \\ \vdots \\ Y_n = F_n^{-1} \circ \mathcal{F}_n(V_1, V_2, ..., V_n) \end{cases}$$

Writing once again each random variable V_j in terms of independant Gaussian random variables G_j with distribution $\mathcal{N}(0,1)$ we construct a functional representation of process X in terms of independant Gaussian processes:

$$\begin{aligned} X(t) = f(G(t)) \ ; \ f = (f_1, ..., f_n) \\ G(t) = (G_1(t), ..., G_n(t)) \\ f_1(x) = F_1^{-1} \circ \mathcal{F}_1 \circ F_G(x_1) \ ; \ ... \ ; \\ f_n(x) = F_n^{-1} \circ \mathcal{F}_n(F_G(x_1), F_G(x_2), ..., F_G(x_n)) \end{aligned} \quad (9)$$

C. Construction of the underlying Gaussian process

1) Multivariate Hermite polynomials: We have seen that process X is written as the image of a n dimension Gaussian process through the map $f : \mathbb{R}^n \to \mathbb{R}^n$. In order to generalize the construction of scalar nonGaussian processes to the vector case we will need to project function f on a basis of the Hilbert space $L(\mathbb{R}^n, \nu(x)dx)$ where $\nu(x)dx$ is the standard Gaussian measure on $\mathbb{R}^n$. We will denote this last space by $\mathcal{L}^2$. Let us start by introducing multi-index notations.

Let $x = (x_1, ..., x_n) \in \mathbb{R}^n$ and $\alpha = (\alpha_1, ..., \alpha_n) \in \mathbb{N}^n$. We denote

$$x^\alpha = (x_1^{\alpha_1}, ..., x_n^{\alpha_n}) \ ; \ \alpha! = \alpha_1! ... \alpha_n! \ ; \ |\alpha| = \alpha_1 + ... + \alpha_n$$

Let $G(t) = (G_1(t), ..., G_n(t))$ a standard Gaussian variable on $\mathbb{R}^n$. Its distribution $\nu(x)dx$ is:

$$\nu(x)dx = \otimes_{i=1}^n \nu_i(x_i)dx_i = \otimes_{i=1}^n \sqrt{2\pi}^{-1} \exp -\frac{x_i^2}{2} dx_i$$

$$\nu(x)dx = (2\pi)^{-n/2} \exp(-||x||_2^2/2)dx$$

Definition 3.3: Let α be a multi-index, the normalized Hermite polynomial $H_\alpha(x)$ is defined by:

$$H_\alpha(x) = \prod_{i=1}^n H_{\alpha_i}(x_i)$$

with $H_{\alpha_i}(x_i)$ being the Hermite polynomial on $\mathbb{R}$.

Proposition 3.4: The family $\{H_\alpha(x)/\sqrt{\alpha!}\}_\alpha$ is an orthonormal basis of the Hilbert space $\mathcal{L}^2$:

$$E(H_\alpha(G)H_\beta(G)) = \int_{\mathbb{R}^n} H_\alpha(x)/\sqrt{\alpha!} \times \frac{H_\beta(x)}{\sqrt{\beta!}} \ \nu(x)dx = \delta_\alpha^\beta$$

(where G is a $\mathcal{N}(0,1)$ - random variable.)

Corollary 3.5: Let $f \in \mathcal{L}^2$,then f can be projected on the above basis:

$$f(x) = \sum_\alpha f_\alpha H_\alpha(x) \ ; \ f_\alpha = (f_{1,\alpha}, ..., f_{n,\alpha}) \in \mathbb{R}^n$$

$$\alpha! \, f_\alpha = < f, H_\alpha >_{\mathcal{L}^2} = \int_{\mathbb{R}^n} f(x) H_\alpha(x) \nu(x) dx \quad (10)$$

In particular, process $X(t)$ can be written as the series:

$$X(t) = \sum_\alpha f_\alpha H_\alpha(G(t)). \quad (11)$$

In the following we will use an approximation $X_M(t)$ of $X(t)$ given by the truncated series:

$$X_M(t) = \sum_{|\alpha| \leq M} f_\alpha H_\alpha(G(t)) \quad (12)$$

2) Melher formula for multivariate processes: In order to finish the construction of multivariate model of the non-Gaussian process we have to determine the autocorrelation of the underlying multivariate Gaussian process. Since we have assumed the Gaussian processes G_j to be independant, we have to determine their scalar autocorrelation function.

Writing the autocorrelation function of $X(t)$ using expansion (11) yields:

$$\begin{aligned} R_X(t) = E(X(s+t)X(s)^T) \\ = \sum_\alpha \sum_\beta f_\alpha f_\beta^T E(H_\alpha(G(t+s))H_\beta(G(s))) \end{aligned} \quad (13)$$

The term $E(H_\alpha(G(t+s))H_\beta(G(s)))$ is computed using the assumption that the processes $\{G_j(t)\}_{j=1,n}$ are independant, therefore:

$$E(H_\alpha(G(t+s))H_\beta(G(s))) = E(\prod_j H_{\alpha_j}(G_j(t+s)) \prod_k H_{\beta_k}(G_k(s)))$$

$$E(H_\alpha(G(t+s))H_\beta(G(s))) = \prod_j \prod_k E(H_{\alpha_j}(G_j(t+s))H_{\beta_k}(G_k(s)))$$

so

$$E(H_\alpha(G(t+s))H_\beta(G(s))) = \left(\prod_{j=1}^{n}\left[R_{G_j}(t)\right]^{\alpha_j}\alpha_j!\right)\delta_\alpha^\beta$$

The autocorrelation function of $X(t)$ can be written in terms of the unknown autocorrelation function of each G_j:

$$R_X(t) = \sum_\alpha f_\alpha f_\alpha^T \alpha! \prod_{i=1}^{n}\left[R_{G_i}(t)\right]^{\alpha_i} \tag{14}$$

As in the scalar case, this equation cannot be solved explicitely and is replaced by the following optimization problem: find n nonnegative definite scalar functions R_{G_j} which minimizes the quantity

$$\|R_X(t) - \sum_{|\alpha|\leq M} f_\alpha f_\alpha^T \alpha! \prod_{i=1}^{n}\left[R_{G_i}(t)\right]^{\alpha_i}\|_{L^2(\mathbb{R},\mathbb{R}^n)}$$

This problem is then written in terms of the spectral measures in order to get rid of the cumbersome positiveness constraint for the autocorrelation functions:

$$\min_{S_{G_j}} \Big\| \int_{\mathbb{R}} e^{i\lambda t} S_X(\lambda) d\lambda - \\ - \sum_{|\alpha|\leq M} \alpha!\, f_\alpha f_\alpha^T \prod_{j=1}^{n}\Big(\int_{\mathbb{R}} e^{i\lambda t} S_{G_j}(\lambda) d\lambda\Big)^{\alpha_j} \Big\|_{L^2} \tag{15}$$

under the constraints: the S_{G_j} are positive and even functions.

3) Convergences:

Proposition 3.6: The series $(X_M(t))_M$ converges in quadratic mean uniformly in t to $X(t)$.

Proof. We have $f_\alpha = (f_{1,\alpha}, ..., f_{n,\alpha})$ with $f_{i,\alpha} = E\left(f_i(G(t))\frac{H_\alpha(G(t))}{\alpha!}\right)$.
As the gaussian process $(G(t), t \in \mathbb{R})$ is stationary, f_α doesn't depend on t. Then for all fixed $t \in \mathbb{R}$, we have

$$X(t) = f\left(G(t)\right) = \sum_\alpha f_\alpha H_\alpha\left(G(t)\right),$$

the series being uniformly convergent in quadratic mean. □

Proposition 3.7: Let R_M be the autocorrelation function of the process $(X_M(t), t \in \mathbb{R})$.

$$\forall t \in \mathbb{R}, \quad \lim_{M\to\infty} R_M(t) = R_X(t).$$

Proof. Owing to the Mehler formula, we have

$$R_M(t) = \sum_{|\alpha|\leq M} f_\alpha f_\alpha^T \alpha! \prod_{i=1}^{n}\left[R_{G_i}(t)\right]^{\alpha_i}.$$

Which gives the result. □

Proposition 3.8: Let $r_M = \sum_{|\alpha|>M} f_\alpha f_\alpha^T \alpha!$ be the rest of the convergent series $\sum_\alpha f_\alpha f_\alpha^T \alpha! = R_X(0) = Id_{m\times n}$. Let consider the coefficients i, j of the considered matrices,

$$\forall t \in \mathbb{R},\ \forall i, j, \quad \Big|\left(R_X(t) - R_M(t)\right)_{ij}\Big| \leq (r_M)_{ij}\,.$$

Proof. The Cauchy-Schwarz inequality permits to write

$$\begin{aligned} \forall t,\ \forall j, \quad \left|R_{G_j}(t)\right| &= \left|E\left(G_j(0)G_j(t)\right)\right| \\ &\leq \sqrt{E(G_j(0)^2)E(G_j(t)^2)} \leq 1 \end{aligned}$$

Then

$$\begin{aligned} \Big|\left(R_X(t) - R_M(t)\right)_{ij}\Big| &= \left|\left(\sum_{|\alpha|>M} f_\alpha f_\alpha^T \alpha! \prod_{k=1}^{n}\left[R_{G_k}(t)\right]^{\alpha_k}\right)_{ij}\right| \\ &\leq \sum_{|\alpha|>M} f_\alpha f_\alpha^T \alpha! \Big|\max_k R_{G_k}(t)\Big|^{M+1} \\ &\leq \sum_{|\alpha|>M} f_\alpha f_\alpha^T \alpha! \\ &\leq r_M. \end{aligned}$$

□

4) Simulation of the nonGaussian multivariate processes: Once the solution S_{G_j} of the optimization problem (15) have been calculated, the simulation of process X is staightforward. Indeed one has only to generate simulated trajectories of n independant scalar Gaussian processes and use relation (12) to generate trajectories of $X_M(t)$.

The simulation of each Gaussian process $G_j(t)$ can be easily done using either a spectral approach or a Markovian model [17]. In any case, due to the number of optimization parameters appearing in problem (15), each unknown spectral density S_{G_j} should be written as, or approximated by a rational function, the same way as it is done in the Markovian approach in [17], in order to introduce a smaller number of parameters. This optimization problem is the difficult numerical part of the procedure and one has to be careful when choosing the domain $t \in [0, T]$ on which the error is constructed. Stochastic algorithm based methods are of course recommended for this particular problem.

Remarks 3.9: 1/ The data used for constructing the numerical model of a nonGaussian process is the distribution of a random vector Y and either the autocorrelation matrix $R_X(t)$ or the spectral matrix $S_X(\lambda)$ of X. One has of course to check the consistency of the data, more precisely that

$$R_X(0) = E(YY^T) = \int_{\mathbb{R}} S_X(\lambda) d\lambda \tag{16}$$

2/ Since the method explicitely use the spectral density of the process in the optimization problem, it can be applied only to the zero-mean process $X(t) - E(X(t))$.

IV. Numerical illustrations

A. Example I

In this example our goal is to simulate a stationnary process $X(t) = (X_1(t), X_2(t))$ whose first order marginal distribution is the uniform distribution over the unit disk :

$D = x^2 + y^2 \leq 1$ and such that its spectral matrix is given by:

$$S_X(\lambda) = \begin{bmatrix} \frac{0.2534}{\pi(1+\lambda^2)} & 0 \\ 0 & \frac{0.25}{\pi(1+\lambda^2)} \end{bmatrix} \quad (17)$$

The two components X_1 and X_2 are uncorrelated but dependant.

The first step is to construct the functional representation between a random vector $Y = (Y_1, Y_2)$ uniformly distributed on D and a vector $U = (U_1, U_2)$ of independant uniform variables. The following algorithm given in Devroye's book [19] can be used in order to simulate Y:

Generate a random variable $Y_1(\omega)$ with distribution

$$f_1(x) = \frac{2}{\pi}\sqrt{1-x^2} \; ; \; |x| \leq 1$$

Knowing the value of Y_1, generate random variable Y_2 with an uniform distribution over $[-\sqrt{1-Y_1^2} \; ; \; \sqrt{1-Y_1^2}]$.

Denoting F_{Y_1} the cumulative distribution function of Y_1, the functional representation can be written:

$$\begin{cases} Y_1 = F_{Y_1}^{-1}(U_1) \\ Y_2 = \sqrt{1-Y_1^2} \times (-1+2U_2) = \sqrt{1-(F_{Y_1}^{-1}(U_1))^2} \times ... \\ \quad ... \times (-1+2U_2) \end{cases} \quad (18)$$

Functions F_{Y_1} and $F_{Y_1}^{-1}$ are constructed numerically, but due to relation (10), $F_{Y_1}^{-1}$ is calculated only for the Gauss points used for computing the integral. In this example we have used 20 Gauss points for computing the Hermite expansion coefficients. We consider now the following memoryless transformation of the Gaussian vector $G(t) = (G_1(t), G_2(t))$:

$$\begin{cases} X_1(t,\omega) = f_1(G_1(t,\omega)) \\ X_2(t,\omega) = f_2(G_1(t,\omega), G_2(t,\omega)) \end{cases} \quad (19)$$

$$X(t,\omega) = f(G_1(t,\omega), G_2(t,\omega)) \quad (20)$$

The function f is projected on the Hermite polynomial basis:

$$f(x,y) = \sum_{i,j=0,N} f_{i,j} H_i(x) H_j(y) \; ; \; f_{i,j} \in \mathbb{R}^2 \; ; \; f_{i,j} = (f_{i,j}^1, f_{i,j}^2)$$

and the optimization problem (15) is solved using a simulated annealing algorithm where the unknown spectral density of Gaussian processes G_j is modeled as:

$$S_{G_j}(\lambda) = \frac{(a_{j,1} + b_{j,1}\lambda)(a_{j,2} + b_{j,2}\lambda)}{(\lambda^2 + 2c_{j,1}d_{j,1}\lambda + d_{j,2}^2)(\lambda^2 + 2c_{j,2}d_{j,2}\lambda + d_{j,2}^2)}$$

which yields 16 unknown parameters to be fitted. The non-Gaussian process can then be simulated using the truncated expansion (12). We first check the quality of the representation of the memoryless transformation as a truncated series (12) by estimating the the distribution of the marginal distribution of X. Figure 1 represents 50000 samples of the following 2

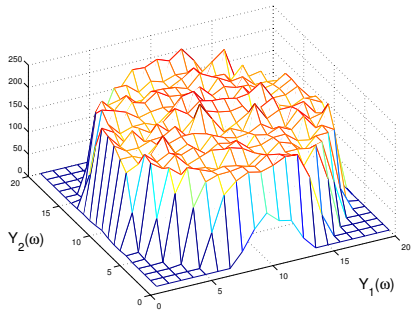

Fig. 1. Histogram of the uniform distribution over the unit disk

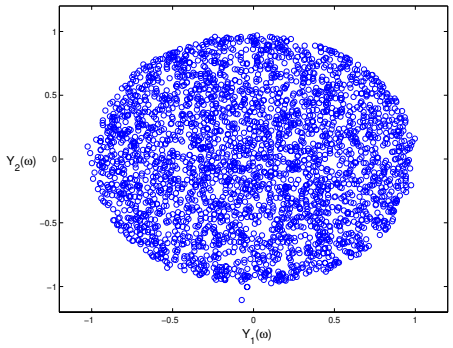

Fig. 2. Simulation using polynomial representation of the Rosenblatt's transformation

dimensional random vector

$$Y_N = \sum_{\alpha_1,\alpha_2 \leq N} f_\alpha H_{\alpha_1}(G_1) H_{\alpha_2}(G_2) \quad (21)$$

where G_1 and G_2 are independant normalized Gaussian variables. The order of truncation is $N = 6$. There are some points which lie outside the unit disk: this is a consequence of the truncation error. Figure 2 represents the histogram built from those samples: one can check that the distribution can be described as uniform over the disk. The next step is to solve the optimization problem (15). Figure 3 shows the comparison between the target autocorrelation function and the one obtained as a solution of the optimisation procedure: the agreement is excellent, even for the intercorrelation which was chosen null. Lastly the expansion (12) is used to simulate 500 trajectories of the nonGaussian process from which its spectral measure is estimated, each trajectory being discretized using 1024 points. Figure 4 shows the comparison between the target spectral measure and the estimated one. And finally a trajectory of process X is drawn on Figure 5.

Remark 4.1: In the same way, it is possible to simulate non-Gaussian processes with uniform distribution on hyperspheres, hyperellipsoids, triangles, etc.

B. Example II

In this last example, the distribution of a stationnary random process $X(t) = (X_1(t), X_2(t))$ will be defined through the marginal distribution of X_1 and X_2 and a copula C. More

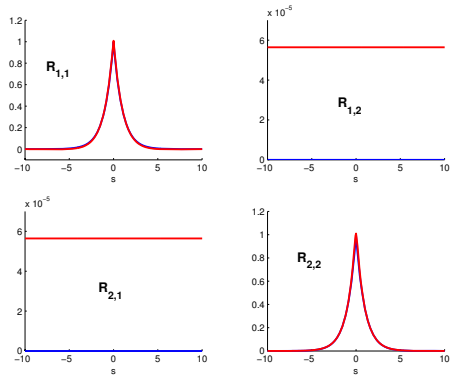

Fig. 3. Comparison between target auto correlation and solution of the optimization problem

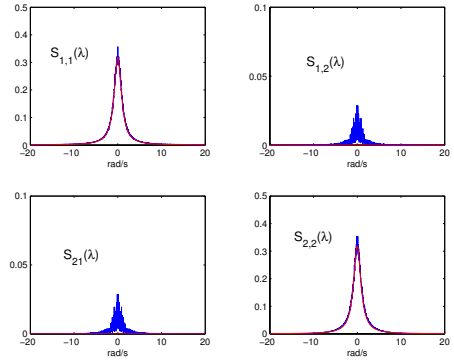

Fig. 4. Comparison between target and estimated spectrum

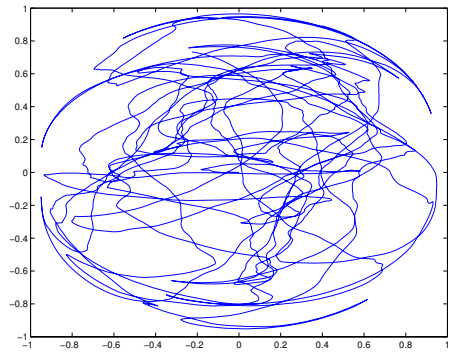

Fig. 5. Trajectory of the nonGaussian process

precisely let X be distributed as a random vector $Y = (Y_1, Y_2)$ such that Y_1 and Y_2 have a parameter 1 exponential distribution and such that the dependancy between Y_1 and Y_2 be defined by the Gumbel copula C:

$$C_\alpha(u,v) = \exp\{-[(-\log(u))^\alpha + (-\log(v))^\alpha]^{1/\alpha}\} \ ; \ \alpha \geq 1.$$

The target autocorrelation function is:

$$S_X(\lambda) = \begin{bmatrix} \frac{1}{2.22(1+\lambda^4)} & \frac{1}{4.44(1+\lambda^4)} \\ \frac{1}{4.44(1+\lambda^4)} & \frac{1}{2.22(1+\lambda^4)} \end{bmatrix} \qquad (22)$$

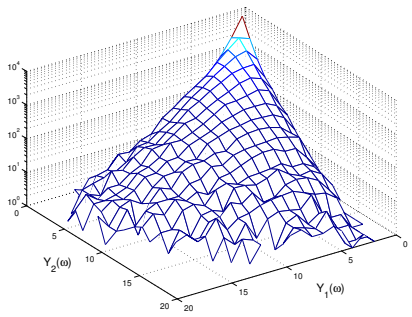

Fig. 6. Histogram of the distribution

The functional representation of X in terms of independant Gaussian processes can be derived by the general relations (9),(9) and (9), but here, we will use a specific and simpler algorithm to simulate a two dimension random vector (U_1, U_2) according the Gumbel's copula:

generate 2 independant uniform variables over $[0,1]$, V_1 et V_2.

solve

$$W = K^{-1}(V_2) \ ; \ K(t) = t - \frac{t\log(t)}{\alpha}$$

construct $U_1$and U_2 using relations

$$U_1 = \phi^{-1}[\phi(W)V_1] \ ; \ U_2 = \phi^{-1}[\phi(W)(1-V_1)]$$

where $\phi(t) = (-\log(t))^\alpha$ defines the Gumbel's copula and $\phi^{-1}(t) = e^{-t^{1/\alpha}}$

Remarks 4.2: This algorithm cannot be generalized to higher dimension copulas.

Function K is the cumulative distribution function of random variable $C(V_1, V_2)$.

The functional representation of X is therefore given by:

$$X_1(t) = -\log(\phi^{-1}[\phi(K^{-1}(F_G(G_2(t)))F_G(G_1(t))])$$

$$X_2(t) = -\log(\phi^{-1}[\phi(K^{-1}(F_G(G_2(t)))(1-F_G(G_1(t)))])$$

As in the two previous examples results of this approach is illustrated through figures 6-10.

ACKNOWLEDGMENT

This research was supported by the French National Research Agency grant ANR-06-CIS6-003.

REFERENCES

[1] Monbet V., Aillot P., and Prevosto M. Survey of stochastic models for wind and sea state time series. *Prob. Eng. Mech.*, 22:113–126, 2007.
[2] Monbet V. and Prevosto M. Bivariate of non stationary and non gaussian observed processes.application to sea state parameters. *Applied Ocean Research*, 23:139–145, 2007.
[3] Fouques S., Myrhaug D., and Nielsen M. Seasonal modeling of multivariate distributions of metocean parameters with application to marine operations. *J. Offshore mech. Artc.*, 126:202–212, 2004.
[4] Gusev A. Peak factors of mexican accelerograms: Evidence of a non-gaussian amplitude distribution. *J. Geophys. Res.*, 101(B9):20083–20090, 1996.

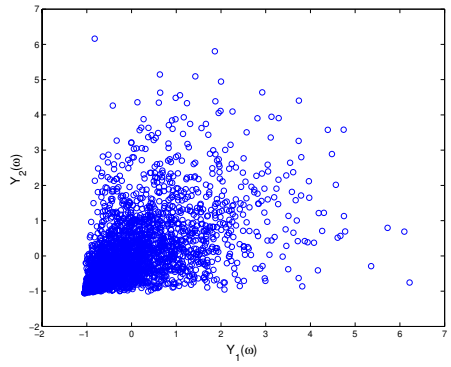

Fig. 7. Simulation using polynomial representation of the Rosenblatt's transformation

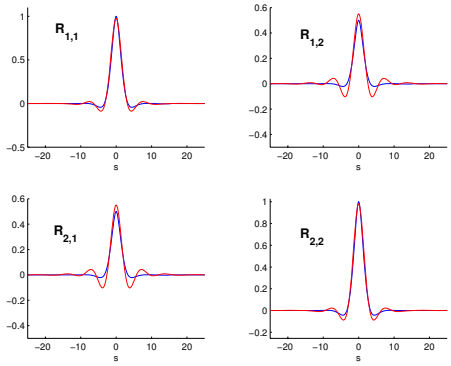

Fig. 8. Comparison between target auto correlation and solution of the optimization problem

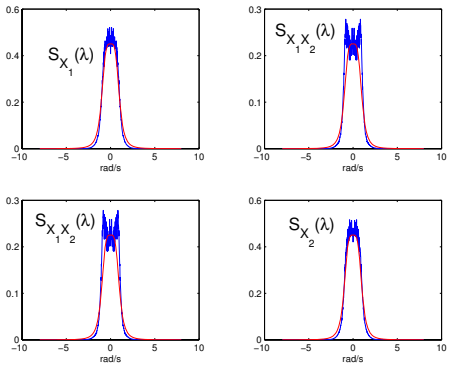

Fig. 9. Comparison between target and estimated spectrum

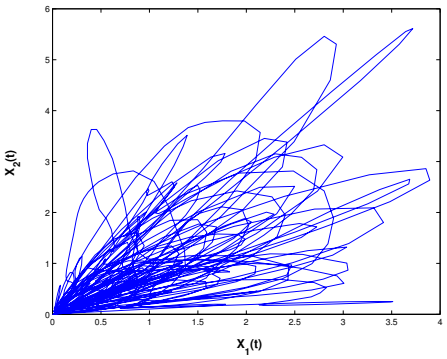

Fig. 10. Trajectory of the nonGaussian process

[5] Grigoriu M. and Kafali G. Non-Gaussian model for spatially coherent seismic ground motions. In Der Kureghian Madanat and Pestanat. , editors, *Application of Staistics and probablity in Civil Engineering*. Millpress, 2003.

[6] Nielsen M., Larsen G., and Hansen K. Simulation of inhomogeneous, non-stationary and non-gaussian turbulentwinds. *J. of Physics: Conference series 75*, 75:1–9, 2007.

[7] Nielsen M., Højstrup J., Hansen K., and Thesberg L. Validity of the assumption of Gaussian turbulence. In *Proceedings of the European Wind Energy Conference, Denmark*, July 2001.

[8] Rocha G. and al. . Simulation of non-gaussian cosmic microwave background maps. *Mon. Not. R. Astron. Soc.*, 357:1–11, 2005.

[9] Grigoriu M. Simulation of stationary non-gaussian translation processes. *j. of Eng. Mech. ASCE*, 124(2):pp12–127, 1998.

[10] Gurley K. and Kareem A. Simulation of non-gaussian processes. In Spanos , editor, *Computational Stochastic Mechanics, CSM98*, pages pp 11–21, Rotterdam, 1999. Balkema.

[11] IASSAR(Report) . A state-of-the-art report on computational stochastic mechanics. In Shinozuka M. and Spanos P.D., editors, *Probabilistic Engineering Mechanics*, volume 12(4), pages 197–321, 1997.

[12] Poirion F. Numerical simulation of homogeneous non-Gaussian random vector fields. *J. of Sound and Vibration*, 160(1):25–42, 1993.

[13] Popescu R., Deodatis G., and Prevost J.H. Simulation of homogeneous nongaussian stochastic vector fields. *Prob. Eng. Mech*, 13(1):1–13, 1998.

[14] Sakamoto S. and Ghanem R. Simulation of non-Gaussian fields with the Karhunen-Loeve and polynomial chaos expansions. In *Engineering Mechanics Conference, Baltimore (USA)*, June 1999.

[15] Sakamoto S. and Ghanem R. Simulation of multi-dimensinal non-Gaussian non-stationary random fields. *Probabilistic Engineering Mechanics*, 17(2):167–176, 2002.

[16] Li L.B., Phoon K.K., and Quck S.T. Comparison between karhunen-love expansion and translation-based simulation of non-gaussian processes. *Computers and Structures*, 85:264–276, 2007.

[17] Puig B., Poirion F., and Soize C. Non-gaussian simulation using hermite polynomial expansion: convergences and algorithms. *Prob. Eng. Mech.*, 17(3):253–264, 2002.

[18] Poirion, F and Puig, B White noise and simulation of ordinary Gaussian processes. *Monte Carlo Methods and Appl.*, 10(1): 9–89,2004.

[19] Devroye L. *Non Uniform Random Variate Generation*. Springer-Verlag, New-York, 1986.

[20] Rosenblatt M. Remarks on a multivariate transformation. *Ann. Math.*, 23:470–472, 1952.

[21] Rubinstein R.Y. *Simulation And The Monte Carlo Method*. John Wiley & Sons, 1981.

[22] Luciano Elisa, Vecchiato Walter, and Cherubini Umberto. *Copula Methods in Finance*. Wiley, John & Sons, Incorporated, 2004.

[23] Xiaohong C., Wu W., and Yanping Y. Efficient estimation of copula-based semiparametric markov models, 2009.

[24] Brdossy A. and Li Y. Geostatistical interpolation using copulas. *Water Resour. Res.*, 44, 2008.

[25] Yan-Heng Li, Bao-ping Shi, and Jian Zhang. Copula joint function and its application in probability seismic hazard analysis. *Acta Seismologica Sinica*, 21(3):296–305, 2008.

[26] Nelsen R. *An Introduction to Copulas*. Springer Series In Statistics. Springer, 2006.