USING SURROGATES AND OPTIMAL TRANSPORT FOR SYNTHESIS OF STATIONARY MULTIVARIATE SERIES WITH PRESCRIBED COVARIANCE FUNCTION AND NON-GAUSSIAN JOINT-DISTRIBUTION

Pierre Borgnat, Patrice Abry, Patrick Flandrin

CNRS, École Normale Supérieure de Lyon, Laboratoire de Physique. Lyon, France

ABSTRACT

Surrogates are investigated as procedures of synthesis for multivariate time series with prescribed properties. First it is shown how to prescribe a multivariate covariance function jointly with the (possibly non-Gaussian) marginal distributions. Second, using histogram matching by approximate optimal transport with the Sliced Wasserstein Distance, the surrogate synthesis is extended to prescribe covariance function and joint-distribution of the components. Algorithms are described and justified, and numerical examples are shown. MATLAB codes are publicly available online.

***Index Terms*—** Surrogate, Numerical Synthesis, Multivariate Series, Optimal Transport, Sliced Wasserstein Distance

1. INTRODUCTION

Improvements in data acquisition leads more and more to multivariate time series. Measurements from sensor networks, computer networks, devices of environmental or health monitoring, are as many examples. These multivariate signals are usually non-Gaussian and correlated. One challenge is the synthesis of stationary multivariate time series that have prescribed probability distributions of values and specified covariance function (both auto- and cross-covariance).

The contribution of this work is to devise synthesis procedures relying on surrogate methods [1, 2]. Whereas synthesis of Gaussian multivariate time series is known, thanks for instance to the circulant embedding methods initially proposed in [3, 4] (see also [5] and references therein), the case of series with prescribed covariance function and non-Gaussian distributions is harder, with few solutions (see [6] and references therein). Surrogates, introduced initially in non-linear physics [1, 2], are a way of synthesising series matching empirical properties of some observed data. Our first contribution is to elucidate how surrogates can be used for an elegant synthesis of many different time series with prescribed properties. Let $X(n)$ be a multivariate series with M components and $x_j(n)$ with $n = 0, ..., N-1$ its j-th component. Let C be a theoretical prescription of the stationary covariance C of X. It is defined as $C_{jk}(n) = \mathbb{E}\{x_j(t)x_k(t+n)\} - \mathbb{E}\{x_j(t)\}\mathbb{E}\{x_k(t+n)\}$, for $n = 0, ..., N-1$ and $i, j = 1...M$. Let also $p_j(x_j)$ be a theoretical prescription of the marginal distribution of component x_j. The first objective is to synthesise multivariate series having this covariance function C with these marginals p_j. We explain how surrogates can be generated as a solution. This is presented in Section 2.

Recent works, inspired by [7], tackle the non-Gaussian synthesis challenge, where non-Gaussian series are obtained by a mapping from Gaussian series. The issue is to inverse this mapping for the covariance. Bivariate series were considered in [8] then multivariate non-Gaussian series in [6] using an elaborated mapping that can be approximately computed and reversed via Hermite expansions. An advantage of the new method with surrogates is that it does not involve such a numerical reversion, a hard step as told in [6].

Furthermore, and this is the contribution in Section 3, surrogate synthesis can go one step further in the prescriptions: not only the marginal distribution can be prescribed, but also the joint-distribution of the series at a given time. For that, we show how to modify the surrogates with optimal transport. A practical solution for optimal transport has been introduced in [9] using a Sliced Wasserstein Distance. It fills nicely the need of multi-dimensional histogram matching in the surrogate method. In Section 3.1, some background is recalled on optimal transport and it is shown in 3.2 how to synthesise series with prescribed covariance and joint-distributions. An example is given in 3.3. Conclusion is in Section 4.

2. PRESCRIBED COVARIANCE AND MARGINALS

2.1. Background: Classical Surrogates of Multivariate Series

From a given series, the main idea of surrogate is to synthesise new stationary data by a randomisation in the Fourier domain [1, 2].

Multivariate Surrogates. For multivariate series, cross-correlations should be kept [10]. The randomisation in the Fourier domain is chosen so that the differences of phase between components stay the same. Let $X(n)$ be a multivariate series with M components ($x_j(n)$ with $n = 0, ..., N-1$ being its j-th component). For initialisation, compute the Fourier transform of each component:

$$(\mathbb{F}x_j)(f) = \sum_{n=0}^{N-1} x_j(n) e^{-i2\pi nf/N} = A_{x_j}(f)\, e^{i\Psi_{x_j}(f)} \qquad (1)$$

The algorithm for multivariate surrogates is as follows:

ALGORITHM 0:
Input $A_{x_j}(f)$ and $\Psi_{x_j}(f)$ for $j = 1, ..., M$, $f = 0, ..., N-1$.
1. Draw a random phase $\Theta(f)$, i.i.d., uniform in $[0, 2\pi]$.
2. Independently for each component j, do a phase randomisation in the Fourier domain by adding $\Theta(f)$ (the same for each j) so that

$$s_j(n) = \frac{1}{N} \sum_{f=0}^{N-1} A_{x_j}(f) e^{i(\Psi_{x_j}(f)+\Theta(f))} e^{i2\pi nf/N}. \qquad (2)$$

3. Form the multivariate surrogate $S(n) = [s_1(n), ..., s_M(n)]^t$.
Output S.

As shown in [10] using the Wiener-Khintchine theorem, the surrogate S has the same cross-covariance structure as X because $(\mathbb{F}s_j)^*(f)(\mathbb{F}s_k)(f) = A_{x_j}(f)A_{x_k}(f)e^{i(\Psi_{x_k}(f)-\Psi_{x_j}(f))}$, hence is equal to $(\mathbb{F}x_j)^*(f)(\mathbb{F}x_k)(f)$. A proof that the surrogates are stationary holds [11, 12]. Finally these surrogates are Gaussian if N is large enough (as sums of randomised Fourier modes) [2].

Synthesis with Gaussian Surrogates. There are two strategies to use these surrogates to synthesise Gaussian series:
1) The target series should follow an **empirical prescription** from some **measured data** X. This is the original framework of surrogates and, given X, the algorithms described above are directly applicable with initialisation by eq. (1).
2) The target series is given through a **model**, with a **theoretical prescription** of stationary covariance function C (standing for the $C_{jk}(n)$ for $n = 0, ..., N-1$ and $i, j = 1...M$). For the surrogate algorithm, C has to be first transformed into Fourier amplitudes $A_{x_j}(f)$ and phases $\Psi_{x_j}(f)$ (these latter impose the cross-covariance) of one realisation X, before generating new realisations by ALGORITHM 0. For this seed Gaussian series X, we advocate the use of circulant embedding methods [3, 4, 5] as the state-of-the-art.

In that case, one could wonder whether there is any need for surrogates if one should know how to synthesise a seed Gaussian series by another method before using surrogates. A first answer is that, being based on two Fourier transforms only, the surrogate algorithm is a quick way to obtain more realisations. Applying this algorithm with independently drawn $\Theta(f)$ will generate new and independent series. A second answer is, and this is the contribution hereafter, that surrogates are easily adapted to prescribe also the marginal of each component (Section 2.2) or their joint-distribution (Section 3).

2.2. Synthesis with Prescribed Covariance and Marginals

Multivariate Surrogate Algorithm for Non-Gaussian Marginals. From [2], a strategy is designed so that the surrogate is also constrained to have the same marginal distribution as the original series. For that, an iterative procedure alternates projection on the two constraints (the covariance function expressed in the Fourier domain, and the prescribed marginal distributions). This is called the **IAAFT** (Iteratively Amplitude Adjusted Fourier Transform) surrogate.

ALGORITHM **IAAFT** surrogate:
Input $A_{x_j}(f)$, $\Psi_{x_j}(f)$ and $x_j(n)$ for $j = 1, ..., M$, and for n and $f = 0, ..., N-1$
Initialisation: $r_j^{(1)}(n)$ is a classical surrogate S from ALGORITHM 0. The prescribed values v_j are the rank-ordered values of x: $v_j = \mathrm{sort}(x_j)$. At iteration l, one applies the two steps:
Step 1. Projection on the prescribed covariance. Form:

$$(\mathbb{F}r_j^{(l)})(f) = \sum_{n=0}^{N-1} r^{(l)}(n) e^{i2\pi \frac{nf}{N}} = A_{r_j^{(l)}}(f)\, e^{i\Psi_{r_j^{(l)}}(f)} \quad (3)$$

and transform it back by replacing the amplitudes by the desired ones $A_{x_j}(f)$ while keeping the phase $\Psi_{r_j^{(l)}}(f)$ of this iteration:

$$s_j^{(l)}(n) = \frac{1}{N} \sum_{f=0}^{N-1} A_{x_j} e^{i\Psi_{r_j^{(l)}}(f)} e^{-i2\pi \frac{nf}{N}}. \quad (4)$$

Step 2. Projection on the prescribed marginal distributions. Independently on each component, apply the **rank ordering mapping** with the prescribed values v_j:

$$r_j^{(l+1)}(n) = v_j(\mathrm{rank}(s_j^{(l)}(n))). \quad (5)$$

Stop iterations: Define the multivariate surrogates $R = [r_1, ..., r_M]^t$ and $S = [s_1, ..., s_M]^t$. Convergence when $R \simeq S$, or when R and/or S do not evolve anymore from one iteration to the next.
Output S and/or R.

Here, recall that the **rank** of each values of a series s_j is defined by $\mathrm{rank}(s_j(n)) = k$ if $s_j(n)$ is the k-th smallest value in s_j.

The algorithm converges to a fixed point $r_j^{(l+1)} = r_j^{(l)}$ so that R has the same marginal distributions as X and approximately its covariance (whereas S has the exact covariance and approximative marginal distributions). In [2], accuracy is further discussed. Another way to look at convergence is to realise that the algorithm is mostly alternating projections on convex sets.

ALGORITHM 1:
Input covariance $C_{jk}(n)$ for $n = 0, ..., N-1$ and and $i, j = 1...M$, and marginal distributions $p_j(v_j)$
1. For the desired covariance $C_{jk}(n)$, create a Gaussian signal X with circulant embedding methods (cf. [3, 4, 5])
2. Compute amplitude $A_{x_j}(f)$ and phase $\Psi_{x_j}(f)$ of the Fourier transform of each component $j = 1, ..., M$, eq. (1)
3.a For each j, draw $v_j(n)$, $n =\in \{1..N\}$ from desired $p_j(v_j)$
3.b Sort values: $v_j = \mathrm{sort}(v_j)$
4. Initialise ALGORITHM **IAAFT** by $R^{(1)} = S$ from ALGORITHM 0 (classical surrogate) with random draw of $\Theta(f)$; see eq. (2)
5. Apply the iterations of ALGORITHM **IAAFT**, eq. (3), (4) and (5)
6. **Stop** if R close enough to S or they do not change
Output R and S

Table 1. **Prescribe Covariance and Marginal Distributions**

Synthesis of Series with Prescribed Covariance Function and Marginals. As in Section 2.1, there are two strategies to synthesise multivariate series with prescribed properties:
1) The usual surrogate framework where a **measured** multivariate series X imposes its empirical Fourier amplitudes $A_{x_j}(f)$ and phases $\Psi_{x_j}(f)$, and its empirical marginals (values of x_j). Per above, surrogates S and R will share these properties with X.
2) If a **model** is given, the covariance C and the desired marginal distributions $p_j(x)$ are prescribed. They are mapped into one realisation of the prescription before generating IAAFT surrogate:

a) The Fourier amplitudes $A_{x_j}(f)$ and phases $\Psi_{x_j}(f)$ are computed by synthesising a seed Gaussian series X having the desired covariance structure, using circulant embedding methods [3, 4, 5], then using Fourier transform (eq. (1)).
b) For each j, $v_j(n)$ are N values drawn i.i.d. from $p_j(v)$ using classical random generators (e.g., inversion method or acceptance-rejection method).

As *Step 2* of the IAAFT surrogates uses rank-ordered distributions of v_j, prescribing these values independently from the covariance is possible. The method ALGORITHM 1 is summarised in Table 1. Each call to ALGORITHM 1 returns a new surrogate S (or R).

2.3. Example

An example using ALGORITHM 1 is reported in Fig. 1. It simulates a non-Gaussian multivariate Moving Average process of order 1 and dimension $M = 3$, defined as follows:
– Covariance function $C(n)$ is prescribed from the recurrence $X(n+1) = \Phi * X(n-1) + E(n)$ where E is i.i.d. Gaussian noise, of variances 1 and $\Phi = [[0.8\ 1.0\ 0.0]; [0.0\ 0.2\ 0.0]; [0.2\ 1.0\ 0.5]]$.
– Marginal p_1 is uniform; p_2 is a triangular distribution; p_3 is a Gamma distribution, with $\alpha = 2.2$ and $\beta = 1.45$.

We were careful to prescribe $C_{jj}(0) = \mathrm{Var}(p_j)$ so that the constraints are compatible. Also, ALGORITHM 1 returns fluctuations of each component around the mean (non-zero for p_3). On Fig. 1 are shown examples of the time series, empirical estimations and prescribed forms for the marginal distributions, the 3 auto-covariances and 3 cross-covariances (non-time reversible). The resulting series and estimates show no discrepancy w.r.t. the target.

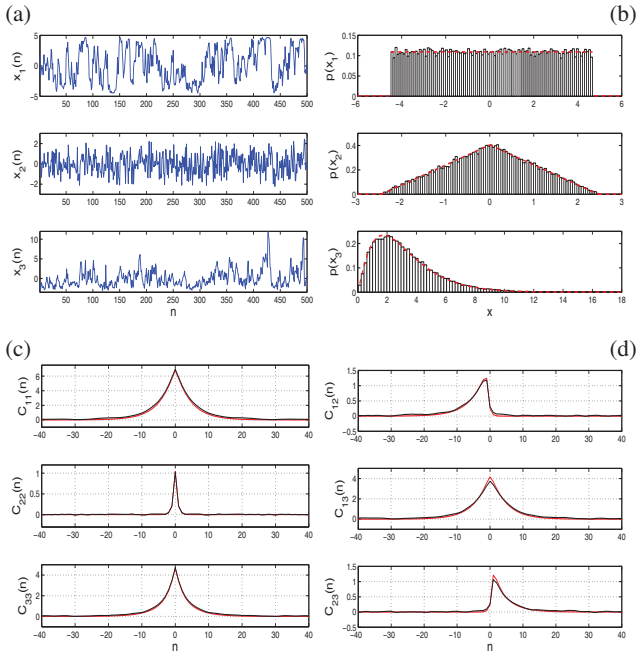

Fig. 1. Example with ALGORITHM 1: Covariance of a Moving Average and marginals as defined in Section 2.3. (a) Zoom on the time series. (b) Empirical estimations (bars) and prescribed forms (dashed red lines) for the marginal distributions. (c) Auto-covariances and (d) Cross-covariance for surrogates (black) and model (red). All empirical estimates on surrogates are done with $N = 2^{15}$.

3. JOINT-DISTRIBUTIONS AND OPTIMAL TRANSPORT

A recent work [9] has studied applications of optimal transport to multi-dimensional histogram matching. It shows how to compute approximately yet practically the optimal transport to go from one histogram to another. From that, not only the marginal distribution p_j can be prescribed for surrogates but also the joint distribution $P(x_1, ..., x_M)$ at a given instant n, assuming that it is stationary (hence independent of n). Indeed, in the IAAFT algorithm, Step 2 of the iteration given by the *rank ordering*, eq. (5), is in fact the *optimal histogram matching mapping in 1D*. Replacing it by the optimal histogram matching in M dimensions to a distribution P allows us to prescribe that the surrogates will have P as joint-distribution.

Let us first recall basics on optimal transport and on the solution proposed in [9] before detailing the new algorithm.

3.1. Optimal Transport with Sliced Wasserstein Distance

The optimal transport between two distributions Y_k and Z_k, $k = 1, ..., N$, is the assignment $k \to \sigma^*(k)$ (where $\sigma^* \in \Sigma_N$, the set of possible permutations of N elements) that minimises the quadratic Wasserstein distance: $W_\sigma(Y, Z)^2 = \sum_k ||Y_k - Z_{\sigma(k)}||^2$. Solution of this problem involves a linear programming with prohibitive computations for large N. An alternative metric coined "Sliced Wasserstein Distance" was proposed in [9]:

$$\tilde{W}_\sigma(Y, Z)^2 = \int_{\theta\in\Omega} \min_{\sigma_\theta \in \Sigma_N} \sum_k ||\langle Y_k - Z_{\sigma_\theta(k)}, \theta\rangle||^2 \mathrm{d}\theta, \qquad (6)$$

where σ_θ is the optimal transport for the points projected on a line defined by the unit vector $\theta \in \Omega = \{u \in \mathbb{R}^M, \text{s.t.} ||u|| = 1\}$. In 1D optimal transport is given by the rank ordering mapping of eq. (5).

It follows that a stochastic gradient descent can minimise $\tilde{W}_\sigma(Y, Z)^2$ and finds an approximate optimal transport. Starting from Y, at each iteration a random direction θ_k is taken and the descent update reads

$$Y^{(k+1)} = Y^{(k)} - \eta_k \left(Y^{(k)} - \langle Z_{\sigma^*_{\theta_k}}, \theta_k \rangle \right) \qquad (7)$$

where $\sigma^*_{\theta_k}$ is the optimal rank ordering from $\langle Z, \theta_k\rangle$ to $\langle Y^{(k)}, \theta_k\rangle$. Convergence is discussed in [9] and works well with $\eta_k \leq 1$ in practice. Not only does it give the distance of eq. (6), but also the optimal transport from Y to Z for this distance. We note it $\tilde{\sigma}^*_{Y,Z}$, and keep in mind that it is computed with the iterative gradient descent.

ALGORITHM 2:
Input covariance $C_{jk}(n)$ for $n = 0, ..., N-1$ and $i, j = 1...M$, and joint-distribution $P(v_1, ..., v_M)$
1. For the desired covariance $C_{jk}(n)$, create a Gaussian signal X with circulant embedding methods (cf. [3, 4, 5])
2. Compute amplitude $A_{x_j}(f)$ and phase $\Psi_{x_j}(f)$ of the Fourier transform of each component $j = 1, ..., M$, eq. (1)
3. Draw N independent vectors $V(n)$ from desired $P(v_1, ..., v_M)$
4. Initialise ALGORITHM **IAAFT** by $R^{(1)} = S$ from ALGORITHM 0 (classical surrogate) with random draw of $\Theta(f)$; see eq. (2)
5. Iterate a modified **IAAFT** ALGORITHM:
a. *Step 1*: apply eq. (3) and (4) to obtain $S^{(l)}$
b. *Step 2*: approximate optimal transport (as in 3.1), to map $S^{(l)}$ to values of V. Result: $R^{(l+1)}(n) = V(\tilde{\sigma}^*_{S^{(l)},V}(n))$
6. **Stop** if R close enough to S or they do not change
Output R and S

Table 2. **Prescribe Covariance Function and Joint-Distribution**

3.2. Synthesis with Prescribed Covariance and Joint-Distribution

When prescribing both the covariance structure and a stationary joint distribution $P(x_1, ..., x_M)$ (this includes marginal distributions $p_j(x) = \int P(x_1, ..., x_M) \prod_{k\neq j} \mathrm{d}x_k$), the proposed surrogate algorithm for synthesis of multivariate time series modifies ALGORITHM 1 by including the computation of approximate optimal transport in replacement of Step 2 of **IAAFT** surrogate. It follows an ALGORITHM 2, sketched in Table 2.

Because the individual algorithms (**IAAFT** and computation of approximate optimal transport) are guaranteed to converge, this algorithm will converge. Again, the algorithm is chiefly an instance of alternating projections on convex sets. It is possible that R and S are not exactly the same if the constraints are not exactly possible jointly. R is exact for the joint-distribution and S for the covariance.

3.3. Example

An example using ALGORITHM 2 is reported in Fig. 2. It simulates a non-Gaussian multivariate of dimension $M = 2$ given as follows:
– Covariance function $C(n)$ is given by exponentially decreasing functions: $C_{jk} = \gamma_{jk} e^{-\alpha_{jk} n}$ with parameters $\alpha_{11} = 0.5$, $\alpha_{22} = 1$, $\alpha_{12} = 0.7$; $\gamma_{11} = 1$, $\gamma_{22} = 1$, $\gamma_{12} = 0.7$.
– Joint-distribution $P(x_1, x_2)$ is so that marginal p_1 is uniform and p_2 is a triangular distribution, and for each point $x_1(n) = x_2(n) + U(n)$ where U is an i.i.d. uniform centred random variable (hence the triangular distribution for the marginal p_2).

We were careful to prescribe $C_{jj}(0) = \mathrm{Var}(p_j)$, and $C_{12}(0) = \int x_1 x_1 P(x_1, x_1) \mathrm{d}x_1 \mathrm{d}x_2$ so that the two constraints are compatible. On Fig. 2, are shown examples of the time series, empirical

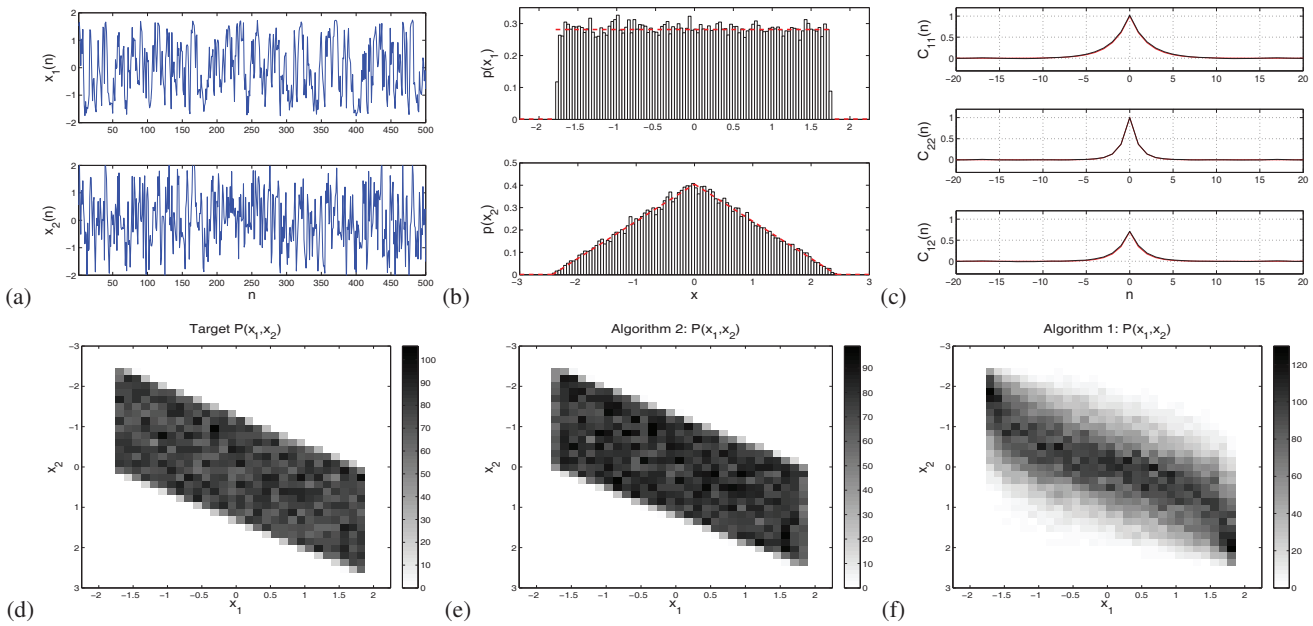

Fig. 2. Example with ALGORITHM 2: Covariance and joint-distributions as defined in 3.3. (a) Zoom on the time series. (b) Empirical estimations (bars) and prescribed forms (dashed red lines) for the marginals. (c) Auto- and Cross-covariances for surrogates (black) and model (red). (d) Target empirical joint-distribution (values $V(n)$ drawn for point 3 of ALGORITHM 2); (e) Obtained empirical joint-distribution with ALGORITHM 2; (f) Comparison to empirical joint-distribution obtained with ALGORITHM 1. All estimates are done with $N = 2^{15}$.

estimations and prescribed forms for the marginal distributions, the 2 auto-covariances and the cross-covariance. The result shows no discrepancy w.r.t. the target. On the second line, the empirical target and obtained joint-distribution are drawn: they are similar. On the contrary, if one would use ALGORITHM 1, the marginals and covariances are the same, whereas the joint-distribution departs from the target: it is shown in Fig. 2 (f). It does not respect the constraint of the prescribed joint-distribution that $x_1(n) \in [x_2(n) + \min(U); x_2(n) + \max(U)]$.

4. CONCLUSION

A method of surrogates is developed to synthesise many different multivariate time-series, where the covariance function and maginal or joint- distributions are prescribed either through a model (C and P or p_j) or by empirical properties of measured series. To obtain more realisations of the same series, it is enough to iterate the proposed algorithms. Future work will involve a thorough comparison in the case of prescribed marginal with the technique of [6]. All computation are done using MATLAB codes that are publicly available[1].

The method can find applications on real-world data. For instance, internet traffic has packet or bytes count time series with an interesting bivariate structure; sensor networks for environmental monitoring are another instance of joint measurements of temperature, pressure, rain,... On the methodological side, the method opens the way to follow [11] and study multivariate stationarity tests.

5. REFERENCES

[1] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. D. Farmer, "Testing for nonlinearity in time series: the method of surrogate data," *Physica D*, vol. 58, no. 1–4, pp. 77–94, 1992.

[2] T. Schreiber and A. Schmitz, "Surrogate time series," *Physica D*, vol. 142, no. 3-4, pp. 346–382, 2000.

[3] A.T.A Wood and G. Chan, "Simulation of stationary Gaussian processes in $[0, 1]^d$," *J. of Comput. and Graph. Stat.*, vol. 3, no. 4, pp. 409–432, 1994.

[4] C.R. Dietrich and G.N. Newsam, "A fast and exact method for multidimensional gaussian stochastic simulations," *Water Resourc. Res.*, pp. 2861–2869, 1993.

[5] H. Helgason, V. Pipiras, and P. Abry, "Fast and exact synthesis of stationary multivariate Gaussian time series using circulant embedding," *Signal Processing*, vol. 95, no. 5, pp. 1123–1133, 2011.

[6] H. Helgason, V. Pipiras, and P. Abry, "Synthesis of multivariate stationary series with prescribed marginal distributions and covariance using circulant embedding," *Signal Processing*, vol. 91, no. 8, pp. 1741–1758, 2011.

[7] R. Price, "A useful theorem for nonlinear devices having Gaussian inputs," *IRE Trans. Inform. Theory*, vol. IT-4, pp. 69–72, 1958.

[8] A. Scherrer and P. Abry, "Synthèse de processus bivariés non Gaussiens à mémoires longues," in *22nd GRETSI Symposium on Signal and Image Processing*, Dijon, 2009.

[9] J. Rabin, G. Peyré, J. Delon, and M. Bernot, "Wasserstein barycenter and its application to texture mixing," in *Proc. SSVM'11*, 2011.

[10] D. Prichard and J. Theiler, "Generating surrogate data for time series with several simultaneously measured variables," *Physical Review Letters*, vol. 73, no. 7, pp. 951–954, 1994.

[11] P. Borgnat, P. Flandrin, P. Honeine, C. Richard, and J. Xiao, "Testing stationarity with surrogates: A time-frequency approach," *IEEE Trans. Signal Processing*, vol. 58, no. 7, pp. 3459–3470, 2010.

[12] C. Richard, A. Ferrari, H. Amoud, P. Honeine, P. Flandrin, and P. Borgnat, "Statistical hypothesis testing with time-frequency surrogates to check signal stationarity," in *Proc. IEEE ICASSP*, Dallas, TX, USA, March 2010, pp. 720–724.

[1] http://perso.ens-lyon.fr/pierre.borgnat/codes.html