Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation 187 (2021) 449–467
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Original articles

Estimation of distribution algorithms for the computation of innovation estimators of diffusion processes

Zochil González Arenas[a,*], Juan Carlos Jimenez[b], Li-Vang Lozada-Chang[c], Roberto Santana[d]

[a] Department of Applied Mathematics, University of the State of Rio de Janeiro, Brazil
[b] Institute of Cybernetics, Mathematics and Physics, Havana, Cuba
[c] Department of Business Intelligence, Freepik Company, Malaga, Spain
[d] Department of Computer Science and Artificial Intelligence, University of the Basque Country, Spain

Received 28 May 2020; received in revised form 4 January 2021; accepted 12 March 2021
Available online 24 March 2021

Abstract

Innovation Method is a recognized method for the estimation of parameters in diffusion processes. It is well known that the quality of the Innovation Estimator strongly depends on an adequate selection of the initial value for the parameters when a local optimization algorithm is used in its computation. Alternatively, in this paper, we use a strategy based on a modern method for solving global optimization problems, Estimation of Distribution Algorithms (EDAs). We study the feasibility of a specific EDA - a continuous version of the Univariate Marginal Distribution Algorithm (UMDAc) - for the computation of the Innovation Estimators. Through numerical simulations, we show that the considered global optimization algorithms substantially improves the effectiveness of the Innovation Estimators for different types of diffusion processes with complex nonlinear and stochastic dynamics.
© 2021 International Association for Mathematics and Computers in Simulation (IMACS). Published by Elsevier B.V. All rights reserved.

Keywords: Innovation Estimators; Estimation of Distribution Algorithms; Parameter estimation; Diffusion process; Numerical simulations

1. Introduction

Diffusion processes defined through Stochastic Differential Equations (SDEs) have became an important mathematical tool for describing the dynamics of several phenomena, e.g., the dynamics of assets prices in the market, the fire of neurons, etc. In many applications, the statistical inference of diffusion processes is of great importance for model building and model selection. Typically, this inference problem consists in the estimation of unknown parameters of the diffusion process given a set of discrete observations. For systems in which all variables of the diffusion are observed without noise, a variety of approximate Maximum Likelihood, Bayesian, M and Z estimators for the parameters have been developed (for a review see, e.g., [19] and references therein). Contrarily, when only noisy observations of some components of the diffusion processes are accessible, just a few

* Corresponding author.
E-mail addresses: zochil@ime.uerj.br (Z.G. Arenas), jcarlos@icimaf.cu (J.C. Jimenez), llozada@freepik.com (L.-V. Lozada-Chang), roberto.santana@ehu.es (R. Santana).

https://doi.org/10.1016/j.matcom.2021.03.017
0378-4754/© 2021 International Association for Mathematics and Computers in Simulation (IMACS). Published by Elsevier B.V. All rights reserved.

estimators are available. They are the maximum likelihood method considered in [47] for linear SDEs with additive noise and the *M*-methods: Prediction Error Recursions [50], the Prediction-Based Estimating Functions [38] and the Innovation method [40]. Among these, the Innovation Estimators stand out for the diversity and number of successful applications (see, e.g. [19,24,56] and references therein).

Like any *M* estimator, the computation of the Innovation Estimators involves the minimization of an objective or fitness function, which is a non-quadratic function of the parameters in most of situations. By using a deep empirical know-how on the possible parameter values of a model for a particular application, this optimization problem has been regularly and successfully solved by means of local optimization algorithms (see, e.g., [41,44,55]). In these works, the known Gaussianity of the fitting-innovation process [22,40,41] is used to validate the results provided by the optimization algorithm. However, since the performance of the innovation estimators computed by local optimization algorithms strongly depends on the quality of the parameter's initial value, a great expertise of the users is needed to obtain satisfactory estimates.

To diminish the effect of a bad selection of initial parameter values and for assisting less experienced users, global optimization methods – such as the Estimation of Distribution Algorithms (EDAs) – arise as a viable alternative to the local optimization methods [26,27,31,36]. In particular, EDAs comprise a group of stochastic optimization heuristics that base the search of an optimal solution on a population of individuals. In the population, each one of the individuals represents a solution to the considered optimization problem. Individuals are evaluated and a subset of them, comprising the solutions with the best objective function values (lowest values for minimization, highest values for maximization), is selected. Further details on the selection method used in this paper are presented in Section 3. EDAs use probabilistic models to represent characteristic patterns of the most promising solutions. In each generation, a probabilistic model is learned from the set of selected solutions and used to sample new solutions. In this paper, we use a product of univariate marginal Gaussian densities to model the distribution of the selected solutions. From these selected solutions, the parameters of the univariate models are learned. To generate a new solution, each variable is sampled from the corresponding univariate Gaussian distribution. In this way, the algorithm evolves in successive generations towards the more promising regions of the search space until a stopping criterion is satisfied. EDAs can be classified according to the type of solution representation and to the way that the learning of the probability model is accomplished. The representations are discrete or continuous. Regarding the way of learning, EDAs can be classified in two classes. One class groups the algorithms that make a parametric learning of the probabilities and the other one comprises those algorithms that perform both, parametric and structural learning of the model.

In this paper, for the computation of the Innovation Estimators of the unknown parameters of diffusion processes, we focus on a class of EDAs with continuous representation and parametric learning. As probabilistic model, we use the univariate marginals estimated from the selected population, which defines the so called Univariate Marginal Distribution Algorithm in continuous domain (UMDAc) [32]. This global optimization strategy is also combined with a local optimization algorithm and both are tested in numerical simulations.

The paper is organized as follows. In Section 2, the estimation problem is clearly defined, and the essentials on the Innovation Method and EDAs are briefly presented. Section 3 focused on the application of the UMDAc to the parameter estimation of SDEs and the resulting algorithms are summarized. In Section 4, the potential of the proposed algorithms is illustrated in the parameter estimation of two types of diffusion processes with complex nonlinear and stochastic dynamics. Their performance is also compared with two optimization algorithms frequently used in practice. Finally, we present the conclusions of the paper and discuss some possible lines of future work.

2. Notations and preliminaries

Let the continuous–discrete state space model be defined by the continuous state equation

$$dx(t) = f(t, x(t); \alpha)dt + \sum_{i=1}^{m} g_i(t, x(t); \alpha)dw^i(t), \text{ for } t \geq t_0 \in \mathbb{R}, \tag{1}$$

and the discrete observation equation

$$z_{t_k} = h_0(t_k, x(t_k)) + \sum_{i=1}^{s} h_i(t_k, x(t_k))\xi_{t_k}^i + e_{t_k}, \quad \text{for } k = 0, 1, \ldots, N \in \mathbb{N}, \tag{2}$$

where $x(t) \in \mathbb{R}^d$ is the state vector at the instant of time t, $z_{t_k} \in \mathbb{R}^r$ is the observation vector at the instant of time t_k, α is a set of p parameters defined on $\Omega \subset \mathbb{R}^p$, w is an m-dimensional standard Wiener process, $\{\boldsymbol{\xi}_{t_k} : \boldsymbol{\xi}_{t_k} \sim \mathcal{N}(0, \boldsymbol{\Lambda}), \boldsymbol{\Lambda} = diag(\lambda_1, \ldots, \lambda_s), k = 0, \ldots, N\}$ and $\{e_{t_k} : e_{t_k} \sim \mathcal{N}(0, \Sigma), k = 0, \ldots, N\}$ are sequences of i.i.d. random vectors independent of w, and f, g_i, h_i are known vector functions. Regularity conditions for the unique identifiability of the state space model (1)–(2) are assumed, which are typically satisfied by stationary and ergodic diffusion processes (see, e.g., [3,29]). Here, the time discretization $(t)_N = \{t_k : k = 0, 1, \ldots, N\}$ is assumed to be increasing, i.e., $t_{k-1} < t_k$ for all $k = 1, \ldots, N$.

Let $x_{t|t_k} = \mathbf{E}(x(t)|Z_{t_k}; \alpha)$ and $P_{t|t_k} = \mathbf{E}((x(t) - x_{t|t_k})(x(t) - x_{t|t_k})^{\intercal}|Z_{t_k}; \alpha)$ for all $t \geq t_k$, where $\mathbf{E}(.|.)$ denotes conditional expectation and $Z_{t_k} = \{z_{t_j} : t_j \leq t_k \text{ and } t_j \in (t)_N\}$ is a sequence of observations. In the case that $t > t_k$, $x_{t|t_k}$ and $P_{t|t_k}$ are called prediction and prediction variance, respectively. If $t = t_k$, $x_{t_k|t_k}$ and $P_{t_k|t_k}$ are called filter mean and filter variance.

With α, $x_{t_0|t_0}$, $P_{t_0|t_0}$ given and for all $k = 1, \ldots, N$, the estimation of the filters $x_{t_k|t_k}$ and $P_{t_k|t_k}$ of the model (1)–(2) from the observations z_{t_k} is known as the nonlinear continuous–discrete filtering problem.

It is worth to note that, in many practical situations, the parameters α are usually unknown. Therefore, in these applications, the estimation of these parameters plays a central role. To be precise, this inference problem – that is the focus of this paper – consists in the estimation of the unknown parameters α of the model (1)–(2), given a time series Z_{t_N} with N observations z_{t_k}.

2.1. Innovation estimators

Given a time series Z_{t_N} with N observations z_{t_k}, the Innovation Estimator $\widehat{\alpha}$ of the parameters α in the model (1)–(2) is defined as in [22]

$$\widehat{\alpha} = \arg\min_{\alpha} \{q(\alpha)\}, \tag{3}$$

where

$$q(\alpha) = \left\{ N \ln(2\pi) + \sum_{k=1}^{N} \ln(\det(\Sigma^{\boldsymbol{\nu}}_{t_k|t_{k-1}})) + \nu_{t_k}^{\top} (\Sigma^{\boldsymbol{\nu}}_{t_k|t_{k-1}})^{-1} \nu_{t_k} \right\}. \tag{4}$$

Here, $\nu_{t_k} = z_{t_k} - \mathbf{E}(h_0(t_k, x(t_k))|Z_{t_k}; \alpha)$ and $\Sigma^{\boldsymbol{\nu}}_{t_k|t_{k-1}}$ are, respectively, the discrete time innovation process and its variance.

The estimator (3) is obtained by maximizing the likelihood of the discrete time innovation process $\{\nu_{t_k}\}$ taking into account that $\nu_{t_1}, \ldots, \nu_{t_N}$ are approximately Gaussian and independent random vectors (see [19,22,40] for details). The innovation estimator defined in this way belongs to the class of the M-estimators or prediction-error estimators depending on the inferential considerations that want to be emphasized [19]. In the case of linear state and observation equations with additive noise [19], the innovation estimator (3) reduces to the maximum likelihood estimator, which is – in general – defined in terms of the transition density function between two consecutive observations of the model (1)–(2).

Since the exact computation of ν_{t_k} and $\Sigma_{t_k|t_{k-1}}$ in (4) is only possible for a few particular models, in general, it is necessary to use approximate formulas. Likewise in [22], for a given value of α and starting with initial filter mean $y_{t_0|t_0} = x_{t_0|t_0}$ and filter variance $Q_{t_0|t_0} = P_{t_0|t_0}$, the approximations $\widetilde{\nu}_{t_k}$ and $\widetilde{\Sigma}^{\boldsymbol{\nu}}_{t_k|t_{k-1}}$ are recursively computed in the second step of the following Local Linearization Filtering (LLF) algorithm [21] for the model (1)–(2). For each $k = 0, \ldots, N-1$, compute

1. Prediction

$$y_{t_{k+1}|t_k} = y_{t_k|t_k} + \int_0^{t_{k+1}-t_k} \left[A_k y_{t_k+t|t_k} + a_k(t)\right] dt$$

$$Q_{t_{k+1}|t_k} = Q_{t_k|t_k} + \int_0^{t_{k+1}-t_k} \{A_k Q_{t_k+t|t_k} + Q_{t_k+t|t_k} A_k^{\intercal} + \sum_{i=1}^{m} B_{i,k} \left(Q_{t_k+t|t_k} + y_{t_k+t|t_k} y_{t_k+t|t_k}^{\intercal}\right) B_{i,k}^{\intercal}$$

$$+ \sum_{i=1}^{m} B_{i,k} y_{t_k+t|t_k} b_{i,k}^{\mathsf{T}}(t) + b_{i,k}(t) y_{t_k+t|t_k}^{\mathsf{T}} B_{i,k}^{\mathsf{T}}$$
$$+ \sum_{i=1}^{m} b_{i,k}(t) b_{i,k}^{\mathsf{T}}(t)\} dt,$$

2. Innovation

$$\widetilde{\nu}_{t_{k+1}} = z_{t_{k+1}} - h_0(t_{k+1}, y_{t_{k+1}|t_k}),$$

$$\widetilde{\Sigma}^{\nu}_{t_{k+1}|t_k} = C_{k+1} Q_{t_{k+1}|t_k} C_{k+1}^{\top} + \Sigma + \sum_{i=1}^{s} \lambda_i D_{i,k+1} \left(Q_{t_{k+1}|t_k} + y_{t_{k+1}|t_k} y_{t_{k+1}|t_k}^{\mathsf{T}} \right) D_{i,k+1}^{\mathsf{T}}$$
$$+ \sum_{i=1}^{s} \lambda_i \left(D_{i,k+1} y_{t_{k+1}|t_k} e_{i,k+1}^{\mathsf{T}} + e_{i,k+1} y_{t_{k+1}|t_k}^{\mathsf{T}} D_{i,k+1}^{\mathsf{T}} + e_{i,k+1} e_{i,k+1}^{\mathsf{T}} \right),$$

3. Filter

$$y_{t_{k+1}|t_{k+1}} = y_{t_{k+1}|t_k} + K_{t_{k+1}} \widetilde{\nu}_{t_{k+1}},$$
$$Q_{t_{k+1}|t_{k+1}} = Q_{t_{k+1}|t_k} - K_{t_{k+1}} C_k Q_{t_{k+1}|t_k},$$

where $K_{t_{k+1}} = Q_{t_{k+1}|t_k} C_k^{\mathsf{T}} \left(\widetilde{\Sigma}^{\nu}_{t_{k+1}} \right)^{-1}$ is the filter gain.

In this algorithm the remaining notations are

$$A_k = J_f(t_k, y_{t_k|t_k}), \qquad B_{i,k} = J_{g_i}(t_k, y_{t_k|t_k}), \quad C_k = J_{h_0}(t_k, y_{t_k|t_{k-1}}), \quad D_{i,k} = J_{h_i}(t_k, y_{t_k|t_k-1})$$

$$a_k(t) = f(t_k, y_{t_k|t_k}) - J_f(t_k, y_{t_k|t_k}) y_{t_k|t_k} + \frac{\partial f(t_k, y_{t_k|t_k})}{\partial s}(t - t_k),$$

$$b_{i,k}(t) = g_i(t_k, y_{t_k|t_k}) - J_{g_i}(t_k, y_{t_k|t_k}) y_{t_k|t_k} + \frac{\partial g_i(t_k, y_{t_k|t_k})}{\partial s}(t - t_k),$$

and

$$e_{i,k+1} = h_i(t_{k+1}, y_{t_{k+1}|t_k}) - D_{i,k+1} y_{t_{k+1}|t_k},$$

where J_v denotes the Jacobian matrix of the vector function v.

For simplicity, in the formal structure of the LLF algorithm given above, the predictions $y_{t_{k+1}|t_k}$ and $Q_{t_{k+1}|t_k}$ are implicitly defined in terms of a system of linear integral equations. Since $y_{t_{k+1}|t_k}$ and $Q_{t_{k+1}|t_k}$ in the LLF are the mean and variance of a bilinear SDE, simplified explicit expression for the exact solution of these integral equations are used in the numerical implementation of the LLF algorithm, which can be found in [16].

Thus, instead of the innovation estimator (3), in practice we use the approximate innovation estimator

$$\widehat{\alpha} = \arg\min_{\alpha} \{\widetilde{q}(\alpha)\}, \tag{5}$$

with

$$\widetilde{q}(\alpha) = \left\{ N \ln(2\pi) + \sum_{k=1}^{N} \ln(\det(\widetilde{\Sigma}^{\boldsymbol{\nu}}_{t_k|t_{k-1}})) + \widetilde{\nu}_{t_k}^{\top} (\widetilde{\Sigma}^{\boldsymbol{\nu}}_{t_k|t_{k-1}})^{-1} \widetilde{\nu}_{t_k} \right\}, \tag{6}$$

where $\widetilde{\nu}_{t_k}$ and $\widetilde{\Sigma}^{\boldsymbol{\nu}}_{t_k|t_{k-1}}$ are computed by means of the LLF algorithm, for all $k = 1, \ldots, N$.

The estimator (5) has been effectively used for the parameter estimation, from actual data, of a variety of neurophysiological, financial and molecular models, among others. Examples are the state space models defined by: (1) the 8-dimensional nonlinear SDEs of the Zetterberg's neural mass model for the alpha-rhythm generators in the human brain, plus a linear observation equation that results of modeling the amplifier of the EEG recording equipment by a cascade of a (second-order) low pass filter and a (first-order) high pass filter [55]; (2) the 15-dimensional nonlinear SDEs defined by an electro-vascular coupling model that explains the dynamics of electrical and vascular

states within a cortical unit in the brain, plus two nonlinear observation equations associated to the data recorded by the EGG and FMRI devices [44]; (3) the system of nonlinear SDEs derived from the multi-factor HJM model for the interest rate dynamics plus a linear observation equation associated to the noisy data of the markets [8]; and (4) the low dimensional SDEs used to approximate various statistical properties associated with steered molecular dynamics simulations of ion transport across a channel protein, and an observation equation for considering the uncertainty attributable to various noisy sources [7].

The success of the approximate innovation estimator (5) in the abovementioned applications have been possible thanks to two main factors: (1) the time distance between the data has been short enough to ensure a suitable approximation $\widetilde{\nu}_{t_{k+1}}$ to the innovation process $\nu_{t_{k+1}}$ provided by the Local Linearization filter; and (2) suitable parameter's initial values have been available for the local optimization algorithms. Recently, based on a new type of approximate minimum variance filters [17], a new type of approximate innovation estimators was introduced [18] to deal with the identification of diffusion processes from a reduced number of observations distant in time. Convergence and asymptotic properties of the new approximate innovation estimators were provided independently of the time distance between observations and practical adaptive algorithms for reducing the estimation bias were also given. For the case that suitable information about parameter's initial values is not available, in this paper, a global optimization algorithm will be proposed – for the first time – as an alternative to the local optimization algorithms for solving the optimization problem (5).

Clearly, the minimization of the function $\widetilde{q}(\alpha)$ in (5) with respect to α remains now as the major difficulty in the computation of the innovation estimators. Therefore, the quality of the estimation strongly depends on the performance of the optimization algorithm to be used. Among the difficulties of this optimization problem are the non-quadratic dependence of the fitness or objective function $\widetilde{q}(\alpha)$ with respect to the parameters α due to the nonlinearity of the model (1)–(2) and the presence of parameters α in the highly nonlinear term corresponding to the innovation variance Σ^{ν}. An additional difficulty is the impossibility of using local optimization methods of high convergence order since calculating the gradient or the Hessian of the objective function with regard to the parameters is usually difficult and sometimes impossible for most of nonlinear inference problems.

2.2. Univariate marginal distribution algorithm

Evolutionary algorithms (EAs) have been extensively applied to optimization problems with non-differentiable and highly complex functions. However, the most effective EAs require the definition of heuristic recombination operators that assumes some knowledge of the optimization problem structure [26]. By replacing the EA step of solution recombination by learning a distribution of the best solutions and sampling from it, EDAs do not depend on the definition of ad-hod recombination operators and they can automatically detect and exploit the patterns shared by the best solutions.

A pseudocode of a general EDA for solving optimization problems is shown in Algorithm 1.

In particular, the Univariate Marginal Distribution Algorithm (UMDA) [32] follows the general scheme in Algorithm 1, but using univariate marginals calculated from the selected population as the probabilistic model computed in the step 5 of this Algorithm 1. UMDA variants are particularly easy to implement since they rely on the computation of univariate distributions which are easier to learn and sample. Theoretical studies, as well as a variety of applications of UMDA have been reported in [14,30,36,46]. In addition, their solid theoretical foundations have permitted the study of convergence properties, an area that has recently attracted more attention [12,57].

On the other hand, applications of EDAs to problems with continuous representation comprise the use of UMDA based on Gaussian models [2,25,48], Gaussian networks [2,15,25], Marginal Histogram in Continuous Domains [52], mixtures of Gaussian distributions [5], and Voronoi based EDAs [39]. UMDA was originally introduced for discrete binary problems and later extended to deal with problems with a continuous representation. While the two variants use a univariate factorization of the distribution, there are important differences between them due to the probabilistic models they use. In its general form, the UMDA for continuous domains (UMDAc) [25] is not restricted to the use of Gaussian distributions, the density function which better fits the optimal solutions is statistically determined for each generation. For more details on EDAs for continuous domains, Refs. [4,6,43] can be consulted.

Table 1
Pseudocode of a general Estimation of Distribution Algorithm (EDA).

Algorithm 1: **EDA**

1 Set $l \Leftarrow 1$. Generate M points randomly from the search space (Initial population D_0).
2 **do** {
3 Evaluate the fitness function at the M points.
4 Select a subset D_{l-1}^{Se} of points according to a selection method.
5 Compute a probabilistic model from D_{l-1}^{Se}.
6 Sample M new points (population D_l) according to the probability distribution previously learnt.
7 $l \Leftarrow l + 1$
8 } **until** Termination criteria are met

3. UMDAc-based innovation estimators for diffusions

With the purpose of solving the optimization problem (5) using EDAs, let us consider the parameters $\alpha = (\alpha_1, \ldots, \alpha_p)$ in (1) as a random vector taking values on the *search space* $\Omega \subset \mathbb{R}^p$. $F_{\alpha_i}(\cdot\,; \theta_i)$ and $F_{\alpha}(\cdot\,; \theta)$ will denote, respectively, the density function of α_i and the joint density function of α depending on the sets of parameters θ_i and $\theta = \{\theta_1, \ldots, \theta_p\}$. In addition, D_l and D_l^{Se} will denote the population at the lth generation and the selected population at the lth generation from which the joint probability distribution of α is learnt.

In the implementation of the UMDAc carried out for this paper, we use a truncated Gaussian distribution for each component α_i at each generation l. In addition, by considering independence among the components of α in the model (1)–(2), we have assumed that the joint density function can be factorized as the product of univariate marginal densities of each component α_i. UMDA has been extensively applied to problems where there is at least some degree of interactions between the variables. For these problems, it can lead to optimal or satisfactory sub-optimal solutions [1,30]. On the other hand, for some particular class of problems with strong interactions which make harder the convergence of EDAs, e.g., with deceptive functions [11], the UMDA behavior will suffer because of the univariate assumption. In any case, our empirical results indicate that the optimization problem we address is not deceptive.

In our optimization problem (5), we consider that each component α_i takes values on the bounded interval $[a_i, b_i]$. We use a multivariate truncated Gaussian distribution for the joint density function of the continuous p-dimensional variable α. Univariate marginal densities for each α_i are computed through

$$F_{\alpha_i}(\alpha; \mu_i, \sigma_i) = \begin{cases} K(\mu_i, \sigma_i)\, e^{-\frac{(\alpha_i - \mu_i)^2}{2\sigma_i^2}} & \text{if } \alpha_i \in [a_i, b_i] \text{ for all } i = 1, \ldots, p \\ 0 & \text{in any other case,} \end{cases} \tag{7}$$

where $K(\mu_i, \sigma_i)$ is the normalization constant of the probability model.

The pseudocode for learning the joint density function by the UMDAc at each generation l, regarding the step 5 in Algorithm 1, is shown in Algorithm 2.

With the specifications above, the point $\hat{\alpha}$ that minimizes a fitness function on Ω is estimated as follows. First, an initial population D_0 of M points α on Ω is generated according to a uniform distribution. In each generation l ($l > 1$), by following the step 4 in Algorithm 1, a subset D_{l-1}^{Se} of L points of D_{l-1} is selected according to certain selection criteria, with $L \leq M$. In order to have a quicker convergence to the optimum, we used *truncation selection* as the selection method of choice [34]. This method selects a given proportion τ of the population solutions with the better values of the fitness function for learning the probabilistic model. Hence, for the selected $L = \tau M$ points, the probabilistic model (7) is estimated with the sampling estimates

$$\widehat{\mu_i} = \frac{1}{L} \sum_{\alpha \in D_{l-1}^{Se}} \alpha_i \quad \text{and} \quad \widehat{\sigma_i} = \sqrt{\frac{1}{L} \sum_{\alpha \in D_{l-1}^{Se}} (\alpha_i - \widehat{\mu_i})^2}$$

Table 2
Calculation of the probabilistic model by the Univariate Marginal Distribution Algorithm for continuous domains.

Algorithm 2: **Probabilistic model: UMDAc with truncated Gaussian distribution**

Let $F_{\alpha_i}(\alpha;\theta_i)$ be the univariate marginal densities given by expression (7).

1 **for** $i := 1$ **to** p {
2 *Obtain* an estimator $\widehat{\theta_i}$ for $\theta_i = (\mu_i, \sigma_i)$
}

The learnt joint density function is expressed as $F_\alpha\left(\cdot\,;\widehat{\theta}^l\right) = \prod_{i=1}^{p} F_{\alpha_i}\left(\cdot\,;\widehat{\theta}_i^l\right)$

of μ_i and σ_i, and so, the joint density function is expressed as

$$F_\alpha\left(\cdot\,;\widehat{\mu}^l,\widehat{\sigma}^l\right) = \prod_{i=1}^{p} F_{\alpha_i}\left(\cdot\,;\widehat{\mu}_i^l,\widehat{\sigma}_i^l\right). \tag{8}$$

It is worth to note that, although $\widehat{\mu_i}$ and $\widehat{\sigma_i}$ are not classic estimators of μ_i and σ_i for the truncated Gaussian distribution, they work well whenever the parameter σ_i is small relative to the length of the supporting interval, $b_i - a_i$. Typically, σ_i tends to zero. Therefore, the effect of truncation is negligible and the estimation can be done by using the maximum likelihood estimator of the parent normal distribution (see Ref. [23] for details).

Now, points for the new population D_l must be sampled by using the learnt probability distribution given by expression (8). At this step, we first use *elitism*, a technique for preserving the best solutions of a population to the new one. In this way, the best solutions found along the algorithm evolution are kept until the final generation. So, elitism works by copying to the new population D_l the ϵ points of the previous population D_{l-1} with the best solutions (minimum values of the fitness function). The remainder new $(M - \varepsilon)$ points for D_l are randomly generated from the multivariate truncated Gaussian distribution defined by the marginal densities (7) with parameters $\widehat{\mu_i}$ and $\widehat{\sigma_i}$ estimated from D_{l-1}^{Se}. Notice that truncation selection and elitism have different effects in the algorithm. Selected solutions are only used to learn the probabilistic distribution, whereas elitist solutions are copied to the next population. In our implementation of the algorithm, elitist solutions are a subset of the selected solutions. This process of generating new populations is repeated until a stopping condition is met. Some stopping conditions as a fixed number of generations or non significant improvement in the minimization of the fitness function after a given number of generations can be considered. At this stage, $\hat{\alpha}$ is set as the point in the last population with the lowest fitness function value. When the optimization algorithm stops, the Gaussianity that should have the fitting-innovation process when the model fits well the data [22,40,41] is then checked by the Kolmogorov–Smirnov test of normality, which is the more important verification required for evaluating the fitness of the model to the data with the estimated parameter $\hat{\alpha}$.

Regarding the algorithmic complexity of the UMDAc, various issues must be considered. Firstly, the UMDAc initialization step, which consists in assigning the values to all the individuals in the initial population. It has complexity $O(pM)$ where p is the number of variables and M the population size. The computational complexity of the evaluation step is problem dependent. Let $cost_f$ be the running time associated to the evaluation of function f, the running time complexity of this step is $O(M\ cost_f)$. The complexity of truncation selection is related to sorting the solutions according to the fitness. In the worst case, the complexity of this step is $O(Mlog(M))$. The complexity of the learning step is related to the estimation of the parameters of the univariate distributions which is $O(pM)$. The cost of the sampling method depends on the complexity of the algorithm used to sample from a Gaussian distribution. Different algorithms exist for this purpose [51]. If we assume the cost of sampling a value from the Gaussian distribution to be $O(1)$ then the computational complexity of sampling is $O(pM)$, and so the computational complexity of UMDAc can be estimated as $O((M\ costf + pM + Mlog(M)) * (g-1) + M\,costf)$ where g is the number of generations.

In addition, we consider the strategy of combining the global and local optimization techniques in which the estimated values of the parameters obtained by EDA are used as initial values of the local technique. The application

Table 3
Algorithm for global optimization.

Algorithm 3: **UMDAc — Innovation estimator**

Set $l \Leftarrow 1$ and generate M points α using a uniform distribution. With $D_0 = \{\alpha^1, ..., \alpha^M\}$ (Initial population for the parameter α),
do {
  $D \Leftarrow D_{l-1}$
  Evaluate the fitness function $\widetilde{q}(\alpha)$, defined in (6), at the M points α in D by means of the LLF algorithm
  Select a subset D_{l-1}^{Se} of points of D according to the selection truncation method
  Estimate the joint density function F_α from D_{l-1}^{Se} applying Algorithm 2
  Consider elitism for the new population of α: keeping ε points α of D with the lowest values of $\widetilde{q}(\alpha)$
  Sample $M - \varepsilon$ new points α for D according to the joint density function F_α estimated in the step 6
  $D_l \Leftarrow D$, $l \Leftarrow l + 1$
} **until** Termination criteria are met

$\hat{\alpha}$ is the point α of the population D_l with the minimum value $\widetilde{q}(\alpha)$ in the last generation l

of EDAs together with local optimization techniques has been reported to notably improve the quality of the solutions for problems from different domains [35,42,58].

For comparison purposes, we will use the MATLAB function *fmincon* since it is the Local Optimization Algorithm (LOA) more frequently reported in applications of the innovation estimators. This function searches for the minimum of a nonlinear multivariate function with constraints and needs an initial parameter estimate for this task. Since an explicit expression for the gradient of the objective function (6) with respect to the parameters is not available, *fmincon* uses a sequential quadratic programming (SQP) method and performs a line search using a merit function. A complete description of this can be found in the online Matlab Documentation of the Optimization Toolbox. Moreover, we compare the results of the proposed optimization methods with other frequently employed strategy when there is not knowledge of the proper initial parameter values to use. This method, known as Random Search, simply starts looking for multiple local optima with a local optimization algorithm at multiple random points from the parameter space, let all those local algorithms run (if they are able to), and finally takes the minimum of all minimums successfully found as the global optimum.

In what follows, we summarize with a pseudocode the four algorithms considered in the paper for computing the approximate innovation estimator $\hat{\alpha}$ – defined in (5) – of the parameters α in the model (1)–(2), given a time series Z_{t_N} with N observations of the model.

To conclude this section, we emphasize that EDAs are global algorithms for optimizing cost functions of any type. In this section, since the proposed EDA is applied to maximize the likelihood function of the innovation process and given that the parameters α in (1) are assumed to be a random vector with *a priori* distribution F_α, the estimates $\hat{\alpha}$ provided by Algorithms 3 and 4 could be seen as Bayesian-type estimators. However, note also that in these algorithms, the maximum of the likelihood function is obtained by sampling parameter values from an *a priori* distribution which changes from the uniform distribution at the initial generation to a truncated Gaussian distribution for the following stages. Besides, the mean and variance of the truncated Gaussian distribution, obtained from the selected points in the previous population, also change from a generation to the next one.

4. Numerical experiments and results

The goal of our numerical experiments is to determine whether the use of UMDAc can improve the computation of Innovation Estimators for unknown parameters of discretely observed diffusion processes. To do so, we compare the Algorithms 3, 4, 5 and 6 described above in the search of the solution to the optimization problem specified in (5) for two test problems. It is known that, with suitable initial value, the local optimization algorithms outperform the global ones in accuracy and computational cost. Therefore, in addition to the typical quantitative comparison

Table 4
Global optimization with Local optimization refinement.

Algorithm 4: **Refined UMDAc — Innovation estimator**

1 Use *Algorithm* 3 to find an estimate $\tilde{\alpha}$ of α.
2 Set $\alpha_0 \Leftarrow \tilde{\alpha}$.
3 Compute a new estimation $\hat{\alpha}$ of α with the MATLAB function *fmincon* starting at α_0.

Table 5
Local optimization strategy (LOA).

Algorithm 5: **LOA — Innovation estimator**

1 Define α_0 (Initial value for α).
2 Compute an estimation $\hat{\alpha}$ of α with the MATLAB function *fmincon* starting at α_0.

Table 6
Global optimization with Random Search Method.

Algorithm 6: **Random Search — Innovation estimator**

1 Set $l \Leftarrow 1$ and generate M initial points α_0 using a uniform distribution. With $S_0 = \{\alpha_0^1, ..., \alpha_0^M\}$,
2 **do** {
3 Compute an estimation $\hat{\alpha}^l$ of α with the MATLAB function *fmincon* starting at α_0^l.
4 $l \Leftarrow l + 1$
5 } **until** $l = M$

$\hat{\alpha} = \arg\min_{\alpha \in S_l} \{q(\alpha)\}$, where $S_l = \{\hat{\alpha}^1, ..., \hat{\alpha}^M\}$

of accuracy versus computational cost between algorithms, we compare the algorithms in two extreme situations that clearly illustrate the advantage of the proposed global optimization algorithms over the local ones, when good initial values for the second ones are not available.

Since the result of the four algorithms to be compared strongly depends on random initial values, their performance cannot be measured just by a simple run, but by the results of many of them. Therefore, for both test problems, given a time series Z_{t_N} of N observations z_{t_k}, we carried out 100 estimations of the parameters α for each one of the four Algorithms and compare the corresponding histograms of results. The information contained in these histograms indicates us which of the compared Algorithm gives, in just one run, a good result with higher probability.

In each one of the 100 estimations, the initial value for the local Algorithm 5, and the initial population for the global Algorithms 3 4 and 6 were randomly selected from a predefined set of possible values for each parameter α_i. The population size M for Algorithms 3 4 and 6 was decided in agreement with the approach suggested in [34], i.e., M was set equal to 20 times the number of parameters p to be estimated. In order to use the truncation selection method for Algorithms 3 and 4, a truncation threshold τ must be specified. An empirical rule previously used in Factorized Distribution Algorithm (FDA) [33] locates it between 0.125 and 0.4. We fixed $\tau = 0.3$ considering also that this choice of τ has been reported in previous EDA applications in continuous domains [9]. On the other hand,

elitism was implemented with 5% of the population. This value, $\epsilon = 5\%$, is a good balance between the goal of keeping the best solutions for the next generation and the goal of having a sufficiently diverse population which is not dominated by a set of solutions very similar among each other. As stopping condition we used a combination of the most common criterion for evolutionary algorithms, which is a predefined maximum number of generations, and the more important verification required for the innovation estimators, as we explained before, in Section 3. We fixed 10 generations as its maximum number, in such a way that – with lowest computational cost – the approximate innovation process $\widetilde{\nu}_{t_k}$ corresponding to the optimal parameter value $\hat{\alpha}$ is approximately a white noise process. All these mentioned values were selected after conducting a set of preliminary experiments.

When the optimization algorithm stops, the known Gaussianity of the fitting-innovation process [22,40,41] is then used to validate the optimal estimated value $\hat{\alpha}$ of the parameters, which is the more important verification required for the innovation estimators.

In the first test model below, the realization of the state equation was computed by means of the order 1 strong Local Linearization scheme for SDEs (see, e.g., [20]) on a time discretization finer than the observation times $t_0,\ldots,$ t_N. Then, a subsample of the approximate solution of $x(t)$ at the time instants $t_0,\ldots, t_N$ was used for evaluating the observation equation and so to obtain a realization Z_{t_N} of the model. Similarly, the realization Z_{t_N} of the second test model below was constructed, but using the known Euler–Maruyama scheme instead of the Local Linearization one. Since, in general, the observations are distant in time, this simulation and subsampling procedure guarantees an accurate simulation of the diffusion process $x(t)$ and, thus, we can have a reliable realization of the state space model at the observation times. On the other hand, in both examples, the time distance between two consecutive observations was set short enough to ensure a suitable approximation $\widetilde{\nu}_{t_{k+1}}$ of the Local Linearization filter to the innovation process $\nu_{t_{k+1}}$, in such a way that the goodness of the estimated parameters depends only on the performance of the optimization algorithm for computing the expression (5).

4.1. Nonlinear state equation with additive noise

Let us consider the stochastic Fitzhugh–Nagumo model defined by the continuous nonlinear state equations

$$dx_1 = \alpha_1(x_1 - \frac{x_1^3}{3} + x_2)dt \tag{9}$$

$$dx_2 = -\frac{1}{\alpha_1}(x_1 - \alpha_2)dt + \alpha_3 dw_2 \tag{10}$$

and the discrete observation equation

$$z_{t_k} = x(t_k) + e_{t_k}\ , \tag{11}$$

where $(x_1(t), x_2(t)) \in \mathbb{R}^2$, $z_{t_k} \in \mathbb{R}^2$, $\alpha_1 = 1$, $\alpha_2 = 1$, $\alpha_3 = 0.1$, $x(0) = (x_1(0), x_2(0)) = (-0.9323, -0.6732)$ and $\{e_{t_k} : e_{t_k} \sim \mathcal{N}(0, 10^{-6}I_{2\times 2})\}$. The stochastic Fitzhugh–Nagumo equations (9)–(10) has been widely used in analytical and simulation studies of neuronal models (see, e,.g., [53,54]). Stability, random attractors, stationarity distribution, spike rates and other dynamical properties of such equations have been studied in a number of papers. See, for instance, [28,49], and references therein.

With $\alpha_3 = 0$, the Fitzhugh–Nagumo equations (9)–(10) have a single equilibrium point $(\alpha_2, \frac{\alpha_2^3}{3} - \alpha_2)$ and the sign of solutions x_1 and x_2 for all $t > 0$ is completely determined by the sign of the parameter α_2. The Jacobian of the linearized equations around this equilibrium point has eigenvalues

$$\lambda_{1,2} = \frac{\alpha_1(1-\alpha_2^2) \pm \sqrt{\alpha_1^2(1-\alpha_2^2)^2 - 4}}{2},$$

and so stability and dynamics of the solutions is well known (see, e.g., [45], pages 11 and 101). In particular, for the parameter values $\alpha_1 = 1$ and $\alpha_2 = 1$, a Hopf bifurcation arises and the solution is a limit circle. When $\alpha_3 \gtrsim 0$, the small additive noise introduced in the equations tends to destabilize the mentioned limit circle. In this situation, it is expected that the eigenvalues λ_1 and λ_2 corresponding to the estimated parameters be complex numbers with a positive real part close to zero. Hence, for the parameters $(\alpha_1, \alpha_2, \alpha_3)$ taking values in a neighborhood of the value (1,1,0.1) the stability and dynamics of the solutions might abruptly switches making hard the simulation and estimation of the model (9)–(11).

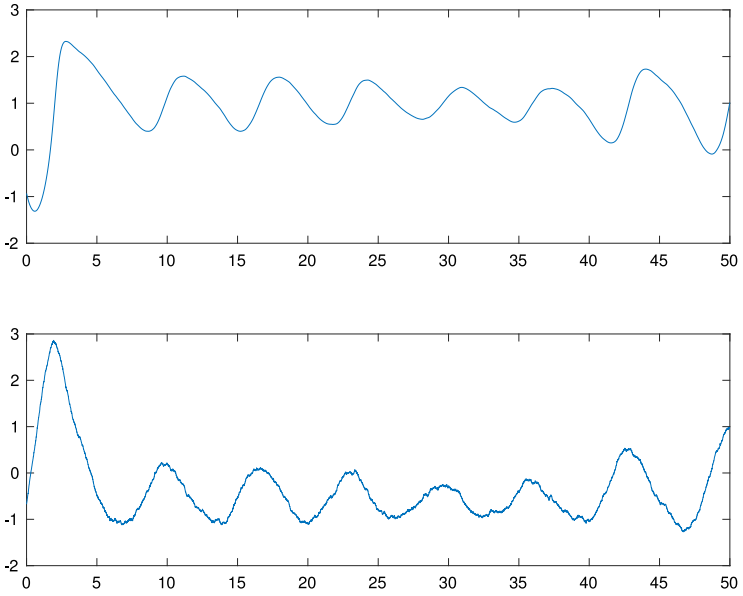

Fig. 1. Realization of the solution of the state equation (9)–(10). Top: $\{x_1(t_j) : j = 1, \ldots, T\}$. Bottom: $\{x_2(t_j) : j = 1, \ldots, T\}$.

A realization $\{x(t_j) = (x_1(t_j), x_2(t_j)) : j = 1, \ldots, T\}$ of the solution of the state equations (9)–(10) is shown in Fig. 1 for the time instants $t_j = jh$, with $h = 0.0005$ and $T = 50$.

Given the state space model (9)–(11) and the single realization $\{z_{t_k} : k = 0, \ldots, N\}$ of the model, with $t_k = k\Delta$, $\Delta = 0.5$ and $N = 500$, 100 estimates of the parameter set $\alpha = (\alpha_1, \alpha_2, \alpha_3)$ were carried out. Each initial estimate of α was uniformly generated inside of their definition intervals

$$\alpha_1 \in [-5, 5], \quad \alpha_2 \in [-5, 5], \quad \alpha_3 \in [0, 1]. \tag{12}$$

Table 7 lists the range of values, defined by the minimum and maximum values, of the parameters estimated by each one of the four considered Algorithms. Significantly, unlike for the estimated parameters by Algorithms 3, 4 and 6 the minimum value of α_1 and the maximum value of α_3 estimated by Algorithm 5 reach the permissible extreme values defined in (12) for these parameters.

This trouble with the estimation of Algorithm 5 is more clearly shown in Fig. 2, which displays the histogram of the estimators obtained by each one of the considered algorithms. In this figure, dot lines correspond to the true value of the parameters, whereas dash and dot lines correspond to the average of the estimations. Observe that, in the third column of the figure, approximately half of the estimations for the parameters α_1 and α_3 lies out of one of the established limits for these parameters, which reflects the failure of Algorithm 5. A more careful analysis of the results shows that only 49 of the 100 estimations made with Algorithm 5 yield a suitable estimation of the true parameter values, which corresponds to the 49 initial values with positive parameter α_1. In this case, all the estimated values of α_1 are positive, the estimated values of α_2 are close and lower than 1 and the estimated values of α_3 are close to 0.1. On the contrary, the 51 failed estimations made with Algorithm 5 results from the initial values with negative values of α_1. In this case, the 51 estimated values of α_1 remain negative and, as well as the 51 estimated values of α_3, reach one of the permissible extreme values defined in (12) for these two parameters, whereas the values of the estimated parameter α_2 are mainly concentrated around the values 1.90 and a few ones around 0. Note that, by looking for the parameters that better fit the model to the data, the sign of the observed noisy variables x_1 and x_2 forces Algorithm 5 to provide positive values of α_2. Moreover, the instability of the oscillations presented in the data forces Algorithm 5 to provide values of α_2 close to 1. However, nothing in the dynamics of the data pushes Algorithm 5 to change the sign of the parameter α_1, which basically explains its failure.

On the contrary, as it is shown in Table 7 and Fig. 2, Algorithm 3 provides satisfactory estimations of the parameters α. The 100 estimations of each parameter are inside of a reduced range of values and the histogram of the estimated values of each parameter are concentrated around a certain value close to the true parameter values.

Table 7
Range of values of the estimated parameters obtained by UMDAc (Alg 3), UMDAc + Local Strategy (Alg 4), Local Strategy (Alg 5), and Random-Search (Alg 6) for the state space model (9)–(11). The range of values is defined by the minimum and maximum values of the estimated parameters in 100 executions of Algorithms.

α	Real	Estimated (Alg 3)	Estimated (Alg 4)	Estimated (Alg 5)	Estimated (Alg 6)
α_1	1	[0.9958, 1.0034]	[0.99998509, 0.99998510]	[−5.0000 , 1.0000]	[0.99997, 0.99999]
α_2	1	[−0.4558, 2.1301]	[0.98495, 0.98496]	[−0.3763, 1.9648]	[0.9847, 0.9849]
α_3	0.1	[0.1319, 0.6941]	[0.1042630, 0.1042639]	[0.1043, 1.0000]	[0.104263, 0.104280]

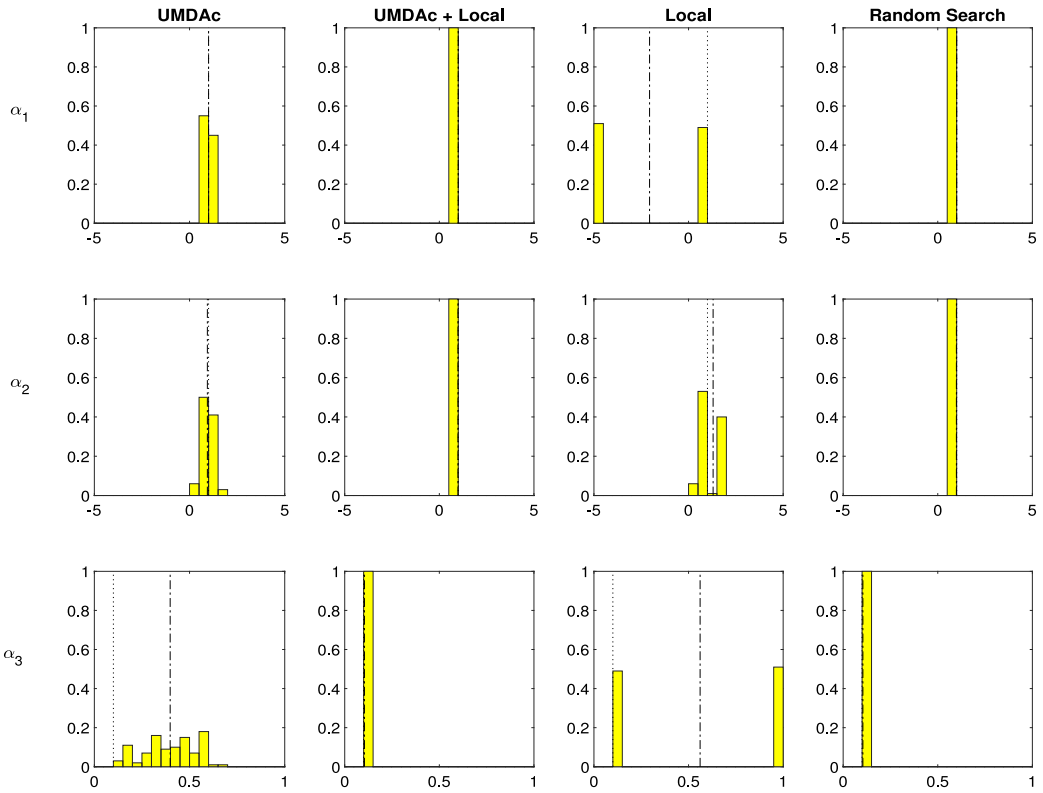

Fig. 2. Histograms of the estimated values of α in 100 estimations of the state space model (9)–(11) by the four estimation algorithms. Dot lines correspond to the true value of the parameters, whereas dash and dot lines correspond to the average of the estimations.

The key of this is the way in which EDAs evolve. Indeed, at steps 5 and 6 in Algorithm 3, a probabilistic model from a limited number of points (the best one according to the fitness function) is built. At steps 7 and 8, UMDAc preserves the best points (according to the fitness function) and generates – from the probabilistic model – new points for replacing the not selected ones. That is, in step 5 and 6, the points with the worst fitness function are not used to construct the probability model, making less likely the generation of new points close to those with the worst fitness function. In the steps 7 and 8, the points with worst fitness function and the unstable ones are replaced by others with larger probability to improve the value of the fitness function. In this way, after various iterations, the populations generated by EDAs typically only contain points close to the true parameter values.

These estimation results can be significantly improved by Algorithm 4 that use each output of Algorithm 3 as initial value of the parameters in the local optimization algorithm. Indeed, this is corroborated in the second column of Fig. 2 which shows the histograms of the estimators of α obtained by Algorithm 4. In addition, Table 7 shows that the 100 estimations are all grouped very close to the true value of the parameters, being all the estimated values of α_1 almost 1 and the estimated values of α_2 close and lower than 1, making the corresponding 100 pairs of eigenvalues λ_1 and λ_2 complex numbers with positive real part close to zero.

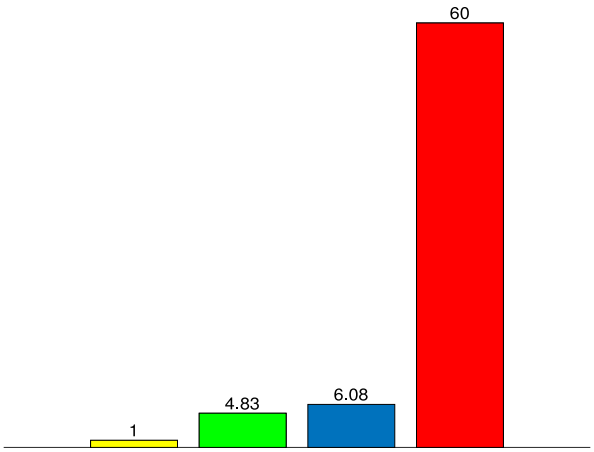

Fig. 3. Relative computational time of each optimization algorithm with respect to that of Algorithm 5 in the parameters estimation of the Fitzhugh–Nagumo equations (9)–(10). From left to right: the Local Strategy (Alg 5), UMDAc (Alg 3), the UMDAc + Local Strategy (Alg 4), and Random-Search (Alg 6).

As it is shown in Fig. 2 and Table 7, such accurate results in the estimation of the parameters of the Fitzhugh–Nagumo equations (9)–(10) are also obtained from the 100 estimations of Algorithm 6. These 100 accurate results are obtained after discarding the estimation of approximately 50% of the local algorithms that – as average – failed to reach a suitable minimum in each one of the 100 runs of Algorithm 6 and taking the global optimum as the minimum of all remaining minima successfully found in each one of the 100 runs. However, as it is shown in Fig. 3, this result is achieved with much higher computational cost. Specifically, Algorithm 6 is 10 times slower than Algorithm 4.

In Fig. 3, each bar presents the relative computational time of each optimization algorithm with respect to that of Algorithm 5. This time ratio works as a simple indicator to compare the total computational cost of each algorithm. Indeed, since each simple run of the local optimization algorithm inside of Algorithm 6 performs – as average – 35 iterations and 4 evaluations of the fitness function per iteration, each run of Algorithm 6 requires – as average – $60 \times 35 \times 4 = 8\,400$ fitness function evaluations to obtain a successful parameter estimation. On the other hand, each run of Algorithms 3 requires only $60 + 57 \times 9 = 573$ evaluations of the same fitness function in the 10 generations with 60 individuals each one (discounting 3 individuals that are preserved in each generation because of the 5% for elitism). Hence, each run of Algorithm 4 requires – as average – a total of $573+140 = 713$ evaluations of the fitness function, *i.e.*, the same 573 evaluations of the fitness function of Algorithms 3 plus $35 \times 4 = 140$ extra evaluations that – as average – is required by the local optimization algorithm to compute the final parameter estimate of Algorithm 4. As mentioned in Section 2.1, each evaluation of the fitness function (6) involves a run of a computationally costly filtering algorithm, which remarks the merit of the proposed Algorithm 4 for computing accurate innovation estimators of diffusion process with a reduced number of evaluations of the fitness function.

4.2. Nonlinear model with multiplicative noise

Let us consider the following nonlinear state space model with multiplicative noise (Example 1 in [21])

$$dx_1 = (\alpha_1 + \alpha_2 x_1)dt + \alpha_3\sqrt{x_1}dw_1 \tag{13}$$

$$dx_2 = \alpha_4 x_2^2 dt + \alpha_5 x_1^2 dw_2 \tag{14}$$

$$z_{t_k} = x_2(t_k) - 0.001x_2^3(t_k) + (x_2(t_k) - 0.01x_2^2(t_k))\xi_{t_k} + e_{t_k}, \tag{15}$$

where $(x_1(t), x_2(t)) \in \mathbb{R}^2$, $z_{t_k} \in \mathbb{R}$, $\alpha_1 = 1$, $\alpha_2 = -1.5$, $\alpha_3 = 0.1$, $\alpha_4 = -1$, $\alpha_5 = 0.01$, $x(0) = (x_1(0), x_2(0)) = (0.5, 0.5)$, w_1 and w_2 are independent standard Wiener processes, $\{\xi_{t_k} : \xi_{t_k} \sim \mathcal{N}(0, 0.01)\}$ and $\{e_{t_k} : e_{t_k} \sim \mathcal{N}(0, 0.01)\}$.

Concerning the estimation of the five parameters, this estimation problem is of greater complexity. The model to be estimated is almost over parameterized (in the sense that some eigenvalues of the Fisher matrix corresponding

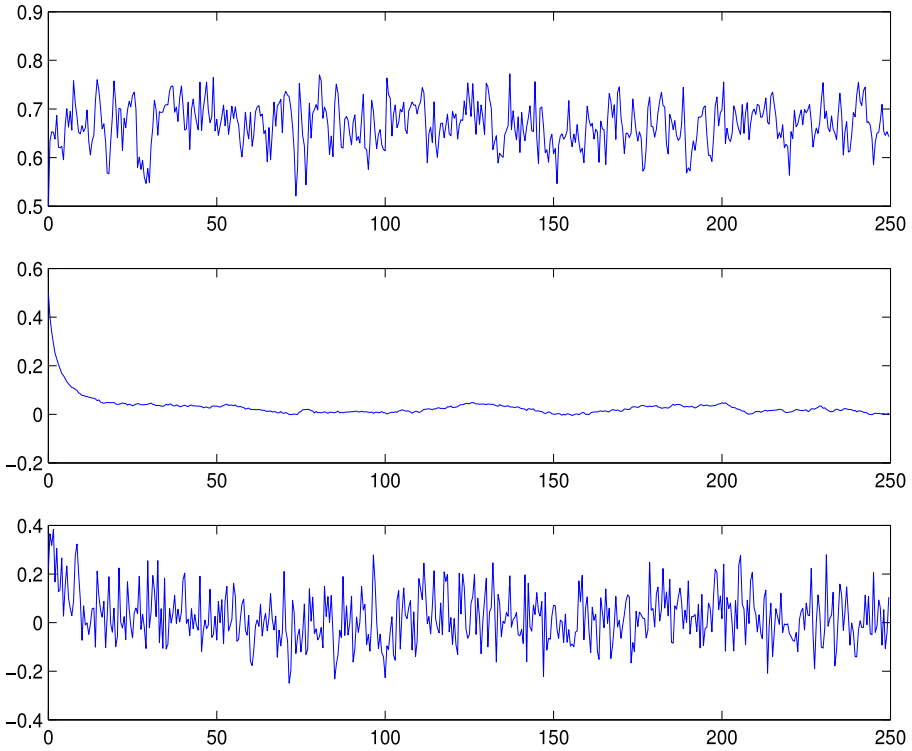

Fig. 4. Realization of the solution of the state space model (13)–(15) Top: $\{x_1(t_j) : j = 1, \ldots, T\}$, Center: $\{x_2(t_j) : j = 1, \ldots, T\}$, Bottom: $\{z_{t_k} : k = 0, \ldots, N\}$.

to the Innovation Estimator are close to zero for some parameter values) and the parameters were set in such a way that the state variables of the diffusion model are strongly dominated by the system noise and in order to have very low signal–noise ratio. Eq. (13) is the well-known Cox–Ingersoll–Ross (CIR) equation proposed in [10] for modeling interest rates in finance. The system of equations (13)–(14) is a particular instance of the general class of stochastic volatility models in finance [13].

In Fig. 4, a realization $\{x(t_j) = (x_1(t_j), x_2(t_j)) : j = 1, \ldots, T\}$ of the solution of the model (13)–(14) at instants of time $t_j = jh$, with $h = 0.005$ and $T = 5 \times 10^4$, is shown. A realization $\{z_{t_k} : k = 0, \ldots, N\}$ of the equation of observations (15) with $t_k = k\Delta$, $\Delta = 0.5$ and $N = 500$ is also shown. Observe in this figure the strong component of noise in the signal and in the observations.

Given the state space model (13)–(15) and a single realization of z_{t_k}, as that shown in Fig. 4, 100 estimates of the parameter set $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5)$ were carried out. Each initial estimate of α was uniformly generated inside of their definition intervals

$$\alpha_1 \in [0, 2], \quad \alpha_2 \in [-3, 0], \quad \alpha_3 \in [0, 0.3], \quad \alpha_4 \in [-3, 0], \quad \alpha_5 \in [0, 0.1].$$

Fig. 5 shows the histograms of the estimated values of α in 100 estimations of the state space model (13)–(15) by the four estimation algorithms. It is clear that estimation resulting from Algorithms 5 and 6 is not useful when considering a uniformly distributed initial value in the specified intervals for each parameter.

On the other hand, an interesting result in the estimation is obtained with the optimization Algorithms 3 and 4 via UMDAc, in the sense that all the results are located around certain value relatively close to the true parameter values. In this case, the estimation of Algorithm 4 does not improve the results of Algorithm 3, which confirms that the local optimization algorithm is not useful for this problem.

For a better explanation of the complexity of this optimization problem, the evaluation of the fitness function around the mean values of the estimated values of the parameters is shown in Fig. 6, by moving the values of only a parameter α_i. In general, the fitness function is almost flat, which is even more evident for the third coordinate. The

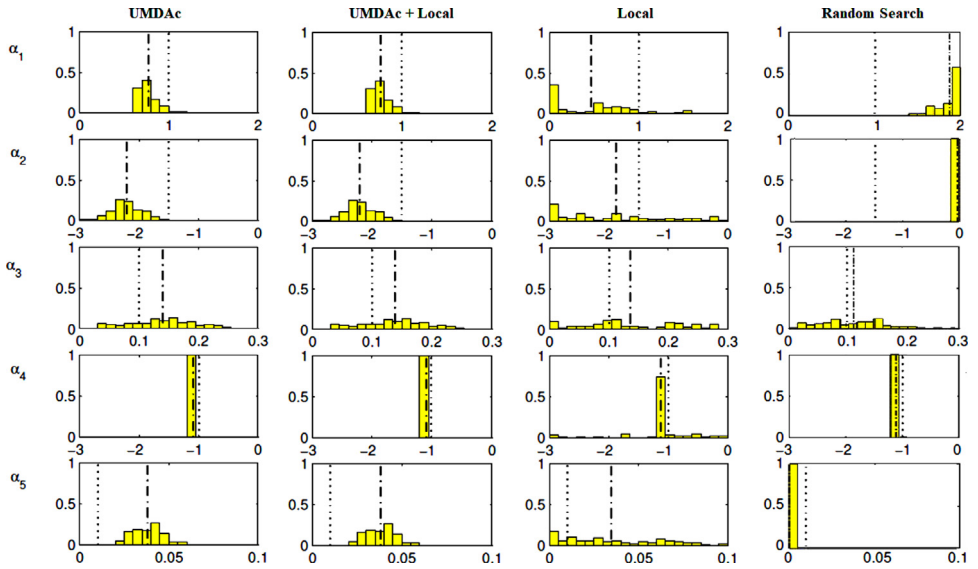

Fig. 5. Histograms of the estimated values of α in 100 estimations of the state space model (13)–(15) by the four estimation algorithms. Dot lines correspond to the true value of the parameters, whereas dash and dot lines correspond to the average of the estimations.

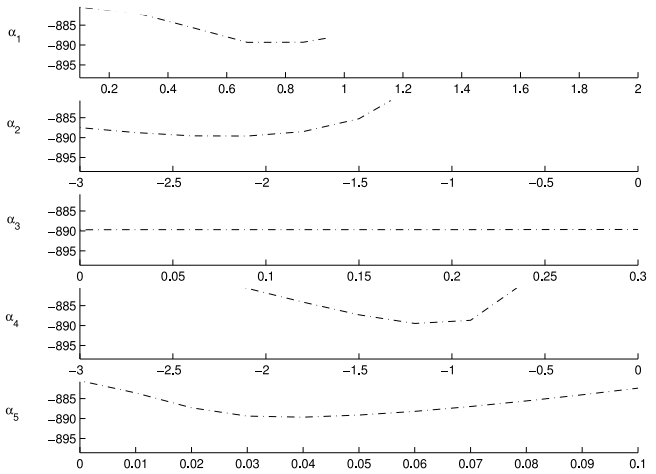

Fig. 6. Evaluation of the fitness function keeping fixed all the coordinates of α except α_i in the model (13)–(15).

existence of a local minimum for the third parameter could be observed making a bigger rescaling in the considered interval. However, note that in this extreme situation the average of the estimated parameters α_1, α_2, α_4, and α_5 are quite close to the minimum value of the fitness function for each parameter, which reveals the good performance of the Algorithms 3 and 4, and demonstrates the robustness of the UMDAc in the case of having small deviations from the typical assumptions required for its use.

Here, it is worth to point out the following. It is known [37,40] that the "exact" innovation estimator (3) is unbiased, i.e., the difference between the parameter value α and the expected value of $\hat{\alpha}$ with respect to the measure on the underlying probability space generating the realizations of the model (1)–(2) tends to zero as the number N of observations z_{t_k} goes to infinite. Therefore, for a model like the one considered in this example (strongly dominated by the system noise and with very low signal–noise ratio) and a short time series of observation distant in time, we cannot expect that the value $\hat{\alpha}$ provided by the estimator (3) - or by its approximation (5) - be close to the parameter value α. Ways to make closer the approximate estimator (5) to α is another important matter, but it is beyond the scope of this paper. What it is really important in this paper is that, for a given realization of a

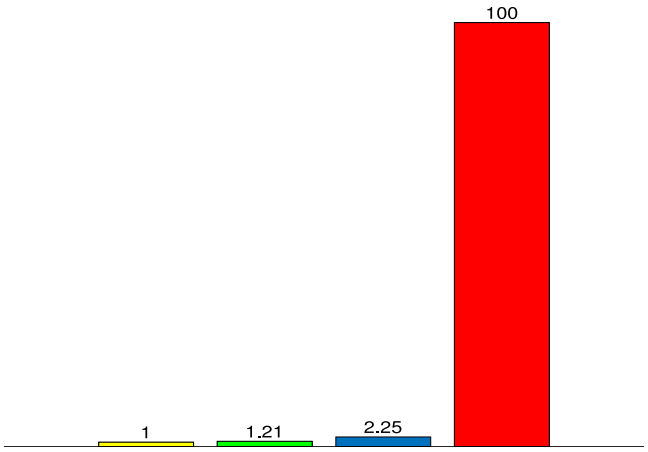

Fig. 7. Relative computational time of each optimization algorithm with respect to that of Algorithm 5 in the parameters estimation of Eqs. (13)–(14). From left to right: the Local Strategy (Alg 5), UMDAc (Alg 3), the UMDAc + Local Strategy (Alg 4), and Random-Search (Alg 6).

state space model, the estimator $\hat{\alpha}$ resulting from the optimization algorithm be close to the minimum value of the fitness function with high probability.

In summary, for this estimation problem, the histograms of the innovation estimator computed 100 times with Algorithms 5 and 6 are, with the exception of the parameter α_4, almost flat or with values concentrated at the extreme of the permissible values of the parameters (indicating that these extreme estimated values for the parameters lie out of the permissible values of such parameters). That is, the marginal distribution of these innovation estimates is quite similar to the uniform distribution of their initial values or with a mean significantly different of the true parameter values, which clearly demonstrates that the results of Algorithms 5 and 6 are irrelevant in this estimation problem. Contrarily, the values of the innovation estimator computed 100 times with Algorithms 3 and 4 are clearly concentrated around certain value inside the interval of permissible values for the parameters and close to the minimum of the fitness function, with more or less dispersion for each parameters. This clearly demonstrates that the proposed Algorithm 3 and 4 outperform Algorithm 5 and 6 in this second estimation problem.

To complete the evaluation on the performance of each optimization algorithm in this estimation problem, Fig. 7 presents the relative computational time of each one of them with respect to that of the local Algorithm 5. In this case, each simple run of the local optimization algorithm inside of Algorithm 6 performs – as average – 154 iterations and 6 evaluations of the fitness function per iteration. Then, each run of Algorithm 6 requires – as average – $100 \times 154 \times 6 = 92\,400$ fitness function evaluations to obtain an unuseful estimation result. On the contrary, obtaining a suitable estimation result, each run of Algorithms 3 requires only $100 + 95 \times 9 = 955$ evaluations of the same fitness function in the 10 generations with 100 individuals each one (discounting 5 individuals that are preserved in each generation because of the 5% for elitism). Hence, to compute the final parameter estimate, each run of Algorithm 4 requires – as average – a total of $955 + 924 = 1\,879$ evaluations of the fitness function, *i.e.*, the same 955 evaluations of the fitness function of Algorithm 3 plus $154 \times 6 = 924$ extra evaluations that – as average – are required by the local optimization algorithm. This remarks again the merit of the proposed Algorithms 3 and 4 for computing accurate innovation estimators of diffusion processes with a reduced number of evaluations of the fitness function.

To conclude we notice that – concerning accuracy and computational cost – the quality of the estimation of the five parameters in Eqs. (13)–(14) by Algorithms 5 and 6 decreases with respect to that in the estimation of the three parameters in Eqs. (9)–(10) by the same algorithms. In general, for estimation problems with larger number of parameters, Algorithms 3 and 4 tend to be more prone to perform better than Algorithms 5 and 4 as most local optimization methods suffer from the curse of dimensionality.

5. Conclusions

In this paper, we have considered two optimization methods based on the Estimation of Distribution Algorithms for computing the Innovation Estimators of unknown parameters of diffusion processes given a set of discrete and

noisy observations. The first method is exclusively based on a variant of the known Univariate Marginal Distribution Algorithm in continuous domain, whereas the second method includes a refinement for the outputs of the first one via a local optimization algorithm. The performance of these two optimization methods were evaluated in the parameter estimation of two types of diffusion models with complex nonlinear and stochastic dynamics. The numerical simulations demonstrate the feasibility of the considered method for the parameter estimation in situations where local optimization algorithms fail. This is particularly relevant in practice when suitable initial values for the parameters to be estimated are not available.

While our results show that UMDAc is able to deal with optimization scenarios where the commonly applied local optimization methods fail, there are still room for improvement. In particular, other EDAs that explicitly model and exploit multivariate interactions between parameters of the fitness function are worth to be evaluated in the computation of Innovation Estimators of diffusion processes. We leave this question as a line of future research.

Acknowledgments

The authors are grateful to the anonymous reviewers that contributed to improve the manuscript. Z.G.A. acknowledges partial financial support from the Brazilian agencies *Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro* (FAPERJ) and *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES) - Finance Code 001. R.S. acknowledges support by the Spanish Ministry of Science and Innovation (projects TIN2016-78365-R and PID2019-104966GB-I00), and the Basque Government (projects KK-2020/00049 and IT1244-19, and ELKARTEK program). All authors made equal contributions to the study and the publication.

References

[1] R. Armañanzas, I. Inza, R. Santana, Y. Saeys, J.L. Flores, J.A. Lozano, Y. Van de Peer, R. Blanco, V. Robles, C. Bielza, P. Larrañaga, A review of estimation of distribution algorithms in bioinformatics, BioData Min. 1 (6) (2008) http://dx.doi.org/10.1186/1756-0381-1-6.

[2] E. Bengoetxea, T. Miquélez, P. Larrañaga, J.A. Lozano, Experimental results in function optimization with EDAs in continuous domain, in: Estimation of Distribution Algorithms, Springer, 2002, pp. 181–194.

[3] T. Bollerslev, J.M. Wooldridge, Quasi-maximum likelihood estimation and inference in dynamic models with time-varying covariances, Econom. Rev. 11 (1992) 143–172.

[4] P.A. Bosman, J. Grahl, Matching inductive search bias and problem structure in continuous estimation of distribution algorithms, European J. Oper. Res. 185 (2008) 1246–1264.

[5] P.A. Bosman, D. Thierens, Expanding from discrete to continuous estimation of distribution algorithms: The IDEA, in: Parallel Problem Solving from Nature - PPSN VI 6th International Conference, in: Lecture Notes in Computer Science, vol. 1917, Springer, Paris, France, 2000.

[6] P.A. Bosman, D. Thierens, Numerical optimization with real-valued estimation-of-distribution algorithms, in: M. Pelikan, K. Sastry, E. Cantú-Paz (Eds.), Scalable Optimization Via Probabilistic Modeling: From Algorithms To Applications, in: Studies in Computational Intelligence, Springer-Verlag, 2006, pp. 91–120.

[7] C.P. Calderon, L. Janosi, I. Kosztin, Using stochastic models calibrated from nanosecond nonequilibrium simulations to approximate mesoscale information, J. Chem. Phys. 130 (14) (2009) 144908.

[8] C. Chiarella, H. Hung, T.-D. Tô, The volatility structure of the fixed income market under the hjm framework: A nonlinear filtering approach, Comput. Statist. Data Anal. 53 (6) (2009) 2075–2088.

[9] D. Cho, B. Zhang, Evolutionary continuous optimization by distribution estimation with variational bayesian independent component analyzers mixture model, in: Parallel Problem Solving from Nature (PPSN VIII), Vol. 3242, Springer, 2004, pp. 212–221.

[10] J.C. Cox, J.E. Ingersoll Jr, S.A. Ross, A theory of the term structure of interest rates, Econometrica 53 (2) (1985) 385–407.

[11] K. Deb, J. Horn, D.E. Goldberg, Multimodal deceptive functions, Complex Syst. 7 (1993) 131–153.

[12] T. Friedrich, T. Kötzing, M.S. Krejca, Unbiasedness of estimation-of-distribution algorithms, Theoret. Comput. Sci. 785 (2019) 46–59, http://dx.doi.org/10.1016/j.tcs.2018.11.001.

[13] E. Ghysels, A.C. Harvey, E. Renault, 5 stochastic volatility, Handbook of Statist. 14 (1996) 119–191.

[14] M. Hashemi, M.R. Meybodi, Univariate marginal distribution algorithm in combination with extremal optimization (eo, geo), in: L. BL., Z. L., K. J. (Eds.), Neural Information Processing – ICONIP 2011, in: Lectures Notes in Computer Science, vol. 7063, Springer, Berlin, 2011, pp. 220–227.

[15] A. Ibañez, R. Armañanzas, C. Bielza, P. Larrañaga, Genetic algorithms and gaussian bayesian networks to uncover the predictive core set of bibliometric indices, J. Am. Soc. Inf. Sci. Technol. 67 (2015) 1703–1721.

[16] J.C. Jimenez, Simplified formulas for the mean and variance of linear stochastic differential equations, Appl. Math. Lett. 49 (2015) 12–19.

[17] J.C. Jimenez, Approximate linear minimum variance filters for continuous-discrete state space models: convergence and practical adaptive algorithms, IMA J. Math. Control Inform. 36 (2019) 341–378.

[18] J.C. Jimenez, Bias reduction in the estimation of diffusion processes from discrete observations, IMA J. Math. Control Inform. 37 (2020) 1468–1505.

[19] J.C. Jimenez, R. Biscay, T. Ozaki, Inference methods for discretely observed continuous-time stochastic volatility models: A commented overview, Asia-Pac. Financial Mark. 12 (2006) 109–141.
[20] J.C. Jimenez, H. de la Cruz, Convergence rate of strong local linearization schemes for stochastic differential equations with additive noise, BIT Numer. Math. 52 (2012) 357–382.
[21] J.C. Jimenez, T. Ozaki, Local linearization filters for non-linear continuous-discrete state space models with multiplicative noise, Internat. J. Control 76 (12) (2003) 1159–1170.
[22] J.C. Jimenez, T. Ozaki, An approximate innovation method for the estimation of diffusion processes from discrete data, J. Time Series Anal. 27 (1) (2006) 77–97.
[23] N.L. Johnson, S. Kotz, Continuous Univariate Distributions – I, John Wiley & Sons, New York, 1970.
[24] S.C. Kamerlin, S. Vicatos, A. Dryga, A. Warshel, Coarse-grained (multiscale) simulations in studies of biophysical and chemical systems, Annu. Rev. Phys. Chem. 62 (2011) 41–64.
[25] P. Larrañaga, R. Etxeberria, J.A. Lozano, J.M. Peña, Optimization By Learning and Simulation of Bayesian and Gaussian Networks, Technical Report EHU-KZAA-IK-4/99, Department of Computer Science and Artificial Intelligence, University of the Basque Country, 1999.
[26] P. Larrañaga, H. Karshenas, C. Bielza, R. Santana, A review on probabilistic graphical models in evolutionary computation, J. Heuristics 18 (5) (2012) 795–819.
[27] P. Larrañaga, J.A. Lozano (Eds.), Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation, Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.
[28] J.R. Leon, A. Samson, Hypoelliptic stochastic fitzhugh–nagumo neuronal model: Mixing, up-crossing and estimation of the spike rate, Ann. Appl. Prob. 28 (4) (2018) 2243–2274.
[29] L. Ljung, P. Caines, Asymptotic normality of prediction error estimators for approximate system models, Stochastics 3 (1979) 29–46.
[30] L. Lozada-Chang, R. Santana, Univariate marginal distribution algorithm dynamics for a class of parametric functions with unitation constraints, Inform. Sci. 181 (11) (2011) 2340–2355.
[31] J.A. Lozano, P. Larrañaga, I. Inza, E. Bengoetxea (Eds.), Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms, Springer, 2006.
[32] H. Mühlenbein, T. Mahnig, Convergence theory and applications of the factorized distribution algorithm, J. Comput. Inf. Technol. 7 (1) (1998) 19–32.
[33] H. Mühlenbein, T. Mahnig, FDA – a scalable evolutionary algorithm for the optimization of additively decomposed functions, Evol. Comput. 7 (4) (1999) 353–376.
[34] H. Mühlenbein, T. Mahnig, Evolutionary computation and beyond, in: Y. Uesaka, P. Kanerva, H. Asoh (Eds.), Foundations of Real-World Intelligence, CSLI Publications, Stanford, California, 2001, pp. 123–188.
[35] H. Mühlenbein, T. Mahnig, Evolutionary optimization and the estimation of search distributions with applications to graph bipartitioning, Internat. J. Approx. Reason. 31 (3) (2002) 157–192.
[36] H. Mühlenbein, G. Paaß, From recombination of genes to the estimation of distributions I. Binary parameters, in: H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature - PPSN IV, in: Lectures Notes in Computer Science, vol. 1141, Springer, Berlin, 1996, pp. 178–187.
[37] J. Nielsen, M. Vestergaard, H. Madsen, Estimation in continuous-time stochastic volatility models using nonlinear filters, Int. J. Theor. Appl. Finance 3 (2) (2000) 1–30.
[38] K. Nolsoe, J.N. Nielsen, H. Madsen, Prediction-Based Estimating Function for Diffusion Processes with Measurement Noise, Technical Reports IMM-REP-2000-10, Informatics and Mathematical Modelling, Technical University of Denmark, 2000.
[39] T. Okabe, Y. Jin, B. Sendhoff, M. Olhofer, Voronoi-based estimation of distribution algorithm for multi-objective optimization, in: Proceedings of the 2004 Congress on Evolutionary Computation CEC-2004, IEEE Press, Portland, Oregon, 2004, pp. 1594–1601.
[40] T. Ozaki, The local linearization filter with application to nonlinear system identifications, in: Bozdogan H. (Ed.), Proceedings of the First US/Japan Conference on the Frontiers of Statistical Modeling: An Informational Approach, Kluwer Academic Publishers, 1994, pp. 217–240.
[41] T. Ozaki, J.C. Jimenez, V. Haggan-Ozaki, The role of the likelihood function in the estimation of chaos models, J. Time Series Anal. 21 (4) (2000) 363–387.
[42] M. Pelikan, Hierarchical Bayesian Optimization Algorithm. Toward a New Generation of Evolutionary Algorithms, in: Studies in Fuzziness and Soft Computing, vol. 170, Springer, 2005.
[43] M. Pelikan, D.E. Goldberg, F. Lobo, A survey of optimization by building and using probabilistic models, Comput. Optim. Appl. 21 (1) (2002) 5–20.
[44] J.J. Riera, J.C. Jimenez, X. Wan, R. Kawashima, T. Ozaki, Nonlinear local electrovascular coupling. ii: From data to neuronal masses, Hum. Brain Mapp. 28 (4) (2007) 335–354.
[45] C. Rocsoreanu, A. Georgescu, N. Giurgiteanu, The FitzHugh-Nagumo Model – Bifurcation and Dynamics, in: Mathematical Modelling: Theory and Applications, vol. 10, Springer Netherlands, 2000.
[46] R. Santana, P. Larrañaga, J.A. Lozano, Side chain placement using estimation of distribution algorithms, Artif. Intell. Med. 39 (1) (2007) 49–63.
[47] F. Schweppe, Evaluation of likelihood function for gaussian signals, IEEE Trans. Inform. Theory 11 (1965) 61–70.
[48] M. Sebag, A. Ducoulombier, Extending population-based incremental learning to continuous search spaces, in: Parallel Problem Solving from Nature - PPSN V, in: Lecture Notes in Computer Science, vol. 1498, Springer, Berlin Heidelberg, 1998, pp. 418–427.
[49] L. Shi, D. Li, X. Li, X. Wang, Dynamics of stochastic Fitzhugh–Nagumo systems with additive noise on unbounded thin domains, Stoch. Dyn. 20 (2020) 2050018.

[50] V. Solo, Some aspects of recursive parameter estimation, Internat. J. Control 32 (1980) 395–410.
[51] D.B. Thomas, W. Luk, P.H. Leong, J.D. Villasenor, Gaussian random number generators, ACM Comput. Surv. 39 (4) (2007) 11.
[52] S. Tsutsui, M. Pelikan, D.E. Goldberg, Evolutionary algorithm using marginal histogram in continuous domain, in: Optimization by Building and Using Probabilistic Models (OBUPM) 2001, San Francisco, California, USA, 2001, pp. 230–233.
[53] H.C. Tuckwell, R. Rodriguez, Analytical and simulation results for stochastic fitzhugh-nagumo neurons and neural networks, J. Comput. Neurosci. 5 (1) (1998) 91–113.
[54] H.C. Tuckwell, R. Rodriguez, F.Y. Wan, Determination of firing times for the stochastic Fitzhugh-Nagumo neuronal model, Neural Comput. 15 (1) (2003) 143–159.
[55] P.A. Valdes-Sosa, J.C. Jiménez, J.J. Riera, R. Biscay, T. Ozaki, Nonlinear EEG analysis based on a neural mass model, Biol. Cybernet. 81 (5–6) (1999) 415–424.
[56] P.A. Valdes-Sosa, J.M. Sanchez-Bornot, R.C. Sotero, Y. Iturria-Medina, Y. Aleman-Gomez, J. Bosch-Bayard, F. Carbonell, T. Ozaki, Model driven eeg/fmri fusion of brain oscillations, Hum. Brain Mapp. 30 (9) (2009) 2701–2721.
[57] C. Witt, Upper bounds on the running time of the univariate marginal distribution algorithm on onemax, Algorithmica (2018) 1–36.
[58] Q. Zhang, J. Sun, E.P.K. Tsang, J.A. Ford, Hybrid estimation of distribution algorithm for global optimization, Eng. Comput. 21 (1) (2003) 91–107.