Chilean Journal of Statistics
Vol. 4, No. 1, April 2013, 35–54

Stochastic Dynamic Systems
Research Paper

Reconstruction of chaotic dynamic systems using non-linear filters

Luis Sánchez[1] and Saba Infante[2,3]

[1]Department Mathematics, Face, University Carabobo, Venezuela
[2]Centre of Analysis, Treatment and Data Modelling, University Carabobo, Venezuela

(Received: 30 June 2011 · Accepted in final form: 03 October 2012)

Abstract

This article proposes a methodology based on sequential Monte Carlo techniques that permits state estimate of chaotic dynamic systems with Gaussian errors and non-linear dynamics in real time. Such systems arise naturally in many varied applications. We illustrate the methodology through the reconstruction of the states of the chaotic maps of Henon, Ikeda, Tinkerbell and Lorenz, using four different algorithms, namely, generic particle filter (GPF), particle filter with re-sampling (PFR), unscented Kalman filter (UKF) and an unscented particle filter (UPF). The performance of the filters was evaluated in terms of the empirical standard deviation and the computation times showing little variance among the estimated errors and a rapid execution of the algorithms.

Keywords: Dynamic systems · Nonlinear filters · Sequential Monte Carlo methods.

Mathematics Subject Classification: Primary 37D45 · Secondary 62L12.

1. Introduction

System dynamics is the physical science that studies the movement of physical phenomena changing and evolving with time. When the state behavior shows complicated, irregular, complex, erratic movements, difficult to predict precisely in space and time and which are very sensitive to the initial conditions and the instability, this leads to the study of chaotic dynamical processes to explain the irregularities observed. A dynamic system is a mathematical model which describes a process in terms of the possible states and a set of rules for determining present and future states in terms of past states. It is defined as

$$\mathbf{x}_t = \mathcal{M}_t(\mathbf{x}_{t-1}) + \mathbf{u}_t, \tag{1}$$

$$\mathbf{y}_t = \mathcal{H}_t(\mathbf{x}_t) + \mathbf{v}_t. \tag{2}$$

[1,2]Corresponding author. Email: sinfante@uc.edu.ve and sinfante64@gmail.com

ISSN: 0718-7912 (print)/ISSN: 0718-7920 (online)

http://www.soche.cl/chjs

Equation (1) represents the dynamic system where $\mathbf{x}_t \in \mathcal{X} \subset \mathrm{R}^n$ denotes the vector of unknown states at time t, $\mathbf{u}_t$ is a random error in the state estimation and $\mathcal{M}_t$ is a transition operator that maps the state space into the same state space. The Equation (2) represents the observed system where $\mathcal{H}_t$ is an operator that maps the state space into the observation space at time t, $\mathbf{y}_t \in \mathcal{Y} \subset \mathrm{R}^n$ is the vector of observations and $\mathbf{v}_t$ is a random observation error.

An importante task is to estimate the unknown states $\mathbf{x_{0:t}} = (\mathbf{x_0}, \ldots, \mathbf{x_t})^T$, based on the measures obtained in the observation process. $\mathbf{y_{1:t}} = (\mathbf{y_1}, \ldots, \mathbf{y_t})^T$. The main idea is to estimate recursively in time t, the posterior distribution $\mathrm{P}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ and its associated characteristics including the marginal distribution $\mathrm{P}(\mathbf{x}_t|\mathbf{y}_{1:t})$, its expected values, modes and variances. If one assumes that the white noise $\mathbf{u}_t \sim \mathrm{N}(0, \sigma^2_{\mathbf{u}})$ and $\mathbf{v}_t \sim \mathrm{N}(0, \sigma^2_{\mathbf{v}})$, it is also necessary to estimate the parameters $(\sigma^2_{\mathbf{u}}, \sigma^2_{\mathbf{v}})$.

In a given time t, the posterior distribution can be estimated using Bayes theorem as

$$\mathrm{P}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) \propto \frac{\mathrm{P}(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})\mathrm{P}(\mathbf{x}_{0:t})}{\int \mathrm{P}(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})\mathrm{P}(\mathbf{x}_{0:t})d\mathbf{x}_{0:t}}.$$

It is possible to obtain a recursive formula to estimate $\mathrm{P}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ by

$$\mathrm{P}(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1}) = \mathrm{P}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})\frac{\mathrm{P}(\mathbf{y}_{t+1}|\mathbf{x}_{t+1})\mathrm{P}(\mathbf{x}_{t+1}|\mathbf{x}_t)}{\mathrm{P}(\mathbf{y}_{t+1}|\mathbf{y}_{1:t})}.$$

In many cases $\mathrm{P}(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1})$ cannot be evaluated accurately, instead, is calculated the marginal distribution $\mathrm{P}(\mathbf{x}_t|\mathbf{y}_{1:t})$ as follows:

Step 1 Predict

$$\mathrm{P}(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int \mathrm{P}(\mathbf{x}_t|\mathbf{x}_{t-1})\mathrm{P}(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}. \tag{3}$$

Step 2 Update

$$\mathrm{P}(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{\mathrm{P}(\mathbf{y}_t|\mathbf{x}_t)\mathrm{P}(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int \mathrm{P}(\mathbf{y}_t|\mathbf{x}_t)\mathrm{P}(\mathbf{x}_t|\mathbf{y}_{1:t-1})d\mathbf{x}_t}. \tag{4}$$

Given the difficulty to calculate the integrals involved in Equations (3) and (4), one of the objectives of this research is to develop mathematical and computational strategies to filter the signals (observations) and use them to reconstruct the vector of states of the system considered in Equation (1). The filtering of a signal in real time is important since it reduces the cost of storage in the database, permits a rapid interpretation of the changes occurring in the phenomena under study and also provides the degrees of freedom for predicting the current states using the past state history of the system. Applications include weather forecasting; analysis of climate change, volatility of econometric series, communication signals and satellite images; development of sensors for the aviation, automobile and navigation industries; and protection of the environment from the propagation of dangerous contaminants.

This article will deal with some chaotic models arising from dynamic systems, which have been widely used to model the behavior of nature and is an area of active research; see Majda and Harlim (2012). Liu and Chen (1998) proposed a general framework based on Monte Carlo computation methods for real time dynamic systems, including importance sampling, resampling, rejection sampling, and Markov Chain techniques, with applications in engineering and econometric disequilibrium models. Meyer and Christensen

(2000) used an algorithm of Markov Chain Monte Carlo, specifically the Gibbs sampler, for estimating parameters in nonlinear models from time series of noisy data (logistic map, Moran-Ricker map, Henon map). Meyer and Christensen (2001) used the extended Kalman filter for the reconstruction of chaotic dynamical systems and compared these results with those obtained using the Gibbs sampler. Bremer and Kaplan (2001) showed how to use Markov Chain Monte Carlo techniques to estimate dynamic parameters on not observed components of a states vector, particularly the Gibbs sampler used to estimate states and reconstruct the attractors of Tinkerbell, Ikeda and Henon maps. Zhang et al. (2006) addressed the problem of secure chaotic communication using sequential Monte Carlo methods, showing that the modulated messages can be estimated using a particle filter algorithm, the signal can be nonlinear, which improves the security level for communication. They simulated using the Holmes map to verify the results. Majda et al. (2010) discuss some difficulties which arise when turbulence signals are filtered in real time in dynamic systems and indicate that common problems include: signals from nature arise from turbulent nonlinear dynamical systems, noisy spatio temporal signals which have amplitude over many spatial scales; imperfect models; the dimension of the state vector is very large order from 10^4 to 10^8; and spatio temporal observations are scarce or partially observed and at different scales. They propose to develop new computational strategies based on stochastic parametrization algorithms (extended Kalman filters, particle filtering) to reduce the dimensionality, to reduce model error, and to improve the filtering as well as the prediction capability. Majda et al. (2010) hope that this article will inspire other mathematicians, scientists and engineers to explore the use of modern applied mathematics in developing new algorithms for filtering turbulent dynamical systems.

In this paper we propose to use four stochastic algorithms to filter signals and reconstruct the states of four models arising from dynamic systems: Henon, Ikeda, Tinkerbell, and the Lorenz model. These models is known to have chaotic behavior, are governed by dynamic equations and the corresponding known time series are generated by equations of motion. In addition to estimating the states and reconstructing the attractors, a comparison in terms of errors and the CPU time between different algorithms is proposed. The considered chaotic models are defined as follows:

(1) **Henon map (Henon, 1976)**:

$$\begin{aligned} x_{t+1} &= y_t + 1 - ax_t^2 + u_{1t}, \\ y_{t+1} &= bx_t + v_{1t}, \end{aligned} \tag{5}$$

where: $u_{1t} \sim \mathrm{N}_p(0, \sigma_{u_1}^2)$, $v_{1t} \sim \mathrm{N}_p(0, \sigma_{v_1}^2)$, u_{1t} and v_{1t} are mutually independent.

(2) **Ikeda map (Casdagli, 1989)**:

$$\begin{aligned} x_{t+1} &= 1 + \rho\left[x_t \cos(\phi_t) - y_t \sin(\phi_t)\right] + u_{2t}, \\ y_{t+1} &= \rho\left[x_t \sin(\phi_t) + y_t \cos(\phi_t)\right] + v_{2t}, \end{aligned} \tag{6}$$

where $u_{2t} \sim \mathrm{N}_p(0, \sigma_{u_2}^2)$, $v_{2t} \sim \mathrm{N}_p(0, \sigma_{v_2}^2)$, u_{2t} and v_{2t} are mutually independent and $\phi_t = 0.4 - 6/[1 + x_t^2 + y_t^2]$.

(3) **Tinkerbell map (Nusse et al., 1997)**:

$$\begin{aligned} x_{t+1} &= x_t^2 - y_t^2 + ax_t + by_t + u_{3t}, \\ y_{t+1} &= 2x_t y_t + cx_t + dy_t + v_{3t}, \end{aligned} \tag{7}$$

where $u_{3t} \sim \mathrm{N}(0, \sigma_{u_3}^2)$ and $v_{3t} \sim \mathrm{N}(0, \sigma_{v_3}^2)$, u_{3t} and v_{3t} are mutually independent.

(4) **Lorenz model (Lorenz, 1963):** is a coupled system of nonlinear differential equations describing fluid dynamics:

$$\begin{aligned} \dot{x} &= s(y - x), \\ \dot{y} &= rx - y - xz, \\ \dot{z} &= xy - bz, \end{aligned} \tag{8}$$

where s, r, b are parameters and $\dot{x} = dx/dt$, $\dot{y} = dy/dt$, $\dot{z} = dz/dt$. The state vector $\mathbf{x} = (x, y, z)^T$ represents a position of the particles in phase space.

The stochastic algorithms proposed in this paper to estimate the states in the dynamical systems under study are: generic particle filter (GPF), particle filter with re-sampling (PFR), unscented Kalman filter (UKF) and an unscented particle filter (UPF). For an extensive review of these methods, see Anderson and Moore (1979), Arulampalam et al. (2002), West and Harrison (1997), Gordon et al. (1993), Kitagawa (1996), Doucet et al. (2000), Liu and West (2000), Pitt and Shephard (1999), Storvik (1999), Fearnhead (2002), Doucet et al. (2001), Fong et al. (2002), Godsill et al. (2004), Lin et al. (2005), Simon (2006), Snyder et al. (2008), Bengtsson et al. (2008), van der Merwe (2004), and Andrieu et al. (2010) among others.

The rest of the article is as follows. Section 2 defines prior, likelihood and posterior models; Section 3 discusses the sequential Monte Carlo techniques; Section 4 defines the measurement of model performance; Section 5 summarizes the main results and Section 6 is a discussion of the results.

2. Prior, Likelihood and Posterior Models

Consider the system of equations given in Equations (1) and (2). The proposed objective is to estimate the states $\mathbf{x}_{0:t}$ based on the observations $\mathbf{y}_{1:t}$; and be able to estimate x_{t+1} given the observations $\mathbf{y_{1:t}}$. Suppose that the prior model is

$$\mathrm{P}(\mathbf{x}_{0:t}) = \mathrm{P}(x_0) \prod_{i=0}^{t-1} \mathrm{P}(x_{i+1}|\mathbf{x}_{0:i}) = \mathrm{P}(x_0) \prod_{i=0}^{t-1} P(x_{i+1}|x_i),$$

where $\mathrm{P}(x_0)$ is known as the prior probability density function for the initial state and $\mathrm{P}(x_{i+1}|x_i) = \mathrm{P}(x_{i+1}|\mathbf{x}_{0:i})$ is a first order Markov process. The likelihood of the model is given by

$$\mathrm{P}(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}) = \prod_{i=1}^{t} \mathrm{P}(y_i|\mathbf{x}_{0:t}) = \prod_{i=1}^{t} \mathrm{P}(y_i|x_{i-1}).$$

Given the prior distribution of the states and the likelihood of the model, the stochastic posterior model is given by

$$\mathrm{P}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = C \times \mathrm{P}(y_1|x_0)\mathrm{P}(x_0) \left[\prod_{i=1}^{t-1} \mathrm{P}(y_{i+1}|x_i)\mathrm{P}(x_i|x_{i-1}) \right] \mathrm{P}(x_t|x_{t-1}),$$

where C is a normalizing constant that may be difficult to evaluate. The forecasting problem can be expressed as

$$\mathrm{P}(x_{t+1}|\mathbf{y}_{1:t}) = \int \cdots \int \mathrm{P}(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t}) dx_0 \cdots dx_t. \tag{9}$$

The above integrals given in Equations (3), (4) and (9) have an analytical solution only if the measurement and state equations are linear and the errors have Gaussian distribution. A solution is given by the Kalman filter. For a nonlinear function this problem may be hard to resolve and one must look for solution strategies using recursive techniques such as proposed in this work.

3. Sequential Monte Carlo Techniques

3.1 Generic particle filter

Given a simulation of N identically distributed independent random variables called particles, $\{\mathbf{x}_{0:t}^{(i)}, i = 1, \ldots, N\}$ according to a distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, an empiric estimator of this distribution is given by

$$\tilde{p}_N(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{1}{N} \sum_{i=1}^{N} \delta(\mathbf{x}_{0:t} - \mathbf{x}_{0:t}^{(i)}),$$

where $\delta(\mathbf{x}_{0:t} - \mathbf{x}_{0:t}^{(i)})$ denotes a point of mass of the function delta de Dirac located in $\mathbf{x}_{0:t}^{(i)}$. Here in lies the importance sampling method (IS). The IS can be modified to obtain the estimator $\tilde{p}_N(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ of $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ without modifying the simulated trajectories $\{\mathbf{x}_{0:t-1}^{(i)}, i = 1, \ldots, N\}$. This involves an importance function $q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ in time t that admits a marginal distribution in time $t-1$, say

$$q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}) q(x_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}), \tag{10}$$

iterating, one obtains

$$q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = q(x_0) \prod_{k=1}^{t} q(x_k|x_{0:k-1}, y_{1:k}).$$

Thus the importance function permits evaluating recursively in time the importance weights

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(y_t|x_t^{(i)}) p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})}.$$

A possible choice for the importance distribution is the prior distribution, that is,

$$q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = p(\mathbf{x}_{0:t}) = p(x_0) \prod_{k=1}^{t} p(x_k|x_{k-1}).$$

In this case the importance weights satisfy

$$w_t^{(i)} \propto w_{t-1}^{(i)} p(y_t|x_t^{(i)}).$$

We use the generic particle filter (GPF) developed by Lee (2005), which can be summarized as follows:

Step 1. Initialization: For $t = 0$; and $i = 1, \ldots, N$. Sample $x_0^{(i)} \sim P(x_0)$.

Step 2. Importance sampling: For $t = 1, \ldots, N$; and $i = 1, \ldots, N$:

- $x_t^{(i)} \sim q(x_t|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})$ and form the set

$$\mathbf{x}_{0:t}^{(i)} = (\mathbf{x}_{0:t-1}^{(i)}, x_t^{(i)}).$$

- We evaluate the importance weights

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(y_t|x_t^{(i)}) p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})}.$$

- Normalize the importance weights

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}}.$$

- Evaluate

$$\hat{N}_{\mathrm{TME}} = \frac{1}{\sum_{i=1}^{N} \left(\tilde{w}_t^{(i)}\right)^2}.$$

Chen (2003) proposed to compare the index ($\hat{N}_{\mathrm{TME}}$) with a predetermined threshold (N_U) to reduce the degeneracy. Thus,

Step 3. If $\hat{N}_{\mathrm{TME}} < N_U = N/2$

- For $i = 1, \ldots, N$, is performed $\tilde{w}_t^{(i)} = 1/N$.

Step 4. Outputs: posteriori distribution, expected value and posteriori covariance.

$$\tilde{p}_N(x_t|\mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} \tilde{w}_t^{(i)} \delta(x_t - x_t^{(i)}), \tag{11}$$

$$\tilde{x}_t = \mathrm{E}[x_t|\mathbf{y}_{1:t}] \approx \sum_{i=1}^{N} \tilde{w}_t^{(i)} x_t^{(i)}, \tag{12}$$

$$\tilde{P}_t \approx \sum_{i=1}^{N} \tilde{w}_t^{(i)} (x_t^{(i)} - \tilde{x}_t)(x_t^{(i)} - \tilde{x}_t)^T. \tag{13}$$

3.2 Particle filter with resampling

A problem that often arises with the generic particle filter is that of degeneration. After a few iterations many of the particles may have insignificant weights implying that the variance of the importance weights may increase in time. Doucet et al. (2000), established two proposals to select an importance function that minimizes the variance of the weights of the trajectories of the simulated states $\{\mathbf{x}_{0:t-1}^{(i)}\}$. An way to measure the degeneracy level in a particle set is to calculate the effective sample size (N_{TME}) introduced in Kong et al. (1994) and Liu (1996), defined as

$$N_{\text{TME}} = \frac{N}{1 + \text{Var}_{q(.|\mathbf{y}_{1:t})}\left[\tilde{w}_t^{(i)}\right]},$$

where $\text{Var}_{q(.|\mathbf{y}_{1:t})}[\tilde{w}_t^{(i)}]$ is the variance of the normalized weights. In practice, the calculation of N_{TME} may be complicated, but it can be estimated by

$$\hat{N}_{\text{TME}} = \frac{1}{\sum_{i=1}^{N}\left(\tilde{w}_t^{(i)}\right)^2}.$$

$\hat{N}_{\text{TME}}$ is then compared with a predetermined threshold $N_U = N/2$. The modified algorithm can be summarized as

Step 1 Initialization: For $t = 0$ and $i = 1, \ldots, N$. Sample: $x_0^{(i)} \sim p(x_0)$.

Step 2 Importance sampling: For $t = 1, \ldots$ and for $i = 1, \ldots, N$

a) Sample: $\tilde{x}_t^{(i)} \sim q(x_t|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})$; and construct the set

$$\tilde{\mathbf{x}}_{0:t}^{(i)} = \{\mathbf{x}_{0:t-1}^{(i)}, \tilde{x}_t^{(i)}\}.$$

b) Evaluate the importance weights

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(y_t|\tilde{x}_t^{(i)})p(\tilde{x}_t^{(i)}|\tilde{x}_{t-1}^{(i)})}{q(\tilde{x}_t^{(i)}|\tilde{\mathbf{x}}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})}.$$

c) Normalize the importance weights

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}}.$$

d) Evaluate

$$\hat{N}_{\text{TME}} = \frac{1}{\sum_{i=1}^{N}\left(\tilde{w}_t^{(i)}\right)^2}.$$

Step 3 Re-sampling:

- If $\hat{N}_{\text{TME}} \geq N_U$, let $\mathbf{x}_{0:t}^{(i)} = \tilde{\mathbf{x}}_{0:t}^{(i)}$, for $i = 1, \ldots, N$.
- Otherwise, if $\hat{N}_{\text{TME}} < N_U$, for $i = 1, \ldots, N$, sample an index $j(i)$ distributed according to a discrete distribution with N elements that satisfy $p_r\{j(i) = l\} = \tilde{w}_t^{(l)}$ for $l = 1, \ldots, N$.

- For $i = 1, \ldots, N$, let $\mathbf{x}_{0:t}^{(i)} = \tilde{\mathbf{x}}_{0:t}^{j(i)}$ and $\tilde{w}_t^{(i)} = 1/N$.

Step 4 The output of the algorithm is as given in the Equations (11), (12) and (13).

3.3 Unscented transformation

The unscented transformation (UT) is a method to calculate the statistics of the first and second order as: the mean and covariance of a random variable that suffers a nonlinear transformation $y = f(x)$ and is based on the beginning probabilistic that says than, it is easier to approximate a distribution of probability than approximating a non-linear arbitrary function (Julier and Uhlmann, 1996); is an elegant and exact way to calculate the mean and covariance of the function y using a Taylor series expansion. Let $\mathbf{x}$ be a vector of dimension n_x with known mean $\bar{x}$ and variance-covariance matrix P_{xx} . The UT calculates the mean and covariance of $y = f(x)$ as follows:

Step 1 Find deterministically $2n_x + 1$ vectors of sigma points x_i weighted by w_i, according to the following procedure:

$$x_0 = \bar{x} \quad ; \quad w_0^{(c)} = \frac{\lambda}{[n_x + \lambda]} + [1 - \alpha^2 + \beta] \quad ; \quad w_0^{(m)} = \frac{\lambda}{[n_x + \lambda]}; \quad i = 0.$$

$$x_i = \bar{x} + \left(\sqrt{[n_x + \lambda]P_{xx}}\right)_i \quad ; \quad w_i^{(c)} = w_i^{(m)} = \frac{1}{2[n_x + \lambda]} \quad ; \quad i = 1, \ldots, n_x.$$

$$x_i = \bar{x} - \left(\sqrt{[n_x + \lambda]P_{xx}}\right)_i \quad ; \quad w_i^{(c)} = w_i^{(m)} = \frac{1}{2[n_x + \lambda]} \quad ; \quad i = n_x + 1, \ldots, 2n_x.$$

$$\lambda = \alpha^2[n_x + \kappa] - n_x.$$

λ is a scale parameter that determines the direction of the sigma points, α determines the dispersion of the sigma points around $\bar{x}$ $[0 \leq \alpha \leq 1]$, κ is a second scale parameter that varies between $0 < \kappa < 3 - n_x$. The parameter β represents the degree of liberty used to introduce prior knowledge of the distribution of x_i; $\left(\sqrt{[n_x + \lambda]P_{xx}}\right)_i$ is ith column of the square root of the matrix $[n_x + \lambda]P_{xx}$ and w_i are weights associated with the ith sigma point such that $\sum_{i=0}^{2n_x} w_i = 1$.

Step 2 Propagate the sigma points by the nonlinear transformation:

$$y_i = f(x_i) \quad ; \quad i = 0, \ldots, 2n_x. \tag{14}$$

Step 3 Calculate the mean and covariance of y as

$$\bar{y} = \sum_{i=1}^{2n_x} w_i^{(m)} y_i \quad ; \quad P_{yy} = \sum_{i=1}^{2n_x} w_i^{(c)} (y_i - \bar{y})(y_i - \bar{y})^T,$$

the precision of the mean and covariance of $y = f(x)$ is guarantied by the Taylor series expansion, independently of the form of $f(x)$. The generated sigma points guarantee the convergence by the mechanism of the Monte Carlo method given in Julier et al. (2000).

3.4 Unscented Kalman filter

The unscented Kalman filter (UKF) is an extension of the UT applied to the state and observation equations given in the Equations (1) and (2). In the UKF the distribution of states is represented by a Gaussian random variable that is specified by a minimal set

of deterministically chosen sample points. It increases the state space model to include the components of the original states and the noise variable $x_t^a = (x_t^T u_t^T)^T$. The selection scheme of sigma points is applied to this new augmented state vector to calculate the corresponding sigma matrix $x_{i,t}^a$. Then the filter updates the average and the covariance by means of a Gaussian approximation of the distribution a posteriori of the states. Use the selection scheme sigma points of the UT to obtain x_t^a, then update the mean and covariance using the Kalman filter equations. The UKF summarized, considering:

Step 1 Given the nonlinear system:

$$\begin{aligned}
&\mathbf{x_t} = \mathcal{M}_t(\mathbf{x}_{t-1}) + \mathbf{u}_t = f_t(\mathbf{x}_{t-1}) + \mathbf{u}_t,\\
&\mathbf{y_t} = \mathcal{H}_t(\mathbf{x}_t) + \mathbf{v}_t = h_t(\mathbf{x}_t) + \mathbf{v}_t,\\
&\mathbf{u}_t \sim \mathrm{N}(0, Q_t),\\
&\mathbf{v}_t \sim \mathrm{N}(0, R_t).
\end{aligned} \tag{15}$$

Step 2 Initialization:

$$\begin{aligned}
\hat{x}_t^a &= \mathrm{E}[x_t^a],\\
P_t^a &= \mathrm{E}[(x_t^a - \hat{x}_t^a)(x_t^a - \hat{x}_t^a)^T].
\end{aligned}$$

Step 3 Generation of the sigma points:

$$x_{i,t}^a = \left[\hat{x}_t^a \quad ; \quad \hat{x}_t^a + \left(\sqrt{[n_x + q + \lambda]P_t^a}\right)_i \quad ; \quad \hat{x}_t^a - \left(\sqrt{[n_x + q + \lambda]P_t^a}\right)_i\right].$$

Step 4 Propagation:

$$\begin{aligned}
x_{i,t+1}^a &= f(x_{i,t}^a),\\
\hat{x}_{t+1}^- &= \sum_{i=0}^{2(n_x+q)} w_i^{(m)} x_{i,t+1}^a,\\
P_{t+1}^- &= \sum_{i=0}^{2(n_x+q)} w_i^{(c)} (x_{i,t+1}^a - \hat{x}_{t+1}^-)(x_{i,t+1}^a - \hat{x}_{t+1}^-)^T,\\
y_{i,t+1} &= h(x_{i,t+1}^a),\\
\hat{y}_t^- &= \sum_{i=0}^{2(n_x+q)} w_i^{(m)} y_{i,t+1},\\
P_{t+1}^{yy} &= \sum_{i=0}^{2(n_x+q)} w_i^{(c)} (y_{i,t+1} - \hat{y}_{t+1}^-)(y_{i,t+1} - \hat{y}_{t+1}^-)^T + R_t,\\
P_{t+1}^{xy} &= \sum_{i=0}^{2(n_x+q)} w_i^{(c)} (x_{i,t+1}^a - \hat{x}_{t+1}^-)(y_{i,t+1} - \hat{y}_{t+1}^-)^T.
\end{aligned}$$

Step 5 Updating:

$$
\begin{aligned}
K_{t+1} &= P_{t+1}^{xy}(P_{t+1}^{yy})^{-1}, \\
\hat{x}_{t+1}^{+} &= \hat{x}_{t+1}^{-} + K_{t+1}\left[y_t - \hat{y}_t^{-}\right], \\
P_{t+1}^{+} &= P_{t+1}^{-} - K_{t+1}P_{t+1}^{yy}K_{t+1}^{T}.
\end{aligned}
$$

3.5 Unscented particle filter

The new filter that results from using the UKF within the structure of the particle filters is called a unscented particle filter (UPF). The main idea is to use the UKF to generate the samples of the proposed distribution required in the PF

$$
x_t^{(i)} \sim q(x_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t}) = \mathrm{N}(\hat{x}_t^{(i)}, \hat{P}_t^{(i)}),
$$

where $\hat{x}_t^{(i)}$ and $\hat{P}_t^{(i)}$ is the mean and covariance of the points $x_t^{(i)}$, generated by the procedure of UKF. A cycle of the UPF may be summarized as follows:

Step 1 Initialization: In time $t = 0$

(1) For $i = 1, \ldots, N$. Sample $x_0^{(i)} \sim p(x_0)$.

(2) For $i = 1, \ldots, N$. Calculate the weights $w_0^{(i)} = p(y_0|x_0^{(i)})$:

$$
\begin{aligned}
\hat{x}_0^{(i)} &= \mathrm{E}\left[x_0^{(i)}\right] \quad ; \quad P_0^{(i)} = \mathrm{E}\left[(x_0^{(i)} - \hat{x}_0^{(i)})(x_0^{(i)} - \hat{x}_0^{(i)})^T\right]. \\
x_0^{a,(i)} &= \begin{pmatrix} x_0^{(i)} \\ w_0 \end{pmatrix} \quad ; \quad P_0^{a,(i)} = \begin{pmatrix} P_0^{(i)} & 0 \\ 0 & Q_0^{(i)} \end{pmatrix}.
\end{aligned}
$$

Step 2 Prediction and updating: For $t \geq 1$

(1) For $i = 1, \ldots, N$; update the particles with UKF.

a) Generate the sigma points:

$$
x_t^{a,(i)} = \left[\hat{x}_t^{a,(i)} \quad ; \quad \hat{x}_t^{a,(i)} + \sqrt{[n_x + q + \lambda]P_t^{a,(i)}}\,; \hat{x}_t^{a,(i)} - \sqrt{[n_x + q + \lambda]P_t^{a,(i)}}\right].
$$

b) Prediction and updating:

$$
\begin{aligned}
x_{t+1}^{x,(i)} &= f(x_t^{a,(i)}), \\
\hat{x}_{t+1|t}^{(i)} &= \sum_{j=0}^{2(n_x+q)} w_j^{(m)} x_{j,t+1}^{x,(i)}, \\
P_{t+1|t}^{(i)} &= \sum_{j=0}^{2(n_x+q)} w_j^{(c)} (x_{j,t+1}^{x,(i)} - \hat{x}_{t+1|t}^{(i)})(x_{j,t+1}^{x,(i)} - \hat{x}_{t+1|t}^{(i)})^T, \\
y_{t+1}^{(i)} &= h(x_{t+1}^{x,(i)}),
\end{aligned}
$$

$$\hat{y}_{t+1|t}^{(i)} = \sum_{j=0}^{2(n_x+q)} w_j^{(m)} y_{j,t+1}^{(i)},$$

$$P_{t+1}^{vv} = \sum_{j=0}^{2(n_x+q)} w_j^{(c)} (y_{j,t+1}^{(i)} - \hat{y}_{t+1|t}^{(i)})(y_{j,t+1}^{(i)} - \hat{y}_{t+1|t}^{(i)})^T + R_t,$$

$$P_{t+1}^{xy} = \sum_{j=0}^{2(n_x+q)} w_j^{(c)} [x_{j,t+1}^{x,(i)} - \hat{x}_{t+1|t}][y_{j,t+1}^{(i)} - \hat{y}_{t+1|t}^{(i)}]^T,$$

$$K_{t+1} = P_{t+1}^{xy}(P_{t+1}^{vv})^{-1},$$

$$\hat{x}_{t+1}^{(i)} = \hat{x}_{t+1|t}^{(i)} + K_{t+1}\left[y_t - h(\hat{x}_{t+1|t}^{(i)})\right],$$

$$\hat{P}_{t+1}^{(i)} = P_{t+1|t}^{(i)} - K_{t+1} P_{t+1}^{vv} K_{t+1}^T.$$

(2) For $i = 1, \ldots, N$; sample from the importance density:

$$x_t^{(i)} \sim \mathrm{N}(\hat{x}_t^{(i)}, \hat{P}_t^{(i)}).$$

(3) For $i = 1, \ldots, N$; calculate:

$$w_t^{(i)} = p(y_t|x_t^{(i)})\tilde{w}_{t-1}^{(i)}$$

and normalize the importance weights:

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_j w_t^{(j)}}.$$

(4) If $\hat{N}_{\mathrm{TME}} < N_U$, re-sample from the population $\{x_t^{(i)}, \tilde{w}_t^{(i)}\}$ to obtain a new set $\{x_t^{(i)}, 1/N\}$ with uniform weights.

Step 3 The output of the algorithm is as given in the Equations (11), (12) and (13).

4. Measurement of Model Performance

To validate the results, the models were implemented with M particles and N runtimes using the empiric standard deviation defined as

$$\sqrt{\mathrm{Var}[x_{t|l}]} = \frac{1}{N}\sum_{t=1}^{N}\left(\frac{1}{M}\sum_{j=1}^{M}\left(x_{t|l}^{(j)} - x_t^{(j)}\right)^2\right)^{\frac{1}{2}},$$

where $x_t^{(j)}$ is the true state simulated in the jth simulation; $x_{t|l}^{(j)} = \sum_{i=1}^{N} \tilde{w}_{t|l}^{(i)} x_t^{j,(i)}$, is the Monte Carlo estimator of $x_{t|l} = \mathrm{E}[x_t|\mathbf{y}_{1:t}]$ for the jth trial signal and $x_t^{j,(i)}$ is the ith simulated trajectory associated with the signal j; and $\tilde{w}_{t|t}^{(i)} = \tilde{w}_t^{(i)}$, is an importance weight.

5. Results

The GPF, PFR, UKF and UPF algorithms were implemented and evaluated in the four chaotic models: Henon, Ikeda, Tinkerbell and Lorenz. In first place, the model given in Equation (5) was considered supposing that the parameters $\theta = (a, b, \sigma^2_{\mathbf{u}_1}, \sigma^2_{\mathbf{v}_1})$ were known and that the states $x_{0:t}$ unknown. In this study, no real observations of the systems were available $y_{1:t}$ (observed variables), so the data was obtained by simulating the linear model $y_{t+1} = 0.3x_t + \eta_t$ (Bremer and Kaplan, 2001), with $x_0 = 1$, $y_1 = 0$, $\eta_t \sim \mathrm{N}(0, 0.001)$. After obtaining the observed sample, the next step was the initialization of the required parameters for each of the proposed algorithms. The parameters used for the particle filters were: $x_0 \sim \mathrm{N}(0.3, 0.01)$, $\mathbf{u}_{1t} \sim \mathrm{N}(0, 10)$, $\mathbf{v}_{1t} \sim \mathrm{N}(0, 0.1)$, for the GPF, and $x_0 \sim \mathrm{N}(0.6, 0.001)$, $\mathbf{u}_{1t} \sim \mathrm{N}(0, 0.007)$, $\mathbf{v}_{1t} \sim \mathrm{N}(0, 0.9)$ for the PFR. The selected parameters were $a = 1.4$ and $b = 0.3$, in order to simplify this study (this methodology could also estimate their values); and the selected importance function was $q_t(\cdot) = \mathrm{N}(\mu, \Sigma)$, where μ and Σ are known; the threshold used in the re-sampling was set at $N_u = N/2$. The specifications a priori used for the UKF and the UPF were $\hat{x}_0^+ = 0.6314$, $P_0^+ = 1$, $Q_t = 0.01$, $R_t = 0.01$, $\kappa = 0.0001$, $\beta = 2$ and $\alpha = 0.001$. The efficacy of each algorithm was evaluated using the empiric standard deviation (ESD) and the execution time (ET) for each filter. The algoritms were implemented in the **Matlab** software platform in a **Pentium Dual-Core** 2.8 GHz. The Figure (1) shows the true Henon model together with the posterior means of the estimated states for the GPF and the PFR. The Figure (2) shows the true Henon map together with the posterior means estimated by the UKF and the UPF. Observe that all of these filters adjust almost perfectly to the original map.

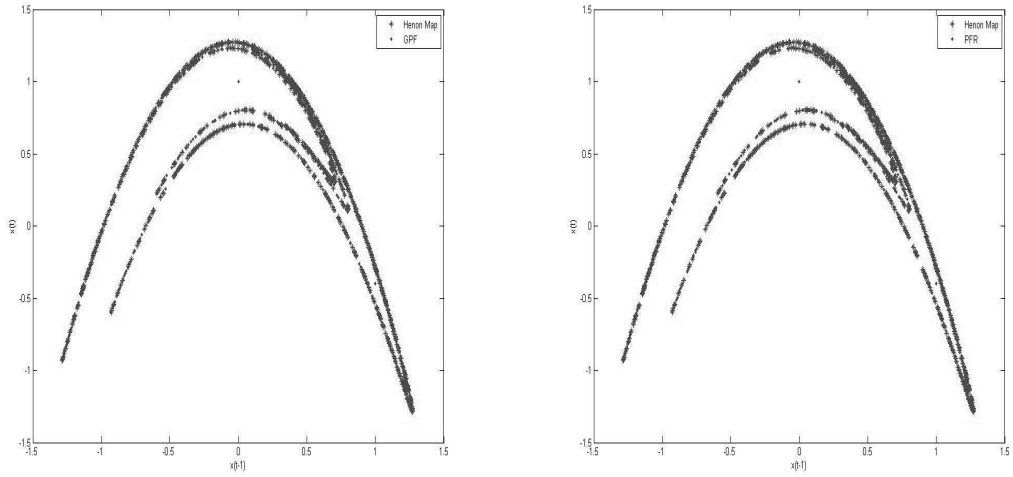


Figure 1. Algorithms GPF and PFR for Henon model.

The Table (1) shows a summary of the empiric standard deviation and execution times of the algorithms for the Henon model for 1000 simulations of length $N = 50, 100, 150, 200$. No significant differences in the estimated errors are observed for the different sample sizes however the execution times of the GPF and UPF do differ from the other filters. In second place, the model given in Equation (6) was considered. The data was obtained by simulating 1000 observations of the model $y_{t+1} = \rho\,[x_t \sin(\phi_t) + y_t \cos(\phi_t)] + \eta_t$, with $x_0 \sim \mathrm{N}(0, 0.001)$, $y_0 \sim \mathrm{N}(0, 0.001)$, $\rho = 0.92$, and $\eta_t \sim \mathrm{N}(0, 0.001)$. The initial values for the GPF and PFR were taken as $x_0 \sim \mathrm{N}(0.1, 0.1)$, $\mathbf{u}_{2t} \sim \mathrm{N}(0, 0.01)$, $\mathbf{v}_{2t} \sim \mathrm{N}(0, 10)$, and the importance function was a normal distribution with known mean and variance. The initial

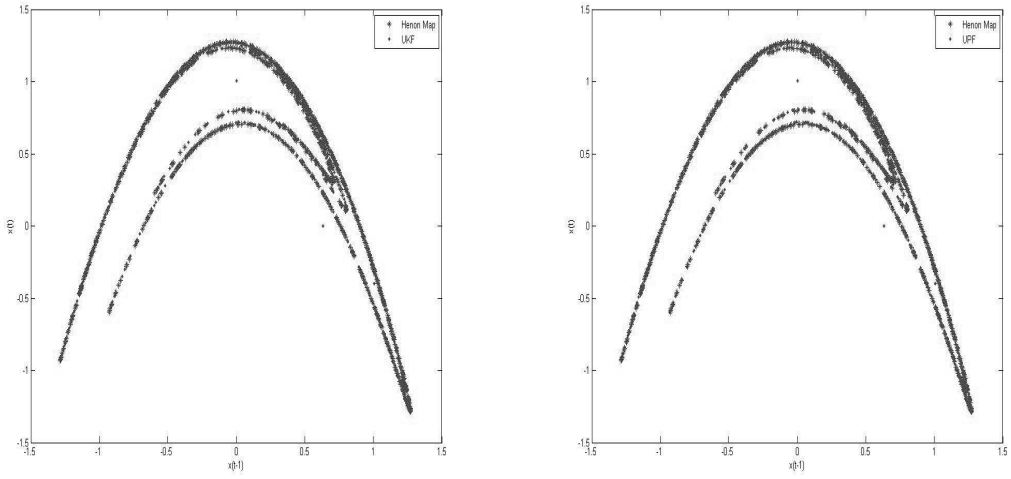


Figure 2. Algorithms UKF and UPF for the Henon model.

Table 1. MC filters: Henon model.

$\sqrt{\text{Var}[x_{t\|t}]}$	GPF	PFR	UKF	UPF
N=50	1.110	0.898	0.982	1.001
N=100	1.107	0.879	0.990	0.990
N=150	1.100	0.874	0.981	1.010
N=200	1.106	0.867	0.981	0.994
CPU Time (seg)	3.298	0.347	0.015	2.620

values for the UKF and UPF filters are $\hat{x}_0^+ = 0.1$, $P_0^+ = 1$, $Q_t = 0.0001$, $R_t = 1$, $\kappa = 0.0001$, $\beta = 2$ and $\alpha = 0.001$. In Figures (3) and (4) are shown the true Ikeda model together with the posterior means of the states estimated by the GPF, PFR, UKF and UPF filters, observing good estimations with respect to the true map. In Table (2) is shown a summary of the empiric standard deviation and the execution times for 1000 simulations of length $N = 50, 100, 150, 200$ for the Ikeda model. No significant differences in the estimated errors were observed for the different sample sizes however the execution times of the UPF did differ quite significantly from the other filters. In third place, the model given in Equation

Table 2. MC filters: Ikeda model.

$\sqrt{\text{Var}[x_{t\|t}]}$	GPF	PFR	UKF	UPF
N=50	2.436	2.513	2.414	2.629
N=100	2.497	2.443	2.447	2.504
N=150	2.493	2.374	2.535	3.225
N=200	2.496	2.394	2.385	2.647
CPU Time (seg)	0.561	0.552	0.019	3.682

(7) was considered. The experimental data was obtained simulating 1000 observations of the model $y_{t+1} = 2x_t y_t + cx_t + dy_t + \eta_t$, with initial values $x_0 = -0.72$, $y_0 = -0.64$, $c = 2$, $d = 0.5$ and $\eta_t \sim \text{N}(0, 0.001)$. To initialize the GPF and PFR the following procedure was used: $x_0 \sim \text{N}(0, 0.1)$, $\mathbf{u}_{3t} \sim \text{N}_p(0, 0.001)$, $\mathbf{v}_{3t} \sim \text{N}_p(0, 10)$, using a normal distribution with known mean and variance as the importance function. The model parameters were chosen

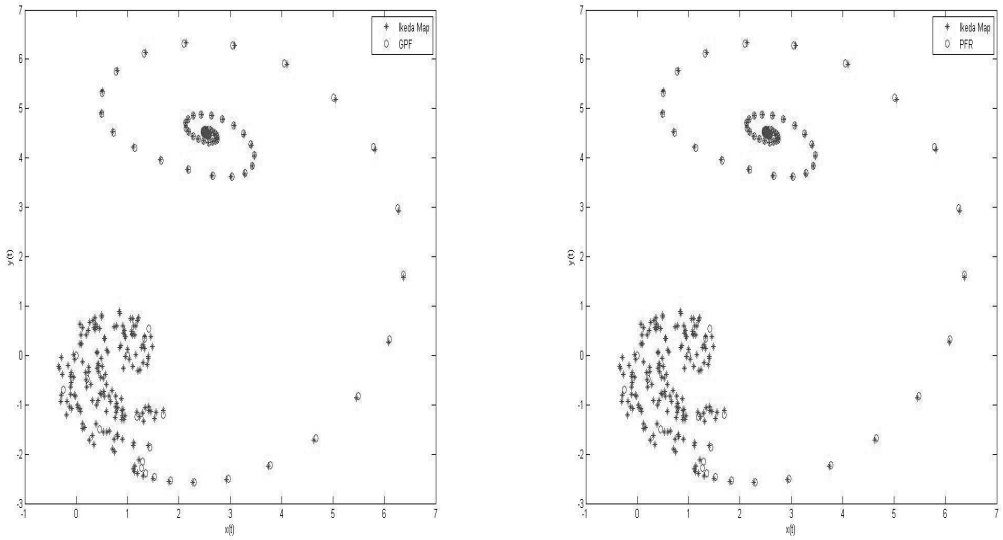


Figure 3. Algorithms GPF and PFR for the Ikeda model.

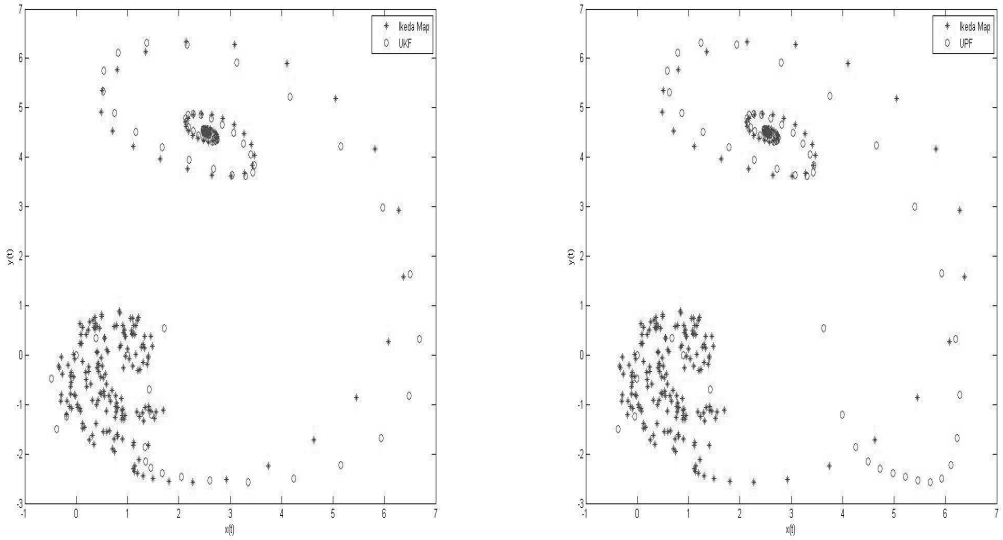


Figure 4. Algorithms UKF and UPF for the Ikeda model.

as $a = 0.9$, $b = -0.6013$, $c = 2$, $d = 0.5$. Similarly, the prior specifications used for the UKF and UPF filters are $\hat{x}_0^+ = 0$, $P_0^+ = 1$, $Q_t = 0.0001$, $R_t = 0.01$, $\kappa = 0.0001$, $\beta = 2$ and $\alpha = 0.001$. In the Figure (5) is seen the true Tinkerbell model along with the posterior means of the states estimated by the GPF and PFR algorithms. In the Figure (6) is seen the true Tinkerbell model along with the estimated posterior means of the states estimated by UKF and UPF; in each case the algorithms reconstruct a good approximation of the true model. In the Table (3) is found a summary of the empiric standard deviation of 1000 simulations of length $N = 50, 100, 150, 200$ and the execution time of the algorithms for the Tinkerbell model. No significant differences were noted in the estimated errors however the UPF was slowest in execution time.

Table 3. MC filters: Tinkerbell model.

$\sqrt{\text{Var}[x_{t\|t}]}$	GPF	PFR	UKF	UPF
N=50	0.243	0.258	0.387	0.355
N=100	0.260	0.251	0.380	0.351
N=150	0.268	0.242	0.387	0.354
N=200	0.252	0.247	0.392	0.358
CPU Time (seg)	0.408	0.389	0.014	2.718

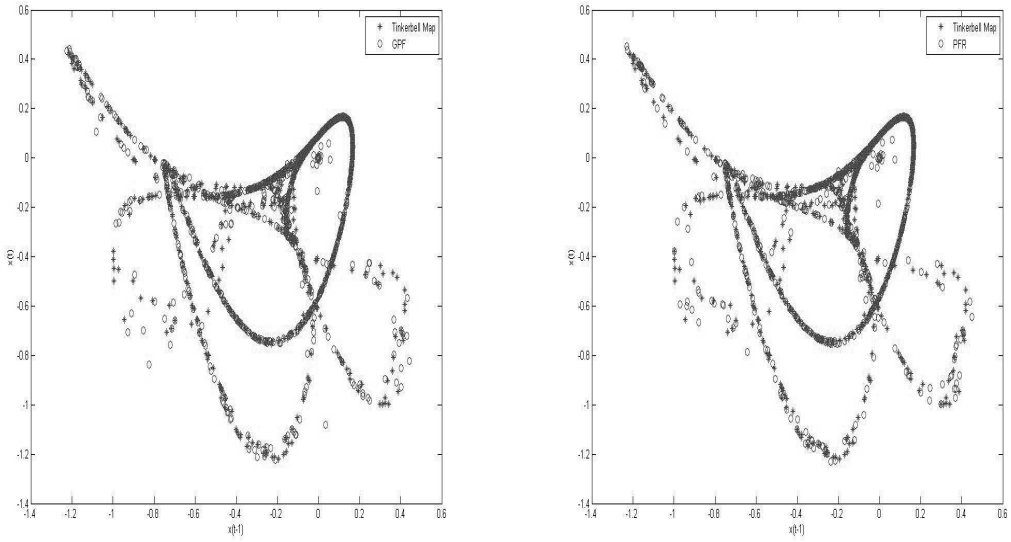


Figure 5. Algorithms GPF and PFR for the Tinkerbell model.

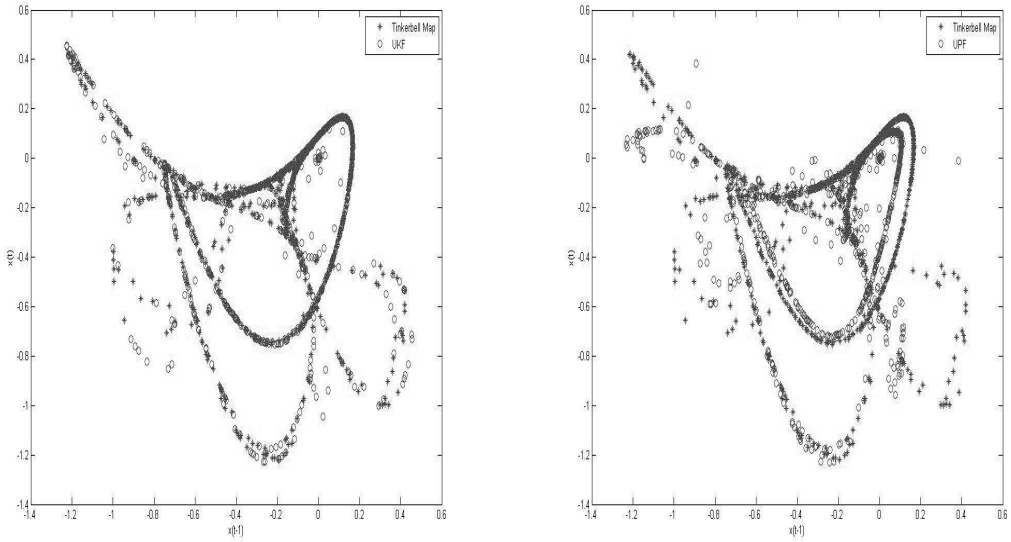


Figure 6. Algorithms UKF and UPF for the Tinkerbell model.

Finally, the model given in Equation (8) was studied. A discrete model was developed by a first order method of Euler considering $x_t = x_{t-1} + hf(x_{t-1})$, with stepsize $h = 0.009$. The discrete evolution equation is

$$\begin{aligned} x_{t+1} &= x_t + h(s(y_t - x_t)) + u_t, \\ y_{t+1} &= y_t + h(rx_t - y_t - x_t z_t) + v_t, \\ z_{t+1} &= z_t + h(x_t y_t - bz_t) + w_t, \end{aligned}$$

where $\mathbf{u}_{4t} \sim \mathrm{N}(0, \sigma^2_{\mathbf{u}_4})$, $\mathbf{v}_{4t} \sim \mathrm{N}(0, \sigma^2_{\mathbf{v}_4})$ and $\mathbf{w}_{4t} \sim \mathrm{N}(0, \sigma^2_{\mathbf{w}_4})$. In each step of time t, the observations are generated by a linear observation Equation (Chui and Chen, 2009)

$$\mathbf{y}_t = \mathbf{x}_t + \eta_t,$$

where $\mathbf{y}_t = (x_t^+, y_t^+, z_t^+)^T$, $\mathbf{x}_t = (x_t, y_t, z_t)^T$, and $\eta_t \sim \mathrm{N}(\mathbf{0}, \sigma^2_\eta \mathbf{I})$, $\mathbf{0}$ is a vector of zeros, and $\mathbf{I}$ is the identity matrix. The GPF and PFR filters were initialized by: $x_0 \sim \mathrm{N}(0.22, 0.1)$, $y_0 \sim \mathrm{N}(1.63, 0.1)$, $z_0 \sim \mathrm{N}(20.81, 0.1)$, $\mathbf{u}_{4t} \sim \mathrm{N}(0, 0.1)$, $\mathbf{v}_{4t} \sim \mathrm{N}(0, 0.1)$, $\mathbf{w}_{4t} \sim \mathrm{N}(0, 0.1)$, $\eta_t \sim \mathrm{N}(0, 100\mathbf{I})$, and a normal distribution with known mean and variance was used as the importance function. The initial values for UKF and UPF are

$$\hat{x}_0^+ = [0.2294, 1.6360, 20.81] \quad , \quad P_0^+ = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$Q_t = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{pmatrix} \quad , \quad R_t = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{pmatrix},$$

$\kappa = 0.0001$, $\beta = 2$ and $\alpha = 0.001$. In the Figure (7) is shown the true Lorenz model along with the posterior means estimated by GPF and PFR. In the Figure (8) is seen the true Lorenz model along with the posterior means of the states estimated by UKF and UPF; in each case observe that the algorithms reconstruct a good approximation of the true chaotic model. The Table (4) gives a summary of the empiric standard deviation and execution times of 1000 simulations of length $N = 50, 100, 150, 200$ for the posterior estimation of the states $x_{0:t}$ of the Lorenz model. No significant differences are noted in the errors and times however the UPF was again the slowest.

Table 4. MC filters: Lorenz model.

$\sqrt{\mathrm{Var}[x_{t\|t}]}$	GPF	PFR	UKF	UPF
N=50	7.741	7.749	6.826	7.532
N=100	7.736	7.776	6.826	7.645
N=150	7.732	7.730	6.815	7.692
N=200	7.722	7.747	6.832	7.727
CPU time (seg)	0.798	0.783	0.030	5.611

Figure 7. Algorithms GPF and PFR for the Lorenz model.

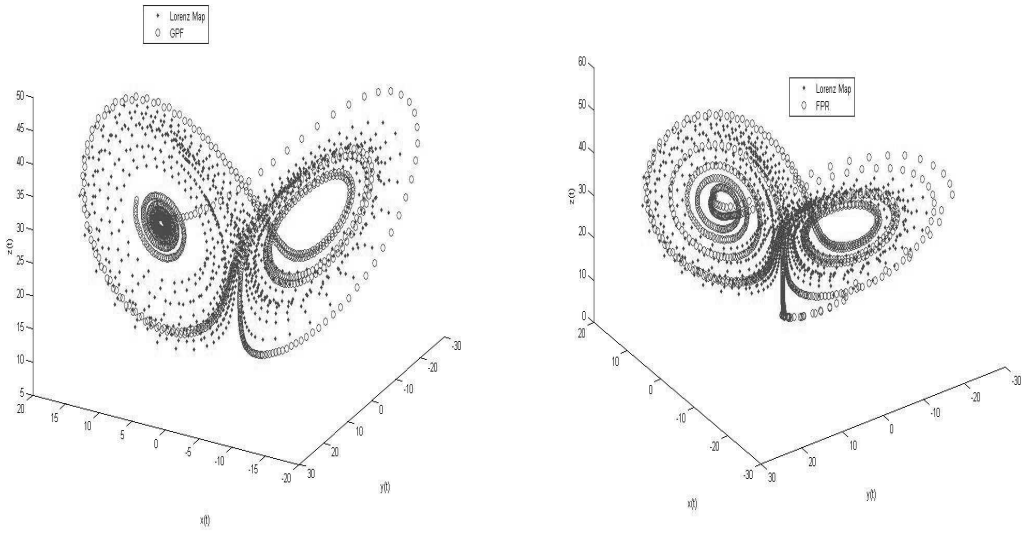
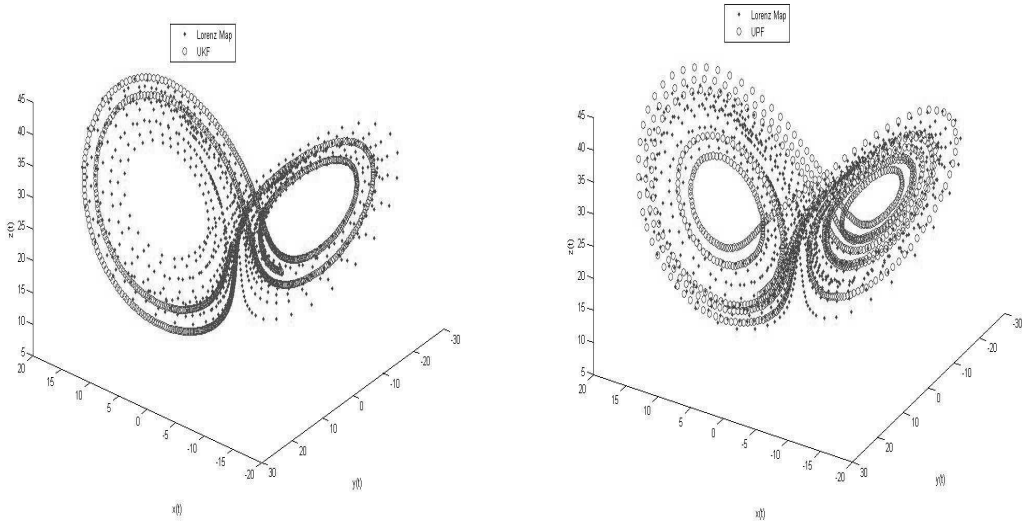


Figure 8. Algorithms UKF and UPF for the Lorenz model.



6. Discussion

In this article are proposed computationally efficient, mathematical strategies to filter the signals of chaotic models. In particular, ways were discussed to sequentially implement four algorithms the generic particle filter, the particle filter with resampling, the unscented Kalman filter and the unscented particle filter in order to estimate the states and reconstruct the attractors generated by the Henon, Ikeda, Tinkerbell and Lorenz maps. The recursive nature of the methods permit that they be computationally efficient and that the estimators obtained be optimal. In this study, the results show that the proposed filters reconstruct very well the states of the four considered models. The GPF, PFR, UKF and

UPF reconstruct exactly the original Henon map. In the Ikeda model, the results show that the GPF and PFR are more efficient that the UKF and UPF in the reconstruction of the original map. In the Tinkerbell model, the GPF and PFR reproduce exactly the original model while the UKF and UPF are less efficient in the reconstruction of the map. In the Lorenz model, the four algorithms performed the reconstruction of the original model adequately. Even though the reconstruction of chaotic maps is not new, here is presented an alternative way to perform the reconstruction. The methods have the advantage that they do not require suppositions in the models; they are applied in discrete models with non-linear structures and with Gaussian and non-Gaussian errors; the approximation is global and the exact solution is obtained when the number of particles tends to infinity. The disadvantage is that in problems of high dimension the algorithms GPF and PFR fail due to the problem of degeneration of the weights. Finally, an adequate measure of the performance of the filters is presented, specifically, the empiric standard deviation (ESD) was tabled for all four algorithms on each of the four considered models, showing low errors and little variability among them. The execution times of the algorithms on the different models were also given showing significantly longer times for the UPF when compared with the other filters.

Acknowledgments

The work is partially supported by the Council of Scientific and Humanistic the University of Carabobo, project CDCH-HM-0462-10.

References

Anderson, B., Moore, J., 1979. Optimal filtering. Englewood Cliffs, Prentice Hall, New Jersey.

Andrieu, C., Doucet, A., Holestein, R., 2010. Particle Markov Chain Monte Carlo methods. Journal of The Royal Statistical Society Series B - Statistical Methodology, 72, 1–33.

Arulampalam, S., Maskell, S., Gordon, N., Clapp, T., 2002. A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. IEEE Transactions on Signal Processing, 50, 174–188.

Bengtsson, T., Bickel, P., Li, B., 2008. Curse of dimensionality revisited: collapse of the particle filter in very large scale systems. Probability and Statistics: Essays in Honor of David Freedman, 2, 316–334.

Bremer, C., Kaplan, D., 2001. Markov Chain Monte Carlo estimation of nonlinear dynamics from time series. Physica D, 160, 116–126.

Casdagli, M., 1989. Nonlinear prediction of chaotic time series. Physica D, 35, 335–356.

Chen, Z., 2003. Bayesian filtering: from Kalman filter to particle filters, and beyond. Technical report. Adaptive System Laboratory, McMaster University.

Chui, C., Chen, G., 2009. Kalman Filtering with Real-Time Applications. Fourth edition. Springer, Berlin.

Doucet, A., De Freitas, J., Gordon, N., 2001. Sequential Monte Carlo in Practice. Springer Verlag, New York.

Doucet, A., Godsill, S., Andrieu, C., 2000. On sequential Monte Carlo sampling methods for Bayesian filtering. Statistics and Computing, 10, 197–208.

Fearnhead, P., 2002. MCMC, sufficient statistics and particle filter. Journal of Computational and Graphical Statistics, 11, 848–862.

Fong, W., Godsill, S., Doucet, A., West, M., 2002. Monte Carlo smoothing with application to audio signal enhancement. IEEE Transactions on Signal Processing, 50, 438–449.

Godsill, S., Doucet, A., West, M., 2004. Monte Carlo smoothing for nonlinear time series. Journal of the American Statistical Association, 99, 156–167.

Gordon, N., Salmond, D., Smith, A.F.M., 1993. Novel approach to nonlinear non Gaussian Bayesian state estimation. IEE Proceedings F, 140, 107–113.

Hnon, M., 1976. A two-dimensional mapping with a strange attractor. Communications in Mathematical Physics, 50, 69–77.

Julier, S., Uhlmann, J., 1996. A general method for approximating nonlinear transformations of probability distributions. Technical report. Department of Engineering Science, University of Oxford.

Julier, S., Uhlmann, J., Durrant, H., 2000. A new method for nonlinear transformati on of means and covariances in filter and estimators. IEEE Transactions on Automatic Control, 45, 477-482.

Kitagawa, G., 1996. Monte Carlo filter and smoother for nonlinear non Gaussian state models. Journal of Computational and Graphical Statistics, 5, 1–25.

Kong, A., Liu, S., Wong, H., 1994. Sequential imputations and Bayesian missing data problems. Journal of the American Statistical Association, 89, 278–288.

Lee, D., 2005. Nonlinear bayesian filtering with applications to estimation and navigation. Ph.D. Dissertation, Texas A & M University.

Lin, M., Zhang, J., Cheng, Q., Chen, R., 2005. Independent particle filters. Journal of the American Statistical Association, 100, 1412–1421.

Liu, J., 1996. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. Statistics and Computing, 6, 113–119.

Liu, J., Chen, R., 1998. Sequential Monte Carlo methods for dynamical systems. Journal of the American Statistical Association, 83, 1032–1044.

Liu, J., West, M., 2000. Combine parameter and state estimation in simulation based filtering. In Doucet, A., De Freitas, J.F.G., Gordon, N.J., (eds.). Sequential Monte Carlo Methods in Practice. Springer Verlag, New York, pp. 197–217.

Lorenz, E., 1963. Deterministic nonperiodic flow. Journal of the Athmospheric Sciences, 20, 130–141.

Majda, A., Harlim, J., Gershgorin, B., 2010. Mathematical strategies for filtering turbulent dynamical systems. Discrete and Continuos Dynamical Systems, 27, 441–486.

Majda, A., Harlim, J., 2012. Filtering Complex Turbulent Systems. Cambridge University Press, Cambridge.

Meyer, R., Christensen, N., 2000. Bayesian Reconstruction of Chaotic Dynamical Systems. University of Auckland, New Zealand.

Meyer, R., Christensen, N., 2001. Fast Bayesian Reconstruction of Chaotic Dynamical Systems Via Extenden Kalman Filtering. University of Auckland, New Zealand.

Nusse, H., Yorke, J., Marden, J., Sirovich, L., 1997. Dynamics: Numerical Explorations. Springer-Verlag, New York.

Pitt, M., Shephard, N., 1999. Filtering via simulation: auxiliary particle filters. Journal of the American Statistical Association, 94, 590–599.

Simon, D., 2006. Optimal State Estimation Kalman, H infinity, and Nonlinear Approaches. John Wiley and Sons, publication. Structural Dynamical Systems, Indian Institute of Science, Bangalore.

Snyder, C., Bengtsson, T., Bickel, P., Anderson, J., 2008. Obstacles to high dimensional particle filtering. American Meteorological Society, 136, 12.

Storvik, G., 2002. Paticle filters in state space models with the presence of uknown static parameters. IEEE Transactions on Signal Processing, 50, 281–289.

van der Merwe, R., 2004. Sigma-Point Kalman filters for probabilistic inference in dynamic state-Space models. Ph.D. OGI School of Science & Engineering at Oregon Health and Science University, Portland.

West, M., Harrison, J., 1997. Bayesian Forecasting and Dynamic Models. Second edition. Springer Verlag, New York.

Zhang, B., Chen, G., Zhou, D., 2006. Chaotic secure communication based on particle filtering. Chaos, Solitons and Fractals, 30, 1273–1280.