Available online at www.sciencedirect.com
ScienceDirect
Digital Signal Processing ••• (••••) •••–•••
Digital Signal Processing
www.elsevier.com/locate/dsp

Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization ☆

Joaquín Míguez

Departamento de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Avenida de la Universidad 30, 28911 Leganés, Madrid, Spain

Abstract

Cost-reference particle filtering (CRPF) is a recently proposed sequential Monte Carlo (SMC) methodology aimed at estimating the state of a discrete-time dynamic random system. The estimation task is carried out through the dynamic optimization of a user-defined cost function which is not necessarily tied to the statistics of the signals in the system. In this paper, we first revisit the basics of the CRPF methodology, introducing a generalization of the original algorithm that enables the derivation of some common particle filters within the novel framework, as well as a new and simple convergence analysis. Then, we propose and analyze a particle selection algorithm for CRPF that is suitable for implementation with parallel computing devices and, therefore, circumvents the main drawback of the conventional resampling techniques for particle filters. We illustrate the application of the methodology with two examples. The first one is an instance of one class of problems typically addressed using SMC algorithms, namely the tracking of a maneuvering target using a sensor network. The second example is the application of CRPF to solve a dynamic optimization problem.

Keywords: Sequential Monte Carlo; Particle filtering; Resampling; Stochastic optimization; Target tracking

1. Introduction

Many problems in physics, engineering, and biology can be modeled using discrete-time random dynamical systems in state-space form [1,2]. Usually, the signal of interest is the system state, which cannot be observed directly but through related noisy observations collected in a sequential manner. Unfortunately, the optimal (Bayesian) estimation of the state given these observations is analytically intractable except in very few specific cases (the most important one being the linear-Gaussian system, which can be handled by means of the well-known Kalman filter [3]). Therefore, both deterministic and Monte Carlo methods have been suggested to numerically approximate the desired state estimates [4,5].

☆ This work has been supported by Xunta de Galicia (project ref. PGIDIT04TIC105008PR), Ministerio de Educación y Ciencia of Spain (project DOIRAS, ref. TIC2003-02602), Comunidad de Madrid (project PRO-MULTIDIS-CM, ref. S0505/TIC/0223), and the EC (Network of Excellence CRUISE, ref. IST-4-027738).

E-mail address: joaquin.miguez@uc3m.es.

1051-2004/$ – see front matter
doi:10.1016/j.dsp.2006.09.003

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

SMC methods, also known as particle filtering (PF) techniques, are a family of recursive algorithms capable of attaining asymptotically optimal estimation of the state of a nonlinear and/or non-Gaussian discrete-time dynamical system [6–9]. They enable the online processing of the observations and their range of applicability is very broad. Indeed, few constraints are imposed on the dynamic system, other than the availability of a probabilistic model for the state and observation noise processes. PF algorithms aim at representing the a posteriori probability distribution of the system state given the available observations by means of an empirical discrete probability measure with random support. This measure consists of particles, which are samples in the state space, with associated weights, and can be updated recursively each time a new observation is collected. The moments of the discrete measure converge asymptotically, as the number of particles grows, to those of the true posterior probability of the state [10].

Their generality notwithstanding, particle filters have some limitations. A general PF algorithm consists of three steps [8]: (i) sampling of new particles according to a proposal distribution, (ii) weighting of the particles according to the observations, and (iii) selection of particles (commonly called resampling), which consists of stochastically discarding low-weight particles while replicating those with higher weights. The fundamental step of assigning weights to particles requires, at least, the ability to draw samples from the state prior probability distribution and to evaluate the likelihood function. Explicit probabilistic models may be readily available for some applications (e.g., digital communications), but may be hard to obtain for others. In general, working out a tractable and realistic probabilistic model often becomes a major task by itself. Another important drawback of conventional SMC techniques is that they are computationally intensive and, although the first two steps (sampling and weighting) can be carried out independently for each particle, the resampling step often becomes a bottleneck for parallel computation, as it involves the joint processing of all particles (see [11] for recent results on this topic).

A new family of recursive methods with the same basic algorithmic structure as particle filters (steps (i), (ii), and (iii) above) was introduced in [12]. Instead of relying on a probabilistic model of the dynamic system to approximate the a posteriori distribution of the state, the new methods perform the dynamic optimization of an arbitrary cost function (not necessarily tied to the statistics of the state and observation processes) that enables a quantitative discrimination of the "good" and "bad" particles. Although it may seem that exchanging one model (the probabilities) by another one (the cost) does not tackle the actual problem, it can be reasonably expected that simple cost functions can be easily found for almost any practical problem. In this way, we may often substitute a complex, or even intractable, probabilistic model by a much simpler one. This simplicity, in turn, translates into algorithms which are easier to design and implement. Moreover, this new class of algorithms has the potential to be distribution-free, since they provide state estimates that minimize the chosen cost function for a variety of different noise distributions in the system. The obvious drawback is that no optimal performance (in any statistical sense) can be claimed. The new methodology was termed cost-reference particle filtering (CRPF) in [12].

In Section 2 of this paper, we revisit the basics of CRPF and propose some generalizations of the original algorithm that enable the derivation of the most common (conventional) particle filters as instances of the new methodology. A simpler analysis of convergence, compared to the one in [12], is also presented. In Section 3, we investigate the selection of particles. We first note that particle selection procedures for CRPF are weaker than resampling algorithms for conventional particle filters, meaning that they are subject to less stringent constraints. Then, we prove that both the standard multinomial resampling scheme and a simpler procedure termed local selection are proper particle selection methods in a certain technical sense. The latter technique was suggested, with a more restrictive form and without analysis, in [12] and it has the advantage of lending itself to straightforward implementation with parallel computing devices. In order to illustrate the theoretical results, we have considered two numerical examples in Sections 4 and 5. The first one is a typical instance of the class of problems usually tackled with SMC algorithms, namely the tracking of a maneuvering target [13]. The second example is a modification of the Hartman 3 problem [14,15], in which a subset of the objective function parameters varies with time, hence a dynamic optimization problem has to be solved. Finally, Section 6 is devoted to the conclusions.

2. Cost-reference particle filtering

2.1. Problem statement

We consider the problem of estimating the sequence of states of a discrete-time Markovian dynamic system which cannot be observed directly. Formally, such systems can be described by the pair of equations

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

$$\mathbf{x}_t = f_x(\mathbf{x}_{t-1}, \mathbf{u}_t) \quad \text{(state equation)}, \tag{1}$$

$$\mathbf{y}_t = f_y(\mathbf{x}_t, \mathbf{v}_t) \quad \text{(observation equation)}, \tag{2}$$

where:

- $\{\mathbf{x}_t \in \mathbb{R}^{n_x}\}_{t\in\mathbb{N}}$ is the state process (the signal of interest),
- $\{\mathbf{u}_t \in \mathbb{R}^{n_u}\}_{t\in\mathbb{N}}$ is the system noise process,
- $f_x : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is the (possibly nonlinear) state transition function,
- $\{\mathbf{y}_t \in \mathbb{R}^{n_y}\}_{t\in\mathbb{N}}$ is the observation (or measurement) process,
- $\{\mathbf{v}_t \in \mathbb{R}^{n_v}\}_{t\in\mathbb{N}}$ is the observation noise process, and
- $f_y : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_y}$ is the (possibly nonlinear) observation function.

We use regular lower-case letters for real scalars (e.g., $a \in \mathbb{R}$), boldface lower-case letters for vectors (e.g., $\mathbf{a} \in \mathbb{R}^n$) and, eventually, boldface capital letters for matrices (e.g., $\mathbf{A} \in \mathbb{R}^{m\times n}$).

The goal is to estimate the sequence $\mathbf{x}_{0:t} \triangleq \{\mathbf{x}_0, \ldots, \mathbf{x}_t\}$ from the observations $\mathbf{y}_{1:t} \triangleq \{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$. Conventional PF algorithms are Bayesian techniques that address this problem by approximating the a posteriori probability density function (pdf) of the states, $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, using a discrete probability measure with random support. They require the availability of a probabilistic model including, at least, the a priori pdf of the state, $p(\mathbf{x}_0)$, the conditional density of the Markovian state process, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, and the likelihood function, $p(\mathbf{y}_t|\mathbf{x}_t)$. In this paper, we consider the alternative approach described next.

2.2. General algorithm

Instead of imposing an explicit probabilistic model on system (1)–(2), let us assume the availability of a lower-bounded real cost function of the form

$$\begin{aligned} &C : \mathbb{R}^{n_x(t+1)} \times \mathbb{R}^{n_y t} \to \left[C_t^{\text{opt}}, \infty\right) \\ &(\mathbf{x}_{0:t}, \mathbf{y}_{1:t}) \rightsquigarrow C(\mathbf{x}_{0:t}, \mathbf{y}_{1:t}) \end{aligned} \quad , \tag{3}$$

where

$$C_t^{\text{opt}} \triangleq \inf_{\mathbf{x}_{0:t}} C(\mathbf{x}_{0:t}, \mathbf{y}_{1:t}), \quad \left|C_t^{\text{opt}}\right| < \infty, \tag{4}$$

is the optimal cost at time t. Function C yields a quantitative assessment of a trial state sequence, $\mathbf{x}_{0:t}$, given the actual sequence of observations, $\mathbf{y}_{1:t}$, and CRPF is a methodology to recursively compute sequences of states with low (ideally minimal) cost. Let us further assume that function C has a recursive structure,

$$C(\mathbf{x}_{0:t}, \mathbf{y}_{1:t}) = \lambda C(\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t-1}) \circledast \triangle C(\mathbf{x}_t, \mathbf{y}_t), \tag{5}$$

where $0 \leqslant \lambda \leqslant 1$ works as a memory factor (similarly to the exponentially-weighted recursive least squares algorithm [2]); the operator $\circledast$ denotes a real function of two real arguments; and $\triangle C : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to [\triangle C_t^{\text{opt}}, \infty)$ is the incremental cost function, where

$$\triangle C_t^{\text{opt}} \triangleq \inf_{\mathbf{x}_t} \triangle C(\mathbf{x}_t, \mathbf{y}_t) \tag{6}$$

is the optimal incremental cost at time t. The use of an arbitrary operation $\circledast$, instead of the simple addition in [12], extends the applicability of the method. Not only a broader class of cost functions can be defined but it also enables the derivation of the most common particle filters as instances of the CRPF method, as will be shown in Section 2.4.

Assume now that a set of M particles and their associated costs up to time $t-1$, i.e., $\Omega_{t-1} = \{\mathbf{x}_{0:t-1}^{(i)}, C_{t-1}^{(i)}\}_{i=1}^{M}$, is available, where $C_{t-1}^{(i)} \triangleq C(\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t-1})$ is a convenient shorthand that we will often use in the sequel. We can introduce a risk function

$$\begin{aligned} &R : \left[C_{t-1}^{\text{opt}}, \infty\right) \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}, \\ &\left(C_{t-1}^{(i)}, \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t\right) \rightsquigarrow R\left(C_{t-1}^{(i)}, \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t\right), \end{aligned} \tag{7}$$

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

where $i \in \{1, \ldots, M\}$, in order to measure the adequacy of a candidate state at time $t-1$ to be propagated up to time t given the new observation, $\mathbf{y}_t$. It is natural to choose the risk function as a prediction of the cost at time t, i.e.,

$$R\big(C_{t-1}^{(i)}, \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t\big) = \lambda C_{t-1}^{(i)} \circledast \triangle C\big(f_x\big(\mathbf{x}_{t-1}^{(i)}, \mathbf{0}\big), \mathbf{y}_t\big), \quad i \in \{1, \ldots, M\} \tag{8}$$

(note the zero-noise argument of function f_x). Another straightforward choice of risk is the blind function that ignores the new observation and simply works with the current $(t-1)$ cost, i.e.,

$$R\big(C_{t-1}^{(i)}, \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t\big) = C_{t-1}^{(i)}, \quad i \in \{1, \ldots, M\}. \tag{9}$$

Using the above elements as building blocks, a CRPF algorithm aims at the online minimization of the cost function C by means of the recursive procedure below.

(1) **Initialization.** In order to avoid that the initial particles, $\{\mathbf{x}_0^{(i)}\}_{i=1}^M$, be infinitely far away from the true state, we assume knowledge of a set $E_0 \subset \mathbb{R}^{n_x}$ such that $\mathbf{x}_0 \in E_0$ (this is a structural rather than a probabilistic assumption). Then, initial samples can be drawn from an arbitrary pdf with domain E_0. Costs are set to a single constant, $C_0^{(i)} = c$, $i = 1, \ldots, M$.

(2) **Recursive loop.** Given the weighted particle set (wps) $\Omega_{t-1} = \{\mathbf{x}_{0:t-1}^{(i)}, C_{t-1}^{(i)}\}_{i=1}^M$, two steps are taken at time t:

(a) **Selection.** Convert Ω_{t-1} into a new wps $\hat{\Omega}_{t-1} = \{\hat{\mathbf{x}}_{0:t-1}^{(i)}, \hat{C}_{t-1}^{(i)}\}_{i=1}^M$, where $\hat{\mathbf{x}}_{0:t-1}^{(i)} \in \{\mathbf{x}_{0:t-1}^{(k)}\}_{k=1}^M$ and either $\hat{C}_{t-1}^{(i)} = C(\hat{\mathbf{x}}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})$ or $\hat{C}_{t-1}^{(i)} = c$ (constant) for all i.
Particles in $\hat{\Omega}_{t-1}$ are selected according to their risks, $R_t^{(i)} \triangleq R(C_{t-1}^{(i)}, \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)$. Deterministic selection is valid (particles with lower risks can be replicated and the ones with the larger risks be discarded). Alternatively, an straightforward stochastic procedure is the common multinomial resampling scheme, using the probability mass function (pmf) $\varsigma_t^{(i)} \propto \mu(R_t^{(i)})$, where $\mu : \mathbb{R} \to [0, \infty)$ is a nonnegative and monotonically nonincreasing function.

(b) **Propagation.** For $i = 1, \ldots, M$, let

$$\mathbf{x}_t^{(i)} \sim p_t\big(\mathbf{x}|\hat{\mathbf{x}}_{t-1}^{(i)}\big), \tag{10}$$

$$C_t^{(i)} = \lambda \hat{C}_{t-1}^{(i)} \circledast \triangle C_t^{(i)}, \tag{11}$$

where $\triangle C_t^{(i)} \triangleq \triangle C_t(\mathbf{x}_t^{(i)}, \mathbf{y}_t)$ and p_t is an arbitrary propagation pdf chosen by the user.

(3) **Estimation.** Whenever necessary, and concurrently with the recursive loop, estimates of the state can be drawn. Several possibilities exist. The straightforward one is to choose the particle with the smallest cost,

$$i_0 = \arg \min_{i \in \{1, \ldots, M\}} \big\{C_t^{(i)}\big\}, \quad \mathbf{x}_t^{\min} = \mathbf{x}_t^{(i_0)}. \tag{12}$$

It is also possible to use a pmf of the form $\pi_t^{(i)} \propto \mu(C_t^{(i)})$ (again, μ denotes a nonnegative and monotonically nonincreasing function) to obtain a *mean cost* estimate

$$\mathbf{x}_t^{\text{mean}} = \sum_{i=1}^{M} \mathbf{x}_t^{(i)} \pi_t^{(i)}, \tag{13}$$

which usually exhibits a smoother time evolution than that of $\mathbf{x}_t^{\min}$.

Many implementations of the CRPF method are possible for a single problem by adequately choosing the operation $\circledast$, the value of the memory factor λ and function μ. The latter will hereafter be referred to as generating function, and it plays an important role in the analysis of the selection step.

2.3. Cost function

The choice of cost function is a major feature of a specific CRPF algorithm. In many practical problems, it is given (e.g., the objective function in standard optimization problems such as the example in Section 5) or there is some "natural" cost to be used (e.g., Euclidean distance in target tracking problems with noisy observations of the

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

Table 1
Standard *bootstrap* filter

1. Initialization: draw M particles from the a priori pdf of the state, i.e., sample $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$, $i = 1, \ldots, M$.
Since there is no associated observation, the weights are uniform, $w_0^{(i)} = 1/M$, $i = 1, \ldots, M$.

2. Recursive step: given the wps $\{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{M}$,

(a) draw new particles
$\mathbf{x}_t^{(i)} \sim q(\mathbf{x}_t) = \sum_{k=1}^{M} w_{t-1}^{(k)} p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(k)})$, $i = 1, \ldots, M$,
where q is called the importance function, and

(b) assign importance weights proportional to the likelihoods,
$w_t^{(i)} \propto p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$
(the weights are normalized to ensure $\sum_{k=1}^{M} w_t^{(k)} = 1$).

Table 2
Derivation of the SBF from the CRPF methodology

1. $\lambda = 0$, $C_0^{(i)} = 0$, $\forall i$, and additive cost increments, $\circledast \equiv +$.

2. Incremental cost and generating function jointly designed to yield $\mu(\triangle C(\mathbf{x}_t, \mathbf{y}_t)) \propto p(\mathbf{y}_t|\mathbf{x}_t)$
(e.g., $\triangle C(x_t, y_t) = |x_t - y_t|^2$ and $\mu(z) = e^{-z/2}$ for a Gaussian likelihood).

3. Initial and propagation densities chosen in agreement with the probabilistic model, i.e., $p_0(\mathbf{x}_0) = p(\mathbf{x}_0)$ and $p_t(\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$, respectively.

4. Selection at each time step with blind risks $R_t^{(i)} = C_{t-1}^{(i)}$ and multinomial resampling with probabilities $\varsigma_t^{(i)} \propto \mu(R_t^{(i)})$.

target position). In general, the only strong requirement for $\triangle C$ is that it should be simple to evaluate, though not necessarily analytically. Therefore, CRPF is suitable to minimize absolute error functions, since no assumptions on differentiability are needed. Besides, lower bounded, real functions of the form $d(f_y(\mathbf{x}_t, 0), \mathbf{y}_{t+1})$ that comply with the triangle inequality, $d(a, b) \leqslant d(a, c) + d(c, b)$, are often good candidates. This includes pseudo-distances, like the Kullback-Leibler divergence (KLD) [16], that can be used in problems where the goal is to optimize a sequence of probability measures (e.g., in order to find the capacity of a communication channel) but even the property $d(a, b) = 0 \Leftrightarrow a = b$ is not strictly needed, since functions with several global maxima can also be handled with CRPF.

2.4. Conventional particle filters

Some of the most widely used PF algorithms can be obtained as instances of the CRPF method. Specifically, we show how a proper choice of the cost and risk functions, forgetting factor, generating function and the binary operator $\circledast$ can yield the standard *bootstrap* filter (SBF) [6] and the sequential importance sampling-resampling (SISR) algorithm [8].

Assume that a probabilistic model is imposed on system (1)–(2). The model depends on the pdf's of the state and observation noise processes, $p(\mathbf{u}_t)$ and $p(\mathbf{v}_t)$, respectively, but it is more conveniently stated in terms of the transition density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and the likelihood function $p(\mathbf{y}_t|\mathbf{x}_t)$. The SBF is a simple algorithm for approximating the so-called filtering pdf, $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, and can be outlined as shown in Table 1 [17]. Note that the selection and propagation of particles are carried out jointly in the SBF, when drawing from the mixture distribution $q(\mathbf{x}_t)$ in step 2(a).

The SBF algorithm can be derived from the general CRPF methodology by the choice of parameters shown in Table 2. The importance weights of the SBF, $w_t^{(i)}$, yield a discrete probability measure, $p_M(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{i=1}^{M} w_t^{(i)} \delta_i(\mathbf{x}_t - \mathbf{x}_t^{(i)})$, where $\delta(\cdot)$ is the Dirac delta function, whose moments converge almost surely (a.s.) to those of the filtering density $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ [10]. It is apparent that the CRPF algorithm specified by Table 2 yields $w_t^{(i)} \equiv \pi_t^{(i)} \propto \mu(C_t^{(i)})$, hence the minimum cost estimator of (12) becomes a maximum a posteriori (MAP) estimate of $\mathbf{x}_t$ and the mean cost estimator (13) yields the minimum mean square (MMSE) estimate of $\mathbf{x}_t$.

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

Table 3
Sequential importance sampling with resampling

1. Initialization: the same as in the SBF.
2. Recursive step: given $\{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{M}$,
(a) draw new particles from an arbitrary importance function, $\mathbf{x}_t^{(i)} \sim q(\mathbf{x}_t)$, $i = 1, \ldots, M$,
(b) assign the (normalized) importance weights, $w_t^{(i)} \propto w_{t-1}^{(i)} p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})/q(\mathbf{x}_t^{(i)})$,
(c) if needed, resample according to the weights $\{w_t^{(i)}\}_{i=1}^{M}$.

Table 4
Derivation of the SISR algorithm from the CRPF methodology

1. $\lambda = 1$, $C_0^{(i)} = 1$, $\forall i$, and multiplicative cost increments, $\circledast \equiv \times$.
2. Incremental cost designed to meet the quotient of probabilities $\triangle C(\mathbf{x}_t, \mathbf{y}_t) \propto q(\mathbf{x}_t)/p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})$.
3. Generating function $\mu(z) = z^{-1}$.
4. Initial pdf $p_0(\mathbf{x}_0) = p(\mathbf{x}_0)$.
5. Propagation densities in agreement with the importance function, $p_t(\mathbf{x}_t) = q(\mathbf{x}_t)$.
6. Selection (if needed) according to the risks $R_t^{(i)} = C_{t-1}^{(i)}$ and particle probabilities $\varsigma_t^{(i)} \propto \mu(R_t^{(i)})$, using the same resampling scheme as in the SISR procedure.

The SISR procedure is summarized in Table 3 [8], while the specification of the equivalent CRPF algorithm is given in Table 4. We note that, depending on the resampling scheme, the selected costs may be reset to $\hat{C}_t^{(i)} = 1/M$, $\forall i$. The same comments as in the SBF can be applied regarding the meaning of the minimum and mean cost estimators of (12) and (13).

2.5. Convergence of the propagation step

The particles in the set Ω_t are, in general, the output of two random procedures, namely (a) the selection step that converts Ω_{t-1} into $\hat{\Omega}_{t-1}$ and (b) the sampling of the propagation pdf's $p_t(\mathbf{x}|\hat{\mathbf{x}}_{t-1}^{(i)})$, $i = 1, \ldots, M$. Assume that both procedures can be jointly described by a single, yet possibly unknown, pdf that we denote as $p_t^s(\mathbf{x}|\{\hat{\mathbf{x}}_{t-1}^{(k)}\}_{k=1}^{M})$. Then we can simply write that $\mathbf{x}_t^{(i)} \sim p_t^s(\mathbf{x}|\{\hat{\mathbf{x}}_{t-1}^{(k)}\}_{k=1}^{M})$.

Resorting to p_t^s, we can define the set of states at time t that can be actually reached after the propagation step, i.e., let

$$\mathcal{X}_t \triangleq \left\{ \mathbf{x} \in \mathbb{R}^{n_x} \colon \forall \varepsilon > 0, \int\limits_{\mathbf{x}' \in \mathbb{R}^{n_x} \colon \|\mathbf{x}-\mathbf{x}'\| \leqslant \varepsilon} p_t^s\left(\mathbf{x}' \middle| \left\{\hat{\mathbf{x}}_{t-1}^{(k)}\right\}_{k=1}^{M}\right) d\mathbf{x}' > 0 \right\}, \tag{14}$$

where $\|\cdot\|$ denotes the norm of a vector in a multidimensional Euclidean space. There is a zero probability that the propagation step produces a particle with incremental cost lesser that $\triangle C_t^* = \triangle C(\mathbf{x}_t^*, \mathbf{y}_t)$, where

$$\mathbf{x}_t^* \triangleq \arg\min_{\mathbf{x} \in \mathcal{X}_t} \left\{ \triangle C(\mathbf{x}, \mathbf{y}_t) \right\} \tag{15}$$

is not necessarily unique. We use the following notation for two particular classes of sets containing the incremental cost $\triangle C_t^*$,

$$I_t(\varepsilon) \triangleq \left[\triangle C_t^*, \triangle C_t^* + \varepsilon \right) \tag{16}$$

and its discrete counterpart,

$$I_t^M(\varepsilon) \triangleq I_t(\varepsilon) \cap \left\{ \triangle C_t^{(i)} \right\}_{i=1}^{M}, \tag{17}$$

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

where $\cap$ denotes set intersection.

The following theorem grants that we always have particles with an incremental cost arbitrarily close to $\triangle C_t^*$ as long as M is sufficiently large.

Theorem 1. *Assume that $\triangle C(\mathbf{x}_t, \mathbf{y}_t)$ is continuous at $\mathbf{x}_t^*$ and let $\delta_M > 0$ be a real function of M. If $\delta_M \geqslant M^{-\psi}$ for any $0 < \psi < 1$, then*

$$\lim_{M\to\infty} \left| I_t^M(\delta_M) \right| = \infty \quad (a.s.). \tag{18}$$

See Appendix A for a proof.

Given a discrete set A, notation $|A|$ indicates the number of elements contained in the set. Note that Theorem 1 alone guarantees that $\lim_{M\to\infty} \min\{\triangle C_{t+1}^{(i)}\}_{i=1}^M = \triangle C_t^*$ a.s., i.e., that for M sufficiently large we will always find some particle arbitrarily close to a the minimum of $\triangle C$ in $\mathcal{X}_{t+1}$. This is actually (a convenient version of) the basic result exploited by random search algorithms (see, e.g., [18] and references therein).

3. Selection methods

A practically inevitable step in conventional PF is resampling, which consists of stochastically replicating the "good" particles (those with a high importance weight) while discarding the "bad" ones, and is aimed at avoiding the degeneracy of the particle filter [8].

In general, the goal of resampling is to obtain a set of particles that still approximate the desired posterior pdf but have more evenly distributed weights [19,20]. Selection procedures for CRPF are possibly weaker, in the sense that they are subject to less stringent constraints. In particular, if we let $\Omega_t = \{\mathbf{x}_{0:t}^{(i)}, C_t^{(i)}\}_{i=1}^{\acute{M}}$ be the wps provided by the algorithm at time t, we do not request that the resulting selected set, $\hat{\Omega}_t = \{\hat{\mathbf{x}}_{0:t}^{(i)}, \hat{C}_t^{(i)}\}_{i=1}^{M}$, be equivalent in any sense to Ω_t (also note that we can allow $\acute{M} \neq M$ for generality). In fact, since the goal is to preserve the particles with smaller risks, we term a selection technique as *proper* if it simply guarantees that

$$\lim_{M\to\infty} \bar{R}_t - \bar{\hat{R}}_t \geqslant 0, \tag{19}$$

where

$$\bar{R}_t = \sum_{i=1}^{\acute{M}} \varsigma_t^{(i)} R_t^{(i)}, \tag{20}$$

$$\bar{\hat{R}}_t = \sum_{i=1}^{M} \hat{\varsigma}_t^{(i)} \hat{R}_t^{(i)} \tag{21}$$

are the mean risk before and after selection, respectively. We abide by the notation introduced in Section 2.2, hence $\varsigma_t^{(i)} \propto \mu(R_t^{(i)})$ is the selection pmf and we use $\hat{R}_t^{(i)} = R_t^{(k)}$ to denote the risk of the selected particle $\hat{\mathbf{x}}_{0:t}^{(i)} = \mathbf{x}_{0:t}^{(k)}$, $i \in \{1, \ldots, M\}$, $k \in \{1, \ldots, \acute{M}\}$. Accordingly, the pmf that yields the mean risk of the selected particles is defined as $\hat{\varsigma}_t^{(i)} \propto \mu(\hat{R}_t^{(i)})$. Intuitively, we say a selection method is proper when it does not cause an increase in the mean risk of particles for large M. No other constraints are imposed. Deterministic selection, as an example, is a valid approach if we accept criterion (19), although we will focus on stochastic selection procedures in this paper.

The advantage of relaxing the design of selection techniques is the possibility to minimize the interaction among particles. Beware that the joint processing of particles required by standard resampling techniques is a bottleneck for the parallelization of the computations, and it is severely restraining the practical implementation of PF algorithms using massively parallel VLSI devices [11].

In the remaining of this section we investigate the use of two stochastic selection approaches. The first one is a straightforward extension of multinomial resampling [8,19,20] to the CRPF context, while the second is a refinement of the easy-to-parallelize method termed local selection in [12]. We prove that both methods are proper according to (19) and briefly discuss the practical advantages of the local approach.

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

Table 5
Global selection preserving the costs of the selected particles

Original wps: $\Omega_t = \{\mathbf{x}_{0:t}^{(i)}, C_t^{(i)}\}_{i=1}^{\acute{M}}$.
Risk computation: $R_{t+1}^{(i)} = R(C_t^{(i)}, \mathbf{x}_t^{(i)}, \mathbf{y}_{t+1})$.
Selection pmf: $\varsigma_{t+1}^{(i)} \propto \mu(R_{t+1}^{(i)})$.
Multinomial resampling: $\hat{\mathbf{x}}_{0:t}^{(i)} \sim p(\hat{\mathbf{x}}_{0:t} = \mathbf{x}_{0:t}^{(k)}) = \varsigma_{t+1}^{(k)}$, for $i = 1, 2, \ldots, M$ and $k \in \{1, \ldots, \acute{M}\}$.
New wps: $\hat{\Omega}_t = \{\hat{\mathbf{x}}_{0:t}^{(i)}, \hat{C}_t^{(i)}\}_{i=1}^{M}$, where $\hat{C}_t^{(i)} = C_t^{(k)}$ if $\hat{\mathbf{x}}_{0:t}^{(i)} = \mathbf{x}_{0:t}^{(k)}$.

Table 6
Local selection with k offsprings per original particle ($M = k\acute{M}$) and preserving the costs of the selected particles

Original wps: $\Omega_t = \{\mathbf{x}_{0:t}^{(i)}, C_t^{(i)}\}_{i=1}^{\acute{M}}$.
Risk computation: $R_{t+1}^{(i)} = R(C_t^{(i)}, \mathbf{x}_t^{(i)}, \mathbf{y}_{t+1})$.
Selection pmf's: for $i = 1, 2, \ldots, \acute{M}$ and $l = i-1, i$,
$\varsigma_{i,t+1}^{(l)} = \mu(R_{t+1}^{(l)}) / \sum_{j=i-1}^{i} \mu(R_{t+1}^{(j)})$,
where $R_{t+1}^{(0)} \triangleq R_{t+1}^{(\acute{M})}$.
Local resampling:
For $i = 1, 2, \ldots, \acute{M}$,
For $r = (i-1)k + 1, \ldots, ik$,
$\hat{\mathbf{x}}_{0:t}^{(r)} \sim p(\hat{\mathbf{x}}_{0:t} = \mathbf{x}_{0:t}^{(l)}) = \varsigma_{i,t+1}^{(l)}$, $l \in \{i-1, i\}$.
New wps: $\hat{\Omega}_t = \{\hat{\mathbf{x}}_{0:t}^{(i)}, \hat{C}_t^{(i)}\}_{i=1}^{M}$, where $\hat{C}_t^{(i)} = C_t^{(k)}$ if $\hat{\mathbf{x}}_{0:t}^{(i)} = \mathbf{x}_{0:t}^{(k)}$.

3.1. Global vs local selection

The extension of the common multinomial resampling is summarized in Table 5 and will be termed *global selection*. Obviously, it requires the joint processing of all particles and, therefore, prevents the implementation of CRPF algorithms using parallel hardware. Nevertheless, it may still be a good choice for relatively simple problems, and it is a proper method.

Theorem 2. *Global selection is proper according to* (19)*, with convergence of the limit in probability*[1] (*i.p.*).

See Appendix B for a proof.

In order to ease parallel implementation, a simple local selection scheme was proposed in [12]. This procedure significantly reduces the interaction among particles by constraining resampling to within $\acute{M}$ independent subsets of three particles, namely $\{\{\mathbf{x}_{0:t}^{(l)}, C_t^{(l)}\}_{l=i-1}^{i+1}\}_{i=1}^{\acute{M}}$. A further simplification leads to sampling within sets including only pairs of consecutive particles, i.e., $\{\{\mathbf{x}_{0:t}^{(l)}, C_t^{(l)}\}_{l=i-1}^{i}\}_{i=1}^{\acute{M}}$, (assume $\mathbf{x}_{0:t}^{(0)} \triangleq \mathbf{x}_{0:t}^{(\acute{M})}$ and $C_t^{(0)} \triangleq C_t^{(\acute{M})}$) which reduces particle interaction to the communication of the pair $\{\mathbf{x}_{0:t}^{(i)}, R_{t+1}^{(i)}\}$ from particle i to particle $i+1$. This is the form of the local selection algorithm summarized in Table 6, which admits a straightforward implementation using parallel computing devices with limited communication capabilities (one-directional and between adjacent processors only). Moreover, the method can be shown to be statistically sound according to the adopted criterion if we assume $M = \acute{M}$ and the following regularity conditions.

(R1) Both $|R_t^{(j)}| < \infty$ and $\mu(R_t^{(j)}) < \infty$, $j = 1, \ldots, M$.
(R2) Let $A_M \triangleq \{1, \ldots, M\}$ and

[1] Statements of the type $\lim_{n\to\infty} a(n) > \lim_{n\to\infty} b(n)$ i.p. should be read as $\lim_{n\to\infty} \text{probability}\{a(n) \leqslant b(n)\} = 0$.

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

$$A_M^+ \triangleq \left\{ i \in A_M \colon\ R_t^{(i)} - \tilde{R}_t^{(i-1)} \geqslant 0 \right\}, \tag{22}$$

$$A_M^- \triangleq \left\{ i \in A_M \colon\ R_t^{(i)} - \tilde{R}_t^{(i-1)} < 0 \right\}, \tag{23}$$

assuming $R_t^{(i)} = R_t^{(j)} \Leftrightarrow i = j$, so that $A_M = A_M^+ \cup A_M^-$, and defining $R_t^{(0)} \triangleq R_t^{(M)}$. Then, the following limits exist

$$\lim_{M\to\infty} \frac{\sum_{i\in A_M^+} R_t^{(i)} - R_t^{(i-1)}}{|A_M^+|} = \lim_{M\to\infty} \frac{\sum_{i\in A_M^-} R_t^{(i-1)} - R_t^{(i)}}{|A_M^-|} = d_R > 0. \tag{24}$$

(R3) $\lim_{M\to\infty} |A_M^+|/|A_M^-| = 1$.

(R4) The following limits exist

$$\lim_{M\to\infty} \frac{\sum_{i\in A_M^+} \mu(R_t^{(i-1)}) - \mu(R_t^{(i)})}{|A_M^+|} = \lim_{M\to\infty} \frac{\sum_{i\in A_M^-} \mu(R_t^{(i)}) - \mu(R_t^{(i-1)})}{|A_M^-|} = d_\mu > 0. \tag{25}$$

Theorem 3. *Assuming $M = \acute{M}$ and the regularity conditions* (R1)–(R4)*, local selection is proper according to* (19)*, with convergence of the limit i.p.*

See Appendix C for a proof.

Independently of the method, performing selection for each t is neither necessary nor necessarily advantageous: both in terms of computational complexity and performance it is usually desirable to select particles only occasionally. No analytical results can be given yet that indicate an adequate strategy for triggering the selection procedure, since the usual estimation of the effective sample size in standard PF [8] cannot be directly translated to CRPF.

4. Application example: Maneuvering target tracking

Consider an object (hereafter the *target*) that emits some radio signal while moving along a two-dimensional space. A region of this space is monitored by a set of sensors that provide noisy measurements of the signal power received at their respective locations. These data are transmitted to a *fusion center*, where they are used to recursively estimate the position and velocity of the target. Below, we present a signal model for this problem, then we describe the algorithms used to track the target and, finally, we show some illustrative simulation results.

4.1. Signal model

The maneuvering target tracking problem can be well represented using a random dynamical model. The system state at time $t = 0, 1, 2, \ldots$ consists of the target position, $\mathbf{r}_t = [r_{1,t}, r_{2,t}]^\top \in \mathbb{R}^2$, and velocity, $\mathbf{v}_t = [v_{1,t}, v_{2,t}]^\top \in \mathbb{R}^2$. Its evolution can be modeled as [21]

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{Q}\mathbf{u}_t, \tag{26}$$

where $\mathbf{x}_t = [\mathbf{r}_t^\top, \mathbf{v}_t^\top]^\top \in \mathbb{R}^4$ is the state vector, $\mathbf{u}_t$ is a two-dimensional zero-mean Gaussian process with covariance matrix $\sigma_u^2\mathbf{I}_2$, i.e., the pdf of $\mathbf{u}_t$ is $N(\mathbf{u}_t|\mathbf{0}, \sigma_u^2\mathbf{I}_2)$ ($\mathbf{I}_k$ denotes the $k \times k$ identity matrix), and the matrices $\mathbf{A}$ and $\mathbf{Q}$ are defined as

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{Q} = \begin{bmatrix} \frac{1}{2}T^2 & 0 \\ 0 & \frac{1}{2}T^2 \\ T & 0 \\ 0 & T \end{bmatrix}, \tag{27}$$

where $T > 0$ is the observation period (i.e., the time between consecutive measurements). The a priori pdf of the state is also Gaussian, namely $p(\mathbf{x}_0) = N(\mathbf{r}_0|\mathbf{0}, 5\mathbf{I}_2) \times N(\mathbf{v}_0|\mathbf{0}, \frac{1}{4}\mathbf{I}_2)$.

Observations are collected through a set of N sensors with known locations, $\mathbf{r}_i \in \mathbb{R}^2$, $i = 1, \ldots, N$, that measure the power of the radio signal received from the target. Assuming the log-normal model widely used in cellular communications [22], the observation at time t in the ith sensor is

$$y_{i,t} = 10\log_{10}\left(\eta + \frac{P_0}{\|\mathbf{r}_t - \mathbf{r}_i\|^\gamma}\right) + g_{i,t}\ (\text{dB}), \quad i = 1, \ldots, N, \tag{28}$$

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

Table 7
Instance of the CRPF algorithm. We use notation $\hat{x}_{k,t}^{(i)}$ for the kth element of $\hat{\mathbf{x}}_t^{(i)}$

1. Initialization:
$E_0 \triangleq \{\mathbf{x} \in \mathbb{R}^4\colon\ x_k \in [-\frac{L}{2}, +\frac{L}{2}], k = 1, \ldots, 4\}$,
$\mathbf{x}_t^{(i)} \sim U(E_0), \quad i = 1, \ldots, M$.
2. Recursive steps: the wps at time t is $\Omega_t = \{\mathbf{x}_{0:t}^{(i)}, C_t^{(i)}\}_{i=1}^M$
Selection:
for $i = 1, \ldots, M$, let $R_{t+1}^{(i)} = \lambda C_t^{(i)} + \|\mathbf{y}_{t+1} - \mathbf{y}(\mathbf{A}\mathbf{x}_t^{(i)})\|$.
Obtain $\hat{\Omega}_t = \{\hat{\mathbf{x}}_{0:t}^{(i)}, \hat{C}_t^{(i)}\}_{i=1}^M$.
Propagation: choose a radius $\rho_{\mathrm{CR}} > 0$.
For $i = 1, \ldots, M$,
$E_{\rho_{\mathrm{CR}}}(\hat{\mathbf{x}}_t^{(i)}) \triangleq \{\mathbf{x} \in \mathbb{R}^4\colon\ x_k \in [\hat{x}_{k,t}^{(i)} - \rho_{\mathrm{CR}}, \hat{x}_{k,t}^{(i)} + \rho_{\mathrm{CR}}]\}$.
$\mathbf{x}_{t+1}^{(i)} \sim U(E_{\rho_{\mathrm{CR}}}(\hat{\mathbf{x}}_t^{(i)}))$.
$\mathbf{x}_{0:t+1}^{(i)} = \{\hat{\mathbf{x}}_{0:t}^{(i)}, \mathbf{x}_{t+1}^{(i)}\}$ and $C_{t+1}^{(i)} = \lambda \hat{C}_t^{(i)} + \|\mathbf{y}_{t+1} - \mathbf{y}(\mathbf{x}_{t+1}^{(i)})\|$.
3. Estimation:
let $\pi_{t+1}^{(i)} \propto \mu(C_{t+1}^{(i)})$, $i = 1, \ldots, M$, be a pmf.
$\mathbf{x}_{t+1}^{\mathrm{mean}} = \sum_{i=1}^M \pi_{t+1}^{(i)} \mathbf{x}_{t+1}^{(i)}$.

where P_0 is the transmitted power, $\gamma > 0$ determines the rate of the (exponential) power decay, $\eta > 0$ yields the sensitivity of the sensor (signals with power less than $10\log_{10}(\eta)$ dB cannot be adequately measured), and $g_{i,t}$ is Gaussian observational noise with pdf $N(g_{i,t}|0, \sigma_g^2)$. We assume the real parameters P_0, γ and η are known, and use $\mathbf{y}_t = [y_{1,t}, \ldots, y_{N,t}]^\top \in \mathbb{R}^N$ to denote the whole collection of observations at time t. The goal is to recursively estimate $\mathbf{x}_t$ from the sequence $\mathbf{y}_{1:t}$.

4.2. Algorithms

We have tested a CRPF algorithm for maneuvering target tracking that results from the following choices and assumptions (see Table 7 for details).

(i) The monitored region is a square of side L (m) centered at the origin. The initial particles are sampled uniformly in this area.

(ii) The cost function is additive,

$$C(\mathbf{x}_{0:t}, \mathbf{y}_{1:t}) = \lambda C(\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t-1}) + \triangle C(\mathbf{x}_t, \mathbf{y}_t) \tag{29}$$

with increments defined as

$$\triangle C\big(\mathbf{x}_t^{(i)}, \mathbf{y}_t\big) \triangleq \big\|\mathbf{y}_t - \mathbf{y}(\mathbf{x}_t^{(i)})\big\|, \tag{30}$$

where $\mathbf{x}_t^{(i)} = \begin{bmatrix} \mathbf{r}_t^{(i)} \\ \mathbf{v}_t^{(i)} \end{bmatrix}$, $\|\mathbf{x} - \mathbf{z}\|$ is the Euclidean distance between vectors $\mathbf{x}$ and $\mathbf{z}$, and $\mathbf{y}(\mathbf{x}_t^{(i)})$ is an $N \times 1$ vector with kth component $y_k(\mathbf{x}_t^{(i)}) = 10\log_{10}(\eta + P_0/\|\mathbf{r}_t^{(i)} - \mathbf{r}_k\|^\gamma)$.

(iii) The risk function is a prediction of the cost,

$$R\big(C_t^{(i)}, \mathbf{x}_t^{(i)}, \mathbf{y}_{t+1}\big) = \lambda C_t^{(i)} + \triangle C\big(\mathbf{A}\mathbf{x}_t^{(i)}, \mathbf{y}_{t+1}\big). \tag{31}$$

(iv) The generating function is [12]

$$\mu(C_t^{(i)}) = \frac{1}{(C_t^{(i)} - \min_{k\in\{1,\ldots,M\}}\{C_t^{(k)}\} + 1/M)^3}, \quad i = 1, \ldots, M. \tag{32}$$

(v) Particles are propagated using a uniform pdf. We have chosen this density to illustrate how the algorithm works when there is no match between the propagation function and the true statistics. The uniform density on a set E is denoted as $U(E)$.

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

Table 8
Auxiliary particle filter (APF)

1. Initialization: the same as in the SBF.

2. Recursive step: given the wps $\{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{M}$,

(a) draw auxiliary indices

$\ell^{(i)} \sim p(\ell) = w_{t-1}^{(\ell)} p(\mathbf{y}_t | h(\mathbf{x}_t^{(\ell)}))$, $i = 1, \ldots, M$,

where $h(\mathbf{x}_{t-1}^{(\ell)})$ is an estimate of $\mathbf{x}_t$ a priori computed from $\mathbf{x}_{t-1}^{(\ell)}$,

(b) draw new particles

$\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(\ell^{(i)})})$, $i = 1, \ldots, M$, and

(c) assign importance weights proportional to the likelihood ratio,

$w_t^{(i)} \propto p(\mathbf{y}_t | \mathbf{x}_t^{(i)}) / p(\mathbf{y}_t | h(\mathbf{x}_{t-1}^{(\ell^{(i)})}))$.

(vi) Selection is performed at each time step. For the simulations, we have considered both global and local selection schemes, according to Tables 5 and 6, respectively, with one offspring per particle (i.e., $\acute{M} = M$).

(vii) A mean estimator of the state, according to (13). The (heuristic) reason for using (13), instead of the straightforward (12), is that the resulting estimate of the target trajectory, $\hat{\mathbf{x}}_{0:t}^{\text{mean}}$, is usually smoother than $\hat{\mathbf{x}}_{0:t}^{\min}$.

For comparison purposes, we have also tackled the same problem with the standard boostrap filter (SBF) [6], the auxiliary particle filter (APF) [23] the sequential importance sampling with resampling (SISR) algorithm [8], and the accelerated random search (ARS) algorithm [18].

The general SBF algorithm was outlined in Table 1. We note that, for the problem at hand, both the state conditional pdf, $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, and the likelihood, $p(\mathbf{y}_t|\mathbf{x}_t)$, are Gaussian (in particular, this implies that particles are drawn from a mixture Gaussian density). The normalized weights are used to approximate the minimum mean square error (MMSE) estimate of the state at time t, i.e., $\hat{\mathbf{x}}_t^{\text{mmse}} = \sum_{i=1}^{M} w_t^{(i)} \mathbf{x}_t^{(i)}$.

The APF algorithm, outlined in Table 8, is a more efficient version of the SBF. An important feature of the APF algorithm is the exploration of the a posteriori probability of the state, given the new observation, before actually generating new particles. It is achieved by sampling the auxiliary indices, denoted $\ell^{(i)}$ in Table 8, according to probabilities that depend on the predictive likelihood, $p(\mathbf{y}_t|h(\mathbf{x}_{t-1}^{(i)}))$, and can be seen as intuitively equivalent to the selection of new particles depending on their risks, as carried out in general CRPF algorithms. However, the procedure for the computation of the costs, and associated probabilities $\pi_t^{(i)}$, in CRPF can be much simpler than the weight calculation in the APF and produce quantitatively very different results. E.g., in a system with Gaussian noise processes, the typical computation of the weight in an APF has the form $w_t^{(i)} \propto \exp\{\|\mathbf{y}_t - f_y(\mathbf{x}_t)\|^2 - \|\mathbf{y}_t - f_y(h(\mathbf{x}_{t-1}))\|^2\}$, which reduces to $\pi_t^{(i)} \propto \mu(\|\mathbf{y}_t - f_y(\mathbf{x}_t)\|^2)$ in a CRPF algorithm (assuming we choose such a quadratic form for the cost function), with freedom to choose function μ. The sampling step is also more flexible in CRPF than in the standard APF.

The SISR algorithm details are given in Table 3. We use it in its simplest form, with the importance function being equal to the state conditional density, i.e., $q(\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$. Thus, the importance weights become $w_t^{(i)} \propto w_{t-1}^{(i)} p(\mathbf{y}_t|\mathbf{x}_{t-1}^{(i)})$. Multinomial resampling is carried out whenever the estimated effective sample size $\hat{M}_{\text{eff}} = 1/\sum_{i=1}^{M} w_t^{(i)2}$ is less than $M/2$ [8]. The MMSE estimate of $\mathbf{x}_t$ is approximated in the same way as with the SBF.

The ARS algorithm is an iterative Monte Carlo optimization method, suitable for searching the global optimum of a cost function in a compact set. Here, we use it to provide a benchmark for the performance of the CRPF technique in minimizing the incremental cost at each time step. The specific algorithm for solving $\min_{\mathbf{x}} \triangle C(\mathbf{x}, \mathbf{y}_t)$ is described in Table 9. For each t, it is initialized with the true target location ($\triangle C(\mathbf{x}_t, \mathbf{y}_t)$ does not depend on the target velocity) and then iterated M times. The algorithm adjustable parameters are the maximum and minimum radii, $\rho_{\max}$ and $\rho_{\min}$, respectively, and the contraction factor $c > 1$. The output is an estimate of the target location, $\mathbf{r}^{[M]}$, that minimizes the incremental cost.

4.3. Numerical results

For the computer simulations, we have assumed that $N = 16$ sensors are uniformly deployed on a square area of side $L = \sqrt{4 \times 10^6}$ m centered at the origin. The background noise power is $\eta = 10^{-7}$ (i.e., the sensitivity is -70 dB),

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

Table 9
Accelerated random search algorithm

1. Initialization:
let $\mathbf{r}^{[0]} = \mathbf{r}_t$, and choose $c > 1$ and $\rho_{\max} > \rho_{\min} > 0$.
Set $\rho^{[1]} = \rho_{\max}$.
2. Iterative step: given $\mathbf{r}^{[n-1]}$:
draw $\acute{\mathbf{r}} \sim U(S(\mathbf{r}^{[n-1]}, \rho^{[n]}))$,
where $S(\mathbf{r}^{[n-1]}, \rho^{[n]}) = \{\mathbf{r} \in \mathbb{R}^2 \colon \|\mathbf{r} - \mathbf{r}^{[n-1]}\| \leqslant \rho^{[n]}\}$.
If $\|\mathbf{y}_t - \mathbf{y}(\acute{\mathbf{r}})\| < \|\mathbf{y}_t - \mathbf{y}(\mathbf{r}^{[n-1]})\|$,
then $\mathbf{r}^{[n]} = \acute{\mathbf{r}}$, $\rho^{[n+1]} = \rho_{\max}$,
else $\mathbf{r}^{[n]} = \acute{\mathbf{r}}^{[n-1]}$, $\rho^{[n+1]} = \rho^{[n]}/c$.
If $\rho^{[n+1]} < \rho_{\min}$ then $\rho^{[n+1]} = \rho_{\max}$.

Table 10
Percentage of successful tracks, averaged over 200 independent simulations (for each value of M)

	SBF	APF	SISR	ARS	CRPF (global)	CRPF (local)
$M = 100$	96.0	97.5	95.5	96.0	98.5	100
$M = 200$	98.5	100	96.0	90.5	98.5	99.0
$M = 400$	99.5	100	100	88.0	100	100

the transmitted power is $P_0 = 1$, the attenuation exponent is $\gamma = 2$, the variance of the power measurements is $\sigma_g^2 = 1$, the observation period is $T = 0.5$ s and the covariance of the state noise is $\mathbf{Q}\mathbf{Q}^\top$ (i.e., $\sigma_u^2 = 1$). The memory factor of the CRPF algorithm is set to $\lambda = 0.9$ and the radius used for uniform propagation of the particles is $\rho_{\mathrm{CR}} = 15$ m. The SBF, the APF and the SISR algorithm are completely specified by the model parameters, and for the ARS procedure we have chosen $\rho_{\max} = 30$ m, $\rho_{\min} = 10^{-4}$ m, and $c = 2$.

We first study the capability of the particle filters to stay locked to the target trajectory. To do this, we define that a track has been successful when mean absolute error (MAE) between the true and estimated trajectories is smaller than 50 m, i.e.,

$$\mathrm{MAE}(\hat{\mathbf{r}}_{t_i:t_f}) = \frac{1}{1 + t_f - t_i} \sum_{t=t_i}^{t_f} \|\hat{\mathbf{r}}_t - \mathbf{r}_t\| < 50 \text{ m}, \tag{33}$$

where $\hat{\mathbf{r}}_{0:t}$ is the trajectory estimate and $[t_i, t_f]$ is the interval where the MAE is calculated (if the target leaves the monitored area at time T_f, we set $[t_i, t_f] = [\lfloor \frac{4}{5} T_f \rfloor, T_f]$, where $\lfloor z \rfloor$ denotes the integer part of the positive real number z). We have run 200 independent simulations (in each simulation we have used the six algorithms, SBF, APF, SISR, ARS, CRPF with global selection and CRPF with local selection, to track the same target) and counted the number of successful tracks for each technique.

Table 10 shows the results obtained when $M = 100$, 200, and 400. It is observed that the five particle filters attain a very similar performance, with success rates between 95.5% (of the SISR algorithm with $M = 100$) and 100%. The two CRPF algorithms, both with global and local selection, are competitive with the conventional particle filters that exploit a full knowledge of the statistics of the state and observation processes. It is interesting to note that the ARS method yields the poorest performance (despite being initialized with the true target position), with a success rate that decreases with the number of iterations. The reason is that the optimization of (30) does not lead to the minimization of the MAE.

Using only the simulation trials that yielded successful tracks, we have computed the average values of the MAE and the mean incremental cost attained by each algorithm for $M = 100$, 200, and 400. The latter magnitude is computed, at time t, as

$$\overline{\triangle C}_t^{\mathrm{PF}} = \sum_{i=1}^{M} \triangle C\left(\mathbf{x}_t^{(i)}, \mathbf{y}_t\right) w_t^{(i)}, \tag{34}$$

$$\overline{\triangle C}_t^{\mathrm{CRPF}} = \sum_{i=1}^{M} \triangle C\left(\mathbf{x}_t^{(i)}, \mathbf{y}_t\right) \frac{\mu(\triangle C_t^{(i)})}{\sum_{k=1}^{M} \mu(\triangle C_t^{(k)})}, \tag{35}$$

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

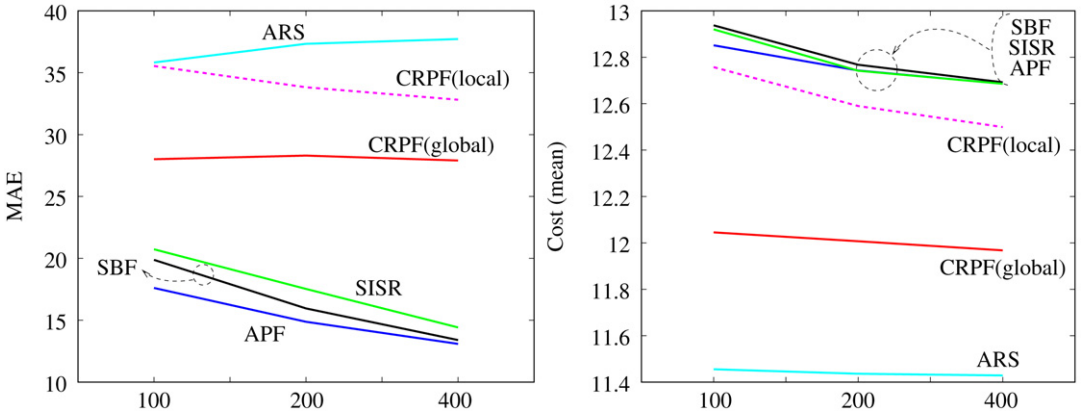

Fig. 1. Left: Average, over the successful tracks only, of the MAE. Right: Average, over the successful tracks only, of the mean cost increments.

for conventional PF algorithms and CRPF methods, respectively. For the ARS techniques, we have computed $\triangle C(\mathbf{r}_t^{[M]}, \mathbf{y}_t)$, i.e., the incremental cost of the last approximation provided by the iterative procedure.

Figure 1 shows the obtained results. The conventional particle filters yield the best performance in terms of MAE, but the worst in terms of the mean incremental cost, $\overline{\triangle C}_t$. On the contrary, the ARS procedure attains very low values of $\overline{\triangle C}_t$ but a very poor MAE (that even grows with M). The CRPF techniques provide a trade off. It is remarkable that, for the algorithm with local selection, both the MAE and the average cost decrease with M. The performance of the CRPF method with global selection is approximately midway between the conventional particle filters and the ARS algorithm in the two plots.

Figure 2 shows the "risk improvement" attained through selection, as predicted by Theorems 2 and 3. Specifically, the plot depicts the difference $\bar{R}_t - \hat{\bar{R}}_t$ (see (20) and (21)) for a sample simulation of the CRPF algorithms with global and local selection. We see that the risk difference is positive for the whole simulation period.

It is also illustrative to study the robustness of the algorithms to mismatches between the assumed dynamical model and the true statistics of the state signal. To do this, we have generated target trajectories according to the model-switching equation

$$\begin{cases} k_t \sim p(k_t|k_{t-1}), \\ \mathbf{x}_t = \mathbf{A}_{k_t}\mathbf{x}_{t-1} + \mathbf{Q}_{k_t}\mathbf{u}_t, \end{cases} \tag{36}$$

where $k_t \in \{1, 2, 3\}$ is the state of a Markov chain with transition probabilities given by (read $\kappa_{ij} = p(k_t = i|k_{t-1} = j)$)

$$\mathbf{K} = \begin{bmatrix} \kappa_{11} = 0.90, & \kappa_{12} = 0.90, & \kappa_{13} = 0.90 \\ \kappa_{21} = 0.01, & \kappa_{22} = 0.01, & \kappa_{23} = 0.09, \\ \kappa_{31} = 0.09, & \kappa_{32} = 0.09, & \kappa_{33} = 0.01 \end{bmatrix}, \tag{37}$$

$$\mathbf{A}_1 = \mathbf{A}_3 = \mathbf{A}, \quad \mathbf{A}_2 = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & \cos(\pi/3) & 0 \\ 0 & 0 & 0 & \sin(\pi/3) \end{bmatrix}, \tag{38}$$

$\mathbf{Q}_1 = \mathbf{Q}_2 = \mathbf{Q}$ and $\mathbf{Q}_3 = \sqrt{20}\mathbf{Q}$.

Table 11 shows the percentages of successful tracks achieved by the tracking algorithms when the target dynamics follow (36). Note that, in this case, the conventional particle filters yield very poor results because the probabilistic assumptions on which they are built do not match the actual statistics of the state process, $\mathbf{x}_t$. The CRPF methods, on the other hand, attain approximately the same proportion of successful tracks as they did without a model mismatch, because they do not rely on specific probabilistic assumptions.

Finally, Fig. 3 shows the average value of the MAE for the different algorithms, computed using only the successful tracks. Even though lost tracks are discarded, the results are fairly different from those of Fig. 1 (left). In particular, the performance of the conventional particle filters degrades significantly and, as a result, the CRPF algorithm with global selection attains a lower MAE than the SBF, APF, and SISR algorithms when $M = 100$ and 200. The performance of CRPF with local selection is approximately the same as in the experiment of Fig. 1 (left).

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

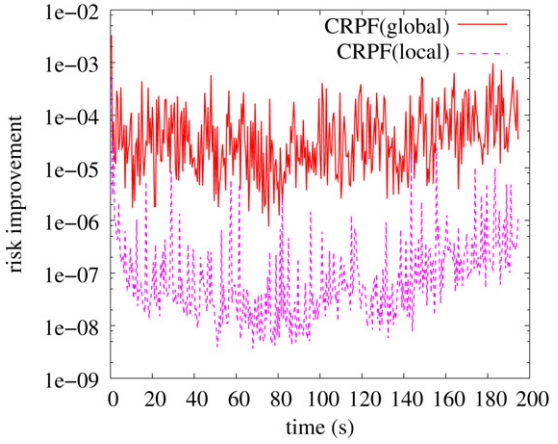

Fig. 2. Difference between the mean risk before propagation and after propagation in the CRPF algorithm with global selection and local selection. $M = 400$ particles.

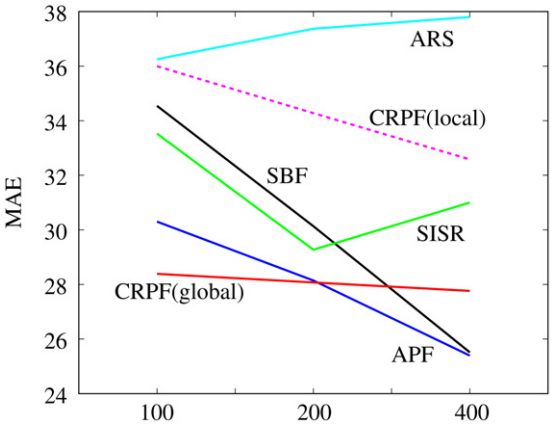

Fig. 3. Average, over the successful tracks only, of the MAE. The state equation used to generate the target trajectories in the simulations was (36), which is mismatched with the probabilistic model used to derive the SBF, APF, and SISR algorithms.

Table 11
Percentage of successful tracks, averaged over 200 independent simulations (for each value of M). The state equation used to generate the target trajectories in the simulations was (36), which is mismatched with the probabilistic model used to derive the SBF, APF and SISR algorithms

	SBF	APF	SISR	ARS	CRPF (global)	CRPF (local)
$M = 100$	26.5	33.5	14.5	96.0	97.0	99.5
$M = 200$	46.5	54.0	29.0	94.0	97.5	100
$M = 400$	47.0	59.0	31.0	92.5	100	100

5. Application example: The Hartman 3 problem

5.1. Problem statement

The Hartman 3 (H3) problem [14] is a typical optimization problem that consists of the minimization of an objective function with a three-dimensional argument and has been recently proposed as part of a benchmark for global optimization algorithms [15]. In order to test the proposed CRPF algorithm, we consider a sequence of H3 problems, specifically (we term the objective function as $\triangle C$ for convenience)

$$\min_{\mathbf{x}_t}\left\{\triangle C(\mathbf{x}_t, \mathbf{y}_t) \triangleq \sum_{i=1}^{4} c_{i,t} \exp\left\{-\sum_{j=1}^{3} a_{ij,t}(x_{j,t} - p_{ij})\right\}\right\}, \quad t = 1, 2, \ldots, \tag{39}$$

where $\mathbf{x}_t = [x_{1,t}, x_{2,t}, x_{3,t}]^\top \in \mathbb{R}^3$, with $0 \leqslant x_j \leqslant 1, \forall j$, $\mathbf{y}_t = [c_{1,t}, \ldots, c_{4,t}, a_{11,t}, \ldots, a_{44,t}]^\top \in \mathbb{R}^{16}$ and the constant p_{ij} is the element in the i-row and jth column of matrix

$$\mathbf{P} = \begin{bmatrix} 0.36890 & 0.11700 & 0.2673 \\ 0.46990 & 0.43870 & 0.7470 \\ 0.10910 & 0.87320 & 0.5547 \\ 0.03815 & 0.57430 & 0.8828 \end{bmatrix}. \tag{40}$$

The time varying parameters collected in $\mathbf{y}_t$ can be decomposed as

$$\mathbf{c}_t = [c_{1,t}, \ldots, c_{4,t}]^\top \quad \text{and} \quad \mathbf{A}_t = \begin{bmatrix} a_{11,t} & \cdots & a_{14,t} \\ \vdots & \ddots & \vdots \\ a_{41,t} & \cdots & a_{44,t} \end{bmatrix},$$

which evolve according to

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

Table 12
Instance of the CRPF algorithm for solving the sequence of H3 problems. It is assumed that the algorithm has no memory, $\lambda = 0$, and that the state transition function is the identity, $f_x(\mathbf{x}) = \mathbf{x}$

1. Initialization: for $i = 1, \ldots, M$
$E_1 \triangleq [0, 1]^3$, $\mathbf{x}_1^{(i)} \sim U(E_1)$, and $C_1^{(i)} = \triangle C(\mathbf{x}_t^{(i)}, \mathbf{y}_1)$.
First solution: $\mathbf{x}_1^{\min} = \mathbf{x}_1^{(i_0)}$ where $i_0 = \arg\min_{k\in\{1,\ldots,M\}} C_1^{(k)}$.
2. Recursive steps: the wps at time t is $\Omega_t = \{\mathbf{x}_t^{(i)}, C_t^{(i)}\}_{i=1}^M$.
Selection:
for $i = 1, \ldots, M$, let $R_{t+1}^{(i)} = \triangle C(\mathbf{x}_t^{(i)}, \mathbf{y}_{t+1})$.
Obtain $\hat{\Omega}_t = \{\hat{\mathbf{x}}_t^{(i)}, \hat{C}_t^{(i)}\}_{i=1}^M$ by local selection.
Propagation: (choose a constant $0 < \rho_{\mathrm{CR}} < 1$)
For $i = 1, \ldots, M$ and $j = 1, 2, 3$,
$E_{\rho_{\mathrm{CR}}}(\hat{x}_{j,t}^{(i)}) \triangleq [\max(0, \hat{x}_{j,t}^{(i)} - \rho_{\mathrm{CR}}), \min(1, \hat{x}_{j,t}^{(i)} + \rho_{\mathrm{CR}})]$.
$x_{j,t+1}^{(i)} \sim U(E_{\rho_{\mathrm{CR}}}(\hat{x}_{j,t}^{(i)}))$.
$C_{t+1}^{(i)} = \triangle C(\mathbf{x}_{t+1}^{(i)}, \mathbf{y}_{t+1})$.
3. Estimation:
tth solution: $\mathbf{x}_t^{\min} = \mathbf{x}_t^{(i_0)}$ where $i_0 = \arg\min_{k\in\{1,\ldots,M\}} C_t^{(k)}$.

$$\mathbf{c}_t = \mathbf{c}_{t-1} + \tilde{\mathbf{c}}, \tag{41}$$

$$\mathbf{A}_t = \mathbf{A}_{t-1} + \tilde{\mathbf{A}}, \tag{42}$$

for $t = 2, 3, \ldots$, with

$$\mathbf{c}_1 = [1.199, 1.001, 3.398, 3.399]^\top, \qquad \tilde{\mathbf{c}} = 10^{-3} \times [-1, 1, -2, -1]^\top, \tag{43}$$

and

$$\mathbf{A}_1 = \begin{bmatrix} 2.602 & 13.980 & 10.100 \\ 0.100 & 13.980 & 54.900 \\ 3.398 & 6.020 & 10.100 \\ 0.100 & 6.020 & 54.900 \end{bmatrix}, \qquad \tilde{\mathbf{A}} = 10^{-1} \times \begin{bmatrix} 0.02 & -0.20 & 1.00 \\ 0.00 & -0.20 & -1.00 \\ -0.02 & 0.20 & 1.00 \\ 0.00 & 0.20 & -1.00 \end{bmatrix}. \tag{44}$$

The above equations yield the exact form of the H3 problem in [15] for $t = 80$.

5.2. Algorithms

Consider the discrete-time dynamical system

$$\mathbf{x}_t = f_x(\mathbf{x}_{t-1}), \tag{45}$$

$$\mathbf{y}_t = f_y(\mathbf{y}_{t-1}), \quad t = 2, 3, \ldots, \tag{46}$$

where f_x is unknown and f_y is defined by (41) and (42) together. If we apply the CRPF algorithm outlined in Table 12 to this system, then we obtain approximate solutions $\triangle C(\mathbf{x}_t^{\min}, \mathbf{y}_t)$, $t = 1, 2, \ldots$, to the sequence of problems (39). For our simulations, the constant radius ρ_{CR}, that appears in the propagation step of Table 12, has been set to $\rho_{\mathrm{CR}} = 0.01$.

For comparison, we have also applied the ARS algorithm to solve the same sequence of problems. For $t = 1$, an initial approximation is chosen uniformly in $[0, 1]^3$, and then the ARS algorithm is applied with maximum radius $\rho_{\max} = 1$, minimum radius $\rho_{\min} = 10^{-7}$ and contraction factor $c = 2$. For $t > 1$, the same parameters are used but the initial approximation is the last solution obtained at time $t - 1$. For each t, the algorithm is iterated M times.

5.3. Numerical results

We have run the CRPF and ARS algorithms 20 times independently (with random initializations) for the sequence of H3 problems from $t = 1$ to 80, both with $M = 1000$ and 5000. Recall that M denotes the number of particles drawn at each time t by the CRPF method and the number of iterations of the ARS technique (therefore, both algorithms

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

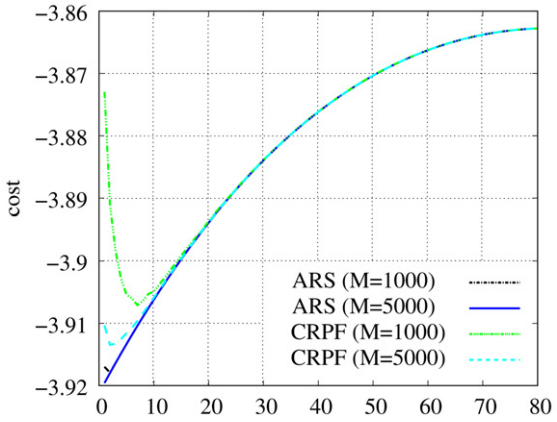

Fig. 4. Value of the cost function $\triangle C(\mathbf{x}_t, \mathbf{y}_t)$, defined by (39), at the minimizers output by the ARS and CRPF algorithms with $M = 1000$ and 5000.

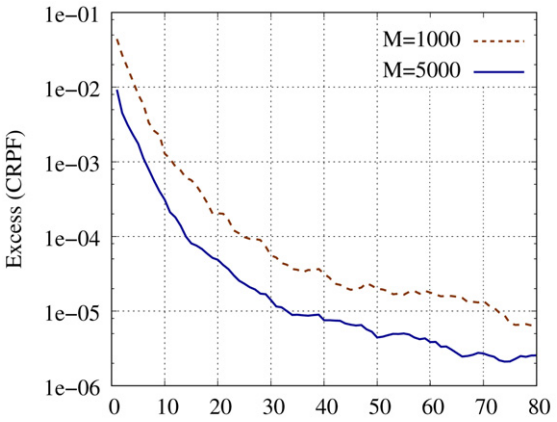

Fig. 5. Excess value of the cost function, computed according to (49), at the minimizers output by the CRPF algorithm with $M = 1000$ and 5000 particles compared with the approximations provided by the ARS technique with $M = 1000$ and 5000 iterations, respectively.

draw M three-dimensional random vectors for each t). Let us denote as $\hat{\mathbf{x}}_{1:80}^{\text{crpf}}(k, M)$ and $\hat{\mathbf{x}}_{1:80}^{\text{ars}}(k, M)$ the sequences of minimizers output by the CRPF and ARS algorithms, respectively, as a result of the kth simulation with M particles.

Figure 4 shows the average value of the cost functions at the minimizers for both algorithms, i.e.,

$$\overline{\triangle C}_{\text{CRPF}}(t, M) = \frac{1}{20}\sum_{k=1}^{20} \triangle C\big(\hat{\mathbf{x}}_t^{\text{crpf}}(k, M), \mathbf{y}_t\big), \tag{47}$$

$$\overline{\triangle C}_{\text{ARS}}(t, M) = \frac{1}{20}\sum_{k=1}^{20} \triangle C\big(\hat{\mathbf{x}}_t^{\text{ars}}(k, M), \mathbf{y}_t\big), \tag{48}$$

for $t = 1, 2, \ldots, 80$ and $M = 1000, 5000$. The curves of the two ARS procedures differ significantly only for $t = 1$, and provide a good benchmark for the CRPF techniques. In particular, we observe that the latter need a convergence period (of 10 to 15 time steps) before they produce solutions that overlap with those of the ARS techniques. After this period, the approximate minima are graphically indistinguishable from those computed with the iterative methods. We also observe a clear improvement (for small t) of the solutions provided by the CRPF algorithm with $M = 5000$ compared to those obtained with only $M = 1000$ particles.

Further insight on the performance of the CRPF algorithm can be drawn by looking at the average "excess" value of $\triangle C(\hat{\mathbf{x}}_t^{\text{crpf}}(k, M), \mathbf{y}_t)$ compared to $\triangle C(\hat{\mathbf{x}}_t^{\text{ars}}(k, M), \mathbf{y}_t)$, i.e.,

$$e(t, M) = \frac{1}{20}\sum_{k=1}^{20} \triangle C\big(\hat{\mathbf{x}}_t^{\text{crpf}}(k, M), \mathbf{y}_t\big) - \triangle C\big(\hat{\mathbf{x}}_t^{\text{ars}}(k, M), \mathbf{y}_t\big), \quad t = 1, \ldots, 80. \tag{49}$$

Figure 5 shows curve (49) for $M = 1000$ and 5000. We observe that the difference between the minima approximated by the CRPF and ARS algorithms is small, and clearly decreasing with t. A significant improvement in the solutions is obtained when increasing the number of particles from $M = 1000$ to 5000.

The computer simulations have shown that a more accurate solution of the sequence of problems (39) is achieved using the ARS algorithm. We should be aware, however, that the ARS procedure is inherently iterative and cannot be parallelized (the nth approximation of the minimizer at time t requires a comparison with the $(n-1)$th approximation, hence the M approximations are computed in a strictly serial fashion). Thus, it is an appealing methodology for batch optimization, but it is not well suited for online applications with stringent time constraints. On the other hand, the CRPF algorithm is particularly suitable for online optimization because the M candidate approximations at time t can be drawn in parallel. Also, their propagation to time $t + 1$ requires a minimal amount of interaction if the local selection procedure is employed (as it has been done in our simulations).

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

6. Conclusions

We have reviewed the recently proposed CRPF methodology and introduced some generalizations that enable the derivation of the most common conventional SMC algorithms (namely, the standard *bootstrap* filter and the sequential importance sampling-resampling algorithm) within the new framework, as well as a simple analysis of convergence of the particle propagation step. Then, we have investigated the issue of particle selection. As a result, we have introduced the notion of proper selection for CRPF algorithms and analyzed a new procedure, termed local selection, that is suitable for implementation with parallel computing devices. In this way, CRPF methods can overcome one of the major practical limitations of conventional particle filters, namely the difficulty for the parallelization of resampling algorithms.

Finally, we have demonstrated the performance of CRPF algorithms by way of two application examples: the tracking of a maneuvering target using a sensor network and a dynamic optimization problem derived from the standard Hartman 3 (H3) problem. We have shown that the CRPF algorithms can perform the target tracking task effectively and are more robust to model mismatches than conventional particle filters. For the dynamic H3 optimization problem, the CRPF has been shown to provide a fast online algorithm with a performance competitive with the accelerated random search technique.

Appendix A. Proof of Theorem 1

Consider some number $\delta > 0$ and the sequence of Bernoulli random variables $\theta^{(m)} \in \{1, 0\}$, $m = 1, \ldots, M$, defined as

- $\theta^{(m)} = 1$, if $\triangle C_t^{(m)} \in I_t(\delta)$,
- $\theta^{(m)} = 0$, otherwise.

Obviously,

$$\sum_{m=1}^{M} \theta^{(m)} = \left| I_t^M(\delta) \right| \tag{A.1}$$

is the number of particles with an incremental cost less than $\triangle C_t^* + \delta$.

The "probability of success" in each Bernoulli trial is

$$p_\theta = \text{probability}\{\theta^{(m)} = 1\} = \text{probability}\left\{\triangle C_t^{(i)} \in I_t(\delta)\right\} \tag{A.2}$$

for arbitrary i. Using the overall pdf that generates the particles through the selection and propagation steps, p_t^s, we can write

$$p_\theta = \int_{\mathcal{X}_t(\delta)} p_t^s\left(\mathbf{x} \middle| \left\{\hat{\mathbf{x}}_{t-1}^{(k)}\right\}_{k=1}^{M}\right) d\mathbf{x} > 0, \tag{A.3}$$

where

$$\mathcal{X}_t(\delta) \triangleq \left\{\mathbf{x} \in \mathcal{X}_t \colon \triangle C(\mathbf{x}, \mathbf{y}_t) \in I_t(\delta)\right\}. \tag{A.4}$$

Note that $\mathcal{X}_t(\delta) \neq \emptyset$ because of the continuity of $\triangle C$ at $\mathbf{x}_t^*$ and $p_\theta > 0$ because of the definition of $\mathcal{X}_t$. If we choose any $\mathbf{x}' \in \mathcal{X}_t(\delta)$, with pdf $p' = p_t^s(\mathbf{x}'|\{\hat{\mathbf{x}}_t^{(k)}\}_{k=1}^M)$ ($p' > 0$ exists [24, Corollary 4.10]) and δ sufficiently small, then $p_\theta = \delta p'$.

Since the mean and variance of $\theta^{(m)}$ are $E[\theta^{(m)}] = p_\theta$ and $\text{Var}[\theta^{(m)}] = p_\theta(1 - p_\theta) < \infty$, we can use the strong law of large numbers [25] to obtain

$$\lim_{M\to\infty} \frac{1}{M} \sum_{i=1}^{M} \theta^{(m)} = p_\theta \quad \text{(a.s.)} \tag{A.5}$$

and, as a consequence,

$$\lim_{M\to\infty} \sum_{i=1}^{M} \theta^{(m)} = \lim_{M\to\infty} M p_\theta. \tag{A.6}$$

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

If we now let δ depend on the number of particles, M, as $\delta_M \geqslant M^{-\psi}$, $0 < \psi < 1$, we arrive at

$$\lim_{M\to\infty} Mp_\theta = \lim_{M\to\infty} M\delta_M p' \geqslant \lim_{M\to\infty} M^{1-\psi} p' = +\infty, \tag{A.7}$$

and, considering (A.1), (A.6), and (A.7) together,

$$\lim_{M\to\infty} \left| I_t^M(\delta_M) \right| = \infty \quad \text{(a.s.).} \qquad \square \tag{A.8}$$

Appendix B. Proof of Theorem 2

Let us consider the initial wps $\Omega_t = \{\mathbf{x}_{0:t}^{(i)}, R_t^{(i)}\}_{i=1}^{\acute{M}}$, with associated pmf $\varsigma_t^{(i)} \propto \mu(R_t^{(i)})$, and the selected wps $\hat{\Omega}_t = \{\hat{\mathbf{x}}_{0:t}^{(i)}, \hat{R}_t^{(i)}\}_{i=1}^{M}$, with associated pmf $\hat{\varsigma}_t^{(i)} \propto \mu(\hat{R}_t^{(i)})$. $\hat{\Omega}_t$ is obtained by sampling the pmf $\{\varsigma_t^{(i)}\}_{i=1}^{\acute{M}}$ M times. The goal is to prove that

$$\lim_{M\to\infty} \bar{R}_t - \bar{\hat{R}}_t = \lim_{M\to\infty} \sum_{i=1}^{\acute{M}} \varsigma_t^{(i)} R_t^{(i)} - \sum_{l=1}^{M} \hat{\varsigma}_t^{(l)} \hat{R}_t^{(l)} \geqslant 0 \quad \text{(i.p.).} \tag{B.1}$$

Let us construct the noninvertible mapping $\alpha : \{1, \ldots, M\} \to \{1, \ldots, \acute{M}\}$ such that $\alpha(j) = i \Leftrightarrow \hat{\mathbf{x}}_{0:t}^{(j)} = \mathbf{x}_{0:t}^{(i)}$. Using this mapping, we further define sets of the form $A_i \triangleq \{j \in \{1, \ldots, M\}\colon \alpha(j) = i\}$, $i = 1, \ldots, \acute{M}$. We observe that $|A_i|$ is the number of offsprings due to particle $\mathbf{x}_{0:t}^{(i)}$.

Using the above notation, we conveniently rewrite $\bar{\hat{R}}_t$ in terms of the original risks,

$$\bar{\hat{R}}_t = \sum_{l=1}^{M} \hat{\varsigma}_t^{(l)} \hat{R}_t^{(l)} = \sum_{i=1}^{\acute{M}} \sum_{j\in A_i} \hat{\varsigma}_t^{(j)} \hat{R}_t^{(j)} = \sum_{i=1}^{\acute{M}} R_t^{(i)} \sum_{j\in A_i} \hat{\varsigma}_t^{(j)} \tag{B.2}$$

$$= \sum_{i=1}^{\acute{M}} R_t^{(i)} \sum_{j\in A_i} \frac{\mu(\hat{R}_t^{(j)})}{\sum_{k=1}^{M} \mu(\hat{R}_t^{(k)})} = \sum_{i=1}^{\acute{M}} R_t^{(i)} \mu(R_t^{(i)}) \frac{|A_i|}{\sum_{k=1}^{M} \mu(\hat{R}_t^{(k)})} \tag{B.3}$$

$$= \sum_{i=1}^{\acute{M}} R_t^{(i)} \mu(R_t^{(i)}) \frac{|A_i|}{\sum_{k=1}^{\acute{M}} |A_k| \mu(R_t^{(k)})}. \tag{B.4}$$

In order to relate (B.4) to $\bar{R}_t$ we need to bring the original pmf into the equality. With this aim, let us divide and multiply by M in (B.4),

$$\bar{\hat{R}}_t = \sum_{i=1}^{\acute{M}} R_t^{(i)} \frac{|A_i|}{M} \frac{\mu(R_t^{(i)})}{\sum_{k=1}^{\acute{M}} \frac{|A_k|}{M} \mu(R_t^{(k)})}. \tag{B.5}$$

According to the weak law of large numbers [16], $\lim_{M\to\infty} |A_i|/M = \varsigma_t^{(i)}$ i.p., hence we write

$$\lim_{M\to\infty} \bar{\hat{R}}_t = \sum_{i=1}^{\acute{M}} R_t^{(i)} \varsigma_t^{(i)} \eta_t^{(i)} \quad \text{(i.p.)}, \tag{B.6}$$

where $\eta_t^{(i)} = \mu(R_t^{(i)}) / \sum_{k=1}^{\acute{M}} \varsigma_t^{(k)} \mu(R_t^{(k)})$. We further observe that $\kappa_t^{(i)} = \varsigma_t^{(i)} \eta_t^{(i)}$, $i = 1, \ldots, \acute{M}$, is a pmf, because $\kappa_t^{(i)} \geqslant 0$,

$$\sum_{i=1}^{\acute{M}} \kappa_t^{(i)} = \sum_{i=1}^{\acute{M}} \varsigma_t^{(i)} \eta_t^{(i)} = \sum_{i=1}^{\acute{M}} \frac{\mu(R_t^{(i)})}{\sum_{k=1}^{\acute{M}} \mu(R_t^{(k)})} \frac{\mu(R_t^{(i)})}{\sum_{k=1}^{\acute{M}} \varsigma_t^{(k)} \mu(R_t^{(k)})} \tag{B.7}$$

and substituting $\varsigma_t^{(k)} = \mu(R_t^{(k)}) / \sum_{l=1}^{\acute{M}} \mu(R_t^{(l)})$ into (B.7) immediately yields $\sum_{i=1}^{\acute{M}} \kappa_t^{(i)} = 1$.

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

A closer look at $\kappa_t^{(i)}$ brings the proof to a conclusion. Indeed, given two particles $\mathbf{x}_{0:t}^{(i)}$ and $\mathbf{x}_{0:t}^{(j)}$ such that $R_t^{(i)} \leqslant R_t^{(j)}$ it follows that $\mu(R_t^{(i)}) \geqslant \mu(R_t^{(j)})$ and, as a consequence, $\eta_t^{(i)} \geqslant \eta_t^{(j)}$. A comparison of the probability mass ratio under the original probability measure, ς_t, and the corrected probability measure, κ_t, yields

$$\frac{\kappa_t^{(i)}}{\kappa_t^{(j)}} = \frac{\varsigma_t^{(i)}\eta_t^{(i)}}{\varsigma_t^{(j)}\eta_t^{(j)}} \geqslant \frac{\varsigma_t^{(i)}}{\varsigma_t^{(j)}} \tag{B.8}$$

and, as a consequence of (B.6), (B.8) and the fact that both $\kappa_t^{(i)}$ and $\varsigma_t^{(i)}$ are pmf's, we obtain

$$\lim_{M\to\infty} \bar{\hat{R}}_t = \lim_{M\to\infty} \sum_{i=1}^{M} \hat{R}_t^{(i)} \hat{\varsigma}_t^{(i)} = \sum_{i=1}^{\acute{M}} R_t^{(i)} \kappa_t^{(i)} \leqslant \sum_{i=1}^{\acute{M}} R_t^{(i)} \varsigma_t^{(i)} = \bar{R}_t \quad \text{(i.p.).} \qquad \square \tag{B.9}$$

Appendix C. Proof of Theorem 3

Let us construct the discrete-time random process

$$z_0 = 0, \qquad z_k = z_{k-1} + \left(R_t^{(k)} - \hat{R}_t^{(k)}\right), \quad k \in \mathbb{N}. \tag{C.1}$$

It is readily seen that $\lim_{M\to\infty} \sum_{i=1}^{M} R_t^{(i)} - \hat{R}_t^{(i)} = \lim_{k\to\infty} z_k$. Furthermore, we can decompose process $\{z_k\}_{k\in\mathbb{N}}$ into two sub-processes $\{z_k^+\}_{k\in\mathbb{N}}$ and $\{z_k^-\}_{k\in\mathbb{N}}$, such that $\lim_{k\to\infty} z_k = \lim_{k\to\infty} z_k^+ - z_k^-$, if we let

$$z_0^+ = 0, \qquad z_0^- = 0, \tag{C.2}$$

$$z_k^+ = z_{k-1}^+ + \left|R_t^{(l_k^+)} - R_t^{(l_k^+-1)}\right| \xi^{(k)^+}, \tag{C.3}$$

$$z_k^- = z_{k-1}^- + \left|R_t^{(l_k^+)} - R_t^{(l_k^+-1)}\right| \xi^{(k)^-}, \tag{C.4}$$

where l_k^+ (correspondingly, l_k^-) is the kth index in A_M^+ (correspondingly, A_M^-), while $\xi^{(k)^+}$ and $\xi^{(k)^-}$ are Bernoulli random variables (r.v.'s) with parameters

$$\phi^{(k)^+} = \frac{\mu(R_t^{(l_k^+-1)})}{\mu(R_t^{(l_k^+-1)}) + \mu(R_t^{(l_k^+)})}, \qquad \phi^{(k)^-} = \frac{\mu(R_t^{(l_k^--1)})}{\mu(R_t^{(l_k^--1)}) + \mu(R_t^{(l_k^-)})}, \tag{C.5}$$

respectively. We note that, by construction of sets A_M^+ and A_M^-, $\phi^{(k)^+} > 1/2$ and $\phi^{(k)^-} < 1/2$ for all k. If we take

$$\phi^+ = \min_{k\in\mathbb{N}}\{\phi^{(k)^+}\}, \qquad \phi^- = \max_{k\in\mathbb{N}}\{\phi^{(k)^-}\}, \tag{C.6}$$

then we can construct the auxiliary processes

$$s_0^+ = 0, \qquad s_0^- \triangleq 0, \tag{C.7}$$

$$s_k^+ = s_{k-1}^+ + \left|R_t^{(l_k^+)} - R_t^{(l_k^+-1)}\right| \zeta^{(k)^+}, \tag{C.8}$$

$$s_k^- = s_{k-1}^- + \left|R_t^{(l_k^+)} - R_t^{(l_k^+-1)}\right| \zeta^{(k)^-}, \tag{C.9}$$

where $\{\zeta^{(k)^+}\}_{k\in\mathbb{N}}$ are i.i.d. Bernoulli r.v.'s with parameter ϕ^+ and $\{\zeta^{(k)^-}\}_{k\in\mathbb{N}}$ are i.i.d. Bernoulli r.v.'s with parameter ϕ^-. Since $1/2 > \phi^- \geqslant \phi^{(k)^-}$ and $1/2 < \phi^+ \leqslant \phi^{(k)^+}$ for all k, and assuming (R1), (R2), and (R3), it follows that $\lim_{k\to\infty} z_k^+ > \lim_{k\to\infty} s_k^+$ and $\lim_{k\to\infty} z_k^- < \lim_{k\to\infty} s_k^-$ i.p. and, as a consequence,

$$\lim_{k\to\infty} z_k > \lim_{k\to\infty} s_k \quad \text{(i.p.).} \tag{C.10}$$

Moreover, by construction of process $\{s_k\}$, and using (R2) and (R3), we can calculate

$$\lim_{k\to\infty} s_k = d_R(\phi^+ - \phi^-) \lim_{M\to\infty} \left|A_M^+\right| = +\infty \quad \text{(i.p.),} \tag{C.11}$$

hence[2]

[2] Statements of the type $\lim_{n\to\infty} a(n) = \infty$ (i.p.) should be read as $\lim_{n\to\infty} \text{probability}\{a(n) \leqslant B\} = 0$ for any finite constant B.

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

$$\lim_{M\to\infty}\sum_{i=1}^{M} R_t^{(i)} - \hat{R}_t^{(i)} = \lim_{k\to\infty} z_k = +\infty \quad \text{(i.p.)}. \tag{C.12}$$

Assumption (R4) enables us to prove that

$$\lim_{M\to\infty}\sum_{l=1}^{M} \mu(R_t^{(l)}) - \mu(\hat{R}_t^{(l)}) = -\infty \quad \text{(i.p.)} \tag{C.13}$$

using a similar procedure.

In the sequel we use (C.12) and (C.13) to complete to proof. Consider two sets of indices,

$$K_0 \triangleq \{k_1, k_2, \ldots, k_n\}, \quad n < M, \tag{C.14}$$

$$K_1 \triangleq \{k_{n+1}, k_{n+2}, \ldots, k_{n+M}\}, \tag{C.15}$$

such that $K_0 \cup K_1 = \{1, 2, \ldots, M\}$, $K_0 \cap K_1 = \emptyset$ and, $\forall i \in K_1$, $R_t^{(i)} = \hat{R}_t^{(i)}$. For an arbitrary set of indices K, let us further define

$$\bar{R}_t(K) \triangleq \sum_{i\in K} R_t^{(i)} \frac{\mu(R_t^{(i)})}{\sum_{l\in K}\mu(R_t^{(l)})}, \tag{C.16}$$

$$p(K) \triangleq \frac{\sum_{i\in K}\mu(R_t^{(i)})}{\sum_{l=1}^{M}\mu(R_t^{(l)})}, \tag{C.17}$$

as well as $\bar{\hat{R}}_t(K)$ and $\hat{p}(K)$, obviously substituting $R_t^{(\cdot)}$ by $\hat{R}_t^{(\cdot)}$ in the equations above. It is apparent from the definition of K_1 that $\bar{R}_t(K_1) = \bar{\hat{R}}_t(K_1)$. Hence, using the notation just introduced, we can write

$$\bar{R}_t - \bar{\hat{R}}_t = p(K_0)\bar{R}_t(K_0) - \hat{p}(K_0)\bar{\hat{R}}_t(K_0) + \big[p(K_1) - \hat{p}(K_1)\big]\bar{R}_t(K_1). \tag{C.18}$$

On the basis of (C.12), (C.13) and the regularity assumption (R1) and (R2), for sufficiently large n and M we can choose sets K_0 and K_1 such that $\bar{R}_t(K_0) - \bar{\hat{R}}_t(K_0) \geqslant 0$ and $\bar{R}_t(K_1) - \bar{\hat{R}}_t(K_0) \geqslant 0$. Also, we have $p(K_1) \geqslant \hat{p}(K_1)$ (because (C.12) ensures $\sum_{l=1}^{M} R_t^{(l)} \geqslant \sum_{l=1}^{M} \hat{R}_t^{(l)}$ for sufficiently large M) and, as a consequence, $\hat{p}(K_0) \geqslant p(K_0)$. Also note that $p(K_1) - \hat{p}(K_1) = \hat{p}(K_0) - p(K_0)$. Therefore,

$$\bar{R}_t - \bar{\hat{R}}_t = p(K_0)\bar{R}_t(K_0) - \hat{p}(K_0)\bar{\hat{R}}_t(K_0) + \big[\hat{p}(K_0) - p(K_0)\big]\bar{R}_t(K_1) \tag{C.19}$$

$$\geqslant \big[p(K_0) - \hat{p}(K_0)\big]\bar{\hat{R}}_t(K_0) + \big[\hat{p}(K_0) - p(K_0)\big]\bar{R}_t(K_1) \tag{C.20}$$

$$= \big[\hat{p}(K_0) - p(K_0)\big]\big[\bar{R}_t(K_1) - \bar{\hat{R}}_t(K_0)\big] \geqslant 0 \tag{C.21}$$

for sufficiently large M (with convergence i.p.). □

References

[1] J. Durbin, S.J. Koopman (Eds.), Time Series Analysis by State Space Methods, Oxford Univ. Press, Oxford, UK, 2001.
[2] S. Haykin, Adaptive Filter Theory, fourth ed., Information and System Sciences Series, Prentice Hall, Englewood Cliffs, NJ, 2001.
[3] R.E. Kalman, A new approach to linear filtering and prediction problems, J. Basic Eng. 82 (1960) 35–45.
[4] G. Kitagawa, Non-Gaussian state-space modeling of nonstationary time series, J. Amer. Statist. Assoc. 82 (400) (1987) 1032–1063.
[5] E. Bolviken, G. Storvik, Deterministic and stochastic particle filters in state-space models, in: A. Doucet, N. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer-Verlag, New York, 2001, pp. 97–116, ch. 5.
[6] N. Gordon, D. Salmond, A.F.M. Smith, Novel approach to nonlinear and non-Gaussian Bayesian state estimation, IEE Proc. F 140 (1993) 107–113.
[7] J.S. Liu, R. Chen, Sequential Monte Carlo methods for dynamic systems, J. Amer. Statist. Assoc. 93 (443) (1998) 1032–1044.
[8] A. Doucet, S. Godsill, C. Andrieu, On sequential Monte Carlo Sampling methods for Bayesian filtering, Statist. Comput. 10 (3) (2000) 197–208.
[9] P.M. Djurić, J.H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M.F. Bugallo, J. Míguez, Particle filtering, IEEE Signal Process. Mag. 20 (5) (2003) 19–38.
[10] D. Crisan, A. Doucet, A survey of convergence results on particle filtering, IEEE Trans. Signal Process. 50 (3) (2002) 736–746.

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003

[11] M. Bolić, P.M. Djurić, S. Hong, Resampling algorithms and architectures for distributed particle filters, IEEE Trans. Signal Process. 53 (7) (2005) 2442–2450.
[12] J. Míguez, M.F. Bugallo, P.M. Djurić, A new class of particle filters for random dynamical systems with unknown statistics, EURASIP J. Appl. Signal Process. 2004 (15) (2004) 2278–2294.
[13] B. Ristic, S. Arulampalam, N. Gordon, Beyond the Kalman Filter, Artech House, Boston, 2004.
[14] L. Dixon, G. Szegö (Eds.), Towards Global Optimization, vol. 2, North-Holland, New York, 1978.
[15] M.M. Ali, C. Khompatraporn, Z.B. Zabinsky, A numerical evaluation of several stochastic algorithms on selected continuous global optimization problems, J. Global Opt. 31 (2005) 635–672.
[16] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley–Interscience, New York, 1991.
[17] A. Doucet, N. de Freitas, N. Gordon, An introduction to sequential Monte Carlo methods, in: A. Doucet, N. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer-Verlag, New York, 2001, pp. 4–14, ch. 1.
[18] M.J. Appel, R. Labarre, D. Radulovic, On accelerated random search, SIAM J. Opt. 14 (3) (2003) 708–730.
[19] J. Carpenter, P. Clifford, P. Fearnhead, Improved particle filter for nonlinear problems, IEE Proc. Radar Sonar Navigat. 146 (1) (1999) 2–7.
[20] R. Douc, O. Cappé, E. Moulines, Comparison of resampling schemes for particle filtering, in: Proceedings of the Fourth International Symposium on Image and Signal Processing and Analysis, 2005.
[21] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, P.-J. Nordlund, Particle filters for positioning, navigation and tracking, IEEE Trans. Signal Process. 50 (2) (2002) 425–437.
[22] T.S. Rappaport, Wireless Communications: Principles and Practice, second ed., Prentice Hall, Upper Saddle River, NJ, 2001.
[23] M.K. Pitt, N. Shephard, Auxiliary variable based particle filters, in: A. Doucet, N. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer-Verlag, New York, 2001, pp. 273–293, ch. 13.
[24] R.G. Bartle, The Elements of Integration and Lebesgue Measure, Wiley, New York, 1995.
[25] H. Stark, J.W. Woods, Probability and Random Processes with Applications to Signal Processing, Prentice Hall, Upper Saddle River, NJ, 2002.

Joaquín Míguez was born in Ferrol, Galicia, Spain, in 1974. He obtained the Licenciado en Informatica (M.Sc.) and Doctor en Informatica (Ph.D.) degrees from Universidade da Coruña, Spain, in 1997 and 2000, respectively. Late in 2000, he joined Departamento de Electrónica e Sistemas, Universidade da Coruña, where he became an Associate Professor in July 2003. From April 2001 to December 2001, he was a visiting scholar in the Department of Electrical and Computer Engineering, Stony Brook University. In September 2004, he moved to Departamento de Teorìa de la Señal y Comunicaciones, Universidad Carlos III de Madrid, also as an Associate Professor. His research interests are in the field of statistical signal processing, with emphasis in the topics of Bayesian analysis, sequential Monte Carlo methods, adaptive filtering, stochastic optimization, and their applications to communications (multiuser and smart antenna systems), sensor networks, target tracking, and vehicle positioning/navigation.

Please cite this article as: J. Míguez, Analysis of selection methods for cost-reference particle filtering with applications to maneuvering target tracking and dynamic optimization, Digital Signal Process. (2006), doi:10.1016/j.dsp.2006.09.003