Analysis of a sequential Monte Carlo method for optimization in dynamical systems

Joaquín Míguez

Department of Signal Theory & Communications, Universidad Carlos III de Madrid, Avenida de la Universidad 30, 28911 Leganés, Madrid, Spain

A R T I C L E   I N F O

Article history:
Received 16 April 2009
Received in revised form 31 August 2009
Accepted 3 November 2009
Available online 12 November 2009

Keywords:
Sequential Monte Carlo
Stochastic optimization
Nonlinear dynamics
Nonlinear tracking
Dynamic optimization

A B S T R A C T

We investigate a recently proposed sequential Monte Carlo methodology for recursively tracking the minima of a cost function that evolves with time. These methods, subsequently referred to as sequential Monte Carlo minimization (SMCM) procedures, have an algorithmic structure similar to particle filters: they involve the generation of random paths in the space of the signal of interest (SoI), the stochastic selection of the fittest paths and the ranking of the survivors according to their cost. In this paper, we propose an extension of the original SMCM methodology (that makes it applicable to a broader class of cost functions) and introduce an asymptotic-convergence analysis. Our analytical results are based on simple induction arguments and show how the SoI-estimates computed by a SMCM algorithm converge, in probability, to a sequence of minimizers of the cost function. We illustrate these results by means of two computer simulation examples.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

We address the extension and analysis of a recently proposed family of algorithms termed “cost reference particle filters” (CRPFs) that can be used for nonlinear tracking in dynamical systems. The original method was proposed in [21,22], with some theoretical developments in [19] and applications to target tracking in [20,3,9].

Although the new methodology was introduced as an algorithmic generalization of conventional particle filters (and, indeed, the standard particle filtering techniques can be interpreted as CRPFs from a purely procedural viewpoint [19]), a CRPF can be plainly described as a sequential Monte Carlo algorithm that tracks the minima of a cost function that evolves with time. To be precise, assume that we wish to sequentially estimate a discrete-time random signal of interest (SoI), \( \{x_t\}_{t \in \mathbb{N}} \), from a series of associated observations, \( \{y_t\}_{t \in \mathbb{N}} \). The estimation criterion is the minimization of a given cost function that depends on both sequences, denoted \( C_t(x_1, \ldots, x_t, y_1, \ldots, y_t) \), i.e., our desired estimate of the SoI is the sequence \( \hat{x}_1, \ldots, \hat{x}_t \) that minimizes \( C_t \) for the observed data. A CRPF generates random paths in the space of \( \{x_t\}_{t \in \mathbb{N}} \), then uses a stochastic mechanism to select the most promising ones (and discard the others) and, finally, ranks the surviving paths according to their cost. The resemblance to particle filters is apparent, but the ultimate goal of the algorithm is the minimization of the prescribed cost and, for this reason, in this paper we adopt the more accurate term “sequential Monte Carlo minimization” (SMCM) for procedures of this type.

The main contribution of this paper is the analysis of the convergence of SMCM methods, which was missing in [21,19]. We prove, using simple induction arguments, that an adequately constructed SMCM algorithm yields estimates of the SoI that converge asymptotically (with time and the number of samples in the signal space), and in probability, to a certain sequence of minimizers of the prescribed cost. Additionally, we provide a description of the SMCM class of algorithms that is concise but more general than the statements in [21,19]. Namely, it...
2. Notation

We write boldface lower-case letters for column vectors, e.g., \( \mathbf{x} \). Sets are denoted with upper-case calligraphic letters, e.g., \( \mathcal{X} \), while lower-case greek letters are used to represent probability density functions (pdf’s) or probability mass functions (pmf’s), e.g., \( \pi \). The joint pdf/pmf, \( \pi \), of two random vectors \( \mathbf{x} \) and \( \mathbf{y} \) is written as \( \pi(\mathbf{x}, \mathbf{y}) \), and the corresponding conditional pdf/pmf is denoted as \( \pi(\mathbf{x}|\mathbf{y}) \). A random sample, \( \mathbf{x}^{(k)} \), drawn from the distribution characterized by the pdf/pmf \( \pi \), is denoted as \( \mathbf{x}^{(k)} \sim \pi(\mathbf{x}) \).

Probability measures are denoted as \( \mathbb{P} \). We may write \( \mathbb{P}_\mathcal{A} \) to make explicit that the associated pdf/pmf is \( \mathcal{A} \). For sequences indexed by discrete time \( t \in \mathbb{N} \), we write \( \mathbb{P}_t \), implying that the associated pdf/pmf is \( \pi_t \). The Borel \( \sigma \)-algebra in the \( n \)-dimensional space \( \mathbb{R}^n \) is denoted as \( \mathcal{B}^n \), while the Lebesgue measure of a measurable set \( \mathcal{A} \subset \mathcal{B}^n \) is distinctly denoted as \( \ell(\mathcal{A}) \).

3. Sequential Monte Carlo minimization

3.1. Cost functions

We are interested in tracking the state of a dynamical system that generates a sequence of vector-valued observations (often called “measurements”) \( \{\mathbf{y}_t \in \mathbb{R}^m\}_{t=0}^\infty \), where \( m \geq 1 \) is the dimension of the observation space and \( t \) denotes discrete time. The SoI is the state system, denoted by the sequence \( \{\mathbf{x}_t \in \mathbb{R}^n\}_{t=0}^\infty \), where \( n \geq 1 \) is the dimension of the signal space. If one represents the dynamics of the state signal, \( \mathbf{x}_t \), and the dependence of the measurements, \( \{\mathbf{y}_t\}_{t=0}^\infty \), on \( \{\mathbf{x}_t\}_{t=0}^\infty \), by means of a Markov state-space model, then the estimation of \( \mathbf{x}_t \) given the observations \( \mathbf{y}_{1:t} \) is the typical kind of problem addressed by particle filters and other stochastic filtering techniques [11,24,1].

In this paper we adopt a different perspective. We assume an estimation criterion that consists in the minimization of a sequence of cost functions that involve both the observations and the states. Specifically, at each time \( t \) we have the ability to evaluate the cost \( C_t(\mathbf{x}_{0:t}, \mathbf{y}_{1:t}) \), where \( \mathbf{y}_{1:t} \) are the available measurements and \( \mathbf{x}_{0:t} \in \mathbb{R}^n \times \cdots \times \mathbb{R}^n \) is any trial sequence in the path-space of the SoI. Our ultimate goal is to find the sequence of states that minimizes the cost at each time step. To be specific, the cost at time \( t \) is a function of \( (t+1)n_0 + n_y \) real arguments, namely

\[
C_t : \mathbb{R}^n \times \cdots \times \mathbb{R}^n \times \mathbb{R}^n \times \cdots \times \mathbb{R}^n \to [0, \infty),
\]

and we aim at solving the optimization problems

\[
S_t(\mathbf{y}_{1:t}) = \arg \min_{\mathbf{x}_{0:t}} C_t(\mathbf{x}_{0:t}, \mathbf{y}_{1:t}), \quad t = 0, 1, 2, \ldots,
\]

where \( S_t(\mathbf{y}_{1:t}) \) denotes the set of sequences in the path-space of the SoI that minimize the cost \( C_t \) for the given observations \( \mathbf{y}_{1:t} \). Note that: (a) \( S_0 \) does not depend on any measurement (not available at that time) and (b) the cost functions can present many global minima, hence \( S_t(\mathbf{y}_{1:t}) \) is not necessarily a singleton.

We wish to search for elements of \( S_t(\mathbf{y}_{1:t}) \) using a sequential and recursive algorithm and, accordingly, we restrict the cost functions to be considered here to be those that can be computed recursively. Specifically, if we let \( C_{0}(\mathbf{x}_0) \) be shorthand for \( C_{0}(\mathbf{x}_0, \mathbf{y}_0) \) and, in general, \( C_{t-1}(\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t-1}) \), then we assume that \( C_t \) can be recursively computed from \( C_{t-1} \) as

\[
C_t = C_t(\mathbf{x}_{0:t}, \mathbf{y}_{1:t}) = H(c_t(\mathbf{x}_{t-1:t}, \mathbf{y}_{t}), C_{t-1}),
\]

where \( c_t : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to [0, \infty) \) is a marginal cost function (that depends only on the \( t \)-th observation) and\footnote{It is not the goal of this paper, however, to discuss the degree of suitability of one cost function or the other. This is, in general, dependent on the SoI and the observations and it is the task of the user to choose the type of cost according to her knowledge of the problem at hand.}
function $H : [0, \infty) \times [0, \infty) \to [0, \infty)$ incorporates the marginal cost into the overall cost.

Typical costs can be additive, e.g., $H(C_{t-1}, \mathbf{y}_t; C_{t-1}) = C_{t-1} + C_{t-1}(\mathbf{x}_{t-1}, \mathbf{y}_t)$, multiplicative, e.g., $H(C_{t-1}, \mathbf{y}_t; C_{t-1}) = C_{t-1}C_{t-1}(\mathbf{x}_{t-1}, \mathbf{y}_t)$, or present other forms. The marginal cost function, $c_t$, may have multiple minima. We do not require continuity (unless explicitly stated) and only the ability to compute $c_{t-1}(\mathbf{x}_{t-1}, \mathbf{y}_t)$ and update $C_{t-1}$ given $C_{t-1}$ is taken for granted.

3.2. Algorithm

The minimization of $C_{t}(\mathbf{x}_{t}, \mathbf{y}_{1:t})$ can be attempted using an algorithm similar to the standard particle filter of [14]. In order to describe the proposed methodology we need to introduce additional notation. Specifically:

- Let $M_t$ denote the number of samples (also called particles) drawn at time $t$. The $k$-th particle at this time, denoted $\mathbf{x}_t^{(k)}$, is a point in the state space $\mathbb{R}^n$ drawn from a probability distribution (to be introduced below).
- Let $C_{t-1}(\mathbf{x}_{t-1}, \mathbf{y}_t)$ be shorthand for the cost of the $k$-th stream of particles, $\mathbf{x}_t^{(k)}$, given the collected measurements $\mathbf{y}_1:t$.
- Let $\lambda_0 \in \mathbb{R}^n$ denote a finite measurable set, hence $\ell(\lambda_0) < \infty$, termed the prior set.
- Let $\pi_t$ be a pmf defined over the set $\{1, \ldots, M_t\}$. We refer to it as the selection pmf.
- Let $s : [0, \infty) \to [0, \infty)$ be termed the selection function.
- Let $(\hat{p}_t(\mathbf{x}_t; \mathbf{x}_{t-1}))_n$ be a family of propagation pdf’s, used to generate new particles in the state space.

The recursive SMCM algorithm investigated in this paper can be outlined as follows.

(1) Initialization: Draw $M_0$ random samples from the prior set $\lambda_0$. The initial costs take on a non-negative constant value, $C_0 = g(\lambda_0, \cdot) > 0$ for all $k$.

(2) Recursive step: Let $\Omega_t = \{\mathbf{x}_t^{(k)}, C_{t-1}^{(k)}\}_{k = 1}^{M_t}$ be the collection of particles and costs at time $t$.

(a) Selection: Draw $M_{t+1}$ random indices from $\{1, \ldots, M_t\}$ according to the pmf $\pi_{t+1}$, i.e.,

\[ \hat{\pi}_{t+1}(i), \quad k = 1, \ldots, M_{t+1}, \; i \in \{1, \ldots, M_t\}. \]

Set $\tilde{\mathbf{x}}_t^{(k)} = \mathbf{x}_t^{(\hat{\pi}_t(i))}$ and $\tilde{C}_{t}^{(k)} = s(C_{t-1}^{(\hat{\pi}_t(i))})$ in order to build $\Omega_t = \{\mathbf{x}_t^{(k)}, C_{t-1}^{(k)}\}_{k = 1}^{M_t}$.

(b) Propagation: Draw new particles and compute new costs,

\[ \mathbf{x}_t^{(k)} \sim \hat{p}_t(\mathbf{x}_t|\mathbf{x}_{t-1}; \tilde{\mathbf{x}}_t^{(k)}), \]

\[ C_{t+1}^{(k)} = H(C_{t}^{(k)}(\mathbf{x}_t^{(k)}), \mathbf{y}_t^{(k)}), \]

in order to build $\Omega_{t+1} = \{\mathbf{x}_t^{(k)}, C_{t+1}^{(k)}\}_{k = 1}^{M_{t+1}}$, where $\mathbf{x}_t^{(k)} = \mathbf{x}_t^{(k)}(\mathbf{x}_t^{(k)}), \mathbf{y}_t^{(k)}(\mathbf{x}_t^{(k)}))$.

Estimation using the particles in $\Omega_{t+1} = \{\mathbf{x}_t^{(k)}, C_{t+1}^{(k)}\}_{k = 1}^{M_{t+1}}$ can be as simple as choosing the sequence of particles with the lowest cost, although averaging techniques can also be used [21,19].

The selection pmf, $\pi_t$, and the selection function, $s(\cdot)$, may be assigned different forms by the user of the algorithm. The selection step above is the counterpart of resampling [12,10] in particle filters, hence its goal is to propagate the most promising particles while discarding high-cost ones. An adequate combination of $s(\cdot)$ and $\pi_{t+1}(\cdot)$ can yield any of the resampling procedures commonly used in particle filtering, as well as other schemes. For instance, if we define normalized weights that decrease exponentially with the costs, $w_t^{(k)} = \exp(-C_{t-1}^{(k)})$, and choose a constant selection function, $s(\cdot) = 1/M_t$, then we reproduce exactly the resampling scheme of the standard particle filter with a likelihood of the exponential family [14]. Ref. [19] contains a study of selection methods. The notation is slightly different from the one in this paper, but it is straightforward (and actually convenient) to rewrite the algorithms in [19] using functions $s(\cdot)$ and $\pi_{t+1}(\cdot)$. For example, the selection scheme of [19, Section 4] is rewritten with $s(\cdot) = c$ and $\pi_{t+1}(\cdot) = C_{0}^{(i)} - \min_{i \in \{1, \ldots, M_t\}} C_{0}^{(i)}(\mathbf{x}_t) + M_t^{-1}$. For the purpose of our convergence analysis in Section 4, we will assume that every particle at time $t$ has a non-zero probability to be selected for propagation at time $t + 1$, i.e., that $\pi_{t+1}(\cdot) > 0$ for all $i \in \{1, \ldots, M_t\}$. No particular form of $s(\cdot)$ will be assumed.

An important feature of the proposed formulation of the method is the definition of the marginal cost component, $c_t$. We allow for the marginal cost to vary with time and depend on the pair $\mathbf{x}_{t-1}, \mathbf{y}_t$, while in [21,19] this term can only depend on $\mathbf{x}_t$ and has a fixed functional form. The family of functions $C_t$ that can now be represented is larger and, in particular, it includes costs that depend only on $\mathbf{x}_{t-1}$ and $\mathbf{y}_t$ that are not penalized when computing the cost. Obviously, the notation $C_t(\mathbf{x}_{t-1}, \mathbf{y}_t)$ also allows for simpler functions that depend only on $\mathbf{x}_{t-1}$ and not on $\mathbf{x}_t$.

3.3. Particle filtering and SMCM

A particle filter (PF) is a Monte Carlo algorithm for the recursive approximation of the sequence of posterior probability distributions of the state of a Markov dynamic model [12,11,6,8]. In this section we briefly review the class of PFs derived from the sequential importance resampling methodology [12] and relate it to the SMCM framework of this paper.

Consider the state-space model described by

\[ \mathbf{x}_0 \sim \tau(\mathbf{x}_0), \quad \mathbf{x}_t \sim \tau(\mathbf{x}_t|\mathbf{x}_{t-1}) \] and $\mathbf{y}_t \sim \lambda(\mathbf{y}_t|\mathbf{x}_t)$,

where $\tau(\mathbf{x}_0)$ is the prior pdf of the state signal $\mathbf{x}_0^{0'}, \tau(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the transition pdf that determines the system dynamics and $\lambda(\mathbf{y}_t|\mathbf{x}_t)$ is the conditional pdf of the observations $\mathbf{y}_t$, i.e., the likelihood of $\mathbf{x}_t$. The posterior pdf of $\mathbf{x}_0$ given the observations $\mathbf{y}_1:t$ (denoted $\pi$) can be
We can also compute proper importance weights recursively, i.e., given the stream \( c_{1:t} \) at every time step. This yields a very simple form, resampling consists in drawing \( x_{i:t} \) few time steps. This difficulty is overcome by introducing resampling \( \text{SIR} \).

The standard PF introduced in [12] presents the weights are normalized, namely

\[
W_i^t \propto \pi(x_i^t | y_{1:t}) \propto \frac{\lambda(y_i | x_i^t) \pi(x_i^t | y_{1:t-1}) W_{i-1}}{\pi(x_i^t | y_{1:t})}. 
\]

Eqs. (10) and (11) yield the sequential importance sampling \( \text{SIS} \) algorithm [12]. Each weighted sample \( (x_{i:t}, W_i^t) \), for \( i = 1, \ldots, M \), is referred to as a particle and the weights are normalized, \( \sum_{i=1}^{M} W_i^t = 1 \). A practical problem of this procedure is the so-called degeneracy of the weights [12], that renders the algorithm useless after a few time steps. This difficulty is overcome by introducing resampling steps [4,12,10,19]. In its conceptually simpler form, resampling consists in drawing \( M \) times from the discrete distribution that assigns probability mass \( W_i^t \) to the \( i \)-th particle, and then assign uniform weights to the resampled particles (this procedure is known as multinomial resampling [10]). The SIS algorithm combined with resampling steps is referred to as sequential importance resampling \( \text{SIR} \).

The standard PF introduced in [14], also termed bootstrap filter \( \text{BF} \), is obtained from the general SIR scheme by choosing \( \psi(x_{i:t} | y_{1:t}) = \tau(x_i | x_{i-1}) \) and resampling at every time step. This yields a very simple algorithm that can be outlined as follows (assume \( M_t = M \) for all \( t \)).

1. Initialization: Draw \( x_{0:t}^i \sim \tau(x_0) \) and set \( w_0^i = 1/M \), for \( i = 1, \ldots, M \).
2. Recursive step: Given \( (x_{i:t}^i, W_i^t) \),
   a. resample to obtain a new set of particles \( x_{i:t}^i, W_i^t \) with equal weights,
   b. draw new samples \( x_{i+1:t}^i \sim \tau(x_{i+1:t} | x_{i:t}) \), for \( i = 1, \ldots, M \), and
   c. assign new weights, \( W_{i+1:t}^i \propto \lambda(y_{i+1:t} | x_{i+1:t}) \).

Note that we do not keep the complete streams of samples \( x_{0:t}^i \) in this simple algorithm. Most integrals with respect to \( (w.r.t.) \) the filtering pdf \( \pi(x_{i:t} | y_{1:t}) \) can be easily approximated by summations using the particles generated by the BF. Specifically, if \( f \) is some integrable function of the state \( x_t \), then we can write

\[
\int f(x_t) \pi(x_t | y_{1:t}) dx_t \approx \sum_{i=1}^{M} f(x_i^t) W_i^t. 
\]

Any SIR algorithm can be easily expressed within the SMCM framework. In particular, a generic SIR procedure with proposal pdf \( \psi(x_{i:t} | x_{i-1}, y_{i:t}) \), multinomial resampling and \( M \) particles can be interpreted as a SMCM algorithm with \( M_t = M \),

\[
\rho_i(x_{0:t-1}) = \psi(x_i | x_{i-1}, y_{i:t}).
\]

\[
H(a, b) = a + b,
\]

and

\[
c_i(x_{0:t-1}, y_{i:t}) = \log(\psi(x_i | x_{i-1}, y_{i:t})) - \log(\lambda(y_i | x_i)) - \log(\tau(x_i | x_{i-1})).
\]

The corresponding importance weights can be recovered as \( W_i^0 \propto \exp(-c_i^0) \) and, as a consequence, the multinomial resampling is implemented within the SMCM framework with \( c_i^0 \propto W_i^0 \) and \( s(c) = 1 \). The two algorithms \( \text{SIR} \) and \( \text{SMCM} \) become identical if we draw initial samples at time \( t = 0 \) from the same distribution, \( \tau(x_0) \), and resample/ select in the same time steps.

4. Asymptotic convergence

4.1. Cost minimizers

Given the statement of the SMCM algorithm in Section 3, the straightforward question to pose is whether the algorithm can produce a sequence of state estimates at a given time \( t \) with a cost which is arbitrarily close to the minimum one. In particular, let \( x_{0:t} = \{x_{0:1}^1, \ldots, x_{0:t}^M \} \in S_t(y_{1:t}) \)

be a sequence of states (possibly not unique) that minimizes the overall cost for the sequence of observations \( y_{1:t} \) and let

\[
\hat{x}_{0:t} = \{\hat{x}_{0:1}, \ldots, \hat{x}_{0:t} \} \in S_t^M(y_{1:t}).
\]

where \( S_t^M(y_{1:t}) \) is the set of approximate minimizers generated by the SMCM algorithm, i.e.,

\[
S_t^M(y_{1:t}) = \arg \min_{x_{0:t} \in S_t^M(y_{1:t})} C_t(x_{0:t}, y_{1:t}).
\]

Can we guarantee that \( \hat{x}_{0:t} \rightarrow x_{0:t}^\ast \) for some \( x_{0:t}^\ast \in S_t^M(y_{1:t}) \) and \( x_{0:t}^\ast \in S_t(y_{1:t}) \)? This is a hard problem in general, unless we impose further constraints on the cost function \( C \). Assume that \( T \) observations, \( y_{1:T} \), are to be collected. In general, the optimal value of the SoI at time \( t < T \), \( x_{t|t}^0 \), cannot be computed before time \( T \). Since any SMCM algorithm draws the samples \( x_{0:t}^1, \ldots, x_{0:t}^M \) at time \( t \) and does not modify them at any later time, it is difficult to ensure the convergence of the sequence \( x_{0:T|T} \) except for cost functions with suitable structures (e.g., when the overall cost is additive and the marginal cost at time \( t \) depends only on \( x_t \) and not on \( x_{t-1} \)).

The original question on the convergence of \( x_{0:t} \) can be simplified, but still retain a good deal of practical interest, if we restrict our attention to the latest realization of the system state, \( x_t \). Thus, we can ask whether a suitably designed SMCM algorithm can ensure that \( x_{0:t} \rightarrow x_{0:t}^\ast \) in some statistical sense. Our answer is, to some extent, positive. Indeed, we can show that, by adequately
choosing the family of propagation densities \( \{ \rho_t \}_{t=0} \), a
SMCM method can produce particles arbitrarily close to the
minimizer \( x^*_t \) at time \( t \) with high probability.

### 4.2. Assumptions and notations

Let us consider an arbitrary but fixed series \( \{ x^n_t \}_{t=0} \),
where \( x^n_t \) is the \((t+1)\)th vector in some sequence \( x^n_t \in S_t(y_{1:t}) \) (note that there may exist several possible
sequences of minimizers to choose from, since the sets
\( S_t(y_{1:t}) \) are not necessarily singletons). All through this
section, as well as in the proofs in Appendix A, we assume
that the sequence of observations, \( y_{1:n} \), is unknown but
fixed. This is a rather common assumption in the analysis of
particle filters (see, e.g., [5,17,1]) and, in our case, it
implies that the minimizers \( \{ x^n_t \}_{t=0} \) are deterministic,
as they depend only on the observations and the form of the
cost function. In the subsequent analysis, randomness is
only due to the selection and propagation steps of the
SMCM algorithm, not to the observations.

We equip the set \( \mathbb{R}^n \) with a proper distance \( d(\cdot, \cdot) \) and
introduce a sequence of functions \( a_t: \mathbb{R}^n \to \mathbb{R}^n \), \( t \in \mathbb{N}^* \),
in the state space. The role of the latter functions is to provide
a (possibly rough) approximation to the dynamics of the sequence of minimizers \( \{ x^n_t \}_{t=0} \). Specifically, we assume that
the \( a_t \)’s comply with the following statements.

**Assumption 1.** For any set \( A \subset \mathbb{R}^n \), let

\[
a_t(A) = \{ x \in \mathbb{R}^n : x = a(x') \text{ for some } x' \in A \}
\]

(19)

denote the image of \( A \) under the function \( a_t \). We assume that,
if \( A \in \mathbb{R}^n \), then \( a_t(A) \in \mathbb{R}^n \) for all \( t \).

**Assumption 2.** There exists a finite constant \( A \in \mathbb{R}^+ \) such
that \( A^t \triangleq d(a_t(x^n_t), x^n_{t+1}) < A \) for all \( t \).

**Assumption 3.** For any pair \( x, x' \in \mathbb{R}^n \) and all \( t \in \mathbb{N}^* \), if
\( d(x, x') < \infty \) then \( d(a_t(x), a_t(x')) < \infty \).

Next, we use the newly introduced functions \( \{ a_t \}_{t=0} \) to
construct the propagation densities of the SMCM algo-
rithm. In particular, we assume that

\[
\rho_t(x^n_t| x^n_{t-1}) = \gamma(x_t; a_{t-1}(x^n_{t-1}), r_t),
\]

(20)

where \( \gamma(x; c, r) \) denotes a pdf which is

- strictly positive in the open ball with center \( c \in \mathbb{R}^n \)
  and radius \( r > 0 \), denoted \( B(c, r) \triangleq \{ x \in \mathbb{R}^n : d(x, c) < r \} \),
  i.e., \( \gamma(x; c, r) > 0 \ \forall x \in B(c, r) \),
- uniformly continuous in \( B(c, r) \) and
- null outside \( B(c, r) \), i.e., \( \int_{\mathbb{R}^n} \gamma(x; c, r) dx = 1 \).

Many densities comply with these assumptions, e.g., the
uniform pdf in \( B(c, r) \), denoted \( U(B(c, r)) \), or a Gaussian
density truncated outside the set \( B(c, r) \). Then, we assume
that the selection pmf, \( \zeta_t(k), k \in \{ 1, \ldots, M_t-1 \} \), is
strictly positive (implying that all particles at time \( t-1 \) have
a non-zero probability to be propagated to time \( t \)) and
formally combine the selection and propagation stages of the
algorithm into a single step. The particles are then
drawn from a mixture density,

\[
X^n_t \sim \rho_t(x^n_t) \triangleq \sum_{k=1}^{M_t-1} \zeta_t(k) \gamma(x^n_t; a_t(x^n_{t-1}), r_t), \quad i = 1, \ldots, M_t
\]

(21)

for some prescribed \( r_t > 0 \). The sequence of radii \( \{ r_t \}_{t=0} \)
must be selected by the user and we will show that, if
every \( r_t \) is finite but sufficiently large, then the SMCM
algorithm is guaranteed to converge in probability.

### 4.3. Asymptotic convergence in probability

We will show that a SMCM algorithm can be designed
in such a way that, for sufficiently large natural numbers
\( M_t \)’s, the state-space samples

\[
x^n_t \triangleq \arg \min_{x \in \mathbb{R}^n} d(x, x^n_{t-1})
\]

(22)

converge to \( x^*_t \), in probability, when \( t \to \infty \). The main
convergence result in this paper is stated below.

**Proposition 1.** If we consider

- a fixed sequence of minimizers \( \{ x^*_t \}_{t=0} \),
- a sequence of functions \( \{ a_t \}_{t=0} \) that satisfy Assumptions 1,
  2 and 3 for the sequence \( \{ x^n_t \}_{t=0} \),
- a sequence of propagation pdf’s of the form in Eq. (21)
  (this implies strictly positive selection pmf’s, \( \zeta_t(i) > 0 \) for
  all \( t \) and all \( i \in \{ 1, \ldots, M_t-1 \} \), and
- a sequence of sufficiently large finite radii \( \{ r_t \}_{t=0} \),

then

\[
\lim_{t \to \infty} \lim_{M_t \to \infty} d(x^n_t, x^*_t) = 0 \text{ in probability}.
\]

To be specific, convergence of the latter limit in probability
means that for any arbitrarily small \( \varepsilon, \delta > 0 \) there exist
\( t_0 \in \mathbb{N} \) and a sequence of natural numbers
\( \{ M_t \}_{t=t_0}^{M_t} \), such that \( P(d(x^n_t, x^*_t) < \varepsilon) > 1 - \delta \)
for all \( t > t_0 \) and all sequences \( \{ M_t \}_{t=t_0}^{M_t} \) that satisfy the inequalities
\( M_t > M_{t+1} \) for \( t = 0, 1, \ldots, T \). Notation \( P(d(x^n_t, x^*_t) < \varepsilon) \) is
shorthand for \( P(A^t) \), where

\[
A^t \triangleq \{ (x^n_1, \ldots, x^n_T) \in \mathbb{R}^n \times \cdots \times \mathbb{R}^n : d(x^n_k, x^*_k) < \varepsilon
\text{ for some } k \in \{ 1, \ldots, M_t \} \}.
\]

(23)

The proof of Proposition 1 is given in Appendix A and a
detailed construction of \( P(A^t) \) is provided in Appendix B.

Formally, Proposition 1 states that it is possible to
design SMCM algorithms that generate, with high prob-
ability, particles arbitrarily close to the minimizers \( x^*_t \)
whenever \( t, r_t \) and \( M_t \) are sufficiently large. Indeed, we can prove that \( P(d(x^n_t, x^*_t) < \varepsilon) \) approaches 1 exponentially fast as
\( M_t \to \infty \) (see Eq. (A.23) in Appendix A for details).

Intuitively, the proof of Proposition 1 shows that
the algorithm generates a sequence of particle sets,
\( A_t = \{ x^n_t \}_{t=0}^{M_t}, t = 0, 1, 2, \ldots \) that converges toward,
and eventually "capture", the minimizers \( x^*_t \) with high
probability. This illustrated in Fig. 1. The particle sets
\( A_1, \ldots, A_4 \) are depicted with solid black lines, while the
corresponding minimizers, \( x^*_1, \ldots, x^*_4 \), are denoted by stars.
At time \( t \) = 2, 3, 4, the image of \( A_{t-1} \), \( a_t(A_{t-1}) \), is
shaded and delimited by a dashed line. The Figure is
"read" from left to right. At time \( t = 1 \), the cloud of
particles, \( A_1 \), is far away from the minimizer, \( x_{0,1}^0 \), but the algorithm is constructed in such a way that, at time \( t = 2 \):

- the distance from \( x_{2,2} \) to the border of \( a_t(A_1) \) is bounded,
- the particle cloud \( A_2 \) is a proper superset of the image \( a_t(A_1) \), i.e., \( a_t(A_1) \subset A_2 \), and
- the distance between \( x_{2,2}^o \) and the boundary of \( A_2 \) is smaller than the distance from \( x_{0,1}^0 \) to the boundary of \( A_1 \).

At time \( t = 3 \) the distance between the particle cloud \( A_3 \) and \( x_{3,3}^o \) is shorter than the distance between \( A_2 \) and \( x_{2,2}^o \) and, finally, at time \( t = 4 \), \( x_{4,4}^o \) lies “within” \( A_4 \). Remarkably, this kind of convergence can be achieved without a precise knowledge of the system dynamics (i.e., we do not use an explicit state-space model of the system). Instead, we need a sequence of functions \( \{a_t\}_t \) that comply with Assumptions 1–3. Such functions provide information on the dynamics of the minimizers, indeed, but this is very rough, since it reduces to a bound on the departure of \( x_{o,t}^o \) from \( a_{t-1}(x_{o,t-1}) \).

Note that Fig. 1 is just a piece of artwork for the illustration of the intuitive ideas underlying the proof of Proposition 1. See Appendix A for the complete formal argument. Fig. 3 in Section 5.1 shows how the particle clouds (in a 1-dimensional space) actually converge to a sequence of exact minimizers for an actual SMCM algorithm.

If \( c_t(x, y) \) is uniformly continuous at \( x = x_{o,t}^o \) for all \( t \in \mathbb{N} \) and the finite radii \( \{\tau_t\}_t \) are sufficiently large, then \( \lim_{t \to \infty} \lim_{M_t \to \infty} d(x_{o,t}^o, x_{o,t}^o) = 0 \) in probability.

Specifically, the Claim states that for any arbitrarily small \( \varepsilon, \delta > 0 \) there exist \( t_\varepsilon \in \mathbb{N} \) and a sequence \( \{M_{t,\varepsilon,\delta} < \infty\}_t \) such that \( P(d(x_{o,t}^o, x_{o,t}^o) > \varepsilon) < 1 - \delta \) for all \( t > t_\varepsilon \) and all sequences \( \{M_t\}_{t \in \mathbb{N}} \) satisfying \( M_t > M_{t,\varepsilon,\delta} \).

A proof can be constructed that follows exactly the same steps as the proof of Proposition 1. The continuity of \( c_t(x, y) \) is simply used to show that a sample sufficiently close to \( x_{o,t}^o \) also has an “almost minimal” marginal cost.

5. Examples

5.1. A nonlinear one-dimensional system

As a first example, we apply a SMCM algorithm to the minimization of the cost function

\[
C_t(x_{o,t}, y_{1:t}) = c_t(x_{t-1:t}, y_t) = (1 - \varepsilon) \left( y_t - \frac{x_t^2}{20} \right) + \varepsilon (x_t - a_{t-1}(x_{t-1}))^2.
\]

\( t \in \mathbb{N} \), \( y_t, x_t \in \mathbb{R} \) are the observations and the Sol, respectively, \( \varepsilon \geq 0 \) is a non-negative weight and we set \( a_t(x) = x + 8 \cos(1.2t) \). When compared to the generic...
of minimizers \( x_{k}^{0} \), analytically and, therefore, it is easier to illustrate the convergence of the SMC algorithm in the way predicted either by Proposition 1 (when \( \varepsilon > 0 \)) or Claim 1 (when \( \varepsilon = 0 \)).

To be specific, if we take \( \varepsilon = 0 \) then we readily find that

\[
x_{t}^{0} = \begin{cases} 
0 & \text{when } y_{t} \leq 0, \\
\pm \sqrt{20 y_{t}} & \text{when } y_{t} \geq 0,
\end{cases}
\]  

(27)

hence there is a unique minimizer when the observation \( y_{t} \) is negative and two optimal (equally valid) solutions when \( y_{t} \) is positive. If we choose a positive weight, \( \varepsilon > 0 \), then the solution is similar, but it also involves \( x_{t-1} \), namely

\[
x_{t}^{0} = x_{t}^{0} \text{ and } x_{t-1}^{0} = x_{t-1}^{0} - 8 \cos(1.2 t).
\]

(28)

This very simple system enables us to illustrate the convergence of both \( d(x_{t}^{0}, x_{t}^{0}) = \| x_{t}^{0} - x_{t}^{0} \| \) and \( d(x_{t}, x_{t}^{0}) = \| x_{t} - x_{t}^{0} \| \) for the simulations, we generate the SoI and the observations from the nonlinear dynamic system

\[
x_{0} \sim N(0, 1),
\]

(29)

\[
x_{t} = 0.5 x_{t-1} + \frac{25 x_{t-1}}{1 + x_{t-1}^{2}} + 8 \cos(1.2 t) + v_{t},
\]

(30)

\[
y_{t} = \frac{x_{t}^{2}}{2} + m_{t},
\]

(31)

where \( x_{t} \in \mathbb{R} \) is the system state at time \( t \in \mathbb{N}^{*} \), \( y_{t} \in \mathbb{R} \) is the corresponding measurement and the noise processes \( v_{t} \sim N(0, \sigma_{v}^{2}) \) and \( m_{t} \sim N(0, \sigma_{m}^{2}) \) are zero-mean Gaussian with variances \( \sigma_{v}^{2} = 1 \) and \( \sigma_{m}^{2} = 2 \), respectively. Model (29)–(31) has been studied by several authors in the stochastic filtering literature [23,14,16]. In our case, it is just a mechanism to produce the signals in a way that departs explicitly from the selected sequence of functions \( a_{t} \) (compare the latter with the dynamic Eq. (30)).

We have applied the simple SMC algorithm below, with a fixed number of particles \( M = M_{t} \), to the sequential optimization of the cost function (26).

(1) **Initialization:** Draw \( M \) i.i.d. samples from a Gaussian distribution centered at \( x = 500 \), \( x_{0}^{(k)} \sim N(0, 500, 1) \), \( k = 1, \ldots, M \), and take \( C_{0}^{(k)} = 0 \) for all \( k \).

(2) **Recursive step:** The selection pdf of choice at time \( t + 1 \) is [21]

\[
\pi_{t+1}^{(k)}(i) \propto \left( R_{t+1}^{(k)} - \min_{j} R_{t+1}^{(j)} + \frac{1}{M} \right)^{-2},
\]

(32)

where \( R_{t+1}^{(k)} = (y_{t+1} - a(x_{t+1}^{(k)}))^{2} \) is a “risk” that we associate to the \( i \)-th particle. Therefore, indices are drawn as \( j^{(k)} \sim \pi_{t+1}^{(k)}(i) \), for \( k = 1, \ldots, M \) and \( i \in \{1, \ldots, M\} \). Taking the identity selection function, \( s(c) = c \), yields the set of selected particles \( \{\tilde{x}_{t+1}^{(k)} = x_{t+1}^{(k)}, \tilde{C}_{t}^{(k)} = C_{t}^{(k)} M_{k} = 1\} \). The propagation pdf of choice is

\[
x_{t+1}^{(k)} \sim \tilde{p}(x_{t+1}^{(k)} | \tilde{x}_{t}^{(k)}) = TN(x_{t+1}; a(x_{t+1}^{(k)}), 10), \quad k = 1, \ldots, M,
\]

(33)

where \( TN(x; m, s^{2}) \) denotes a Gaussian density with mean \( m \) and variance \( s^{2} \) truncated outside the interval \((m - 10 s, m + 10 s)\). The cost of the \( k \)-th particle is \( C_{t}^{(k)} = C_{t}(\tilde{x}_{t+1}^{(k)}, \tilde{y}_{t+1}) \).

Let us note that the SMC algorithm described above strictly complies with the assumptions of Proposition 1 and Claim 1. Indeed,

- the sequence of minimizers is deterministic (once the observations \( y_{t} \) for the simulation have been generated);
- the truncated Gaussian densities \( TN(x; m, s^{2}) \) with support in \((m - 10 s, m + 10 s)\) comply with the definition of a density \( \gamma(x; m, s) \) as specified in Section 4.2;
- the functions \( a_{i}(x) = x + 8 \cos(1.2 t) \) satisfy Assumptions 1, 2 and 3; and
- the radius \( r_{t} = 10 s \) for (all \( t \)) has been found to be large enough to attain convergence in all simulations.

In particular, Assumptions 1 and 3 are straightforward to check, using the continuity of \( a_{t} \), while, for Assumption 2, we note just that the upper bound \( A \) can be explicitly computed for each sequence (since the sequence of observations has a finite length).

For our first simulation, we have independently generated 500 sequences, \( y_{0:T}(j) \), with \( T = 200 \) and \( j = 1, 2, \ldots, 500 \), and computed the corresponding sequences of minimizers \( x_{t}^{(k)}(j) \), for \( t = 0, \ldots, T \) and \( j = 1, \ldots, 500 \). Then, we have applied the SMC algorithm with \( M = 10,000 \) particles to compute the sequence of approximate minimizers \( x_{t}^{(k)}(j) \), \( j = 1, \ldots, 500 \), for every sequence of observations.

Fig. 2 (left) shows the distance between the true and approximate minimizers as a function of time and averaged over the 500 simulations, i.e., \( \frac{1}{500} \sum_{j=1}^{500} \| x_{t}^{(k)}(j) - x_{t}^{(k)(j)} \| \), as a solid line. It is clearly seen that the distance is very large during the initial time steps (notice the mismatch between the prior density of \( x_{0} \) and the initialization of the SMC algorithm, which has been introduced for the purpose of illustrating the convergence of the algorithm with time explicitly). However, the proposed procedure converges steadily and, for \( t > 40 \) time steps, \( x_{t}^{(k)}(j) \) is already a good approximation of \( x_{t}^{(k)} \), with a mean distance of \( \approx 4 \times 10^{-4} \). In the same plot, we also depict the averaged distance between \( x_{t}^{(k)} \) and the minimum-cost particle. When \( \varepsilon = 0 \), the minimum-cost particle and \( x_{t}^{(k)} \) coincide (i.e., if \( \varepsilon = 0 \) then \( x_{t}^{(k)} = x_{t}^{(k)} = \tilde{x}_{t}^{(k)} \) as defined in Eq. (25) and used in Claim 1), but this is not necessarily the case when \( \varepsilon > 0 \). Specifically, the dashed line in the figure shows this averaged distance when \( \varepsilon = 0.1 \). The reason for this mismatch is that the second term in the cost function (26) depends on the sample \( x_{t+1}^{(k)} \), which is not updated by the SMC algorithm at time \( t \). This fact illustrates the
need for a careful choice of the costs in practical problems where the aim is the estimation of the system state. Be aware, nevertheless, that this result is not contradictory with Proposition 1. Indeed, the simulation shows that $x_{1}^{10} \rightarrow x_{2}^{10}$ both with $\epsilon = 0$ and $\epsilon > 0$, but in the latter case $x_{1}^{10}$ is not the minimum-cost particle.

While convergence with time is clearly observed in Fig. 2 (left), the accuracy of the steady-state distance between the approximate and true minimizers depends on the value of $M$. Fig. 2 (right) illustrates the convergence of the approximate minimizers with the number of particles, $M$. It shows, with a solid line connecting hollow circles, the distance between $x_{1}^{M}$ and $x_{2}^{M}$ averaged over the last 100 time steps of 200 independent simulations. It can be seen how the distance decreases steadily with $M$. The average distance between the minimum-cost particle, for $\epsilon = 0.1$, and the true minimizers is depicted with a dashed line connecting black squares. This distance also decreases with $M$, but apparently converges to a positive “floor” value of 0.1. Again, this is due to the second term in the cost function (26) and the fact that the sample $x_{i-1}^{M}$ is inherited from time $t-1$ and not updated when $y_{t}$ is received.

We conclude this example with a plot that shows how the particle sets $A_{t} = \{x_{i}^{M}\}_{i=1}^{M}$ eventually “capture” the minimizers $x_{i}^{M}$ as time evolves. Only for this simulation, we set $M = 500$. Fig. 3 is a numerical counterpart of the (merely illustrative) diagram in Fig. 1. It shows the minimizers $x_{i}^{M}$, represented by dark-colored “+” signs, for $t = 10, 20, \ldots, 80$, together with the particle sets (light-colored ‘-‘ signs). Initially, the samples $x_{i}^{M}$ are far away from the $x_{i}^{M}$ but, as time evolves, this distance is reduced. At times $t = 60, 70$ and 80, at least one minimizer lies among the particles in $A_{t}$.

5.2. A chaotic CO$_2$ laser model

As a second example, we consider a more challenging system that describes the dynamics of a CO$_2$ laser with modulated losses in a chaotic regime [18]. A deterministic model of the laser dynamics is given by the set of differential equations

$$
\dot{x}_{1}(s) = Q_{0} x_{1}(s) x_{2}(s) - Q_{3} x_{1}(s),
$$

$$
\dot{x}_{2}(s) = -\gamma_{1} x_{2}(s) - 2Q_{0} x_{1}(s) x_{2}(s) + G x_{3}(s) + x_{4}(s) + P,
$$

$$
\dot{x}_{3}(s) = -\gamma_{1} x_{3}(s) + G x_{2}(s) + x_{5}(s) + P,
$$

$$
\dot{x}_{4}(s) = -\gamma_{2} x_{4}(s) + Z x_{2}(s) + G x_{5}(s) + ZP,
$$

$$
\dot{x}_{5}(s) = -\gamma_{2} x_{5}(s) + Z x_{3}(s) + G x_{4}(s) + ZP,
$$

(34)

where $s \in \mathbb{R}$ denotes continuous time, $\dot{x}_{i}(s) = dx_{i}(s)/ds$ denotes the time derivative of the $i$-th state variable, $Q_{0}(s) = Q_{0}[1 + z \sin^{2}(B_{0} + F(s))]$ and $F(s) = H \sin(2\pi f s)$ is an external sinusoidal forcing signal with frequency $f$ (Hz). In the above equations, $x_{1}(s)$ represents the laser output intensity, $x_{2}(s)$ is the population inversion between the two resonant levels, and $x_{3}(s), x_{4}(s)$ and $x_{5}(s)$ account for molecular exchanges between the two levels resonant with the radiation field and the other rotational levels of
the same vibrational band. See [18] for physical details, including the meaning of the fixed parameters \(Q_0, \alpha, B_0, H, \gamma_1, \gamma_2, G, Z \) and \(P\). The values selected for the simulation ensure chaotic behavior and are shown in Table 1.

We address the problem of tracking the latent variables \(\{x_i(s)\}_{i=1}^{n+1}\) in (34). In order to simulate the evolution of the system, we transform the continuous-time deterministic model (34) into a discrete-time dynamic random system. Time discretization is most easily carried out by Euler’s method and we assume a white Gaussian perturbation of the latent variables, i.e.,

\[
\begin{align*}
    x_{1,t} &= x_{1,t-1} + \frac{Q_0 x_{1,t-1} x_{2,t-1} - Q_1 x_{1,t-1}}{T} + \sigma_1 \epsilon_{1,t}, \\
    x_{2,t} &= x_{2,t-1} + (\gamma_1 + 2Q_0 x_{1,t-1}) x_{2,t-1} + G x_{3,t-1} + X_{4,t-1} + P + \sigma_2 \epsilon_{2,t}, \\
    x_{3,t} &= x_{3,t-1} + \frac{-\gamma_2 x_{3,t-1} + G x_{2,t-1} + X_{5,t-1} + P}{T} + \sigma_3 \epsilon_{3,t}, \\
    X_{4,t} &= x_{4,t-1} + \frac{-\gamma_2 x_{4,t-1} + G x_{3,t-1} + X_{3,t-1} + ZP}{T} + \sigma_4 \epsilon_{4,t}, \\
    X_{5,t} &= x_{5,t-1} + \frac{-\gamma_2 x_{5,t-1} + G x_{4,t-1} + X_{3,t-1} + ZP}{T} + \sigma_5 \epsilon_{5,t},
\end{align*}
\]

(35)

\[ t \in \mathbb{N}, \text{ where } T = 5 \times 10^{-3} \text{ time units is the time step used to approximate the derivatives by differences, } Q_{1,t} = Q_1(s = t), \ x_{1,t} = x_1(s = t) \text{ and } \epsilon_{i,t} \sim N(0, \sigma^2_{i}), \quad i = 2, 3, 4, 5, \text{ are Gaussian noise terms with zero mean and variance } \sigma^2_{i} = 10^{-6}. \text{ Note that } x_{1,t} \text{ is deterministic given } \{x_{i-1}\}_{i=1}^{n+1} \text{ (otherwise the features of the dynamics are changed abruptly compared to the deterministic continuous-time case). The state of system (35) consists of } x_0 = [x_{1,1}, \ldots, x_{5,1}]^T \in \mathbb{R}^5 \text{ and we start the simulation with the state value}
\]

\[ x_0 = [8.0221 \times 10^{-3}, 1.5817, 1.6770, 1.6384 \times 10, 1.6722 \times 10]^T. \]

(36)

For convenience, we represent model (35) as a vector function \(g_i, \text{ i.e., } x_i = g_i(x_{i-1}, v_i), \text{ where } v_i = [0, v_2, \ldots, v_{5}]^T \) is the noise vector. Furthermore, we use the recursive notation

\[ g^k(x_{i-1}, v_{i+k-1}) = g(x_{i-1}, v_{i+k-1}), \quad k = 2, 3, \ldots, \]

(37)

with \( g^1 = g \) to represent \( k \) successive iterations of the random dynamic model. To indicate \( k \) deterministic steps, we write \( g^{k} (x_{i-1}, 0) \).

The only observable in an experimental setup is the laser intensity \( x_{1,t} \) [18]. Moreover, it is not possible in general to collect measurements with a frequency as small as \( 1/T \). Therefore, for our simulations we assume that an observation is available every \( 10T \) time units, i.e., every 10 steps of the discrete-time model (35). The \( n \)-th observation is denoted \( y_n = z_{1,n} + m_n \), where \( m_n \sim N(0, \sigma^2_m) \) is a zero-mean Gaussian perturbation with variance \( \sigma^2_m = 10^{-3}, z_{1,n} = x_{1,10n} \) is the laser intensity variable and \( n \in \mathbb{N} \). If we let \( z_n = [z_{1,n}, \ldots, z_{n,n}]^T \) with the obvious notation \( z_{1,n} = x_{1,10n} \), then we can rewrite the state-space model used for the simulation of the CO2 laser as

\[
\text{(state eq.) } z_n = g^{10}(z_{1,n}, v_{10(n-1)} + 1) + n_{10n},
\]

(38)

\[
\text{(observation eq.) } y_n = z_{1,n} + m_n.
\]

(39)

In this example we study the performance of the SMCM algorithms built from four different cost functions

\[ C_{a,n}(z_{0:n}, y_{1:n}) = c_{a,n}(z_n, y_n) = |y_n - z_{1,n}|, \]

(40)

\[ C_{b,n}(z_{0:n}, y_{1:n}) = \frac{1}{2} C_{b,n-1}(z_{0:n-1}, y_{1:n-1}) + c_{a,n}(z_n, y_n), \]

(41)

\[ C_{c,n}(z_{1:n}, y_{1:n}) = c_{c,n}(z_n, y_n) = \begin{cases} 1 - z_{1,n}^2/y_n^2 & \text{if } y_n \geq U, \\ 1 & \text{if } y_n < U, \end{cases} \]

(42)

\[ C_{d,n}(z_{1:n}, y_{1:n}) = \begin{cases} C_{d,n-1}(z_{1:n-1}, y_{1:n-1}) & \text{if } y_n \geq U, \\ 1 & \text{if } y_n < U, \end{cases} \]

(43)

where \( n = 1, 2, \ldots, U = 1.6 \times 10^{-4} \) is a threshold value. Note that functions \( C_{a,n} \) and \( C_{c,n} \) do not depend on the history of the Sol but only on the last value, i.e., they are of the form \( H(a, b) = a \), while functions \( C_{b,n} \) and \( C_{d,n} \) are constructed from the same marginal costs as \( C_{a,n} \) and \( C_{c,n} \), respectively, but they depend on the past values of the Sol. The reason for the use of a threshold value, \( U \), in the definition of \( C_{c,n} \) and \( C_{d,n} \) above is that the laser intensity can take very small values that are completely masked by the observational noise. In this case, the observation is not useful for the algorithm.

The generic SMCM algorithm with fixed \( M = M_n = 500 \) is constructed as follows.

(1) **Initialization:** Produce a sample of size \( M \) by setting \( z_{1:0} = x_{1:0} \) and drawing \( z_{i:0} \sim U(x_{i:0} - 1, x_{i:0} + 1), i = 2, \ldots, 5, k = 1, \ldots, M \). Assign initial costs, \( c_{0:k} \) for \( k = 1, 2, \ldots, M \). This is different for the different functions, namely, \( c_{i:0} = c_{b:0} = 0 \) and \( c_{i:0} = c_{d:0} = 1 \).

(2) **Recursive step:** Assume \( \Omega_n = \{z_{n:k}, c_{n:k} \}_{k=1}^{n+1} \) is available. The selection pmf at time \( n+1 \) is

\[ \pi_{n+1} (i) \propto (R_{n+1}^{(i)} - \min_{k} R_{n+1}^{(k)} + 1/M)^{-2}, \]

(44)

where \( R_{n+1}^{(k)} = |y_{n+1} - |g^{k} (z_{n+1}, 0)||_1 \) and \( |u|_1 \) denotes the \( i \)-th element of vector \( u \). Note that \( R_{n+1}^{(k)} \) can be interpreted as a prediction of the cost of the \( k \)-th particle.

Indices are drawn as \( i \sim \pi_{n+1} (i), \text{ for } k = 1, \ldots, M \text{ and } i \in \{1, \ldots, M\}, \) and the selected particles are \( z_{n+1}^{(k)} = z_{i}^{(k)} \). We propagate the particles using Eq. (38) to obtain \( z_{n+1}^{(k)} = k, 1, \ldots, M, \) and then compute the corresponding costs \( C_{n+1} \).

Note that in this algorithm the composition of functions \( g^{10} \) plays the role of \( a_{n} \) in the general scheme.
The propagation density does not match the assumptions of Proposition 1, though, as it does not have a finite support.

In order to assess the tracking performance of the proposed method we have also applied the standard bootstrap filter (BF) described in Section 3.3 to the state-space model given by Eqs. (38) and (39).

Fig. 4 illustrates the dynamics of the laser model and the performance of the SMCM algorithm with cost function $C_{c,n}$ and the BF. Fig. 4 (left) shows a typical realization of the output intensity variable, $z_{1,n}$, the absolute value of the associated observations, $|y_{n}|$, and the estimates from the two Monte Carlo algorithms. It can be seen that the Gaussian noise completely hides the variable $z_{1,n}$ for relatively long periods of time. Only some spikes can be actually observed with accuracy, but tracking can be achieved, nonetheless, both with the BF and the SMCM algorithm. Valid estimates can also be computed for the latent variables $z_{2,n}, \ldots, z_{5,n}$. In particular, Fig. 4 (right) shows the signal $z_{2,n}$ and its estimates.

We have numerically approximated the normalized mean square error (NMSE) for each variable $z_{i,n}$, $i = 1, \ldots, 5$. Let $\hat{z}_{i,n}(k)$ denote the estimate of variable $z_{i,n}(k)$ obtained in the $k$-th simulation trial. For the SMCM algorithms, $\hat{z}_{i,n}(k)$ is the $i$-th component of the particle with the least cost, while for the BF the estimate results from the weighted mean of the particles. If we assume that $N_{s}$ independent simulations have been carried out, each one consisting of a sequence of $N_{j}$ observations, then the error is approximated as

$$E_{i}^{\text{NMSE}} = \frac{1}{N_{s}N_{j}} \sum_{k=1}^{N_{s}} \sum_{n=1}^{N_{j}} \frac{|z_{i,n}(k) - \hat{z}_{i,n}(k)|^2}{\sum_{j=1}^{M} |z_{i,j}(k)|^2},$$

(45)

Table 2 shows the values $E_{i}^{\text{NMSE}}$, $i = 1, \ldots, 5$, computed from $N_{s} = 200$ independent simulations, for each algorithm. With $M = 500$ particles, all methods
guarantee an adequate tracking of all variables and attain a similar performance (the error in the estimation of $z_{1,n}$ is only significantly higher for the SMCM algorithm with cost function $C_5$.).

From the perspective of the SMCM methodology, it is interesting to look into the relative performance when there is some discrepancy between the assumed model and the BF. In particular, we have simulated nonlinear observations of the form $y_n = \max\{10^{-4}, z_{1,n} + m_n\}$, with $m_n \sim N(0,10^{-5})$ as before. We have applied the five algorithms without any modification (in particular, the BF is still designed assuming linear observations) and evaluated the average error values, $E_{\text{AISE}}$, through another 200 independent simulation trials.

Our results, shown in Table 3, illustrate the superior robustness of the SMCM algorithms, which attain error values very similar to than those shown in Table 2, whereas the performance of the BF degrades considerably for $z_{1,n}$, $z_{2,n}$ and $z_{4,n}$.

6. Conclusions

We have revisited the sequential Monte Carlo minimization method originally introduced in [21,22,19]. As a result, we have:

- proposed an extended description of the methodology that allows to tackle the optimization of a larger class of cost functions and
- analyzed the asymptotic convergence of SMCM algorithms both with time and the number of particles.

Our results show that properly designed SMCM algorithms can generate, with high probability, arbitrarily accurate estimates of a sequence of cost minimizers. The proof, shown in Appendix A, is lengthy but based on elementary results and a simple induction argument (an intuitive sketch of the argument has been provided in Section 4.3). One remarkable consequence of the analysis is that the convergence of SMCM algorithms depends very weakly on the dynamics of the signal of interest. Indeed, virtually any function that provides a rough approximation of the evolution of the cost minimizer at time $t$ into the minimizer at time $t+1$ enables the design of a convergent algorithm. These results have been illustrated by way of a simple example involving a nonlinear one dimensional system. Finally, we have also considered the problem of tracking the latent physical variables in the dynamic model of a CO$_2$ laser. In this case, we have compared the tracking performance of four SMCM algorithms (built around different cost functions) and a standard particle filter. We have shown that, at least for this example, the SMCM algorithm is more robust to variations in the form of the observations than the particle filter.

---

2 This model makes practical sense because experimental observations also show such a “floor” effect, i.e., very small values of the intensity variable $x_1$, are not practically observable [18].

**Acknowledgements**

This work has been partially supported by the Ministry of Science and Innovation of Spain (Project MONIN, Ref. TEC-2006-13514-C02-01/TCM, and program Consolider-Ingenio 2010, Project CSD2008-00010 COMONSENS) and the Autonomous Community of Madrid (Project PROMULTIDIS-CM, Ref. S-0505/TIC/0233).

**Appendix A. Proof of Proposition 1**

In this appendix we address the formal proof of Proposition 1. We start with two auxiliary results, that involve the introduction of some new notation, and then proceed with the main proof. The latter is based on a simple induction argument and a classical inequality for the sum of bounded random variables.

**A.1. Auxiliary results**

We first use the sequence of functions $\{a_t\}_{t=1}^{\infty}$, to establish a connection between the dynamics of the minimizers, $\{x^n_t\}_{t=1}^{\infty}$, and the dynamics of an adequately defined sequence of sets. In particular, let $\{X_t\}_{t=1}^{\infty}$ be a sequence of sets in $\mathbb{R}^n$, and let us assume that they satisfy the following assumptions.

**Assumption 4.** For all $t \in \mathbb{N}$, $X_t = \bigcup_{i=1}^{n} O_i$, where $n_t$ is a (finite) natural number and $O_{1,t}, \ldots, O_{n_t,t}$ are bounded open cells in $\mathbb{R}^n$.

Let $A, B$ be proper subsets of $\mathbb{R}^n$ and let $d(x,x')$ denote a proper distance between points $x, x' \in \mathbb{R}^n$. We define the distance between $A$ and $B$ as

$$d(A,B) = \inf_{x \in A} d(x,x'),$$

and the complement of $A$ as $\overline{A} = \{ y \in \mathbb{R}^n : y \notin A \}$.

**Assumption 5.** For all $t \in \mathbb{N}$ we assume that $d(X_t, A_{t-1}(X_{t-1})) > 0$. Equivalently, $A_{t-1}(X_{t-1})$ is a proper subset (not equal to) $X_t$.

The class of sets that complies with Assumption 4 is very large. In particular, note that for any measurable set $E \subset \mathbb{R}^n$, such that $\mu(E) < \infty$, and for any $\varepsilon > 0$, there exists a sequence of bounded open cells $(O_1, \ldots, O_n)$, with $n < \infty$, such that the measure of the symmetric difference between the set $E$ and its approximation $E_n = \bigcup_{i=1}^{n} O_i$ is less than $\varepsilon$ [2, Theorem 15.10], i.e., $\mu(E \Delta E_n) = \mu((E - E_n) \cup (E_n - E)) < \varepsilon$.

Next, we introduce some notation needed to express the properties of the sequence $\{X_t\}_{t=1}^{\infty}$ relative to the sequence of minimizers $\{x^n_t\}_{t=1}^{\infty}$. The distance between a point $x \in \mathbb{R}^n$ and a set $X \subset \mathbb{R}^n$ is defined as $d(x,X) = \inf_{x \in X} d(x,x')$. In our case, it is of interest to upper bound the separation between a minimizer $x^n_{t+1}$ and the set $X_{t+1}$, given the previous distance $d(x^n_t, X_{t+1})$. We start with a bound on the departure between two arbitrary points $x, x' \in \mathbb{R}^n$ when they are propagated
through the function \(a_t\), namely

\[
D_{t+1}^{sup} = \sup_{x \in \mathbb{R}^n} d(a_t(x), a_t(x')) - d(x_t, x_{t+1}) \tag{A.2}
\]

while the corresponding upper bound for the distance increment is

\[
A_{t+1}^{sup} = D_{t+1}^{sup} - d(x_t, x_{t+1}) \tag{A.3}
\]

Note that, because of Assumption 3, if \(d(x_t, x_{t+1}) < \infty\) then the supremum in (A.2) is bounded, i.e., \(D_{t+1}^{sup} < \infty\). From Assumption 5, we can also define the sequence of positive real numbers

\[
d_{t+1}^{out} = d(x_{t+1}, a_t(x_t))^2 > 0, \quad t \in \mathbb{N}^*
\]

and, taking together (A.3), (A.4) and Assumption 2, we introduce the sequence

\[
K_{t+1} = d_{t+1}^{out} - A_t - A_{t+1}^{sup}, \quad t \in \mathbb{N}^*
\]

with maximum lower bound denoted as \(K = \inf_{t < n; t < K_t}\). It can be shown (see the proof of Lemma 1 below) that \(K_{t+1}\) is the worst-case reduction of the distance \(d(x_t, x_{t+1})\) that we achieve at time \(t+1\), i.e., \(K_{t+1} \leq d(x_t, x_{t+1}^*) - d(x_{t+1}, x_{t+1+1}^*)\).

**Lemma 1.** If the following assumptions hold for some sequence \(\{x_t^*\}_{t=1}^n\):

(a) \(K \geq 0\).

(b) There exists \(L \in (0, +\infty)\) such that \(d(x_t, x_{t+1}^*) \leq L\).

(c) Given \(L\), for all \(\varepsilon > 0\) there exists \(t_\varepsilon \in \mathbb{N}\) such that, for all \(t > t_\varepsilon\),

\[
\sum_{n=2}^t |K_n| < L - \varepsilon.
\]

Then, for all \(t > t_\varepsilon\), \(d(x_t, x_{t+1}^*) < \varepsilon\). Moreover, if \(L > d(x_t, x_{t+1}^*) + \varepsilon\) then \(d(x_t, x_{t+1}^*) = 0\) for all \(t > t_\varepsilon\).

**Proof.** From Assumption 5 in the definition of sequence \(\{x_t\}_{t=1}^n\), we deduce an initial upper bound of \(d(x_t, x_{t+1}^*)\), \(t \geq 2\),

\[
d(x_t, x_{t+1}^*) \leq \max[0, d(a_{t-1}(x_{t-1}), x_{t+1}^*) - d_{t-1}^{out}],
\]

which using the triangular inequality,

\[
d(a_{t-1}(x_{t-1}), x_{t+1}^*) \leq d(a_{t-1}(x_{t-1}), a_{t-1}(x_{t+1}^* - x_t)) + d(a_{t-1}(x_{t+1}^* - x_t), x_{t+1}^*)
\]

and Assumption 2 yields

\[
d(x_t, x_{t+1}^*) \leq \max[0, d(a_{t-1}(x_{t-1}), a_{t-1}(x_{t+1}^* - x_t)) + A_{t-1} - d_{t-1}^{out}].
\]

From the definition of \(D_{t+1}^{sup}\) and \(A_{t+1}^{sup}\), we have

\[
d(a_{t-1}(x_{t-1}), a_{t-1}(x_{t+1}^* - x_t)) \leq D_{t+1}^{sup} + d(x_{t+1} - x_{t+1}^* - x_t)
\]

and, substituting (A.9) into (A.8), we arrive at the inequality

\[
d(x_t, x_{t+1}^*) \leq \max[0, d(x_{t-1}, x_{t+1}^* - x_t) + A_{t-1} - d_{t-1}^{out}]
\]

and, since we have previously defined \(K_t = d_{t-1}^{out} - A_{t-1}\), we readily obtain

\[
d(x_t, x_{t+1}^*) \leq \max[0, d(x_{t-1}, x_{t+1}^* - x_t) - K_t].
\]
that, for all \( M_0 > M_{0,\varepsilon,\delta} \),
\[
P\left\{ d(x, x_0^{(k)}) < \frac{\varepsilon}{2} \quad \text{for some } k \in \{1, \ldots, M_0\} \right\} > 1 - \delta. \tag{A.13}
\]

Note that the event "\( d(x, x_0^{(k)}) < \varepsilon/2 \) for some \( k \in \{1, \ldots, M_0\} \)" implies the event "\( \min_{k \in \{1, \ldots, M_0\}} d(x, x_0^{(k)}) < \varepsilon/2 \)".

Next, we build the sequence \( \{X_t\}_{t \geq 0} \) recursively. Our induction hypothesis is that \( X_{t-1} \) is a finite Borel set that contains all discrete points \( \{x_0^{(m)}\}_{M_0 + 1} \) drawn from a pdf \( \rho_{t-1} \), where \( M_0 + 1 \) is a natural number (hence, \( M_0 = \infty \)). This is actually weaker than Assumption 4 but is sufficient for our argument. Let us define
\[
\eta \triangleq \sup_{x \in X_{t-1}, \eta \in \mathbb{R}^d} d(x, x_0^{(k)}). \tag{A.14}
\]

Since \( \lambda(X_{t-1}) < \infty \) by assumption, \( \eta < \infty \) and the set \( \{x_0^{(m)}\}_{M_0 + 1} \) is a finite \( \eta \)-net of \( X_{t-1} \) (recall that \( M_0 + 1 \) can be arbitrarily large but finite). As a consequence, it turns out that
\[
a_{t-1}(X_{t-1}) \subseteq \bigcup_{k = 1}^{M_0 + 1} a_{t-1}(B(x_0^{(k)}), \eta)). \tag{A.15}
\]

where \( B(x, \eta) \triangleq \{x' \in \mathbb{R}^d : d(x, x') < \eta\} \) denotes an open ball centered at \( x \) with radius \( \eta > 0 \). Moreover, if we construct \n\[
X_t \triangleq \bigcup_{k = 1}^{M_0 + 1} B(a_{t-1}(x_0^{(k)}), r_t), \tag{A.16}
\]

with sufficiently large \( r_t \), then we can write \n\[
a_{t-1}(X_{t-1}) \subseteq \bigcup_{k = 1}^{M_0 + 1} a_{t-1}(B(x_0^{(k)}), \eta)) \subseteq X_t. \tag{A.17}
\]

In particular, (A.17) holds if we take
\[
r_t > \sup_{x \in \mathbb{R}^d, d(x, x') < 2\eta} d(a_{t-1}(x), a_{t-1}(x')) \geq \sup_{x \neq x' \in X_{t-1}} \inf_{x'' \in \mathbb{R}^d} d(x, x'') < 2\eta \tag{A.18}
\]

(note that the suprema are finite because of Assumption 3). Given \( M_{t, \varepsilon, \delta} < \infty \) (A.16) and (A.18) ensure that (A.17) holds and, therefore, \( X_t \) complies with Assumptions 4 and 5.

We naturally associate the propagation pdf \( \rho_t \) defined in Eq. (21) to the set \( X_t \). Recall that, since we have assumed \( \zeta_t(k) > 0 \) for all \( k \in \{1, \ldots, M_{t-1}\} \) (which is possible because \( M_{t-1} < \infty \)), it follows that \( \rho_t \) is uniformly continuous and strictly positive in \( X_t \).

Let \( x_0^{(k)} \sim \rho_t(x), k = 1, \ldots, M_0 \). We need to prove that we can draw samples arbitrarily close to an arbitrary point \( x' \in X_t \). First, we note that, from the construction of \( X_t \) and the choice of \( \rho_t \) we obtain
\[
P\left\{ x_0^{(k)} \in B\left( x', \frac{\varepsilon}{2}\right) \right\} \triangleq \varepsilon_m > 0 \tag{A.19}
\]

(note that \( \lambda(X_t) < \infty \), since \( M_{t-1} < \infty \) and \( r_t < \infty \)). We also introduce the auxiliary random variables
\[
z_t^{(k)} \triangleq \begin{cases} 1 & \text{if } x_0^{(k)} \in B\left( x', \frac{\varepsilon}{2}\right), \\ 0 & \text{otherwise}, \end{cases} \tag{A.20}
\]

and the (random) sample mean \( Z_t \triangleq (1/M_t) \sum_{k = 1}^{M_t} z_t^{(k)} \leq 1 \).

The expected value of \( Z_t \) is non-zero, namely
\[
E[Z_t] = \frac{1}{M_t} \sum_{k = 1}^{M_t} E[z_t^{(k)}] = \varepsilon_m, \tag{A.21}
\]

where \( \varepsilon_m \) is independent of \( M_t \).

For conciseness, let \( Q(\varepsilon, M_t) \) denote the event "\( d(x_0^{(k)}, x') < \varepsilon/2 \) for some \( k \in \{1, \ldots, M_t\} \)". Obviously, \( Q(\varepsilon, M_t) \) is equivalent to the event "\( Z_t > 0 \)" and we can write
\[
P(\varepsilon, M_t) = P(Z_t > 0) > P(\varepsilon_m > \varepsilon) \tag{A.22}
\]

for any small constant \( 0 < \varepsilon < \varepsilon_m \). Therefore,
\[
P(\varepsilon, M_t) > 1 - P(\varepsilon_m > \varepsilon) \geq 1 - 2\exp(-2M_t\varepsilon^2), \tag{A.23}
\]

where we have applied one of Hoeffding’s inequalities [15, Theorem 1] to obtain the bound. Since \( \lim_{M_t \to \infty} \exp(-2M_t\varepsilon^2) = 0 \), for any \( \delta > 0 \) we can find \( M_{t, \varepsilon, \delta} \) such that, whenever \( M_t > M_{t, \varepsilon, \delta} \), the inequality \( 1 - 2\exp(-2M_t\varepsilon^2) > 1 - \delta \) is satisfied.

A.2.2. Application of Lemma 2

In order to apply Lemma 2 to the sequence \( \{X_t\}_{t \geq 0} \) we need to guarantee that \( d^{out} = d(x_t, a_{t-1}(X_{t-1})) \) be large enough. Specifically, the inequality
\[
d^{out} > A_{t-1} + A_{t-1}^{sup} \tag{A.24}
\]

must be satisfied for all \( t \). We recall that \( A_{t-1} = d(x_0^{(k)}, a_{t-1}(X_{t-1})) \) by Assumption 2 and \( A_{t-1}^{sup} \) is defined by (A.3) and (A.2). In our case, from the definition of \( d^{out} \), (A.16) and (A.18) we obtain that \( d^{out} \) must be chosen to satisfy the inequality
\[
d^{out} \geq r_t - \sup_{x \in \mathbb{R}^d, d(x, x') < 2\eta} d(a_{t-1}(x), a_{t-1}(x')). \tag{A.25}
\]

Since it is always possible to choose \( r_t \) such that
\[
r_t > \sup_{x \in \mathbb{R}^d, d(x, x') < 2\eta} d(a_{t-1}(x), a_{t-1}(x')) + A_{t-1}^{sup} + A_{t-1}, \tag{A.26}
\]

then it is also always possible to make the inequality (A.24) hold for all \( t \) and, as a consequence, \( K = \inf_{t \in \mathbb{N}} d^{out} - A_{t-1} - A_{t-1}^{sup} > 0 \).

Therefore, we can apply Lemma 2 to prove the existence of \( t_0 \in \mathbb{N} \) such that, for all \( t > t_0 \),
\[
d(x_t, x_0^{(k)}) < \frac{\varepsilon}{2} \tag{A.27}
\]

Also, from the definition of the sequences \( \{X_t\}_{t \geq 0} \) and \( \{\rho_t\}_{t \geq 0} \), there exists \( M_{t, \varepsilon, \delta} \) such that, for all \( M_t > M_{t, \varepsilon, \delta} \) and all \( x \in X_t \),
\[
P(\varepsilon, M_t) > 1 - \delta. \tag{A.28}
\]

Taking together (A.27) and (A.28), we arrive at
\[
P\left\{ d(x_0^{(k)}, x') < \varepsilon/2 \quad \text{for some } k \in \{1, \ldots, M_t\} \right\} > 1 - \delta. \tag{A.29}
\]

Since the event "\( d(x_0^{(k)}, x') < \varepsilon \) for some \( k \in \{1, \ldots, M_t\} \)" implies the event "\( \min_{k \in \{1, \ldots, M_t\}} d(x_0^{(k)}, x') < \varepsilon/2 \)", the proof is complete.
Appendix B. The probability measure for the random vector $\mathbf{x}_t$

Recall that the sequence of observations $\mathbf{y}_{1:t}$ is fixed, hence the randomness in the SMCM method comes exclusively from the sampling process using the sequence of pdf's $\rho_t$ defined in Eq. (21).

Let us put together the samples at time $t$ in order to construct the random vector $\mathbf{x}_t^t = [\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_t^{(M_t)}]^\top \in \mathbb{R}^{n_t M_t}$. At time $t = 0$, $\mathbf{x}_0^0 \sim \rho_0^0(\mathbf{x}_0^0) = \prod_{i=1}^{M_0} \rho_0(\mathbf{x}_i^0)$, while, at time $t$ and given $\mathbf{x}_{t-1}^t$, $\mathbf{x}_t^t \sim \rho_t^t(\mathbf{x}_t^t) = \prod_{i=1}^{M_t} \sum_{k=1}^{M_{t-1}} \rho_t^t(\mathbf{x}_i^{(k)}|\mathbf{x}_{t-1}^t) f_t(k)$. The process $\{\mathbf{x}_t^t\}_{t\geq 0}$ is first-order Markovian and, as a consequence, we can write down the joint pdf for the sequence $\mathbf{x}_{0:t}$, namely

$$
\rho_0^t(\mathbf{x}_{0:t}^t) = \prod_{i=0}^{t} \rho_0^t(\mathbf{x}_i^t). \tag{B.1}
$$

The event “$d(\mathbf{x}_i^t, \mathbf{x}_{ij}^t) < \epsilon$” in Proposition 1 is the set $\mathcal{A}^e \in \mathfrak{B}^{n(M_{t-1} + n_t)}$ defined by Eq. (23), hence, $P(d(\mathbf{x}_i^t, \mathbf{x}_{ij}^t) < \epsilon) = P^E_\rho(\mathcal{A}^e)$. 

References