Evolutionary Particle Filters: Model-free Object Tracking
Combining Evolution Strategies and Particle Filters

Silja Meyer-Nieberg, Erik Kropat and Stefan Pickl

Department of Computer Science, Universität der Bundeswehr München,
Werner-Heisenberg Weg 37, 85577 Neubiberg, Germany

Keywords: Tracking, Dynamical Systems, Particle Filter, Dynamic Optimization, Evolutionary Algorithms.

Abstract: Tracking situations or more generally state estimation of dynamic systems arise in various application contexts. Usually the state-evolution equations are assumed to be known up to certain parameters. But what can be done if this is not the case? This paper presents an innovative approach to solve this difficult and complex situation by using the inherent tracking abilities of evolution strategies. Combining principles of particle filters and evolution strategies leads to a new type of algorithms: evolutionary particle filters. Their tracking quality is examined in simulations.

1 INTRODUCTION

Particle filters play an important role in the estimation of dynamical systems in many areas ranging from engineering over financial modeling to physical and biological systems. Further applications include tracking and localization tasks, for instance, estimating autonomous robot positions via GPS measurements or tracking the position of air planes in air traffic control. But what can be done if the evolution equations for the state variables are unknown? Problems like these appear for instance in ballistic target tracking or hand-held GPS-receivers (Johansson and Lehmann, 2009). In this case, we cannot apply common particle filters since these usually require the knowledge of the complete probabilistic model. The present paper investigates the potential direct use of the inherent tracking ability of evolution strategies, a specific evolutionary algorithm. In the area of dynamic optimization, it was shown in (Arnold and Beyer, 2006) that evolution strategies can follow moving optimizers which should allow their application to model-free tracking in principle. The question remains, however, which measure should be used to guide the search if information on the system is scarce.

The paper is organized as follows: It starts with a general description of the dynamic estimation problem and gives a concise sketch on particle filters. Afterwards, evolution strategies with the current state-of-the-art adaptation of the search direction and step-size are introduced. This is followed by a brief review on related approaches, that is, either publications using evolutionary and related algorithms in the area of particle filtering or approaches where it was not assumed that the state model is completely known. Afterwards, the main ideas for the new algorithms are presented before they are investigated closer in the experimental section. Conclusions and potential follow-up work constitute the last part of the paper.

1.1 A Dynamic Estimation Problem and Particle Filters

Tracking applications consider variants of the general system

$$\begin{aligned} \mathbf{x}_{k+1} &= f_{k+1}(\mathbf{x}_k, \vec{\varepsilon}_{k+1}) \\ \mathbf{z}_{k+1} &= h_{k+1}(\mathbf{x}_{k+1}, \vec{\omega}_{k+1}) \end{aligned} \tag{1}$$

(Arulampalam et al., 2002) describing the evolution of the state variables $\mathbf{x}_k$ and the sequence of the measurements $\mathbf{z}_k, k \geq 1$. The state variables denote the "true" position of the target which cannot be obtained directly. Instead only noisy measurements $\mathbf{z}_k$ can be taken from which the position has to be inferred. The random variables $\vec{\varepsilon}_k$ and $\vec{\omega}_k$ denote measurement and process noise, respectively and are assumed to be independent. In tracking applications, the aims are to obtain the posterior density $p(x_k|\mathbf{z}_{1;k})$, with $p(\mathbf{x}_k|\mathbf{z}_{1;k}) := p(\mathbf{x}_k|\mathbf{z}_1, \ldots, \mathbf{z}_k)$, statistical estimates of interesting characteristics, or to derive certain statistical moments - for instance the mean of the target position. Usually, the non-linear functions f_k, h_k, $k \geq 1$ are assumed to be known.

Meyer-Nieberg S., Kropat E. and Pickl S..
Evolutionary Particle Filters: Model-free Object Tracking - Combining Evolution Strategies and Particle Filters.
DOI: 10.5220/0004284300960102
In *Proceedings of the 2nd International Conference on Operations Research and Enterprise Systems* (ICORES-2013), pages 96-102
ISBN: 978-989-8565-40-2

Following a Bayesian approach, the posterior density is obtained in two steps: a prediction and an update step. The prediction step obtains the density of the present target position given the past measurements

$$p(\mathbf{x}_k|\mathbf{z}_{1;k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1;k-1})\mathrm{d}\mathbf{x}_{k-1}. \quad (2)$$

The required posterior can be given once new measurements are made as

$$p(\mathbf{x}_k|\mathbf{z}_{1;k}) = \frac{p(\mathbf{x}_k|\mathbf{z}_{1;k-1})p(\mathbf{z}_k|\mathbf{x}_k)}{p(\mathbf{z}_k|\mathbf{z}_{1;k-1})} \quad (3)$$

with

$$p(\mathbf{z}_k|\mathbf{z}_{1;k-1}) := \int p(\mathbf{z}_\mathbf{k}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1;k-1})\mathrm{d}\mathbf{x}_k. \quad (4)$$

However, exact analytical solutions for (3) exist only if quite restrictive assumptions are fulfilled. Otherwise approximations are required. One of these is the well-known particle filter which is described shortly in the following. For a more detailed description, we refer to (Doucet and Johansen, 2011). It can be shown that the density can be approximated

$$p(\mathbf{x}_k|\mathbf{z}_{1;k}) \approx \sum_{i=1}^{N_s} w_{i,k}\delta(\mathbf{x}_k - \mathbf{x}_{i,k}) \quad (5)$$

using N_s particles $\mathbf{x}_{i,k}$ which means that the sum converges to the density for $N_s \to \infty$. The symbol $\delta(\mathbf{u})$ denotes the Dirac delta function and the $w_{i,k}$ are positive weights summing up to one.

It can be shown (Arulampalam et al., 2002) that the weights can be updated according to

$$w_{i,k} \quad = \quad w_{i,k-1}\frac{p(\mathbf{z}_k|\mathbf{x}_{i,k})p(\mathbf{x}_{i,k}|\mathbf{x}_{i,k-1})}{q(\mathbf{x}_{i,k}|\mathbf{x}_{i,k-1},\mathbf{z}_k)} \quad (6)$$

with $q(\mathbf{x}_{i,k}|\mathbf{x}_{i,k-1},\mathbf{z}_k)$, the importance density, to be defined. Since the performance of the particle filter depends strongly on the importance density its good choice is critical. Common choices include e.g. the transition density $q(\mathbf{x}_k|\mathbf{x}_{i,k},\mathbf{z}_k) = p(\mathbf{x}_k|\mathbf{x}_{i,k})$ although this can lead to problems. Several approaches exist which differ in the preconditions they make. Most approaches assume that the state evolution equations are known. Only a few publications exist which address the problem of unknown parameters.

In this paper, we assume that the functions describing the evolution of the non-measurable state variables are unknown. Therefore, it is not possible to compute the transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ explicitly. Only the measurements can be taken into account to derive and predict the target position.

1.2 Evolution Strategies

Evolutionary algorithms (EAs) are population-based stochastic search and optimization algorithms. They construct an implicit probabilistic model based on good candidate solutions (parent population). The model is then used to create new solutions (the offspring) which are incorporated into the population if they are sufficiently good. An evolutionary algorithm starts with an initial population which may be drawn randomly. Evolution strategies (ESs) (Beyer and Schwefel, 2002) are a variant of evolutionary algorithms that is predominantly applied for continuous search spaces. Mutation, that is, movement according to random perturbations, is the main search operator. Recombination is usually performed by calculating the weighted mean of the μ parents, although other forms exist (Beyer and Schwefel, 2002). The result is then mutated – usually by adding a normally distributed random variable with zero mean and covariance matrix $\sigma^2\mathbf{C}$. Afterwards, the individuals are evaluated using the function to be optimized or a derived function which allows an easy ranking of the population. An important topic in evolution strategies is the continuous adaptation of the covariance matrix. Evolution strategies with ill-adapted parameters show slow convergence or are unable to find the optimal state at all. Therefore, methods for adapting the scale factor σ or the full covariance matrix have received a lot of attention (see (Meyer-Nieberg and Beyer, 2007)) – cumulating in evolution strategies with covariance matrix adaptation.

1.3 Updating the Covariance Matrix

First, the update of the covariance matrix is addressed. In evolution strategies two types exist: one used by the *covariance matrix adaptation evolution strategy* (CMA-ES) (Hansen, 2006) which considers past information from the search and an alternative used by the *covariance matrix self-adaptation evolution strategy* (CMSA-ES) (Beyer and Sendhoff, 2008) which takes only present information into account.

The covariance matrix update of the CMA-ES is explained first. The CMA-ES uses weighted intermediate recombination, in other words, it computes the weighted centroid of the μ best individuals of the population. This mean $\mathbf{m}^{(g)}$ is used for creating all offspring by adding a random vector drawn from a normal distribution with covariance matrix $(\sigma^{(g)})^2\mathbf{C}^{(g)}$, i.e., the actual covariance matrix consists of a general scaling factor (step-size) and the matrix denoting the directions. Following usual notation in evolution strategies this matrix $\mathbf{C}^{(g)}$ will be referred to as *co-*

variance matrix in the following.

The basis for the CMA update is the common estimate of the covariance matrix using the newly created population. Instead of considering the whole population for building the estimates, though, it introduces a bias towards good search regions by taking only the μ best individuals into account. Furthermore, it does not estimate the mean anew but uses the weighted mean $\mathbf{m}^{(g)}$. Following (Hansen, 2006),

$$\mathbf{y}_{m:\lambda}^{(g+1)} := \frac{1}{\sigma^{(g)}}\left(\mathbf{x}_{m:\lambda}^{(g+1)} - \mathbf{m}^{(g)}\right)$$

are determined with $\mathbf{x}_{m:\lambda}$ denoting the mth best off the λ particle according to the fitness ranking. The rank-μ update then obtains the covariance matrix as

$$\mathbf{C}_{\mu}^{(g+1)} \quad := \quad \sum_{m=1}^{\mu} w_m \mathbf{y}_{m:\lambda}^{(g+1)} (\mathbf{y}_{m:\lambda}^{(g+1)})^{\mathrm{T}} \qquad (7)$$

To derive reliable estimates larger population sizes are usually necessary which is detrimental with regard to the algorithm's speed. Therefore, past information, that is, past covariance matrizes are usually also considered with parameter c_μ determining the effective time-horizon. In CMA-ES it has been found that enhancing the general search direction in the covariance matrix is usual beneficial. For this, the concepts of the *evolutionary path* and the *rank-one-update* are introduced. As its name suggests, an evolutionary path considers the path in the search space the population (i.e., the weighted mean) has taken so far. The evolutionary path $\mathbf{p}_c$ gives a general search direction that the ES has taken in the immediate past. In order to bias the covariance matrix accordingly, the rank-one-update is used

$$\mathbf{C}_{1}^{(g+1)} \quad := \quad \mathbf{p}_c^{(g+1)} (\mathbf{p}_c^{(g+1)})^{\mathrm{T}}. \qquad (8)$$

Together the components constitute the covariance update of the CMA-ES

$$\mathbf{C}^{(g+1)} := (1 - c_1 - c_\mu)\mathbf{C}^{(g)} + c_1 \mathbf{C}_1^{(g+1)} + c_\mu \mathbf{C}_\mu^{(g+1)}$$

see (Hansen, 2006) for details. The CMA-ES is one of the most powerful evolution strategies. However, as pointed out in (Beyer and Sendhoff, 2008), its scaling behavior with the population size is not good. The CMSA-ES (Beyer and Sendhoff, 2008) updates the covariance matrix differently by considering

$$\mathbf{y}_{m:\lambda}^{(g+1)} \quad := \quad \frac{1}{\sigma_m^{(g+1)}}\left(\mathbf{x}_{m:\lambda}^{(g)} - \mathbf{x}_p^{(g)}\right) \qquad (9)$$

with $\mathbf{x}_p^{(g)}$ the base vector of the mutation leading to $\mathbf{x}_{m:\lambda}^{(g+1)}$. Using (weighted) recombination, Eq. (9) equals the rank-μ update of the CMA-ES. The covariance update then reads

$$\begin{aligned}\mathbf{C}^{(g+1)} \quad := \quad & (1 - \frac{1}{c_\tau})\mathbf{C}^{(g)} + \\ & \frac{1}{c_\tau}\sum_{m=1}^{\mu} w_m \mathbf{y}_{m:\lambda}^{(g+1)} (\mathbf{y}_{m:\lambda}^{(g+1)})^{\mathrm{T}} \end{aligned} \qquad (10)$$

with the weights usually set to $w_m = 1/\mu$. See (Beyer and Sendhoff, 2008) for information on the free parameter c_τ.

1.4 Updating the Step-size

The CMA-ES uses the so-called *cumulative step-size adaptation* (CSA) to adapt the scaling parameter (also called *step-size*, *mutation strength* or *step-length*). To this end, the CSA (Hansen, 2006) determines again an evolutionary path $\mathbf{p}_\sigma$ by summing up the movement of the population centers eliminating the influence of the covariance matrix and the step length. For a detailed description see (Hansen, 2006). The length of the path $\mathbf{p}_\sigma$ is important. If the path length is short, several movements of the centers counteract each other which indicates that the step-size is too large and should be reduced. If on the other hand, the ES takes steps approximately in the same direction, progress and algorithm speed would be improved, if the ES could make larger changes. Therefore, long path lengths are seen as an indicator for a required increase of the step length. Ideally, the CSA should result in uncorrelated steps leading to

$$\ln(\sigma^{(g+1)}) = \ln(\sigma^{(g)}) + \frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_\sigma^{(g+1)}\| - \mu_{\chi_n}}{\mu_{\chi_n}}\right) \qquad (11)$$

as the CSA-rule. The parameter μ_{χ_n} in (11) stands for the mean of the χ-distribution with n degrees of freedom and serves as the ideal value for the comparison. It can be shown that the original CSA encounter problems in large noise regimes. Therefore, uncertainty handling procedures and other safeguards are recommended. An alternative approach for adapting the step-size is *self-adaptation* mainly developed in (Schwefel, 1981). It subjects the strategy parameters of the mutation to evolution. In other words, the scaling parameter or in its full form, the whole covariance matrix, undergoes recombination, mutation, and indirect selection processes. The working principle is based on an indirect stochastic linkage between good individuals and appropriate parameters: On average good parameters should lead to better offspring than too large or too small values or mis/leading directions. Today, self-adaptation is used mainly to adapt the step-size or a diagonal covariance matrix.

In the case of the mutation strength, usually a log-normal distribution $\sigma_l^{(g)} = \sigma_{\text{base}} \exp(\tau \mathcal{N}(0,1))$ is used for mutation. The parameter τ is called the *learning rate*. The variable σ_{base} is either the parental mutation strength or the result of recombination. For the step-size, it is possible to apply the same type of recombination as for the positions although different forms – for instance a multiplicative combination – could be used instead. The self-adaptation of the step-size is referred to as σ-*self-adaptation* (σSA) in the remainder of the paper. The newly created mutation strength is then directly used in the mutation of the offspring. If the offspring is sufficiently good, it is passed to the next generation. The baseline σ_{base} is either the mutation strength of the parent or – if recombination is used – the recombination result. Self-adaptation with recombination has been shown to be "robust" against noise (Beyer and Meyer-Nieberg, 2006) and is used in the CMSA-ES (Beyer and Sendhoff, 2008) as update rule for the scaling factor.

2 EVOLUTION STRATEGIES FOR MODEL-FREE TRACKING

This paper provides – to our knowledge – the first attempt to use an ES directly for tracking tasks. We propose to adapt evolution strategies to object tracking by guiding the search using the remaining available information. Since the state evolution equations are unknown, tracking can only take the measurements and the observation equation into account. We assume that it is possible to evaluate $p(\mathbf{z}_k|\mathbf{x})$ point-wise, for $\mathbf{z}_k$ given, since we will use this density as the fitness function

$$f_k(\mathbf{x}) \quad = \quad p(\mathbf{z}_k|\mathbf{x}) \tag{12}$$

to guide the search of the evolution strategy. One of the aims of the paper is to investigate whether this approach is feasible. Since the measurements are overlaid with noise, information from the search so far is valuable to avoid moving towards false optima provided the true positional changes of the state variables are not too large. We postulate that it is possible to recover the target movement to some extend by considering the search process of the ES. The maximizer of (12) or of some derivatives of (12) should provide a rough estimate on the target position. The dynamical nature of the optimization problem should keep the ES's population from collapsing to a point solution. An ES that tries to optimize the dynamic problem should provide a better guess for the true position especially as it keeps statistics of past searches and therefore is influenced by $\mathbf{z}_1, \ldots, \mathbf{z}_k$.

Two main approaches are considered, see Figs. 1 and 2. The first called evolutionary particle filter (EPF) does not use recombination similar to usual particle filters. Instead it selects one particle of the parent population as the mean of the mutation vector. The second approach will use intermediate recombination and is notated as EPF_{rec}. Both will be combined with CMA and CMSA – adapted for object tracking. Let us first address the EPF. It allows

Algorithm 1:
Evolutionary Particle Filter
g=0: Initialize $\sigma_m^{(0)} = \sigma_{m;k-1}$, $\mathbf{C}_m^{(0)} = \mathbf{C}_{m;k-1}$, $\mathbf{x}_m^{(0)} = \mathbf{x}_{m;k-1}$, $w_m^{(0)} = w_{m;k-1}$, $m = 1, \ldots, \mu$.
REPEAT
Draw λ parents $\mathbf{x}_m^{(g)}$ with resampling (uniform or according to weights)
Determine $\sigma_m^{(g)}$, $\mathbf{C}_m^{(g)}$
Set $w_l^{(g)} = w_m^{(g)}$
Create λ particles according to

$$\mathbf{x}_\lambda^l = \mathbf{x}_m^{(g)} + \sigma_m^{(g)} \mathcal{N}(\mathbf{0}, \mathbf{C}_m^{(g)}) \tag{13}$$

Determine the fitness

$$f(\mathbf{x}_\lambda^l) = w_l^{(g)} p(\mathbf{z}_k|\mathbf{x}_\lambda^l) \tag{14}$$

Select the μ-best particles
Calculate the weights

$$w_m^{(g+1)} = \frac{f(\mathbf{x}_m^{(g+1)})}{\sum_{k=1}^{\mu} f(\mathbf{x}_k^{(g+1)})} \tag{15}$$

$g \to g+1$
UNTIL STOP
$w_{m;k} = w_m^{(g_{end})}$, $\sigma_{m;k-1} = \sigma_m^{(g_{end})}$, $\mathbf{C}_{m;k} = \mathbf{C}_m^{(g_{end})}$, $\mathbf{x}_{m;k} = \mathbf{x}_m^{(g_{end})}$, $m = 1, \ldots, \mu$

Figure 1: Evolutionary particle filter without recombination.

a slightly different approach since its base vector is a concrete point of the parent population: Setting the importance density in (6) equal to the transition density leads to the weight update $w_{ik} = w_{ik-1} p(\mathbf{z}_k|\mathbf{x}_{ik})$ which can be used as alternative to the fitness (12) for the ES in the evolutionary particle filter in Figs. 1. In this way past measurements influence the selection via the old weights. The EPF_{rec} strategy uses recombination which circumvents the usage of previous weights. Therefore, Eq. (12), $p(\mathbf{z}_k|\mathbf{x})$, will serve as fitness. Recombination can be realized in several ways. In ES, static weights are common. This is in contrast to typical procedures in particle filters which use a fitness dependent relative weight. Investigating from which procedure the algorithms benefitbest or whether a combination shall be applied is an interesting point for further research. The parameters of the normal distribution are updated by adapting the concepts of the CMA-ES and the CMSA-ES to the task at hand. New measurements can arrive during optimization leading to a dynamic optimization problem. In the case of evolution strategies, problems may appear if the magnitude of the change is very large which will lead to a longer adaptation times if the scaling factors have become too small. This will occur especially if measurements are taken infrequently. We assume that in general convergence of the ES will take longer than the appearance of new measurements (this can be forced by limiting the search time of the ES).

Since both, the CMA and the CMSA, rely on past information, finding an appropriate generation time window will probably important. This paper considers the performance of systems with short generation time horizons which will require larger populations. The ESs are running systems, that is, they are started in the beginning of the measurements and incorporate new information as soon as a generation cycle is complete. In the following, the adaptation of the covariance matrix and step-size update are discussed.

In the case of the EPF_{rec}, the common ES-update rules appear appropriate. Using the EPF necessitates changes to the covariance update. The first step in the covariance update is the rank-μ update with the mean of the previous population as the base point. This, however, does not have a good justification if recombination is not used. The population mean of the new population could be used, instead. However, the question arises whether this method should be applied at all since there are different base points for the mutation for each offspring. Alternatives will be investigated in further research. The question remains whether the evolutionary path which considers the movement of the population center is suitable for the covariance update in case of the EPF-algorithm. Again, this will be investigated experimentally, however on first sight, path information as direction appears more suitable for EPF_{rec} than EPF.

The update of the scaling parameter remains to be addressed. As pointed out, two main approaches exist for ESs: the CSA-rule and the σSA-rule. Again, the concept of an evolutionary path lends itself more easily to the EPF_{rec} than to the EPF. In the latter case, the question occurs whether the movement of the mth best individual might be used, instead. The information from this path evolution will probably be overlaid by large stochastic fluctuations requiring possible longer time horizons. In a first approach, the usual CSA-rule with small time horizon will be used. The σSA-rule requires not any changes and could be transferred directly to both main EPF-types. However, the finding the most suitable recombination form for the step-size is an important research task.

Algorithm 2:
Evolutionary Particle Filter with Recombination
g=0: Initialize $\sigma^{(0)} = \sigma_{\mu;k-1}$, $\mathbf{C}^{(0)} = \mathbf{C}_{k-1}$, $\mathbf{x}_m^{(0)} = \mathbf{m}_{k-1}$, $w_m^{(0)} = w_{m;k-1}$, $m = 1, \ldots, \mu$.
REPEAT
Compute the weighted mean

$$\mathbf{m}^{(g)} = \sum_{m=1}^{\mu} w_m^{(g)} \mathbf{x}_m^{(g)} \tag{16}$$

Create λ particles according to

$$\mathbf{x}_\lambda^l = \mathbf{m}^{(g)} + \sigma^{(g)} \mathcal{N}(\mathbf{0}, \mathbf{C}^{(g)}) \tag{17}$$

Determine the fitness

$$f(\mathbf{x}_\lambda^l) = p(\mathbf{z}_k | \mathbf{x}_\lambda^l) \tag{18}$$

Select the μ-best particles
Calculate the weights

$$\text{a)} \quad w_m^{(g+1)} = \frac{1}{\mu} \tag{19}$$

$$\text{b)} \quad w_m^{(g+1)} = \ln\left(\frac{\mu+1}{2}\right) - \ln(m) \tag{20}$$

Determine $\sigma^{(g+1)}$, $\mathbf{C}^{(g+1)}$
$g \to g+1$
UNTIL STOP
$w_{m;k} = w_m^{(g_{end})}$, $\sigma_\mu = \sigma_\mu^{(g_{end})}$, $\mathbf{C}_k = \mathbf{C}^{(g_{end})}$, $\mathbf{m}_k = \mathbf{m}^{(g_{end})}$, $m = 1, \ldots, \mu$

Figure 2: Evolutionary particle filter (EPF_{rec}) using recombination.

We will use the following notation in the remainder of the paper. Let EPF-CMSA denote the evolutionary particle filter without recombination using the update rules of the CMSA-ES, EPF_{rec}-CMSA denote the algorithm with recombination and the CMSA-rule, whereas EPF-CMA and EPF_{rec}-CMA apply the rules from the CMA-ES most notably the cumulative search path adaptation.

3 EXPERIMENTS

This paper is devoted to an investigation as to whether evolution strategies can be applied in a tracking where the state evolution equations are unknown and aims at providing a proof-of-concept. Therefore, finding optimal parameter settings for the algorithms is not attempted and is left for further work.

3.1 Experimental Set-up

The algorithms have several parameters that can and should normally be tuned for a practical use. The experiments presented will start from the default values in CMA-ES and CMSA-ES and conduct only limited investigations into finding better parameters. The following questions are addressed in our first investigation of the algorithms using two simple dynamic models: Are evolution strategies able to track the moving target without knowledge of the true positions and state equations? This leads to the next topics: Are there differences in the tracking quality between CMA and CMSA using EPF_{rec}? Are there differences between the tracking using EPF and EPF_{rec}?

Following particle filter literature, the average root mean squared error (average RMSE) is considered together with the relative success frequency P_{succ} of having a tracking distance below a given threshold. In this paper, we investigate the algorithms using two simple systems. A one-dimensional system which is inspired by (Uosaki and Hatanaka, 2005), where only the step-size adaptation mechanism is investigated and a two-dimensional system which requires the adaptation of a diagonal covariance matrix. For a proof-of-concept we consider the following systems: The first, one-dimensional, dynamical system is given by

$$\begin{aligned} x_t &= \frac{x_{t-1}}{2} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2t) + 10\mathcal{N}(0,1) \\ z_t &= \text{sign}(x_t)\frac{x_t^2}{20} + \mathcal{N}(0,1) \end{aligned} \tag{21}$$

with the maximum of $p(z_i|x)$ directly discernible. No covariance matrix is necessary, just step-size adaptation is required. A two-dimensional system is defined by

$$\begin{aligned} \mathbf{x}_k &= \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \mathbf{x}_{k-1} + \\ & \begin{pmatrix} 8\cos(1.2t) \\ 8\sin(1.2t) \end{pmatrix} + \begin{pmatrix} \frac{25x_{1k-1}}{1+x_{1k-1}^2} \\ \frac{25x_{2k-1}}{1+x_{2k-1}^2} \end{pmatrix} + \begin{pmatrix} 5\mathcal{N}(0,1) \\ 5\mathcal{N}(0,1) \end{pmatrix} \\ \mathbf{z}_k &= \begin{pmatrix} \text{sign}(x_{1k})\frac{x_{1t}^2}{20} + \mathcal{N}(0,1) \\ \text{sign}(x_{2k})\frac{x_{2t}^2}{20} + \mathcal{N}(0,1) \end{pmatrix}. \end{aligned} \tag{22}$$

Again, optimal points of the density can be obtained easily.

3.2 Discussion

In a first analysis, the common CMA-ES and CMSA-ES versions were applied with the usual parameter setting excepting the population size λ and the learning rates of the CMSA-ES. The EPF_{rec}-CMA-ES uses $\mu = \lambda/2$ parents applying weighted recombination, whereas the EPF_{rec}-CMSA-ES uses $\mu = \lambda/4$ parents with equal weights. The fitness independent weights enable the use of a different fitness function instead of the normal probability density in (21) as long as the optimizer remains the same. The offspring population size was set to $\lambda = 100$ which is quite large for ESs but relatively small for particle filters. Preliminary runs revealed that in the case of the first dynamical system, good results were obtainable for small population sizes around $\lambda = 10$ for the CMA-variant whereas self-adaptation requires larger population sizes. In the case of the second system, larger populations appear necessary. The learning rates τ and τ_c have to be adjusted. In a first approach, the parameters were set to $\tau = 1/\sqrt{10}$ and $\tau_c = 1/(\mu N)$. As a safeguard against strong statistical fluctuations in the case of the EPF_{rec}-CMSA-ES, the mean of the mutation strengths was substituted by the median since this is the more robust estimate. We allowed three generation cycles before new measurements arrived. We conducted 20 different runs of the dynamical system (21) simulating the system for $k_{\max} = 100$ movements/measurements. For each of these realizations, we conducted 30 runs of the EAs. A tracking is called successful, if the distance between the true value and the algorithmic estimate is smaller than one. Overall, we find that the EPF_{rec}-CMA realizes lower average root mean squared errors. In the case of dynamical system (21), the minimal average mean squared error is 7.96, the median lies at 9.31 and the maximal average error reads 11.96. In contrast, the values read 13.52, 19.57, and 24.65 for EPF_{rec}-CMSA. We assume that this may be due either to the parameter setting chosen or that the stronger stochastic influences on the CMSA-ES have lead to stronger fluctuations in the tracking quality and some extreme values. Concerning the success frequency, an interesting finding emerges. For the first dynamical system, the EPF_{rec}-CMSA has success frequencies between 0.32 and 0.39 with an average RMSE during tracking time of around 0.17-0.18 whereas the success frequency is below 0.1 for the CMA-version.

In the case of the two-dimensional system (22), both strategies benefit from larger population sizes and longer generation times until new measurements arrive. The number of runs of the dynamical system was reduced to ten and 30 repeats were used. In contrast to the system (21), the EPF_{rec}-CMSA variant lead to better results (see Tables (1) and (2)).
First experiments with the EPF-strategy revealed stability problems if a fitness dependent weight was

Table 1: EPF$_{rec}$-CMA: Average RMSE-values (best and worst) for ten different runs of the system (22), each averaged over 30 repeats.

λ	g	best	worst
100	1	13.221	15.755
100	2	12.612	14.769
100	3	12.076	15.469
100	4	12.161	14.795
100	5	12.706	14.746
200	1	14.733	16.843
200	2	14.977	17.796
200	3	15.93	19.895
200	4	15.991	19.721
200	5	16.424	19.956

Table 2: EPF$_{rec}$-CMSA: Average RMSE-values for ten different runs of the system (22), each averaged over 30 repeats.

λ	g	best	worst
100	1	10.335	9.263
100	2	7.39	8.778
100	3	6.072	8.312
100	4	5.54	6.958
100	5	5.592	7.632
200	1	8.829	10.576
200	2	6.505	8.156
200	3	6.358	7.924
200	4	5.81	7.428
200	5	5.832	6.870

used. Therefore, usual rank-dependent weights of the CMSA-ES were applied. We compared the performance of the EPF-strategy with the EPF$_{rec}$-CMSA for system (21), that is, without adapting the covariance matrices. The experiments reveal no advantage for using the EPF strategy. Instead the RMSE is always larger than the error for the EPF$_{rec}$-CMSA. Not using recombination probably requires larger populations in order to cope with stochastic fluctuations.

4 CONCLUSIONS

This paper presents new approaches for state estimation in dynamical systems. State estimations of dynamical systems have various application areas ranging from tracking and localization tasks in autonomous robots to air traffic control and fault detection. In contrast to nearly all approaches, the paper does not assume that the probabilistic model governing the evolution of the state variables of the system is completely known. Instead it assumes that the evolution equation cannot be recovered analytically and only measurement information arrives during the tracking. We propose the use of evolution strategies which have been successfully applied to noisy and dynamic optimization. The paper presents a proof-of-concept applying selected evolutionary particle filters to simple dynamical systems. Since the first experiments were successful, in-depth investigations will be carried out in order to fine-tune the approaches.

REFERENCES

Arnold, D. V. and Beyer, H.-G. (2006). Optimum tracking with evolution strategies. *Evolutionary Computation*, 14:291 – 308.

Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174 –188.

Beyer, H.-G. and Meyer-Nieberg, S. (2006). Self-adaptation of evolution strategies under noisy fitness evaluations. *Genetic Programming and Evolvable Machines*, 7(4):295–328.

Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strategies: A comprehensive introduction. *Natural Computing*, 1(1):3–52.

Beyer, H.-G. and Sendhoff, B. (2008). Covariance matrix adaptation revisited - the CMSA evolution strategy -. In Rudolph, G. et al., editors, *PPSN*, volume 5199 of *Lecture Notes in Computer Science*, pages 123–132. Springer.

Doucet, A. and Johansen, A. M. (2011). A tutorial on particle filtering and smoothing: Fifteen years later. In Crisan, D. and Rozovsky, B., editors, *Oxford Handbook of Nonlinear Filtering, 2011,*. Oxford University Press.

Hansen, N. (2006). The CMA evolution strategy: A comparing review. In Lozano, J. et al., editors, *Towards a new evolutionary computation. Advances in estimation of distribution algorithms*, pages 75–102. Springer.

Johansson, A. and Lehmann, E. (2009). Evolutionary optimization of dynamics models in sequential monte carlo target tracking. *Evolutionary Computation, IEEE Transactions on*, 13(4):879 –894.

Meyer-Nieberg, S. and Beyer, H.-G. (2007). Self-adaptation in evolutionary algorithms. In Lobo, F., Lima, C., and Michalewicz, Z., editors, *Parameter Setting in Evolutionary Algorithms*, pages 47–76. Springer Verlag, Heidelberg.

Schwefel, H.-P. (1981). *Numerical Optimization of Computer Models*. Wiley, Chichester.

Uosaki, K. and Hatanaka, T. (2005). Evolution strategies based gaussian sum particle filter for nonlinear state estimation. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 3, pages 2365 – 2371.