Localisation and Tracking of Underwater Acoustic Source Using a Modified Particle Filter

Dragana Carevic
Maritime Operations Division, DSTO
Bldg A-51, HMAS Stirling
Rockingham, WA 6958, Australia
Email: dragana.carevic@dsto.defence.gov.au

***Abstract*—This paper is concerned with the problem of recursively estimating the kinematics of a target moving in underwater environments. The measurements are noise-corrupted time differences of arrival (TDOA) of the signal emitted from the source to pairs of spatially separated sensors. The errors associated with these measurements are very small so standard nonlinear filtering techniques, such as particle filter, can not be directly applied. The reason is that the likelihood function may concentrate in a region of the state space that is too small to contain samples of the particle system. We apply the improved approximation based on progressive correction principle to efficiently handle this difficult problem and describe a new approach to finding an optimal sequence of fictitious matrices to be used in the correction substeps. The performance of the proposed filter is tested using Monte Carlo simulations and the resulting mean-square errors in the positition and velocity are compared to their theoretical Cramer-Rao lower bounds.**

I. Introduction

The problem of recursively estimating the kinematics of a moving target in an underwater environment using passive sensors is of considerable interest in a number of surveillance applications. Sensors can be mounted on board a ship, towed, or deployed in water, and their position is assumed to be known. The measurements are the noise-corrupted differences in the propagation travel time of an emitted signal from the source to each of the sensors; these measurements are commonly referred to as the time differences of arrival (TDOA) or relative time delays.

This problem is highly nonlinear and requires nonlinear filtering techniques for its solution. To solve it in an online application, sequential Monte Carlo methods, or particle filters (PF) could be used [1], [2]. The PF is a nonlinear recursive Bayesiang filtering method based on representing the posterior probability density of the state vector as a set of random samples (or particles) with associated weights. When a measurement arrives, the PF works by propagating the set of samples and then recomputes the weights to incorporate the likelihood of the new measurement.

The PF is easy to implement and has shown to exibit good performance in nonlinear problems. However, the particle set is usually a discrete approximation of the continuous posterior distribution. This introduces the problem of loss of diversity of particles that may result in poor representation of the posterior density. A number of techniques have been proposed that attempt to improve the diversity among particles. One is to perform regularisation of the empirical distribution associated with the particle system using a kernel method, called the regularised particle filter (RPF) [3]. Another approach is to use Monte Carlo Markov chain (MCMC) move step approach [2] based on Metropolis-Hastings algorithm.

The problem of loss of diversity among particles is even more pronounced in the cases where the observation variance is very small. The likelihood function associated with such a measurement may concentrate in a region of the state space that is too small to contain samples of particle system. To efficiently handle this difficult problem Musso *at al.* [3] and Oudjane and Musso [4] propose the improved approximation based on progressive correction (PC) principle. The correction step, performed when a new measurement arrives, is carried out via a number of substeps whereby the initial sub-correction is carried out using a large (fictitious) variance matrix, and the variance matrix is decreased (or deflated) in subsequent steps. A similar algorithm, called the annealed particle filter (APF), is proposed in the computer vision literature. It applies the simulated annealing principle to perform a particle based stochastic search for the global maximum of the weighting function [5].

Because the velocity of sound propagation in water is great, measured TDOAs are usually small and the TDOA measurement errors have extremely small variances. The particle filter based on progressive corrections proposed in [3] may not produce a monotonically decreasing sequence of fictitious matrices resulting in incorrect state estimates. In this paper we describe a modification to this approach that enables efficient filtering under these constraints.

II. Problem Formulation

We consider a scenario where a single source is moving in three-dimensional (3-D) observation region. The source is located at the spatial coordinates $\mathbf{t} = (x,\, y,\, z)^T$ and moves in the horizontal xy plane with a nearly constant velocity $(\dot{x},\, \dot{y}\,)^T$. Let the source state vector be given by

$$\mathbf{x} = [x,\, y,\, z,\, \dot{x},\, \dot{y}\,]^T. \tag{1}$$

Then the discrete-time state equation for this problem, based on a white noise acceleration kinematic model, can be written as [6]

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{v}_k \tag{2}$$

978-1-4244-5222-4/10/$26.00 ©2010 CROWN

where $\mathbf{F}$ is the transition matrix defined by

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & T & 0 \\ 0 & 1 & 0 & 0 & T \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

and $\mathbf{v}_k$ is a 5 x 1 i.i.d. Gaussian distributed zero-mean process noise vector with the covariance matrix $\mathbf{Q}$ defined by

$$\mathbf{Q} = \begin{bmatrix} \frac{q_1 T^3}{3} & 0 & 0 & \frac{q_1 T^2}{2} & 0 \\ 0 & \frac{q_1 T^3}{3} & 0 & 0 & \frac{q_1 T^2}{2} \\ 0 & 0 & q_2 T & 0 & 0 \\ \frac{q_1 T^2}{2} & 0 & 0 & q_1 T & 0 \\ 0 & \frac{q_1 T^2}{2} & 0 & 0 & q_1 T \end{bmatrix}. \tag{4}$$

In (4) q_1 and q_2 denote, respectively, the process noise scaling factor for x and y coordinates (in $\mathrm{m}^2/\mathrm{s}^3$), and for z coordinate (in m^2/s). Also, T in (3) and (4) is the sampling time interval.

It is assumed that the acoustic signal emitted by the source is received, at time step k, by P_k spatially separated sensors. The sensors are either stationary or they can be moving. Their spatial positions are given by $\mathbf{s}_{k,p} = (x^s_{k,p}, y^s_{k,p}, z^s_{k,p})^T$, $p = 1, \ldots, P_k$, and are assumed to be known.

Using these received signals, we obtain L_k TDOAs measured between distinct pairs of sensors. The lth TDOA, $l = 1, \ldots, L_k$, is measured between the pair of sensors $\mathbf{s}_{k,p_l}$ and $\mathbf{s}_{k,q_l}$, where $p_l, q_l \in \{1, \ldots, P_k\}$ and $p_l \neq q_l$. This measurement is given by

$$z_{k,l} = h_l(\mathbf{x}_k) + w_{k,l} \tag{5}$$

where $w_{k,l}$ is a zero mean independent Gaussian noise with variance $\sigma^2_{k,l}$ and

$$h_l(\mathbf{x}_k) = \frac{\|\mathbf{t}_k - \mathbf{s}_{k,p_l}\| - \|\mathbf{t}_k - \mathbf{s}_{k,q_l}\|}{c} \tag{6}$$

is the exact TDOA between the sensors $\mathbf{s}_{k,p_l}$ and $\mathbf{s}_{k,q_l}$. In (6) $\|\cdot\|$ denotes the vector 2-norm and c is the velocity of sound propagation in water. All TDOA measurements $z_{k,l}$ are stacked to form an L_k-dimensional observation vector $\mathbf{z}_k = [z_{k,1}, \ldots, z_{k,L_k}]^T$. Also, the corresponding L_k x 1 vector of the measurement functions in (6) is given by

$$H(\mathbf{x}_k) = [h_1(\mathbf{x}_k), \ldots, h_{L_k}(\mathbf{x}_k)]^T. \tag{7}$$

Let $\mathbf{Z}_{1:k} = \{\mathbf{z}_j, j = 1, \ldots, k\}$ denote the sequence of measurements obtained up to time k. Given $\mathbf{Z}_{1:k}$ and the target motion model described by (2)-(4) our aim is to obtain filtered estimates of the state vector $\mathbf{x}_k$. However, there are two features to note about this problem. First, the problem is nonlinear as the measurements in (6) are nonlinearly related to the state vector. Second, the TDOA measurement errors in (5) are very small due to the fact that the velocity of sound propagation in water has a large value. In this way the techniques available for the estimation of the state vector related to this problem are limited.

III. Particle Filter

The problem of nonlinear filtering is to compute, at each time step k, the posterior density $p(\mathbf{x}_k|\mathbf{Z}_{1:k})$ given a realisation of the observation sequence $\mathbf{Z}_{1:k} = \{\mathbf{z}_j, j = 1, \ldots, k\}$. Suppose that the pdf $p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})$ at time step $k-1$ is available. Then the posterior $p(\mathbf{x}_k|\mathbf{Z}_{1:k})$ may be obtained recursively in two stages, prediction and update (or correction) [2]. The prediction stage involves applying the state transition probability $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ to $p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})$, i.e.,

$$p(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1}) d\mathbf{x}_{k-1} \tag{8}$$

and the correction stage involves updating of the prediction pdf via the Bayes' rule

$$p(\mathbf{x}_k|\mathbf{Z}_{1:k}) = \frac{1}{C} p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{Z}_{1:k-1}). \tag{9}$$

where $p(\mathbf{z}_k|\mathbf{x}_k)$ is the likelihood function (or measurement density) and C is a normalising constant. The recurrence relation of (8) and (9) constitutes the formal solution to the Bayesian recursive estimation problem.

A particle filter is a nonlinear filtering technique that approximates (8) and (9), through Monte Carlo simulation, by a set of particles (representing samples of the state vector) associated with discrete probability masses or importance weights. Starting with a weighted set of particles $\{\mathbf{x}^{(i)}_{k-1}, w^{(i)}_{k-1}\}^N_{i=1}$ that is approximately distributed according to $p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})$, new samples are generated from a suitably designed proposal distribution $q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{Z}_k)$, which is also known in the literature as the importance function. Choosing this distribution to be equal to the transition probability kernel $p(\mathbf{x}_k|\mathbf{x}^{(i)}_{k-1})$, we have

$$\mathbf{x}^{(i)}_k \sim p(\mathbf{x}_k|\mathbf{x}^{(i)}_{k-1}) \tag{10}$$

for $i = 1, \ldots, N$. Next, in the update stage, the importance weights are set to

$$w^{(i)}_k = w^{(i)}_{k-1} p(\mathbf{z}_k|\mathbf{x}_k) \tag{11}$$

and are subsequently normalised to $\sum w^{(i)}_k = 1$, as

$$w^{(i)}_k = \frac{w^{(i)}_k}{\sum^N_{j=1} w^{(j)}_k} \quad \text{for} \quad i = 1, \ldots, N. \tag{12}$$

The resulting particle set $\{\mathbf{x}^{(i)}_k, w^{(i)}_k\}^N_{i=1}$ may be resampled in order to eliminate samples with low importance weights and to multiply samples with high importance weights. Generally, resampling involves a mapping of a set $\{\mathbf{x}^{(i)}_k, w^{(i)}_k\}^N_{i=1}$ to a new discrete representation $\{\mathbf{x}^{(j)}_k, w^{(j)}_k\}^N_{j=1}$ where $\mathbf{x}^{(j)}_k = \mathbf{x}^{(i)}_k$ independently with probability $a^{(i)}$, $i = 1, \ldots, N$, and $w^{(j)}_k \propto w^{(i)}_k / a^{(i)}$ [7]. A pseudocode of this algorithm is given in Table I.

A good choice of the discrete distribution $a^{(i)}$ can help steer the particle set towards the right solution. One option is to set $a^{(i)} = w^{(i)}$, in which case the above algorithm is equivalent to the standard resampling procedure described in [8], [1]. But having the additional flexibility in choosing the sampling

TABLE I
GENERAL RESAMPLING ALGORITHM

$\{\mathbf{x}_k^{(i^*)}, w_k^{(i^*)}\}_{j=1}^N$ = RESAMPLE$[\,\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N, \{a^{(i)}\}_{i=1}^N\}\,]$

- For $i = 1$ to N do {
  - Generate $j \in \{1, \ldots, N\}$ with $P(j = l) = a^{(l)}$
  - Set $\mathbf{x}_k^{(i^*)} = \mathbf{x}_k^{(j)}$
  - Set $w_k^{(i^*)} = w_k^{(j)}/a^{(j)}$

  }
- Normalise $w_k^{(i^*)}$ as given by (12)

TABLE II
THE REGULARISATION STEP OF THE RPF

$[\,\{\mathbf{x}_k^{(i^*)}\}_{i=1}^N\,]$ = REGULARISE $[\,\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N\,]$

- Calculate the empirical covariance matrix S_k of $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$
- Compute D_k such that $D_k D_k^T = S_k$
- For $i = 1$ to N do {
  - Generate $j \in \{1, \ldots, N\}$ with $P(j = l) = w_k^{(l)}$
  - Draw $\epsilon^{(i)} \sim K$ from Epanechnikov/Gaussian kernel
  - Set $\mathbf{x}_k^{(i^*)} = \mathbf{x}_k^{(j)} + h_{opt} D_k \epsilon^{(i)}$

  }

weights can potentially be very useful, and we will consider this in more details later in this paper.

A. Regularisation

The particle filter described by (10) and (11) suffers from the problem of loss of diversity among particles, particularly in the cases where the underlying true posterior distribution is continuous. This may lead to a poor representation of the posterior density and result in "particle collapse". A good measure of the diversity among particles is given by the effective sample size that is estimated as follows:

$$N_{eff} = \frac{1}{\sum_{i=1}^N (w^{(i)})^2}. \tag{13}$$

A small N_{eff} indicates that the particle diversity is small and that there is a severe particle degeneracy, and vice versa.

A number of techniques have been proposed that attempt to improve particle diversity (for some examples see [2]). Widely used is the regularised particle filter (RPF) [3] that performs regularisation of the empirical distribution associated with the particle system using a kernel method. In particular the RPF uses a continuous approximation of the posterior density

$$p\left(\mathbf{x}_k|\mathbf{Z}_{1:k}\right) \approx \sum_{i=1}^N w^{(i)} K_h(\mathbf{x}_k - \mathbf{x}_k^{(i)}) \tag{14}$$

where

$$K_h(\mathbf{x}) = \frac{1}{h^{n_x}} K\left(\frac{\mathbf{x}}{h}\right) \tag{15}$$

is the rescaled kernel density $K(\cdot)$, $h > 0$ is the kernel bandwidth, and n_x is the dimension of the state vector $\mathbf{x}$. The kernel density is a symmetric pdf with finite variance. The kernel and the bandwidth are chosen to minimise the mean integrated square error between the true posterior density and the corresponding empirical regularised representation. In the special case of an equally weighted sample, the optimal choice of the kernel is the Epanechnikov kernel [4] and the optimal choice of bandwidth for this kernel h_{opt} is also given [4]. In order to reduce the computational cost, the samples can be generated from a Gaussian kernel instead of the Epanechnikov kernel, with the optimal bandwidth in this case being defined by [4]

$$h_{opt} = A \cdot K^{-\frac{1}{n_x+4}} \text{ with } A = [4/(n_x+2)]^{\frac{1}{n_x+4}}. \tag{16}$$

Given a weighted particle set $\{w_k^{(i)}, \mathbf{x}_k^{(i)}\}_{i=1}^N$, the regularisation step of the RPF is described by the algorithm given in Table II. It can be seen that this algorithm is a resampling procedure based on a set of weights $\{w_k^{(i)}\}_{i=1}^N$ where the resampling is done by generating samples from a countious rather than from a discrete distribution.

B. Progressive Corrections

The progressive correction particle filter (PCPF), a version of which is described in [4], combines the resampling scheme given in Table I with the regularisation step in Table II so as to obtain a solution to the nonlinear filtering problem that arises in the cases where the measurement error is very small. In particular, at a time step k, the resampling stage is performed iteratively, through a series of M_k substeps each of which is carried out using distinct sampling weights $w_{k,m}^{(i)}$, $m = 1, \ldots, M_k$. The initial distribution $w_{k,1}^{(i)}$ is chosen so as to be very broad, representing the overall trend of the search space, and it becomes more peaked as m increases.

Consider the tracking problem defined in Section II where the measurement error is Gaussian distributed. Given the set of particles $\{\mathbf{x}_k^{(i)}\}_{i=1}^N$ and the parameter $\beta_{k,m} \in [0, 1)$ at the resampling substep m, we compute the fictitious likelihood $p(\mathbf{z}_k|\mathbf{x}_k^{(i)}; \beta_{k,m})$ as a function of $\beta_{k,m}$ as

$$\begin{aligned} &p_m(\mathbf{z}_k|\mathbf{x}_k^{(i)}; \beta_{k,m}) = \\ &= \frac{1}{C}\exp\left[-\frac{\beta_{k,m}}{2}\left(\mathbf{z}_k - H_k(\mathbf{x}_k^{(i)})\right)^T R_k^{-1}\left(\mathbf{z}_k - H_k(\mathbf{x}_k^{(i)})\right)\right] \end{aligned} \tag{17}$$

for $i = 1, \ldots, N$, where $R_k = \text{diag}[\sigma_{k,1}^2, \ldots, \sigma_{k,L_k}^2]$ is the measurement error covariance matrix. The corresponding normalised weights are then obtained as

$$w_k^{(i)}(\beta_{k,m}) = \frac{p_m(\mathbf{z}_k|\mathbf{x}_k^{(i)}; \beta_{k,m})}{\sum_{j=1}^N p(\mathbf{z}_k|\mathbf{x}_k^{(j)}; \beta_{k,m})}. \tag{18}$$

Let $\{\beta_{k,m}\}_{m=1}^{M_k}$ be the parameters used to compute the normalised weights $w_k^{(i)}(\beta_{k,m})$ in (18) in the $M_k > 1$ substeps

of the resamping stage. We require that $\{\beta_{k,m}\}_{m=1}^{M_k}$ satisfy the following conditions:

1) $\sum_{m=1}^{M_k} \beta_{k,m} = 1$
2) $0 < \beta_{k,1} < \beta_{k,2} < \ldots < \beta_{k,M_k-1} < 1$.

The first condition ensures that the output of this resampling stage is an uniformly weighted measure. The second condition makes sure that the distribution $w_k^{(i)}(\beta_{k,m})$ is broader for $m = 1$, and that it becomes more peaked as m increases. In particular, note that the likelihood $p(\mathbf{z}_k|\mathbf{x}_k^{(i)}; \beta_{k,i})$ in (16) is computed using the effective measurement error covariance matrix $R/\beta_{k,i}$. Then, if the condition 2) is valid, the likelihoods $p_i(\mathbf{z}_k|\mathbf{x}_k^{(i)}; \beta_{k,m})$, $m = 1, 2, \ldots$ are computed using a decreasing sequence of the effective error covariance matrices $\big(R/\beta_{k,1}, R/\beta_{k,2}, \ldots\big)$. In this way the corresponding weighting distributions in (18) are more peaked for the increased values of m.

To ensure optimal propagation of particles from one resampling substep to another at each time step k the decomposition parameters $\beta_{k,1}, \ldots, \beta_{k,M_k}$ need to be determined adaptively with respect to the characteristics of the underlying particle set. In [4] the optimal value of $\beta_{k,m}$ is chosen as

$$\beta_{k,m} = \frac{2\log\big(\delta_{max}\big)}{\max\limits_{1\le i\le N}\big(\mathbf{z}_k - H_k(\mathbf{x}_k^{(i)})\big)^T R_k^{-1}\big(\mathbf{z}_k - H_k(\mathbf{x}_k^{(i)})\big)} \tag{19}$$

where δ_{max} is a constant. In some cases this scheme fails to produce an increasing sequence of $\beta_{k,m}$ as required by condition 2), and, as a consequence, the algorithm may become unstable.

In this paper we use another approach to selecting $\beta_{k,m}$. It is based on the notion that the effective sample size $N_{eff}(\beta_{k,m})$ at the resampling substep m, defined by

$$N_{eff}\big(\beta_{k,m}\big) = \frac{1}{\sum_{j=1}^{N}\big(w_k^{(j)}(\beta_{k,m})\big)^2}, \tag{20}$$

is a good measure of the number of particles that are propagated to the next substep [5]. We therefore attempt to find $\beta_{k,m}$ so as to keep the effective sampling size at a required level

$$N_{eff}\big(\beta_{k,m}\big) = \alpha_m N \tag{21}$$

where α_m is a constant, $0 > \alpha_m > 1$. Since $N_{eff}\big(\beta_{k,m}\big)$ is monotonically decreasing in $\beta_{k,m}$ for a given α_m (21) has a unique solution for β [5]. Also, when $\alpha_1 = \ldots = \alpha_{M_k} = \alpha$ the sequence of the estimated parameters $\{\beta_{k,1}, \ldots, \beta_{k,M_k}\}$ is monotonically increasing.

In the practical implementation we compute $\beta_{k,m}$ as the solution to the nonlinear equation $N_{eff}\big(\beta_{k,m}\big) - \alpha_m N = 0$ using *fsolve* function from Matlab Optimization Toolbox [9]. This function applies trust-region dogleg method [10] and requires that the starting value for the search $\beta_{k,m}^0$ is provided. Therefore we set

$$\beta_{k,1}^0 = \frac{2\log\big(\delta'_{max}\big)}{\max\limits_{1\le i\le N}\big(\mathbf{z}_k - H_k(\mathbf{x}_k^{(i)})\big)^T R_k^{-1}\big(\mathbf{z}_k - H_k(\mathbf{x}_k^{(i)})\big)} \tag{22}$$

for $m = 1$ and $\beta_{k,m}^0 = \beta_{k,m-1}$ for $m = 2, \ldots M_k$.

TABLE III
MODIFIED PROGRESSIVE CORRECTION PARTICLE FILTER

$[\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N]$ = MPCPF $[\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N, \mathbf{z}_k, \alpha]$

- For $i = 1$ to N do {
  - Draw $\mathbf{x}_k^{(i)} \sim p\,(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)})$ (see (10))
  }
- Set $m = 1$
- Repeat {
  - Using $\{\mathbf{x}_k^{(i)}\}_{i=1}^N$ find $\beta_{k,m}$ such that $N_{eff}\big(\beta_{k,m}\big) = \alpha N$
  - If $m = 1$ and $\beta_{k,1} >= 1$ do {
    - $\beta_{k,1} = 1$
    }
  - Elseif $\sum_{i=1}^{m} \beta_{k,i} >= 1$ do {
    - $\beta_{k,m} = 1 - \sum_{i=1}^{m-1} \beta_{k,i}$
    }
  - For $i = 1$ to N do {
    - Compute $p(\mathbf{z}_k|\mathbf{x}_k^{(i)}; \beta_{k,m})$ using (17)
    }
  - For $i = 1$ to N do {
    - Compute $w_k^{(i)}(\beta_{k,m})$ using (18)
    }
  - Resample using the regularisation algorithm is Table II
    $[\{\mathbf{x}_k^{(i)}\}_{i=1}^N]$ = REGULARISE $[\{\mathbf{x}_k^{(i)}, w_k^{(i)}(\beta_{k,m})\}_{i=1}^N]$
  - Set m = m+1

  } until $\sum_{i=1}^{m-1} \beta_{k,i} = 1$
- For $i = 1$ to N do {
  - Set the output weight $w_k^{(i)} = N^{-1}$
  }

The pseudocode of a single time step update of the modified progressive correction particle filter (MPCPF) is given in Table III. The algorithm iterates through the regularising step in Table II using, in each iteration m, a set of weights that are computed with the parameter $\beta_{k,m}$ estimated based on (21). The iteration stops when it is detected that $\sum_{j=1}^{m} \beta_{k,j} >= 1$ in which case we set $\beta_{k,m} = 1 - \sum_{j=1}^{m-1} \beta_{k,j}$. In this way the resulting set of parameters $\{\beta_k, j\}_{j=1}^{M_k}$ satisfies the condition $\sum_{j=1}^{M_k} \beta_{k,j} = 1$ and the output of the algorithm is a uniformly weighted measure $\{\mathbf{x}_k^{(i)}, N^{-1}\}_{i=1}^N$. Also note that it is possible that in the first iteration, i.e. for $m = 1$, we have $\beta_{k,1} >= 1$. This indicates that the true effective sample size $N_{eff}(\beta)$ computed using $\beta = 1$ is greater or equal to the required αN. In this case we set $\beta_{k,1} = 1$ and resample the particle set using the resulting weights. This is equivalent to the standard resampling procedure described in [8], [1].

IV. CRAMER-RAO LOWER BOUND

In this section we present the computation of the posterior Cramer-Rao Lower Bound (CRLB) for the nonlinear filtering

problem described in Section II. Let $\hat{\mathbf{x}}_k$ denote an unbiased estimator of the state vector $\mathbf{x}_k$ (1) based on measurement sequence $\mathbf{Z}_{1:k} = \{\mathbf{z}_j, j = 1, \ldots, k\}$ and initial density $p(\mathbf{x}_0)$. The covariance matrix of $\hat{\mathbf{x}}_k$ has a lower bound (also referred to as the CRLB) given by

$$E[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^T] \geq \mathbf{J}_k^{-1} \tag{23}$$

where $\mathbf{J}_k$ is the information matrix defined by

$$\mathbf{J}_k = E[(\nabla_{\mathbf{x}_k} \log p(\mathbf{x}_k, \mathbf{Z}_{1:k}))(\nabla_{\mathbf{x}_k} \log p(\mathbf{x}_k, \mathbf{Z}_{1:k}))^T]. \tag{24}$$

According to [11], the matrix $\mathbf{J}_k$ can be recursively computed as

$$\mathbf{J}_{k+1} = \mathbf{D}_k^{22} - \mathbf{D}_k^{21}(\mathbf{J}_k - \mathbf{D}_k^{11})^{-1}\mathbf{D}_k^{12}. \tag{25}$$

For the special case of linear Gaussian state space model in (2) and nonlinear Gaussian measurement model in (5) matrices $\mathbf{D}_k^{ij}$ in (25) are given by

$$\mathbf{D}_k^{11} = \mathbf{F}_k^T \mathbf{Q}_k^{-1} \mathbf{F}_k \tag{26}$$

$$\mathbf{D}_k^{12} = -\mathbf{F}_k^T \mathbf{Q}_k^{-1} = (\mathbf{D}_k^{12})^T \tag{27}$$

$$\mathbf{D}_k^{22} = \mathbf{Q}_k^{-1} + E\{\tilde{\mathbf{H}}_{k+1}^T \mathbf{R}_{k+1}^{-1} \tilde{\mathbf{H}}_{k+1}\} \tag{28}$$

where $\mathbf{F}_k = \mathbf{F}$ and $\mathbf{Q}_k = \mathbf{Q}$ are the transition matrix and process noise covariance matrix defined by (3) and (4), respectively, and $E\{\cdot\}$ is the expectation with respect to $\mathbf{x}_{k+1}$. In order to define $\tilde{\mathbf{H}}_k$ in (28) we first compute the Jacobians of the measurement functions $h_l(\mathbf{x}_k)$ in (6) defined by

$$\tilde{H}_{k,l} = \left[\frac{\delta h_l(\mathbf{x}_k)}{\delta x} \quad \frac{\delta h_l(\mathbf{x}_k)}{\delta y} \quad \frac{\delta h_l(\mathbf{x}_k)}{\delta z} \quad \frac{\delta h_l(\mathbf{x}_k)}{\delta \dot{x}} \quad \frac{\delta h_l(\mathbf{x}_k)}{\delta \dot{y}} \right] \tag{29}$$

where

$$\frac{\delta h_l(\mathbf{x}_k)}{\delta x} = \frac{1}{c}\left(\frac{x_k - x_{k,p_l}^s}{\|\mathbf{t}_k - \mathbf{s}_{k,p_l}\|} - \frac{x_k - x_{k,q_l}^s}{\|\mathbf{t}_k - \mathbf{s}_{k,q_l}\|} \right) \tag{30}$$

$$\frac{\delta h_l(\mathbf{x}_k)}{\delta y} = \frac{1}{c}\left(\frac{y_k - y_{k,p_l}^s}{\|\mathbf{t}_k - \mathbf{s}_{k,p_l}\|} - \frac{y_k - y_{k,q_l}^s}{\|\mathbf{t}_k - \mathbf{s}_{k,q_l}\|} \right) \tag{31}$$

$$\frac{\delta h_l(\mathbf{x}_k)}{\delta z} = \frac{1}{c}\left(\frac{z_k - z_{k,p_l}^s}{\|\mathbf{t}_k - \mathbf{s}_{k,p_l}\|} - \frac{z_k - z_{k,q_l}^s}{\|\mathbf{t}_k - \mathbf{s}_{k,q_l}\|} \right) \tag{32}$$

$$\frac{\delta h_l(\mathbf{x}_k)}{\delta \dot{x}} = \frac{\delta h_l(\mathbf{x}_k)}{\delta \dot{y}} = 0 \tag{33}$$

for $l = 1, \ldots, L_k$. The partial derivatives in (30)-(32) are evaluated at the true value of $\mathbf{x}_k$. The L_k x 5 matrix $\tilde{\mathbf{H}}_k$ is then computed as:

$$\tilde{\mathbf{H}}_k = \begin{bmatrix} \tilde{H}_{k,1} \\ . \\ . \\ . \\ \tilde{H}_{k,L_k} \end{bmatrix}. \tag{34}$$

For larger values of q_1 and q_2 in (4) the expectation in (28) can be computed using Monte Carlo approximation, whereby one needs to create an ensemble of state vector realisations and the expectation is computed as the average over this ensemble. In the cases where q_1 and q_2 are small the expectation operator in (28) can be ignored since the process noise is small and the variation of the state vector realisation is negligible.

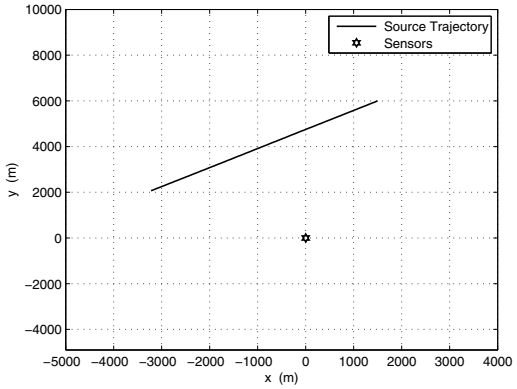

Fig. 1. Projection of the 3D tracking geometry used in the simulations on the xy plane.

The recursion in (25) requires that the initial value of the information matrix is specified. The initial distribution of each component of the state vector (1) is taken to be uniform. Denote by A_j and B_j the limits of the initial uniform density of the jth component of the state vector. Then the initial $\mathbf{J}_0$ is approximated as a diagonal matrix with:

$$\mathbf{J}_0(j,j) \approx \frac{12}{(B_j - A_j)^2}, \quad j = 1, \ldots, 5. \tag{35}$$

V. Simulation Results

We compare the performance of the MPCPF to the CRLB which indicates the best possible performance that one can expect for a given scenario and a set of parameters. In the simulations the source is initially positioned at $\mathbf{t}_0$ =(1500 m, 6000 m, 135 m) and moves with constant velocity v =10.3 m/s and a course of 230°. Note that the coordinate system is oriented such that xy plane overlaps with the water surface and positive z axis corresponds to depth.

An array of five receiving sensors is used in the time delay analysis. The sensors are stationary and are positioned at $\mathbf{s}_1$ = (0 m, 0 m, 100 m)^T, $\mathbf{s}_2$ = (30 m, 0 m, 100 m)^T, $\mathbf{s}_3$ = (-15 m, $15\sqrt{3}$ m, 100 m)^T, $\mathbf{s}_4$ = (-15 m, -$15\sqrt{3}$ m, 100 m)^T and $\mathbf{s}_5$ = (0 m, 0 m, 95 m)^T. The observation time is 600 s and the sampling interval is T = 5 s. The TDOAs are measured with respect to the reference sensor $\mathbf{s}_1$ so that there are $L_k = L = 4$ measurements at each time step k. The measurement accuracy is $\sigma_{k,l} = \sigma_l$ = 15 μs for $l = 1, \ldots, L$ and the velocity of sound propagation in water is $c = 1500$ m/s. The projection of the 3D tracking geometry used in the simulations on the xy plane is shown in Fig. 1.

Initially, the state vector components are set to be independent and uniformly distributed. The upper and lower limits of the initial uniform density for the x and y components are -10 km and 10 km, for the z component 0 m and 500 m, and for the x and y components of the velocity -25 m/s and 25 m/s. The process noise parameters in the MPCPF are set

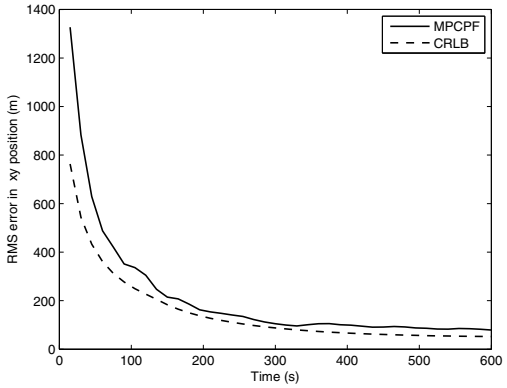

Fig. 2. RMS error in the xy position as a function of time.

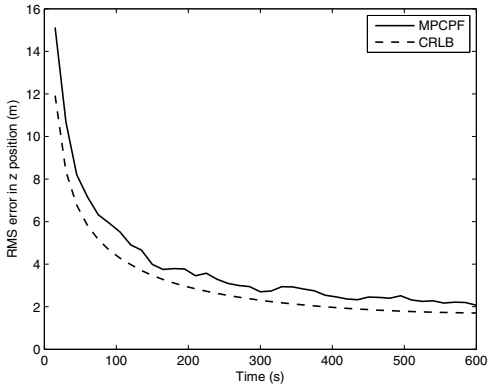

Fig. 3. RMS error in the z position (depth) as a function of time.

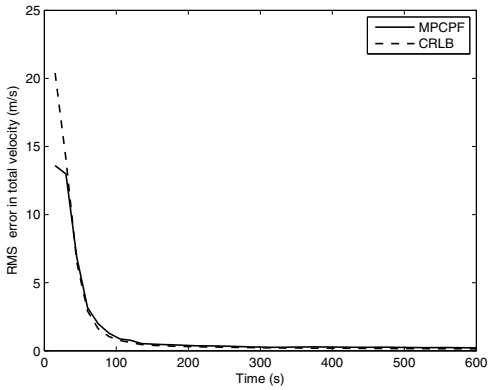

Fig. 4. RMS error in velocity as a function of time.

to $q_1 = 10^{-6}$ m^2/s^3 and $q_2 = 10^{-2}$ m^2/s and the number of particles is N = 1500. We also used $\alpha = 0.2$ to define the effective sample size in each iteration of the resampling stage of the MPCPF.

Monte Carlo simulation are carried out on the above scenario, and the results, based on $M = 200$ runs, are used to compute three MPCPF performance metrics. These metrics are (1) root-mean square (RMS) error in the xy position, (2) RMS error in the z position (depth), and (3) RMS error in total velocity. Let (x_k^i, y_k^i, z_k^i) and $(\hat{x}_k^i, \hat{y}_k^i, \hat{z}_k^i)$ denote true and estimated source position at time step k at the ith Monte Carlo run, and $\dot{x}_k^i$ and $\hat{\dot{x}}_k^i$, and $\dot{y}_k^i$ and $\hat{\dot{y}}_k^i$, be true and estimated source velocities in x and y directions, respectively. The RMS error in the xy position, the RMS error in the z position, and the RMS error in velocity at time k are respectively computed as

$$RMS_{xy}(k) = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left((\hat{x}_k^i - x_k)^2 + (\hat{y}_k^i - y_k)^2\right)} \quad (36)$$

$$RMS_z(k) = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(\hat{z}_k^i - z_k)^2} \quad (37)$$

$$RMS_v(k) = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left((\hat{\dot{x}}_k^i - \dot{x}_k)^2 + (\hat{\dot{y}}_k^i - \dot{y}_k)^2\right)}. \quad (38)$$

We compute the CRLB corresponding to (36)-(38) as follows. Let $\mathbf{J}_k^{-1}[i,j]$ denote the ijth element of the inverse information matrix computed for our problem at time step k. Then the CRLBs for (36)-(38) can be written as

$$CRLB_{xy}(k) = \sqrt{\mathbf{J}_k^{-1}[1,1] + \mathbf{J}_k^{-1}[2,2]} \quad (39)$$

$$CRLB_z(k) = \sqrt{\mathbf{J}_k^{-1}[3,3]} \quad (40)$$

$$CRLB_v(k) = \sqrt{\mathbf{J}_k^{-1}[4,4] + \mathbf{J}_k^{-1}[5,5]}. \quad (41)$$

The results are graphically presented in Figs. 2 - 4. Fig. 2 shows the RMS error in the xy position for the MPCPF and the corresponding CRLB as a function of time; Fig. 3 shows the RMS error in depth for the MPCPF and its CRLB, and Fig. 4 shows the same values computed for the estimated velocity. It can be seen that the estimation errors for the source position and velocity resulting from applying the MPCPF are close to the theoretical lowest limit given by the CRLB. In particular, the error in depth is very small, so depth is estimated with high accuracy. The sensor $\mathbf{s}_5 = (0\,\text{m},\ 0\,\text{m},\ 95\,\text{m})^T$ is elevated 5 meters above the horizontal plane in which the other four sensors are placed; it contributes much to the accurate estimation of the depth of the source.

We note that it is not necessary to search for the optimal value of the parameter β using *fsolve* Matlab function in the MPCPF in the cases where initially, in the first iteration of the algorithm, $\beta_{k,1} \geq 1$ (see the pseudocode in Table III). This condition can be detected by computing the effective sample size as per (20) using $\beta_{k,1} = 1$ and comparing the resulting

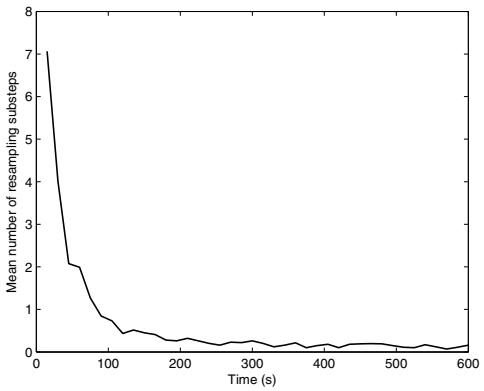

Fig. 5. Mean number of substeps that involve the search for the optimal β over M=200 simulations as a function time.

$N_{eff}(1)$ to the required αN. If $N_{eff}(1) \geq \alpha N$ we simply resample the particle set using the weights computed with $\beta_{k,1} = 1$ and terminate the algorithm. If $N_{eff}(1) < \alpha N$, the search for the optimal value of the parameter β in a number of substeps is initiated as shown in Table III. In this way we can reduce the number of searches using *fsolve* procedure which may be computationally intensive, and thus reduce the computational cost of the proposed algorithm. This improved version of the MPCPF algorithm is used in the Monte Carlo simulations described in this section. Fig. 5 shows the mean number of substeps or iterations performed in the resampling stage of the MPCPF algorithm over M = 200 simulations. It can be seen that the number of substeps is initially relatively great, when the uncertainty is high, but it rapidly declines, and is on average close to zero for the larger time steps k.

VI. Conclusion

This paper was concerned with the problem of recursively estimating the kinematics of a moving target in underwater environments. The measurements were noise-corrupted differences in the propagation travel time of an emitted signal from the source to pairs of spatially separated sensors. The errors associated with these measurements are very small so standard nonlinear filtering techniques, such as particle filter, can not be directly applied. The reason is that the likelihood function may concentrate in a region of the state space that is too small to contain any samples of the particle system. In this paper we applied an improved approximation based on progressive correction principle to efficiently handle this difficult problem and described a new approach to finding an optimal sequence of fictitious matrices to be used in the correction substeps. The performance of the proposed filter is tested using Monte Carlo simulations and the resulting mean-square errors in the positition and velocity are compared to their theoretical Cramer-Rao lower bounds.

References

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 174–188, June 2002.

[2] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond Kalman Filter: Particle filters for tracking applications*. Boston: Artech House, 2004.

[3] C. Musso, N. Oudjane, and F. L. Gland, "Improving regularized particle filters," in *Sequential Monte Carlo methods in practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. Springer-Verlag, New York, 2001, pp. 247–271.

[4] N. Oudjane and C. Musso, "Progressive correction for regularized particle filters," in *Proc. Third Int. Conf. Inform. Fusion*, July 2000, pp. 26–33.

[5] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Proc. Computer Vision and Pattern Recogn. Conf. Volume 2*, July 2000, pp. 126 – 133.

[6] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with application to tracking and navigation*. John Willey and Sons, 2001.

[7] J. S. Liu, R. Chen, and T. Logvinenko, "A theoretical framework for sequential importance sampling with resampling," in *Sequential Monte Carlo methods in practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. Springer-Verlag, New York, 2001, pp. 224–246.

[8] N. Gordon, D. J. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. Radar and Signal Process.*, vol. 140, no. 2, pp. 107 – 113, 1993.

[9] *Matlab Optimization Toolbox*. The MathWorks, Inc., 1984-2008.

[10] J. J. More, B. S. Garbow, and K. E. Hillstrom, "User guide for minpack 1," Argonne National Laboratory, Parkville, Australia, Tech. Rep. Rept. ANL-80-74, 1980.

[11] P. Tichavsky, C. H. Muravchik, and A. Nehorai, "Posterior Cramer-Rao bounds for discrete-time nonlinear filtering," *IEEE Trans. Signal Process.*, vol. 46, no. 5, pp. 1386–1396, 1998.