

Particle Filter Design using Importance Sampling for Acoustic Source Localisation and Tracking in Reverberant Environments

Eric A. Lehmann^{†,*} and Robert C. Williamson[‡]

[†]Western Australian Telecommunications Research Institute
35 Stirling Highway, Perth WA 6009, Australia
E-mail: Eric.Lehmann@watri.org.au

[‡]National ICT Australia *and* Australian National University
Locked Bag 8001, Canberra ACT 2601, Australia
E-mail: Bob.Williamson@nicta.com.au

Abstract—Sequential Monte Carlo methods have been recently proposed to deal with the problem of acoustic source localisation and tracking using an array of microphones. Previous implementations make use of the basic bootstrap particle filter, whereas a more general approach involves the concept of importance sampling. In this paper, we develop a new particle filter for acoustic source localisation using importance sampling, and compare its tracking ability with that of a bootstrap algorithm proposed previously in the literature. Experimental results obtained with simulated reverberant samples and real audio recordings demonstrate that the new algorithm is more suitable for practical applications due to its reinitialisation capabilities, despite showing a slightly lower average tracking accuracy. A real-time implementation of the algorithm also shows that the proposed particle filter can reliably track a person talking in real reverberant rooms.

Index Terms—acoustic source localisation and tracking, microphone array, Bayesian filtering, particle filter, importance sampling.

I. INTRODUCTION

The concept of *acoustic source localisation and tracking* (ASLT) plays an important role in many practical speech acquisition systems. Domains of application include teleconferencing, multimedia information processing, and hands-free telephony, to name but a few. Other applications, such as automatic speech recognition and speaker identification systems, are also very sensitive to the quality of the audio input signals. In most cases, exact knowledge of the speaker position is the key to acquiring clean speech using such tools as beamforming or equalisation principles.

The multipath propagation of acoustic waves in practical environments however constitutes a major challenge to overcome for any tracking algorithm. Recently, methods based on a state-space approach (*Bayesian filtering*) have been developed to deal with this problem [1–3]. Because Bayesian filtering algorithms deliver location estimates based on a series of past

measurements rather than the current observation only, these methods are more efficient at dealing with the spurious effects of acoustic reverberation than traditional ASLT algorithms. Also, a tracker based on state-space filtering involves a model of the specific target dynamics, providing information regarding how the source is more likely to evolve from one time step to the next. This enables the tracker to effectively discriminate between observations originating from the true target and erroneous observations resulting from acoustic disturbances.

Among the different methods based on Bayesian filtering, the concept of *particle filtering* (PF) appears as a promising approach to tackle the ASLT problem [2–4]. As a sequential Monte Carlo method, the PF technique can be used to deal with nonlinear and/or non-Gaussian problems, making it superior to algorithms such as the Kalman filter and its derivatives. This is of particular importance for ASLT, where the observations typically result from a nonlinear process due to the chosen localisation procedure (such as steered beamforming [5], cross-correlation [6] or eigenvalue decomposition [7]). Also, the observation noise in ASLT problems is usually non-Gaussian due to the effects of acoustic reverberation. Particle filtering can then be used to consider several observations per sensor, in order to represent multi-modal density functions reflecting the multiple hypotheses that each of the measurement modalities might originate from the target (see e.g. [2]).

Previous research works on particle filtering applied to ASLT, such as [3, 4, 8], make use of the basic *bootstrap* particle filter, introduced by Gordon, Salmond and Smith in [9]. The conceptual simplicity of this algorithm leads to straightforward practical implementations and moderate computational requirements. The bootstrap PF however suffers from a major drawback: during each iteration, the particles are relocated in the state space without knowledge of the current observations. The PF might hence omit some important regions of the state space when searching for the target, which mainly precludes the PF from reinitialising after a target disappears or becomes occluded for a short period of time. Despite showing promising results, this algorithm con-

*This work was performed while Eric Lehmann was working with National ICT Australia.

sequently still lacks some important characteristics necessary for a smooth operation in practical scenarios, such as the automatic detection of new targets and the ability to recover from track loss.

In this research, we develop a particle filtering method based on the more general concept of *importance sampling* (IS), in which particles are generated during each iteration on the basis of both the particle set at the previous time step and the current measurement. This provides the resulting algorithm with the important property of *reinitialisation*. Importance sampling further allows the combination of different types of observations in a global statistical framework.

The development of a robust acoustic source tracking algorithm for reverberant environments is the main motivation behind the research described in this paper. In the next section, we review the generic approach to the problem of ASLT. The basic concepts of bootstrap filtering and importance sampling are briefly explained in Sections III and IV. We then develop a particle filter for ASLT using the IS approach in Section V, and finally present the results of experimental tests that demonstrate the performance of the newly proposed algorithm in Section VI.

II. SOURCE TRACKING AND BAYESIAN FILTERING

Consider an array of M acoustic sensors distributed at known locations in a reverberant environment with known acoustic wave propagation speed c . Assuming a single sound source, the problem is to estimate the location of this “target” for each time step $k = 1, 2, \dots$, based on the signals $s_m(t)$, $m \in \{1, \dots, M\}$, provided by the array. Let \mathbf{x}_k represent the *state variable* at time k , corresponding to the position and velocity of the target in the state space:¹

$$\mathbf{x}_k = [x_k \ y_k \ \dot{x}_k \ \dot{y}_k]^T.$$

At each time step, each microphone in the array delivers a frame of audio signal which can be processed using some localisation technique such as, for instance, steered beamforming (SBF) or time-delay estimation (TDE). Let \mathbf{y}_k denote the *observation variable* (or measurement) which, in the case of ASLT, typically corresponds to the localisation information resulting from this processing of the audio signals.

Using a Bayesian filtering approach and assuming Markovian dynamics, this system can be globally represented by means of the following two equations:

$$\mathbf{x}_k = g(\mathbf{x}_{k-1}, \mathbf{u}_k), \quad (1a)$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{v}_k), \quad (1b)$$

where $g(\cdot)$ and $h(\cdot)$ are possibly nonlinear functions, and \mathbf{u}_k and \mathbf{v}_k are possibly non-Gaussian noise variables. (1a) is the *transition equation* describing the dynamics of the state variable, and (1b) is the *observation equation* that determines how the measurements are obtained from the unobserved state variable. Ultimately, one would like to compute the so-called *posterior* probability density function (PDF) $p(\mathbf{x}_k | \mathbf{y}_{1:k})$,

¹Note that this research focuses on a two-dimensional problem setting where the height of the source is considered known. The developments can however be easily generalised to handle the third dimension if necessary.

where $\mathbf{y}_{1:k} = \{\mathbf{y}_1, \dots, \mathbf{y}_k\}$ represents the concatenation of all measurements up to time k . The posterior PDF $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ contains all the statistical information available regarding the current condition of the state variable \mathbf{x}_k . An estimate $\hat{\mathbf{x}}_k$ of the state then follows, for instance, as the mean or the mode of this PDF.

The solution to this Bayesian filtering problem consists in the following two steps of *prediction* and *update* [9]. Assuming that the posterior density $p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1})$ is known at time $k-1$, the posterior PDF $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ for the current time step k can be computed using the following equations:

$$p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1}, \quad (2a)$$

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) \propto p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k-1}), \quad (2b)$$

where $p(\mathbf{x}_k | \mathbf{y}_{1:k-1})$ is the *prior* PDF, $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ is the *transition* density, and $p(\mathbf{y}_k | \mathbf{x}_k)$ is the so-called *likelihood* function.

III. BOOTSTRAP PARTICLE FILTER

Particle filtering is an approximation technique that implements the recursion of (2) by representing the posterior density as a set of samples of the state space $\mathbf{x}_k^{(n)}$ (*particles*) with associated likelihood weights $w_k^{(n)}$, $n \in \{1, \dots, N\}$. A basic PF variant is the bootstrap filter [9] which can be described as follows. Assume that the set of particles and weights $\{(\mathbf{x}_{k-1}^{(n)}, w_{k-1}^{(n)})\}_{n=1}^N$ is a discrete representation of the posterior density $p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1})$. The bootstrap PF then implements the following three iteration steps:

1. Resampling: draw N samples $\tilde{\mathbf{x}}_{k-1}^{(n)}$, $n \in \{1, \dots, N\}$, from the existing set of particles $\{\mathbf{x}_{k-1}^{(i)}\}_{i=1}^N$ according to their likelihood weights $w_{k-1}^{(i)}$.
2. Prediction: propagate the particles through the transition equation, $\mathbf{x}_k^{(n)} = g(\tilde{\mathbf{x}}_{k-1}^{(n)}, \mathbf{u}_k)$.
3. Update: each particle is assigned an unnormalised likelihood weight, $\tilde{w}_k^{(n)} = p(\mathbf{y}_k | \mathbf{x}_k^{(n)})$. Then normalise the weights so that they add up to unity:

$$w_k^{(n)} = \frac{\tilde{w}_k^{(n)}}{\sum_{i=1}^N \tilde{w}_k^{(i)}}. \quad (3)$$

As a result, the set of particles and weights $\{(\mathbf{x}_k^{(n)}, w_k^{(n)})\}_{n=1}^N$ is approximately distributed as the current posterior density $p(\mathbf{x}_k | \mathbf{y}_{1:k})$. The sample set approximation of the posterior PDF can then be obtained via:

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) \approx \sum_{n=1}^N w_k^{(n)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(n)}), \quad (4)$$

where $\delta(\cdot)$ is the Dirac delta function, and an estimate $\hat{\mathbf{x}}_k$ of the target state for the current time step k follows as:

$$\begin{aligned} \hat{\mathbf{x}}_k &= \int \mathbf{x}_k \cdot p(\mathbf{x}_k | \mathbf{y}_{1:k}) d\mathbf{x}_k \\ &\approx \sum_{n=1}^N w_k^{(n)} \mathbf{x}_k^{(n)}. \end{aligned} \quad (5)$$

The disadvantage of this algorithm is that during the prediction step, the particles are relocated in the state space without knowledge of the current measurement \mathcal{Y}_k . Some regions of the state space with potentially high posterior likelihood might hence be omitted during the iteration, leading to a decreased tracking performance. This drawback can be addressed using the concept of importance sampling.

IV. IMPORTANCE SAMPLING

Assuming perfect Monte Carlo sampling, let $\{\mathcal{X}_k^{(n)}\}_{n=1}^N$ be a set of N random samples drawn from the density $p(\mathcal{X}_k|\mathcal{Y}_{1:k})$, with uniform weights $w_k^{(n)} = 1/N, n \in \{1, \dots, N\}$. This sample set allows the approximate computation of any statistical quantity of interest based on the PDF $p(\mathcal{X}_k|\mathcal{Y}_{1:k})$ such as its mean or mode, which can be used as an approximation of the current target state. In practice however, the posterior density is not usually available and it is hence impossible to sample directly from it.

An alternative solution is the use of importance sampling (IS), see e.g. [10]. This method consists in choosing a so-called *importance density* $q(\mathcal{X}_k|\mathcal{Y}_{1:k})$ from which particles are easy to sample, $\mathcal{X}_k^{(n)} \sim q(\cdot)$. Then, for the approximation in (4) to remain a truthful representation of the desired posterior density $p(\mathcal{X}_k|\mathcal{Y}_{1:k})$, the computation of the weight must be updated to (see e.g. [11]):

$$\begin{aligned} w_k^{(n)} &\propto \frac{p(\mathcal{X}_k^{(n)}|\mathcal{Y}_{1:k})}{q(\mathcal{X}_k^{(n)}|\mathcal{Y}_{1:k})} \\ &\propto p(\mathcal{Y}_k|\mathcal{X}_k^{(n)}) \cdot \frac{p(\mathcal{X}_k^{(n)}|\mathcal{Y}_{1:k-1})}{q(\mathcal{X}_k^{(n)}|\mathcal{Y}_{1:k})}, \end{aligned} \quad (6)$$

where the second line follows from (2b). The importance weights are hence defined as the product of the likelihood function and a correction term that compensates for a potentially uneven distribution of the particles that might result from the process of sampling the importance function. The generic IS algorithm can be summarised as follows:

1. Sample N particles according to the importance function, $\mathcal{X}_k^{(n)} \sim q(\mathcal{X}_k|\mathcal{Y}_{1:k}), n \in \{1, \dots, N\}$.
2. For each particle, compute the unnormalised importance weight as defined in (6):

$$\tilde{w}_k^{(n)} = p(\mathcal{Y}_k|\mathcal{X}_k^{(n)}) \cdot \frac{p(\mathcal{X}_k^{(n)}|\mathcal{Y}_{1:k-1})}{q(\mathcal{X}_k^{(n)}|\mathcal{Y}_{1:k})}.$$

Then normalise the weights according to (3).

The set of particles and weights $\{(\mathcal{X}_k^{(n)}, w_k^{(n)})\}_{n=1}^N$ is then approximately distributed as the current posterior PDF $p(\mathcal{X}_k|\mathcal{Y}_{1:k})$, and an estimate of the current state can be computed using (5). To emphasize the fact that the particles are here sampled according to a specific PDF (rather than propagated from the previous time step as in the bootstrap implementation), the term *importance particles* will be used from now on to denote the samples $\mathcal{X}_k^{(n)}$ generated by drawing from the importance function $q(\cdot)$.

Note that, although described in this work as a separate algorithm, the bootstrap PF of Section III corresponds to a

special case of the IS algorithm presented here. The bootstrap filter can indeed be derived from the IS procedure with the simplifying assumption $q(\cdot) \triangleq p(\mathcal{X}_k|\mathcal{X}_{k-1})$, emphasising the fact that particles are sampled without taking the current observations into account. Further information on existing PF algorithms and other Monte Carlo methods can be found in [10–12].

The importance sampling principle allows a decreased estimate variance by virtue of an improved sample-based representation. In terms of minimising the variance of the weights, which constitutes the so-called *degeneracy* problem in PF implementations, the optimal importance density $q_{\text{opt}}(\cdot)$ has been shown to be [10]:

$$q_{\text{opt}}(\mathcal{X}_k|\mathcal{Y}_{1:k}) \triangleq p(\mathcal{X}_k|\mathcal{X}_{k-1}, \mathcal{Y}_k). \quad (7)$$

It can be seen that this choice of importance density takes into account both the previous state \mathcal{X}_{k-1} and the current observation \mathcal{Y}_k , making the IS algorithm more robust than the bootstrap method.

In theory however, any density (subject to some weak assumptions) could potentially be chosen as importance function, the main purpose of which is to redirect some of the particles in regions of the state space with potentially high posterior likelihood. In previous literature for instance, the importance function $q(\cdot)$ was implemented to take advantage of measurements from auxiliary sensors (see e.g. [13]), which provides an efficient way of fusing data obtained from different observations. Similarly, the algorithm presented in [14] implements the IS method to draw on information obtained from two different measurement processes derived from the same raw data. Contrary to the method consisting in combining the different observations in the representation of \mathcal{Y}_k , the IS technique hence offers a principled way of including these in a common framework, even when the statistical relationship between the different measurements is not completely known or hard to determine. This specific approach is applied here to the ASLT problem.

V. IMPORTANCE SAMPLING FOR ASLT

A. Algorithm Design

It can be seen that three design choices need to be made for a practical implementation of the IS principle, regarding the definition of the target dynamics, the likelihood function, and the importance function. These issues are discussed in detail below.

1) **Target dynamics:** in order to remain consistent with previous literature [2, 3], a Langevin process is used to model the dynamics equation (1a). This model is typically used to characterise various types of stochastic motion, and it has proved to be a good choice for the current application. The source motion in each of the Cartesian coordinates is assumed to be an independent first-order process, which can

be described by the following equation:

$$\mathbf{x}_k = \underbrace{\begin{bmatrix} 1 & 0 & aT_U & 0 \\ 0 & 1 & 0 & aT_U \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & a \end{bmatrix}}_{\mathbf{G}} \cdot \mathbf{x}_{k-1} + \mathbf{u}_k, \quad (8a)$$

with the noise variable:

$$\mathbf{u}_k \sim \mathcal{N} \left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \underbrace{\begin{bmatrix} b^2 T_U^2 & 0 & 0 & 0 \\ 0 & b^2 T_U^2 & 0 & 0 \\ 0 & 0 & b^2 & 0 \\ 0 & 0 & 0 & b^2 \end{bmatrix}}_{\mathbf{Q}} \right), \quad (8b)$$

where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the density of a multi-dimensional Gaussian random variable with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The parameter T_U corresponds to the time interval separating two consecutive updates of the particle filter. The model parameters in (8) are defined as:

$$\begin{aligned} a &= \exp(-\beta T_U), \\ b &= \bar{v} \sqrt{1 - a^2}, \end{aligned}$$

with \bar{v} the steady-state velocity parameter and β the rate constant. The transition PDF $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ then simply follows from the noise characteristics defined in this model:

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k; \mathbf{G}\mathbf{x}_{k-1}, \mathbf{Q}),$$

with $\mathcal{N}(\boldsymbol{\alpha}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ the density of a Gaussian variable with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ evaluated at $\boldsymbol{\alpha}$.

2) **Likelihood function:** experimental results from previous research carried out on particle filtering for ASLT have shown that steered beamforming (SBF) delivers an improved tracking performance compared to TDE-based methods [3, 15]. The SBF principle is hence used here to implement a *pseudo-likelihood* (PL) function, as introduced in [3].² With $S_m(\omega) = \mathcal{F}\{s_m(t)\}$ the Fourier transform of the m -th signal data, the likelihood function is defined as the output $\mathcal{P}_\Omega(\boldsymbol{\ell})$ of a delay-and-sum beamformer (DSB) steered to the location $\boldsymbol{\ell} = [x \ y]^T$, and computed over the frequency domain Ω :

$$\mathcal{P}_\Omega(\boldsymbol{\ell}) = \int_\Omega \left| \sum_{m=1}^M S_m(\omega) \exp(j\omega \|\boldsymbol{\ell} - \boldsymbol{\ell}_m\| c^{-1}) \right|^2 d\omega, \quad (9)$$

where $\boldsymbol{\ell}_m = [x_m \ y_m]^T$ is the known position of the m -th microphone. In the sequel, the likelihood function is hence computed according to $p(\mathcal{Y}_k | \mathbf{x}_k) \triangleq \mathcal{P}_{\Omega_L}(\boldsymbol{\ell})$, with the location vector $\boldsymbol{\ell}$ reflecting the current state of the variable \mathbf{x}_k and with the integration in (9) carried out over the frequency range $\Omega_L : \omega \in 2\pi \cdot [300\text{Hz}, 3000\text{Hz}]$.

3) **Importance function:** the purpose of $q(\cdot)$ is to relocate some of the particles in the state space taking the current observation into account, and potentially also taking advantage

of a different measurement process. Rather than a fine-scale and accurate representation of the particle sampling areas, the importance function is typically meant to give a coarse indication of where the particles should be sampled in the state space. Based on the signals received at the sensors, several principles could be used to implement this function. The SBF output computed for low frequencies is however known to possess these desired properties. The SBF beampattern at high frequencies generally exhibits a narrow main lobe and suffers from aliasing effects which typically generate spurious peaks in the observations.³ For low frequencies however, the aliasing effects are reduced and the width of the main lobe in the beampattern becomes more important, leading to less accurate but also less ambiguous localisation results. Hence, this approach is of particular interest in the context of importance sampling, and the importance function is defined here as $q(\cdot) \propto \mathcal{P}_{\Omega_s}(\boldsymbol{\ell})$, which is computed according to (9) with the integration carried out over the frequency band $\Omega_s : \omega \in 2\pi \cdot [100\text{Hz}, 400\text{Hz}]$. Note that because the importance function is typically evaluated on a grid defined across the entire state space (see Section VI-A), this function can be easily normalised and it is hence not defined as a pseudo-density.

B. Proposed IS Algorithm for ASLT

The proposed IS algorithm for ASLT, which will be denoted SBF-IS from now on, is given in Algorithm 1. It must be noted that the previously defined importance function is only a coarse approximation of the optimal density $q_{\text{opt}}(\cdot)$ defined in (7), since it only relies on the current SBF measurements. In order to generate some of the state samples on the basis of the previous particle set $\{\mathbf{x}_{k-1}^{(n)}\}_{n=1}^N$, a standard bootstrap option is included in the algorithm (iteration step C). Also, in a manner similar to [14], the reinitialisation step (iteration option A) has been added to allow the PF to deal efficiently with speech pauses or detect a new target entering the scene. This procedure can be seen as a mixed-state bootstrap step, with particles distributed according to a combination of the original bootstrap density and the reinitialisation density. To this purpose, the reinitialisation density has been simply defined to be the same PDF as the importance function, implicitly defining iteration option A of Algorithm 1 as an importance sampling step without compensation of the corresponding importance weights.

The resampling process involved in iteration step C of the IS algorithm can be easily implemented using a scheme based on a cumulative weight function [9]. Alternatively, several other resampling methods are also available from the particle filtering literature, see e.g. [11]. Any of these methods may also be used to efficiently implement the process of sampling particles from the (discrete) importance function $q(\cdot)$, in steps A and B of Algorithm 1.

²The pseudo-likelihood is defined as a pseudo-density, which differs from a true PDF in that it is not necessarily suitably normalised. The reader is referred to [3, 8] for a description of the pseudo-likelihood approach.

³Spatial aliasing is a well-known phenomenon in the microphone array literature [16]. This effect is especially pronounced with widely spaced microphones, which is the type of arrays considered in this work.

Assumption: at time $k - 1$, the set of particles and weights $\{(\mathcal{X}_{k-1}^{(n)}, w_{k-1}^{(n)})\}_{n=1}^N$ is a discrete representation of the posterior distribution $p(\mathcal{X}_{k-1} | \mathcal{Y}_{1:k-1})$.

Iteration: for each particle, i.e. for $n = 1, \dots, N$, choose randomly one of the following sampling methods according to their respective probabilities:

- A. **Reinitialisation** (probability P_R): sample the particle $\mathcal{X}_k^{(n)} \sim q(\mathcal{X}_k | \mathcal{Y}_{1:k})$ and compute the unnormalised importance weight $\tilde{w}_k^{(n)} = p(\mathcal{Y}_k | \mathcal{X}_k^{(n)})$.
- B. **Importance sampling** (probability P_S): sample the particle $\mathcal{X}_k^{(n)} \sim q(\mathcal{X}_k | \mathcal{Y}_{1:k})$, and compute the unnormalised importance weight according to (6):

$$\tilde{w}_k^{(n)} = p(\mathcal{Y}_k | \mathcal{X}_k^{(n)}) \cdot \frac{p(\mathcal{X}_k^{(n)} | \mathcal{Y}_{1:k-1})}{q(\mathcal{X}_k^{(n)} | \mathcal{Y}_{1:k})}.$$

- C. **Bootstrap** (probability $1 - P_R - P_S$): draw a sample $\mathcal{X}_{k-1}^{(i)}$ from the set $\{\mathcal{X}_{k-1}^{(n)}\}_{n=1}^N$ with probability $w_{k-1}^{(i)}$, then propagate it through the transition equation, $\mathcal{X}_k^{(n)} = g(\mathcal{X}_{k-1}^{(i)}, \mathbf{u}_k)$. Compute the unnormalised importance weight $\tilde{w}_k^{(n)} = p(\mathcal{Y}_k | \mathcal{X}_k^{(n)})$.

Finally, normalise the weights according to (3).

Result: the new set $\{(\mathcal{X}_k^{(n)}, w_k^{(n)})\}_{n=1}^N$ of particles and weights is approximately distributed as the posterior density $p(\mathcal{X}_k | \mathcal{Y}_{1:k})$, and the current target state can be estimated according to (5).

Algorithm 1: SBF-IS, importance sampling algorithm for ASLT.

C. Discussion of Practical Implementation Aspects

The respective probabilities of each sampling method are free parameters in the IS algorithm. They can be determined in various ways, including setting them to constant values, as done in [14]. Here, these probabilities are determined at every time step on the basis of whether the current importance function is suitable for sampling or not. Ideally, the importance function is expected to present one peak only, explicitly defining one single region where particles are to be generated. If this function presents several local maxima, it is obviously not appropriate for single-target tracking. Hence, during each PF iteration, the importance function is first computed across the state space, and the number N_p of peaks above a certain threshold (defined here as 90% of the largest measured value) is then determined. The reinitialisation and bootstrap probabilities are then computed as: $P_R = \bar{P}_R / N_p$ and $P_S = \bar{P}_S / N_p$, where \bar{P}_R and \bar{P}_S are the prior probabilities of each method, respectively, and have been optimised on the basis of practical tests as $\bar{P}_R = 0.01$ and $\bar{P}_S = 0.25$.

In practice, the density $p(\mathcal{X}_k^{(n)} | \mathcal{Y}_{1:k-1})$ in the computation of the importance sampling weights (iteration step B) can be

approximated as follows, using (2a) and (4):

$$p(\mathcal{X}_k^{(n)} | \mathcal{Y}_{1:k-1}) \approx \sum_{i=1}^N w_{k-1}^{(i)} p(\mathcal{X}_k^{(n)} | \mathcal{X}_{k-1}^{(i)}). \quad (10)$$

However, because the importance particles are sampled in the state space in a manner that usually violates the propagation model described by (8), the transition PDF $p(\mathcal{X}_k | \mathcal{X}_{k-1})$ in (10) must be updated in order to allow these sampled particles to be given non-zero weights. In the sequel, the following transition PDF will be used in the implementation of (10):

$$p(\mathcal{X}_k | \mathcal{X}_{k-1}) \triangleq (1 - \psi) \cdot \mathcal{N}(\mathcal{X}_k; \mathbf{G}\mathcal{X}_{k-1}, \mathbf{Q}) + \psi \cdot \mathcal{U}(\mathcal{X}_k),$$

where $\mathcal{U}(\cdot)$ denotes the uniform distribution (defined over the considered state space), and the background probability ψ is set to a small constant to account for the fact that importance particles are not governed by the same dynamics model as particles used in a standard bootstrap step. More information about tracking models with switching parameters is provided in [17].

Finally, it can be seen that the importance function $q(\cdot)$ defined in Section V-A only contains *spatial* information about the state vector \mathcal{X}_k . As a result, the velocity component of the importance particles is here set to some random value upon sampling from the importance density:

$$\begin{bmatrix} \dot{x}_k^{(n)} \\ \dot{y}_k^{(n)} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} b^2 & 0 \\ 0 & b^2 \end{bmatrix}\right).$$

VI. PRACTICAL EXPERIMENTS

A. Experimental Setup

The setup defined for the following experiments was based on a medium-sized room measuring roughly $2.9\text{m} \times 3.8\text{m} \times 2.7\text{m}$, and fitted with an array of $M = 8$ omnidirectional microphones positioned at a constant height and organised as one pair on each wall. In each pair, the distance between the sensors was 0.6m .

The microphone signals used in the experiments were samples of audio data sampled at 8kHz , either recorded in a real office room or generated using the image method [18]. For the practical recordings, the sound source was simulated with a loudspeaker moving along a predefined path across the enclosure. The signals were split into frames of 512 samples (processed using a Hamming window), and subsequently used as observation to compute both the importance and likelihood functions. The data processing was carried out using a 50% overlapping factor, yielding the update interval $T_U = 0.032\text{s}$. The numerical values defined for the transition model parameters were set to $\bar{v} = 0.7\text{m/s}$ and $\beta = 10\text{Hz}$.

For the SBF-IS algorithm, the importance function was computed over a horizontal grid of points uniformly distributed across the state space with a spacing of 0.1m .

In the following results, the performance of the IS algorithm is compared to that of the SBF-PL method, a bootstrap-only algorithm described in [3]. For both methods, the number of particles was set to $N = 30$. Other algorithm-specific parameters were optimised empirically to achieve a satisfactory tracking performance, using a reference sample of real audio data recorded in the environment described above.

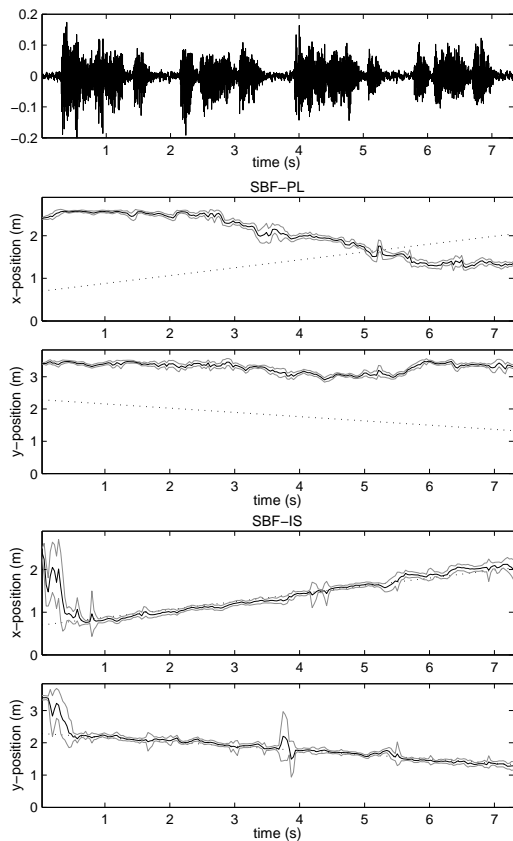


Fig. 1: Tracking results obtained with an IS-based and a non-IS method. *Top plot:* example of signal recorded with one array sensor for this simulation. *Bottom plots:* true source position (dotted line), source location estimate (solid lines) and lines representing \pm one standard deviation of the particle set (grey lines).

B. Tracking Examples

A typical example of the tracking results achieved with the SBF-IS algorithm is depicted in Figure 1. It contains the plots of the estimated source position versus time resulting from the two PF methods. The grey lines above and below the estimated source position represent plus/minus one standard deviation of the particle set for both the x and y -coordinates. The audio data used in this example was recorded in a real office room with reverberation time $T_{60} = 0.39$ s and average SNR 9.4dB. The acoustic source was moving at a constant speed along a straight line over a distance of about 1.6m. The signal recorded with one of the array sensors is given as an example in the top plot of Figure 1. This practical result also demonstrates the reinitialisation capabilities of the IS method, with the set of particles purposely initialised in a random room location at the start of the simulation, about 2m away from the true start position of the target. As soon as the source starts emitting an acoustic signal, the IS method is able to relocate its particles towards the true source position and subsequently tracks the target as it moves across the state space. The non-IS filter is unable to detect the source due to the current measurement data not being taken into account when propagating the particles. The situation described in

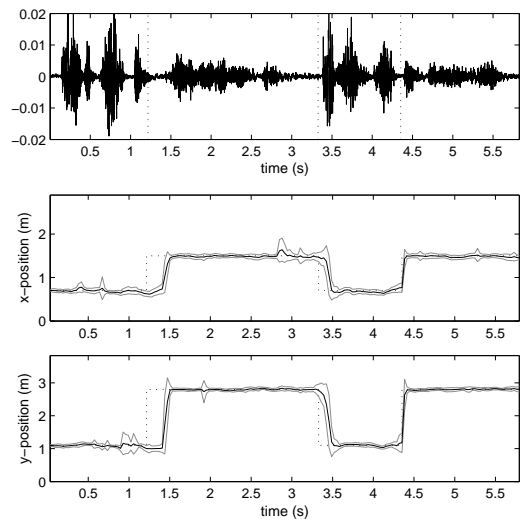


Fig. 2: SBF-IS tracking results with alternating conversation scenario. *Top plot:* example of audio signal generated for one of the array sensors. Vertical dotted lines denote a change of speaker. *Bottom plots:* tracking results in x and y -coordinates. Dotted lines represent the position of the active source.

Figure 1 typically constitutes an example of target detection (track acquisition), for which the IS method clearly shows its superiority over a pure bootstrap implementation. More results on the tracking performance of algorithm SBF-PL can be found in [3].

The results depicted in Figure 2 were obtained with a scenario where two speakers take part in an alternating conversation. The simulation was carried out using the image method to generate signals originating from two different locations in the above mentioned setting, with a reverberation time $T_{60} = 0.35$ s. White noise was added to the microphone signals with an SNR level of about 20dB. The plot at the top of Figure 2 shows an example of signal resulting for one of the sensors. The vertical dotted lines represent time instants at which a speaker change occurs in the original source signal. The plots at the bottom of Figure 2 show the tracking results obtained with the SBF-IS algorithm. This demonstrates once again the efficiency of this method which automatically switches between talkers as soon as a speech signal is detected at a different location in the state space.

C. Image Method Results

Results presented in the previous section specifically demonstrate the performance of the IS algorithm during the phase of target detection, i.e. in *localisation* mode. This section deals with a more specific assessment of the PF operating in *tracking mode* only. To this purpose, the particles were initialised at the true source location at the beginning of each simulation in the following results.

For this experiment, the microphone signals were generated with the image method [18] for varying values of reverberation time T_{60} . White noise was added to the resulting signals with an approximate SNR level of 20dB. A single example

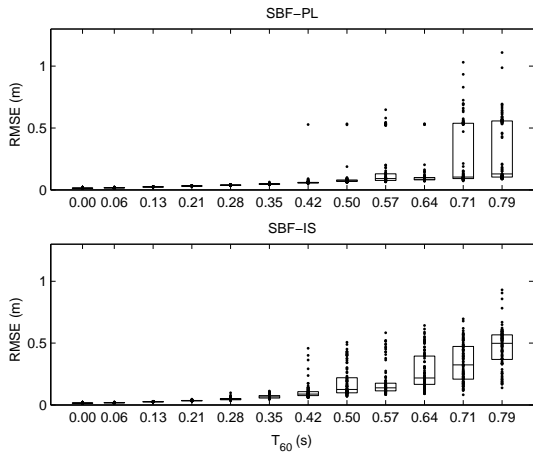


Fig. 3: Statistical tracking performance results obtained with simulated reverberant data (image method) for various levels of reverberation. In each boxplot, the dots represent RMSE data points, the lines at the top and bottom of the box correspond to the 75th and 25th percentile of the data set, respectively, and the horizontal line in the middle of the box is the median of the data set.

of target trajectory and source signal was considered, with a path corresponding to a 1.6m straight line across the room. The source signal was a sentence uttered by a male speaker, defining a 7.3s audio sample.

The results presented in Figure 3 were obtained by simulating each PF algorithm 100 times for the considered audio data. For each run, an estimate of the tracking accuracy was computed as the average deviation (root mean squared error, RMSE) of the PF location estimate from the true source trajectory. The statistical distribution of this assessment parameter (for each value of T_{60}) is plotted in Figure 3 using a boxplot representation, which contains information about interquartile range and median of the RMSE data set.

For low to medium reverberation times, i.e. up to $T_{60} \approx 0.6$ s, these results show that the median tracking accuracy of both IS-based and non-IS methods are similar. Simulation runs for which the PF does not recover after losing track of the target result in the appearance of a second mode in the distribution of the RMSE parameter. This effect can be seen easily in the SBF-PL results for reverberation times greater than about 0.4s, whereas the reinitialisation capabilities of the SBF-IS method allows such cases to be mostly avoided. On the other hand, algorithm SBF-IS exhibits distributions of the RMSE results that are more spread out: the outliers appear here as the tail of the distribution rather than a separate mode. This results from the SBF-IS algorithm occasionally reinitialising off-track (i.e. erroneously) and then recovering, rather than due to a complete and definitive loss of the target as with SBF-PL.

D. Further Discussion

When designing any tracking algorithm, a compromise must be found between its *localisation* ability and its *tracking* accuracy. With the proposed IS algorithm, this can be achieved very efficiently by tuning the prior probabilities of the reinitialisation and importance sampling options, \bar{P}_R and

\bar{P}_S respectively. A bootstrap implementation constitutes an extreme limit in this tradeoff with $\bar{P}_R = \bar{P}_S = 0$.

On the basis of a (non-optimised) Matlab implementation, it can be seen that the SBF-IS algorithm requires roughly twice more computational power than SBF-PL to process the same amount of input data. This is of course due to the additional task of computing the importance function over a fixed grid of points across the state space. However, a real-time implementation of the SBF-IS algorithm, running on a 1.7GHz computer in conjunction with a 16 sensor array, shows that this additional processing power requirement does not represent any difficulties for modern desktop computers. Given this hardware setup, the number of particles in the IS algorithm can be increased up to 120 before reaching the limits of the system resources, which proves to be more than sufficient for the considered application. This practical implementation demonstrates the robustness of the IS algorithm when localising sources and tracking fast target motions in the setting of a $3.5\text{m} \times 4.5\text{m} \times 2.7\text{m}$ office room with a practically measured reverberation time $T_{60} = 0.5$ s. Demonstration movies (originally recorded in real-time) showing some typical examples of the IS algorithm output delivered by this implementation can be found online at <http://www.watri.org.au/~ericl/tracking.html>.

Finally, it must be kept in mind that the tracking performance of the IS method developed in this paper can be potentially largely improved by using some additional information (such as e.g. voice activity detection) to adjust the reinitialisation probability \bar{P}_R . The use of a more elaborate beamforming principle providing improved localisation estimates would also lead to a better tracking performance.

VII. CONCLUSION

Speaker localisation and tracking are complicated array processing applications, made especially challenging by complex reverberation effects and the discontinued nature of speech signals. Adopting a Bayesian filtering approach to this problem leads to superior tracking performance compared to traditional acoustic localisation methods. In this paper, we have developed a particle filtering technique using the principle of importance sampling. The resulting algorithm is able to automatically recover from track loss, detect a new source entering the acoustic scene, and switch between speakers taking turns, thus making it more suitable than bootstrap methods in practice. In a practical tracking system, a bootstrap-only algorithm would typically necessitate additional processing units to deal with such scenarios, whereas the IS method already integrates these functionalities at a low level in the algorithm.

ACKNOWLEDGMENT

National ICT Australia is funded by the Australian Government's Department of Communications, Information Technology and the Arts, the Australian Research Council through Backing Australia's Ability, and the ICT Centre of Excellence programs. We would also like to thank the reviewers for their comments.

REFERENCES

- [1] T. Dvorkind and S. Gannot, "Speaker localization exploiting spatial-temporal information," in *Proc. Int. Workshop on Acoustic Echo and Noise Control*, Kyoto, Japan, September 2003, pp. 295–298.
- [2] J. Vermaak and A. Blake, "Nonlinear filtering for speaker tracking in noisy and reverberant environments," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol. 5, Salt Lake City, USA, May 2001, pp. 3021–3024.
- [3] D. Ward, E. Lehmann, and R. Williamson, "Particle filtering algorithms for tracking an acoustic source in a reverberant environment," *IEEE Trans. Speech and Audio Processing*, vol. 11, no. 6, pp. 826–836, November 2003.
- [4] D. Ward and R. Williamson, "Particle filter beamforming for acoustic source localization in a reverberant environment," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol. 2, Orlando, USA, May 2002, pp. 1777–1780.
- [5] J. H. DiBiase, H. F. Silverman, and M. S. Brandstein, "Robust localization in reverberant rooms," in *Microphone arrays: signal processing techniques and applications*, M. Brandstein and D. Ward, Eds. Berlin: Springer-Verlag, 2001, pp. 157–180.
- [6] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. ASSP-24, no. 4, pp. 320–327, August 1976.
- [7] J. Benesty, "Adaptive eigenvalue decomposition algorithm for passive acoustic source localization," *Journal of the Acoustical Society of America*, vol. 107, no. 1, pp. 384–391, January 2000.
- [8] D. Ward, "Nonlinear filtering of the generalized cross-correlation function for source localization," in *Proc. IEE Workshop on Nonlinear and Non-Gaussian Signal Processing*, Peebles Hydro, UK, July 2002.
- [9] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, April 1993.
- [10] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.
- [11] 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, February 2002.
- [12] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo methods in practice*. New York: Springer-Verlag, 2001.
- [13] J. Vermaak, M. Gangnet, A. Blake, and P. Pérez, "Sequential Monte Carlo fusion of sound and vision for speaker tracking," in *Proc. IEEE Int. Conf. on Computer Vision*, vol. 1, Vancouver, Canada, July 2001, pp. 741–746.
- [14] M. Isard and A. Blake, "ICONDENSATION: unifying low-level and high-level tracking in a stochastic framework," in *Proc. European Conf. on Computer Vision*, vol. 1, Freiburg, Germany, June 1998, pp. 893–908.
- [15] E. Lehmann, D. Ward, and R. Williamson, "Experimental comparison of particle filtering algorithms for acoustic source localization in a reverberant room," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol. 5, Hong Kong, China, April 2003, pp. 177–180.
- [16] M. Brandstein and D. Ward, Eds., *Microphone arrays: signal processing techniques and applications*. Berlin: Springer-Verlag, 2001.
- [17] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: particle filters for tracking applications*. Boston – London: Artech House, 2004.
- [18] J. Allen and D. Berkeley, "Image method for efficiently simulating small-room acoustics," *Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, April 1979.