# Acoustic Source Localisation and Tracking using Track Before Detect

Maurice F Fallon, *Member, IEEE,* and Simon Godsill, *Member, IEEE*

***Abstract*—Particle Filter-based Acoustic Source Localisation algorithms attempt to track the position of a sound source — one or more people speaking in a room — based on the current data from a microphone array as well as all previous data up to that point. This paper first discusses some of the inherent behavioural traits of the Steered Beamformer localisation function. Using conclusions drawn from that study, a multi-target methodology for acoustic source tracking based on the Track Before Detect (TBD) framework is introduced. The algorithm also implicitly evaluates source activity using a variable appended to the state vector. Using the TBD methodology avoids the need to identify a set of source measurements and also allows for a vast increase in the number of particles used for a comparitive computational load which results in increased tracking stability in challenging recording environments. An evaluation of tracking performance is given using a set of real speech recordings with two simultaneously active speech sources.**

***Index Terms*—Tracking Filters, Sequential Estimation, Particle Filtering, Acoustic Source Localisation, Multi-target Tracking.**

## I. Introduction

LOCALISATION and tracking of speech sources — known as Acoustic Source Tracking (AST) or Localisation — has become an increasingly active area of research with applications in the fields of video conferencing and speech. The aim is to use an array of distributed microphones, with no specific arrangement, to track a speaking person as they move around a room based on the path delays between the source and microphones as determined from the sound recordings at the microphones.

Tracking speech sources is, however, complicated by several factors

1) background noise due to the environment
2) other active sound sources
3) reverberation of the source signal itself

which leads to a complex data processing problem. Furthermore speech is, by its nature, highly non-stationary - alternating between periods of high activity during an utterance and silence.

We will be using a particle filtering-based approach to address this problem [1], [2], [3], [4]. This approach has advanced recently from tracking single-source recordings in synthetic environments [5], to tracking in real and challenging environments [6]. An extension has also been proposed which quickly alternates between tracking two speakers taking part in a conversation [7], while another estimates overall speech activity and uses these estimates to more reliably track the speaker during speech silence [8].

While significant progress has been made, some of the properties of the underlying localisation function have not yet been explicitly recognised and accommodated. More specifically, a disproportionate amount of algorithm computation is often devoted to the raw evaluation of the localisation function for particles located very close to one another (within fractions of a centimetre) despite the frequency content of the incoming signals precluding the estimation of the function to such precision. The effect of this is that to maintain real-time operation of the algorithm, either fewer particles or more computational power must be used.

In the following paper we propose a novel algorithm which utilises the Track Before Detect (TBD) methodology to more evenly distribute computation. This algorithm allows us to utilise a much larger particle set, which results in a more stable performance.

The proposed method will *directly* model speaker activity from the localisation function (the Steered Beamformer) without recourse to typical Voice Activity Detection (VAD) algorithms (which are an indirect measure of the activity of the localisation functions). This will address the highly non-stationary nature of speech; facilitating stable and realistic source tracking during speech inactivity.

After a brief overview of the Sequential Monte Carlo framework (commonly referred to as Particle Filtering) and the models we are using in Section II, the results of a series of experiments which study the behaviour of speech and a common localisation function, the Steered Beamformer, are discussed in Section III.

Using these experiments as motivation, a novel likelihood, using the TBD-based methodology, is then presented in Section IV. An extension of this method allows for a straightforward multi-target tracking (Section V).

Finally Sections VI and VII present a series of illustrative and comparative tracking results for both single source and two source tracking.

## II. Proposed Framework

We will define the source state vector at time $k$ to be

$$\alpha_k \triangleq (x_k, \dot{x}_k, y_k, \dot{y}_k, \lambda_k) \tag{1}$$

where $x_k$ and $\dot{x}_k$ are position and velocity, respectively, of the source in the X-direction and similarly for $y_k$ and $\dot{y}_k$ in the Y-

Maurice Fallon is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. Email: mfallon@mit.edu

Simon Godsill is with Signal Processing and Communications Laboratory, Cambridge University Engineering Dept, Trumpington Street, Cambridge, CB2 1PZ, UK. Email: sjg30@cam.ac.uk

Tracking examples can be viewed at http://people.csail.mit.edu/mfallon

This work was funded by Microsoft Research through the European PhD Scholarship Programme.

Manuscript received April 19, 2005; revised January 11, 2007.

direction. The parameter $\lambda_k$, a source activity indicator, will be introduced in Section IV-C. The solution of the tracking problem will require the estimation of the source position portion of this vector, $(x_k, y_k)$, at each time step.

The generic tracking problem requires recursive estimation of the posterior filtering distribution, $p(\alpha_k|\mathbf{Z}_{1:k})$, using Bayes' Theorem as follows

$$\begin{aligned} p(\alpha_k|\mathbf{Z}_{1:k-1}) &= \int p(\alpha_k|\alpha_{k-1})p(\alpha_{k-1}|\mathbf{Z}_{1:k-1})d\alpha_{k-1} \\ p(\alpha_k|\mathbf{Z}_{1:k}) &\propto p(\mathbf{Z}_k|\alpha_k)p(\alpha_k|\mathbf{Z}_{1:k-1}). \end{aligned} \quad (2)$$

This two step process firstly requires a *prediction step* in which the posterior distribution from the previous time step, $p(\alpha_{k-1}|\mathbf{Z}_{1:k-1})$ is propagated using a model for the expected dynamics of a person, $p(\alpha_k|\alpha_{k-1})$, to give us the predictive density $p(\alpha_k|\mathbf{Z}_{1:k-1})$. In the second step - the *update step* the likelihood function (formed from the measurement model) is combined with the predictive density to obtain the posterior distribution at the current time.

### A. Particle Filtering

The above problem may be both non-linear and multi-modal, while the measurement noise may also be non-Gaussian. As such, there exists no closed form solution to the problem. An alternative approach is Sequential Monte Carlo (SMC) which attempts to estimate the distribution by carrying out the above integrations on a large set of weighted discrete samples, also known as particles, which can then be used to form an estimate for the posterior density.

Efficient particle filtering was initially put forward by Gordon et al. [9] as a simple bootstrap filter with weight resampling. It remains an area of active research activity with a large body of published work. A general overview of the principles and background to SMC filtering, or *particle filtering* as it is generally known, can be found in [2], [1]. In the following section the various components that are required to implement such a system will be introduced.

### B. Source Dynamical Model

Source movement in the $\mathcal{X}$ and $\mathcal{Y}$ dimensions[1] will be assumed to be independent and can be decoupled as a result. State dynamics will be modelled by a first-order Langevin Markov process whose specifics were first proposed by Vermaak [5] and retained by Ward et al. [6]. The model will be specified by its initial state and state transition distributions which are of the form $p(\alpha_0)$ and $p(\alpha_k|\alpha_{k-1})$ respectively. The discrete time equations for the $\mathcal{X}$ dimension of the source state will be

$$\begin{aligned} \dot{x}_k &= a_x\dot{x}_{k-1} + b_xF_x \\ x_k &= x_{k-1} + \triangle T\dot{x}_k \\ a_x &= e^{-\beta_x\triangle T} \\ b_x &= v_x\sqrt{1-a_x^2} \end{aligned} \quad (3)$$

[1]Tracking in the vertical dimension is not explored herein as it would require much more extensive microphone coverage than the two dimensional case. It is assumed to be straight forward to extend any successful state vector approach, such as the particle filter, to the third dimension.

where $F_x = \mathcal{N}(0,1)$. A suitable choice of parameter values for $\beta_x$ and $v_x$ will allow us to simulate realistic human motion.

However in subsequent sections the tracking algorithm will be extended to track more than one source and to this end a modification will be introduced to the dynamical model, in Section V-C, which adds a repulsive force to a pair of sources should they drift close to one another.

## III. Characteristics of the SBF for Audio Data

In this section a number of parameters of the Steered beamformer (SBF) localisation function and of speech itself are studied and suggestions of how these observations might best be integrated into the AST framework are discussed.

The SBF function has been chosen as the localisation function for this algorithm for two reasons. Firstly, it provides more accurate tracking performance as demonstrated by Ward et al. [6]. Secondly, as it is an ensemble localisation function, it avoids the complications of speaker crossing and speaker directivity which hamper the Generalised Cross-Correlatins (GCC) [10] in multi-target environments (as further discussed in V-A).

Within the literature, usage of the SBF function is becoming more widespread. Limitations which have previously prohibited its use as a localisation function have lessened due to increasing computational power.

**The Steered Beamformer (SBF) function** is a measure of correlation across a batch of signals for a set of relative delays, and is often seen as an indirect measure of how likely it is that the full batch of audio recordings, from a microphone array, originated at a specific location. The delay-and-sum beamformer[2], expressed by convention as a continuous Fourier Transform (although we will of course implement this using Discrete Fourier Transforms), steered to the physical location $l = [x\ y]$ is given by

$$\mathcal{S}(l) = \int_\Omega \left| \sum_{m=1}^{N_m} S_m(\omega)W_m(\omega)\mathrm{e}^{j\omega T_m(l)/c} \right|^2 d\omega \quad (4)$$

where the measured quantity itself is known as the Steered Response Power (SRP). The Euclidean distance between the steering location and the known position of the $m$th microphone, $l_m$, is $T_m(l) = \|l - l_m\|$. The number of microphones used is denoted $N_m$.

$S_m(\omega)$ is the Fourier transform of the recording made at microphone $m$, while the weighting function, $W_m(\omega)$, is chosen to be the phase transform, $W_m(\omega) = (|S_m(\omega)|)^{-1}$, which is commonly known as the PHAT transform.

The frequency range over which the integration is carried out is denoted $\Omega$. In what follows this range will be chosen to be 200–6000Hz, which corresponds to 371 discrete frequency bins ($N_{\text{freq}}$) when using a discrete Fourier transform with a frame length of 1024 samples and a sampling rate of 16000Hz.

[2]For continuity we will maintain the same notation used by Lehmann and Johnansson [11].

### A. Distribution of Steered Response Power Values

Figure 1 illustrates the distribution of all the steered response power values for a fully evaluated 4x4m SBF function grid for 8 minutes of continuously active speech taken from two different speakers undergoing a number of different paths and trajectories. The distribution illustrated in red corresponds to all grid values **more than 30cm** away from the true source location, i.e. the approximate noise, or 'clutter', distribution. The distribution illustrated in blue corresponds to the SBF function peak values **within 30cm** of the true location i.e. the source distribution. Note that it is common to normalise the steered response range by dividing by $N_m^2 N_{\text{freq}}$; this has not been carried out here.

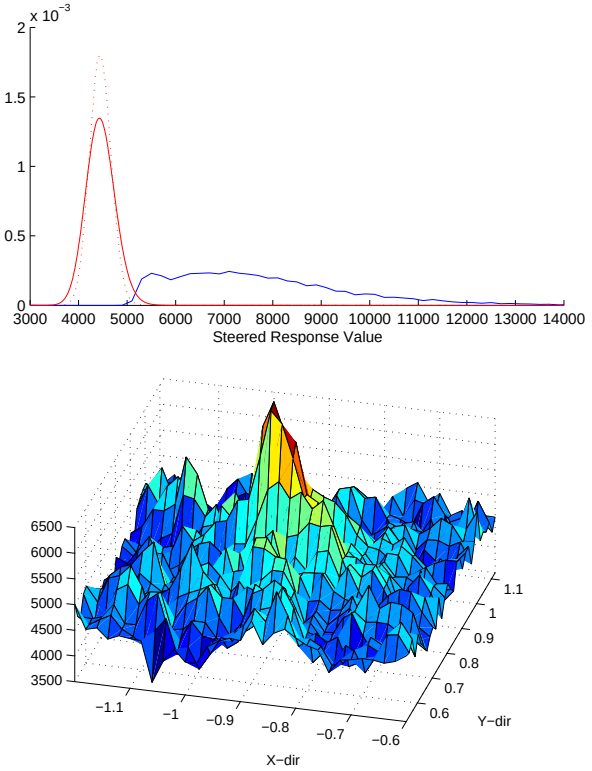


Fig. 1. *Upper: Signal and noise distributions of SBF values, $S(l)$, for 8 minutes of speech. Blue: signal distribution. Red: noise distribution. Dotted red: simulated noise distribution. See Section III-A for more details. $N_{freq} = 371$, $N_m = 12$. Lower: An example of the full 2 dimensional SBF surface evaluated at a 2cm resolution. Each of the points on this surface is equivalant to the pixels mentioned in Section III-A, however this surface is NOT fully evaluated in our algorithm. Note that this surface only corresponds to a small portion of the recording room.*

**Distribution of Clutter Measurements:** Now consider what defines the distribution corresponding to clutter alone. The complex phasor component due to microphone $m$, in Equation 4, is denoted as follows

$$R_m = S_m(\omega) W_m(\omega) e^{j\omega T_m(l)/c}, \tag{5}$$

which is defined to have unit magnitude when using the phase transform. Should each of the $N_m$ signal phasors be entirely uncorrelated with one another, the phases will be uniformly distributed in the range $[0, 2\pi]$, as follows

$$R = e^{j\theta}, \qquad \theta \sim \mathcal{U}(0, 2\pi) \tag{6}$$

In this case, the distribution formed by the summation in Equation 4 (across all of the microphones) is non-standard.

However, once we have summed over all microphones and all frequencies, the central limit theorem applies approximately and we can compute the appropriate mean and variance of the resulting normal distribution as 4430 and 295 units respectively. Clearly the mean and variance of the resulting distribution are directly dependent on the number of frequencies and microphones used in the integration. A simulated noise distribution, drawn according to Equations 4 and 6 with $N_m = 12$ and $N_{\text{freq}} = 371$ and broadly corresponding to the experimental noise distribution (solid red line), is illustrated with a dotted red line in Figure 1 (upper).

**Distribution of Signal Measurements:** The distribution of SBF peak values illustrated in Figure 1 illustrates that when a person is speaking the recordings of the spoken signal at the array of microphones are significantly correlated with one another. The SRP value for which the origin (either clutter or source) is equally likely is approximately 5500 SRP units.

Further experiments presented in [12] have illustrated that this threshold is robust to the (grid) density at which the SBF function is evaluated and also to the level of additive noise present in the recording environment. For these reasons the non-linear CDF-based soft mapping suggested in Section IV-B uses 5500 units as its mean value.

### B. Shape of the Steered Beamformer Function

As suggested by Lehmann [7], the minimum density at which the SBF surface must be implemented to avoid aliasing is defined by the range of integration frequencies used to calculate the surface. However, implementing this surface using a very dense grid of points requires substantial computational power, hence a trade-off will be sought such that only this minimum density is used. See Figure 1 (right) for an example of the full SBF surface at a high density. The figure also illustrates that the density used, $\delta x = \delta y = 0.02$m, is above what would be required to observe the underlying surface.

For example, if a low range of integration (such as 100–200Hz) is chosen, the width of SBF peaks will be much broader because the wavelengths of these frequencies are much longer. Experiments have shown, [12], that a grid with cell density of 0.8m will typically observe these peaks.

For the SBF integrated over the full range, a speech signal, with maximum frequency in the range of 4000–6000Hz, will result in an SBF suface with source peaks with 3dB width of 5–10cm. For this reason, the SBF surface will be discretised to a grid with cell density of 10cm in what follows.

### C. Other Important Issues

**Frequency of Useful Measurements:** As previously observed [13], speech is a highly non-stationary signal whose frequency content and activity varies widely from one frame to the next. Typically the SBF trace for a speech source will consist of a sequence of useful measurements followed by a sequence of silent or corrupted frames containing no useful measurements. This behaviour is difficult to model as each syllable, word or sentence can vary in length from speaker to

speaker and from utterance to utterance. Instead, we will tackle this problem with a data-reactive Markovian activity detector in Section IV-C.

**Interference between sources:** Interference between two sources simultaneously active in the same acoustic field greatly reduces the frequency of useful measurements when compared to the single source scenario. This is due to signal-to-signal interference. We will adjust the tracking parameters of the multi-target tracking extension proposed in Section V to account for this; a more explicit solution would be to perform some form of source separation at the outset to suppress other active sources.

### D. Accommodating Physical Observations

In this section a new likelihood function for the particle filter to best accommodate the underlying characteristics of the SBF function as identified in the previous section is designed.

Firstly classical approaches to tracking typically involve an initial step in which a small number of useful position measurements are extracted from the sensor output (e.g. from raw radar scans) using sensor signal processing. However this step usually requires a thresholding process; which as well as often being subjective, leads to a loss of information and limits the generality of the tracking algorithm.

Secondly for AST with the SBF function, this step would initially require the calculation of the SBF surface in the full region of interest so as to determine possible candidate peaks from the surface. To calculate this function at a sufficient density of points to guarantee the observation of all candidate source peaks (using the full frequency range of interest) is computationally prohibitive [6], [7]. Instead, the authors proposed limiting the frequency range to a small band of low frequencies. This allows the evaluation of the entire surveillance region at a low density (which can then be normalised). This surface is then used to provide particle proposals while later the full frequency SBF is used to evaluate the particle likelihoods.

If particles are closely positioned and the gradient of SBF surface is constrained by the frequency content of speech, it may be considered unwise to persist in calculating likelihoods what can only be very marginally different. We instead propose evaluating the likelihood function on the points of a grid which have a carefully chosen density. As a result the likelihood function of neighbouring particles need only be calculated once only (and shared when required). The overall number of SBF evaluations required falls dramatically. This alternative approach is introduced in the following sections, drawing on the Track Before Detect (TBD) framework [14], [15].

## IV. Track Before Detect

In the field of Electro-Optical sensor-based tracking, it is assumed that at each time step $k$, a pixel grid of $IJ$ resolution cells is read simultaneously and that an individual pixel $(i, j)$ has an intensity of $z_{ij}(k)$. The complete sensor measurement is denoted

$$\mathbf{Z}(k) = \{z_{ij}(k) : i = 1, \ldots, I, j = 1, \ldots, J\}. \quad (7)$$

Furthermore if a target is present it may only influence the pixel measurement[3] in which it is located. As a result the likelihood can be represented as

$$\begin{aligned} p(\mathbf{Z}|\alpha) &= \prod_{i,j} p(z_{ij}|\alpha) \quad (8) \\ &= \prod_{i,j \in C(\alpha)} p_{S+N}(z_{ij}) \prod_{i,j \notin C(\alpha)} p_N(z_{ij}) \end{aligned}$$

where $C(\alpha)$ is the set of subscripts of pixels affected by the target, with state vector $\alpha$:

$$C(\alpha) = \{(i, j);\ |i\Delta - x| < \Delta/2, |j\Delta - y| < \Delta/2\} \quad (9)$$

The likelihood functions for pixels in noise and in a combination of signal and noise are $p_N(\cdot)$ and $p_{S+N}(\cdot)$ respectively. Using the particle filter technique the update stage of the filter is achieved using weighted resampling in proportion to the particle likelihoods. The resampling weights are thus $w(\alpha) \propto p(\mathbf{Z}|\alpha)$. However because this weight need only be evaluated up to a scaling factor the likelihood function can be divided by $\prod_{i,j} p_N(z_{ij}|x, y)$ giving a likelihood ratio

$$q(\mathbf{Z}|\alpha) \propto \prod_{i,j \in C(\alpha)} l(z_{ij}) \quad (10)$$

where

$$l(z_{ij}) = \frac{p_{S+N}(z_{ij})}{p_N(z_{ij})}. \quad (11)$$

This key step means that the likelihood ratio, Equation 11, need only be calculated for the individual pixel in which the particle is located if using a single pixel model (or for the set of pixels located in the immediate vicinity of the particle, if using a more defined pixel model).

Moreover, when tracking accurately the particles will typically form a tight cluster around the true source location. This means that the measurement value for a particular pixel, $z_{ij}$, may be shared by many of the particles corresponding to a particular target. This leads to a dramatic computational reduction as the SBF calculation is the computationally intensive step in this algorithm, since each pixel likelihood now only needs to be evaluated once and then stored. The benefit of this is studied and discussed in Section IX. Finally, resampling of the particles is carried out at the end of each iteration.

### A. Adapting TBD to Acoustic Source Tracking

The TBD framework was then applied to the AST problem. Once again, the idealised assumption that only one single discrete cell of the discretised SBF surface is affected by each active source was made.

As discussed in Section III an SBF grid density of 10cm is sufficient to observe the majority of promising peaks (although increasing this density may lead to even more accurate results). The SBF grid will thus be discretised with this density for the results of this paper.

[3]More accurate sensor models may allow the target to contribute to more than one pixel, however we have observed substantial variabilty in the shape of the peak from frame to frame. Future work could perhaps consider more accurately modelling the peak shape via a point spread function.

In standard TBD, Equation 11 requires that the SBF values be normally distributed with known mean and variance statistics. However, the actual range of the SBF values is not distributed in this manner. As a result a non-linear mapping will be used to adjust the SBF values onto a more balanced range.

### B. Magnitude Mapping

As proposed in the previous section, the particle likelihood function will be based on a measurement function calculated for a set of pixels rather than a continuous function. As such the measurement related to a particular particle, $\alpha_k$ as defined in Equation 1, is that of the point at the centre of the *pixel* in which it lies,

$$z(x_k, y_k) = z_{ij} \qquad \text{for } (i,j) \in C(\alpha) \tag{12}$$

From the study of the behaviour of the SBF function in Section III, it was noted that for a particular recording environment and experimental setup the SBF function results in distributions of signal-and-noise and noise-only measurements with different mean and variance statistics. In an attempt to better understand the measurement function a nonlinear mapping to the SBF values, $\mathcal{S}(x,y)$, is applied as follows:

$$z(x,y) = \Phi(\mathcal{S}(x,y); \bar{\mathcal{S}}, \sigma^2_{\mathcal{S}}) \tag{13}$$

where $\Phi$ is a normal cumulative distribution function with mean $\bar{\mathcal{S}} = 5500$ and variance $\sigma^2_{\mathcal{S}} = 500$. As a result the measurements have been mapped onto the range $z \in [0,1]$, such that the distribution mean lies between the noise measurements (lower end) and active measurements (higher end) without applying a hard threshold. This approximate mapping does not produce normal distributions, as there is substantial variabilty in the distributions over time. However, it has been found to be robust in practical operation. Finally, the parameters can be calibrated in advance or online and the choice of parameter values is determined only by the number of microphones and the frequency integration range.

Following the framework proposed by Salmond and Birch [14] we shall assume that the background noise is modeled as a zero mean Gaussian with variance of $\sigma^2_N$ for all pixels $(i,j)$. As a result the noise and signal distributions will become normal distributions centred on zero and 1, respectively and with limits of $[0,1]$ as follows

$$\begin{aligned} l(z_{ij}) &= \frac{p_{S+N}(z_{ij})}{p_N(z_{ij})} \\ &= \frac{c_{S+N}\mathcal{N}(z_{ij}; 1, \sigma^2_{S+N})}{c_N \mathcal{N}(z_{ij}; 0, \sigma^2_N)} \end{aligned} \tag{14}$$

where $c_N$ and $c_{S+N}$ are the respective normalisation constants as a result of truncation of $z$ to $[0,1]$.

The truncation constant for the normal distribution, $N(z_{ij}, 0, \sigma^2_N)$, used to evaluate the noise likelihood function is

$$c_N = \left[\int_0^1 p_N(z_{ij})dz\right]^{-1} = 2\left(\text{erf}\left[\frac{1}{\sqrt{2}\sigma_N}\right]\right)^{-1} \tag{15}$$

while the truncation constant for the normal distribution, $N(z_{ij}, 1, \sigma^2_{S+N})$, used to evaluate the signal and noise likelihood function is

$$c_{S+N} = \left[\int_0^1 p_{S+N}(z_{ij})dz\right]^{-1} = 2\left(\text{erf}\left[\frac{1}{\sqrt{2}\sigma_{S+N}}\right]\right)^{-1} \tag{16}$$

The variances may be chosen to be non-identical and other forms of the likelihood function might, instead, have been used if a better match to the data can be found.

For *identical variances*, ($\sigma_{S+N} = \sigma_N$), these constants will cancel out. The likelihood ratio for a pixel will simplify to

$$l(z_{i,j}) = \exp\left[\frac{2z_{ij} - 1}{2\sigma^2_N}\right] \tag{17}$$

Example likelihood functions, as well as the steered response power mapping, are illustrated in Figure 2. The standard deviation used in this figure, $\sigma_{S+N} = \sigma_N = 0.5$, were tuned manually and gave a reasonable modeling of the data on average. Experimentation with non-symmetrical likelihood functions could be a source of future work. This values were used in the experiments carried out in Sections VI and VII.

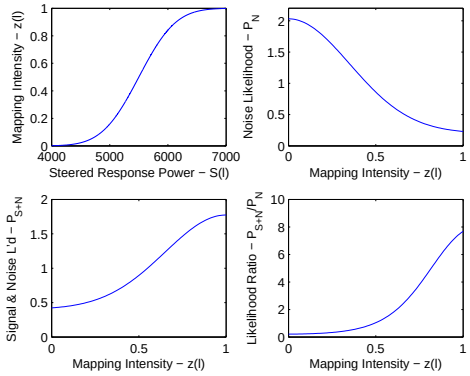


Fig. 2. *Illustration of the functions used in calculating the likelihood ratio. First the raw steered response values are mapped onto the range* $[0-1]$ *using a normal CDF (upper-left). Then the likelihood ratio (lower-right) is evaluated as a ratio of the noise-only likelihood function (upper-right) and the signal-and-noise likelihood function (lower left).*

### C. Activity Indicator Variable

As discussed previously [13], the temporally discontinuous nature of speech must be recognised to allow for a complete AST system. The authors of [13] introduced a model which uses a direct measure of voice activity as a parameter of the tracking system. Generally, when the source is deemed to be inactive, the particles are allowed to drift according to the dynamical model without recourse to the measurement data (the publication goes on to propose a number of different versions which soften the judgement of speaker inactivity). This approach is reasonable and the behaviour presented when the speaker became silent was as one would logically expect - a gradual increase in location estimate variance.

However the Voice Activity Detector proposed therein operated on the actual recorded speech signal rather than on the measurement function itself. This solution is one degree removed from the level of measurement we ideally require: whether the target can be *observed* by the SBF or not. Such

a system will perform poorly should another source be active simultaneously in the room or, for example, if there was a loud noise for a short period elsewhere in the room. To counter these issues we propose to instead detect activity directly from the SBF function, while also integrating the proposed detection mechanism directly within the Bayesian tracking algorithm.

As mentioned above, we have added an activity indicator variable, $\lambda_k \in \{0, 1\}$, to the state vector in a similar way to [14]. This variable will attempt to track the instantaneous activity of the source: if the source is *observable* via the measurement function at the time-frame in question. For single source operation it is anticipated that this will be broadly analogous to syllable-level activity estimation. The parameter is not intended to determine overall longer term speaker activity — a problem which has been examined in [16].

The activity indicator variable will evolve according to a Markovian switching process with pre-determined transition probabilities — where

$$\text{Prob}\{\lambda_{k+1} = a|\lambda_k = b\} = \Pi_{a,b} \tag{18}$$

is the probability of a transition between states $a$ and $b$, where $a, b \in \{0, 1\}$.The optimisation of these parameters is discussed in Section VI-B2b. Following optimisation, we chose the probability of birth, $\Pi_{1,0} = 0.05$, and the probability of death, $\Pi_{0,1} = 0.05$ which were seen to perform well. Particles with an inactive state will drift via the dynamical model with the likelihood ratio set to unity, so that the final likelihood weighting function will become (time index omitted)

$$q(\mathbf{Z}|\alpha) \propto \prod_{\lambda=1,\ (i,j)\in C(\alpha)} \frac{p_{S+N}(z_{ij})}{p_N(z_{ij})} \tag{19}$$

When the target is actually speaking, the likelihood ratio, $l(z_{ij}|\lambda = 1)$, of particles deemed to be active will typically be greater than one. Meanwhile, the ratio of particles deemed to be inactive is defined to be unity. In this way active particles will eventually proliferate upon resampling.

Pseudo-code for the Track Before Detect AST algorithm is given in Algorithm 1.

*a) Overall Source Activity Estimation:* As each particle's activity variable discretely determines the source to be either active or inactive, the overall probability of activity of the source can simply be estimated as the proportion of active particles as

$$p(\lambda_k = 1|\mathbf{Z}_k) \approx \frac{\sum_{p=1}^{N_p}(\lambda_k^{(p)})}{N_p}. \tag{20}$$

where $\lambda_k^{(p)}$ is the $p$th particle of a filter of $N_p$ particles. This will allow us to track source activity directly — as distinct from signal energy activity (the typical VAD output).

Furthermore as this activity variable is dependent only on SBF activity in the region of the particle cluster (which coincides with the estimated source location), it is possible to track the activity of multiple sources simultaneously in different regions of the room — something that would not be possible with a generic voice activity detector.

**Algorithm 1**: *Track Before Detect Acoustic Source Tracking Algorithm*

**for** $p \in \{1 : N_p\}$ **do**
- Predict $\alpha_k^{(p)}$ by drawing from $p(\alpha_k|\alpha_{k-1}^{(p)})$
- Draw a new activity state, $\lambda_k^{(p)}$, using (18)
- Evaluate $q(\mathbf{Z}|\alpha)$ using (19)
- Update weight $w_k^{(p)*} = w_{k-1}^{(p)} q(\mathbf{Z}|\alpha)$

For each $p$ set $w_k^{(p)} = w_k^{(p)*} / \sum_{p=1}^{p=N_p}\{w_k^{(p)*}\}$
Resample if necessary

## V. Multi Target Acoustic Source Tracking

The modification of single target AST algorithms to track more than one simultaneously active target would seem at first glance to be a simple and logical extension. However, the acoustic field within a typical room is affected by each sound source's activity - something that is not the case for a radar scanning system, for example. This means that signal-to-signal correlation — required to produce effective GCC or SBF function — is severely compromised. In turn this reduces the proportion of frames providing useful measurements — this before even considering the problem of source-to-measurement data association.

Multiple target acoustic source tracking using the GCC as a measurement function has been attempted by Ma et al., [17], [18]. The experiments carried out to test this algorithm's performance used signals simulated using the image method and assumed the speakers to be ideal point sources. Both of these simplifications are unrealistic and the resulting method is unlikely to successfully operate when using real recordings.

Furthermore the number of microphone pairs (4) used in the presented simulations is insufficient to provide adequate coverage of a typical room when using the GCC measurement function. As mentioned previously, at least two GCC angle estimates are necessary at all times to provide a 2-D location estimate. Given the effect of speaker orientation, approaching 10 pairs would be necessary to track two real sources using the GCC.

The complexity caused by an unknown number of speakers regularly criss-crossing each other's path in each of the GCC functions, while simultaneously fading in and out of activity during silence, makes for a very difficult data association problem. Figure 3 illustrates this problem and for these reasons the SBF will instead be used as the measurement function.

### A. Multi Target Track Before Detect

Multi-target TBD is a relatively new extension of the TBD methodology, [15]. According to the TBD methodology we have assumed that the source may influence only the pixel value corresponding to the cell in which it is located (or the region surrounding the source location if sensor smearing has occurred). Therefore as Kreucher et al [19] suggest, we will consider the sources to behave independently when widely separated. Tracking in this scenario will be identical to the single source case in Section IV. Alternatively, when sources

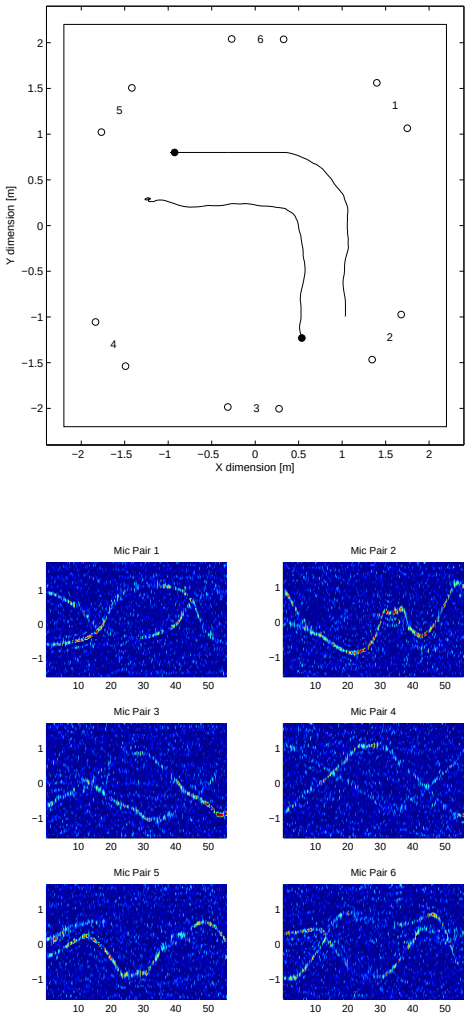


Fig. 3. *Upper: Paths taken by two sources moving in a room. Illustrated also is the room boundary and the location of the microphone pairs. Lower: Illustration of the GCC delay paths for the two moving sources. This figure shows the GCC delays that would be expected for each of the microphone pairs. Each source begins at the circular marker and over 60 seconds moves around the room and back to the markers (following the same path). One complication is illustrated at 15 seconds when the target traces cross in 5 of the 6 microphone pairings - causing considerable target identification complication. A second complication is illustrated in the trace corresponding to microphone pair 2 (for the sample duration and in other microphone traces to a lesser extent). Because source two is facing away from this microphone pair it cannot be observed for the duration.*

are closely spaced a joint likelihood will be considered. The transition between these two states is explained in Section V-D.

Two human speakers moving in a room[4] will generally not separate or coalesce. To preclude this behaviour (within our algorithm) we will introduce a source-to-source repulsive effect for closely spaced targets. A joint particle state will then represent the sources' combined behaviour and track the sources jointly. These two scenarios — joint and disjoint tracking — will be explained in the following sections.

[4]This work will concern itself only with a two source scenario. The possibility of extension to three or more sources is discussed in Section IX but would require a further coding effort without affecting the core algorithm.

### *B. Joint Tracking: Widely spaced sources*

Two widely separated sources will be considered to be independent of one another and will behave as seperate individual targets. This is an approximation which is well-justified in the case of widely spaced sources where there is negligible interaction between their likelihood ratios. A state vector for the source $s$ at time frame $k$ will be

$$\alpha_k^s = (x_k^s, \dot{x}_k^s, y_k^s, \dot{y}_k^s, \lambda_k^s) \tag{21}$$

As in the case of single source tracking, the generic dynamical model (Section II-B) will be used as the transition prior in the prediction step.

Because the sources are widely separated, it will be assumed that only SBF pixels in the vicinity of the true source position will be affected by the source's speech signal. As a result, the likelihood ratio for source $s$ will be identical to the single source case and evaluated in a similar way to Equation 19.

### *C. Tracking more than one closely positioned source*

When considering two sources located close to one another, at time $k$, a joint state vector will be used

$$\begin{aligned} \alpha_k &= (\alpha_k^1, \alpha_k^2) \\ \alpha_k^1 &= (x_1, y_1, \dot{x}_1, \dot{y}_1, \lambda_k^1) \\ \alpha_k^2 &= (x_2, y_2, \dot{x}_2, \dot{y}_2, \lambda_k^2), \end{aligned} \tag{22}$$

with a single associated weighting $w_k$ for the entire particle target cluster. As in the case of joint source tracking, the individual sources will be propagated according to the Langevin model, however, we will modify the dynamical model subtly to discourage the coalescence of two speech sources.

Since the repulsive forces are computable directly from the previous state value at time $k-1$, the causal Markovian structure of the dynamical model is retained. Now, however, sources are explicitly modeled as dependent upon one another. See also Khan, Balcher and Dellaert [20] for an alternative non-causal repulsion mechanism that uses a Markov random field formulation.

*Source Repulsion Mechanism:* The distance between the two target positions can be obtained by

$$d_{12} = \sqrt{(x_1 - x_2)^2 - (y_1 - y_2)^2} \tag{23}$$

with a relative angle between them of

$$\theta_{12} = \angle((x_1, y_1), (x_2, y_2)). \tag{24}$$

We shall propose that beyond a certain particle separation, $d_{12} > d_{\text{rep}}$, the sources are neither attracted to one another nor repelled (modelled as two independent sources using the Langevin motion model of Section II-B). However, when sources become closer than this, $d_{12} \leq d_{\text{rep}}$, a repulsive effect will force them apart. This force is modelled as an accelerating force applied in the opposite relative direction of $\theta_{12}$. A simple squared function has proven to work satisfactorily,

$$F_{\text{rep}}(\alpha_k) = \begin{cases} a_{\text{rep}}(d_{12} - d_{\text{rep}})^2 & \text{if } d_{12} \leq d_{\text{rep}} \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

where $d_{\text{rep}} = 0.3$m is an approximate lower limit of how close two people would typically approach one another while

speaking. $a_{\text{rep}}$ determines the magnitude of the repulsion force and was chosen empirically to give reasonable behaviour. The function is illustrated in Figure 4. This force is then decomposed into its separate $\mathcal{X}$ and $\mathcal{Y}$ components, which for the first source is

$$\begin{aligned} F^1_{\text{rep,x}}(\alpha_k) &= \cos(\theta_{12})F_{\text{rep}}(\alpha_k) \\ F^1_{\text{rep,y}}(\alpha_k) &= \sin(\theta_{12})F_{\text{rep}}(\alpha_k) \end{aligned} \quad (26)$$

while the force applied to the second source is the equal opposite force

$$\begin{aligned} F^2_{\text{rep,x}}(\alpha_k) &= -F^1_{\text{rep,x}}(\alpha_k) \\ F^2_{\text{rep,y}}(\alpha_k) &= -F^1_{\text{rep,y}}(\alpha_k). \end{aligned} \quad (27)$$

See Figure 4 for a graphical illustration of the decomposition. These resultant vectors are added to the original dynamical model (in this case for the $\mathcal{X}$-coordinate of source $s$)

$$\dot{x}^s_k = a_x\dot{x}^s_{k-1} + b_xF_x + F^s_{\text{rep},x}(\alpha_{k-1}) \quad (28)$$

$$x^s_k = x^s_{k-1} + dT\dot{x}^s_k. \quad (29)$$

Finally for each source, $s$, the subset of pixels affected by the source is given by

$$C(\alpha^s) = \{(i,j); |i\Delta - x^s| < \Delta/2, |i\Delta - y^s| < \Delta/2\} \quad (30)$$

and the resultant likelihoods are given by

$$q(\mathbf{Z}|\alpha^s) \propto \prod_{\lambda^s=1,\ (i,j)\in C(\alpha^s)} \frac{p_{S+N}(z_{ij})}{p_N(z_{ij})} \quad (31)$$

Following from Equation 9 and assuming that the sources may not occupy the same pixel cell, $C(\alpha^1) \cap C(\alpha^2) = \emptyset$, the product of the two likelihood ratios is calculated to give an overall likelihood ratio for the joint particle cluster

$$q(\mathbf{Z}|\alpha) = \prod_{s=1}^{N_s} q(Z|\alpha^s). \quad (32)$$

### *D. Transition between tracking mechanisms*

Transitions between the joint particle filter and two individual particle filter systems will be decided based on the MMSE estimate of the source particles and their variances[5]. While this may not be as accurate an estimator as the Kullback-Leibler Divergence, for example, it has proven to be sufficient in practice.

We shall denote $I_c = 1$ as the case of the two targets treated jointly using the closely-spaced algorithm, and $I_c = 0$ as the case of independent filtering of the two targets.

The transition decision is then taken according to

$$I_c = \begin{cases} 1 & \text{if } d_{12,\text{MMSE}} \le d_{\text{thres}} \\ & \text{or } (d_{12,\text{MMSE}} - \sigma_1 - \sigma_2) \le \sigma_{\text{thres}} \\ 0 & \text{otherwise.} \end{cases} \quad (33)$$

where $d_{12,\text{MMSE}}$ is the distance between the MMSE estimates of the source positions, $\sigma_1$ and $\sigma_2$ are the standard deviations of respective source particle clusters and $d_{\text{thres}}$ is an empirically chosen separation threshold set to 0.65m in what follows. This value represents the range beyond which the SBF surface is uneffected by a particular source, hence two sources can be assumed to be independent beyond this seperation.

[5]Note: the transition between the joint and disjoint particle filters uses the distance between the MMSE estimates of *the entire particle set*. The unrelated repulsion mechanism, in Section V-C, uses the distance between *individual* targets within a single (multi-target) particle.

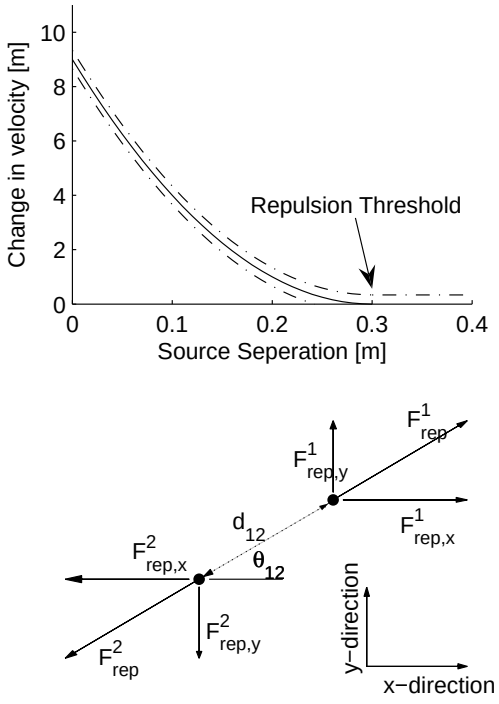


Fig. 4. *Upper: Illustration of the repulsion effect: as source separation, $d_{12}$, falls below the threshold, $d_{rep}$, the force becomes increasingly significant. Lower: The decomposition of the resultant force into dimensional components.*

The integration of this state transition into the algorithm is detailed in Algorithm 2 for *two sources*.

## VI. Experimental Performance Testing

To test the algorithm a set of recordings were made in a office room with twelve microphones spaced around a space roughly 5m x 5m, as illustrated in Figure 5. The recording setup and other details are identical to those used in [21]. Note that the source used was a computer loudspeaker emitting typical conversational speech. The audio sample used was a recording of a BBC radio presenter and has been posted on our webpage. The main sources of background noise were the ventilation system and cooling system of the recording laptop. While the $RT_{60}$ time of the room was not determined, the acoustic behaviour of the room was compariable to a typical office or living room. The tracking algorithm is tested using 250 particles, which allows for realtime operation in MATLAB on a typical PC. A sample rate of 16kHz, frame lengths of 1024 samples and frame overlap of 50% were used in all experiments.

Firstly, we shall determine the stability of the proposed algorithm in increasingly difficult circumstances while comparing it to other particle filter strategies (Section VI-A). We will then go on to demonstrate the optimisation of some of the key parameters of the system (Section VI-B). In Section VI-C, the

**Algorithm 2**: *Switching Acoustic Source MTT Algorithm*

**for** $p \in \{1 : N_p\}$ **do**
- Determine $I_c^{(p)}$ using (33)
- **if** $I_c = 1$ **then**
  - Predict joint $\alpha_k^{(p)}$ using $\alpha_{k-1}^{(p)}$ and repulsion dynamical model
  - Evaluate $q(\mathbf{Z}_k|\alpha^{(p)})$ using (31)
  - Update weight $w_k^{(p)*} = w_{k-1}^{(p)} q(\mathbf{Z}_k|\alpha_k^{(p)})$
  - **if** $D_{MMSE} > D_{thres}$ **then**
    - Resample each target seperately, $w_{s,k}^{(p)} = w_k^{(p)}$
  - Divide State Vector into individual Source State Vectors
  - **else**
    - Resample if necessary
- **else**
  - **for** $s \in \{1 : 2\}$ **do**
    - Predict $\alpha_k^s$ using $\alpha_{k-1}^s$ and dynamical model
    - Weight $w_{s,k}^{(p)*}$ according to (19)
  - **if** $D_{MMSE} > D_{thres}$ **then**
    - **for** $s \in \{1 : 2\}$ **do**
      - Resample if necessary
  - **else**
    - Resample
    - **for** $s \in \{1 : 2\}$ **do**
      - Randomly combine individual state vectors: $\alpha_k^{(p)} \leftarrow [\alpha_{1,k}^{(p)}, \ldots, \alpha_{N_s,k}^{(p)}]$

algorithm will be compared to existing algorithms using some common metrics.

Finally, results illustrating the performance of the Multi-Target Tracker will be presented in Section VII.

### A. Comparison with other algorithms

As mentioned in Section III, a GCC-based measurement function fails to utilise all available signal-to-signal correlation information. This means that particle filter tracking with this measurement function will be unstable for certain source positions, paths and recording scenarios. For example, Figure 5 illustrates a path in which the GCC measurements rely principally on only a single microphone pair (pair number 1) for the first half of the recording and for the second half pair number 4. Because of this the tracking algorithm becomes unstable as the frequency of useful location estimates from secondary pairs (numbered 2,3,5,6) is low due to the directionality of human speech [22].

To simulate increasingly challenging recording conditions, white noise[6] was added to each of the 12 recorded audio samples afterwards. The average signal-to-added noise ratio of the samples were as follows:

1) No noise added
2) 30dB (noise barely noticeable)
3) 20dB (noise becoming noticeable)
4) 10dB (noise level significant)
5) 5db (noise beginning to drown out speaker)
6) 0db (speaker slightly drowned out)
7) -5db (speaker substantiality drowned out)

For each scenario a particle filter was run using each of three measurement functions: the GCC, the SBF using Lehmann and Williamson's *Pseudo-Likelihood* method [23] and the proposed Track-Before-Detect SBF. Each filter utilised identical dynamical model settings, resampling schemes and 250 particles.

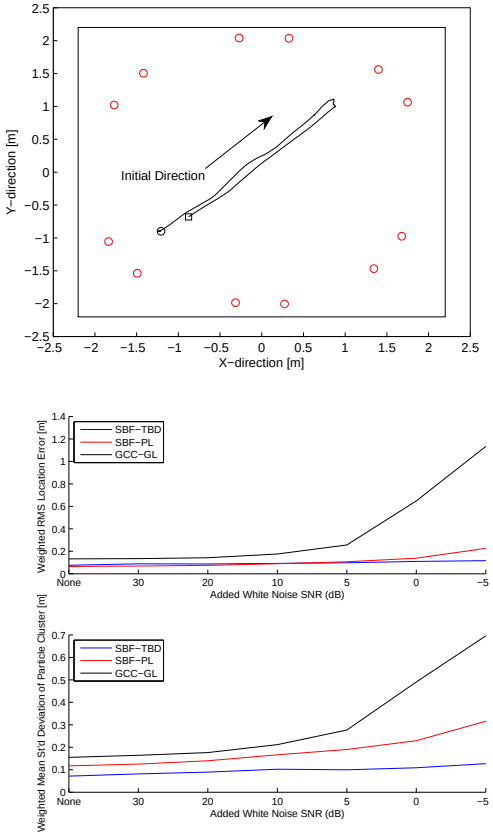


Fig. 5. *Upper: Path taken by a source moving in a room. Note that the speaker begins and ends in the lower left corner and always faces in the direction it is moving. The microphone pairs mentioned in Section VI-A are numbered 1-6 from the upper right corner clockwise. Lower: Tracking performance for three tracking algorithms tracking a speaker in increasing levels of added white noise. Note the collapse of the performance of GCC-based tracker when the SNR falls below 5dB.*

Figure 5 (right) illustrates the performance of the algorithms for each scenario, averaged over 50 Monte Carlo simulations. Although the SBF frameworks consistently out-perform[7] the GCC version, the very poor performance[8] of the GCC framework when the added SNR falls from 5dB to 0dB is of particular note. In these scenarios the secondary microphone pairs fail to provide sufficient bearing estimates to localise in

[6]The implementation of this test with gradually increasing *reverberation* would, of course, have been more insightful but a varechoic chamber was not available.

[7]The superior tracking accuracy afforded by the SBF measurement function has previously been identified by Ward et al. [6].

[8]Note that an average error of one metre corresponds to complete tracking failure, given the dimensions of the experiment

2 dimensions with the only useful measurements being those from pair 1 (and later pair 4) — illustrating the deficiency of the GCC measurement model.

**Comparison with other SBF Algorithms:** While the performance of the TBD algorithm does marginally outperform the pseudo-likelihood algorithm, it should be mentioned that the parameters of neither algorithm were specifically optimised for this particular audio sample. The following highlights algorithmic issues which illustrate the advantages of the proposed TBD algorithm.

Firstly, by its nature, the distribution proposed by Lehmann et al. [7] cannot be properly normalised as recognised by the authors. This means the framework's treatment of particle weightings in successive frames may not be entirely equitable. This issue has been addressed by the TBD method.

Secondly, the correct implementation of the TBD algorithm allows for a large reduction in SBF computation. When multiple particles are located within the same TBD pixel the SBF calculation need only be carried out once and used for all such particles. The computation time required for each of the algorithms (when implemented using MATLAB on a typical desktop PC) was as follows:

- Pseudo-likelihood SBF-based particle filter: 5.94 times real time
- Track-Before Detect SBF-based particle filter: 1.09-1.84 times real time
- GCC-based particle filter: 1.3 times real-time

Note that the computation required for the TBD particle filter is variable as the number of computations will increase when the particle cloud becomes more diffuse, because more SBF likelihood evalutions are required. This occurs during pauses in speech activity and when there is greater speaker location uncertainty as illustrated in Figure 6, again averaging over 50 Monte Carlo runs. In the following section the correct choice of activity variables is shown to remove this instability entirely.

Regardless of these issues, so as to maintain stable tracking during an extended speaker pause a large particle cluster diffuse enough to explore all plausible regions of the state space, so as to ensure the target's eventual re-detection, is required. The Track-Before Detect framework allows us to do this while balancing the computation of the overall algorithm.

### *B. Variable Optimisation*

In the following section the optimisation of two of the more important parameters of the TBD algorithm is illustrated.

*1) Density of the SBF Track Before Detect Grid:* Varying the density of the SBF TBD grid affects the particle filter tracking accuracy as well as the required computation. In this section the optimal grid density is sought. The experiment is related to but distinct from the experiment in Section III. In Section III the minimum grid density required so as to observe the full set of SBF surface peaks was examined with different frequency ranges. No particle filter was implemented therein.

For the experiment presented in this section a particle filter tracked the speaker path illustrated in Figure 9 (upper plot). The results for a set of simulations are presented in Figure 6. Each data point is the average of 50 simulation runs, each tested on the same 30 seconds of recorded data.

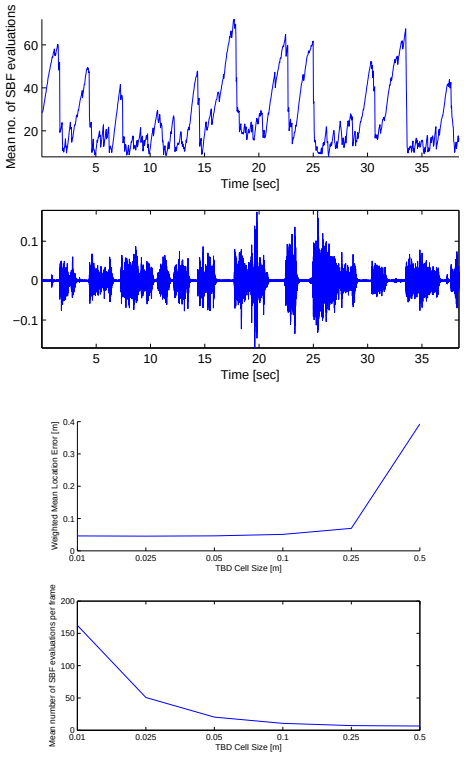


Fig. 6. *Upper: Variation in the number of SBF evaluations (upper plot) carried out using the TBD framework over the course of a speech sample (lower plot). Note how the number of evaluations required grows during silent periods. Lower: The effect of varying the density of the Track-Before Detect grid cells. See Section VI-B for more details.*

The RMS estimated source location error (upper plot) was evaluted while varying the SBF TBD grid density. Meanwhile, the lower plot illustrates the mean number of SBF evaluations required at each grid density per frame.

This means that for a grid density of 0.1m, on average, each steered response value was evaluated just once — yet shared across 23 different particles. When the grid density is set to 0.01m each evaluation was shared across an average of just 2 particles — a huge reduction in efficiency. The cell size of the TBD SBF grid is set to 0.1m for all other simulations in this paper.

The mean location error for the particle filter falls as the cell size is reduced. However it can be seen that there is little improvement gained by reducing the cell size below 10–25cm[9]. Furthermore, the number of SBF evaluations required (and the associated increase in computational expense) quickly increases when the density is reduced below 10–25cm. As such this cell size, 10–25cm, represents a *sweet spot* in which accuracy of the particle filter and the computational demand of the beamformer are balanced. This optimal grid size is determined by the number of microphones used.

*2) Parameters of Activity Indicator and Stability during Speaker Silence:*

[9]The mean location error for the highest grid density - approximately 5cm - gives an indication of the upper bound on the performance of this or any localisation algorithm.

*a) Illustrative Example: Speaker with Silent Pauses:* As mentioned previously the activity indicator allows us to determine directly the activity of a speaker from the particle filter behaviour. Knowledge of the speaker activity is useful both as an algorithm output but also in improving computational stability during speaker silence. In this section the main parameters of this system, the birth and death probabilities, $P_b$ and $P_d$, are experimented with.

Figure 7 illustrates the algorithm results for a speaker moving in the room described above. The speaker is silent on two occasions — between 5–9 seconds and between 18–23 seconds (with its location during silence indicated by dotted lines). The particle filter tracks the speaker location accurately during speaker activity. However, note how the uncertainty of the X and Y position estimates grow, as expected, during sections of speaker inactivity.

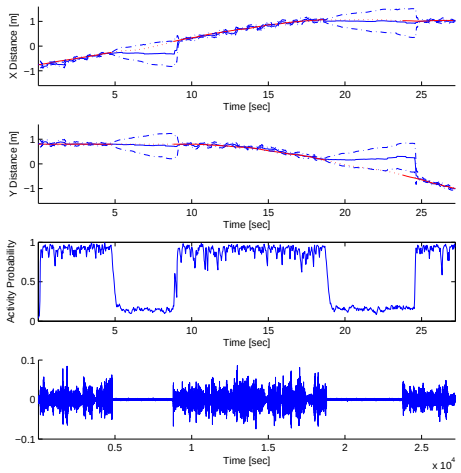


Fig. 7. *Example of single target tracking with speech pauses. Tracking performance in the X and Y directions is shown in upper and centre-top figures respectively. The correct path is shown by a red line (solid when active, dotted when silent), the particle filter mean location estimate with a solid blue line and estimate variance bars are shown either side of the estimate in dashed blue. Note how the variance of the position estimate increases during the two periods of extended silence. The lower centre plot shows the evolution of the overall probability of activity, $p(\lambda_{overall})$, while the bottom plot is of one of the recorded speech signals.*

Previous filtering algorithms, [23], required a sufficiently large particle set to adequately sample the entire surrounding region during this silent pause. Instead, using the TBD approach, if a particle is proposed to be inactive, using the Markovian birth/death process detailed in Section IV-C, then the associated target likelihood ratio is *defined to be unity* (Equation 19). This means that the set of currently inactive particles will have little effect on computational load.

*b) Optimisation of Activity Switching Parameters:* So as to optimise the switching parameters, $P_b$ and $P_d$, the speech source, illustrated in Figure 7, was tracked while varying the probabilities of birth and death. Figure 8 represents the results of 50 runs of the algorithm, each with 250 particles.

The time taken for the filter to recognise that the speaker has resumed speaking (after a pause) and to resume accurate tracking is illustrated in the upper plot (for the two different speech pauses illustrated in Figure 7). For $P_b = P_d = 0.025$, this shows that after approximately 0.6 seconds the filter has resumed typically accurate source tracking (defined here as a mean error of less than 0.15m). Setting the parameter to this or larger values achieved adequately responsive performance.

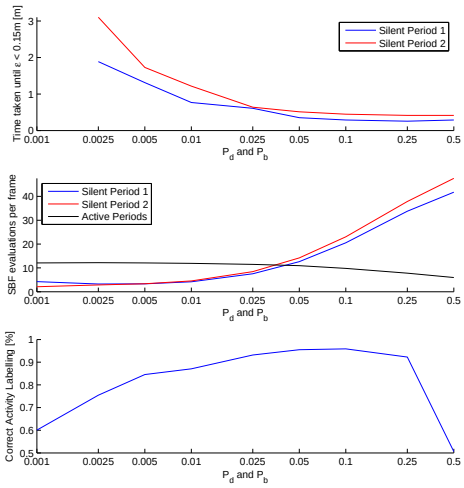


Fig. 8. *The effect that varying $P_b$ and $P_d$ has on the ability for the TBD filter to correct itself after a period of speaker silence (middle) and the amount of computation required during such a silence when compared to what is typically required during typical active tracking (upper). Also shown is the percentage of correct activity labellings over the entire tracking segment. The x-axis of each plot has a logarithmic scale.*

However, the responsiveness of the system for large values of $P_b$ and $P_d$ must be traded against increased SBF computation - due to a more diffuse particle cluster. The centre plot illustrates that, when the activity parameter is set to a larger value ($(P_b; P_d) > 0.05$), greater SBF computation is required when the speaker is silent (red and blue) than when it is active (black) as the particle cluster is more defuse. For $P_b$ and $P_d$ in the region of 0.025–0.05 it can be seen that the filter is computationally stable regardless of speaker activity.

The lower plot illustrates, over the entire speech sample the percentage of iterations in which the speaker activity was correctly labelled (including both active and inactive periods). Again, the best performance is seen in the mid-range — with correct labelling of more than 95% of frames.

For the remainder of these experiments the switching parameters were set to be $P_b = P_d = 0.05$. Note that in principle one could estimate the parameters $P_b$ and $P_d$ by ML or Bayesian methods; this has not been done here. Finally, figure 7 presents an example of accurate activity estimation.

## C. Monte Carlo Simulation Results

A final set of tests of this single source TBD algorithm was carried out against some common AST metrics. The results presented in Table I provide a comparison between the performance of the proposed SBF Track Before Detect algorithm and the GCC-based particle filter, [5], as well the Pseudo-Likelihood and Gaussian Likelihood SBF-based particle filters, [6]. The paths of the sources are illustrated in Figure 9. Performance is measured in terms of the mean squared error ($\bar{\epsilon}$), the mean standard deviation of the particle cluster (MSTD) as well as a measure of the percentage of

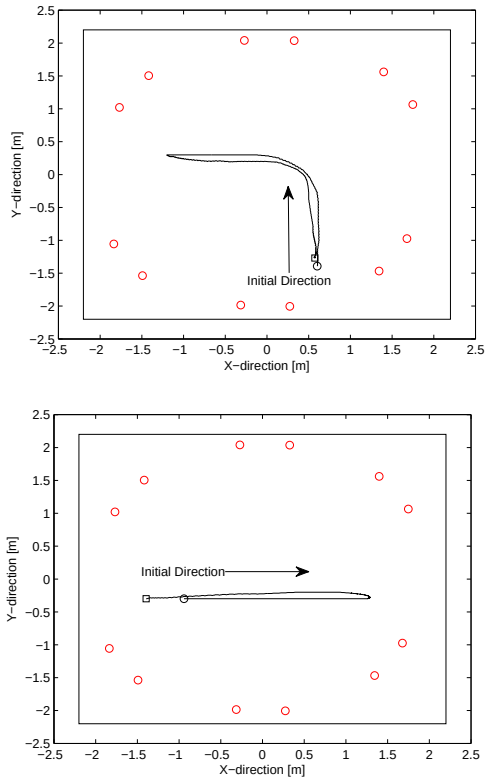


Fig. 9. *Paths taken by the sources tested in Section VI-C. Example 1 is to the upper figure, while example 2 is to the lower figure. Note that in each case the targets double back upon the original path an return to the original location. Performance is not shown.*

tracks which fail completely (Track Loss Percentage, TLP) which was introduced in [11].

Parameters of the dynamical model and other common system settings were set equal in each algorithm, while parameters unique to a particular algorithm follow those quoted in their respective papers. The particle numbers used for each of the algorithms vary in this experiment, but an effort has been made to equalise the algorithm runtime instead. For this reason, 1000 particles have been used for the TBD algorithm but only 100 for the SBF-PL algorithm .

The average tracking error of the proposed algorithm is shown to be similar to that of each of the other algorithms. However, the purpose of illustrating this experimental result is not to identify the superior performance of the TBD filter but rather to illustrate that it gives similar tracking accuracy to previous methods — while reducing the uncertainty of the position estimate. The MSTD for the TBD filter, indicative of uncertainty of the the position estimate and the stability of the tracking algorithm, is substantially lower than for the other algorithms — which was achieved without increasing the overall computation time.

The computation time of the TBD algorithm remains reasonable because of the vast reduction in the proportion of likelihoods that need be calculated using the SBF-TBD. It is anticipated that the SBF-TBD with thousands of particles will comfortably run in real-time on a typical modern computer.

| Method | $\bar{\epsilon}$ (m) | MSTD (m) | TLP | $N_p$ | Time (sec) |
|---|---|---|---|---|---|
| | Example 1 — 60.8sec | | | | |
| GCC-GL | 0.076 | 0.109 | 0 | 100 | 74.73 |
| SBF-PL | 0.073 | 0.310 | 0 | 100 | 126.64 |
| SBF-GL | 0.105 | 0.359 | 0 | 100 | 560.67 |
| SBF-TBD | 0.083 | 0.061 | 0 | 1000 | 73.09 |
| | Example 2 — 33.6sec | | | | |
| GCC-GL | 0.088 | 0.128 | 0 | 100 | 35.27 |
| SBF-PL | 0.108 | 0.330 | 18 | 100 | 71.74 |
| SBF-GL | 0.125 | 0.353 | 22 | 100 | 311.14 |
| SBF-TBD | 0.108 | 0.053 | 0 | 1000 | 48.65 |

TABLE I
*Comparative Results for GCC based bootstrap, SBF based bootstrap and SBF TBD particle filters tracking a single source. Each figure has been averaged for 50 algorithm runs*

## VII. Multiple Target Tracking Results

An evaluation of the tracking performance of the multi-target tracking algorithm is presented in this section. The test recordings were carried out using the same system described in Section V-A. Each source was recorded using the system described in VI and were then linearly mixed before the MTT algorithm was run.

### A. Illustrative Results

Figure 10 shows an illustration of the tracking performance for two different examples of two source tracking. The duration of the two samples, 32 seconds and 54 seconds respectively, is long compared to what has been tested previously in the literature. Source 1 in each case is a female speaker and Source 2 is a male speaker.

The particle filter is seen to track the two targets successfully. Note how the variance of the location estimate varies — particularly for Source 2 in Example 1 (upper plot). This coincides with a portion of audio in which Source 1 dominates the second source. Because Source 2 is unobservable the size of particle cluster (as represented by the uncertainty ellipses) will expand to represent this uncertainty. This is similar to the algorithmic behaviour observed during a silent gap of a single source sample. When the target is observable once more, the particle filter returns to tracking accurately.

### B. Performance Evaluation

Comparative results for the proposed tracking algorithm are presented in Table II. The results show that accurate tracking of two sources speaking simultaneously is successful and only a very slight degradation in performance is displayed relative to the single source case (Table I), despite the fact that dual source recordings will have a much lower proportion of useful peak measurements due to cross-signal interference. Additionally, the computation time is increased only by about a factor of two, which we regard as satisfactory. Using the TBD framework has allowed us to avoid the data association problem which is often computationally intensive in multiple target tracking algorithms.

## VIII. Further Work

A number of issues are yet to be addressed by this algorithm. Firstly, as mentioned above, there does exist the

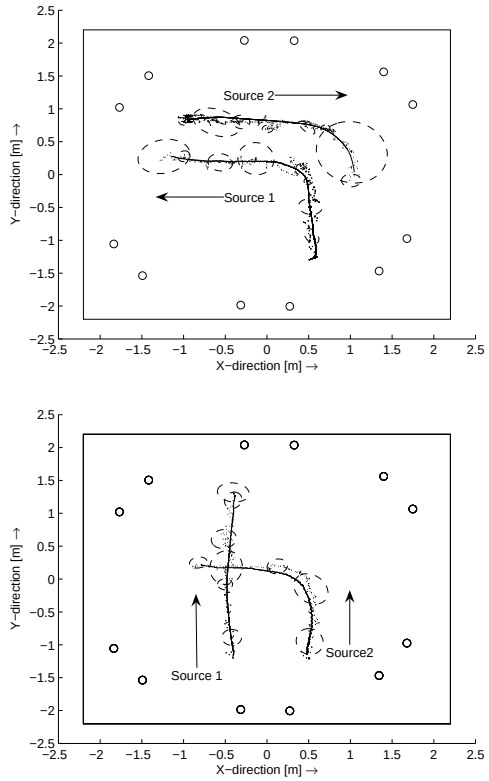


Fig. 10. *Two sample recordings of two sources moving in a room, which were used to test the performance of the algorithm. The ground truth measurements are in black. An example of the tracking performance is overlayed on each plot (dotted black lines). Example 1 is the top plot. The microphone positions are indicated with circles. Uncertainty ellipses are shown every 100 frames.*

| Source | $\bar{\epsilon}$ (m) | MSTD (m) | TLP (%) | $N_p$ | Time (sec) |
|---|---|---|---|---|---|
| | Example 1 — 32sec | | | | |
| 1 | 0.110 | 0.076 | 2 | 1000 | 92.14 |
| 2 | 0.114 | 0.118 | 10 | | |
| | Example 2 — 54.4sec | | | | |
| 1 | 0.11937 | 0.11275 | 2 | 1000 | 184.66 |
| 2 | 0.108 | 0.330 | 4 | | |

TABLE II
*Illustrative Results for the SBF TBD particle filter tracking two sources for the examples in Section 10. Each figure is the average for 50 algorithm runs*

possibility of instability during extended silence. A high level algorithm component, which halts tracking after extended silence, would be required in such a scenario. More generally, the assumption of continuous source activity throughout the algorithm run needs to be relaxed. In current work we have begun to address this, see [12], [16], and this will be reported in a future publication.

In future work, an algorithm to handle initiation and removal of source tracks will be considered.

Secondly, as discussed by Salmond and Birch [14] the performance of the TBD algorithm can be improved if the resolution of the measurement grid is improved so that the source may illuminate more than a single grid point (which would then be modelled as a scattered measurement). It is possible this could increase tracking accuracy further.

## IX. Conclusions

We have proposed a multi-target Track Before Detect algorithm which can track multiple simultaneously active speech sources.

Current Steered Beamformer methodologies are limited by the computational inefficiency of tracking targets using a dense cloud of individual particles — each evaluating the measurement function at minutely different physical locations.

This paper proposes an algorithm using the pixel-based TBD framework. The algorithm reduces the proportion of likelihoods which are typically calculated per particle which allows for a vast increase in the number of particles to be used for a simular computational effort. While the tracking accuracy of the algorithm was shown to be slightly better than other single source AST algorithms, the much larger particle cluster yielded greater stability. This illustrates the proposed algorithm's utility lies in the most challenging conditions.

An extension of the algorithm to track two sources was also detailed. Performance for two source tracking examples was seen to be comparable to the single source scenario and with an increase in computation of only a factor of two. Tracking stability for closely spaced targets was maintained using a novel repulsion mechanism.

Future work should consider extending the approach to the full 3 dimensional case and investigation of a more accurate source point spread model. Furthermore, the multi-track tracking framework should be relaxed to consider the recognition of and response to extended speech silences explicitly within the filter.

## Acknowledgment

The authors would like to thank Jonathan Cameron and Joan Lasenby for their help with the PhaseSpace motion capture system [24].

## References

[1] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo methods in practice*. Springer-Verlag, 2000.

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

[3] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and computing*, vol. 10, pp. 197–208, 2000.

[4] O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

[5] J. Vermaak and A. Blake, "Nonlinear filtering for speaker tracking in noisy and reverberant environments," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, pp. 3021–3024, 2001.

[6] D. B. Ward, E. A. Lehmann, and R. C. Williamson, "Particle filtering algorithms for tracking an acoustic source in a reverberant environment," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 826–836, Nov. 2003.

[7] E. A. Lehmann and R. C. Williamson, "Particle filter design using importance sampling for acoustic source localisation and tracking in reverberant environments," *EURASIP Journal on Applied Signal Processing*, 2006, Article ID 17021, 11 pages.

[8] A. M. Johansson, E. A. Lehmann, and S. Nordholm, "Real-time implementation of a particle filter with integrated voice activity detector for acoustic speaker tracking," in *Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, December 2006, pp. 1004–1007.

[9] N. Gordon, D. Salmond, and A. F. M. Smith, "Novel approach to nonlinear and non-Gaussian Bayesian state estimation," *IEE Proceedings (F)*, vol. 140, pp. 107–113, 1993.
[10] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 24, pp. 320–327, Aug. 1976.
[11] E. A. Lehmann and A. M. Johansson, "Experimental performance assessment of a particle filter with voice activity data fusion for acoustic speaker tracking," in *Proceedings of the IEEE Nordic Signal Processing Symposium*, Reykjavik, Iceland, June 2006, pp. 126–129.
[12] M. Fallon, "Acoustic source tracking using sequential Monte Carlo," Ph.D. dissertation, University of Cambridge, Jun. 2008.
[13] E. A. Lehmann and A. M. Johansson, "Particle filter with integrated voice activity detection for acoustic source tracking," *EURASIP Journal on Advances in Signal Processing*, 2007, Article ID 50870, 11 pages.
[14] D. J. Salmond and H. Birch, "A particle filter for track-before-detect," in *Proceedings of the American Control Conference*, vol. 5, 2001, pp. 3755–3760.
[15] Y. Boers and J. Driessen, "A particle filter multi-target track before detect application," *IEE Proceedings - Radar, Sonar and Navigation*, vol. 151, no. 6, pp. 351–357, 2004.
[16] M. Fallon and S. Godsill, "Multi target acoustic source tracking with an unknown and time varying number of targets," in *Proceedings of Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, May 2008.
[17] W.-K. Ma, B.-N. Vo, S. Singh, and A. Baddeley, "Tracking an unknown time-varying number of speakers using tdoa measurements: A random finite set approach," *IEEE Transactions on Signal Processing*, vol. 54, pp. 3291–3304, Sep. 2006.
[18] B.-N. Vo, W.-K. Ma, and S. Singh, "Localizing an unknown time-varying number of speakers: A bayesian random finite set approach," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, pp. 1073–1076, Mar. 2005.
[19] C. Kreucher, M. Morelande, K. Kastella, and A. Hero, "Particle filtering for multitarget detection and tracking," in *IEEE Aerospace Conference*, Mar. 2005, pp. 2101–2116.
[20] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-based particle filter for tracking multiple interacting targets," in *in Proc. European Conf. on Computer Vision (ECCV*, 2004, pp. 279–290.
[21] M. Fallon and S. Godsill, "Multi-Target acoustic source tracking using track before detect," in *Proceedings of the Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct. 2007, pp. 77–80.
[22] H. Dunn and D. Farnsworth, "Exploration of pressure field around the human head during speech," *Journal of the Acoustical Society of America*, vol. 10, pp. 184–199, January 1939.
[23] E. Lehmann, D. Ward, and R. Williamson, "Experimental comparison of particle filtering algorithms for acoustic source localization in a reverberant room," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, pp. 177–180, Apr. 2003.
[24] PhaseSpace, "http://www.phasespace.com," Website, 2008.