THE UNIVERSITY
of EDINBURGH

Acoustic Source Localisation and Tracking Using Microphone Arrays

Ashley Hughes

A thesis submitted for the degree of Doctor of Philosophy.
The University of Edinburgh.
June 2015

Abstract

This thesis considers the domain of acoustic source localisation and tracking in an indoor environment. Acoustic tracking has applications in security, human-computer interaction, and the diarisation of meetings. Source localisation and tracking is typically a computationally expensive task, making it hard to process on-line, especially as the number of speakers to track increases. Much of the literature considers single-source localisation, however a practical system must be able to cope with multiple speakers, possibly active simultaneously, without knowing beforehand how many speakers are present. Techniques are explored for reducing the computational requirements of an acoustic localisation system. Techniques to localise and track multiple active sources are also explored, and developed to be more computationally efficient than the current state of the art algorithms, whilst being able to track more speakers.

The first contribution is the modification of a recent single-speaker source localisation technique, which improves the localisation speed. This is achieved by formalising the implicit assumption by the modified algorithm that speaker height is uniformly distributed on the vertical axis. Estimating height information effectively reduces the search space where speakers have previously been detected, but who may have moved over the horizontal-plane, and are unlikely to have significantly changed height. This is developed to allow multiple non-simultaneously active sources to be located. This is applicable when the system is given information from a secondary source such as a set of cameras allowing the efficient identification of active speakers rather than just the locations of people in the environment.

The next contribution of the thesis is the application of a particle swarm technique to significantly further decrease the computational cost of localising a single source in an indoor environment, compared the state of the art. Several variants of the particle swarm technique are explored, including novel variants designed specifically for localising acoustic sources. Each method is characterised in terms of its computational complexity as well as the average localisation error. The techniques' responses to acoustic noise are also considered, and they are found to be robust.

A further contribution is made by using multi-optima swarm techniques to localise multiple simultaneously active sources. This makes use of techniques which extend the single-source particle swarm techniques to finding multiple optima of the acoustic objective function. Several techniques are investigated and their performance in terms of localisation accuracy and computational complexity is characterised. Consideration is also given to how these metrics change when an increasing number of active speakers are to be localised.

Finally, the application of the multi-optima localisation methods as an input to a multi-target tracking system is presented. Tracking multiple speakers is a more complex task than tracking single acoustic source, as observations of audio activity must be associated in some way with distinct speakers. The tracker used is known to be a relatively efficient technique, and the nature of the multi-optima output format is modified to allow the application of this technique to the task of speaker tracking.

Lay Summary

This thesis considers acoustic source localisation and tracking in an indoor environment. Acoustic tracking has applications in security, human-computer interaction, and the diarisation of meetings. Source localisation and tracking is typically a computationally expensive task, particularly as the number of speakers to track increases. Much of the literature considers single-source localisation, however a practical system must be able to cope with multiple speakers, possibly active simultaneously, without knowing beforehand how many speakers are present. Techniques are explored for reducing the computational requirements of an acoustic localisation system. Techniques to localise and track multiple active sources are also explored, and developed to be more computationally efficient than the current state of the art algorithms, whilst being able to track more speakers.

The thesis considers the modification of a single-speaker source localisation technique, and improves the localisation speed by considering speaker heights. Estimating height information effectively reduces the search space where speakers have previously been detected, but who may have moved across a room. This is developed to allow multiple non-simultaneously active sources to be located.

This thesis also considers the application of a technique to significantly further decrease the computational cost of localising a single source in an indoor environment, compared the state of the art. Several variants of the technique are explored, including novel variants designed specifically for localising acoustic sources. Each method is characterised in terms of its computational complexity as well as the average localisation error. The techniques' responses to acoustic noise are also considered, and they are found to be robust.

The thesis also studies techniques to localise multiple simultaneously active sources by the extension of the single-source localisation techniques already considered. Several such techniques are investigated, and their performance in terms of localisation accuracy and computational complexity is characterised. Consideration is also given to how these metrics change when an increasing number of active speakers are to be localised.

Finally, the application of the multi-target localisation methods as an input to a multi-target tracking system is presented. Tracking multiple speakers is a more complex task than tracking a single acoustic source, as observations of audio activity must be associated in some way with distinct speakers. The tracker used is known to be a relatively efficient technique, and the nature of the localiser output format is modified to allow the application of this technique to the task of speaker tracking.

Declaration of Originality

I hereby declare that the research recorded in this thesis and the thesis itself was composed and originated entirely by myself in the School of Engineering at The University of Edinburgh.

Ashley Hughes
Edinburgh
June 2015

Acknowledgements

Many people have helped me in my experience as a PhD student. My friends and family have been very supportive throughout, and deserve more thanks than I can ever give.

Firstly, many thanks must go to my supervisor, Dr. James R. Hopgood for his useful feedback, support, and never-ending patience and positive outlook. His suggestions and corrections proved to be invaluable when writing this thesis.

I would also like to express my gratitude to my colleagues at Dialog Semiconductor for their patience, and allowing me to continue to work part time, in order to finish my write-up.

Finally, I would like to express my deepest gratitude to my partner, Maddy, who has had tremendous patience and always encouraged me to keep working, even when it would have been more convenient if I had helped out more with our child. My parents have also been patient and encouraging, and also deserve thanks.

Ashley Hughes
Edinburgh
June 2015

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

2D 2-dimensional

3D 3-dimensional

ALE average location error

ASL acoustic source localisation

ASLT acoustic source localisation and tracking

AV audio-visual

AVS acoustic vector sensor

BSS blind source separation

CL curvilinear

CU coordinate uncoupled

CUDA compute unified device architecture

dB decibels

DOA direction of arrival

DUET degenerate unmixing estimation technique

EKF extended Kalman filter

FE functional evaluation

FIR finite impulse response

FISST finite set statistics

GCC generalised cross correlation

GCC-PHAT generalised cross correlation with phase transform

GCPSO guaranteed convergence particle swarm optimiser

GM-PHD Gaussian mixture probability hypothesis density

GMM Gaussian mixture model

GPU graphics processing unit

HE height estimation

HE-SRC height-estimated stochastic region contraction

HMM hidden Markov model

i.i.d independent and identically distributed

IEKF iterated extended Kalman filter

IIR infinite impulse response

ILD interaural level difference

ISM image-source model

JPDAF joint probabilistic data association filter

MAP maximum *a posteriori*

MHT multiple hypothesis tracker

ML maximum likelihood

MMSE minimum mean-square error

MUSIC multiple signal classification

OMAT optimal mass transfer

OSPA optimal subpattern assignment

PDF probability density function

PHAT phase transform

PHD probability hypothesis density

PSO particle swarm optimisation

RFS random finite set

RIR room impulse response

RMS root mean square

SBF steered beam-former

SINR signal to interference plus noise ratio

SMC sequential Monte Carlo

SNR signal to noise ratio

SPL sound pressure level

SRC stochastic region contraction

SRP steered response power

TDOA time difference of arrival

VAD voice activity detection(or detector)

WoSP waves of swarm particles

Nomenclature

$\bar{\alpha}$	Eyring absorption coefficient
α_i	Sabine energy absorption coefficient for a surface
$\beta_{t\,\vert\,t-1}(x\,\vert\,\zeta)$	PHD Recursion spawn intensity
β_i	Reflection coefficient for a surface
$\delta(\circ)$	Dirac-delta function
δ_l	Locust Swarm new particle position offset
γ_a	Ideal gas adiabatic index
$\gamma_t(x)$	PHD Recursion birth intensity
$\lambda_{\min}$	Smallest measurable acoustic signal wavelength of signal in a a discrete-time sampled microphone system
τ	Time delay
c	The speed of sound in air/an ideal gas
c_c	OSPA cut-off parameter
g_l	Locust Swarm gap parameter
$h_d(\mathbf{x},\,\mathbf{p},\,t)$	Direct-path impulse response
$H_R(\omega)$	Frequency domain representation of an RIR
J_i	Number of locations to evaluate in SRC algorithm at step i
k_B	Boltzmann constant
L_a	Microphone array aperture length
m_a	Eyring air attenuation constant
M_g	Mass of a molecule of an ideal gas

m_t	Niching PSO merging radius threshold
N_i	Number of locations to select for the next SRC algorithm iteration search volume
n_l	Locust Swarm Gaussian random variable
$\mathbf{p}$	The position of a microphone
$\mathbf{p}_g$	Global historical best position of a particle swarm
$\mathbf{p}_i$	Historical best position of particle i
p_o	OMAT/OSPA metric order parameter
$\|r_f\|$	Acoustic far-field threshold distance
r_l	Locust Swarm range parameter
s_l	Locust Swarm spacing parameter
S_r	The acoustically reflective surface area of a room
$s\,(t)$	The signal emitted by an acoustic source
T_1	SRC algorithm convergence scaling factor
T_{60}	Reverberation time of a room, measured as the time taken for the SPL from a source to to decay 60dB below the original level
T_g	Temperature of an ideal gas in Kelvin
u_l	Locust Swarm Uniform random variable
V_i	The search volume in SRC algorithm at step i
$V_i\,(t)$	The velocity of the i$^{\text{th}}$ particle in a particle swarm at time step t
v_l	Locust Swarm new velocity scaling factor
V_r	The volume of a room
V_u	The unit voxel
$\mathbf{x}$	The position of an acoustic source
$X_i\,(t)$	The position of the i$^{\text{th}}$ particle in a particle swarm at time step t

$z_d(\mathbf{x}, \mathbf{p}, t)$ The direct-path signal received at a microphone in position $\mathbf{p}$, from a source at position $\mathbf{x}$

Chapter 1
Introduction

Speech is a primary natural method of communication between people and, because it is so intuitive, it is highly desirable to interact in the same way with machines. Evidence of the trend towards speech interaction with computers can be seen with the recent release of several commercial products such as Siri and Google Voice. It is also desirable to be able to recall exactly what somebody has said in many circumstances. Diarisation - the process of recording who said what, when and where - is the key step to this process. This is especially true in environments such as meetings, where there may be many speakers, and it would be beneficial to be able to review the discussion after it has taken place.

These situations require computers capable of processing audio in order to best extract speech and speaker information, maximising the clarity of the speech and determining speaker positions and identities. This is a complicated task involving many layers of processing, and a possible example system block diagram for an audio diarisation system is shown in Figure 1.1. Hardware is needed to capture the acoustic signals; acoustic sources need to be localised; speakers need to be identified and tracked; their speech signals need to be extracted, possibly interpreted (for example, converted to text) and the data needs to be stored in a suitable format.

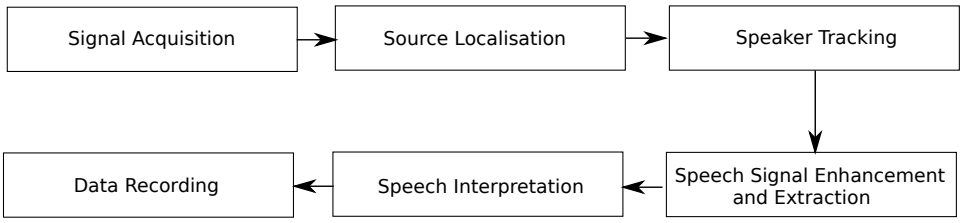


Figure 1.1: An example of an audio diarisation system block diagram.

In addition, audio processing capabilities are useful in a variety of other contexts, for example surveillance networks. In this situation, one might like to track multiple speakers across an area, whilst sometimes focussing on what one particular person is saying. This is somewhat similar to the 'cocktail party problem' [1, 2], which refers to the brain's ability to concentrate on a single voice within a crowded room full of talking people.

Acoustic source localisation is an important component of a speech processing system, and is the primary focus of this thesis, which considers techniques for the localisation and tracking of both single acoustic sources and multiple concurrent sources. For this to be successful, some understanding of acoustics is required, as well as methods of tracking moving targets based on observations of the real world. To this end, the thesis starts off with an introduction to the acoustic source tracking problem and continues with a discussion of some of the issues encountered by these systems.

This chapter of the thesis introduces the problem of acoustic speaker localisation and tracking. A brief review of some of the state of the art methods and approaches to tackling this problem are described and, finally, the structure of the rest of the thesis will be given as a guide to the following chapters.

1.1 Motivation

This thesis considers the problem of speaker tracking in a closed, noisy and reverberant environment, a scenario which is typical of a meeting room or somebody giving a presentation to an audience. Tracking speakers in such an environment might allow the automatic creation of searchable meeting minutes, or allow a security system to identify and track targets by their acoustic activity.

It is assumed that the area of interest is equipped with suitable hardware - namely a microphone array which surrounds the entire area, along with optional video cameras. This setup is illustrated in Figure 1.2, which also illustrates some of the problems associated with speaker localisation.

Note that the use of cameras and microphones together has the potential to make a system robust to non-simultaneous interference or occlusions from the acoustic and visual domains respectively. Combining the two modalities also allows separation between tracking the locations of people within an area and the tracking of which of those people are speaking at any given time.

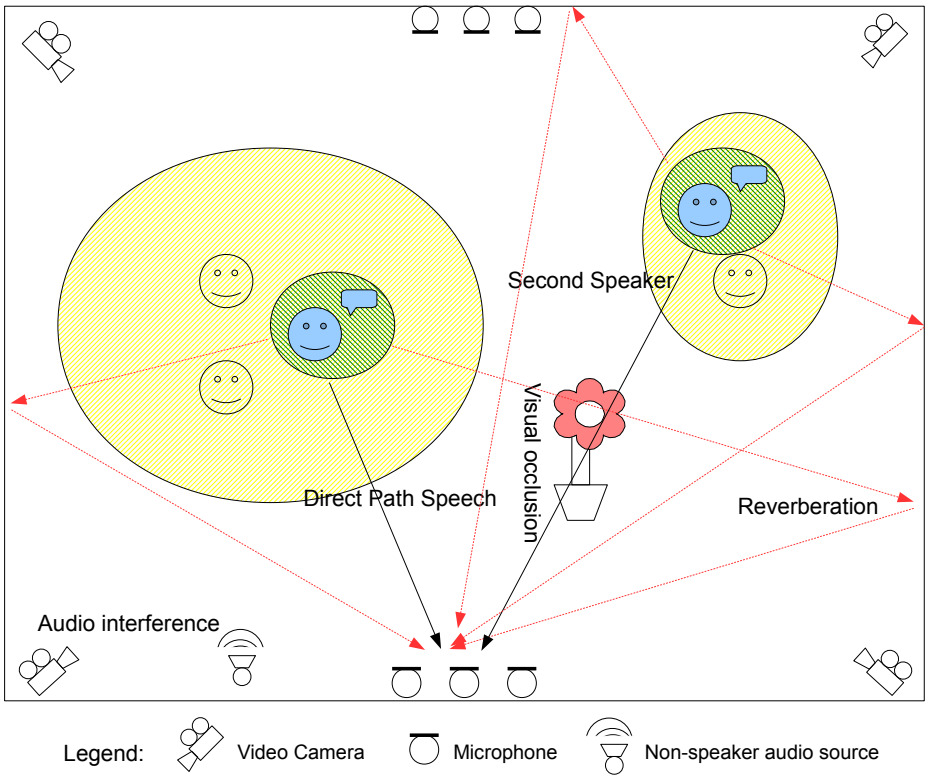


Figure 1.2: A visual overview of the speaker localisation, showing multiple sources, visual occlusion and reverberation effects.

The diagram shows that there are multiple concurrent speakers, which poses a problem, as in order to make any captured audio data intelligible, speech from the target speakers has to be separated out from that of other speakers, as well as from background noise. In addition, sounds made within a room are subject to reverberation, which is essentially a large number of continued reflections of the sound received by an observer after the source has stopped emitting acoustic energy. These continue to be audible for a short time after the sound is created due to reverberation [3], which causes problems for speech processing algorithms. Thus, it becomes useful to be able to infer the location of a speaker from, for example, a microphone array placed around the edges of a room. This might be done in order to better extract speech to be processed from an area detected as containing a speaker, for example using beamforming.

The diagram also indicates that cameras can be used to track people too, with similar problems such as visual occlusion. Note that people can occlude each other or be occluded by other objects within a room. Camera systems also have to extract 3-dimensional (3D) information

about the position of people within a room from their 2D view of the world, and don't generally identify who is speaking without help from an audio system. Typically, an array of cameras can detect the positions of people by identifying face, for example using the Viola-Jones face detector [4].

Such a system cannot typically identify speakers, however, as lip movement would have to be detected for multiple sources at a large distance. Furthermore, lip movement is no guarantee of speech being produced, and it is very possible that speakers may not be directly facing a camera. This means that the tracking of speakers might be considered as tracking the sub-set of people tracked by a camera who are talking. Fusion of data from both domains is desirable to achieve this goal, in order to provide a solution robust to occlusion, as well as potentially ruling out acoustic sources not corresponding to speakers.

As such, speaker tracking systems can be made using joint audio-visual systems, where the hope is that brief occlusion or noisy and unreliable observations in one domain can be mitigated by observations from the other. People detected in video moving behind some occluding object might continue to be tracked by virtue of their association with an audio source. Spurious audio sources or interference from, for example, a radio might be ignored as there is no person tracked on camera to associate them with. One such system is briefly considered in this thesis, in Chapter 4.

1.2 Acoustic Source Localisation

The task of acoustic source localisation (ASL) makes use of an array of microphones with known locations to determine the position of a sound source. Because the locations of the microphones are known, the same acoustic signal arriving at these different positions can be sampled and compared to triangulate an acoustic source. Samples are typically acquired in frames for processing, usually between tens or a few hundreds of milliseconds long, and the signals from the microphones are used to estimate the position of one or more speakers within that time-frame.

Acoustic sensor array behaviour is governed by several factors. Firstly, a distinction is made between near-field and far-field sources, and assuming that a source is in the far-field of a pair of microphones can simplify the mathematics and algorithms used [5] because the source can be assumed as coming from the same direction of arrival (DOA) when considering each

microphone. However, this assumption is generally untrue for the acoustic source localisation and tracking (ASLT) task in a small acoustic environment such as a meeting room, as discussed in Section 2.1.2, and so this thesis assumes that acoustic sources are always in the near-field.

Secondly, the array layout influences the ability of that array to be robust against spatial aliasing. An aperture is a spatial region which, in this case, receives propagating sound waves and a microphone array can be thought of as a continuous aperture sampled at discrete points in space. Similar to the well-known Nyquist minimum sampling frequency for capturing a signal, microphone arrays have a maximum inter-device spacing $d_{\max}$ which must be adhered to in order to avoid spatial aliasing [5]. This is given in Equation (1.1), where $\lambda_{\min}$ is the minimum wavelength of interest. This becomes important in the case of room equipped with an array for voice capture, as comparing signals from microphones situated across the room from each other is likely to result in spatial aliasing. Therefore, when processing signals from a smart-room based microphone array, care must be taken to process data appropriately.

$$d_{\max} = \frac{\lambda_{\min}}{2} \tag{1.1}$$

As an example, consider that the speed of sound is approximately $330\mathrm{ms}^{-1}$. If speech sources are to be located, with an acceptable sampling frequency for speech being 16kHz, and therefore a maximum speech frequency of 8kHz, then the shortest wavelength to be considered will be, approximately, $\lambda_{\min} = 4.1\mathrm{cm}$. This leads to a maximum microphone spacing of $d_{\max} = 2\mathrm{cm}$.

1.3 Acoustic Source Tracking

Source localisation stands in contrast to acoustic source tracking, where the positions of acoustic sources are estimated over multiple frames and an association made between measurements which likely correspond to speech from the same source. At each frame, the result of an ASL step can be used to update an inter-frame estimation of source localisations. This allows moving sources to be continuously tracked, as well as allowing probabilistic methods to be used to attempt to minimise the localisation error.

Tracking of acoustic sources is made harder due to the conditions of the room. There may be sources of acoustic interference or noise, and even without these, sources are likely to be subject to reverberation, all of which can lead to noisy measurements and spurious sources

being detected sporadically. Attempting to track multiple speakers is another problem, as the incoming acoustic wave at each microphone is a mixture of the signals from each source and must be processed to detect each one individually. In addition, the number of sources is usually unknown and must be estimated.

When considering tracking multiple speakers, a distinction must be made between tracking simultaneously speaking sources, and tracking non-concurrent sources. The former problem might be approached by tracking the single speaker at any time, and attempting to detect when the speaker changes, for example by a very sudden change in speaker localisation observations. In contrast, the task of tracking multiple concurrent speakers requires that a localisation algorithm be able to identify multiple source positions concurrently. It also requires that a tracking algorithm be able to identify the number of sources and estimate multiple independent speaker states (tracked positions) from those observations.

The challenging task of multiple-source acoustic speaker tracking is logically split to systematically deal with these various complicating factors. Firstly, measurements must be extracted in some form from the microphone array. These can be in the form of direct Cartesian coordinates for multiple sources, or indirect measurements such as a set of time difference of arrival (TDOA) figures between different microphone pairs.

The measurements are then fed into a tracking algorithm, which can understand how to translate them into absolute positions if necessary. The tracking algorithm deals with associating measurements with tracked targets and updating the estimate of their positions in a way which minimises the localisation error. It does this in the presence of noise, interference sources, and the possibility of an unknown number of multiple speakers. This stage also has to take into account the possible movement of sources, and so often dynamics modelling is used to help estimate an updated position of a speaker given their last known position, their trajectory and a new measurement.

1.4 State of the Art

State of the art ASLT systems employ Bayesian techniques as source tracking algorithms. These include systems which make use of Kalman filter based approaches as well as more complex particle filter based approaches. Whilst Kalman filters are computationally less complex than particle filters, particle filters do not require assumptions about the state propagation lin-

earity or the shape of any noise present in the system. As such, particle filter based approaches are well placed to deal with the non-linear dynamics models of moving speakers [6].

As well as there being many different approaches to the tracking stage, there are also several techniques which can be used for the localisation stage. These measurement extraction techniques often use the indirect DOA measurements returned by generalised cross correlation (GCC) based algorithms such as generalised cross correlation with phase transform (GCC-PHAT), which use TDOA information to estimate which direction a sound came from relative to a microphone pair. By using a number of these DOA measurements together, the position of the source can be determined. Converting these indirect measurements to true source positions is a non-linear process, and the tracker used has to take account of this, however these indirect methods are convenient to use as they are much less computationally expensive to implement than steered beam-former (SBF) methods [7].

An examples of a recent novel single-source localisation techniques is a method of directly tracking a source from TDOA measurements using an extended Kalman filter (EKF) [8]. Another recent technique attempts to implement low complexity localisation by estimating the DOA of a source from techniques based on the multiple signal classification (MUSIC) algorithm. This method is extended in [9] by using particle swarm optimisation (PSO) to search for multiple peaks of the MUSIC spectrum function. Another use of PSO in the context of localisation is given in [10], which extends the particle filter approach of [11] to decrease the localisation time for a single source. PSO is also used in combination with a blind source separation (BSS) method in [12] to extend [11] to track two acoustic sources.

An alternative is to attempt to extract the source position directly using SBF methods, where the 'focal point' of the acoustic array is steered such that the acoustic power picked up is maximised. Because the typical ratio of room sizes, measurable in cubic meters, and the resolution achievable by an acoustic array, measurable in cubic centimetres, is very high, there are very many possible voxels (the 3D analogue of pixels) which could be evaluated within a room. As an example, in an 8x5x3m room, with a volume of 120 cubic metres, is equivalent to 120 million cubic centimetres. A typical microphone might only be able to resolve down to, say, a 3 cubic centimetre voxel, but this is still over 13 million points which could potentially be interrogated in order to find the globally maximum valued voxel.

As such, it is generally impractical to do an exhaustive search to find the global maximum

and so, for example, one technique to use SBF methods for localisations samples the search area randomly and iteratively constricts it until a source is found [13]. The same authors also improve upon this method by altering their sampling strategy to use a set of increasingly fine grids [14].

Whilst these methods on their own are applicable to the single-source tracking case, the multiple source case is more complicated. Measurements made of possible speaker locations must be associated with the most likely currently tracked speaker, a problem known as multi-target data association. The traditional approach to this has been to use a multiple hypothesis tracker (MHT) [15] or a joint probabilistic data association filter (JPDAF) [16].

More recent attempts for multiple source tracking have re-derived Bayesian filtering techniques in the context of finite set statistics [17,18]. This allows the tracker to deal with an unknown and time-varying number of sources, along with spurious and missed measurements. Recent studies have used a random finite set (RFS) based particle filtering approach, however as the number of sources to be tracked increases, the complexity of the algorithm increases too, limiting its ability to perform well in scenarios with a large number of concurrent speakers. It has been suggested that first-order statistical moment approximation of the RFS Bayes filter such as that presented in [19] may be successful [6] in tracking more people whilst keeping the complexity of the calculations involved low compared to particle filter based methods.

Particle filter based systems have been the focus of several multi-source localisation studies recently. Fallon et al. presented a system [20,21] which uses an SBF to evaluate the audio field and determine if sources are active within an 'existence grid'. Sources are then tracked using a particle filter which takes account of whether a source is born (that is, new to the tracker), surviving or becoming inactive. Another approach by Zhong [6] uses a Rao-Blackwellised particle filter to first marginalise out source positions, and then perform data association using the particle filter. A further example of a particle filter based approach is given by Lehmann, who uses the SBF as a pseudo-likelihood function for importance sampling [11,22,23], however this is limited to only tracking one source.

Other state of the art localisation for multiple sources include [24], which models multiple peaks in the TDOA between two microphones as a Gaussian mixture. This allows the regions of space to be identified which are likely to contain acoustic sources, and these are finally located within each selected region using numerical optimisation on a restricted subset of the

TDOAs. This has the disadvantage of the observations acquired being non-linearly related to the source positions, and this thesis avoids this limitation by the use of SBF methods whose results correspond directly to source positions. Some state of the art work makes use of only two microphones and time-frequency masking to perform tracking of two speakers [25], although the error increases as the number of sources to localise is increased.

Other recent novel approaches to the problem have attempted to use BSS methods to perform joint audio separation and localisation [26]. Independent component analysis (ICA) can be used, for example, to perform TDOA estimation [27, 28]. In addition, the degenerate unmixing estimation technique (DUET) BSS method has been used to localise sources [6] and also perform tracking when combined with a modern RFS based multi-target filter [29]. The difficulty involved with these is typically the high computational requirements of the filters used for tracking more than two targets. Zhong [6] has suggested using the Gaussian mixture probability hypothesis density (GM-PHD) filter to address this problem. There is also a difficulty in the unknown complexity and ability of these localisation methods to extract the positions of more than two speakers for filtering. This thesis addresses this issue by presenting a technique for extracting an arbitrary number of sources with only a linear increase in computational requirement for each additional speaker.

Finally, recent work has investigated the use of acoustic vector sensors (AVSs) to perform source localisation [23]. AVSs are special pieces of hardware which measure the acoustic particle velocity as well as the acoustic pressure, as in a traditional microphone. Whilst this approach enables more complex localisation frameworks which utilise the extra sensor data available, these frameworks require special hardware.

1.5 Scope of the work

This work will focus on studying source localisation methods which can efficiently extract positions of multiple concurrent sources. The work will also study tracking systems which can make efficient use of these extracted measurements to track multiple acoustic sources in the presence of noise and reverberation. Such a system would be most useful if it could process data in real time, and so attention is given to improving the efficiency of ASLT methods without sacrificing accuracy.

1.5.1 Section Overview

The second chapter of this thesis introduces some of the basic concepts and tools used when working with acoustic signals. This will cover the basics of some theory about audio, acoustic environments, microphone arrays and acoustic array signal processing. An overview will be given of the acoustic environments used in the work done for this thesis, consisting of both a real room used to make audio recordings and a simulated area used to evaluate the principles used.

Chapter 3 will introduce the Bayesian filtering paradigm, and go on to develop some of the standard tracking algorithms relevant to acoustic source tracking. This will start with an introduction to probabilistic Bayes filtering, followed by practical realisations, namely the Kalman filter and its non-linear extensions. Consideration will then be given to how Bayesian filtering can be used to track multiple sources, and one such implementation - the Gaussian mixture probability hypothesis density filter - will be described.

Chapter 4 will concentrate on non-concurrently active acoustic source localisation and tracking. An existing SBF method of source localisation will be studied and further developed to take account of the vertical axis in such a way that both extends the search area and reduces the number of computations required to arrive at a position estimate.

This technique will be used with a simple single-source tracking system and compared with another, similar, existing technique. The developed technique is found to improve localisation accuracy when compared to the original method, as well as requiring less computational effort.

Chapter 5 explores the use of particle swarms for acoustic source localisation. Particle swarm methods hold potential as a multi-target acoustic localisation technique, as the family of particle swarm algorithms includes multi-target optimisation routines which are all based on the same concept. As such, investigating the performance of particle swarm techniques on the single source scenario is an important first step to building a simultaneous multi-target speaker localisation and tracking system.

The performance of several different swarm localisation techniques are measured and compared, both to each other and to existing source localisation methods. The methods studied are found to be robust to noise and to provide a good localisation performance. Importantly, they also greatly reduce the computational effort required to localise a source.

Chapter 6 expands upon the problem of multiple-source localisation. Inspiration is taken from existing methods of SBF single-source localisation to apply an optimisation technique to the problem of finding multiple local maxima of the SBF response. The performance of this technique in terms of accuracy and complexity is then considered.

Chapter 7 considers the problem of tracking multiple acoustic sources across an area using the localisation technique developed in the previous chapter. This work investigates how multiple speakers can be tracked over many audio frames, and consideration is given to whether or not the number of speakers is known.

Chapter 8 concludes the thesis, providing a summary of the key points of the work undertaken and the conclusions reached. It also points to some suggestions for interesting further work which could be undertaken, built on the results of this thesis.

1.5.2 Contributions

The first contribution is the formalisation of an assumption made about where to search for an acoustic source is on the vertical axis and a method to ensure that all of the vertical space is explored. This method also allows the localisation method to use information from a previous frame to inform a subsequent frame of where it doesn't necessarily have to consider the entire vertical axis, as a nominal head-height has been determined. This method has been shown to reduce the number of calculations required to localise a source whilst at the same time preserving localisation accuracy. The method described is tunable in such a way that if the parameters are chosen correctly, the algorithm reduces to a search on the original search area.

Furthermore, this novel technique is developed fully, such that it can be used in conjunction with a video based system which tracks potential speakers (people) across a room. Peoples' heights can be extracted, which allows the localisation effort to be minimised even with a change in speaker.

The second contribution is the development of a single source localisation method which greatly reduces the computational requirements, when compared to similar steered response power (SRP) based techniques [13]. This is whilst maintaining an acceptable level of localisation error. Novel modifications to the method, which try to adapt it to the specific objective function being used, are made and their effects characterised.

The third contribution is the adaptation of the developed single-source optimisation technique as a localisation method for multiple acoustic sources. The novel application of several variants of this multi-optima localisation technique is explored, and some of the intrinsic associated problems encountered are considered. One of the variants is found to perform very well in terms of both complexity and multi-target localisation error.

The final contribution is the application of a closed-form RFS multiple-source tracker to the acoustic tracking problem. These trackers are tested using the low-complexity localisation techniques previously developed. This result of this is a system which can localise and track more sources than some recently developed tracking techniques [6, 7], whilst maintaining low complexity and low localisation error.

Chapter 2
Background Knowledge

This chapter provides an overview of some of the basic methods used for acoustic signal processing, focusing on techniques relevant to source localisation. Acoustic signal propagation is considered first, followed by a description of the effects caused by the acoustic environment on the signal received by a microphone. Descriptions are given of the simulated and real experimental environments used in the course of this work, finally followed by a short discussion of various useful audio processing techniques such as voice activity detection (VAD).

2.1 Acoustics

A basic understanding of room acoustics and acoustic signal propagation is helpful when formulating the problem of acoustic source localisation and tracking. This section reviews these principles by discussing wave propagation within a room, and the effects of the environment on the received signal at the position of a listener.

2.1.1 Acoustic Waves

Sound waves travel through a medium - in the case considered here, air - from an acoustic source to a listener via multiple paths. The first path to consider is the direct path, which is the shortest path from the source to the listener. This path takes the least time to arrive at the position of the listener, and the speed of an acoustic wave is determined by the propagation medium. Additionally, it is assumed that the medium is homogeneous over the area being considered, and that the speed of sound doesn't vary across the room, or with time. Of course, this is not true as even in an ideal gas, the speed will change with temperature [30], as shown in Equation (2.1), where c is the speed of sound; γ_a is the gas-specific adiabatic constant; k_b is the Boltzmann constant; M_g is the mass of single molecule of the gas and T_g is the absolute temperature in Kelvin. Nevertheless, it is common practice to simplify the model and assume a constant speed of sound.

$$c = \sqrt{\frac{\gamma_a k_B T_g}{M_g}} \tag{2.1}$$

Furthermore, in the context of walking speakers in a room prepared for speech recording, it is reasonably assumed that the speakers won't be moving very quickly (typically around $1\mathrm{ms}^{-1}$, especially relative to the speed of sound (around $340\mathrm{ms}^{-1}$). As such, the Doppler effect is ignored, as the frequency shifts will only be very small.

2.1.2 Near-Field and Far-Field

A source can be assumed to be in the far-field if it is further away than some distance $|r_f|$ from the centre of the aperture formed by an array. $|r_f|$ is given in Equation (2.2), where L_a is the length of the aperture and $\lambda_{\min}$ is the smallest wavelength measurable given the sampling rate [6]. This Equation comes from the Fresnel parameter for acoustic signal intensity at a given distance from a source, which is derived using a signal's wavelength [31].

$$|r_f| > \frac{L_a^2}{\lambda_{\min}} \tag{2.2}$$

As mentioned in Section 1.2, assuming that all sources are in the far-field allows the slight simplification of the acoustic models to be used. Using this assumption, the signals arriving at each microphone from a single source are considered to be far enough away that they can be approximated as plane waves, and therefore they all have the same angle of arrival. Figure 2.1 demonstrates the arrival of sound waves from an acoustic source in the far-field, where the dotted line represents a wavefront. However, this simplification is not applicable to this thesis, and the near-field model must be used. Figure 2.2 demonstrates the arrival of sound waves from a source in the near-field. Note that in Figure 2.2, the wavefronts incident on the microphone are spherical waves, and that the angle of incidence at each microphone is different.

The acoustic sources in this thesis are assumed to be in locations such that they are always in the near-field of all microphone pairs. In the formulation of the Steered Response Power (SRP) later in this thesis, each pair of microphones will be considered as an array, thus the microphone separation will be considered as the array aperture. The near-field assumption

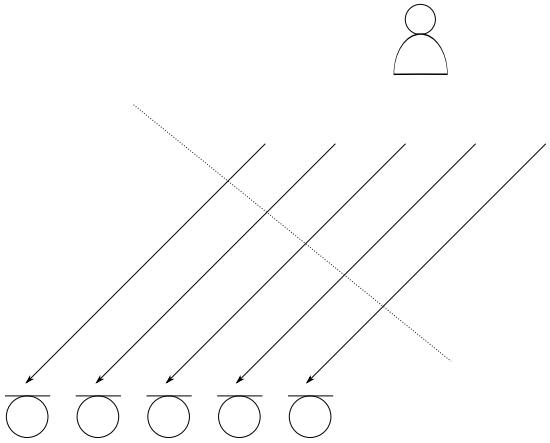


Figure 2.1: Acoustic plane waves received from a source in the far-field

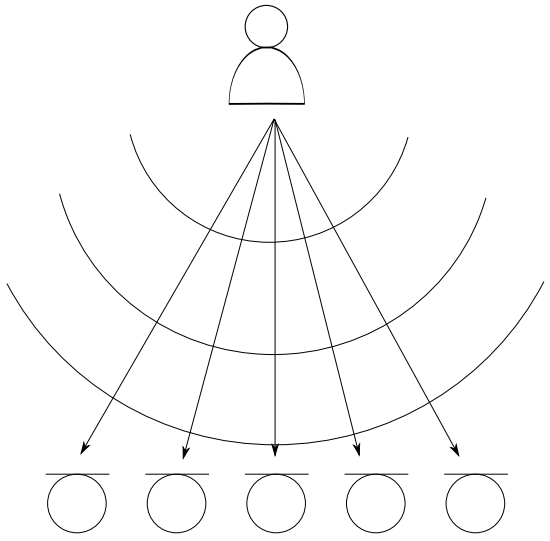


Figure 2.2: Acoustic spherical waves received from a source in the near-field

can be justified by considering a typical sampling frequency of 44.1kHz (maximum received frequency of 22.05kHz because of Nyquist limit), which leads to a minimum wavelength of $\lambda_{\min} = 0.015\mathrm{m}$. With a microphone separation of 30cm, this leads to a cut-off of 6.01m. This violates the far-field assumption in most of the office environment considered (the largest room considered is $(8 \times 5 \times 3)\,\mathrm{m}^3$).

2.1.3 Reverberation

Reverberation is an effect caused by multiple echoes of a sound source arriving at the position of a listener (or listening device) at multiple short times after the arrival of the sound from the direct path. The differing time delays are caused by the different lengths of the multiple different indirect paths that the audio takes, where the speed of sound in air is assumed to be constant across the environment. To the human ear, echoes arriving around 50ms [32] after the initial arrival of the direct path sound are indiscernible from each other, and this causes the multiple echoes in reverberation to be indistinguishable from each other. When attempting source localisation, these multiple echoes can degrade the performance of many algorithms and as such, it is important to take them into account when studying these techniques.

The sound waves from a source can be broadly split into three categories:

- The direct path wave - this is the wave which travels directly from the sound source to the observer.
- The early reflections - these reflections typically arrive at the observer's position up to 100ms after the direct path wave.
- The late reflections - the arrival of these reflections are separated by very small times and are indistinguishable from each other, which effectively results in a diffuse sound field. A diffuse sound field consists of an infinite number of sound waves arriving from uniformly distributed directions, with random phase relations [33].

The relative arrival times and attenuations are illustrated in Figure 2.3, where the attenuations of subsequent reflections can be explained by the frequency dependent absorption of sound waves by air [34].

The pattern of arrival times of the different reflections is highly dependent on the source location

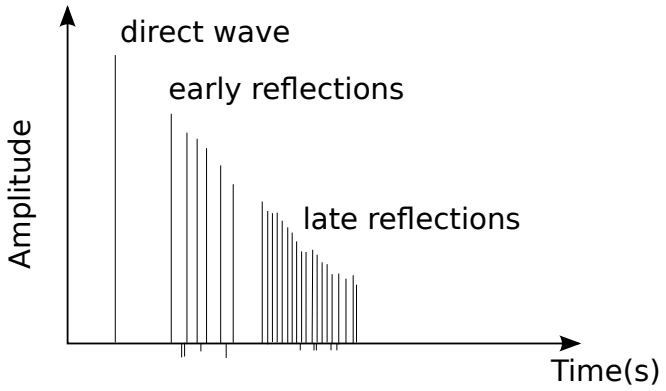


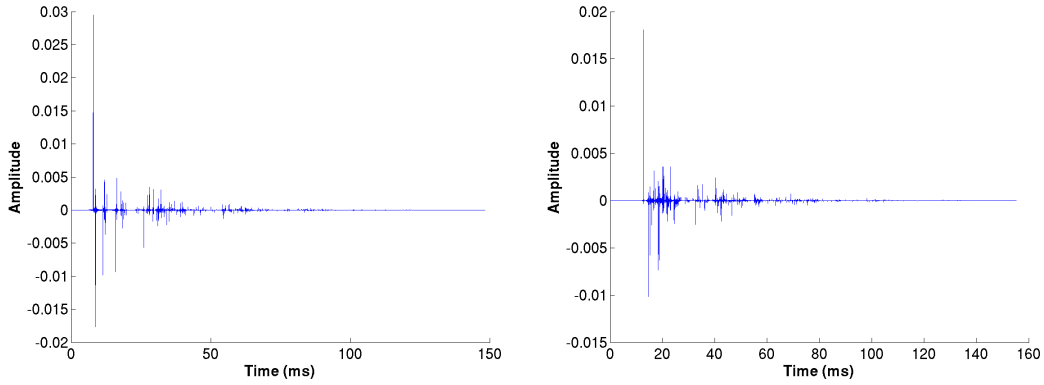
Figure 2.3: Illustration of the arrival of a sound wave and its echoes

within the room, the location of the observer, and the characteristics of the room itself. This is especially relevant to acoustic source localisation, as this characterisation, known as the room impulse response (RIR), changes for each source to be tracked as it moves, as well as when microphones move or the room characteristic itself changes, such as when a door is opened or a table is moved. Figures 2.4a and 2.4b show the RIR for a source within a simulated room for two different microphones.

2.1.4 Room Impulse Response

In order to understand the signal received by a microphone from a source, the effects on the wave caused by reverberation must be considered. The signal arriving from a source at position vector $\mathbf{x}$, at the position of a microphone at location vector $\mathbf{p}$, will be the sum of the direct path signal and all of its reflections.

Firstly, the received direct path signal $z_d(\mathbf{x}, \mathbf{p}, t)$ will be modelled as the convolution of the emitted signal $s(t)$ with direct path impulse response $h_d(\mathbf{x}, \mathbf{p}, t)$, as shown in Equation (2.3) [35]. This assumes that delay and attenuation are not frequency dependent, as the acoustic medium is air in a small indoor environment as opposed to, say, the large distances and different medium involved with underwater acoustics such as sonar.



(a) Microphone 1 (b) Microphone 2

Figure 2.4: Simulated room impulse responses (RIRs) of a source position at two different microphone positions

$$\begin{aligned} z_d(\mathbf{x},\, \mathbf{p},\, t) &= s(t) * h_d(\mathbf{x},\, \mathbf{p},\, t) \\ &= s(t) * \frac{a}{r}\delta(t-\tau) \\ &= \frac{a}{r}s(t-\tau) \end{aligned} \tag{2.3}$$

This set of equations shows that the direct path impulse response is simply a delayed and scaled version of the signal emitted by the source. The delay τ is given by Equation (2.4) which is simply the Euclidean distance between the microphone and the source, r, divided by the speed of sound, c.

$$\begin{aligned} \tau &= \frac{r}{c} \\ \tau &= \frac{\parallel \mathbf{x}-\mathbf{p} \parallel}{c} \end{aligned} \tag{2.4}$$

The scaling factor $\frac{a}{r}$ is determined by an acoustic medium dependent constant a and also varies inversely with r. Note that this is a consequence of the inverse square law for acoustic waves. The sound intensity is subject to the inverse square law, but the sound pressure, where an acoustic wave is described as a pressure wave, is subject to an inverse distance law. $\delta(t-\tau)$ is the time-delayed standard Dirac-delta function. The Dirac-delta has a value of zero everywhere

except at $t = 0$, and an integral of one over x. It is defined mathematically in Equation (2.5).

$$\delta(t) = \begin{cases} +\infty, & t = 0 \\ 0, & t \neq 0 \end{cases} \tag{2.5a}$$

$$\int_{-\infty}^{+\infty} \delta(t)\,\mathrm{d}t = 1 \tag{2.5b}$$

The reflected signals are similarly modelled using an impulse response, as shown in Equation (2.6), where $z_r(\mathbf{x},\, \mathbf{p},\, t)$ is the signal received at the microphone from reflections, and $h_r(\mathbf{x},\, \mathbf{p},\, t)$ is the impulse response of all of the reflecting surfaces for a source at position $\mathbf{x}$ and microphone at position $\mathbf{p}$.

$$z_r(\mathbf{x},\, \mathbf{p},\, t) = s(t) * h_r(\mathbf{x},\, \mathbf{p},\, t) \tag{2.6}$$

This modelled room impulse response will be similar to the real examples shown in Figures 2.4a and 2.4b. Typically, an audio model includes reflections from the walls, ceiling and floor. In reality, there are also reflections from the surface of all objects within a room, but for simplicity these are not modelled. This is firstly because they are not easily captured by the practical impulse response modelling algorithm described in Section 2.1.6, and also because it is assumed that their overall contribution to the room impulse response will be small compared to that of large reflective surfaces such as walls.

In a real room, there will be many objects which reflect sound, each with a different coefficient of absorption. Finally, Equation (2.7) shows the complete model for the signal received at a microphone $z(\mathbf{x},\, \mathbf{p},\, t)$ due to a sound source.

$$\begin{aligned} z(\mathbf{x},\, \mathbf{p},\, t) &= s(t) * (h_d(\mathbf{x},\, \mathbf{p},\, t) + h_r(\mathbf{x},\, \mathbf{p},\, t)) \\ &= \frac{a}{r} s(t - \tau) + s(t) * h_r(\mathbf{x},\, \mathbf{p},\, t) \end{aligned} \tag{2.7}$$

2.1.5 Reverberation Time

The reverberation time of a room is defined by the time taken for the sound pressure level (SPL) from an acoustic source to decay to some threshold below the original level. The level used to describe this effect is often -60dB, given the symbol T_{60}. Other levels can be used however, such as -20dB (T_{20}). Equation (2.8) gives the Eyring expression for T_{60} [35] in terms of the room volume V_r; the total reflecting surface area S_r; m_a, a constant which takes air attenuation into account and $\bar{\alpha}$, an absorption coefficient given by Equation (2.9). This equation allows each reflecting surface to be modelled as having a different absorption coefficient, which allows different surface materials, such as hard walls or soft carpet, to be taken into account. In Equation (2.9), the surface area for the i^{th} surface in the room is denoted as S_i, and the absorption coefficient for that area is denoted α_i.

$$T_{60} = \frac{0.161 V_r}{-S_r \ln(1-\bar{\alpha}) + 4 m_a V_r} \tag{2.8}$$

$$\bar{\alpha} = \frac{1}{S_r} \sum_i S_i \alpha_i \tag{2.9}$$

There are several variations on the reverberation time equation [35, 36], for example the Sabine equation takes account of the absorption coefficients as an expansion of the logarithm term, ignoring all but the initial term of the expansion. However the variations all have a similar form and produce similar results, namely that the decay time is increased in the presence of large reflective surfaces. Finally, Kuttruff notes that the Millington-Sette formula can result in reverberation time of 0 when a reflecting area has an absorption coefficient of 1, which doesn't make sense in real life!

2.1.6 Image Method for Room Impulse Response

With an understanding of reverberation time and the nature of the room impulse response, it can be seen that modelling acoustic sources within a simulation environment might be complex. However, approximations of the transfer functions between sources and microphones can be made, allowing the effects of different reverberation times to be studied.

For simulation purposes, it is useful to be able to determine the RIR of a given environment (room size and reverberation time), source and set of microphone locations. These can then be convolved with the output of an acoustic source model in order to provide a reasonable approximation of how that output might be observed by a microphone. The RIR changes for every speaker location relative to a set of microphones, so to simulate a moving speaker, a RIR must be calculated for every location the speaker visits on their path. It must also be calculated separately for every microphone at each location along the speakers path, which ultimately means that a large number of RIRs must be calculated for a realistic simulation of a moving acoustic source being recorded by a set of microphones.

The image-source model (ISM) method [37] assumes that the room of interest is rectangular in shape, as this simplifies the model and and still allows the calculation of RIRs for realistic rooms. Using this technique, a speaker is modelled as a point source within a rectangular area, emitting a single frequency pressure wave. The boundary condition of a wall is then satisfied by placing an image of the point source symmetrically on the far side of the wall. This is repeated for each wall, and extended by repeatedly mirroring each mirror image, as shown in 2D in Figure 2.5 where the white circle represents the original sound source and the black circles represent image sources. The pressure wave from each wall can be summed with the wave from the original source to form the wave received at a microphone location. Because this pressure wave represents the frequency response of the room to the original source, the RIR can be found via the Fourier transform of this expression [37].

The standard ISM method allows surfaces (walls, the ceiling and the floor) to be modelled as non-rigid surfaces by allowing each of them to have an associated reflection coefficient, β_i, given in Equation (2.10) where α_i is the Sabine energy absorption coefficient for that surface, as used in Equation (2.9). By taking these into account when considering the frequency response of a room, the Fourier transform of this frequency response will yield the RIR as before, but with non-rigid surfaces.

$$\alpha_i = 1 - \beta_i^2 \tag{2.10}$$

Because the response of a room at an individual microphone to an input signal at a given position is given by the convolution of that signal with the RIR, the RIR can be conveniently

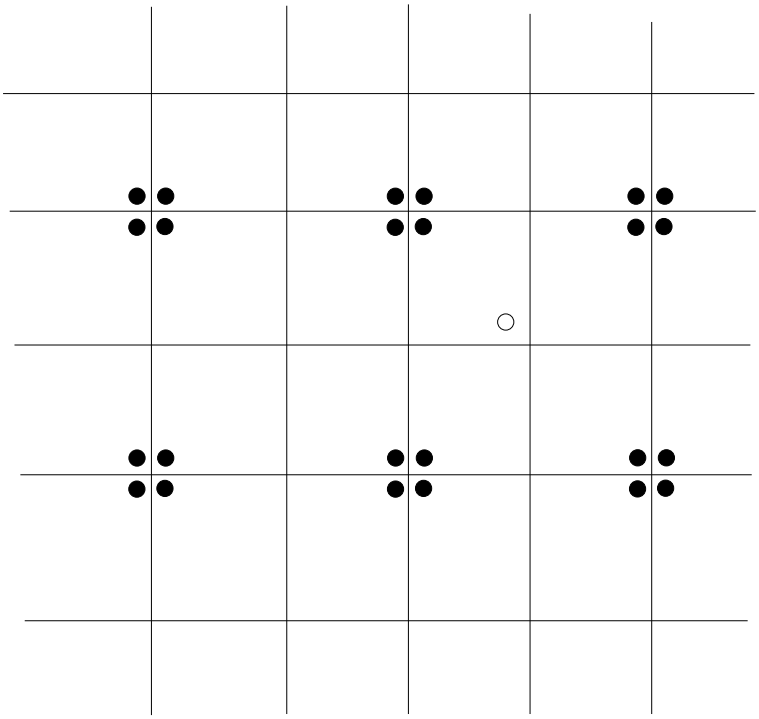


Figure 2.5: Illustration of the image method for room impulse response. The source position is marked by the white circle in the corner of a square representing a room. The other squares represent mirror images of the room, containing mirror images of the source marked by black dots.

implemented using a finite impulse response (FIR) filter. Note that the number of filter taps will be very large, especially in a highly reverberant environment, and so it is of practical interest to approximate the RIR using an infinite impulse response (IIR) filter [38]. Furthermore, the RIR needs to be calculated separately for every combination of source position and each microphone position, leading to a large number of large filters.

2.1.6.1 Improved Image method

The time delay used in [37] is limited in simulation to being a discrete integer multiple of the sampling frequency used. This results in a coarse RIR which requires high pass filtering to be more realistic. It is suggested [39] that the ISM calculations be implemented in the frequency domain, as shown in Equation (2.11). This would allow these time delays to be represented as required. The RIR can then be found using the inverse Fourier transform of $H_R(\omega)$, the representation of the RIR in the frequency domain. $A(\mathbf{u}, \mathbf{l})$ is an amplitude attenuation function [39], parametrised by an image source indexed by $\mathbf{u}$ and a target position for RIR generation, $\mathbf{l}$. $\tau(\mathbf{u}, \mathbf{l})$ represents the time delay of the original audio signal from the image source location $\mathbf{u}$ to the target position $\mathbf{l}$. Note that the summation over $\mathbf{u}$ is used to represent the sum over each dimension of the vector $\mathbf{u}$, and the sum over $\mathbf{l}$ represents the sum over the dimensions of l, over each possible mirror position, of which there are in theory infinitely many.

$$H_R(\omega) = \sum_{\mathbf{u}=0}^{1} \sum_{\mathbf{l}=-\infty}^{\infty} A(\mathbf{u}, \mathbf{l})\, \mathrm{e}^{-j\omega\tau(\mathbf{u}, \mathbf{l})} \tag{2.11}$$

By using the frequency domain approach, the tail of the RIR decays unrealistically, as the reflection coefficient β can become negative, where it is assumed to be positive in the original algorithm. It is suggested that an alternative definition of β be used as shown in Equation (2.12) [35, 39], where ψ is the angle of incidence at a boundary, and ζ is a reflection coefficient for that boundary.

$$\beta = \frac{\zeta \cos(\psi) - 1}{\zeta \cos(\psi) + 1} \tag{2.12}$$

The energy decay envelope $E(t)$, given in Equation (2.13), is expressed in decibels (dB) and can be used to estimate the reverberation time, T_{20} or T_{60}, using a known RIR in the time

domain, $h\left(t\right)$.

$$E\left(t\right) = 10\log_{10}\left(\frac{\int_{t}^{\infty} h^{2}\left(\xi\right)\mathrm{d}\xi}{\int_{0}^{\infty} h^{2}\left(\xi\right)\mathrm{d}\xi}\right) \tag{2.13}$$

The work in [39] approximates the energy decay curve, which allows for the development of a fast method for RIR simulation [40]. This method, which is the method used in this thesis for simulation, models the early reflections using the enhanced ISM method described in [39]. The late reflections are modelled as decaying random noise, where the rate of decay is calculated from the energy decay envelope, also described in [39].

2.2 Signal Model

This section covers the modelling of the signals received from a set of sources by a set of microphones. The model will take account of multiple speakers, and the modelling of reverberation is simplified to aid the development of source localisation algorithms.

2.2.1 Microphone Model

In this thesis, it is assumed that the positions of the microphones within the environments under consideration are known. This is because the area to be considered is assumed to be a room equipped specifically for the ASLT task. In a system containing L microphones, the position of each microphone is denoted by $\mathbf{p}_{l}$, with l being an integer between 1 and L. Within the room, there are a number of concurrent active speakers whose positions are unknown, and it is these positions that we would like to find. Audio data from a microphone is typically processed in frames of a given length of time, typically tens or a few hundreds of milliseconds long. It is assumed that the movement of an acoustic source within a frame is insignificant and that therefore, the position of the m^{th} source out of M_{t} active sources at time t within an audio frame is a constant within that frame, denoted by the vector $\mathbf{x}_{m,t}$.

By the principle of superposition, the received acoustic signal, $z_{l}\left(t\right)$, at microphone l is given by sum of each signal produced by each source, subject to a reverberant environment, as well as some independent noise. This is illustrated in Equation (2.14), where $*$ denotes the convolution operation and $h\left(\mathbf{p}_{l},\, \mathbf{x}_{m,t}\right)$ is the RIR between the source at position $\mathbf{x}_{m,t}$ and the microphone

at position $\mathbf{p}_l$, as discussed in Section 2.1.3.

$$z_l(t) = \sum_{m=1}^{M_t} s_m(t) * h(\mathbf{p}_l, \mathbf{x}_{m,t}) + \bar{v}_l(t) \tag{2.14}$$

In this equation, $s_m(t)$ represents the signal produced at acoustic source m, and $\bar{v}_l(t)$ represents the noise term which consists of the sum of two independent components. The first of these components is the channel noise, denoted by $\bar{v}_{1,l}(t)$, and the second component is the interference signal created by multiple sound sources which are not directly of interest. These sources are typically directional sources of unwanted interference such as footsteps or the noise generated by a computer fan. Similar to Equation (2.14), these sources are subject to superposition and reverberation, and as such, their contribution to $\bar{v}_l(t)$, $\bar{v}_{2,l}(t)$ is given in Equation (2.15).

$$\bar{v}_{2,l}(t) = \sum_{j=1}^{J_t} \bar{s}_j(t) * h(\mathbf{p}_l, \bar{\mathbf{x}}_{j,t}) \tag{2.15}$$

This equation assumes that there are J interfering sources, whose signals are represented by $\bar{s}_j(t)$. As in Equation (2.14), these signals are each convoluted by the RIR and summed together to form the total interference signal at the microphone position $\mathbf{p}_l$.

2.2.2 Free-field Model

The free field model, also known as the direct field model, of an acoustic environment takes account of reverberation by including reverberant signals as noise terms. The received signal at a microphone is then taken to consist of the direct path signals from the sources to the microphone, with added noise. This effectively means that any TDOA or other position information is assumed to come from a true source, not from reflection. This assumption breaks down in highly reverberant environments, as the reflection signals can be at relatively high levels, even compared to the direct path signal. The number of incorrect time delay estimates in a delay estimating system has been found to start increasing with ($T_{60} > 0.15$s), and to dominate the results with ($T_{60} > 0.5$s [41]. Highly reverberant environments can also invalidate the assumption that the noise term is independent from the direct paths signal, as well as the assumption that noise term fits a Gaussian normal distribution.

The free field model is mathematically defined by considering Equation (2.7) in the context of multiple speakers. Equation (2.16a) shows the multi-source received signal obtained by considering Equation (2.7) for multiple sources. Equation (2.16b) shows the reverberant part of the model grouped into a noise variable, $v_l(t)$. In these equations, $\tau_{m,l}$ represents the time delay of the signal from source m to microphone l.

$$z_l(t) = \sum_{m=1}^{M_t} \frac{a}{r} s_m(t - \tau_{m,l}) + \sum_{m=1}^{M_t} s_m(t) * h_r(\mathbf{p}_l, \mathbf{x}_{m,t}) + \bar{v}_l(t) \tag{2.16a}$$

$$z_l(t) = \sum_{m=1}^{M_t} \frac{a}{r} s_m(t - \tau_{m,l}) + v_l(t) \tag{2.16b}$$

Equation (2.17) shows that as before, the noise at each microphone is the sum of of several components. In this case, the total noise contains both the channel noise, $\bar{v}_{1,l}(t)$, and the noise from interference sources, $\bar{v}_{2,l}(t)$, as well as a new term describing the environmentally dependent reflected signals of the source signal.

$$v_l(t) = \bar{v}_l(t) + \sum_{m=1}^{M_t} s_m(t) * h_r(\mathbf{p}_l, \mathbf{x}_{m,t}) \tag{2.17}$$

2.3 Audio Localisation Methods

As mentioned in Chapter 1, there are two categories of source localisation methods - those which return direct measurements of source locations and those which return indirect TDOA measurements. Direct methods are typically based on steered beamforming methods. The output of these direct methods can be used directly by a tracker to smooth position estimates over time. On the other hand, TDOA methods require that a tracker extract the sources' true positions from the non-linear relationship between multiple TDOAs and a position in Cartesian coordinates.

This section will introduce some of the commonly used methods to implement both direct and indirect localisation. It will start off by introducing the GCC and how it can be used to obtain TDOA measurements. It will then discuss the steered beamforming using microphone arrays, which forms the basis for the direct measurement method of localisation.

2.3.1 Time Difference of Arrival

The TDOA of a signal from an acoustic source at a pair of microphones is related to the angle of arrival of that signal at the array. This relationship is dependent on the inter-microphone spacing, as shown in Equation (2.18) (for the far-field), where θ is the angle of arrival, d is the microphone separation, and t is the TDOA of the signal between the two microphones. In the near-field, the TDOA doesn't relate directly to a an angle of arrival, but rather to a set of angles and associated distances. TDOA can be used for acoustic source localisation by estimating the time difference of a signal arriving at a pair of microphones using cross-correlation. This cross-correlation technique can then be developed into a technique (Section 2.3.2.3) which uses an arbitrary time-difference to interrogate a location in space regardless of whether or not it is in the near-field or far-field.

$$\cos\left(\theta\right) = \frac{ct}{d} \tag{2.18}$$

2.3.1.1 Cross Correlation

The cross-correlation of two signals gives a measure of their similarity when one of the signals is delayed relative to the other, and is defined in Equation (2.19) given a time delay τ and and two functions of time, f and g.

$$\left(f \star g\right)\left(\tau\right) = \int_{-\infty}^{\infty} f^{*}\left(t\right) g\left(t+\tau\right) \mathrm{d}t \tag{2.19}$$

This can also be expressed in the frequency domain, as shown in Equation (2.20). Here, $G\left(\omega\right)$ and $F\left(\omega\right)$ are the Fourier transforms of $g\left(t\right)$ and $f\left(t\right)$ respectively.

$$\left(f \star g\right)\left(\tau\right) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F^{*}\left(\omega\right) G\left(\omega\right) \mathrm{e}^{j\omega\tau} \mathrm{d}\omega \tag{2.20}$$

Whilst working in the context of the free-field model, the Fourier transform of the emitted signal $s_m\left(t\right)$ can be represented at each microphone as a time delayed version using the shift property of the Fourier transform. This is shown in Equation (2.21), where $Z_l\left(\omega\right)$ is the Fourier

transform of the signal received at the microphone, $\tau_{m,\,l}$ is the time delay, as before, and $S_m\,(\omega)$ is the Fourier transform of $s_m\,(t)$. This equation also includes a noise term in the frequency domain, $V\,(\omega)$, which is the Fourier transform of $v_l\,(t)$.

$$Z_l\,(\omega) = \mathrm{e}^{-j\omega\tau_{m,\,l}} S_m\,(\omega) + V_l\,(\omega) \tag{2.21}$$

The cross-correlation of the received signals at two microphones is then denoted by $R_{l_1,\,l_2}\,(\tau)$ and is expressed in Equation (2.22), where v_1 and v_2 are the noise signals at microphones 1 and 2 respectively.

$$\begin{aligned} R_{l_1,\,l_2} &\approx \int_{-\infty}^{\infty} S_m\,(\omega)\, S_m^*\,(\omega)\, \mathrm{e}^{-j\omega\left(\tau_{m,\,l_1} - \tau_{m,\,l_2}\right)} \mathrm{e}^{j\omega\tau}\mathrm{d}\omega \\ &= R_{s_m s_m}\left(\tau - \left(\tau_{m,\,l_1} - \tau_{m,\,l_2}\right)\right) + R_{v_1 v_2}\,(t) \end{aligned} \tag{2.22}$$

Here, $R_{s_m s_m}$ is the autocorrelation of the source signal $s_m\,(t)$ and $R_{v_1 v_2}\,(t)$ is the cross correlation of the two noise signals. Because these noise signals are assumed to be independent and identically distributed (i.i.d), their cross correlation should be 0. Therefore, the cross-correlation of the signals from the microphones should be maximised at the time delay τ which is equal to the TDOA, $\tau_{m,\,l_1} - \tau_{m,\,l_2}$. The assumption that the noise terms do not contribute holds only so far as their being i.i.d holds true. In a highly reverberant environment, this assumption breaks down [41], and peaks of the cross-correlation between two microphones no longer necessarily reflect time delays at which there is a true TDOA.

With this in mind, the TDOA of a source can be estimated as $\hat{\tau}_{l_1 l_2}$ from the cross-correlation of the signals from two microphones by finding the time delay τ which maximises this function, as expressed in Equation (2.23).

$$\hat{\tau}_{l_1 l_2} = \arg\max_{\tau} R_{l_1,\,l_2} \qquad \tau \in [-\tau_{\max},\, \tau_{\max}] \tag{2.23}$$

τ_{max} is defined as the maximum delay possible, which occurs when the microphone pair and the source positions are collinear. It is expressed in Equation (2.24) as the time taken for the sound wave to travel directly from one microphone to the other.

$$\tau_{max} = \frac{\| \mathbf{p}_1 - \mathbf{p}_2 \|}{c} \tag{2.24}$$

In practice, this set of possible time delays is limited by the sampling frequency used, and the cross-correlation calculation is made on a discrete set of samples, limited in number by the frame size.

2.3.1.2 Generalised Cross Correlation (GCC)

Because the period of observation is limited - samples are processed in frames - the true cross-correlation cannot be exactly calculated and must be estimated using the available data. Equation (2.25) shows the cross correlation, described in the frequency domain in Equation (2.20), described more simply as the inverse Fourier transform of the cross spectral density G_{l_1,l_2} by the Wiener-Khinchin theorem.

$$R_{l_1,l_2}(\tau) = \int_{-\infty}^{\infty} G_{l_1,l_2}(\omega)\, \mathrm{e}^{j\omega\tau} \mathrm{d}\omega \tag{2.25}$$

The GCC introduces two filters, one applied to each signal, which are applied before the multiplication of the signals. When these filters, $H_{l_1}(\omega)$ and $H_{l_2}(\omega)$, are equal to one at all frequencies, ω, then the filtering step is effectively removed and the algorithm reduces to the simple approximation of the cross-correlation.

Equation (2.26) shows the complete general correlation between two input signals in the frequency domain, taking into account the pre-filtering step, with the frequency weighting $\psi(\omega)$ given by these filters expressed in Equation (2.27).

$$R_{l_1,l_2}(\tau) = \int_{-\infty}^{\infty} \psi(\omega)\, G_{l_1,l_2}(\omega)\, \mathrm{e}^{j\omega\tau} \mathrm{d}\omega \tag{2.26}$$

$$\psi(\omega) = H_{l_1}(\omega)\, H^*_{l_2}(\omega) \tag{2.27}$$

The weighting $\psi(\omega)$ must be chosen as a compromise between the output of sharpness of peaks at time offsets corresponding to true time delays, and the sensitivity of the output to

errors caused by the finite observation time. This factor can cause instability in the output, leading to peaks surrounded by noise.

2.3.1.3 Generalised Cross Correlation Weightings

Several different weightings are examined in [42], and the most prevalent of these in the literature is the phase transform (PHAT) weighting, defined in Equation (2.28).

$$\psi(\omega) = \frac{1}{|G_{l_1, l_2}(\omega)|} \tag{2.28}$$

Developed as an ad-hoc technique to avoid the spreading out of peaks of the cross-correlation, the PHAT weighting has been shown to be relatively robust to reverberation when estimating TDOAs, along with a technique based on linear regression [43].

2.3.1.4 Localisation from Generalised Cross Correlation

Given the GCC between signals from two microphones, the TDOA of a single source will correspond to a peak of the GCC function at that delay. This can be converted to an angle of arrival as described in Section 2.3.1 for far-field sources.

When multiple microphone pairs are available, the angle of arrival can be calculated for each pair given that pair's estimated TDOA. These DOA lines should all intersect at roughly the same place in space, as shown in Figure 2.6, where the black circle represents the true source location. Of course, calculated TDOAs are estimations, and so it is unlikely that the intersections between more than two DOA lines will be at a single point. This can be taken into account by finding a point that is close to all lines in the least-squares sense using linear intersection [44]. Multiple TDOAs are used directly in [8] to estimate a source location using a Kalman filter.

2.3.2 Steered Beamforming

The steered beamforming approach to audio source localisation uses the microphone array to enhance the signal power from a chosen angle of arrival. Beamforming can be thought of as a spatial filter, where signals not coming from the direction of interest are attenuated according to the array directivity pattern. Using beamforming techniques, the angle of interest can be

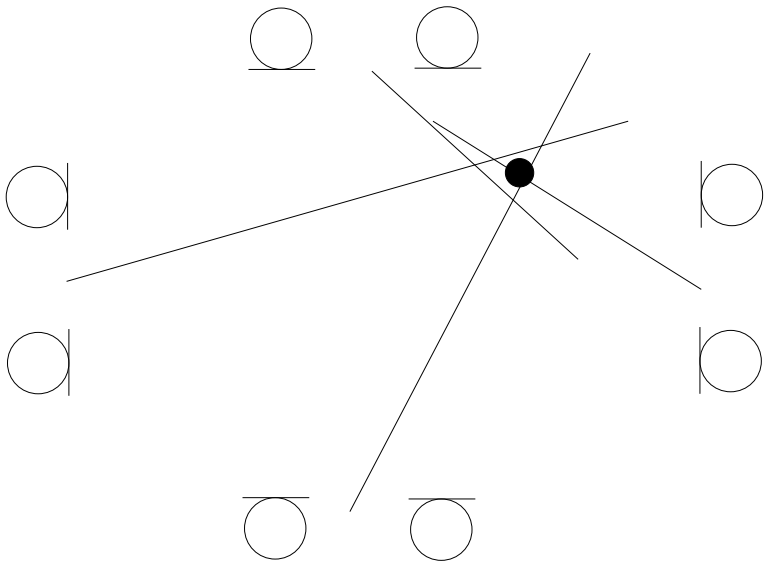


Figure 2.6: Intersection of DOA lines at a source position

changed without physically moving the microphone array.

2.3.2.1 Microphone Arrays

To perform beamforming, a microphone array can be configured in one of two ways, which determines the beamforming methods which can be used [45]. The broadside array configuration is constructed with a line of microphones arranged perpendicularly to the expected direction of arrival of sound waves, as illustrated in Figure 2.7. The frequency response of a broadside array depends on the angle of arrival of the incident sound wave, with a flat response at 0°and angle dependent null frequencies introduced for signals incident at other angles [5]. Note that altering the relative delays between the signals allows these null frequency angles to move, which is the basis of steered beamforming.

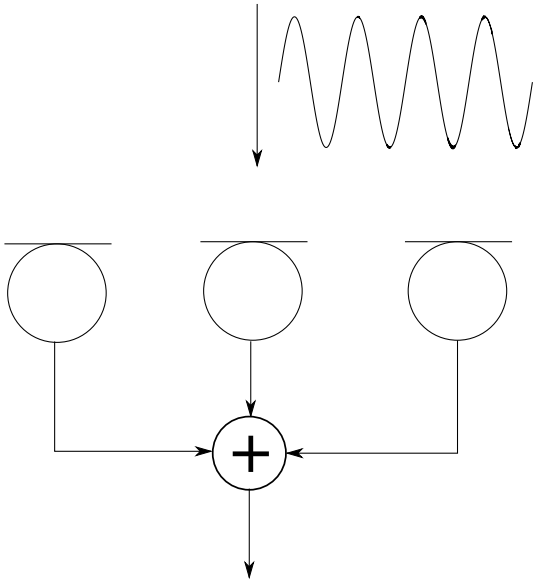


Figure 2.7: Broadside Array Configuration

In contrast, an endfire array- also called a differential array - consists of microphones arranged in a line parallel to the expected direction of arrival of sound waves, as illustrated in Figure 2.8. This figure shows an example array set up for second-order differential beamforming [45], and the important point to note is that the microphones are colinear with the expected direction of a source, and so a simple delay-and-sum beamformer is not used. Note that broadside arrays are not generally intended to be able to localise sources which are colinear with the microphones, and the arrays used in this thesis are broadside arrays, so endfire arrays are not considered further.

This configuration allows the creation of cardioid pick-up patterns in the direction of the array line using an inversion and summing operation. Figure 2.8 shows how this might be achieved for an array of three microphones. An array in this configuration typically has a high pass frequency response with a notch, or null frequency, dictated by the microphone spacing.

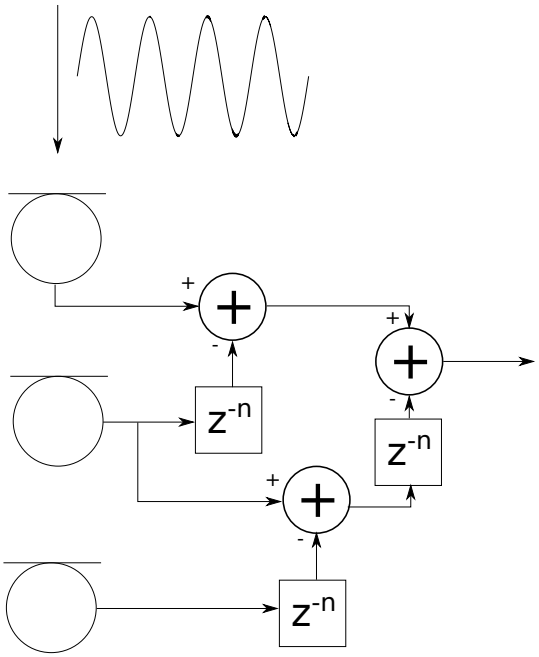


Figure 2.8: Endfire Array Configuration

The microphone arrays used in this thesis to perform beamforming are used in the broadside array configuration, as it is simple to use this configuration to create a steerable delay and sum beamformer.

2.3.2.2 Delay and Sum Beamformer

The delay and sum beamformer is one of the simplest forms of beamformer. By modifying the broadside array structure shown in Figure 2.7 to include a variable delay on each microphone line as shown in Figure 2.9, the array directivity pattern can be rotated.

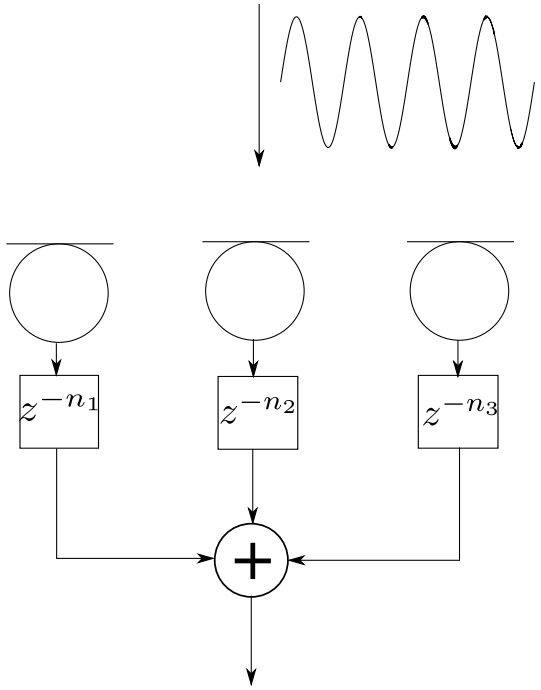


Figure 2.9: Delay and Sum Beamforming Configuration

Figure 2.10 shows the far-field directivity pattern for an example microphone array with three elements separated by a distance of 7.5cm. Note that the sources in this thesis are assumed to come from the near-field, so this Figure only serves to illustrate that the delay and sum configuration can result in a steerable directivity pattern in general. A near-field beamformer would use different delays for each microphone signal, resulting in a more complex beam-pattern. The concept of a simple delay and sum beamformer on individual microphone pairs is used in Section 2.3.2.3.

The diagram shows the response of the array to several frequencies, illustrating the frequency dependent nature of the directivity pattern. Note that the plot is in polar coordinates and was created by first considering the response of the array to a sinusoid of a defined frequency coming from an angle defined in the polar plot. The phase delay for the signal was calculated based on the TDOA of the signal to each microphone, (and the applied steering delay, zero degrees in this case). The signals from each microphone were summed, and the magnitude (of the resulting sinusoid in complex form) calculated and normalised by the number of microphones. This was then plotted in the log domain for various different frequency sinusoids.

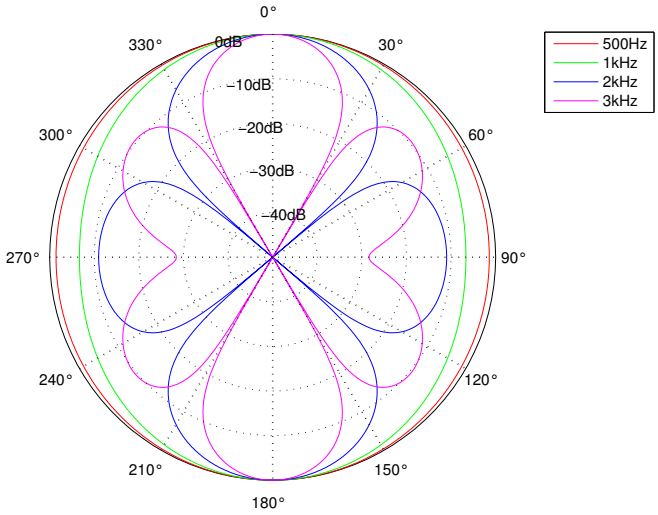


Figure 2.10: Broadside array example directivity pattern

The output $y(k)$ of a delay and sum beamformer is given in Equation (2.29), where $z_l(t)$ is the signal picked up by microphone l and τ_l is the steering time delay for microphone l, given by Equation (2.30). In this equation, the angle θ to which the main lobe of the directivity pattern should be steered can be related to the steering delay using Equation (2.18). Figure 2.11 shows the steered directivity plot for a time delay which corresponds to an angle of -30°, which was generated in the same way as Figure 2.10. The main lobe refers to the directivity pattern where the angle is such that the sensitivity is maximised over all frequencies considered.

$$y(t) = \sum_{l=1}^{L} z_l(t + \tau_l) \tag{2.29}$$

$$\tau_l = \frac{\| \mathbf{x}_m - \mathbf{p}_l \| - d_r}{c} \tag{2.30}$$

Note that the distance d_r is the distance of the steered position to a reference position, which is typically the centre of the microphone array, which allows multiple microphones to be used together by sharing a common reference. This compensates for the microphones' different positions within the array.

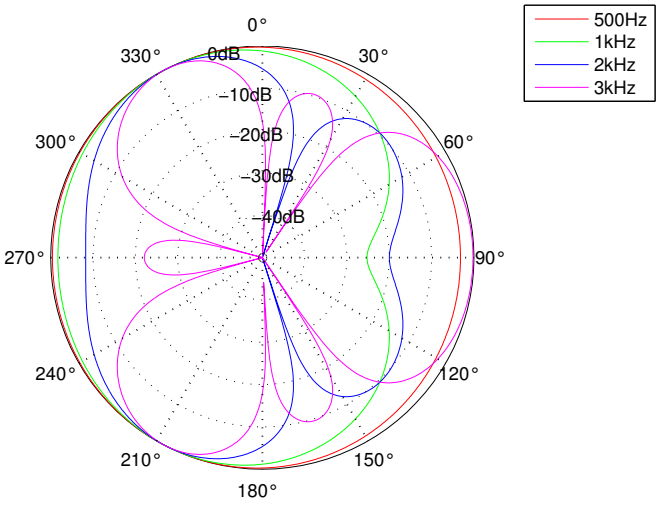


Figure 2.11: Steered array example directivity pattern

It should also be noted that there are many other beamforming methods which can be used with a broadside microphone array, for example, super-directive beamforming [46, 47] aims to maximise the directivity of the beamformer, minimising the noise coming from all but the target direction, without adversely affecting the signal from that direction.

2.3.2.3 Steered Response Power

Delay and sum beamforming can be used to perform localisation by steering the beamformer to a position $\mathbf{x}$ and calculating the acoustic energy present in the beamformer output signal. In theory, the SRP $P(\mathbf{x})$ will be at a maximum when the beamformer is steered to the true position of a source. Thus, the estimate $\hat{\mathbf{x}}$ of a single source position is formulated as shown in Equation (2.31), where $\mathbf{x}'$ is the point to which the beamforming array is steered.

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}'} P\left(\mathbf{x}'\right) \tag{2.31}$$

Of course, for multiple speakers, the SBF function will be multi-modal across space, and so multiple peaks must be found in order to locate multiple sources. The SRP is defined [13] in Equation (2.32), where n represents the n^{th} time frame, T is the length of the frame and

$\tau_{\mathbf{x}',l}$ is the time delay required to be applied to the signal from microphone l to steer the array directivity pattern in the direction of the candidate position $\mathbf{x}'$. Note that this time delay can be calculated directly from the distances between each microphone and the candidate position $\mathbf{x}'$, and is thus applicable to sources from both the near-field and the far-field.

$$P\left(\mathbf{x}'\right) = \int_{nT}^{(n+1)T} \left(\sum_{l=1}^{L} z_l \left(t - \tau_{\mathbf{x}',l} \right) \right)^2 \mathrm{d}t \tag{2.32}$$

The SRP may also be calculated by summing together the GCCs for each microphone pair in the array of size L [48]. This is expressed in the frequency domain in Equation (2.33).

$$P\left(\mathbf{x}\right) = \sum_{k=1}^{L} \sum_{l=1}^{L} \int_{-\infty}^{\infty} W_k\left(\omega\right) W_l^*\left(\omega\right) S_k\left(\omega\right) S_l^*\left(\omega\right) \mathrm{e}^{j\omega\left(\tau_{\mathbf{x},\,l} - \tau_{\mathbf{x},\,k}\right)} \mathrm{d}\omega \tag{2.33}$$

As with the GCC described in Section 2.3.1.2, the weightings $W_k\left(\omega\right)$ and $W_l^*\left(\omega\right)$ represent a pre-filtering option, and a common choice is the PHAT weighting.

2.3.2.4 Frequency Integration Range

The frequency range over which beamforming is implemented affects the spatial grid density over which the SBF can be evaluated without encountering aliasing [7]. By varying the range of integration over frequency in a steered beamformer, it was concluded by the Fallon in [7] that as the maximum frequency increased, the density of grid points also had to be increased in order to accurately find the location of the maximum point of acoustic energy.

Choosing maximum frequencies of 2000Hz and 3000Hz resulted in the localisation results being affected only negligibly when the grid density was increased from a 0.02m spacing to a 0.04m spacing. Because this was true at both frequencies, this suggested that there is no real benefit to localisation accuracy by using frequency information above 2000Hz. In turn, this suggests that there is an upper bound to the accuracy of SBF based localisers. Consideration to frequency limitation in beamforming is also given in [49], where restricting the frequency range reduced the number of spurious peaks on the beamformer output over the search area.

2.3.2.5 Localisation from Steered Response Power

Maximisation of the SRP function to estimate a source position can be achieved by evaluating the function at a number of positions. This search can be exhaustive, whereby all possible position vectors are evaluated and the maximum value can be extracted with certainty. Unfortunately, this requires a very large number of evaluations of the SRP function, and becomes less tractable as the room size and number of microphones increases. If an SRP based localiser can resolve down to 5 cubic centimetre accuracy, then an $(10 \times 10 \times 3)\,\mathrm{m}^3$ room would contain $2,400,000$ points to be evaluated. Instead, a subset of the search space can be interrogated, and a search for the maximum value carried out iteratively [50].

2.3.3 Other Localisation Methods

Localisation techniques are not limited to GCC or SBF techniques. In addition to these, there are phase unwrapping techniques [6], which make use of the cross-spectrum between a microphone pair to extract TDOAs. Adaptive Eigenvalue decomposition [51] can also extract TDOA measurements, and the difference in attenuation caused by different path lengths between each microphone and the speaker can be used to estimate the range of the speaker. This level difference information, referred to as interaural level difference (ILD), can be used to generate a circle of potential source locations, as opposed to the straight line at an angle of arrival created by TDOA techniques. Similarly though, the point of intersection of these circles in a system with multiple microphones can be found to estimate a source location. Work has also been undertaken which allows a single microphone pair to perform localisation using the intersection of the ILD circle and the TDOA line [52]. Other work fuses the measurements within a Bayesian framework [53], and several other methods have been considered, however these other localisation techniques are not the main focus of this thesis. Instead, this thesis concentrates on SBF based methods which have been shown to be robust to noise and reverberation, and which also allow the relatively easy localisation of multiple sources, as will be demonstrated.

2.4 General Methods used in Acoustic Source Localisation and Tracking

This section covers several concepts useful to many implementations of acoustic source localisation and tracking (ASLT). Firstly, source dynamics models, which model the way in which acoustic sources move, will be introduced. The application of VAD methods to ASLT will then be discussed.

2.4.1 Motion Models

When considering the problem of source tracking, account must be taken of how those sources might move within an area. This is especially true for probabilistic trackers, which predict how the state of a target evolves over time, and then corrects that prediction based on observations made. It is therefore important to have a good model for the source motion over time.

Many different source dynamics models have been developed [54], and these can be broadly split into two groups: curvilinear (CL) models and coordinate uncoupled (CU) models. CU models make use of Cartesian coordinates to represent a target state vector, typically including the state position and velocity, an example of which is given in Equation (2.34). This equation demonstrates a state which consists of the estimation of both a source's position and its velocity, however more complicated models might make use of higher order time derivatives of the source position.

$$\mathbf{s} = [\mathbf{x},\, \dot{\mathbf{x}}] \tag{2.34}$$

At a discrete time step k, the source location can be predicted from the previous source state simply as shown in Equation (2.35), where ΔT is the time period between step k and step $k-1$.

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \Delta T \dot{\mathbf{x}}_k \tag{2.35}$$

Similarly, the state of source in a CL model is expressed in polar co-ordinates, with the state update equation taking account of this.

Several promising models are investigated in [55], and the CU method of source modelling

with Langevin dynamics is popular due to its simplicity and accuracy when applied to acoustic sources. The velocity of a source can also be modelled as a random walk in both CU and CL models, along with more complex techniques which also model the source acceleration using, for example, another random walk model.

The Langevin model [11, 56] assumes that each Cartesian component is independent. For each component $\mathcal{X}$, the model for that component is given in Equations (2.36a) to (2.36d).

$$\dot{\mathcal{X}}_t = a_{\mathcal{X}} \dot{\mathcal{X}}_{t-1} + b_{\mathcal{X}} F_{\mathcal{X}} \tag{2.36a}$$

$$\mathcal{X}_t = \mathcal{X}_{t-1} + \Delta T \dot{\mathcal{X}}_t \tag{2.36b}$$

$$a_{\mathcal{X}} = \mathrm{e}^{-\beta_{\mathcal{X}} \Delta T} \tag{2.36c}$$

$$b_{\mathcal{X}} = v_{\mathcal{X}} \sqrt{1 - a_{\mathcal{X}^2}} \tag{2.36d}$$

Here, $F_{\mathcal{X}}$ is a normally distributed random variable with variance $\sigma^2_{F_{\mathcal{X}}}$, $\dot{\mathcal{X}}$ is the speed in the direction of the $\mathcal{X}$ component and $v_{\mathcal{X}}$ is the root mean square (RMS) speed in that same direction.

Some work simplifies the source movement even further [8] by considering only stationary sources whose only movement is from the influence of process noise, expressed in Equation (2.37). $\mathbf{x}_t$ is the source state at time t and $\mathbf{v}(t)$ is the process noise. This model has been found to be good enough to model sources in a recorded environment [8], although it is known that the choice of motion model can significantly change tracker performance [55].

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{v}(t) \tag{2.37}$$

With the assumption that the noise in each Cartesian direction is independent, the covariance matrix $Q(t)$ for the process noise can be written for a 3D system as shown in Equation (2.38), where σ_v^2 is the process noise power and T is the time since the last update of the source state.

$$Q(t) = \sigma_v^2 T^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.38}$$

The source dynamics model used in this thesis will mainly be the simple model of movement under Gaussian noise, which makes it relatively easy to compare algorithms such as the work by Klee [8].

2.5 Experimental Environments

This section describes the experimental environments used to provide the results for this thesis. The results are derived from both simulated environments and a real room equipped with a microphone array. Simulated environments allow experiments to be conducted on a wide variety of possible input configurations, such as scenarios involving a large and changing number of concurrent moving speakers, or a diverse set of reverberant conditions. As the ground truth is known exactly in simulations, localisation methods can be reliably tested to obtain an indication of their accuracy. They also allow the investigation of tracking methods applied to potentially difficult situations for both trackers and localisers, such as when two concurrent sources pass very close to each other.

A real environment is also useful because it allows the testing of algorithms without the modelling limitations of a simulated environment. This includes true reverberation effects and real background noise from within the room from various sources.

Several simulation environments are considered in this thesis. One such environment is that shown in Figure 2.12, which was chosen for its similarity to the setup in [13], which allowed direct comparisons to be made between the results obtained for this thesis and the work carried out in that paper. This environment was also replicated in the acoustic lab.

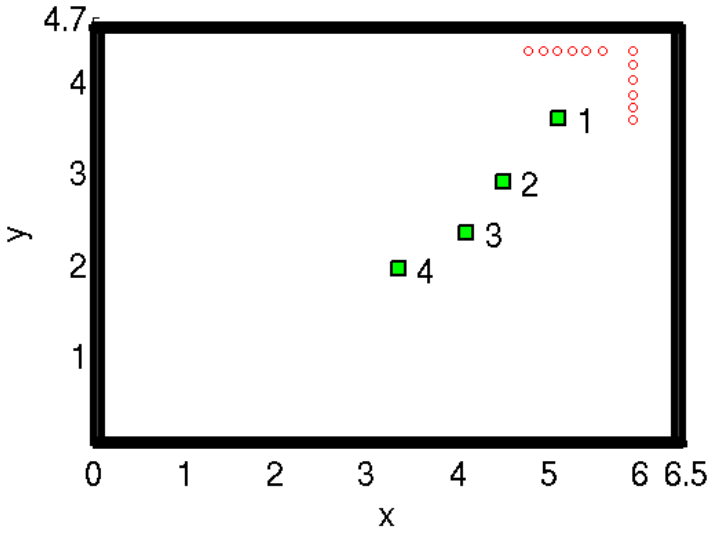


Figure 2.12: First simulated room environment, with numbered speakers marked by green squares, and microphones marked by red circles.

Figure 2.13 shows the layout of the real lab environment used in this thesis to perform experiments. This room is the same audio lab as described in [6], although the layout of the microphone array is different. As in [6], there are 16 microphones available, and the configuration of 12 of these are indicated in the diagram by microphones symbols. Exact positions shown in Figure 2.12, where the positions of the microphones and speakers in the simulated environment are set to their corresponding real-room positions. The final microphones are placed beneath the central microphone in each group of three, a 'T' arrangement. This provides the ability to discriminate the height of sources, which is not assumed to be known or constant between speakers. This is in contrast to [6], which focusses solely on the 2D localisation problem. These microphones are placed on metal T-bar stands and sampled at 96kHz. A high sampling rate was used to capture data as this was the default setup of the acoustic equipment, however this was sub-sampled to 44.1kHz, using the Audacity software package, for processing. The positions of the speakers used in the real room environment are also shown in Figure 2.12, where the top right hand side of that figure corresponds to the top right hand side of Figure 2.13.

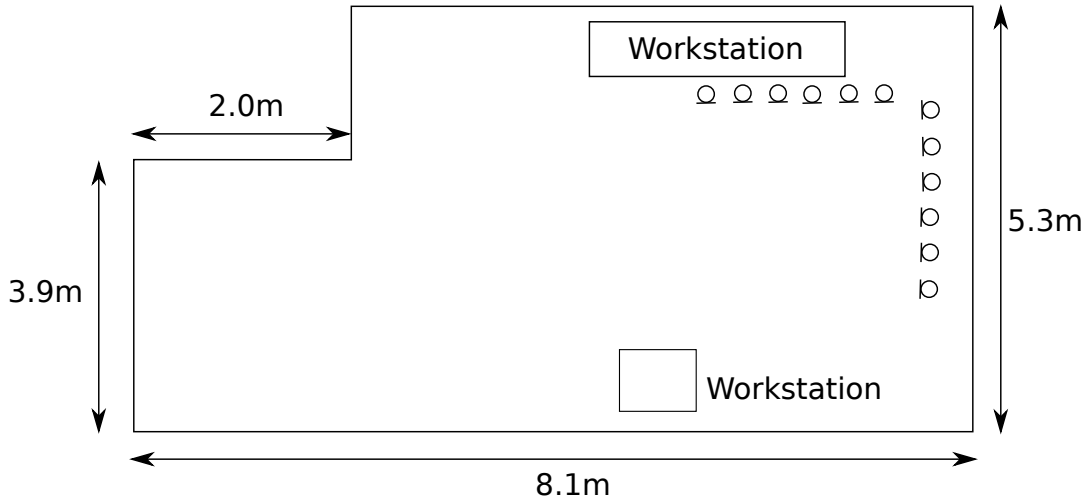


Figure 2.13: Real room environment - the Audio Processing Lab

The room is highly reverberant, with a reverberation time of 0.836s, as well as a complex room shape compared to the simple cuboid of the simulated environment. Multiple simultaneous speakers were created using recordings made using lapel microphones, to create a relatively reverberation free source of audio. These recordings were played through multiple loudspeakers placed at the various positions within the room to recreate speech sources with precisely known positions. In order to allow investigation of changing the number of speakers, the individual speakers were recorded through the microphone arrays separately, and simultaneous speech scenarios were created by summation of the corresponding microphone signals from each recording.

Another simulation environment is shown in Figure 2.14, where the width and length are indicated as being 8m and 5m respectively, and the height of the room is 3m. This relatively simple environment was chosen to represent the size of a typical office. The simulated room contains microphones placed around the edges of the room as indicated by the microphone symbols in the diagram, although some experiments made use of different array configurations. The room was a chosen to be a simple cuboid, as this is both typical of an office, and it allows reverberation to be simulated using the image method technique described in Section 2.1.6.1. The room does not contain any furniture, which is not typically simulated in the literature, however it would change the RIR in a real room environment. The value of T_{60} used for this simulation environment was 0.2s.

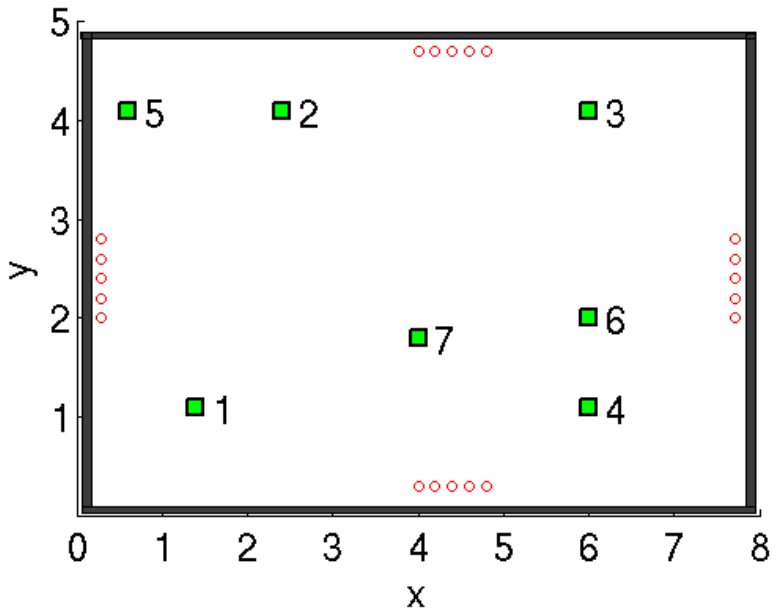


Figure 2.14: Second simulated room environment, with numbered speakers marked by green squares, and microphones marked by red circles.

2.6 Summary

This Chapter has introduced the basic framework around which the task of ASLT is based. A signal model was introduced for signals from acoustic sources. The model considers sources within a reverberant acoustic environment which affects the signal received by a microphone. Following on from this, some standard localisation methods were introduced. These make use of the acoustic model to extract both direct and indirect measurements of a source's position. Some of these methods will later be used to develop efficient single and multi-source localisation techniques.

This was followed by a brief introduction to some more general audio processing techniques. These techniques, whilst not directly localisation methods, are nonetheless useful when considering the localisation and tracking techniques used in this thesis. Finally, the environments in which experiments were carried out in this thesis were described. This introduced the distinction between simulated room environments which provide experimental flexibility, and a real acoustic lab, which allows experiments to be carried out on true non-idealised speech signals in a truly reverberant setting.

Chapter 3
Bayesian Filtering

3.1 Bayesian Estimation

This chapter introduces techniques which are fundamental to the acoustic source tracking problem, and underlie the remainder of the thesis. As such, it introduces Bayes' theorem and the recursive Bayesian estimator, which allow posterior probabilities to be calculated from *a priori* knowledge. This is followed by an introduction to the Kalman filter, the classical, optimal, closed form solution to the state tracking problem, which deals with linear system models which have Gaussian noise terms. The EKF is also discussed, which is a standard extension of the Kalman filter to deal with non-linearities in the system and observation models.

To deal with non-Gaussian noise models along with non-linear system models, the particle filter is introduced, which is a standard way to deal with these difficult yet common problems. Finally, because the aim of this thesis is to study multi-source speaker localisation, random finite sets are introduced, which can be used for multiple source Bayesian filtering.

3.1.1 Bayes Theorem

In tracking applications, there is typically an object to be tracked which has a true target state that we wish to estimate, given a set of distorted observations of that object's state. Bayes' theorem is used to relate these observations and their prior probabilities to a posterior probability of the system state given these observations. Bayes theorem is given in Equation (3.1), where the state to be estimated is given by x and the observations are represented by z.

$$p\left(x \mid z\right) = \frac{p\left(x\right) p\left(z \mid x\right)}{p\left(z\right)} \tag{3.1}$$

Here, $p\left(x \mid z\right)$ represents the posterior probability of the state x, given observations z, which is generally what we would like to know. This is calculated from the prior probability density function (PDF) of the state, $p\left(x\right)$; the likelihood of the observations given the state, $p\left(z \mid x\right)$

and finally the evidence term $p(z)$. The evidence term is the marginal likelihood of the observation over the state space X, given in Equation (3.2), and normalises the posterior probability. This term is a constant, as it is the marginal probability of the evidence term over the entire state space, i.e. does not depend on the value of x [57]. Equation (3.1) can therefore be simplified to the proportionality given in Equation (3.3), with proportionality constant α.

$$p(z) = \int_X p(x)\, p(z \mid x)\ \mathrm{d}x \tag{3.2}$$

$$p(x \mid z) \propto\ \alpha\, p(x)\, p(z \mid x) \tag{3.3}$$

3.1.2 Recursive Bayesian Estimation

Recursive Bayesian estimation attempts to calculate estimates of the unknown PDF recursively over time steps as new observations are available. The true state of a process to be estimated by recursive Bayesian estimation is assumed to be a Markov process, where the state x_t at (discrete) time t is dependent only on the previous state at time $t-1$, not on any other previous states, which is illustrated in Figure 3.1 and written probabilistically in Equation (3.4), where x_0 is the initial system state. Figure 3.1 also shows the observations available, where an observation z_t is dependent only on the state at time t. As the true state is hidden, and only distorted observations are available, this is a hidden Markov model (HMM) [58].

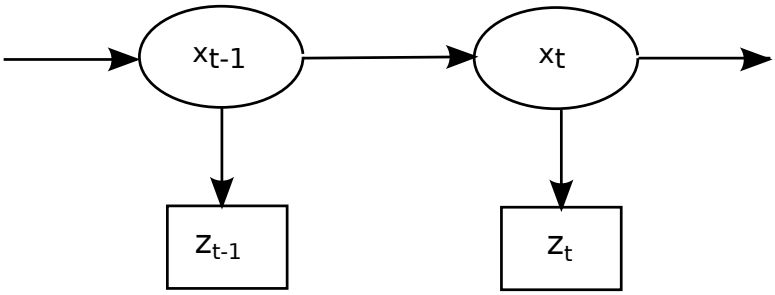


Figure 3.1: Illustration of Markov process dynamic Bayesian network, showing the dependence of states over time, as well as the observations of those states.

$$p(x_t \mid x_0, \ldots, x_{t-1}) = p(x_t \mid x_{t-1}) \tag{3.4}$$

The variables x_t and z_t are generated at each time step by some process or measurement function, respectively, defined in Equations (3.5a) and (3.5b), where v_t and w_t are independent random variables representing noise with known PDFs.

$$x_t = f_t\left(x_{t-1}, v_t\right) \tag{3.5a}$$

$$z_t = h_t\left(x_t, w_t\right) \tag{3.5b}$$

It follows that, similarly to Equation (3.4), an observation at time t is only dependent on the current state, and not any previous state, expressed in Equation (3.6).

$$p\left(z_t \mid x_0, \ldots, x_t\right) = p\left(z_t \mid x_t\right) \tag{3.6}$$

For Bayesian target tracking, the posterior distribution, $p\left(x_t \mid z_{1:t}\right)$ is required, and this can be found using Bayes' rule and the Chapman-Kolmogorov equation, given in Equation (3.7), which describes the prediction step of a recursive Bayesian estimator.

$$p\left(x_t \mid z_{1:t-1}\right) = \int p\left(x_t \mid x_{t-1}\right) p\left(x_{t-1} \mid z_{1:t-1}\right) \, \mathrm{d}x_{t-1} \tag{3.7}$$

This prediction is then updated with observations of the system using Bayes' rule, as shown in Equation (3.8), with the normalising constant defined in Equation (3.9), which depends on the likelihood $p\left(z_t \mid x_t\right)$, which is defined by the measurement function, Equation (3.5b). $p\left(x_0 \mid z_0\right)$ can be initialised as $p\left(x_0\right)$.

$$\begin{aligned} p\left(x_t \mid z_{1:t}\right) &= p\left(x_t \mid z_t, z_{1:t-1}\right) \\ &= \frac{p\left(z_t \mid x_t, z_{1:t-1}\right) p\left(x_t \mid z_{1:t-1}\right)}{p\left(z_t \mid z_{1:t-1}\right)} \\ &= \frac{p\left(z_t \mid x_t\right) p\left(x_t \mid z_{1:t-1}\right)}{p\left(z_t \mid z_{1:t-1}\right)} \end{aligned} \tag{3.8}$$

$$p\left(z_t \mid z_{1:t-1}\right) = \int p\left(z_t \mid x_t\right) p\left(x_t \mid z_{1:t-1}\right) \, \mathrm{d}x_t \tag{3.9}$$

With these recurrence relations and knowledge of the normalising constant $p(z_t \mid z_{1:t-1})$, optimal state estimates can be made based on, for example, the maximum *a posteriori* (MAP) criterion of the minimum mean-square error (MMSE) criterion. The MAP estimate is the maximisation of $p(x_t \mid z_{1:t})$, as defined in Equation (3.10), and the MMSE estimate is the conditional expectation of x_t, defined in Equation (3.11).

$$\hat{x}_{t\,|\,t}^{\mathrm{MAP}} \triangleq \arg\max_{x_t} p(x_t \mid z_{1:t}) \tag{3.10}$$

$$\hat{x}_{t\,|\,t}^{\mathrm{MMSE}} \triangleq \mathbb{E}[x_t \mid z_{1:t}] = \int x_t\, p(x_t \mid z_{1:t})\, \mathrm{d}x_t \tag{3.11}$$

3.1.3 Kalman Filter

Different estimates can be made using Bayesian inference, in particular, the posterior PDF can be maximised, as in MAP methods, or an error function can be minimised, for example with MMSE techniques. All estimates rely heavily on good *a priori* knowledge, which is assumed to be available.

Kalman filters, first introduced in 1960 [59], are used to estimate the state of a linear system from a set of noisy measurements. The filters are recursive, which means that they only need a current estimate of the state of the system and one set of measurements - no measurement history is required. The filter is optimal in the MMSE sense, but this is conditional on the linearity of the system and the noise terms are Gaussian. If the noise is not Gaussian, but second-order statistics are known, then the Kalman filter is the best linear filter [60], but non-linear filters such as particle filters can produce lower variance state estimates. This filter is therefore a constrained closed form solution of the Bayesian estimator, and is derived from the Bayesian estimator steps using the MMSE criterion.

A Kalman filter attempts to estimate an n-dimensional state at discrete time t, represented by $\mathbf{x}_t \in \Re^n$, of a process whose difference equation is given in Equation (3.12).

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_{t-1} + \mathbf{w}_{t-1} \tag{3.12}$$

The true state is observed via an m-dimensional measurement, $\mathbf{z}_t \in \Re^m$, shown in Equa-

tion (3.13).

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t \tag{3.13}$$

In these equations, both $\mathbf{w}_{t-1}$ and $\mathbf{v}_t$ are random variables representing process noise and measurement noise respectively. Importantly, these random variables are taken to be independent zero-mean white Gaussian noise, with covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ defined respectively for each variable. In Equation (3.12), the matrix $\mathbf{A}$ is the state transition matrix which relates the states between time steps. Matrix $\mathbf{B}$ converts an (optional) input to the system, $\mathbf{u}_{t-1}$, to a matrix which can be added directly with the state update. Finally, the matrix $\mathbf{H}$ converts the true state of the system to the observation vector $\mathbf{z}_t$.

The filtering algorithm can be thought of in two stages - prediction and correction. In the prediction stage, the filter attempts to estimate the next state based on the current state estimate and knowledge of system dynamics. It also estimates the next time step state estimation error covariance matrix from the current one.

During the correction stage, the initial state estimation is updated using the initial error covariance updates and the observations made of the system. Equations (3.14a) and (3.14b) describe the prediction stage, and Equations (3.14c) to (3.14e) describe the correction stage.

$$\hat{\mathbf{x}}'_t = \mathbf{A}\hat{\mathbf{x}}_{t-1} + \mathbf{B}\mathbf{u}_{t-1} \tag{3.14a}$$

$$\mathbf{P}'_t = \mathbf{A}\mathbf{P}_{t-1}\mathbf{A}^\intercal + \mathbf{Q} \tag{3.14b}$$

$$\mathbf{K}_t = \mathbf{P}'_t\mathbf{H}^\intercal\left(\mathbf{H}\mathbf{P}'_t\mathbf{H}^\intercal + \mathbf{R}\right)^{-1} \tag{3.14c}$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}'_t + \mathbf{K}_t\left(\mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}'_t\right) \tag{3.14d}$$

$$\mathbf{P}_t = \left(\mathbf{I} - \mathbf{K}_t\mathbf{H}\right)\mathbf{P}'_t \tag{3.14e}$$

Equation (3.14a) simply takes the *a priori* state estimate from the previous time step, $\hat{\mathbf{x}}_{t-1}$ and predicts an intermediate estimation of the next state estimation, $\hat{\mathbf{x}}'_t$. Similarly, Equation (3.14b) creates an intermediate prediction for the next state estimation error covariance, $\mathbf{P}'_t$.

The correction stage begins with Equation (3.14c) calculating $\mathbf{K}_t$, known as the Kalman gain.

This gain is then used with the observation vector in Equations (3.14d) and (3.14e) to determine the *a posteriori* state estimate and estimate error covariance matrix. This process of prediction and correction can be repeated recursively over multiple time steps.

3.1.4 Extended Kalman Filter

The Kalman filter assumes that the system being filtered is linear with Gaussian noise. These conditions are often not satisfied and the EKF is one way to deal with the case of Gaussian noise with a non-linear state transition function and observation function. The idea behind the EKF is that if these functions are known, then they can be approximated by a linear function (linearised) around some point - namely the current state estimate - at each time step, allowing the Kalman filter algorithm to continue in an otherwise standard way.

The Kalman filter was based on Equation (3.12), the state difference function, as well as Equation (3.13), the observation function, being a linear. These can be redefined as in Equations (3.15) and (3.16) as general non-linear functions with, as before, independent white Gaussian noise variables. Unfortunately, the distributions of the noise terms are no longer Gaussian as they are transformed by the non-linear functions, which means that the EKF is not an optimal estimator [61].

$$\mathbf{x}_t = f\left(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, \mathbf{w}_{t-1}\right) \tag{3.15}$$

$$\mathbf{z}_t = h\left(\mathbf{x}_t, \mathbf{v}_t\right) \tag{3.16}$$

The state and the measurements can be approximated as in Equations (3.17) and (3.18), which neglect the noise variables. In Equation (3.17), $\hat{\mathbf{x}}_{t-1}$ is an *a posteriori* estimate of the system state from the previous time step.

$$\tilde{\mathbf{x}}_t = f\left(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}, 0\right) \tag{3.17}$$

$$\tilde{\mathbf{z}}_t = h\left(\tilde{\mathbf{x}}_t, 0\right) \tag{3.18}$$

The system state and observations can be linearised as shown in Equations (3.19) and (3.20), which are truncated Taylor series expansions around the approximations in Equations (3.17) and (3.18). Here, the notation is as before, with additional Jacobian matrices, $\mathbf{A}_t$, $\mathbf{W}_t$, $\mathbf{H}_t$ and $\mathbf{V}_t$. Note that $\mathbf{A}_t$ is the Jacobian matrix of partial derivatives of $f\left(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, 0\right)$ with respect to $\mathbf{x}_t$; $\mathbf{W}_t$ is the Jacobian matrix of partial derivatives of $f\left(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}, 0\right)$ with respect to $\mathbf{w}_t$; $\mathbf{H}_t$ is the Jacobian matrix of partial derivatives of $h\left(\tilde{\mathbf{x}}_t, 0\right)$ with respect to $\mathbf{x}_t$, and $\mathbf{V}_t$ is the Jacobian matrix of partial derivatives of $h\left(\tilde{\mathbf{x}}_t, 0\right)$ with respect to $\mathbf{v}_t$.

$$\mathbf{x}_t \approx \tilde{\mathbf{x}}_t + \mathbf{A}_t\left(\mathbf{x}_{t-1} - \hat{\mathbf{x}}_{t-1}\right) + \mathbf{W}_t\mathbf{w}_{t-1} \tag{3.19}$$

$$\mathbf{z}_t \approx \tilde{\mathbf{z}}_t + \mathbf{H}_t\left(\mathbf{x}_t - \tilde{\mathbf{x}}_t\right) + \mathbf{V}_t\mathbf{v}_t \tag{3.20}$$

The EKF algorithm structure is the same as the Kalman filter structure, that is, there is a state prediction phase and an update phase based on observations of the system being filtered. Equation (3.21a) shows that the state estimate is simply calculated using the non-linear state transition function with no noise term.

$$\hat{\mathbf{x}}'_t = f\left(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_{t-1}, 0\right) \tag{3.21a}$$

$$\mathbf{P}'_t = \mathbf{A}_t\mathbf{P}_{t-1}\mathbf{A}_t^\intercal + \mathbf{W}_t\mathbf{Q}_{t-1}\mathbf{W}_t^\intercal \tag{3.21b}$$

$$\mathbf{K}_t = \mathbf{P}'_t\mathbf{H}_t^\intercal\left(\mathbf{H}_t\mathbf{P}'_t\mathbf{H}_t^\intercal + \mathbf{V}_t\mathbf{R}_t\mathbf{V}_t^\intercal\right)^{-1} \tag{3.21c}$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}'_t + \mathbf{K}_t\left(\mathbf{z}_t - h\left(\hat{\mathbf{x}}'_t, 0\right)\right) \tag{3.21d}$$

$$\mathbf{P}_t = \left(\mathbf{I} - \mathbf{K}_t\mathbf{H}_t\right)\mathbf{P}'_t \tag{3.21e}$$

Equation (3.21b) changes the state estimate error covariance update to include the Jacobian matrices, ands similarly, the update equations are modified to take account of the system linearisation and also make use of the defined Jacobian matrices.

3.1.5 Iterated Extended Kalman Filter

The EKF can be refined using local iterations of a linearised update step [62] to attempt to better estimate the state of systems with highly non-linear measurement functions. The filter algorithm is the same as the EKF, but with Equation (3.21d) replaced with a linearised iteration function given by Equation (3.22). This iteration function is a re-linearisation around the updated state estimate, with η_0 set to $\hat{\mathbf{x}}'_t$, meaning that with only one iteration, the iterated extended Kalman filter (IEKF) reduces to the EKF.

$$\eta_l = \hat{\mathbf{x}}'_t + \mathbf{K}_{t,\eta_{l-1}} \left(\mathbf{z}_t - h\left(\eta_{l-1}, 0\right) - \mathbf{H}_{t,\eta_{l-1}} \left(\hat{\mathbf{x}}'_t - \eta_{l-1} \right) \right) \tag{3.22}$$

The iterations stop at iteration l after the difference between iteration result η_l and η_{l-1} is small, that is, less than some tolerance value. The final state estimation for the Kalman filter iteration t is then set to η_l and $\mathbf{P}_t$ is calculated again. For each iteration, the Kalman gain is recalculated, as is the observation function $h\left(\eta_{l-1}, 0\right)$ and the associated Jacobian matrix $\mathbf{H}_{t,\eta_{l-1}}$.

3.1.6 Numerical Stability

Kalman filters are known to be numerically unstable [63]. This can be demonstrated by considering Equation (3.14b), also known as the Ricatti equation. If this is propagated by one time step, as in Equation (3.23), Equation (3.14e) can be substituted in to give Equation (3.24).

$$\mathbf{P}'_{t+1} = \mathbf{A}\mathbf{P}_t\mathbf{A}^\intercal + \mathbf{Q} \tag{3.23}$$

$$\mathbf{P}'_{t+1} = \mathbf{A}\mathbf{P}'_t\mathbf{A}^\intercal - \mathbf{A}\mathbf{K}_t\mathbf{H}\mathbf{P}'_t\mathbf{A}^\intercal + \mathbf{Q} \tag{3.24}$$

The Kalman gain, Equation (3.14c), can be re-written as shown in Equation (3.25), and then Equation (3.24) can be substituted in as shown in Equation (3.26).

$$\left(\mathbf{H}\mathbf{P}'_t\mathbf{H}^\intercal + \mathbf{R}\right)\mathbf{K}_t^\intercal = \mathbf{H}\mathbf{P}'_t \tag{3.25}$$

$$\mathbf{P}'_{t+1} = \mathbf{A}\mathbf{P}_t\mathbf{A}^\intercal - \mathbf{A}\mathbf{K}_t\left(\mathbf{H}\mathbf{P}'_t\mathbf{H}^\intercal + \mathbf{R}\right)\mathbf{K}_t^\intercal\mathbf{A}^\intercal + \mathbf{Q} \tag{3.26}$$

Both $\mathbf{P}'_{t+1}$ and $(\mathbf{H}\mathbf{P}'_t\mathbf{H}^\intercal + \mathbf{R})$ are covariance matrices, and so they must be positive-definite. If $\mathbf{A}$ and $\mathbf{K}_t$ are both full-rank matrices, then Equation (3.26) can be thought of as the difference of two positive-definite matrices. Because these calculations are performed by computers, which have finite accuracy when it comes to representing numbers, the computed result of Equation (3.26) ($\mathbf{P}'_{t+1}$) can become an indefinite matrix, although it should in theory remain positive-definite.

This problem can cause the Kalman filter state estimate to diverge, which is a problem for practical implementations of the filter. A common way to deal with this is to update the square-root of $\mathbf{P}_{t+1}$ at each time step, which can be found using the Cholesky decomposition. Equation (3.27) then redefines the covariance matrix of the predicted state error, $\mathbf{K}_t$. Whilst this method resolves the divergence problem and moreover, effectively increases the numerical precision of the result [64], there is a computational cost due to the square-root computation and other methods have been developed with the aim of reducing this cost [65].

$$\mathbf{P}_{t+1} \triangleq \mathbf{P}_{t+1}^{1/2}\mathbf{P}_{t+1}^{\intercal/2} \tag{3.27}$$

3.2 Bayesian Filtering for Multiple Sources

Multi-target acoustic source tracking is concerned with updating the states of multiple sources, which is complicated by the appearance and disappearance of sources as they start and stop talking. Further, tracking systems have to deal with spurious measurements not originating from any source of interest, as well as missed measurements, where a real speaker isn't detected correctly. Thus, multi-target trackers must estimate both the number of targets at a given time step, as well as their states. Figure 3.2 illustrates this problem.

3.2.1 Random Finite Sets

The RFS formulation is justified [66] by considering the estimation error of a tracking system. If the multi-target state is represented by an array of individual target state vectors $\mathbf{X}$, as shown

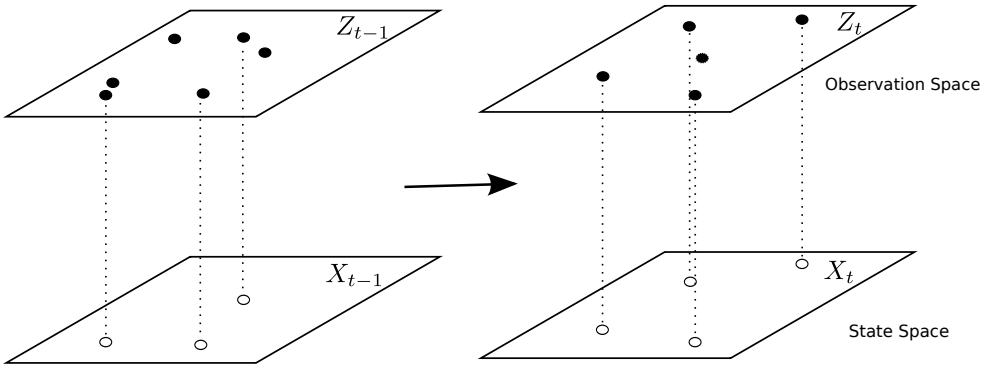


Figure 3.2: System model for multi-target tracking, showing spurious and missed measurements as well as a change in the number of speakers

in Equation (3.28), then the estimation error is hard to define, partly because the order in which to stack them is unknown. In Equation (3.28), n target states are arranged into a matrix with n rows, each row being one of the target states. Each target state is a row vector with m components, i.e. $\mathbf{x}_n = [x_{n,1} \dots x_{n,m}]$.

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} \tag{3.28}$$

Figure 3.3 illustrates such a problem, where the state of two targets is defined by the stacking of their Cartesian coordinates. Despite the estimated state $\mathbf{X'}$ having the correct numbers, they are in the wrong order, leading to an estimation error of $\|\mathbf{X} - \mathbf{X'}\|^2 = 2$. This could be dealt with by defining the estimation error as the minimum error over all stacking permutations of the vector, but there are still problems.

Importantly, using a stacked vector representation causes problems with calculating the error when the cases illustrated in Figure 3.4 are considered: firstly when the estimated number of targets is different from the true number of targets; and secondly when there are no real targets at all. The same thing can be argued for measurements data - the number of measurements at a given time can change, and their order is insignificant.

A RFS is specified by a discrete distribution which characterises the number of points in the set (its cardinality) and a set of joint distributions which specify the locations of the points

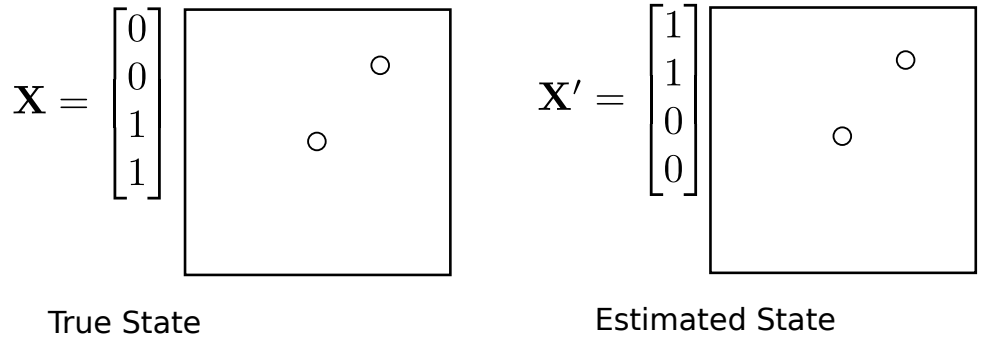


Figure 3.3: Multi-target states represented by a stacked vector of Cartesian coordinates

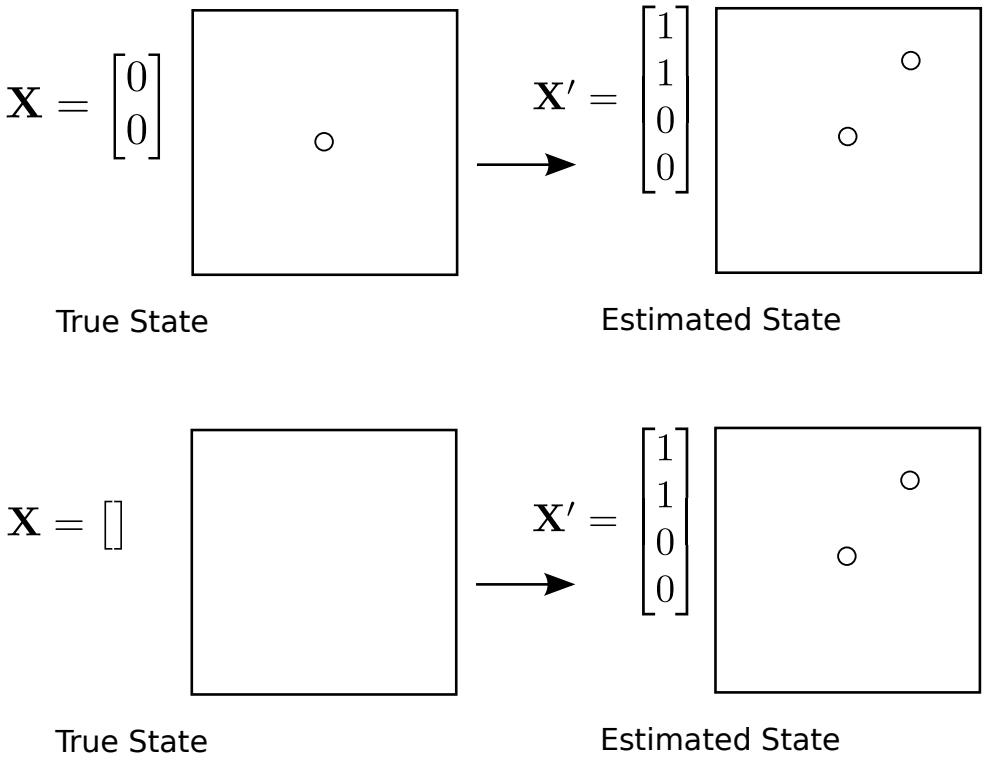


Figure 3.4: Estimated number of states differing from the true number of states using a stacked vector representation

conditional on that cardinality. In RFS based trackers, the multiple individual states of each source are treated as a set-valued single state to be tracked [19]. The RFS formulation for multi-target filtering is set up in [19] as follows.

The number of targets at time t is defined by $M(t)$, and at time $t-1$, the individual states of the targets are $x_{1,t-1},\, \ldots,\, x_{M(t-1),t-1}$. At time t, the sensor array gives $N(t)$ observations $z_{1,t-1},\, \ldots,\, z_{N(t-1),t-1}$, whose order is unknown because the true source of each measurement is unknown, as mentioned previously. Some of these measurements will be spurious - not corresponding to any true source; measurements corresponding to a true source may be missing, as they simply haven't been observed. Equations (3.29) and (3.30) represent the target states and measurements at time t as finite sets, where $\mathcal{F}(\mathcal{X})$ and $\mathcal{F}(\mathcal{Z})$ are the power sets of the state space $\mathcal{X}$ and the observation space $\mathcal{Z}$ respectively. Note that the power set $\mathcal{P}(S)$ of a set S is defined as the set of all possible subsets of S, including S and the empty set, $\emptyset$.

$$X_t = \left\{x_{1,t},\, \ldots,\, x_{M(t),t}\right\} \in \mathcal{F}(\mathcal{X}) \tag{3.29}$$

$$Z_t = \left\{z_{1,t},\, \ldots,\, z_{N(t),t}\right\} \in \mathcal{F}(\mathcal{Z}) \tag{3.30}$$

Moving from time step $t-1$ to t, each state x_{t-1} in X_{t-1} continues to exist with some probability, $p_{S,t}(x_{t-1})$, which leads to the source 'death' probability of $1 - p_{S,t}(x_{t-1})$. Each state also has a transition function, $f_{t\,|\,t-1}(x_{t-1})$. Together, these provide the state behaviour at time t from time $t-1$ as the RFS $S_{t\,|\,t-1}(x_{t-1})$, which can be either $\{x_t\}$ or $\emptyset$. The multi-target state X_t at time t is expressed in terms of the previous multi-target state X_t by Equation (3.31), which is a union of the RFS of surviving targets, the RFS of spontaneous target births Γ_t, and the RFS of new targets, referred to as 'spawned' targets, $B_{t\,|\,t-1}(\zeta)$.

$$X_t = \left[\bigcup_{\zeta=X_{t-1}} S_{t\,|\,t-1}(\zeta)\right] \cup \left[\bigcup_{\zeta=X_{t-1}} B_{t\,|\,t-1}(\zeta)\right] \cup \Gamma_t \tag{3.31}$$

The RFS measurement model is then defined by considering the probability of a single state, x_t, being detected, $p_{D,t}(x_t)$. The PDF of obtaining a measurement z_t from state x_t, conditional on being detected, is defined by $h_t(z_t \mid x_t)$. Therefore, each state at each time step gener-

ates an RFS given by Equation (3.32). Note that explicit expressions for both $h_t\left(z_t \mid x_t\right)$ and $f_{t\,|\,t-1}\left(X_t \mid X_{t-1}\right)$ must be derived from the dynamics and sensing models of the system being filtered using finite set statistics (FISST).

$$\Theta_t\left(x_t\right) = \begin{cases} \{z_t\} & \text{if state detected} \\ \emptyset, & \text{otherwise} \end{cases} \tag{3.32}$$

Similar to $S_{t\,|\,t-1}$, this state is either $\{z_t\}$ if the target is detected, or $\emptyset$ if not. The sensors also receive a set of false measurements, often referred to as clutter, represented by K_t. This leads to the representation of the multi-target measurement given in Equation (3.33), which is the union of the set of clutter observations and the set of observations generated by targets.

$$Z_t = K_t \cup \left[\bigcup_{x \in X_t} \Theta_t\left(x\right)\right] \tag{3.33}$$

This leads to a multi-source recursive Bayesian filter, with prediction and update steps expressed in Equations (3.34) and (3.35) respectively.

$$p_{\,t\,|\,t-1}\left(X_t \mid Z_{1:t-1}\right) = \int f_{t\,|\,t-1}\left(X_t \mid X\right) p_{t-1}\left(X \mid Z_{1:t-1}\right) \mu_s \,\mathrm{d}X \tag{3.34}$$

$$p_t\left(X_t \mid Z_{1:t}\right) = \frac{h_t\left(Z_t \mid X_t\right) p_{\,t\,|\,t-1}\left(X_t \mid Z_{1:t-1}\right)}{p_t\left(Z_t \mid Z_{1:t-1}\right)} \tag{3.35}$$

In these equations, μ_s is a reference measure [67] on the power set $\mathcal{F}\left(\mathcal{X}\right)$ and $p_t\left(Z_t \mid Z_{1:t-1}\right)$ is the Bayes normalisation factor expressed in Equation (3.36).

$$p_t\left(Z_t \mid Z_{1:t-1}\right) = \int h_t\left(Z_t \mid X\right) p_{\,t\,|\,t-1}\left(X \mid Z_{1:t-1}\right) \mu_s \,\mathrm{d}X \tag{3.36}$$

The recursion given by this set of equations calls for computationally intractable set integrations. Sequential Monte Carlo (SMC) methods to approximate these integrals have been studied [68], however the computational cost is still high.

3.2.2 Probability Hypothesis Density Filter

The probability hypothesis density (PHD) filter attempts to reduce the complexity of the multi-target tracking problem by propagating through time steps a first-order statistical moment of the multi-target posterior state, rather than multi-target posterior density. This parallels the Kalman filter, which propagates the expectation of a single target state.

The intensity, or PHD, of an RFS X with probability distribution P is the first-order moment of X, and is denoted as the function $v\left(x\right)$ in Equation (3.37). In this equation, $\mathcal{X}$ is the entire set state space, and S is a region in $\mathcal{X}$, $S \subseteq \mathcal{X}$.

$$\int |X \cap S| = \int_{S} v\left(x\right) \, \mathrm{d}x \tag{3.37}$$

The interpretation of this is that the integration of the PHD gives the expected number of elements of the set X - the expected number of targets.

Poisson RFSs are described completely by their PHDs, and can be used to model birth and clutter RFSs, for example those represented by Γ_t, $B_{t\,|\,t-1}$ and K_t given in Section 3.2.1. A Poisson RFS is one which has a Poisson cardinality distribution of X, with some mean $\hat{N}$, and for a finite cardinality, the elements of X are i.i.d.

3.2.3 Linear Gaussian Probability Hypothesis Density Recursion

Similar to the standard Bayes recursion, the PHD recursion requires extra assumptions about the target model in order to obtain a closed form solution akin to the Kalman filter. The linear Gaussian assumption for individual target states must still hold for a closed form solution. Further, it is assumed that the RFS predicted using $p_{t\,|\,t-1}$ is a Poisson RFS; the survival and detection probabilities defined previously are independent, expressed in Equation (3.38), and finally that the birth intensity $\gamma_t(x)$ and the spawn intensity $\beta_{t\,|\,t-1}(x \,|\, \zeta)$ are Gaussian mixtures as described in Equations (3.39) and (3.40) respectively.

$$\begin{aligned} p_{S,t}\left(x\right) &= p_{S,t} \\ p_{D,t}\left(x\right) &= p_{D,t} \end{aligned} \tag{3.38}$$

$$\gamma_t(x) = \sum_{i=1}^{J_{\gamma,t}} w_{\gamma,t}^{(i)} \mathcal{N}\left(x; m_{\gamma,t}^{(i)}, P_{\gamma,t}^{(i)}\right) \tag{3.39}$$

$$\beta_{t\,|\,t-1}(x\,|\,\zeta) = \sum_{j=1}^{J_{\beta,t}} w_{\beta,t}^{(j)} \mathcal{N}\left(x; F_{\beta,t-1}^{(j)}\zeta + d_{\beta,t-1}^{(j)}, Q_{(j)}^{\beta,t-1}\right) \tag{3.40}$$

The birth parameters are similar to the standard Gaussian mixture model (GMM) parameters in that the system is a sum of $J_{\gamma,t}$ Gaussian distributions, weighted with weights $w_{\gamma,t}^{(i)}$ and each distribution with mean $m_{\gamma,t}^{(i)}$ and covariance matrix $P_{\gamma,t}^{(i)}$. Similarly, the spawn parameters represent a sum of $J_{\beta,t}$ Gaussians, each weighted by $w_{\beta,t}^{(j)}$, with mean $F_{\beta,t-1}^{(j)}\zeta + d_{\beta,t-1}^{(j)}$ and covariance matrix $Q_{(j)}^{\beta,t-1}$. It can be shown [19] that the propagation of these Gaussian mixtures through the PHD results in Gaussian mixtures at the output, which allows the recursion to be meaningfully repeated.

The final GM-PHD recursion is given in Equations (3.44) and (3.46), where Equation (3.44) corresponds to the Bayesian prediction step and Equation (3.46) corresponds to the update step. In these equations, the linear Gaussian state transition model and measurement model are represented by Equations (3.41) and (3.42), with a state transition matrix $F_{t-1}\zeta$; process noise covariance matrix Q_{t-1}; measurement matrix H_t and measurement noise covariance matrix R_t.

$$f_{t\,|\,t-1}\left(x\,|\,\zeta\right) = \mathcal{N}\left(x; F_{t-1}\zeta, Q_{t-1}\right) \tag{3.41}$$

$$h_t\left(z\,|\,x\right) = \mathcal{N}\left(z; H_t x, R_t\right) \tag{3.42}$$

Equation (3.44) assumes that the posterior intensity at time step $t-1$ is a Gaussian mixture of the form given in Equation (3.43). In this Equation, $w_{t-1}^{(i)}$ is the mixture weight for the i^{th} of J_{t-1} mixture components in the model, where J_{t-1} represent the number of tracked sources at time $t-1$. Similarly, in Equation (3.44), $J_{\beta,t}$ represents the number of spawned targets at a time step, each with weight $w_{\beta,t}^{(l)}$.

$$v_{t-1}\left(x\right) = \sum_{i=1}^{J_{t-1}} w_{t-1}^{(i)} \mathcal{N}\left(x; m_{t-1}^{(i)}, P_{t-1}^{(i)}\right) \tag{3.43}$$

$$v_{t\,|\,t-1}(x) = v_{S,t\,|\,t-1}(x) + v_{\beta,t\,|\,t-1}(x) + \gamma_t(x) \tag{3.44a}$$

$$v_{S,t\,|\,t-1}(x) = p_{S,t} \sum_{j=1}^{J_{t-1}} w_{t-1}^{(j)} \mathcal{N}\left(x; m_{S,t\,|\,t-1}^{(j)}, P_{S,t\,|\,t-1}^{(j)}\right) \tag{3.44b}$$

$$m_{S,t\,|\,t-1}^{(j)} = F_{t-1} m_{t-1}^{(j)} \tag{3.44c}$$

$$P_{S,t\,|\,t-1}^{(j)} = Q_{t-1} + F_{t-1} P_{t-1}^{(j)} F_{t-1}^{\mathsf{T}} \tag{3.44d}$$

$$v_{\beta,t\,|\,t-1}(x) = \sum_{j=1}^{J_{t-1}} \sum_{l=1}^{J_{\beta,t}} w_{t-1}^{(j)} w_{\beta,t}^{(l)} \mathcal{N}\left(x; m_{\beta,t\,|\,t-1}^{(j,l)}, P_{\beta,t\,|\,t-1}^{(j,l)}\right) \tag{3.44e}$$

$$m_{\beta,t\,|\,t-1}^{(j,l)} = F_{\beta,t-1}^{(l)} m_{t-1}^{(j)} + d_{\beta,t-1}^{(l)} \tag{3.44f}$$

$$P_{\beta,t\,|\,t-1}^{(j,l)} = Q_{\beta,t-1}^{(l)} + F_{\beta,t-1}^{(l)} P_{\beta,t-1}^{(j)} \left(F_{\beta,t-1}^{(l)}\right)^{\mathsf{T}} \tag{3.44g}$$

Similar to Equation (3.43), the predicted intensity at time t is assumed to be a Gaussian mixture of the form given in Equation (3.45), where the variables are as defined previously, along with $\kappa_t(z)$, which represents the intensity of K_t, the clutter RFS at time step t.

$$v_{t\,|\,t-1}(x) = \sum_{i=1}^{J_{t\,|\,t-1}} w_{t\,|\,t-1}^{(i)} \mathcal{N}\left(x; m_{t\,|\,t-1}^{(i)}, P_{t\,|\,t-1}^{(i)}\right) \tag{3.45}$$

$$v_t(x) = (1 - p_{D,t})\, v_{t\,|\,t-1}(x) + \sum_{z \in Z_t} v_{D,t}(x; z) \tag{3.46a}$$

$$v_{D,t}(x; z) = \sum_{j=1}^{J_{t\,|\,t-1}} w_t^{(j)} \mathcal{N}\left(x; m_{t\,|\,t}^{(j)}, P_{t\,|\,t}^{(j)}\right) \tag{3.46b}$$

$$w_t^{(j)}(x) = \frac{p_{D,t} w_{t\,|\,t-1}^{(j)} q_t^{(j)}(z)}{\kappa_t(z) + p_{D,t} \sum_{l=1}^{J_{t\,|\,t-1}} w_{t\,|\,t-1}^{(l)} q_t^{(l)}(z)} \tag{3.46c}$$

$$q_t^{(j)}(z) = \mathcal{N}\left(z; H_t m_{t\,|\,t-1}^{(j)}, R_t + H_t P_{t\,|\,t-1}^{(j)} H_t^{\mathsf{T}}\right) \tag{3.46d}$$

$$m_{t\,|\,t}^{(j)}(z) = m_{t\,|\,t-1}^{(j)} + K_t^{(j)} \left(z - H_t m_{t\,|\,t-1}^{(j)}\right) \tag{3.46e}$$

$$P_{t\,|\,t}^{(j)} = \left[I - K_t^{(j)} H_t\right] P_{t\,|\,t-1}^{(j)} \tag{3.46f}$$

$$K_t^{(j)} = P_{t\,|\,t-1}^{(j)} H_t^{\intercal} \left(H_t P_{t\,|\,t-1}^{(j)} H_t^{\intercal} + R_t \right)^{-1} \tag{3.46g}$$

Finally, Equations (3.47) and (3.48) give the expectation of the predicted number of targets and the expectation of the number of targets after the update step respectively.

$$\hat{N}_{t\,|\,t-1} = \hat{N}_{t-1} \left(p_{S,k} + \sum_{j=1}^{J_{\beta,t}} w_{\beta,t}^{(j)} \right) + \sum_{j=1}^{J_{\gamma,t}^{(j)}} w_{\gamma,t}^{(j)} \tag{3.47}$$

$$\hat{N}_t = \hat{N}_{t\,|\,t-1} \left(1 - p_{D,t} \right) + \sum_{z \in Z_t} \sum_{j=1}^{J_{t\,|\,t-1}} w_t^{(j)} \left(z \right) \tag{3.48}$$

As time progresses, the number of Gaussians in the mixture model increases so far as to become computationally hard to deal with. To remedy this, the number of Gaussians propagated from step $t-1$ to step t can be reduced in a process called pruning. This process takes into account a maximum allowable number of Gaussian terms, $J_{\max}$; a threshold T for Gaussians with weights under which should be discarded and a threshold U, which allows similar Gaussian components to be grouped together and approximated with a single Gaussian component. Algorithm 1 [19] presents the pruning algorithm, given the sets $\left\{ w_t^{(i)}, m_t^{(i)}, P_t^{(i)} \right\}_{i=1}^{J_t}$.

Finally, state extraction is slightly complicated by the fact that the height of each Gaussian peak is dependent not only on its covariance, but on its weight as well. Choosing the $\hat{N}_t$ highest peaks as the state estimates might pick Gaussian components with low weights. It is suggested that when selecting the components, only those components with a weight over a threshold E be used.

3.2.3.1 Track Continuity

Whilst the GM-PHD filter can extract multiple tracked states at each time frame, Gaussians corresponding to a single source across time frames aren't explicitly associated with each other. To remedy this, a label $l_t^{(i)}$ can be applied to each Gaussian component. At the prediction step, surviving Gaussians keep their labels whilst spawned and birthed components are each assigned a new label. At the updated and pruning stage, components which should have the same label as in the prediction step need to be identified, as well as deciding which label to keep when

$l = 0$
$I = \left\{ i = 1, \ldots, J_t \mid w_t^{(i)} > T \right\}$
repeat
  $l = l + 1$
  $j = \arg\max_{i \in I} w_t^{(i)}$
  $L = \left\{ i \in I \mid \left(m_t^{(i)} - m_t^{(j)} \right)^{\intercal} \left(P_t^{(i)} \right)^{-1} \left(m_t^{(i)} - m_t^{(j)} \right) \leqslant U \right\}$
  $\tilde{w}_t^{(l)} = \sum_{i \in L} w_t^{(l)}$
  $\tilde{m}_t^{(l)} = \frac{1}{\tilde{w}_t^{(l)}} \sum_{i \in L} w_t^{(i)} x_t^{(i)}$
  $\tilde{P}_t^{(l)} = \frac{1}{\tilde{w}_t^{(l)}} \sum_{i \in L} w_t^{(i)} \left(P_t^{(i)} + \left(\tilde{m}_t^{(l)} - m_t^{(i)} \right) \left(\tilde{m}_t^{(l)} - m_t^{(i)} \right)^{\intercal} \right)$
  $I = I \setminus L$
until $I = \emptyset$
if $l > J_{\max}$ **then**
  Replace $\left\{ w_t^{(i)}, m_t^{(i)}, P_t^{(i)} \right\}_{i=1}^{l}$ by the $J_{\max}$ Gaussians with the largest weights
end if
return $\left\{ w_t^{(i)}, m_t^{(i)}, P_t^{(i)} \right\}_{i=1}^{l}$

Algorithm 1: GM-PHD Gaussian pruning algorithm

components are merged [69].

3.3 Summary

This chapter introduced a Bayesian framework for filtering and tracking which is the basis for much work in the area of acoustic source localisation. Bayesian estimation was presented, followed by the Kalman filter, which is a closed-form solution derived from Bayesian estimation for linear systems with Gaussian noise. The EKF was then presented, which is an extension of the Kalman filter to non-linear systems, along with the IEKF, a similar technique.

These techniques are important for the tracking of speakers in the single-source scenario, and are used in the literature as well as the early chapters of this thesis for that purpose. However, they cannot deal with tracking multiple simultaneous speakers, which is one of the considerations of this thesis.

For that purpose, RFSs were introduced, which are the basis for multi-target tracking PHD filter techniques. The GM-PHD was introduced, which provides a powerful tool for tracking the states of multiple speakers. PHD filters replace multiple target states with a single, RFS based multi-target state. The Bayesian recursion can be defined for this set-state, leading to

practical filters. The GM-PHD filter is one such filter which has a closed form given some assumptions, and avoids computationally expensive sequential Monte Carlo (SMC) methods.

Chapter 4
Single Source Audio Localisation

This chapter considers the relatively simple case of acoustic source localisation and tracking for a single speaker at a time. This specific case of the ASLT problem is considered first as it avoids the problems inherent when considering an unknown number of concurrent speakers. By assuming that only a single continuous source is active, all location information extracted can be considered as coming from that source, neglecting reverberation and other noise.

The work in this chapter builds on work which utilises the SRP, which is a useful measure of the acoustic power from a point in space within a room. The SRP is attractive for source localisation because it is robust to both noise and reverberation, but also because it has the potential to be used for the localisation of multiple sources. This is because individual sources create distinct peaks of the SRP function corresponding to their position in space. The localisation of multiple peaks of this function will be considered in Chapter 6.

The localisation technique studied in this Chapter, known as stochastic region contraction (SRC), is a method of evaluating the SRP iteratively, to close in on the location of a source. This technique will be described, and this will be followed by the introduction of a new technique, the HE-SRC method. This technique was developed to more efficiently use the SRP for source localisation by feeding the height of a source detected in previous time steps back in to the localisation algorithm, in order to improve the localisation speed. Height information might not be considered particularly useful for a speaker *tracker*, as speakers can be assumed never to occupy the same position on the horizontal-plane. Nevertheless, it will be shown that the use of height information in the *localisation* algorithm can improve tracker results. Some work assumes that speaker heights are known to be within some narrow range of the vertical axis. This chapter makes use of height information to speed up the two dimensional localisation task, whilst simultaneously allowing the entire vertical axis to be explored.

Finally, it should be remarked that the HE-SRC method is fully developed to be able to take advantage of potential speaker height information from other sources, such as from a video processing algorithm. If a set of cameras can be used to locate the positions of people within a

room, then their heights can be extracted to allow the current speaker to be identified relatively quickly. This might be particularly useful in scenarios where the speakers are not situated in the vertical axis where some algorithms assume them to be, that is, distributed over some implicitly defined region of the vertical axis. This is acceptable if all of the people within an area are around the same height, and standing up. However, this assumption might be violated in scenario where, for example, somebody is standing up and giving a presentation to a seated audience within a meeting room. In such a scenario, knowing the positions of potential speakers - and therefore their heights - would allow an audio localisation algorithm to quickly identify speaker locations as and when the active speaker changed, for example when questions were asked of and answered by the presenter.

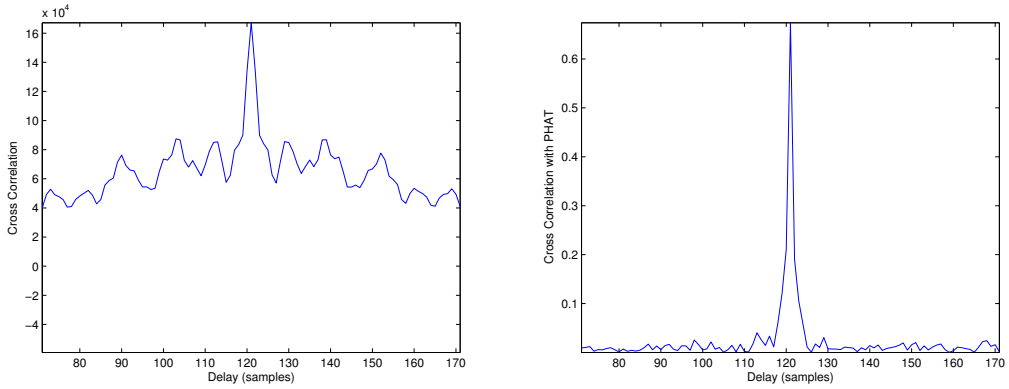
4.1 Steered Response Power

The SRP is a useful measure of the acoustic power originating from a particular location in space within a room, which has been shown to be relatively robust to reverberation [70, 71]. The GCC-PHAT from a set of microphones is used by the SRP algorithm to build up a 3D map of this power, as detailed in Section 2.3.2.3.

The GCC-PHAT [42] is a technique used to estimate the TDOA of an audio signal to a set of microphones. The method is popular as it is relatively simple and can give good results, even in noisy and reverberant environments [72]. The cross-correlation function is visually compared to the cross-correlation with PHAT weighting in Figure 4.1, which both show a peak at a time delay caused by an acoustic source. These plots were generated by a simulated speaker, subject to a RIR generated by the image method. The peak given by the correlation with PHAT weighting is much more distinct than that given by correlation with no weighting applied, with correlation values relatively low everywhere except at the true time offset. This makes the PHAT weighting useful for reliably extracting a peak corresponding to a TDOA.

4.1.1 Computational Complexity

Since the volume of a room is very large compared to the spatial resolution generally required by source tracking applications (see Section 2.3.2.5), and because of the computationally expensive nature of the algorithm [13], the calculation of the SRP across an entire room is infeasible. This means that an exhaustive search, where every point in space is interrogated for its



(a) Cross-correlation

(b) Cross-correlation with PHAT weighting

Figure 4.1: GCC-PHAT weighting comparison

power and the point with the maximum power deemed to be the source location, cannot practically be implemented. There are various methods [13,70,73] for finding the global maximum of the 3D array produced by searching over the room volume, however SRP based audio localisation also has the potential to locate and track multiple speakers more easily than the traditional maximal GCC TDOA methods [74]. This gives a strong motivation to develop techniques to efficiently search the SRP energy map across a room, and one such example is the technique presented in [75]. This method uses an inverse mapping to attempt to limit the spatial search to areas of maximal power. This reduces the computation time required to localise a source when compared to an exhaustive search of every possible grid point within the search volume. However, this comes at the cost of reduced spatial coverage, meaning possible missed sources, depending on the parameters used in the algorithm.

4.2 Stochastic Region Contraction

SRC [13] is a technique for locating a single audio source within a search space using the SRP. Instead of exhaustively searching through every possible location to find the peak SRP value, it speeds up the process by iteratively choosing points in space at which the SRP is to be evaluated. The search space is then reduced by restricting the next iteration of the algorithm to a search within an area bounded by a subset of the results of the current iteration. The boundary is defined as the cuboid volume containing only the locations of a subset of the highest SRP values. In this context, each SRP calculation is called a functional evaluation (FE).

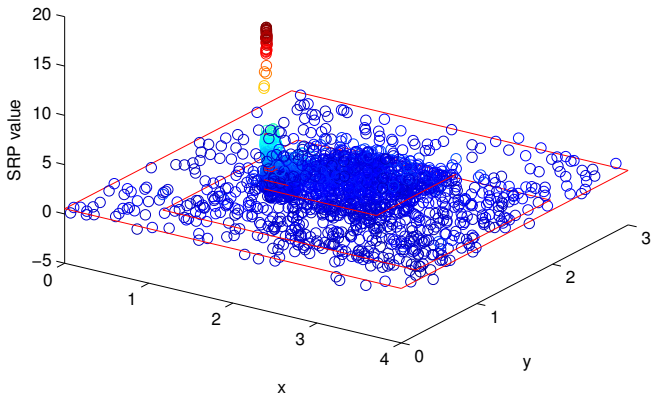


Figure 4.2: A 2D example of SRC, showing the search region contracting around the global maximum at each iteration i

4.2.1 Algorithm Description

The SRC uniformly takes samples of the SRP from across the search space and attempts to contract that search space to the area given by a set of the highest valued samples [13]. Because the higher valued SRP samples will generally be centred around a peak, caused by a sound source, the search area should quickly shrink. If a peak is not detected (the samples drawn miss the peak or any higher SRP values surrounding it), then in the single-speaker case, the remainder of the samples should be around the same magnitude.

The best samples will then be uniformly distributed across the search space, and the search space will not shrink appreciably. This allows the next iteration to sample the space again, and the source should be detected. Algorithm parameters determining the number of samples to be drawn have the obvious effect of influencing how little or often peaks are missed. Using very few samples will miss peaks more often than using many samples, resulting in higher search times. However, there is a balance to be made between reducing the number of samples to draw in order to reduce the number of FEs used, and using so few samples that the search area constricts very slowly and ends up requiring an even larger number of FEs.

By repeating this process, the search space will become an area sufficiently small enough to be considered the point which is the maximum of the SRP function and therefore the source of the sound. This is illustrated in Figure 4.2.

The initial number of random points to evaluate, J_0 is estimated in [13] by considering the probability of one of the uniformly distributed points being within the volume V_{peak}, which is the volume containing points with values higher on average than points in the surrounding area. J_i is then defined as the number of random points required for evaluation of the next iteration i of the algorithm. N_i is defined as the number of points needed to define the next search volume, V_{i+1}, which is a cuboid sub-volume of the original search volume. The algorithm also defines Φ, the maximum total number of FEs allowed to be evaluated, along with FE_i, which tracks the total number of FEs calculated. Also defined are V_u, the unit voxel which represents the smallest volume of space in which a FE can be calculated for, and T_1, a parameter normally set to about 10, which allows the algorithm to stop if V_{i+1} never quite gets close enough to V_u to stop within the specified maximum number of FEs allowed (Φ). The SRC algorithm for finding the global maximum is then given in Algorithm 2.

Initialise $i = 0$
Set parameters J_0, N_0 and $V_0 = V_{room}$
Calculate $S(x, y, z)$ for J_i points
Sort to find the highest N_i points
loop
    Contract the search volume V_{i+1} to only contain these N_i points
    if V_i is the unit voxel, or $FE_i > \Phi$ and $V_{i+1} < T_1 V_u$ **then**
        Stop and keep the result
    else if $FE_i > \Phi$ **then**
        Stop and ignore the result
    else
        Keep subset G_i of the points that are greater then the mean of the N_i points.
    end if
    Find J_{i+1} new points in V_{i+1}
    Set N_{i+1} = $G_i \cup B_i$, where B_i is the set of the $N_i - G_i$ highest new points from V_{i+1}
end loop

Algorithm 2: SRC Algorithm

4.2.2 Stochastic Region Contraction Variants

The parameters J_i and N_i can be chosen in several different ways, leading to several different variants of the SRC algorithm. The authors of [13] found it best to set a fixed value of N_i for all iterations based on their experimental results. From there, they defined three variants of the algorithm which each chose J_i using a different method.

The first method, SRC-I, sets J_i to the number of FEs needed to find $(N_i - G_i)$ points greater

than the average of the current set, μ_i. It uses a finite value for Φ. SRC-II sets J_i to the number of FEs needed to find $(N_i - G_i)$ points higher than the minimum of the complete set N_i and it also uses a finite value for Φ. Finally, for SRC-III, J_i is fixed to some value J and the $(N_i - G_i)$ highest points are chosen at each iteration. In this case, Φ is set to infinity.

4.3 Height-Estimated Stochastic Region Contraction

In order to reduce the number of FEs required for convergence, the nature of the different room axes should be considered. Whilst speakers might reasonably be located anywhere on the horizontal-plane within a room, their position in the vertical axis is normally more restricted. This is dealt with implicitly in some case, for example by limiting the search range over the vertical axis [13], centred around a likely head-height. This approach excludes speakers at unusual places such as up at the ceiling, or on floor. It also limits the general applicability of the method to scenarios where speakers are at different heights, for example, a speaker on a stage taking questions from an audience.

To choose head height, existing knowledge of the current positions and heights of people in a room can be used. In an audio-visual (AV) system, this is easy to initialise as video data can be used to make an initial estimation of the heights which should be searched in the audio domain. In addition, existing audio domain search techniques such as the full SRC algorithm can be used to make the first head height estimation. After they have been found initially, the tracked locations of people, both speakers and non-speakers, from both audio and visual sources will allow a good estimate of the height to be used across the room.

From a set of people sparsely distributed across a room, the head height to be used at every x-y co-ordinate in the SRP map will be defined. If there is only one person within a room, and that person has been previously localised in the audio domain, then their previously detected height will be used at their last known positions, under the assumption that a person might have moved across the horizontal plane, but is unlikely to have changed height.

If the positions of multiple potential speakers in a room are known, possibly from a people-tracking video system, then their heights can be extracted and used to reduce the acoustic search complexity for finding active speakers. Estimated height between potential speakers will be calculated by interpolating between the extracted heights. Interpolation can only be done between points, and because people are unlikely to be located in the corners of the search

area, estimated heights need to be defined differently for areas which are not between potential speakers.

This can be achieved by defining points at the corners of the room being searched, such that they are included in the set of points to be interpolated. A height needs to be defined for these extra locations, and this means that an assumption about the outer elements of the set of potential speakers and how they relate to the height at the edge of the search area must be made. This work uses the speaker closest to a corner to specify the height at that corner.

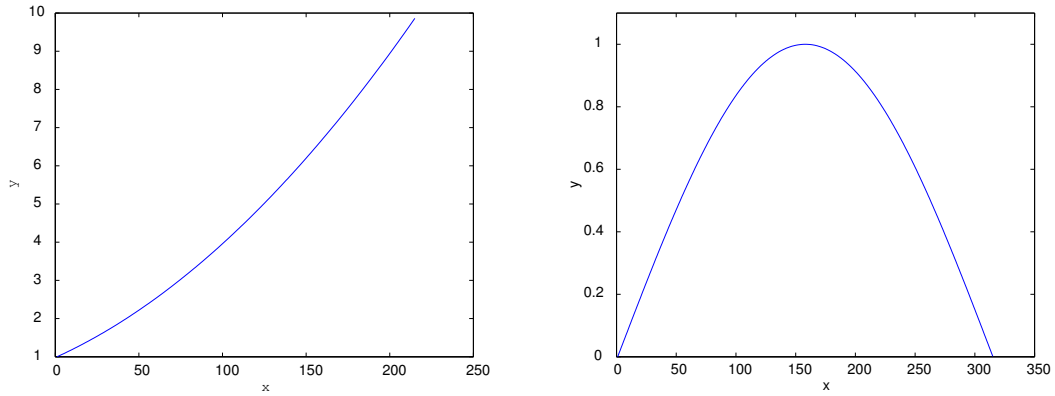
4.3.1 Interpolation

In order to define the head-height across an entire room, the height of known speakers can be used as points for interpolation, leading to a surface representing head-height over the entire horizontal-plane. When doing interpolation, there is a trade-off between the smoothness of the curve produced and the size of ripples produced. The interpolation should not contain severe ripples as they would lead to large errors in the head height estimation across the room. Ideally, there would be no ripples at all - the surface should be monotonic between all known points used for the interpolation.

4.3.1.1 Monotonic Interpolation

To guarantee that there are no ripples on the interpolated surface, the interpolating function should be monotonic between any two points, where a monotonic function maintains the order of the set of points. Figure 4.3 illustrates a monotonically increasing function and a non-monotonic function. In one dimension, monotonic interpolation can be achieved using linear interpolation or monotonic cubic interpolation [76].

Unfortunately, the extension of linear interpolation into 2D, bilinear interpolation, doesn't result in linear functions of the set of points, which means that monotonicity is not guaranteed. This is because whilst the interpolant function is linear across each axis x and y, it is the product of those two linear functions over any other straight line, i.e. quadratic interpolation.



(a) Monotonically increasing function

(b) Non-monotonic function

Figure 4.3: Illustration of a monotonic and a non-monotonic function in one dimension

4.3.1.2 Delaunay Triangulation

The lack of monotonicity in linear interpolation when applied to a set of points in more than one dimension required that another technique be found. Delaunay triangulation [77] can be used as an interpolation method which sacrifices the smoothness of the resulting interpolated surface for a surface which could be used reliably for height estimation, as it is monotonic. Delaunay triangulation obtains a piecewise linear interpolation of the set of speakers, which is a set of points in 2D. This creates a surface which can be evaluated at any 2D point within the convex hull (see Section 4.3.1.3) of the speaker set, which in turn allows an estimate for the head height to be taken between potential speakers within a room. To enable this ability across an entire room, the data also needs to be extrapolated from the convex hull of the set of speakers out to the edges of the room, which is detailed in Section 4.3.2. Figure 4.4 demonstrates an example surface created using Delaunay triangulation based interpolation.

This is in comparison to a plate-splines method [78] illustrated in Figure 4.5, which demonstrates the problem of ripples at the edge of the area. Whilst not orders of magnitude larger than the heights of the sources, indicated by green squares, the ripples produced are still a metre too high in some places. This is a large distance compared to the height of a person, and the amount of ripple is not generally controllable.

Note that in each diagram, there are four true sources in the centre of the room, two of which are lower than the others, to simulate a situation where some speakers are sitting down some are standing up. The green squares which mark the corners of the area - corresponding to

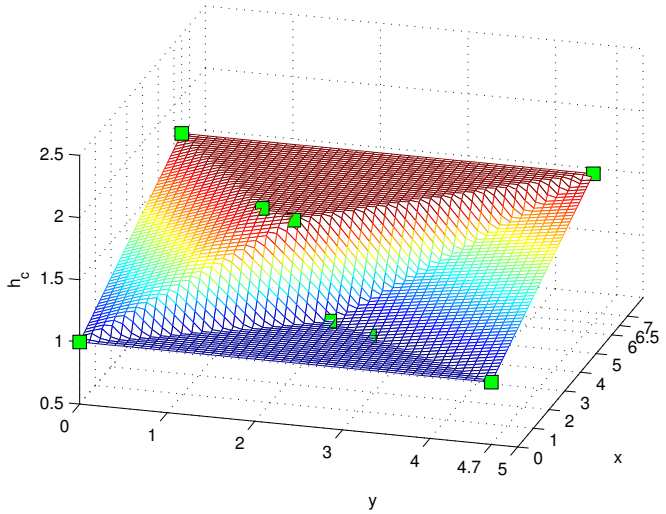


Figure 4.4: Delaunay triangulation method for estimating head height (h_c) as a function of position $(\mathbf{x}, \mathbf{y})$

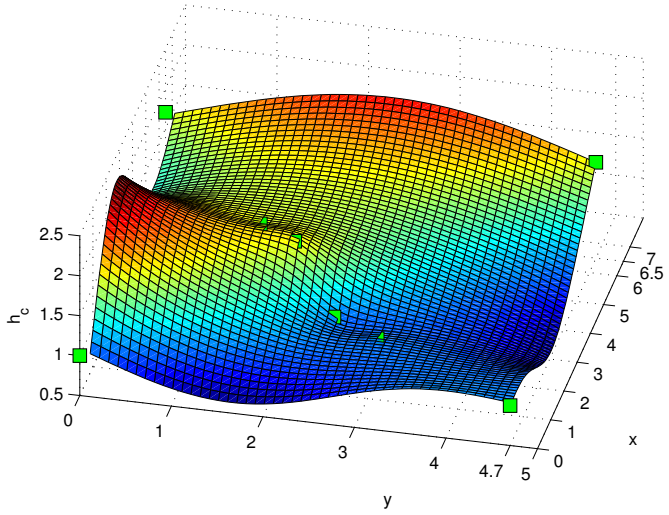


Figure 4.5: Plate-splines method for estimating head height (h_c) as a function of position $(\mathbf{x}, \mathbf{y})$

the corners of the room under consideration, have been extrapolated according the method described in 4.3.2. The room width and length are represented by the x and y axes and the interpolated heights h_c form the set $\mathbf{H}$ across the area of the room. Together, these diagrams show that the Delaunay method can avoid the problem of ripples, although it leads to a less smooth interpolation.

The Delaunay triangulation of a 2D set of points is the division of the plane containing those points into a set of triangles with those points as vertices. The triangulation tries to avoid triangles with small angles, as illustrated in Figure 4.6, which illustrates a set of points highlighted in red, with lines between the vertices indicating the Delaunay triangulation of these points. Also shown are the circumcircles of these point. The Delaunay condition (in 2D) requires that the circumcircle of a triangle not contain any other point from the set. This method can be extended to higher dimensions, using higher dimensional point sets and equivalents to planes and triangles. A widely used algorithm to calculate the Delaunay triangulation is Quickhull [79], which is used within MATLAB via the freely available QHull implementation. Quickhull calculates the Delaunay triangulation of a d-dimensional set of points by converting the problem into a calculation of the convex hull of the set in $(d + 1)$ dimensions.

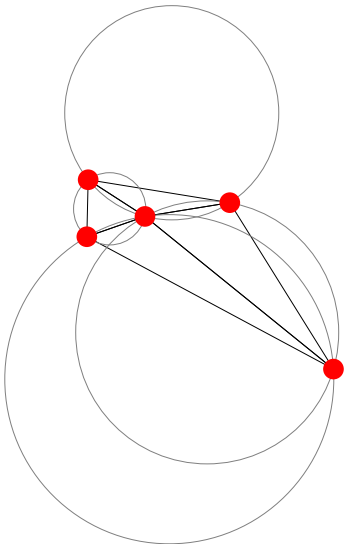
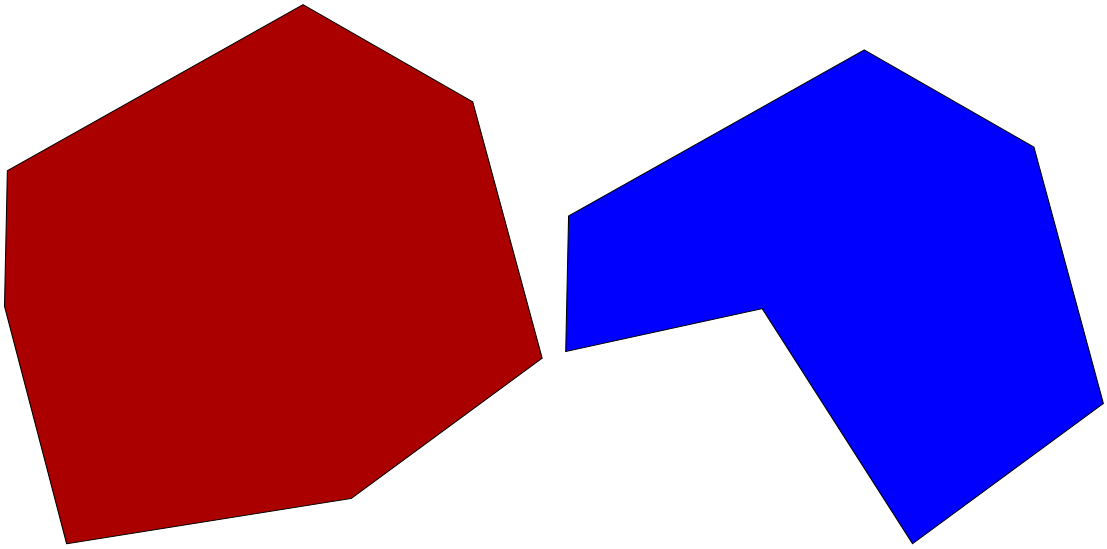


Figure 4.6: 2D Delaunay triangulation showing circumcircles

4.3.1.3 Convex Hull

The convex hull of a set of points S is the smallest convex subset of those points which contains that whole set, S. In 2D, this is a convex polygon, which is a polygon which does not intersect itself, and, for every pair of points on the boundary of the polygon, every point on a straight line joining that pair is also within the polygon. Figure 4.7 illustrates the difference between convex and non-convex polygons in 2D space.



(a) Convex Polygon (b) Non-convex Polygon

Figure 4.7: Illustration of convex and non-convex polygons in 2D

Figure 4.8 shows a visualisation of the elastic band analogy, which provides an intuitive view of the convex hull in 2D. If an elastic band is stretched out to enclose the entire set, it takes the shape of the convex hull of the set when it is released.

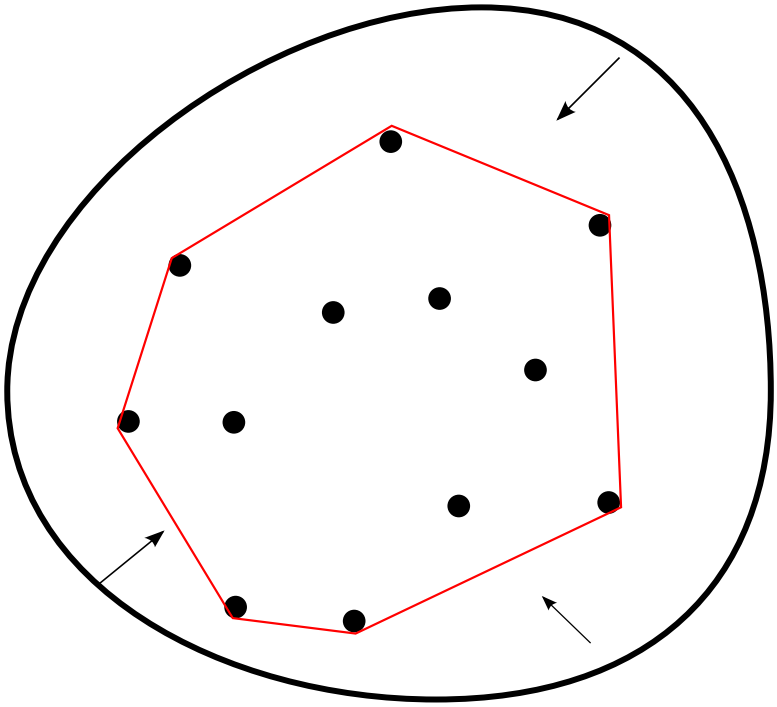


Figure 4.8: Elastic band analogy for the convex hull of a set of 2D points

Using the concept of convex hull, it is clear that given a set of speakers, the Delaunay triangulation of their positions will only be useful for deriving height information strictly between them. This means that an alternative has to be found for possible speaker locations outside of this convex hull.

4.3.2 Extrapolation

To perform the extrapolation task required, the Delaunay triangulation technique is further utilised by artificially expanding the convex hull of the set of speaker locations to cover the entire room. Note that in this work only simple cuboid areas of interest are considered for this task.

In order to extrapolate correctly, room corners must be pre-allocated nodes, which are assigned heights dynamically based on existing estimated speaker heights within a room. There are several options for choosing the height h_{c_j} at each of these j nodes (where in a rectangular room, $j = 4$), such as choosing the height at a corner point to be the same as the height of the nearest speaker, as shown in Equation (4.1a), where z_i is the height component of r_i, the position of known node i and r_{c_j} is the position of corner j. An alternative is to use

Equation (4.1b), the expected height of a speaker from all known node heights z_i. If it is assumed that there are a limited number of speakers then finding the nearest node to a corner poses no computational problems.

$$h_{c_j} = \arg\min_{z_i}[r_{c_j} - r_i] \tag{4.1a}$$

$$h_{c_j} = \mathbb{E}\left[z_i\right] \tag{4.1b}$$

With the heights of the corner nodes chosen, triangulation based interpolation can be used across the entire room, an example of which is shown in Figure 4.4.

4.3.3 Estimating the Height

Because the head height, H, is only an estimate, its accuracy varies across the room. To compensate, the head height to be used in the SRC algorithm is drawn from a PDF which ensures that most of the time, samples are taken around head height without being overly restrictive and a small amount of time from less likely areas, so as not to entirely neglect large portions of the search space. The interpolated head height from previously detected sources is used to model the mean of a Gaussian distribution whose variance changes depending on its proximity to a known source. This Gaussian distribution is then combined with a Uniform distribution across h_r, the entire height of the room, in a mixing model. This allows the search to concentrate on areas likely to contain people whilst at the same time, not neglecting to check for possible outliers. The height h_{sub} to use at each time step for every 2D point $\mathbf{p}_2 = (x_{p_2},\, y_{p_2})$ is drawn from the mixture model described in Equation (4.2) where $\mathbb{T}$ is the set of known speaker locations. In these equations, $\varphi\left(z \mid \mathbf{p}_2\right)$ represents the conditional probability of a height given the point $\mathbf{p}_2$, and α_0 is the mixing coefficient.

$$\begin{aligned} \varphi\left(z \mid \mathbf{p}_2\right) &= \alpha_0 \mathcal{N}\left(\mu_h,\, \sigma_h^2\right) + (1-\alpha_0)\,\mathcal{U}\left(0,\, h_r\right) \\ \mu_h &= H[\mathbf{p}_2] \\ \sigma_h^2 &= \hat{q}(\mathbf{p}_2,\, \mathbb{T}) \end{aligned} \tag{4.2}$$

This can be repeated n times to create an array where $h[n] = h_{sub}$, drawn from the mixture

distribution, each time. The resulting SRP value for the point $\mathbf{p_2}$ can either be the maximum value found as in Equation (4.3a) or the expectation (Equation (4.3b)) of the values, in which case as n increases, $SRP_{\mathbf{p_2}}$ tends towards the marginalisation of the SRP over z, the room height.

$$SRP_{\mathbf{p_2}} = \max_{z}[S(x_{p_2}, y_{p_2}, h[n])] \tag{4.3a}$$

$$SRP_{\mathbf{p_2}} = \mathbb{E}\left[S(x_{p_2}, y_{p_2}, h[n])\right] \tag{4.3b}$$

Around each person, we can be relatively confident of their height. Further away from them, the decreasing confidence is modelled by increasing the variance of the sampling PDF. The variance at a distance l metres from a speaker is chosen to be modelled by a sigmoid function, q, such as Equation (4.4a), which is a scaled error function, or Equation (4.4b), which is also sigmoid in shape.

$$\mathrm{q}(l) = \alpha_1 \operatorname{erf}(\alpha_2 l) \tag{4.4a}$$

$$\mathrm{q}(l) = \alpha_1(1 - e^{-l/\alpha_2}) \tag{4.4b}$$

Both of these equations evaluate to 0 at their origins and asymptotically approach constants as their arguments tend towards infinity. With an appropriate choice of coefficients, these functions can be used to limit the search height at a previously estimated speaker location to a narrow range. Moving away from the source position, the range of heights to be considered can be smoothly increased up to a predefined limit. It is convenient to choose this limit to match the original assumption made in the SRC algorithm, such that the sigmoid function limits to the upper and lower bounds of the original height range to be searched.

Whilst this chapter primarily considers single source localisation, the height estimation method can be extended to accommodate multiple speakers for integration with multi-source localisation techniques, particularly if video localisation methods are used in conjunction with the audio data. If two speakers are positioned close together, then the sigmoid functions associated with each speaker might overlap. As such, a decision needs to be taken about the variance to be used at points in space which are part of the overlap. The variances are combined to form a

global variance in Equation (4.5).

$$\begin{aligned} \mathbb{L}_{\mathbf{p},\mathbb{T}} &= \{l : (\exists \mathbf{q} \in \mathbb{T})(l = |\mathbf{p} - \mathbf{q}|)\} \\ \hat{q}(\mathbf{p_2}, \mathbb{T}) &= \min_{l \in \mathbb{L}_{\mathbf{p},\mathbb{T}}} q(l) \end{aligned} \tag{4.5}$$

At any point $\mathbf{p}$ in space, the appropriate variance $\hat{q}$ to use will be the sigmoid function q of the minimum of the set of all 2D Euclidean distances $\overline{\mathbf{pq}}$ to known sources, where an element of $\mathbb{T}$ is denoted as $\mathbf{q}$. The minimum is chosen to ensure that the change in variance remains smooth even for overlapping sigmoids from multiple sources.

4.3.4 Algorithm Description

The algorithm for finding the global maximum using the estimated head height is given in Algorithm 3, where DT denotes the Delaunay Triangulation operation.

Initial search for a speech source
while $running$ **do**
  $\hat{\mathbb{T}} = \mathbb{T}$
  for all room corners **do** ▷ Add room corners to $\hat{\mathbb{T}}$
    $\mathbf{n} \leftarrow (x_{\text{corner}},\, y_{\text{corner}},\, z_{\text{nearest member of } \mathbb{T}})$
    $\hat{\mathbb{T}} \leftarrow \hat{\mathbb{T}} \cup \{\mathbf{n}\}$
  end for
  $\hat{\mathbb{H}} \leftarrow \mathrm{DT}(\hat{\mathbb{T}})$ ▷ Delaunay Triangulation of the set
  for all $\mathbf{p_2} = (x_{p_2},\, y_{p_2}) \in \mathbb{A}$ **do** ▷ Whole search area
    $\hat{\mathbb{H}}_0 \leftarrow h_{sub} \sim \varphi\,(z \mid \mathbf{p}_2)$ ▷ Choose a height
  end for
  Perform SRC with heights from $\hat{\mathbb{H}}_0$
  $\mathbb{T} = \mathbb{T} \cup \{\text{new speaker positions}\}$
end while

Algorithm 3: HE-SRC Algorithm

4.3.5 Selection of Stochastic Region Contraction Parameters

The variant of SRC used was SRC-I, due to its relatively fast convergence compared to the other variants. SRC-I fixes the number of points to be evaluated at the first iteration, J_0, to a constant J and then calculates J_i FEs at each iteration i of the algorithm, where J_i is determined dynamically [13]. In this variant, the parameter N used to contract the search region [13] can be chosen to adjust the localisation accuracy of the algorithm. The original authors chose a value

of $N = 100$, which gave them the lowest computational cost without sacrificing accuracy.

However, because the algorithm has been changed, it is necessary to investigate again the selection of the parameters used. Figure 4.9 shows the average number of FEs required to localise a source for different values of N using the Height Estimated variant of SRC-I. Similarly, Figure 4.10 shows that as N is increased, the average localisation error decreases. As such, in our experiments using Height Estimated SRC-I (HE-I), we chose $N = 35$ to keep the total number of FEs low and provide a reasonably low average error. Note that the experiments to determine these parameters were performed as a Monte Carlo experiment. The results come from the average results of the algorithm run over each individual speaker 100 times, resulting in thousands of HE-SRC trials being run.

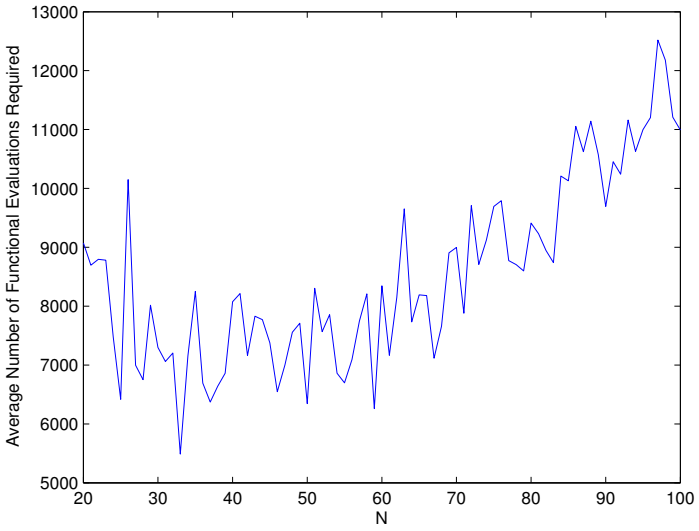


Figure 4.9: FE's of HE-SRC using SRC-I required to localise a source as a function of N

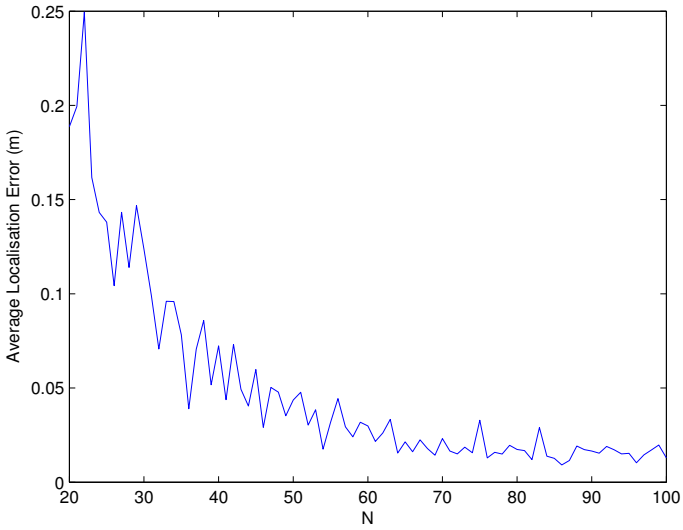


Figure 4.10: Localisation error of HE-SRC using SRC-I as a function of N

4.4 Experimental Results

The algorithms were run in the environment shown in Figure 2.12 on recorded data, where the red circles represent 12 of the microphones used (placed along the edges of the room, similar to the panels used in [80]. This set up is used to enable a direct comparison, as the HE-SRC is an extension of the work in [80]). There were 16 microphones in total, where two microphones were placed under each line of microphones in the diagram, forming two sets of microphones in 'T' configurations. The green squares represent the speaker positions. The data was recorded in the (4.7x6.5)m room, as described in [80] in order to make a direct comparison. A minute of data was recorded for each speaker at 96kHz, which was downsampled to 44.1kHz (to reduce the memory load of the algorithm), which gave each around 160 audio windows based on a window size of 2^{14} samples (0.37s). As before, these experiments were run as a Monte Carlo experiment, this time around 200 times over each speaker. This allowed representative averaged results to be collected.

Speakers did not talk at the same time and the two speakers furthest away from the array were at the lower height of 1m, rather than 1.6m, in order to show that speaker height can be successfully accounted for by algorithm as designed. With no height information available from a camera system, these experiments simply used the previously tracked speaker height from the acoustic data to estimate the height data across the room. This meant that the only interpolation

done was between the speaker and the points at the four corners of the room, defined to be at the same height as the speaker. This resulted in a plane parallel to the horizontal plane as the estimated speaker height, and the variance of the Gaussian distribution of the vertical axis was varied around that position.

α_0 (Equation (4.2)) was chosen to be 0.95 in order to concentrate the search within head height, allowing for the very small possibility that a source might at some point be located, for example, on the ground. Lower values weight the distribution to uniformly draw from across the height of the room, making the search similar to the original SRC algorithm, but with fewer assumptions and therefore slower searches. α_1 was chosen to be 0.5, allowing most of the Gaussian distribution to concentrate on an area 1m tall, similar to the 1m tall Uniform distribution used for height in the original SRC algorithm. Finally, α_2 was generated by choosing the radius l, at which the sigmoid function should be 99% of the way towards α_1, to be 1m, which assumes people have some personal space whilst talking.

Data was evaluated using an average location error (ALE) - the mean of the Euclidean distances of each set of results to their corresponding ground truths. Because the search space was reduced by the height estimation, the number of samples J_i at each stage was lowered to improve overall search times, trading off against accuracy. In the first instance, HE-I, only 350 samples were taken at the first iteration with only $N = 35$ used for region contraction. Accuracy decreased as the sound source was further away from the microphone array, implying a lower signal to noise ratio (SNR) as in [13], but this may be acceptable in a system whose tracker accounts for noisy state observations. For HE-II, J_0 was set to 1000 and N to 60, which increased the accuracy across all sources whilst keeping the number of FEs low. In HE-III, J_0 was set to 3000 and N to 60, the value as used in [13].

Table 4.1 shows the results of first (SRC-I) variation of the SRC algorithm from [13] on the data set and compares these configurations with the HE variants. It shows the average number of FEs used within an audio frame and the ALE, where Source 1 is the closest to the microphone array and Source 4 is the furthest. Note that there results come from a Monte Carlo simulation, where the algorithm was run on the same data sets around 200 times. The averages are therefore across thousands of runs of the localisation algorithm, as each Monte Carlo trial consisted of hundreds of time-steps, on each of which the algorithm was executed.

The results show that with prior information about head height within a room, the SRC can

Algorithm	Source 1		Source 2		Source 3		Source 4	
	ALE (m)	# FEs	ALE (m)	# FEs	ALE (m)	# FEs	ALE (m)	# FEs
SRC-I	0.26	61,1001	0.31	61,1001	0.45	61,001	0.6	61,001
HE-I	0.32	17,156	0.35	21,939	0.44	31,811	0.58	35,053
HE-II	0.12	34,022	0.22	35,136	0.26	41,228	0.5	39,402
HE-III	0.11	40,721	0.15	40,736	0.23	42,900	0.34	44,111

Table 4.1: Comparison of SRC Methods

Algorithm	Source 1		Source 2		Source 3		Source 4	
	ALE (m)	# FEs	ALE (m)	# FEs	ALE (m)	# FEs	ALE (m)	# FEs
HE-I	0.45	20,115	0.49	23,849	0.51	36,253	0.6	37,962
HE-II	0.23	35,117	0.24	36,140	0.41	47,281	0.47	48,294
HE-III	0.22	43,783	0.24	43,548	0.32	55,352	0.46	56,667

Table 4.2: FEs required to find a source with no prior

be sped up whilst maintaining accuracy. Because in HE-III the parameters are similar to the SRC-I parameters, the algorithms are expected to perform similarly when there is no known audio source. In this case, the mean of the Gaussian is set to the same offset as that used in the algorithm and the variance is again set to 0.5.

Table 4.2 shows the average number of FEs required to find a source using the algorithm without prior information. The results indicate an increased computational load with HE-III, but still within the tractable range of tens of thousands of FEs and close to the performance of SRC-I, as expected. For lower values of J_0 and N, the results are improved. In particular, HE-II provides good accuracy and good performance, with or without prior information, so much so that it is suitable as an audio estimator for the initial height information in this situation.

4.4.1 Visual Height Cues

As previously noted, detecting and tracking speakers in both the audio and visual domains is made harder by the problems such as visual occlusion and audio silence where a speaker moves after having been detected previously and then starts speaking again. By fusing the data from both domains, one might hope that at any one time, a speaker will be locatable using at least one of these modalities. The work in this chapter has been adapted with a video based tracking system to provide in a joint effort with Heriot-Watt University [81].

An experiment was set up in which a set of cameras was used to extract the positions of a pair of speakers having a conversation. This information was fed into a joint audio-visual detection system, represented in Figure 4.11. As a first step in the process, the height information of

the people in the room - both potentially speakers, but not concurrent speakers - detected by the video system was used to cue the initial height information used for interpolation in the HE-SRC algorithm. This processing step is indicated in the system diagram by the arrow marked with a '1'. In return, the joint detector results were improved over using the original SRC as the audio processing block, indicated by the arrow marked '2', where both acoustic and visual information is fused to perform tracking. The output of this step is then used for height estimation by the acoustic module, and this feedback step is indicated by the arrow marked '3'.

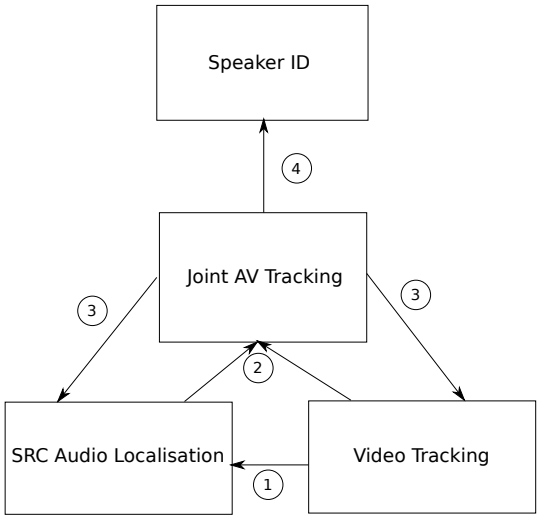


Figure 4.11: Joint detection system

4.5 Tracking

The HE-SRC method developed returns a location estimate for each frame in Cartesian co-ordinates. Bayesian filtering methods are commonly used to track a series of localisation results, allowing the localisation error to be reduced over time by taking account of source movement models and compensating for noisy, spurious and missed measurements.

This section applies a simple Kalman filter to the data localised by the HE-SRC, similar to the application of an extended Kalman filter to a contemporary piece of work which localises sources from raw TDOA measurement. The application of this filter results in reduction in the localisation error.

4.5.1 Time-Difference Of Arrival Extended Kalman Filter Tracker

In [8], an EKF is developed to track a speaker directly from a series of TDOA measurements, and it is to this work that the tracking scheme used in this Chapter is to be compared. For reference, in their paper they note that for maximum likelihood (ML) based localisation, the error function $\epsilon(\mathbf{x})$, given in Equation (4.6), must be minimised. In this equation, $\hat{\tau}_i$ is the TDOA value observed by microphone pair i, and σ_i^2 is the observation error covariance. $T_i(\mathbf{x})$ is the true TDOA between microphone pair i for a speaker at position $\mathbf{x} \in \mathbf{R}^3$.

$$\epsilon(\mathbf{x}) = \sum_{i=0}^{N-1} \frac{1}{\sigma_i^2} [\hat{\tau}_i - T_i(\mathbf{x})]^2 \tag{4.6}$$

Unfortunately, $T_i(\mathbf{x})$ is non-linear in $\mathbf{x}$ and so minimising Equation (4.6) is non-trivial. To work around this, the partial derivative is taken by Klee et al. in Equation (4.7), and used in Equation (4.8a) to approximate the TDOA function, $T_i(\mathbf{x})$, with a first-order Taylor series using the previous estimate of the speaker position, $\hat{\mathbf{x}}(t-1)$. This allows the error criterion to be minimised over a series of time points, which takes into account that a speaker's position cannot change instantaneously.

$$\nabla_{\mathbf{x}} T_i(\mathbf{x}) = \frac{1}{s} \left[\frac{\mathbf{x} - \mathbf{m}_{i1}}{d_{i1}} - \frac{\mathbf{x} - \mathbf{m}_{i2}}{d_{i2}} \right] \tag{4.7}$$

$$T_i(\mathbf{x}) \approx T_i(\hat{\mathbf{x}}(t-1)) + \mathbf{c}_i^T(t) \left[\mathbf{x} - \hat{\mathbf{x}}(t-1)\right] \tag{4.8a}$$

$$\mathbf{c}_i^T(t) = \left[\nabla_{\mathbf{x}} T_i(\mathbf{x})\right]_{\mathbf{x}=\hat{\mathbf{x}}}^T (t-1) \tag{4.8b}$$

The linearisation can be used to approximate the error function (Equation (4.6)), which is shown in Equation (4.9). $\bar{\tau}_i(t)$ is defined in Equation (4.10).

$$\begin{aligned} \epsilon(\mathbf{x}; t) &\approx \sum_{i=0}^{N-1} \frac{1}{\sigma_i^2} \left\{ \hat{\tau}_i - T_i(\mathbf{x}(t-1)) - \mathbf{c}_i^T(t) \left[\mathbf{x} - \hat{\mathbf{x}}(t-1)\right] \right\}^2 \\ &= \sum_{i=0}^{N-1} \frac{1}{\sigma_i^2} \left[\bar{\tau}_i(t) - \mathbf{c}_i^T(t) \mathbf{x} \right]^2 \end{aligned} \tag{4.9}$$

$$\bar{\tau}_i(t) = \hat{\tau}_i(t) - T_i(\hat{\mathbf{x}}(t-1)) + \mathbf{c}_i^T(t)\hat{\mathbf{x}}(t-1) \tag{4.10}$$

With the error function to be minimised now defined, an EKF (Section 3.1.4) or an IEKF (Section 3.1.5) can be used to facilitate speaker tracking. To do so, a state transition model which corresponds to the expected movement of a speaker is required. The model used in [8] is simply to assume that a source is stationary except for slight movement modelled by additive Gaussian noise. The non-linearity of the TDOA measurements is taken into account using a non-linear observation function, which precludes the use of a simple linear Kalman filter.

4.5.2 Height Estimated Steered Response Power Kalman Filter Tracker

Because the HE-SRC returns direct position measurements, the formulation of a Kalman filter based tracker for a single source can be simplified slightly. Because there is no longer a non-linear relationship between the measurements and the system state and the state transition function is assumed to be linear by the choice of source dynamics model, the simple Kalman filter can be used. Similar to Equation (4.6), the error function $\epsilon(\mathbf{x})$ defined in Equation (4.11) must be minimised.

$$\epsilon(\mathbf{x}) = \sum_{i=0}^{N-1} \frac{1}{\sigma_i^2} \left[\hat{\mathbf{x}}_i - \mathbf{x}\right]^2 \tag{4.11}$$

As before, this is the minimisation of the squared difference between an estimation and the ground truth. However in this case, it is not the TDOA error which is being minimised, but the raw source position estimation error. Furthermore, there is only one observation for each time frame, as the SRC returns a single position estimate using data from all microphones in the system. As such, the error function $\epsilon(\mathbf{x},\, t)$ at time step t reduces to Equation (4.12).

$$\epsilon(\mathbf{x},\, t) = \frac{1}{\sigma^2} \left[\hat{\mathbf{x}} - \mathbf{x}\right]^2 \tag{4.12}$$

Because the observations are now linear, there is no need to use a non-linear observation function in the Kalman filter. Instead, the observation matrix $\mathbf{H}$ can be used simply as an identity matrix, leading to the standard Kalman filter equations shown in Equations (4.13a) to (4.13e).

There is no need for a control input in Equation (4.13a), however the rest of the Kalman filter equations are simply the regular Kalman equations as presented in Chapter 3.

$$\hat{\mathbf{x}}'_t = \mathbf{A}\hat{\mathbf{x}}_{t-1} \tag{4.13a}$$

$$\mathbf{P}'_t = \mathbf{A}\mathbf{P}_{t-1}\mathbf{A}^\intercal + \mathbf{Q} \tag{4.13b}$$

$$\mathbf{K}_t = \mathbf{P}'_t\mathbf{H}^\intercal \left(\mathbf{H}\mathbf{P}'_t\mathbf{H}^\intercal + \mathbf{R}\right)^{-1} \tag{4.13c}$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}'_t + \mathbf{K}_t \left(\mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}'_t\right) \tag{4.13d}$$

$$\mathbf{P}_t = \left(\mathbf{I} - \mathbf{K}_t\mathbf{H}\right)\mathbf{P}'_t \tag{4.13e}$$

Because the speakers are only assumed to be moving under process noise, the state vector doesn't include a velocity component and the state transition matrix $\mathbf{A}$ can be set to an identity matrix. This source dynamics model is somewhat limited, however its use in this chapter has allowed for the development of a simple tracker along the same lines as that developed by Klee et al [8]. When considering a moving source, more complex models of source movement might be used, and several examples are given in Section 2.4.1.

4.5.3 Tracking Integration and Results

The model of the observation noise covariance matrix $\mathbf{R}$ can be obtained by considering the ALE of the HE-SRC localiser. It is assumed that the estimation noise is independent and also the same in each dimension, such that there are no cross-correlation terms. $\mathbf{R}$ is therefore expressed as in Equation (4.14), where $\sigma_{\mathbf{R}}$ is the localisation standard deviation. Given the localisation techniques' typical ALE, it is reasonable to assume this value is around 30cm.

$$\mathbf{R} = \sigma^2_{\mathbf{R}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.14}$$

To integrate the tracking algorithm, the localisation process is run on every audio block as before, however the final state estimate from an audio frame is obtained by using the Kalman filter with the localisation observation. Using the same stationary source model as Klee et al.,

Algorithm	Source 1		
	ALE (m)	Filtered ALE (m)	# FEs
HE-I	0.32	-	17,156
HE-I + KF	0.199	0.115	20,268

Table 4.3: Comparison of HE-SRC with and without Kalman filtering

and assuming that the ground truth is accurate to within 10cm, the filtered results are used as input to the height interpolation and extrapolation algorithm.

This simple technique has been used to demonstrate the improved source localisation accuracy of using a tracking algorithm in conjunction with a raw position estimation algorithm. This is shown in Table 4.3, where the results for the HE-SRC algorithm for the first speaker in the recorded audio data are included for comparison. Note that using the Kalman filter decreased the raw average localisation error, as the technique prevented detections of spurious sources from significantly affecting the localisation results of subsequent audio frames. This was because incorrect localisations do not immediately significantly alter the filter state, and therefore the initialisation of the algorithm is not significantly affected by the occasional spurious measurement.

This shows that as expected, introducing a tracking algorithm can lead to a reduction in the estimation error whilst maintaining the computational efficiency. Between two cases, the average number of functional evaluations is close, but there is a distinct advantage to using a Kalman filter.

4.6 Conclusions

This work contributes a method of speeding up and increasing the accuracy of the SRC algorithm by estimating the height at which to search from prior information, obtainable either via a camera based system, or from information from the previous iteration of the algorithm. The key to this technique is to estimate an average head height across an area by interpolating and extrapolating heights of known potential speakers and forming a probability distribution of head height using this data. This allows a single audio source to be localised quickly whilst still searching across the room to find new source, for example when there is a speaker change. This localisation technique was combined with a simple Kalman filter, which reduced the average localisation error for a stationary source.

The audio data used for these experiments consisted of non-concurrent stationary speakers, which allowed a simple model to be used within the Kalman filter. Because the work in this chapter built on previous single source localisation work, only non-concurrent speakers could be considered. Both of these limitations are unrealistic for a practical ASLT system, and multi-speaker systems with moving sources will be considered in subsequent chapters.

It should be noted that the individual FEs are parallelisable within each iteration of the algorithm, and with this extension, the significant reduction in the number of FEs required suggests that the algorithm could be suitable for practical implementation using graphics processing unit (GPU) programming techniques such as compute unified device architecture (CUDA) to attempt to provide real time source localisation.

Chapter 5
Particle Swarm Methods for Audio Source Localisation

The SRP has potential to be used for multiple source localisation as individual acoustic sources appear as distinct individual peaks of the function over space. This chapter concentrates on a novel technique for source localisation, primarily in the case of a single speaker, with a view to being extensible to the multi-speaker case.

This is in contrast to the work in Chapter 4, which might not be readily extendible to multi-target localisation. Running the SRC algorithm on multi-target data can cause the contracting cuboid to become stuck if encapsulates multiple distinct peaks, and so a more flexible scheme would be preferable. In contrast, the chosen scheme is part of a broad family of search techniques which include multi-target optimisers. Therefore, it is of interest to show that the chosen optimisation scheme can be made to work well in the single source case, before examining the use of multi-target optimisers from the same family.

Neglecting peaks caused by reverberation, which are a problem for any localisation method, the task of acoustic source localisation can be thought of as the task of finding a set number of maxima of the SRP function. Particle swarm techniques are investigated in this chapter as a solution to this problem.

In this chapter, it is firstly assumed that there is only one speaker active at any time. With this assumption in place, the number of SRP maxima to be found in any time window is fixed, and a localisation algorithm can finish when it has found a single peak. This approach might be extended to the case of multiple and an unknown number of speakers by using an estimate of the number of speakers in each frame to provide an upper bound for the number of peaks to localise, and using multi-optima detection techniques.

This chapter considers the case of a single speaker to be localised, as in Chapter 4, and applies PSO to the problem to provide a point estimate of the source location. Several variants of the particle swarm algorithm are considered, and evaluated on both their accuracy and rate of

convergence on both recorded and simulated data. The work is then expanded to use height estimation information in an effort to speed up the search, similar to the work in Chapter 4.

5.1 Particle Swarm Optimisation

Localising a peak of the SRP function can be seen as a numeric optimisation problem. As described in Chapter 4, the SRC attempts to maximise the SRP by iteratively contracting the search space around a region which has a high SRP value. This method assumes that the area of space approaching a peak of the objective function will also be at a relatively high level. In contrast to many optimisation techniques however, the method does not attempt to make use of the gradient of the function to direct the search. Such an approach might be considered inappropriate because the SRP result is generally quite noisy, making gradients across space potentially misleading.

When considering the optimisation of the SRP, we restrict ourselves to considering metaheuristic techniques which, whilst imperfect, attempt to find solutions without guaranteeing that they will find a global optimum. A promising alternative to the iterative SRC, known as PSO has been applied to speaker detection in the context of an extension [10] of a tracking system developed by Ward et al. [11], which introduces swarm dynamics to a particle filter.

The work in [11] makes use of the SRP as a pseudo-likelihood function for a particle filter approach to source tracking. The work in [10] extends this by allowing the particles to interact according to the PSO paradigm, as an integral part of the particle filtering framework. This speeds up the filter's convergence of a source.

In contrast, the focus of this Chapter is the application of PSO as a raw localisation routine. This will provide point estimate results for the single speaker localisation problem, which can then be filtered as a second stage. This can be done, for example, with a Kalman filter, which is less computationally intensive than a particle filter. The use of PSO techniques is further justified as they are found to be computationally efficient. This is important, because even though the SRC is much more efficient than an exhaustive search, it is still hard to implement in an online solution. Work has been done to create parallelised implementations [82, 83], in an attempt to rectify this situation. With the potential to perform multi-target localisation, PSO is attractive as it is also potentially parallelisable.

The performance of this method over a range of noise conditions is studied and the relative performance of several different variants of the algorithm will also be explored. Methods of adapting the algorithm to the audio localisation problem will be considered, with the fact that speakers might be spread uniformly across the horizontal-plane, but their distribution is not uniform over the vertical axis.

5.1.1 Description

Particle swarm optimisation is similar to SRC in that the algorithm makes use of a set (known as a swarm) of candidate locations which could provide the final solution to the problem. These candidates, referred to as particles, are moved around the search space in an attempt to converge on an optimum point. In contrast to SRC, the particle movement is determined by a particle's local knowledge of the value of the objective function as well as the swarm's overall knowledge.

5.1.2 General Particle Swarm Optimisation Algorithm

PSO was developed as a stochastic optimisation method inspired by the behaviour of large groups of animals such as flocks of birds [84]. Each particle in the swarm moves around the search space with a dynamically adjusted velocity which is dependent on its own observations and the overall observations of the entire particle swarm.

At a time step t, the velocity of the i^{th} particle $\mathbf{V}_i$ is given by Equation (5.1a) and the position $\mathbf{X}_i$ of that particle is given in Equation (5.1b).

$$\mathbf{V}_i\left(t+1\right) = \mathbf{V}_i\left(t\right) + c_1 r_1 \left[\mathbf{p}_i\left(t\right) - \mathbf{X}_i\left(t\right)\right] + c_2 r_2 \left[\mathbf{p}_g\left(t\right) - \mathbf{X}_i\left(t\right)\right] \tag{5.1a}$$

$$\mathbf{X}_i\left(t+1\right) = \alpha_p \mathbf{X}_i\left(t\right) + \beta_p \mathbf{V}_i\left(t+1\right) \tag{5.1b}$$

The vector $\mathbf{p}_i$ records the best historical position (that is, the position with the highest value of the SRP objective function) of that particle and $\mathbf{p}_g$ records the best historical position found over the entire swarm. α_p and β_p are position control parameters, and c_1 and c_2 are weighting variables for the two components with a recommended default value of 2. Finally, r_1 and r_2 are random variables both with a range of $[0,\ 1]$.

The standard procedure for performing a PSO search is detailed in Algorithm 4, where typical stopping conditions include a maximum number of iterations completed (g_{best}) or the best value not changing by an amount over some threshold (g_{thresh}) between estimations.

Initialisation
for $i = 1$ to Swarm Size **do**
    Set $\mathbf{X}_i$ randomly within the search range
    Set $\mathbf{V}_i$ randomly within the permissible velocity range
    Assign the particle's best position to the current (initial) position
    Evaluate the objective function at the particle position
end for
Identify $\mathbf{p}_g$, the swarm's best result
while Stopping Conditions Unmet **do**
    for $i = 1$ to Swarm Size **do**
        Initialise random variables r_1 and r_2
        Update particle velocity and position
        Evaluate the objective function at the new particle position
        Update the particle's best known position
    end for
    Identify $\mathbf{p}_g$, the swarm's best result
end while

Algorithm 4: PSO Algorithm

5.1.3 Commom Particle Swarm Optimisation Variants

There are several commonly used variants [85, 86] of the PSO algorithm, which are worth considering when applying the technique to speaker localisation. These versions can all be applied to and evaluated on an acoustic data set for a performance comparison.

5.1.3.1 Shi

Shi and Eberhart extended Eberhart's original work by introducing new parameters which control the velocity update step [87]. The updated particle speed and position formulae are given in Equations (5.2a) and (5.2b) respectively, which introduce the new variable ω_p. Shi et al. also recommend that the variables c_1 and c_2 remain set to 2.

$$\mathbf{V}_i(t+1) = \omega_p \mathbf{V}_i(t) + c_1 r_1 \left[\mathbf{p}_i(t) - \mathbf{X}_i(t)\right] + c_2 r_2 \left[\mathbf{p}_g(t) - \mathbf{X}_i(t)\right] \tag{5.2a}$$

$$\mathbf{X}_i(t+1) = \alpha_p \mathbf{X}_i(t) + \beta_p \mathbf{V}_i(t+1) \tag{5.2b}$$

The coefficient ω_p is an inertial weight, where a smaller value favours more local exploration [88]. This factor allows a trade off to be made between an aggressive local search and a wide-ranging search, which can then be used to optimise the convergence time and number of steps used in the algorithm.

Furthermore, it is suggested that the inertial factor used can be a function of time, such that as the search progresses, more effort is made to search a smaller area. Many different inertia weight strategies exist, and [89] provides an overview of their formulae and compares their effectiveness given different goals such as minimising the number of iterations or minimising the average error.

One such inertia weight management scheme is to decrease the weight linearly between a minimum and maximum value [90, 91], as shown in Equation (5.3). In this equation, the inertia weight w_t at time step t is simply decreased relative to the current time step, the maximum allowable number of time steps $t_{\max}$. The value of w_t is constrained between a starting value, w_s and a final value, w_e.

$$w_t = (w_s - w_e)\,(t_{\max} - t)\,\frac{1}{t_{\max}} + w_e \tag{5.3}$$

5.1.3.2 Clerc

The PSO extension by Clerc and Kennedy [92] introduced a constriction coefficient, χ, with several different classes of solution. Clerc's Type 1″ has been widely used due to its relative simplicity, and the updated formulae are indicated in Equations (5.4a) and (5.4b). φ is the sum of two random variables, φ_1 and φ_2. φ_1 and φ_2 are both uniformly distributed between 0, and $\varphi_{\max,1}$ and $\varphi_{\max,2}$ respectively, which are parameters to the algorithm mapping to the weighting constants, in the regular PSO, c_1 and c_2.

$$\mathbf{V}_i\,(t+1) = \chi\,(\mathbf{V}_i\,(t) + \varphi\mathbf{Y}\,(t)) \tag{5.4a}$$

$$\mathbf{Y}_i\,(t+1) = -\chi\mathbf{V}_i\,(t+1) + (1 - \chi\varphi)\mathbf{Y}_i\,(t) \tag{5.4b}$$

The constriction coefficient χ for the Type 1″ variant is calculated as shown in Equation (5.5), with the constriction parameter κ with limits $0 \leq \kappa \leq 1$. Note that Clerc's notation is simplified

by writing the new attractor as $\mathbf{Y}(t) = \mathbf{p}_c - \mathbf{X}_i(t)$, with $\mathbf{p}_c$ defined in Equation (5.6).

$$\chi = \begin{cases} \sqrt{\frac{2\kappa}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}}} & \text{for } \varphi = \varphi_1 + \varphi_2 > 4 \\ \kappa & \text{for } \varphi = \varphi_1 + \varphi_2 \leq 4 \end{cases} \tag{5.5}$$

In these equations, Clerc simplifies the problem by first expressing the latter half of Equation (5.2a) as a single expression, given in Equation (5.6).

$$\mathbf{p}_c = \frac{\varphi_1 \mathbf{p}_i + \varphi_2 \mathbf{p}_g}{\varphi_1 + \varphi_2} \tag{5.6}$$

The particle velocity is then modified by only one attractor, $\mathbf{p}_c$, multiplied by the constant $\varphi = \varphi_1 + \varphi_2$.

5.1.3.3 Trelea

Finally, Trelea's variant [93] of the algorithm is expressed in Equations (5.7a) and (5.7b). This deterministic version of PSO simply replaces the random variables and their weights with constant weights.

$$\mathbf{V}_i(t+1) = a\mathbf{V}_i(t) + b\left[\mathbf{p}_i(t) - \mathbf{X}_i(t)\right] + b\left[\mathbf{p}_g(t) - \mathbf{X}_i(t)\right] \tag{5.7a}$$

$$\mathbf{X}_i(t+1) = \alpha_p \mathbf{X}_i(t) + \beta_p \mathbf{V}_i(t+1) \tag{5.7b}$$

After some analysis, Trelea recommended two sets of parameters, both with α_p and β_p set to 1. The first of Trelea's parameter sets, denoted 'Trelea 1', has $a = 0.6$ and $b = 1.7$, and the second set, 'Trelea 2', has $a = 0.729$ and $b = 1.494$.

5.1.4 Boundary Conditions

When considering the motion of individual particles, care must be taken to ensure that their new position at each time step is within the allowed solution space - that is, particles should not be positioned outside the confines of the room or area being considered. If particles were to be positioned outside of this area, the algorithm would be wasting computationally expensive

FEs on positions whose values are not strictly useful and which contribute to an increased convergence time, or even an entirely incorrect result.

This boundary limitation can be classed as a hard boundary condition [94], where the positions of the particles are strictly prevented from leaving the search domain. The alternative is a soft boundary condition, where the velocities of the particles are limited such that their magnitudes never exceed a user defined maximum. This has the effect of largely confining the particles to the search space, however it is still possible for them to escape these restrictions.

Several strategies exist for dealing with particles which do exceed the system boundaries [95], and several useful methods are summarised below:

1. Absorbing Walls: Particle velocity in the direction normal to the boundary is set to zero when the particle comes into contact with the wall, and the particle position can be clipped to the boundary. The boundary effectively absorbs the particle energy, and that particle is then set in motion again in further iterations.
2. Reflecting Walls: Particle velocity in the direction normal to the boundary is simply multiplied by minus one, to ensure that particles fly back into the acceptable search space.
3. Wraparound Method: Particles which exceed the boundaries are wrapped around that boundary, and appear within the valid search space on the opposite corresponding of that dimension.
4. Invisible Walls: Particles are allowed to escape the boundary conditions, however if they do, then the objective function is not evaluated for that particle. It is noted in [95] that the rationale behind this is to save computation time for problems where the objective function is a computationally expensive operation. This is the case for speaker localisation, where we are aiming to localise a speaker using a minimal number of FEs.

These are considered in the context of hard boundary conditions, where the particle positions are clipped, in [94], and more elaborate techniques are considered in [96].

5.2 Single User Localisation using Particle Swarm Optimisation

PSO can be used for single source localisation by simply setting the objective function as the SRP. Thus, each particle in the swarm directly explores the received audio power at that point,

and the swarm attempts to converge on a peak in space. The algorithm parameters can then be altered to investigate how they affect performance and accuracy. Because evaluating the SRP over a large number of points is computationally expensive, the number of FEs is used as a performance metric, as before. Similarly, the ALE is used as a performance metric, and it should ideally be as small as possible. Finally, because the PSO localisation method returns a direct estimate of the source position in Cartesian co-ordinates, the result can be tracked using a simple Kalman filter to reduce the mean square error.

5.2.1 Initial Particle Distribution

Because we are only attempting to localise one acoustic source, which may change locations gradually, or - in the case of multiple non-simultaneous speakers - very suddenly, we cannot focus solely on likely updated positions like a particle filter. As we must still search the entire space for a possible change of speaker, we must initialise the search according to the normal PSO paradigm, that is, distributed uniformly over the search space. Of particular interest is how the number of particles used in a swarm affects the average number of FEs used to find a source. It has been noted that a low number of particles yields good results, and it is of interest to observe the behaviour on the SRP. As an objective function, the SRP changes between frames, and is discretised according to the available resolution of the microphone array. Further, an appropriate number of particles to use might also depend on the size of the room being considered.

5.2.2 Effect of Signal to Noise Ratio

In order to thoroughly investigate the robustness of the algorithm to SNR, the simulation environment was used to repeat the experiments using a range of SNR levels, by changing the amplitude of the noise signal which was combined with the raw speech signal. This allowed the algorithm to be evaluated using known SNR levels, rather than the unknown and somewhat variable levels present in the recorded data set.

5.2.3 Height-Estimation Extension

In Chapter 4, prior knowledge of the position of existing speakers was used to reduce the cost of localisation in successive audio frames. Knowledge of speaker height was used to

limit the search space, and a similar idea can be applied to the PSO localisation method. The SRC method drew random samples from within the search space and iteratively contracted that search region until a source was identified. Because PSO doesn't draw random samples at each iteration, another method must be found if height estimation is to be considered.

As described in Section 5.1.3, Shi's variant of the PSO algorithm offers a parameter which controls the extent to which a swarm trades off between a thorough exploration of a smaller local area or a further reaching search of the surrounding area. As such, we seek to restrict the search to a locally intensive search over head height, particularly in an area where that head height has previously been estimated. Further, we wish to return to the more wide ranging configuration in other areas, returning to the more general assumption of searching over most of the height of a room.

5.2.3.1 Inertia Control

To achieve this goal, the inertia control parameter ω_p is modified to affect the different dimensions individually. The scalar ω_p can be replaced with a vector $\boldsymbol{\omega}_p$, with elements equal to ω_p, as shown in Equation (5.8) for a normal 3D search space.

$$\boldsymbol{\omega}_p = \begin{pmatrix} \omega_p \\ \omega_p \\ \omega_p \end{pmatrix} \tag{5.8}$$

The scalar multiplication of the particle velocity at time step i is replaced with a Hadamard product, thus Equation (5.1a) can be rewritten as shown in Equation (5.9). This allows the elements of vector $\boldsymbol{\omega}_p$ to be updated independently, as shown in Equation (5.10), which effectively means that the inertia control is independent for each dimension of the search. Because we expect speakers to be spread around a room, but their height to be relatively similar, we can now favour a more intensive local search in the vertical direction, whilst searching across the entire length and breadth of a room. This is achieved by using a relatively small value for $\omega_{z,\,p}$ compared to those used for $\omega_{x,\,p}$ and $\omega_{y,\,p}$, resulting in a comparatively aggressive local search over the vertical axis, compared to a wide-ranging search over the horizontal plane.

$$\mathbf{V}_i\,(t+1) = \boldsymbol{\omega}_p \circ \mathbf{V}_i\,(t) + c_1 r_1\left[\mathbf{p}_i\,(t) - \mathbf{X}_i\,(t)\right] + c_2 r_2\left[\mathbf{p}_g\,(t) - \mathbf{X}_i\,(t)\right] \tag{5.9}$$

$$\boldsymbol{\omega}_p = \begin{pmatrix} \omega_{x,\,p} \\ \omega_{y,\,p} \\ \omega_{z,\,p} \end{pmatrix} \tag{5.10}$$

5.2.3.2 Adaptive Inertia Control

We can further extend the modification of Shi's PSO algorithm by taking the adaptive inertial weight strategies into account. Instead of using a single set of maximum and minimum inertial weights, these variables are also split into component vectors for each dimension of the search. Equation (5.3) (in the linearly decreasing case) can then be modified as shown in Equation (5.11) to calculate the weight vector $\boldsymbol{\omega}_{p,\,t}$ at each step, where the start weight w_s is replaced by the start weight vector $\mathbf{w}_s$ and the end weight w_e is similarly replaced with the vector $\mathbf{w}_e$. $t_{\max}$ represents the number of iterations over which the inertial weight will be lowered.

$$\boldsymbol{\omega}_{p,\,t} = \begin{pmatrix} \omega_{x,\,p} \\ \omega_{y,\,p} \\ \omega_{z,\,p} \end{pmatrix}_t = (\mathbf{w}_s - \mathbf{w}_e)\,(t_{\max} - t)\,\frac{1}{t_{\max}} + \mathbf{w}_e \tag{5.11}$$

5.2.3.3 Initial Distribution

We also attempt to speed up the search for a speaker by modifying the initial distribution of particles in subsequent audio frames. Because we do not need to concern ourselves so much with a wide-ranging vertical search, we can initialise the particle heights around speakers detected in a previous frame to be non-uniformly distributed. As in Chapter 4, we choose an initial vertical distribution as a weighted mixture of a Normal distribution centred around what is assumed to be head height, and a uniform distribution which encompasses the entirety of the vertical search space. This distribution is described mathematically in Equation (4.2).

5.2.4 Single Source Results

The PSO algorithm was used to perform speaker localisation for a set of stationary single sources on both simulated audio data and a recorded data set, and the effects of various parameter settings were explored. Each variation of the PSO algorithm was run with and without

Parameter	Value
Maximum Iterations	500
c_1	0.002
c_2	0.02
$t_{\max}$	50
g_{thresh}	1×10^{-55}
g_{epochs}	70
$\omega_{s,\,z}$	0.04
$\omega_{s,\,x},\ \omega_{s,\,y}$	0.09
$\omega_{e,\,x},\ \omega_{e,\,y},\ \omega_{e,\,z}$	0.01

Table 5.1: Key PSO Parameters

a modified initial distribution, over a range of swarm sizes. The effect of varying SNR was also investigated. Note that the data sets used were those same data sets used in Chapter 4, to ensure a fair comparison on the single speaker ASLT task. Table 5.1 shows the algorithm parameter settings for the parameters which were not varied.

5.2.4.1 Comparison of Particle Swarm Optimisation Variants

Initially, four PSO variants were compared on the simulated data set of the room setup described in Section 4.4. The graphs presented refer to the results obtained using the second speaker as the target, and the lack of difference between results for each speaker on the recorded data set prompted further investigation using known SNRs on the simulated data set. The results were expected to differ, as previous work indicated a change in the computational requirements which varied inversely with the signal SNR. The SNR was designed to be lower for each successive speaker, as each was positioned progressively further from the microphone array, with the inverse square law affecting the received signal at each microphone.

For the first experiments, the Trelea types 1 and 2 were used, as well as Clerc's variant, and the decreasing inertial weight method. The average accuracy, filtered accuracy, and the number of FEs required to complete the search on the simulated data set are shown in Figures 5.1 to 5.3 respectively. The experiments were all set up with a maximum number of PSO iterations of 500, which was found to be more than enough, but meant that there was a reasonable cap to the amount of processing which could possible be done on each audio frame. These initial experiments were run using the reflecting walls boundary conditions.

The size of the particle swarm affects the number of FEs used to arrive at a result because for

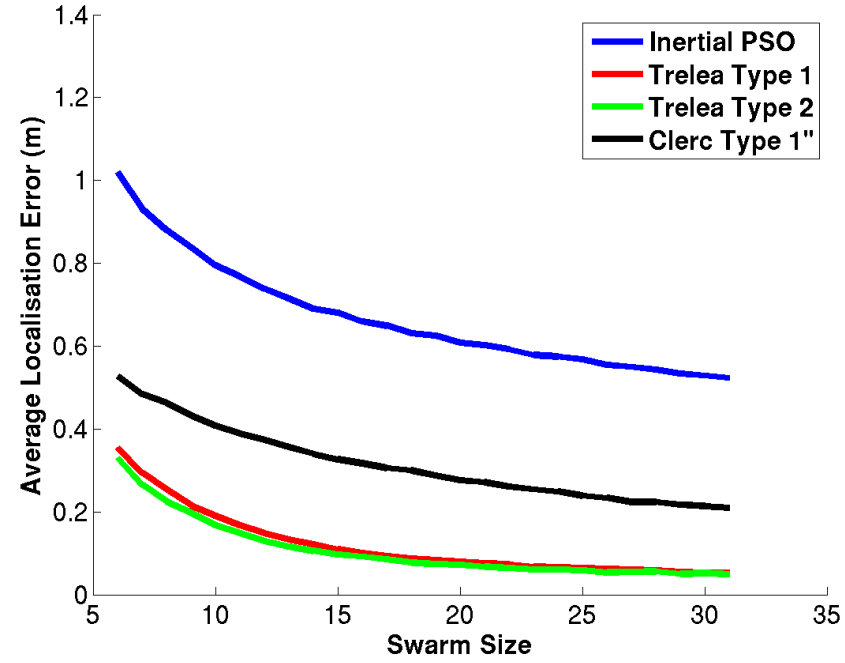


Figure 5.1: ALE vs Particle Swarm Size for multiple PSO variants on simulated data, showing the decrease in average error as the swarm size is increased.

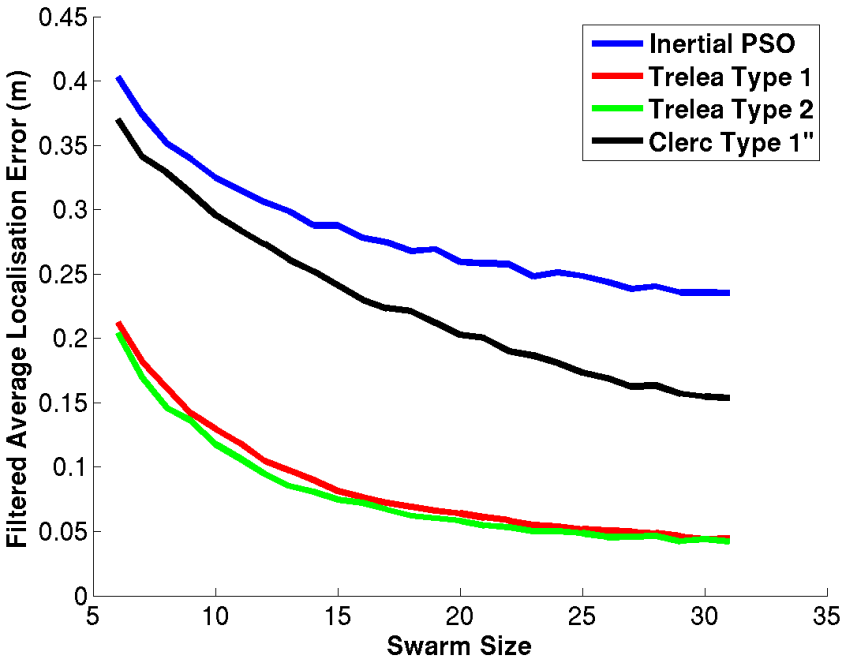


Figure 5.2: Filtered ALE vs Particle Swarm Size on simulated data, showing the decrease in average error as the swarm size is increased, with overall lower levels of error compared to the unfiltered case.

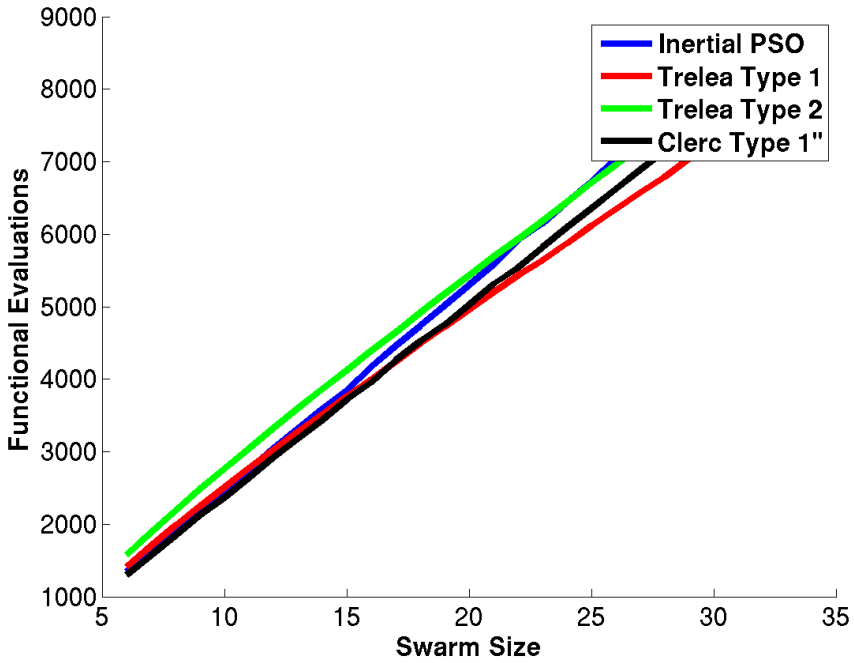


Figure 5.3: FEs vs Particle Swarm Size for multiple PSO variants on simulated data, showing the approximately linear increase required as the swarm size is increased.

a larger swarm size, more particles must be evaluated before any particle is allowed to change location. Whilst a larger swarm means more of the FE operations could theoretically be done in parallel, more work will be done overall. It is therefore of interest to plot the number of PSO iterations (also referred to as epochs) used as the size of the swarm is increased, and this is shown in Figure 5.4.

The filtered ALE shown in Figure 5.2 is much lower than the raw ALE, and this is not unreasonable. Despite the lack of interfering sources, the optimisation algorithm can still become stuck at local optima, even if the value of the objective function isn't as large as the value at the true target position. This causes incorrect localisation results, placed anywhere over the horizontal-plane, to occasionally be returned. This cannot be avoided, as there are occasionally audio frames where a speaker has paused between speech segments. This results in no peak at the target location on the objective function, which means that the algorithm converges on some other maximum of the function unrelated to the target of interest. This problem seems to be particularly problematic when there are very few particles in the swarm, and the localisation accuracy increases somewhat when a larger swarm is used. Note however, that the filtering can still deal with these spurious measurements, and that the filtered accuracy reaches a quite

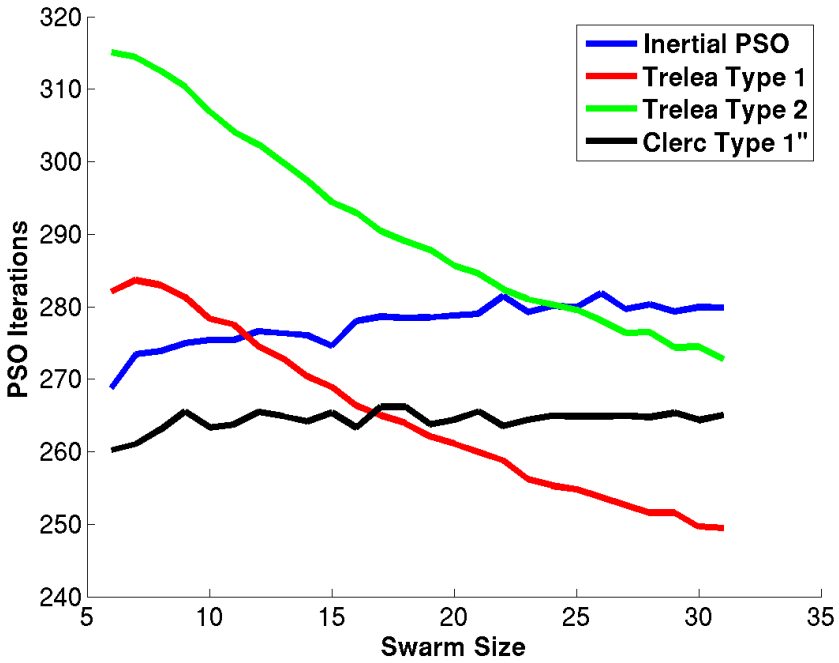


Figure 5.4: PSO Epochs vs Particle Swarm Size for multiple PSO variants on simulated data, showing the differing effect the swarm size has on the number of PSO Epochs required for each variant.

reasonable average error of around 40cm even for very small swarm sizes, in the worst case. The level of accuracy achieved with a small swarm size is in line with the general consensus in particle swarm studies, that conclude that large swarms are not necessary to achieve good results.

Similarly, the number of epochs required for the swarms to converge was almost constant for the Inertial PSO and for the Clerc Type $1''$, whereas the two Trelea variants display a decreasing Epochs requirement as the swarm size is increased on the simulated data set. The number of FEs are directly related to the swarm size, so whilst there are differences in the number of PSO iterations required, the number of FEs required is still strongly dependent on the swarm size. As such, the results suggest that there is no computational advantage to increasing the swarm size, and only marginal accuracy advantage to be gained.

Finally, it can be seen that the results for each variant of the PSO algorithm are almost identical in terms of FEs, and this suggests that if this algorithm were to be implemented in an online and real-time system, then the variant with the least computational complexity should be used, as there is no advantage to using ever more complex schemes to control the particles. Note

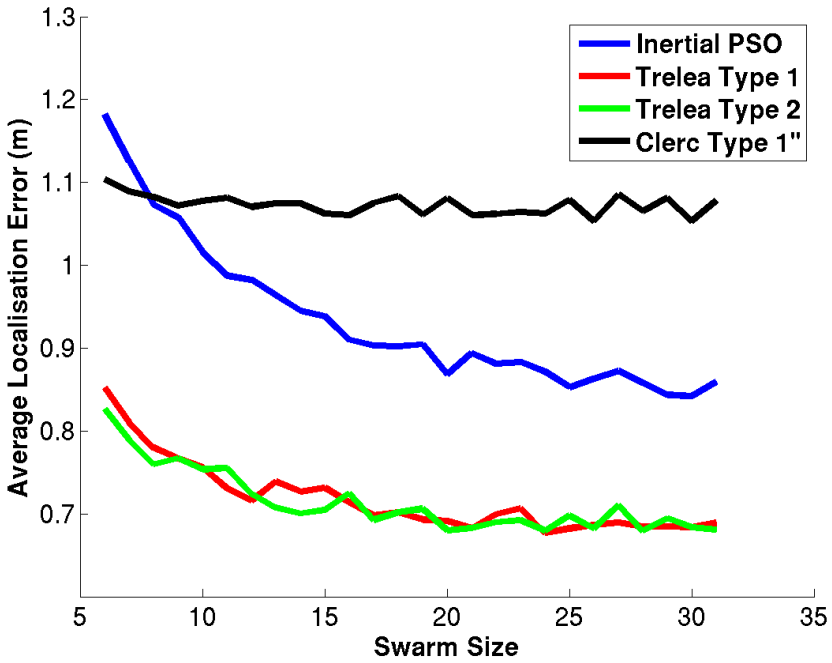


Figure 5.5: ALE vs Particle Swarm Size for Recorded Data, for multiple PSO variants, showing the effect of increasing the swarm size on each variant.

however, that the parameters used in each case were adapted so that the algorithms would run successfully on the data - in particular, numerical constants were scaled such that particles moved within the target room, rather than fly straight out of it on the first iteration!

The same techniques were also tested on data recorded in a room equipped with microphones - specifically, the same recorded data as described in the room setup in Section 4.4. Figure 5.5 shows how the localisation error is increased on the real audio data set, which is largely due to the presence of interfering noise sources at various points throughout the recording. Additionally, the ground truth is not perfectly known for the recorded data, so errors on that set are expected to be larger than errors on the simulated set, where the ground truth is known precisely. The PSO algorithms converge upon optima each frame, and where there are large errors, the SRP is generally higher than that at the target location - that is, the optimisation routine is finding a global maximum, but that maximum does not necessarily correspond to the source that is being targeted. On the recorded data, the decreasing Epochs requirement of the Trelea variants is not nearly so apparent.

Although there is an overall increase in localisation error when run on recorded data (which was

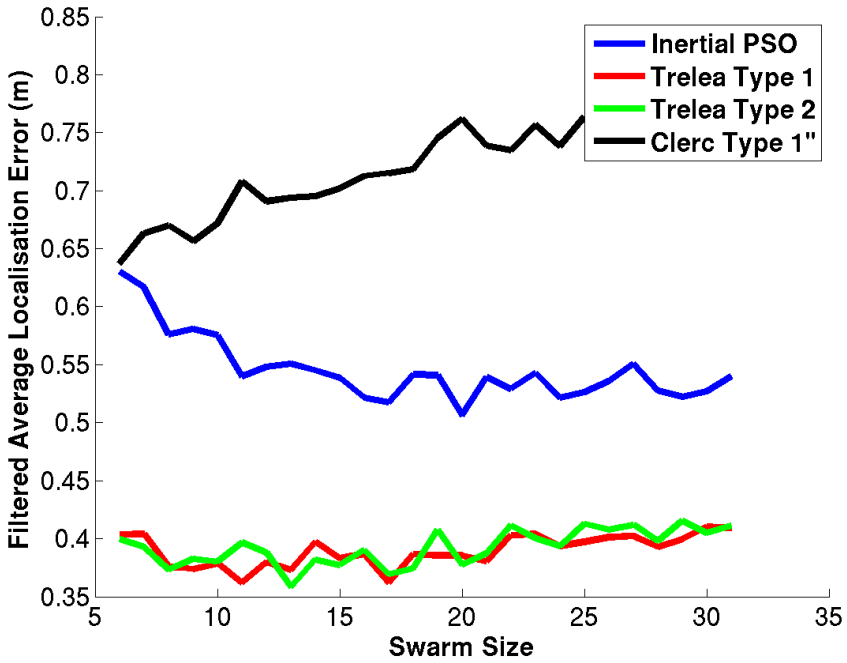


Figure 5.6: Filtered ALE vs Particle Swarm Size for Recorded Datafor multiple PSO variants, showing the effect of increasing the swarm size on each variant, with an overall lower level of error compared to the unfiltered results.

captured in a highly reverberant environment!), the time taken for the algorithms to converge did not suffer greatly. As shown in Figure 5.7, the number of PSO epochs required is about the same as in the simulated data case. On the recorded set, the ALE was at a consistent level for all of the newly developed PSO variants.

5.2.4.2 Effect of Boundary Strategies

The experiments were repeated to study the effect of changing the boundary conditions of the search. In particular, this was expected to affect the number of FEs required in each case, as the different boundary strategies can, for example, avoid FEs when the particle in question is outside of the target area. Figure 5.8 shows the average number of FEs required for each PSO variant with different boundary conditions applied. The figure shows the results of the experiment applied to the Trelea Type 1 variant of the algorithm.

There is no clear advantage to using any one of these boundary conditions over the others, as neither accuracy nor convergence time is noticeably affected. This is conceivably because the

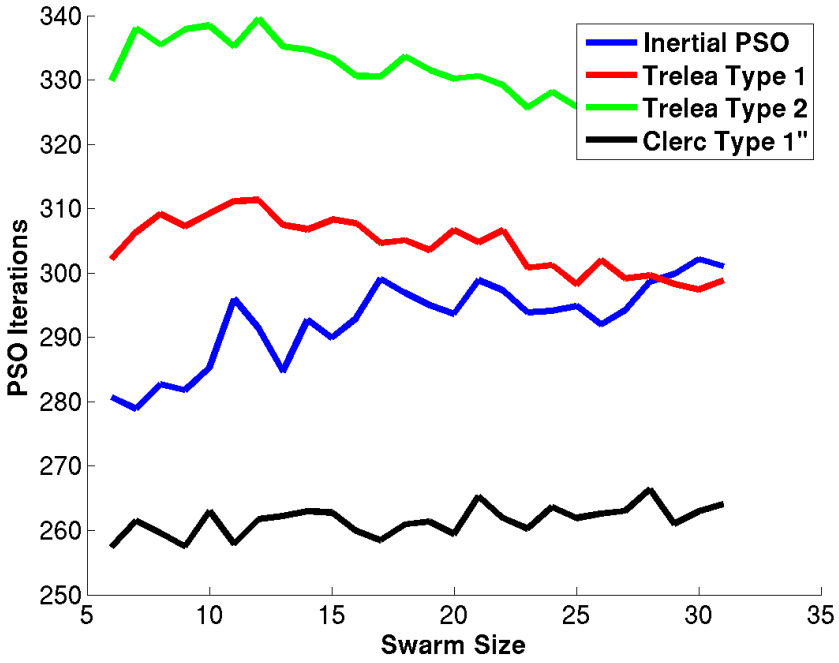


Figure 5.7: PSO Epochs vs Particle Swarm Size for Recorded Data, for multiple PSO variations.

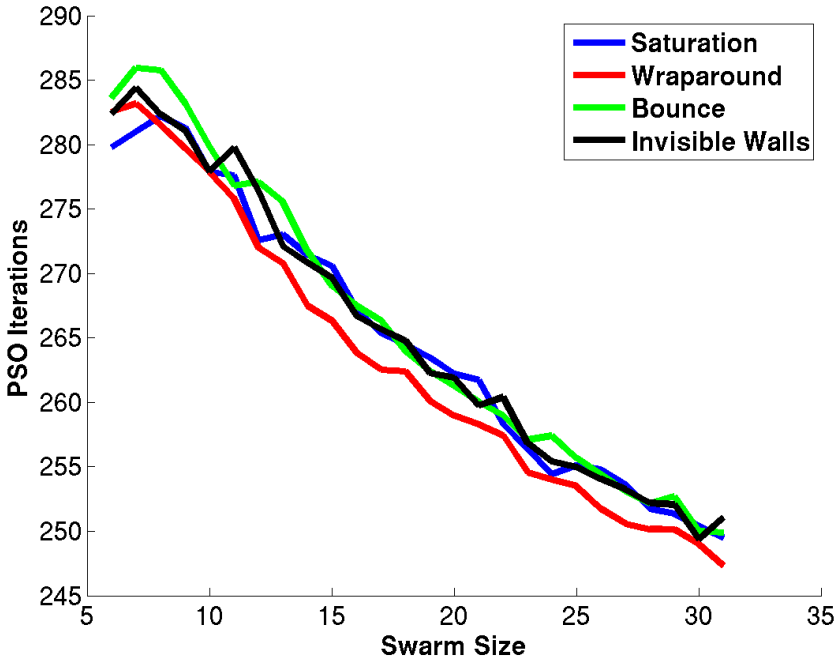


Figure 5.8: PSO Iterations vs Particle Swarm Size for a multiple boundary conditions using the Trelea Type 1 PSO variant, showing the decreasing number of iterations required as the swarm size is increased, regardless of the boundary condition used.

sources being studied aren't close to the boundaries of the room, and therefore the boundary conditions are rarely evoked by any significant number of particles. It might therefore be simplest to choose the invisible walls boundary conditions, as it might marginally save some effort when localising a source near a wall. Again, considering sources near walls, the wraparound approach may be inappropriate, as it might cause a search near convergence to start reinvestigating the opposite side of the room.

5.2.4.3 Exploring Height Estimation

In order to compare the height-estimated extensions competitively with the pre-existing PSO variants, each test was run with the same parameters as in Section 5.2.4.1. Firstly, the effect of varying the initial particle distribution was studied against Shi's linearly decreasing inertial weight PSO. Secondly, the per-dimension inertial control technique was run on the same data. For these experiments, the invisible wall boundary condition was used on both recorded and simulated data.

As before, each variant was considered over a range of particle swarm sizes, and the performance in terms of the number of FEs and the ALE recorded. Figures 5.9 and 5.11 graph the ALE and the number of FEs used, respectively, against the size of the particle swarms.

Simply initialising the swarms with a height biased towards what is approximately head-height, based on the results from previous frames, has a small negative effect on the localisation accuracy on the simulated data. In contrast, using the Hadamard product has little effect on the ALE, as expected, and the results are in-line with running the standard Inertial PSO variant. Both techniques lead to a reduction in the number of number of PSO epochs required for convergence, with the Hadamard product having a less pronounced effect than the other two new variants. Using a combination of the two methods does not produce the expected combined effect of better accuracy with lower computational complexity, suggesting that biased initialisation (height-initialised) is the most useful technique.

Finally, the same techniques were used on the recorded data set, with similar results. As shown in Figure 5.13, the localisation error is consistent with the Inertial PSO variant across all new variants of the algorithm, rather than being slightly worse, as in the case of the simulated data. For the number of epochs required, Figure 5.14 shows that both techniques decrease this number, as on the simulated data. The lack of improvement in the localisation error can be

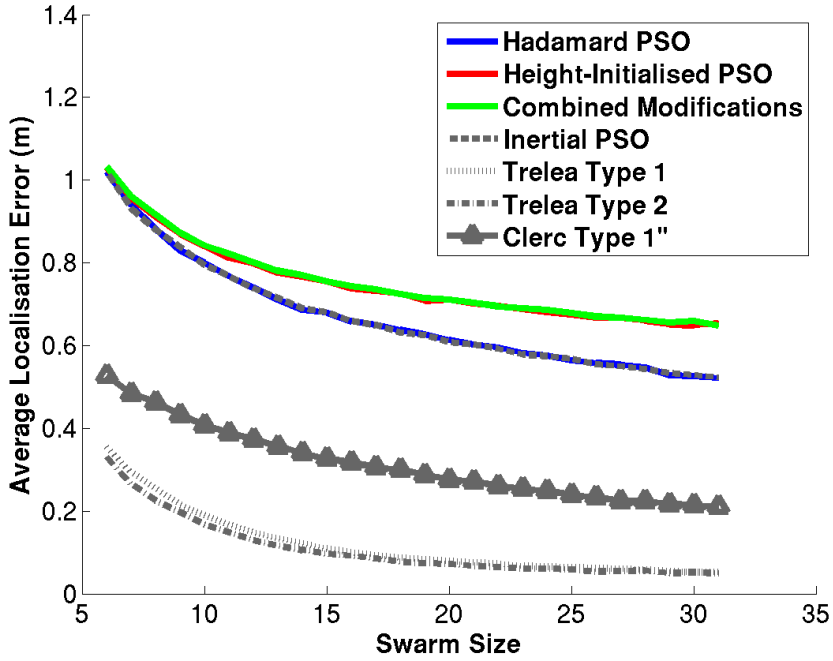


Figure 5.9: ALE vs Particle Swarm Size on simulated data, showing modified PSO variants.

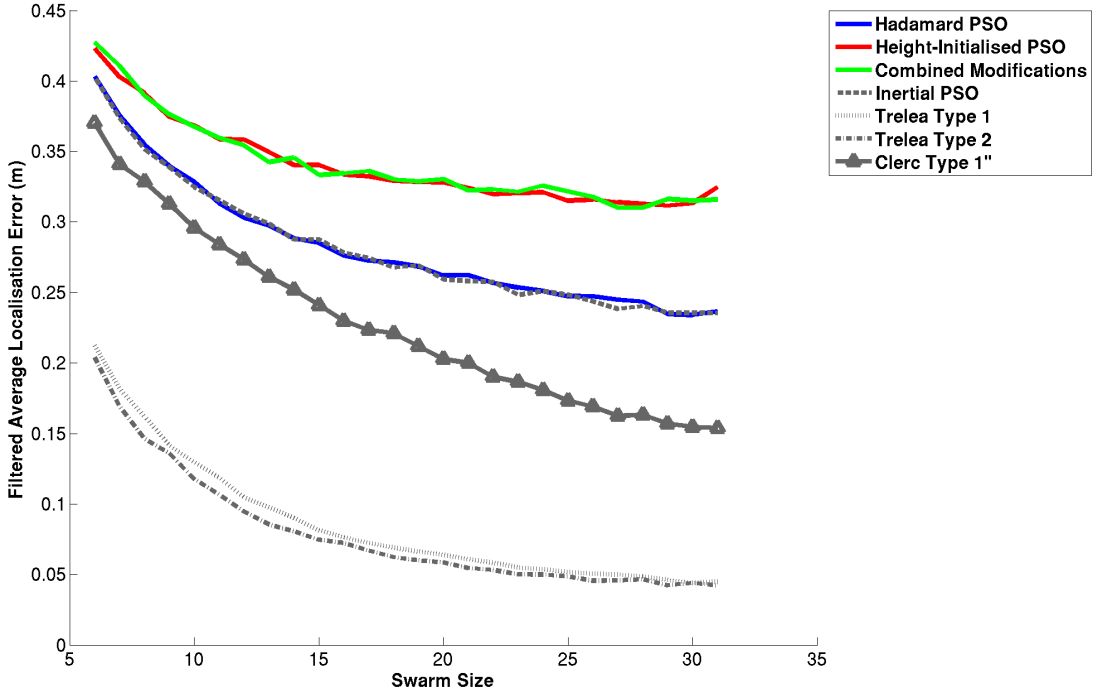


Figure 5.10: Filtered ALE vs Particle Swarm Size on simulated data, showing modified PSO variants.

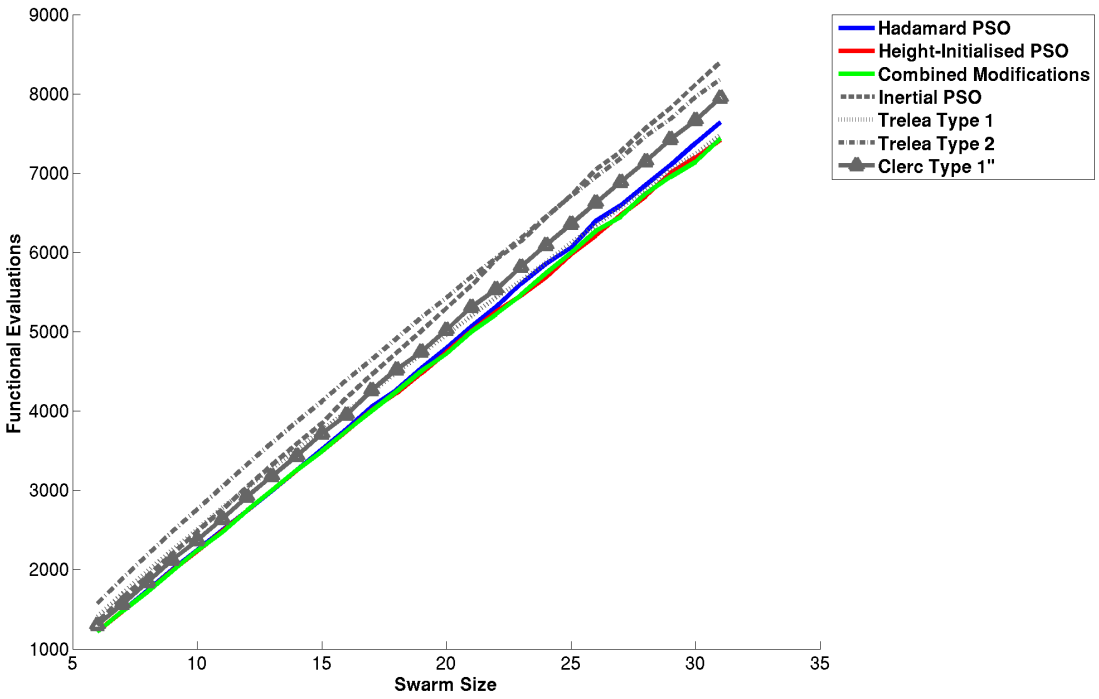


Figure 5.11: FEs vs Particle Swarm Size on simulated data, showing modified PSO variants.

attributed to the modifications failing to help the swarms avoid interfering sources which had a larger SRP value than the desired target. As might be expected, given that neither variant had as good an improvement on the recorded data as on the simulated data, the combined initialisation and Hadamard product variant did not provide any further advantage.

5.2.4.4 Robustness to Noise

As in Chapter 4, the algorithm was run over 4 separate speakers in order to gain an insight into the robustness to increasing SNR as the speech signal comes from a source further and further away from the microphone array. This yielded no discernible effect, and so the simulated data was modified to run simulations over a wide range of artificial white noise power levels. The noise was added to each microphone signal as before, and the performance of the algorithm measured as the SNR was changed from a negative to a positive value.

Figure 5.15 shows the effect of SNR on the Hadamard variant, for various swarm sizes. Figure 5.16 shows the related measure, the number of Epochs required. Whilst there is a sharp

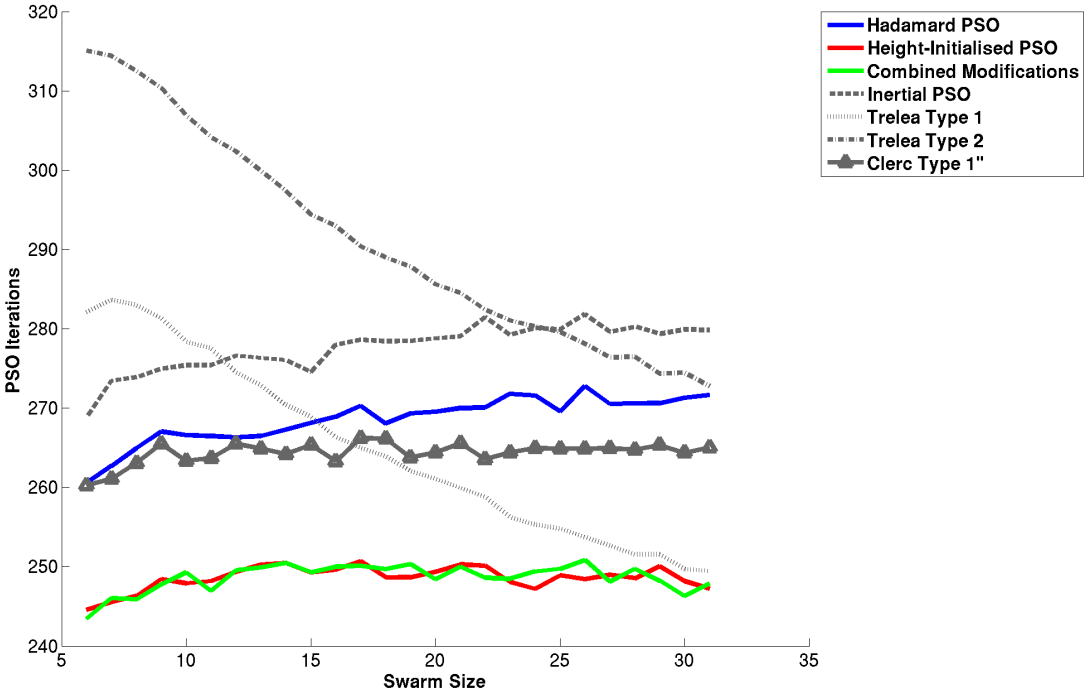


Figure 5.12: PSO Epochs vs Particle Swarm Size on simulated data, showing modified PSO variants.

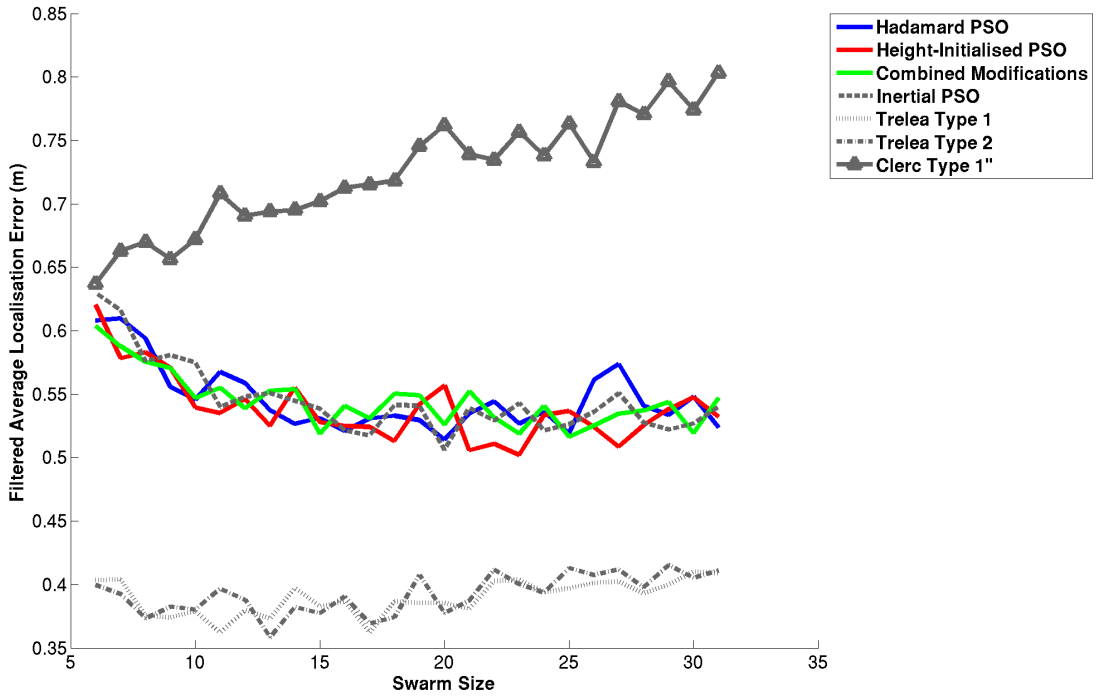


Figure 5.13: Filtered ALE vs Particle Swarm Size on recorded data, showing modified PSO variants.

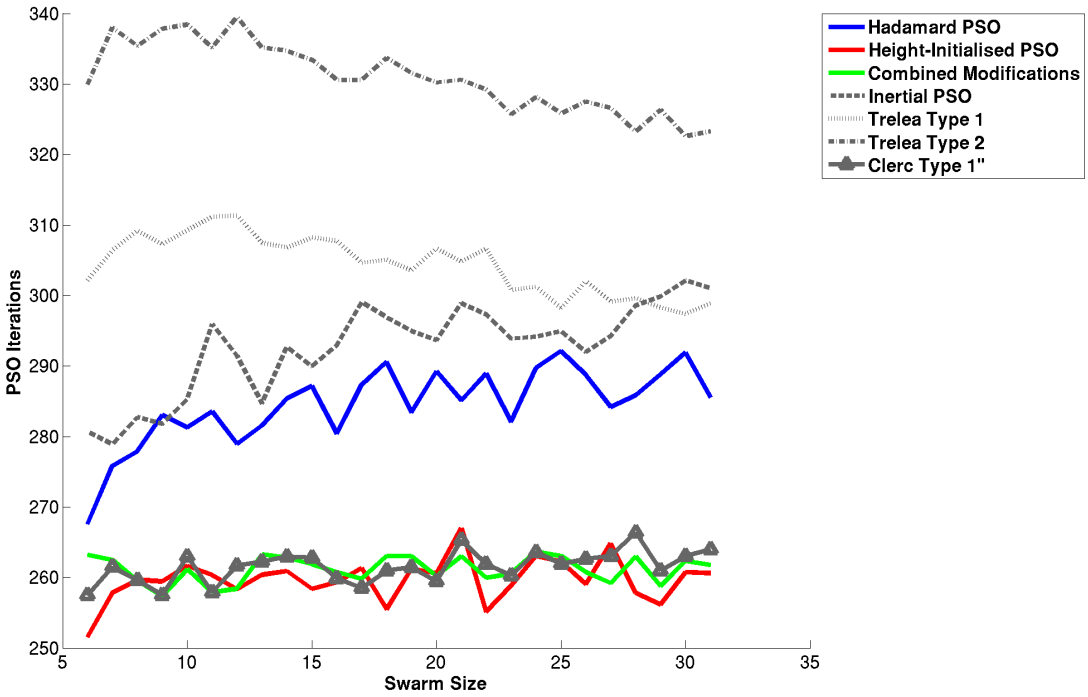


Figure 5.14: PSO Epochs vs Particle Swarm Size on recorded data, showing modified PSO variants.

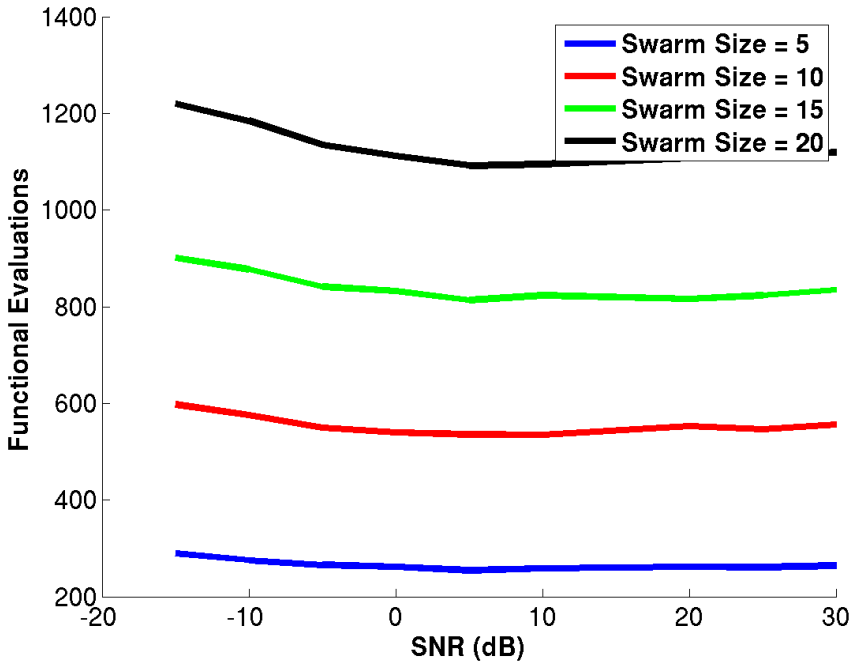


Figure 5.15: Functional Evaluations vs SNR for the Hadamard PSO variant, also showing the effect of swarm size.

decline in the required Epochs as the SNR increases, this trend is not so apparent when the number of FEs is considered. This pattern repeats itself over each of the PSO variants considered.

Finally, the localisation error is graphed against changing SNR in Figure 5.17. There is a clear trend in all but the smallest of swarm sizes, of a small increase in localisation error for negative SNRs, which is to be expected because a peak in the SRP will become less distinct as the SNR decreases. This suggests that the underlying objective function, the SRP, is itself fairly robust to acoustic noise, and consistently produces a peak at a speaker location which can be localised by the PSO technique. Good performance over a range of SNRs is not unreasonable for this technique, as it was chosen specifically for its ability to cope with noisy, non-smooth objective functions, on which gradient-based optimisation functions are expected to perform badly.

5.3 Conclusions

Compared to SRC, use of the PSO family of algorithms results in a significant reduction in the number of FEs required to perform the localisation task. This result is generally scalable by

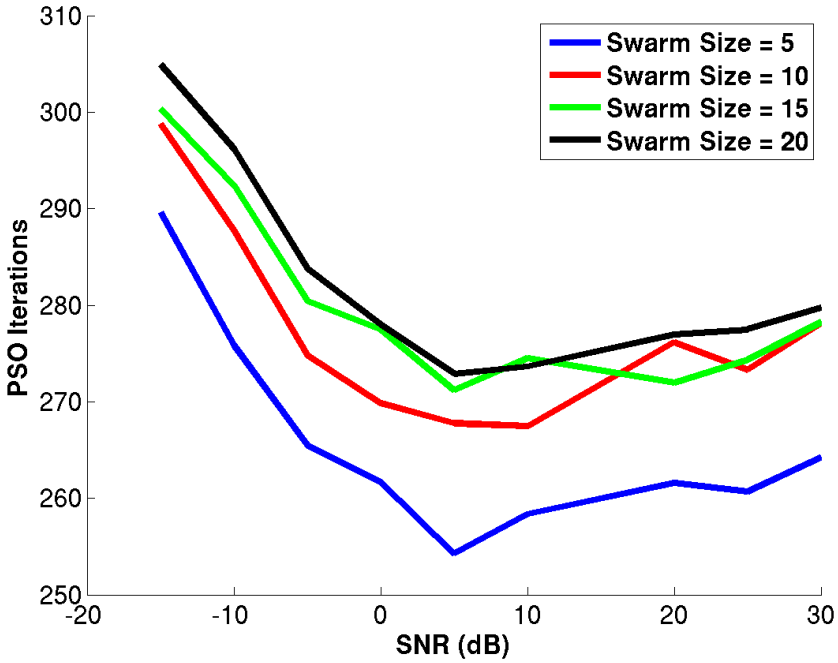


Figure 5.16: PSO Epochs vs SNR for the Hadamard PSO variant, also showing the effect of swarm size.

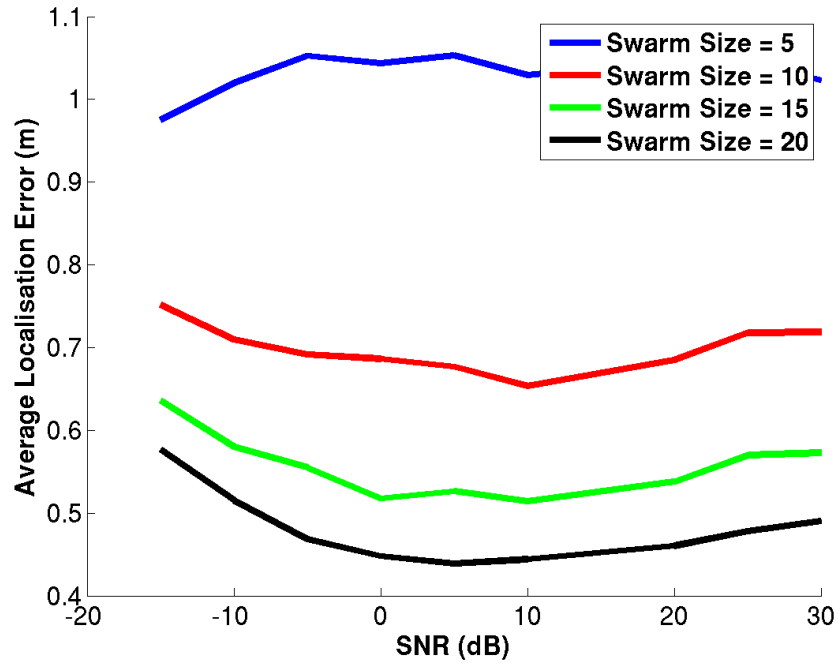


Figure 5.17: ALE vs SNR for the Hadamard PSO variant, also showing the effect of swarm size.

altering the PSO swarm size, such that when more FEs are used, a lower localisation error is achieved, with diminishing returns. The localisation error itself is generally on par with the SRC technique for medium sized swarms, and, after filtering with a Kalman filter, still acceptable even with small swarms consisting of only 5 particles.

As well as proving the suitability of using the SRP as an objective function for optimisation using PSO, this chapter has characterised the response of such a system over different PSO algorithm variants, boundary conditions, swarm sizes and SNRs. The general technique is robust to low SNR environments and provides a consistently low computational load.

Finally, modifications were made to the standard PSO algorithms in an attempt to adapt the technique to the practicalities of an acoustic search environment. The primary advantage of these techniques is maintaining an acceptable error level, whilst performing well with a very small swarm size. This significantly reduces the computational load required to perform the speaker localisation task. In particular, the number of FEs required can be consistently brought down to the order of low thousands, as opposed to the high tens or even hundreds of thousands required by the SRC methods.

Chapter 6
Multi-source Particle Swarm Optimisation Localisation

This section will discuss the use of multi-swarm PSO to find multiple peaks of the SRP objective function, which we expect to correspond to multiple simultaneous speakers. The premise of this technique is to extend the behaviour of a single swarm of particles to include multiple, potentially interacting, particle swarms on the same objective function. Note that this is distinct from multi-objective optimisation, where there are multiple objective functions and a Pareto optimum must be found.

6.1 Multi-Optima Particle Swarm Optimisation Variants

The lowering of the computational complexity for single source localisation is particularly useful as it provides an opportunity to utilise similar techniques for locating multiple speakers, with the aim of only performing as many FEs as might previously have been required to locate a single source. The SRP objective function is already well suited for multiple source localisation, and there exist PSO techniques which attempt to find multiple peaks of an objective function.

There exist many multiple-optima optimising techniques, several of which are studied in this section. The nature of the fitness/objective function - namely its noisiness and the absence of a symbolically differentiable formula limit the optimisation methods available. Methods based on particle swarms are of course of particular interest.

This section presents several swarm-based algorithms which are then applied to the problem of multiple source localisation. As before, the localisation accuracy of each technique is an important metric, as is the computational load required to perform the localisation task. The accuracy metric in particular must be carefully considered in this case, as the comparison is now between two sets - the set of known speaker locations and the set of observations - rather than between a single observation and a known lone speaker location.

6.1.1 Niching Particle Swarm Optimisation

Niching methods are an approach to multiple solution finding in genetic algorithms [97]. These methods have been applied to PSO [98] to search for multiple solutions on general functions containing multiple optima, and niching methods can be applied to find multiple solutions in parallel or sequentially. The Niching PSO makes use of two variant of the general PSO algorithm; the guaranteed convergence particle swarm optimiser (GCPSO) and the cognition-only model.

The cognition-only PSO variant [99] simply modifies the velocity update step, as shown in Equation (6.1), to allow each particle to consider its own best position, with no swarm interaction via the global best. This limits the reach of individual particles across the search space, so each particle considers only its own local space.

$$V_i\left(z+1\right) = \omega_p V_i\left(z\right) + c_1 r_1\left[\mathbf{p}_i\left(z\right) - x_i\left(z\right)\right] \tag{6.1}$$

The GCPSO algorithm [100, 101] modifies the speed update as shown in Equation (6.2), but only for the globally best particle at each epoch, indexed by τ. These equations represent an attempt to solve a limitation of PSO in that when two particles both arrive at the current best position at time step z, their velocity update depends entirely on the inertial term. This is problematic as it means convergence isn't guaranteed, as particles can stop moving and converge on a position which is only the best seen so far, rather than a global optimum. As such, the search can halt prematurely, and the problem is known as stagnation. The GCPSO provides guaranteed convergence onto a local optimum.

$$V_\tau\left(z+1\right) = -X_\tau\left(z\right) + \hat{y}\left(z\right) + \omega V_\tau\left(z\right) + \rho\left(z\right)\left(1-2r_2\right) \tag{6.2}$$

This update equation makes use of $\hat{y}\left(z\right)$, which is the globally best recorded particle position, to ensure that at least one particle keeps moving until an optimum has been found. The algorithm is forced into performing a local search using an area surrounding the global best, and the radius of this search is controlled by the function $\rho\left(z\right)$, defined in Equation (6.3).

$$\rho(z+1) = \begin{cases} 2\rho(z) & \text{if } \#\text{successes} > s_c \\ \frac{1}{2}\rho(z) & \text{if } \#\text{failures} > f_c \\ \rho(z) & \text{otherwise} \end{cases} \tag{6.3}$$

As this is a recurrence relation, an initial value, $\rho(0)$ is required, and this can simply be set to 1 [101]. In this context, a failure is defined as occurring when the value of the objective function at the global best position $\hat{y}(z)$ is the same as the objective function value at $\hat{y}(z-1)$. The number of successes denotes the number of *consecutive* successes, and the number of failures is the number of *consecutive* failures. As such, when either of these counters is incremented, the other must be reset to 0. The parameters s_c and f_c are threshold parameters to the algorithm, and their values must be set appropriately for the objective function, determined by experimentation.

The steps of the NichePSO are given in Algorithm 5 [98]. Particles are initialised at random positions drawn uniformly across the search space, and each sub-swarm S_j, indexed by j is given a radius R_j as defined in Equation (6.4). In this equation, g is the index of the best particle in the sub-swarm, and the sub-swarm $S_{j,i}$ represents each of the particles (indexed by i) in the set S_j, excluding the particle $i = g$.

$$R_j = \max\{\| S_{j,g} - S_{j,i} \|\} \tag{6.4}$$

Initialise main swarm
Update main swarm particle positions using one iteration of cognition-only PSO
Evaluate the objective function for each particle in the main swarm
for Each Sub-swarm **do**
    Update sub-swarm particle positions using one iteration of GCPSO
    Evaluate the objective function for each particle in the sub-swarm
    Update sub-swarm radius
end for
Merge sub-swarms if possible
Allow particles in main swarm to be absorbed into main swarms they move into
Identify particles in main swarm which can be partitioned, and create new sub-swarms based on those particles and each of their nearest neighbours

Algorithm 5: NichePSO Algorithm

With the sub-swarm radius defined, the conditions required for sub-swarms to merge can be

given. Swarms that intersect are liable to converge upon the same optima, and therefore intersecting swarms can be merged together. Equations (6.5a) and (6.5b) form the two conditions for sub-swarms $S_{j1,\,g}$ and $S_{j2,\,g}$, with radii R_{j1} and R_{j2} respectively, merging.

$$\| S_{j1,\,g} - S_{j2,\,g} \| < (R_{j1} + R_{j2}) \tag{6.5a}$$

$$\| S_{j1,\,g} - S_{j2,\,g} \| < \mu_s \tag{6.5b}$$

Equation (6.5b) is required for sub-swarms that have converged on the same optimal solution, and both have a radius of 0. Thus, if $R_{j1} = R_{j2} = 0$, then Equation (6.5b) can still be used to merge two swarms given some small threshold, μ_s. Similar to merging sub-swarms, particles from the main swarm can be merged when, for particle $\mathbf{x}_i$, the condition described by Equation (6.6) is met.

$$\| \mathbf{x}_i - S_{j,\,g} \| < R_j \tag{6.6}$$

Finally, the niching part of NichePSO is the part of the algorithm where the main swarm of particles is partitioned, with the creation of new sub-swarms. There are many niching methods available, such as a sequential niching technique [102] which de-rates the objective function in areas where the algorithm converges. [97] details many different approaches, however the method taken by NichePSO is to extend previous work [103], which thresholds the objective function values. For values over the threshold, the particle at that position is removed from the swarm and that position is labelled as a solution. The objective function is then, 'stretched' to stop other particles exploring the area. If the objective function value is less than the threshold, then more particles are added to explore that area more closely.

NichePSO modifies this method as it recognises that the threshold value, denoted ϵ_n, is dependent on the objective function used which therefore requires tuning. To avoid this, the change in the fitness of each particle is monitored such that new sub-swarms are created when there is very little change over a small number of iterations. The variance σ_j of particle j's fitness is kept over a number e_σ of iterations, and this is compared to a threshold ϵ_t. If the variance is less than the threshold, then a new sub-swarm is created using the particle in question and its nearest neighbour in the Euclidean sense from the main swarm.

6.1.2 Wave of Swarm Particles (WoSP)

The waves of swarm particles (WoSP) [104, 105] extends the standard PSO algorithm to find multiple optima by allowing the particles to undergo a cycle of behaviour known as, 'converge disperse'. This technique allows a single swarm to converge upon an optimum value, and then disperse away from that area to search for another local optimum position. This ejection is achieved by means of a strong short range attractive force between particles which causes particles to fly past each other when they get close, due to the discrete-time updates used. As this is most likely to occur at optima of the objective function, particles converging at these points must be made to not return to that optima via the normal particle swarm dynamics.

To achieve this, each particle is assigned a wave number which is incremented when it undergoes an ejection event, a process which is also referred to as promotion. Further, each particle only responds to global and local best results from other particles within the same wave. Each particle keeps a history of locations from where it has been ejected. If a particle comes within a certain distance - the search scale parameter of the algorithm - then the particle is unable to 'report' for that iteration. In this context, being unable to report means that the particle's velocity is not allowed to update normally, rather, it obeys the update rule shown in Equation (6.7). In these equations, $\hat{P}_c$ is a unit vector in the direction of the closest previous promotion point, and $\hat{P}_t$ is a unit vector in the direction of the smallest component of $V_i(z)$.

$$V_i(z+1) = V_i(z) - c_1\hat{P}_c - c_2\hat{P}_t \tag{6.7}$$

For particles which are allowed to report, the velocity update step is updated as shown in Equation (6.8), which introduces an extra velocity term, V_{SRF}.

$$V_i(z+1) = \omega_p V_i(z) + c_1 r_1 \left[\mathbf{p}_i(z) - x_i(z)\right] + c_2 r_2 \left[\mathbf{p}_g(z) - x_i(z)\right] + V_{\mathrm{SRF}} \tag{6.8}$$

Each component of V_{SRF} is calculated as the sum of all the short range forces along that axis that a particle experiences due to every other particle. Each i^{th} component of this sum is of the form given in Equation (6.9), where the S_f is a factor which determines the magnitude of the short range force; S_P is a parameter which determines how quickly the force changes with the distance D between the particle being updated and the i^{th} other particle. Finally, V^i is the distance between the two particles in the dimension of the component of V_{SRF} being calculated.

$$\mathrm{SRF}^i = S_f * \frac{V^i}{D^{S_P}} \tag{6.9}$$

Note that this search scale parameter is particularly relevant to the case of source localisation, as it might be beneficial to define an area of personal space around a person, where another source is unlikely to be found. The complete WoSP algorithm is given in Algorithm 6 [104].

Initialise all particles to random positions and speeds
Set all particles to be in wave 0
repeat
  for each particle **do**
    Update particle position
    Evaluate Objective function, update local and wave bests
    Check distance of particle from its promotion points, decide if particle is allowed to report or not
    Update velocity of particle
    Mark particle for promotion if necessary
  end for
  for each particle in descending order of fitness **do**
    if marked for promotion **then**
      Move particle to highest wave, creating new wave if particle is already in the highest wave.
      Add particle's current position to its list of promotion points
      If a wave is left with only one particle, promote that particle too, but do not record the position as a promotion point for that particle.
    end if
  end for
  for each wave in descending order **do**
    Promote particles from lower numbered waves to this wave if their best position has a lower fitness than that of this wave. Do not record promotion points
    When the last particle leaves a wave, a gradient-based local search is done and the result reported as one of the optima found
  end for
until termination conditions met

Algorithm 6: WoSP Algorithm

Finally, it is noted that the algorithm can be terminated when a maximum number of iterations have been completed, or when a minimum number of optima have been found.

6.1.3 Locust Swarms

The Locust Swarm technique is explicitly designed to search for multiple optima, and is derived from the WoSP algorithm [106]. This technique attempts to find multiple optima by making use of 'coarse search, greedy search' tactic, whereby PSO is used to perform a coarse search of an area, before a greedy search technique, 'devours' local areas to find local optima. As each small area is devoured, particles are sent out from that area as, 'scouts' in order to find nearby optima.

The Locust Swarm procedure is introduced in Algorithm 7. The method attempts to address a perceived weakness of the WoSP, in that WoSP sends new particle waves in random directions, when it would be preferable to launch them towards areas which look like they might contain an optimum, but which haven't yet been searched extensively.

Generate R random points
Identify a number S of the best points
Give each point a random velocity
Run PSO for a limited number n of iterations
Optimise using fmincon
for 2 to Number of Swarms, N **do**
    Generate another R points around the previous optimum
    Identify a number S of the best points
    Give each particle a velocity directed away from the previous optimum
    Run PSO for a limited number n of iterations
    Optimise using fmincon
end for
Return best optimum

Algorithm 7: Locust Swarms Algorithm

When generating the R new particles around a previous optimum, their distance δ_l from that point is decided for each dimension of the objective function according to the formula in Equation (6.10), where r_l is a parameter specifying the allowable range of the dimension in question; g_l is a parameter specifying the minimum gap between the original particle and the new ones. The spacing parameter, s_l is then used with the random variable, $u_l \sim \mathcal{N}(0, 1)$ to provide variations.

$$\delta_l = \pm r_l \left(g_l + |u_l s_l|\right) \tag{6.10}$$

To ensure outward exploration away from the previous optimum, the initial speed of the new particles along each dimension is set according to Equation (6.11), where v_l is a velocity scaling parameter, and $u_l \sim \mathcal{U}(0,\, 1)$ is a uniformly distributed variable.

$$V_i(0) = v_l \delta_l + 0.05 \frac{r_l}{2} (2u_l - 1) \tag{6.11}$$

By limiting the number of PSO iterations, the PSO algorithm does not converge, but it provides the standard Matlab 'fmincon' search a local area to start searching. The fmincon function (by default) uses an interior-point method to solve the optimisation problem. This method assumes that the function to be optimised is convex, which, importantly, means that a local optimum is also a global optimum. This is generally not true for acoustic data, as not only are there optima from target sources and interference sources, the SRP is not generally smooth in the area between speakers. However, in close proximity to a speaker, the SRP may well be smooth enough to successfully localise an audio source. If the SRP surface is not smooth, this might manifest itself as many maxima, and so a multi-optima approach should be able to eventually find the true maximum, albeit with more spurious localisations than might otherwise have been recorded.

One final point to note is that the interior-point method used assumes that the function being optimised is continuous. For sampled data, this is not strictly true, and there is a limit to the resolution of the SRP surface, which means that when attempting to optimise a point, locations in space will be so close together that they evaluate on the same grid point. This work investigates the suitability of this method, as even with this limitation, typical available resolution is enough to localise a source to within an acceptable distance. Therefore, careful tuning of the parameters used by fmincon should allow the function to converge upon a point which is accurate enough, before the non-continuity problems arise.

6.2 Particle Swarm Optimisation for Acoustic Multi-source Localisation

It is important that the multi-source localisation schemes described in Section 6 be properly adapted for the speaker localisation problem. These optimisation techniques are often trialled

on high-dimensional problems [107]. The source localisation problem has only three dimensions, so this affects constants which, for example, limit the number of iterations used. Where it might be appropriate to limit to a few thousand iterations of PSO on a 30-dimensional test function in the Locust Swarms algorithm, a few thousand iterations have been shown in Section 5.2.4 to be enough to allow a single-source finding PSO to converge completely. Thus, limiting this parameter further is part of the tuning process when adapting the algorithm to the acoustic source problem.

Similarly, practical points can be taken into account regarding the physical location of possible speakers. It is unlikely (but not impossible) that two sources will be very close, unless they are, say, whispering to each other. This can be taken into account in the WoSP algorithm by changing the scale parameter such that optima are afforded some personal space. Furthermore, as we are not considering the (unusual) situation of one set of microphones attempting to localise over split levels, it is reasonably fair to exclude maxima which only significantly differ in position along the vertical axis, i.e., speakers don't stand on top of each other.

In applying multi-optima techniques to the source localisation problems, we expect to be able to adapt them to the problem appropriately. Several considerations are discussed in this section.

6.2.1 Known Sources and Initialisation

Because multiple optima are to be found, there is some scope to exploit different particle initialisation strategies [108] in order to improve the search performance.

In a speaker tracking system where audio is processed on a frame-by-frame basis, it might make sense to exploit information about sources located in previous audio frames. If these results truly correspond to audio sources which are continuing to emit sound between frames, then the SRP objective function should be very similar at these locations between frames. It might therefore be expected that swarms in subsequent frames should also cluster around these points. Therefore, a reasonable approach might be to initialise the systems used so that they naturally find sources which are present between frames very quickly.

For the Niching PSO algorithm, this simply means initialising a subset of the main swarm particle positions close to previously found optima. The algorithm should then naturally create sub-swarms for any continuing audio sources after a few iterations. The rest of the main swarm should be initialised uniformly across the search space to continue searching for new or

previously undiscovered sources.

In the context of WoSP and locust swarms, particles can be initialised near only one previously known source, as these algorithm find new optima sequentially. If the source still exists in a new audio frame, then it will quickly be found and the algorithms will continue as normal, searching for other audio sources. This only provides limited benefit however, as it only affects the first source to be found. Further, if a source is no longer present, the algorithms are put at some disadvantage.

6.2.2 Avoiding Duplicate Optima

When searching for multiple optima, there is strong chance of swarms reporting the same optima multiple times, especially in sequential techniques where swarms are not aware of each other. This is handled in WoSP through the use of promotion points and restricting when particles are allowed to report their values to the rest of the swarm.

The Locust swarm algorithm handles this requirement by assigning outward velocities from the discovered optimum position to the particles in a subsequent PSO search. This was modelled in Matlab as a particle at the point of the previous optimum which exerted a repulsive force on nearby active particles. This was expressed as an extra velocity component to be added to the standard PSO velocity update equations, the magnitude of which is given in Equation (6.12). The direction of this force is the direction of the vector from the optimal point to the particle in question. This is modelled on an inverse-square law repulsive force system, such that nearby particles are repelled strongly and further away particles feel little effect.

$$v_r = \frac{\alpha_r}{r^2} \tag{6.12}$$

The additional velocity component, v_r is dependent on the distance r between a particle and the repelling position, scaled by an appropriate factor α_r. This factor must be chosen to force the velocities produced by nearby interacting particles to be around the same order of magnitude as the velocities used by the swarm for general exploration. Too small a velocity would result in an ineffective force which does not prevent duplicate results, and too large a resultant velocity would result in particles being pushed away from previous optima they are nowhere near in the context of the room being explored.

Although not explicitly stated in [106], the implementation of Locust swarms for localising more than two speakers requires that all previously discovered optima are avoided, rather than simply the most recently found optimum point. This was simply achieved by extending the repulsive force model to all previously discovered optima, such that particles were forced to move on from the most recently discovered optimum and not allowed to come near to any of the others.

This scheme was implemented in MATLAB by calculating the repulsive velocity for each particle as the sum of all of the repulsive velocities resulting from the interaction of that particle with each previously discovered optimum position. This is simply expressed in Equation (6.13), which can then be added to the regular PSO velocity update equation.

$$\mathbf{v}_r = \sum_{n=1}^{N} \mathbf{v}_{r,\,n} \tag{6.13}$$

This scheme was also used in the Niching PSO implementation. The Niching algorithm was developed with this problem in mind, and it is something which has to be dealt with for general genetic algorithms, on which the Niching method is partially based. One way of dealing with the problem is to 'derate' the objective function around previously discovered optima, although this has been thought to be a flawed solution, as it may create new false maxima. In any case, this was not implemented for the acoustic search, as the repulsive force scheme had already been implemented for Locust swarms and found to be effective, so it was simply ported across as the chosen 'derating' method.

6.2.3 Vertical Axis Restrictions

Whilst the PSO searches employed operate on 3 dimensions, some of the particle interactions are more simply expressed and more appropriate to the problem when implemented to deal with 2 dimensions only. This is particularly relevant to the repulsive force calculations. Because it is assumed that sources cannot be on directly on top of each other, the repulsive force is applied only on the horizontal plane. Particles are therefore not forced down or up from existing optima, rather, they are forced to another part of the horizontal plane. The calculation of the distance between the two points also limits itself to the horizontal plane, such that particles far enough above or below an optimum point to not otherwise be affected are counted as being close. This

forces an emphasis on searching for speakers over the horizontal plane, rather than at different heights when considering one position on the horizontal plane.

Similarly, all swarm management routines - absorbing and merging of sub-swarms - consider only the horizontal-plane component of particles' positions. This has the added advantage of slightly reducing the computational load required when calculating distances.

6.2.4 Limited Resolution

The SRP objective function is fundamentally limited in resolution, and so there comes a point when trying to further narrow down the position of an optimum value of the function becomes impossible. Both the WoSP and Locust swarms algorithms specify that the PSO searches are to be followed by local gradient based searches in the areas of high likelihood for a peak indicated by the output of the PSO searches.

These parts of the algorithm were not implemented for the acoustic search experiments for two reasons. Firstly, PSO techniques already find peaks to a suitable degree of resolution, and a gradient based search would be unable to improve on any true optimum found by PSO. Secondly, gradient based searches are not well-suited for the noisy environment of the SRP objective function.

6.2.5 An Unknown Number of Speakers

Similar to the SRC method, the multi-modal PSO method may fail to converge if there are fewer targets than expected - that is, if the number of sources has been overestimated. Because real tracking systems must estimate the number of speakers, it seems likely that this will occur when processing real signals. As such, it is worthwhile considering when to halt the search for a speaker. If all true sources have been found, and there are no other peaks, then algorithms will no longer necessarily converge quickly if the termination conditions mandate that a certain number of sources be found. In this case, the total number of FEs must also be used to terminate the algorithm.

Spurious localisation results can still occur when the number of speakers has been overestimated, and it is desirable for a localisation algorithm to detect possible spurious results and eliminate them. This could be achieved by the detection of non-speech acoustic sources at

search time, although such a method is considered any further in this work. The algorithms used can also prune the number of results returned by applying a threshold to the SRP at the optima positions. A threshold can be determined by requiring that an optimum position be a certain factor greater than the mean SRP value over all interrogated positions. This effectively filters optima based on the signal to interference plus noise ratio (SINR) levels at their positions, and the value of this is expected to decrease as the number of sources increases, and therefore the SINR decreases. As such, it is of interest to determine how well this scheme scales with the number of speakers.

6.2.6 Iteration Limitation

The total number of FEs available is of course limited by the size of the area being searched and the spatial resolution of the microphone array. This is an impractical limit however, and an acceptable maximum will be much lower than this exhaustive search. Consideration might be given to the number of sources that are expected to be found. The relationship between the number of sources and the number of FEs required to find them all must be considered, as it may not be a linear one. With an idea of this scaling, the number of sources to be found could then also give rise to an expected number of FEs. This could help avoid problems such as early termination due to exceeding too optimistic a limit.

6.2.7 Performance Metrics

The number of FEs is an important metric when considering multiple sources, just as it is in the single source case. Similarly, the ALE is still useful for each individual source, but is complicated by the fact that each optima found must be assigned to a known audio source to evaluate the methods. In our studies, the ground truth is known, but obviously this luxury is unavailable in practical online systems. Also of interest is the number of optima found before matches for real speakers are made. This informs us about the expected number of spurious results, which is useful when considering tracking schemes such as the PHD family of trackers, which explicitly cater for optima appearing which don't correspond to any source.

As before, the number of FEs and algorithm epochs is recorded. In addition, the number of optima found in each case is recorded, allowing calculation of the number of spurious sources found or acoustic sources missed. In order to assign a localisation accuracy score, a measure

known as optimal subpattern assignment (OSPA) is adapted.

6.2.7.1 Optimal Subpattern Assignment (OSPA)

The OSPA metric is designed to be used with multi-optima processes, to detect how close a set of observed positions is to the set of known source positions [109]. This metric overcomes some inconsistencies in other metrics, such as the Hausdorff metric and the optimal mass transfer (OMAT) metric. The Hausdorff metric does not perform well when the cardinalities of the two sets being compared are different, as is in the case of a missed target, or the case of detection of spurious sources. The OMAT metric improves upon this, however it is still sensitive to differing cardinalities, as well as the geometry of the two sets being compared.

There are three practical steps to be taken in order to find the OSPA distance between the two sets being considered, X and Y:

- Find the optimal subset of Y that is closest to X in terms of the p_o^{th} order OMAT metric.
- For each element (j) in Y, assign the variable α_j to a cut-off value, c_c if there is no point in X assigned to it, or to the minimum of c_c and the distance of the element to its assigned point in X, if it exists.
- Calculate the p_o^{th} order average over all α_j

The parameter p_o determines the averaging 'order', and increasing it increases the sensitivity of the metric to outlier points - points in the measurement set which aren't easily assigned to point in the ground-truth set. This effect is tempered somewhat by the cut-off parameter c_c, which assigns a fixed value to points which effectively not assignable, and can be used to trade-off between the metric more heavily penalizing cardinality errors or localisation distance errors. The OSPA metric is defined mathematically in Equation (6.14), with parameters c_c and p_o. Note that m and n are the cardinalities of sets X and Y respectively, and that Equation (6.14) is only valid for $m \leq n$. In the case of $m > n$, $d_{p_o}^{(c_c)}(X,\, Y) \coloneqq d_{p_o}^{(c_c)}(Y,\, X)$.

$$d_{p_o}^{(c_c)}(X,\, Y) = \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} d^{(c_c)}\left(x_i,\, y_{\pi(i)}\right)^{p_o} + c_c^{p_o}\,(n-m) \right) \right)^{\frac{1}{p_o}} \tag{6.14}$$

$d^{(c_c)}\left(x_i,\, y_{\pi(i)}\right)$ is defined in Equation (6.15), and the distance $d\,(x,\, y)$ can be simply defined

as the Euclidean distance between the two points in space, x and y.

$$d^{(c_c)}\left(x_i,\, y_{\pi(i)}\right) = \min\left(c_c,\, d\left(x,\, y\right)\right) \tag{6.15}$$

Finally, note that Π_n is the set of all permutations of the set of integers up to n, so that π represents a permutation on which points in each set are considered.

6.2.7.2 Interpreting Optimal Subpattern Assignment

The OSPA metric on the localisation accuracy is not as simple to interpret as the single source ALE. Where the ALE is reasonably expressed in metres, the OSPA metric cannot be so simply thought of as an error in units of difference of Euclidean distance. The metric deals with differences in distance between nominally associated members of two sets, but it also deals with differences in the cardinalities of the sets. As such, it is instructive to consider how the metric changes when an observation set with known errors is compared to a known ground truth. When using the metric in practice, two parameters must first be chosen; the *order* parameter p_o, and the *cut-off* parameter c_c.

The appropriate choice of these parameters is discussed in [109]. To summarise, the *order* parameter determines how harshly outlier estimates (which are not close to any object in the ground truth) are penalised. The *cut-off* parameter limits this penalty by defining a point where an observation is considered, "unassignable" if it is further than the cut-off from any object within the set of ground-truth elements. By altering c_c, the relative weighting between cardinality errors and localisation errors is controlled. In the special case of $p_o = 1$, the cut-off is the error assigned to a point estimate which is considered to not be associated with any of the elements of the ground-truth set. The case of $p_o = 2$ is also noted as a practical choice, as second order metrics are common [109] and the curves produced are reasonably smooth as the localisation error increases. OSPA metrics with both $p_o = 1$ and $p_o = 2$ are calculated in this thesis.

A value for c_c must be chosen depending on the problem being considered. Values of c_c are roughly categorised as either being small, moderate, or large, and the boundaries depend on the typical localisation errors encountered as well as typical expected distances between the objects being tracked. Broadly, small values of c_c which are close to or less than typical localisation

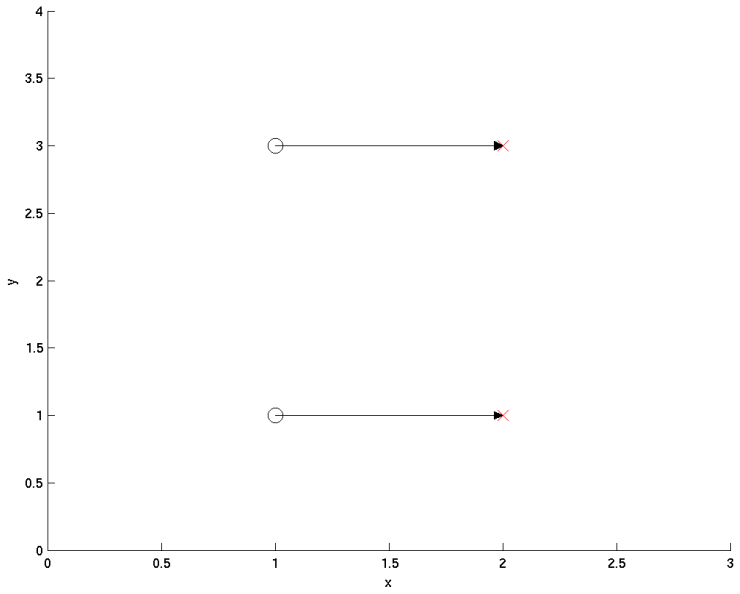


Figure 6.1: Example Data for OSPA Metric - Black Circles are true source positions, arrows show observation positions as they are moved along to their final positions, marked by red crosses.

errors are classified as being small and have the effect of emphasising localisation errors over cardinality errors. Values of c_c corresponding to the maximum distance between observable objects are considered large, and emphasise cardinality errors.

Values of c_c between the typical localisation error distance and the maximal object distance are classed as moderate, and represent a trade-off between penalising either error type too heavily. In this thesis, a value of $c_c = 0.7$ (metres) is used. This is slightly larger than a typical single-source localisation error, but smaller than the maximum distance between sources, which is limited by the size of the room under consideration.

To aid in the interpretation of OSPA results, the metric is calculated for a range of known localisation errors and the chosen parametrisation ($c_c = 0.7$ with $p_o = 1$ and $p_o = 2$) shown in the following Figures. In Figure 6.1, the two elements of the ground truth set are the 2D Cartesian points $(1, 1)$ and $(1, 3)$, marked by black circles. A set of observation are generated, such that the observation of each element is moved progressively further away from their true positions. These are shown in the Figure as arrow, indicating the change in observation up to the final observations marked by red crosses.

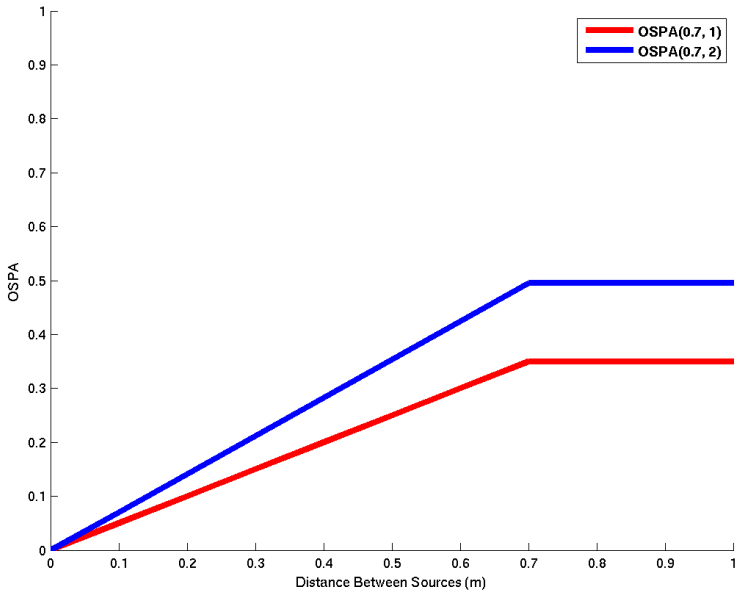


Figure 6.2: OSPA metric for two source observations as the localisation error increases at the same rate for each source.

The OSPA metric is calculated between the ground truth and the observations as the observations move away from the true positions at the same speed. Figure 6.2 shows the difference between the two OSPA parametrisations used, and demonstrates how the metric peaks when the cut-off is reached on the x-axis. Note that in these tests, there is no cardinality error, and that the functions are labelled with their parametrisations as $\mathrm{OSPA}(c_c,\, p_o)$.

Figure 6.3 demonstrates a slightly different situation, where there localisation error of the different sources is not the same. Whilst the observations are the same as in the previous case, one of the ground truth elements has been offset, such that the localisation error for the observation of that source is a constant 0.2m greater than the other localisation error. Figure 6.4 shows how the OSPA metrics change as the localisation error increases, where the x-axis corresponds to the smaller of the two errors. The larger error is a constant 0.2 greater than x at all points along the axis.

6.3 Multiple Source Results

Each of the algorithms was implemented in MATLAB and run on several data sets involving multiple simultaneous speakers. The recorded data set was limited in the number of speakers

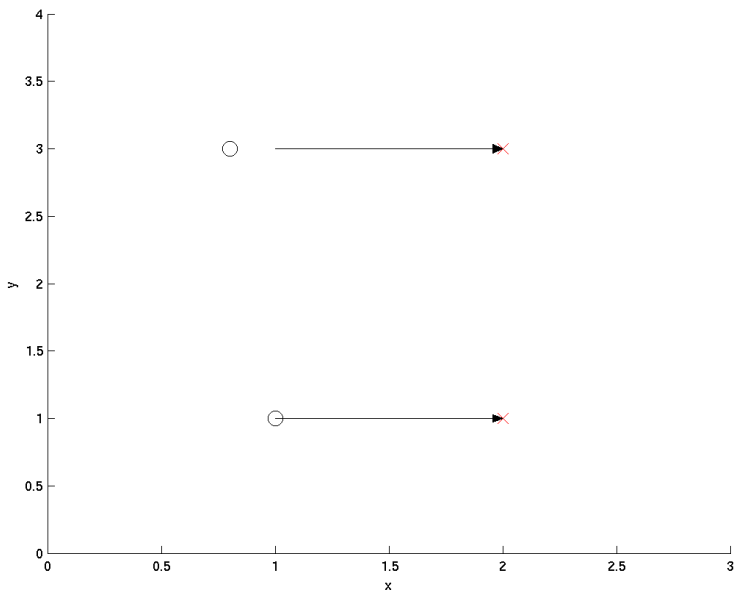


Figure 6.3: Example Data for OSPA Metric - Black Circles are true source positions, arrows show observation positions as they are moved along to their final positions, marked by red crosses.

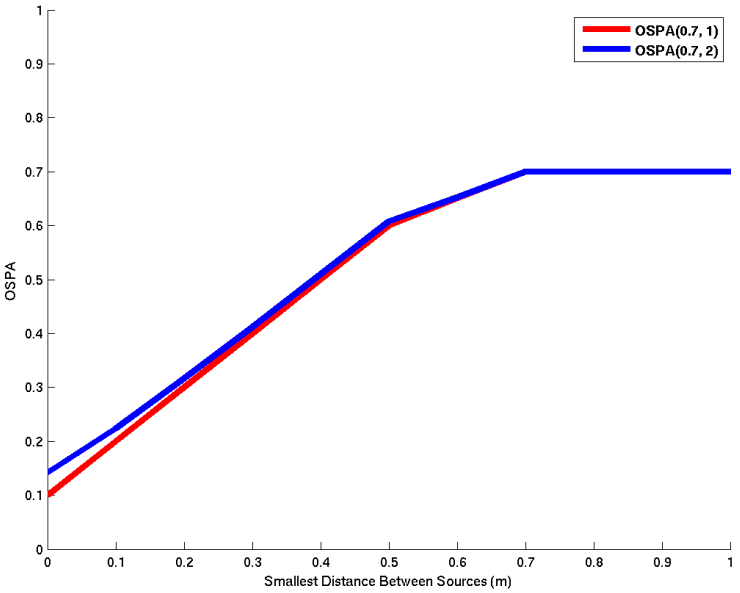


Figure 6.4: OSPA metric for two source observations where the localisation error is different for each source.

available, and only stationary speakers were considered. This data set was the same data used in Chapter 5, except that the source signals were combined to create signals containing multiple speakers in combinations of 2, 3 and 4 simultaneous speakers. All experiments were performed as Monte Carlo simulations, allowing average performance figures to be presented.

The multi-source localisation methods described in Section 6.1 are all capable of being directed to produce a set number of outputs by simply declaring that a completion criteria is to have found a set number of optima. This is simply an extension of the single source localisation case, where the algorithm finishes after finding a single optimum position, and is used to evaluate the multi-optima algorithms' response to various environmental conditions and parameters when dealing with a known number of speakers. In each test performed, the number of speakers is known and the algorithms are tested for their ability to find these known sources, with measurements made of the computational effort involved and the localisation error incurred. Thus, the cardinality error contribution for the OSPA metric is ideally zero for these tests. Note however, that for a tracking framework, the number of sources is not known, and only estimated. Furthermore, a tracking framework must make extra effort to detect new sources, so has to search for more optima than the number of sources which it has tracked. Further consideration of this problem is given in Chapter 7, and this section focusses only on the capabilities of localisation methods.

One of the initial results which came from this work was that the boundary conditions on the PSO searches are much more important in the multiple source case than in the single source case. The invisible walls boundary condition was used initially as it was used successfully with the single source PSO experiments. This occasionally resulted in particles being forced out of the search region when they interacted with the repulsive forces generated by previously discovered optima. Once these particles were ejected, they typically could not re-enter the valid search space due to continued interaction with the repulsive forces. For searches involving multiple optima this situation became more common, often resulting in entire swarms being forced out of the search space. The solution to this problem was to revert to using the wraparound boundary condition, which encouraged further exploration away from known optima. Whilst this may seem counter-intuitive for the single source case, it was previously shown that the choice of boundary condition makes very little difference. As such, allowing the wraparound condition to be used should only have the positive effect of ensuring ejected particles are not permanently excluded in the multiple source case.

6.3.1 Simulated Data Set

Seven simulated speakers were created, and sets of data created with combinations of two up to seven speakers by superimposing the microphone signals. The speakers were simulated in the simple simulation environment described in Section 2.5, with the original speech signals around 30 seconds long. Their positions are shown in Figure 2.14, marked by green squares. Eash speaker is numbered, such that in a set with a given number of active concurrent speakers, the speakers from 1 up to that number are the ones which are active. In addition, two of the speakers (speakers 1 and 3 were simulated moving towards each other at the (slightly high!) constant speed of around $5\mathrm{ms}^{-1}$, which was determined by requiring that the speakers crossed the room in the 30 second time period. The source signals were clean speech sampled at 44.1kHz, and the audio frame size used was the same as used in Chapters 4 and 5.

6.3.2 Wave of Swarm Particles

The WoSP algorithm proved to be hard to tune for the acoustic data, and a tuning which generated consistent results, even across audio frames in a single data set, was not achieved. This was largely due to the noisy nature of the objective function. The algorithm performance was dependent on the small noisy peaks of the objective function, which could cause new waves to be created very regularly, resulting in many optimal positions being reported, most of which were nowhere near the target optima. Furthermore, it was unclear when to stop the algorithm, as peaks were found sequentially, but in no particular order.

This behaviour was inconsistent, however, with some data frames producing optimal estimates very slowly, requiring a large number of FEs without any gain in the localisation accuracy achieved. The difference in behaviour can be tuned somewhat by changing the conditions required for a new wave to be created - the promote factor. However, changing this does not lead to a tuning which is useful over a range different input vectors.

The difficulty encountered in tuning the WoSP algorithm, along with the trouble encountered trying to implement it correctly in MATLAB, meant that it was not considered any further past the initial testing stage. Whilst the algorithm may be suited to the speaker localisation task, the problems encountered in getting it to run consistently mean that this variant is not considered further in this thesis.

6.3.3 Niching Particle Swarm Optimisation

The Niching PSO algorithm encountered much the same issue with the SRP as the objective function as the WoSP algorithm did. The fundamental problem is made quite clear in the context of the Niching algorithm - it is not trivial to decide which positions are suitable for more local (niched) exploration based on the criterion of a local area looking promising as a potential peak of the objective function.

There are several advantages and disadvantages of the Niching technique over the WoSP algorithm. In addition to generating a large amount of clutter in the results, the Niching algorithm has a tendency to either quickly settle on a set of optima, or to return with only one optimum point. By stepping through the code as it progressed, this was found to be caused by the tendency of sub-swarms to balloon in size, merging with other sub-swarms and having an eventual single sub-swarm covering the search area. This problem is particularly prominent when a small number of initial particles are used. When a single particle is marked for the creation of a new sub-swarm, that particle takes with it its nearest neighbour from the main swarm. With a low density of particles across the search area, this can lead to new sub-swarms having a large radius, enveloping many optima caused by both noise and genuine acoustic sources. Because sub-swarms concentrate on a local search, they move comparatively slowly towards local optima, particularly when they encompass multiple peaks. This means that larger sub-swarms do not quickly constrict in size.

This problem is exacerbated by sub-swarms absorbing main swarm particles. Further, even small sub-swarms suffer when they are close to each other, as when they merge, they suddenly encompass multiple optima. Thus, the swarm radius is increased, with the ability to only find one of the local optima. These slightly larger sub-swarms then tend to quickly merge with other sub-swarms until all sub-swarms are merged.

To the Niching algorithm's advantage however, is the consistent computational and timing behaviour. A large main swarm population can be used, resulting in many optima returned effectively in parallel. This allows the search space to be explored uniformly and, importantly, optima found across the search area within a constant time. This is in contrast to the WoSP algorithm, which finds optima sequentially, and is an advantage because it in theory allows the algorithm to be trivially parallelised to a large degree. For example, with a swarm size of 100 particles, which achieves a reasonable particle density for the room sizes considered, each

iteration of the algorithm could be parallelised and most of the sub-swarms would converge (assuming they don't balloon into one sub-swarm) after around the same number of iterations. Because the algorithm operates a parallel search for multiple optima, parallelising the individual FEs for the large swarm means that the algorithm might be kept computationally competitive when compared to a sequential search using the same number, or even significantly fewer particles. The number of algorithm iterations is dependent on the convergence criteria specified for the local-search PSO part of the algorithm, and because noisy peaks are smaller in magnitude than peaks caused by acoustic sources, convergence can take many iterations, resulting in a high computational cost. Lowering the conditions for a convergence allows peaks to be found quickly, but with a high amount of clutter.

Whilst both the Niching algorithm and the WoSP algorithms suffer from optima clutter, the WoSP method suffers from unpredictable computational burden which is somewhat dependent on the noise environment. The Niching algorithm has problems with swarms merging and taking over the entire search space, however this is not insurmountable. Consideration is given in 7 to how this can be overcome to produce a method which very quickly localises optima on the objective function, and whose main problem is the large level of clutter compared to the number of optima created by acoustic sources. This is generally a poor localisation function, however, it is feasible on the context of a filtering system which accounts for and even expects such clutter.

6.3.4 Locust Swarms

The performance of the locust swarms algorithm is in stark contrast to the Niching PSO and the WoSP algorithm. Locust swarms effectively deal with the niching problem by defining niches as the result of a fully completed PSO search. The, 'devour and move on' paradigm means that the locust swarm implementation is effectively repeating PSO searches on the objective function, with previous optima ruled out as potential results. This means that the niching condition is as good as it can be - desirable maxima corresponding to previously discovered global maxima are used, rather than speculatively marked potential solutions, as is the case with WoSP and Niching PSO. Note that, the swarm size used for the particle swarm experiments was 15 particles, as this represented a good trade off between localisation speed and accuracy, as demonstrated by the experiments in Chapter 5. The PSO variant used as the underlying search routine was the standard inertial PSO, with the remainder of the PSO parameters used set to the

same values used in Chapter 5, initially using the invisible walls boundary condition.

The result of this is that the algorithm generally returns the desired optima. If the number of optima required is known, then exactly that many can be returned. Depending on the objective function at the time window being considered, these generally correspond to the global maxima, rather than just peaks in the noise floor. This allows the algorithm to be considered with no cardinality error, which allows the assessment of its ability to localise a given number of acoustic sources. Further consideration of how this must be adapted for an unknown number of sources is given in Chapter 7.

6.3.4.1 Repelling Force

The first experiment run was to investigate the effect of varying the strength of the repulsive force applied from positions of previously discovered optima. The initial value given to the forces was to ensure that the maximum particle velocity was applied when a particle was within half a meter (on the horizontal-plane) of a historical optimum position. With the maximum per-dimension particle velocity set to 0.1m per iteration, the target repulsive force coefficient was determined to be 0.025 by simple manipulation of Equation (6.12), as shown in Equation (6.16).

$$0.1 = \frac{\alpha_r}{0.5^2} \tag{6.16}$$

To study the effect of changing this value, it was varied around this ideal value as shown in Figure 6.5. This figure shows that the average number of FEs required to find two sources doesn't change when the strength of the repulsive force changes around this desired value. This can be explained when considering the clustering behaviour of the particles. When particles are ejected from the site of a previous optimum value, they move at the maximum permissible speed out of that site's region of significant influence. Because the magnitude of the force is inversely proportional to distance between the two positions, the contribution of a repelling position to a particle's velocity is insignificant for most of that particle's life.

Similarly, Figure 6.6 shows that if the repulsive force coefficient is kept within a sensible range, it has no effect on the localisation performance of the algorithm other than ensuring that the same optimum is not found twice.

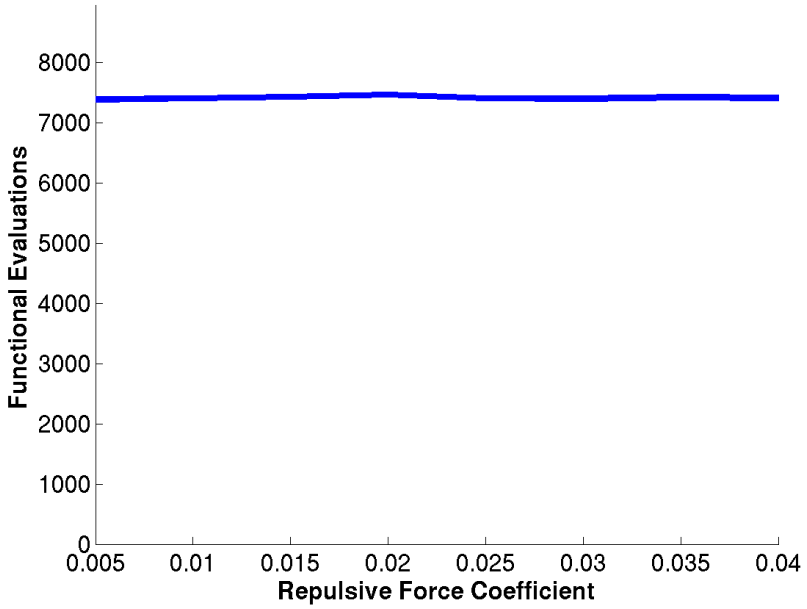


Figure 6.5: Functional Evaluations required to localise 2 sources using the Locust Swarms algorithm, against the size of the repulsive force coefficient applied.

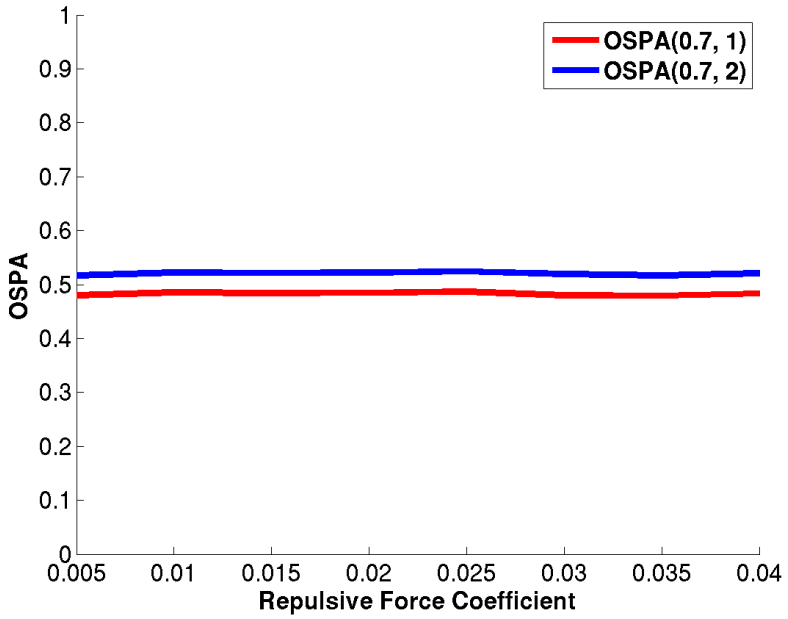


Figure 6.6: OSPA metric against the size of the repulsive force coefficient applied for localisation using the Locust Swarms algorithm, with two OSPA parametrisations shown.

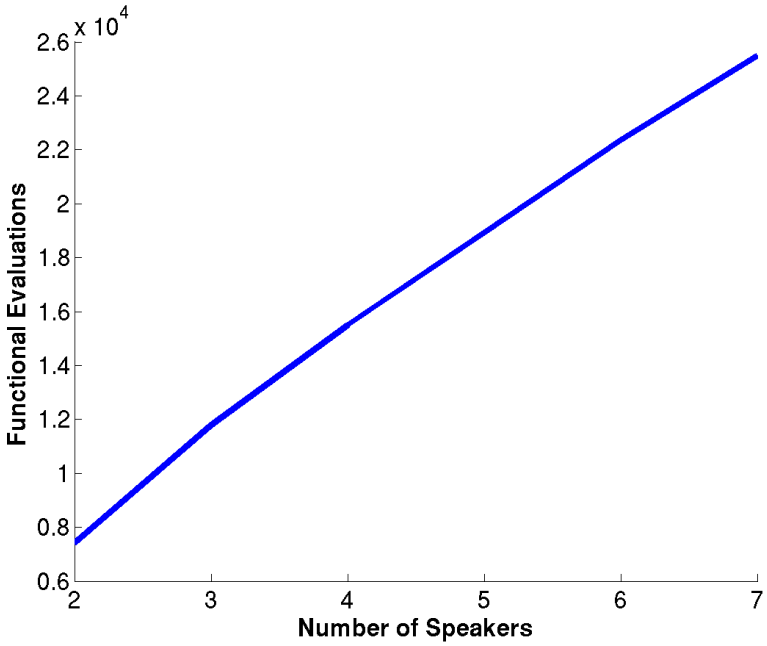


Figure 6.7: Functional Evaluations required to localise a given number of sources on simulated data, using the Locust Swarms algorithm

6.3.4.2 Variable Number of Sources

Figure 6.7 shows how the total number of FEs required to localise a number of sources scales as the number required is increased when the algorithm is run on the simulated data set. The graph shows that the number of FEs increases linearly with the number of optima to be found, and this corresponds to an extra PSO search per additional source required. Because the algorithm is a repeated PSO search, this result is not unprecedented - subsequent searches are kept away from previous optima and the algorithm continues as normal. A foreseeable caveat to this is that a sufficiently dense population of sources might cause particles to become trapped, however this scenario was not encountered, and so was not considered any further than as a potential pitfall.

Figure 6.8 demonstrates how the OSPA score average stays constant regardless of how many sources are searched for. These scores appear to compare well with the general level of error encountered for single sources, and this is not unwarranted given that in practice, the algorithm is a repeated single-source search. Figures 6.9 and 6.10 show the same algorithm run on the recorded data set. Whilst there were fewer available sources to deal with, the trends are the same. There is a linear increase in the number of FEs required to localise every additional source, and the average OSPA metric is unaffected by the number of sources to be found. The

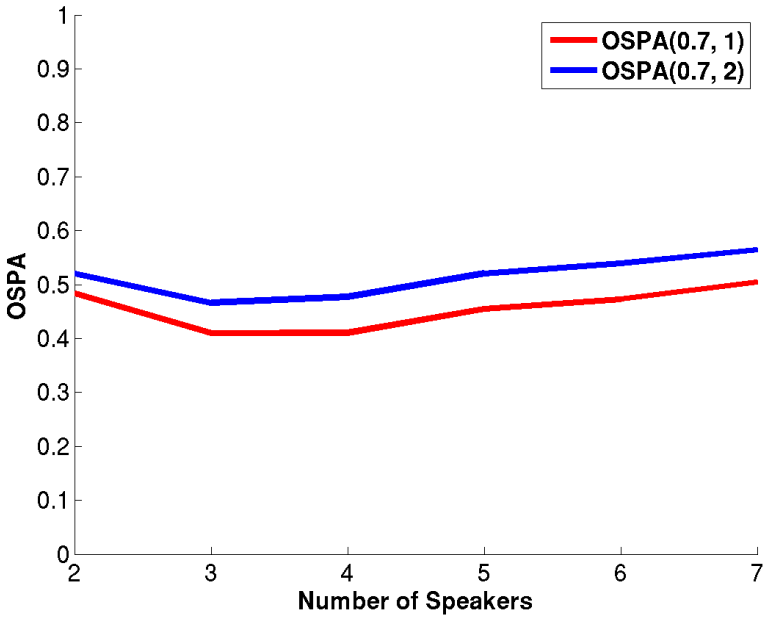


Figure 6.8: Average OSPA score versus number of sources localised on simulated data, using the Locust Swarms algorithm.

main difference is the slight increase in both FEs required and localisation error when compared to the results for the simulated data set. This is likely due to the highly reverberant nature of the acoustic lab, and mirrors the results of the single-source PSO experiments. Nevertheless, the results are still within usable ranges - the number of FEs per speaker is still in the order of thousands rather than tens or hundreds of thousands, and the localisation error is in the same order of magnitude as that obtained on the simulated data set.

6.3.4.3 Robustness to Noise

Finally, as was the case with the single-source PSO localisation, it was of interest to study the algorithm's robustness to acoustic noise. As before, the level of artificial acoustic noise added to the simulated data set microphone signals was varied to achieve different SNR levels. The algorithm was then re-run over a wide range of these levels.

Given the single-source PSO localisation algorithm's good performance in the face of low SNR it was expected that the Locust swarm algorithm would perform similarly. As Figure 6.11 shows, much the same effect was observed on the number of FEs required. For high SNR conditions, the algorithm was unaffected, and only in very low SNR conditions was there a

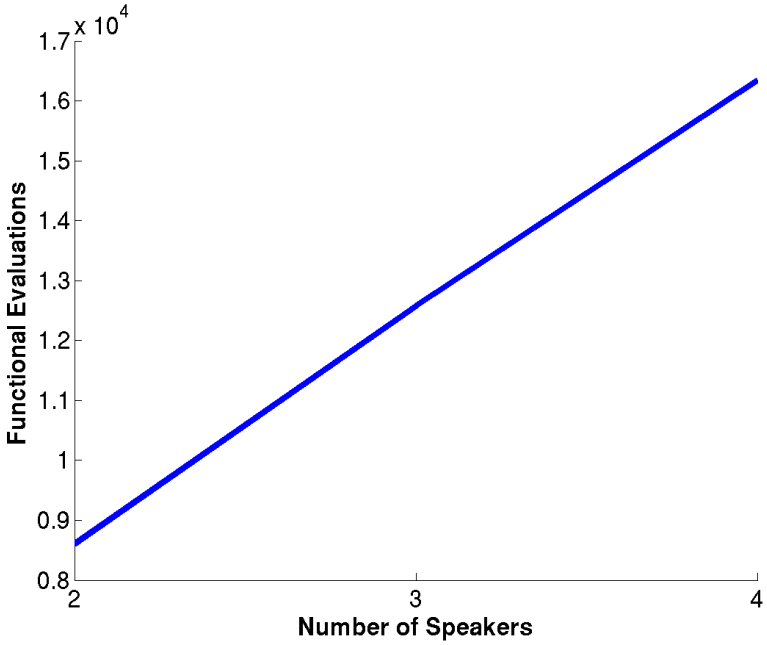


Figure 6.9: FEs required to localise a given number of sources on recorded data, using the Locust Swarms algorithm.

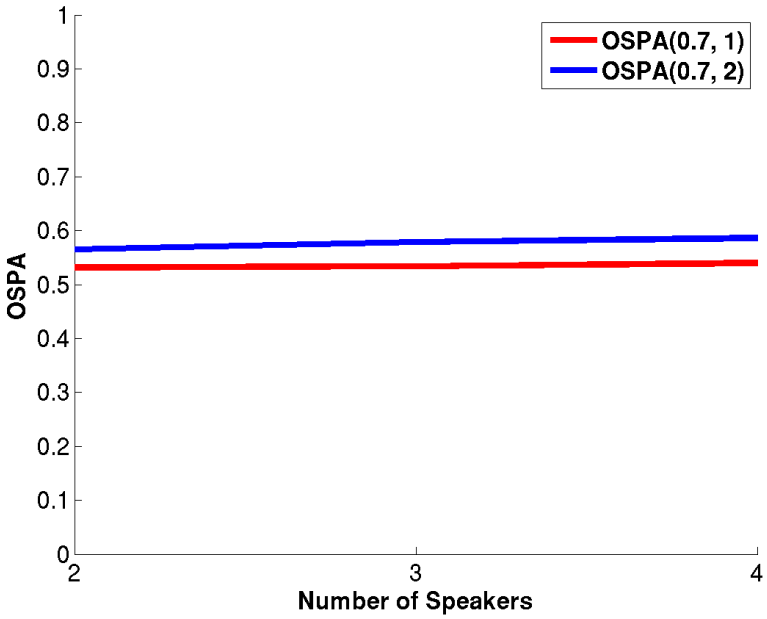


Figure 6.10: Average OSPA score versus number of sources localised on recorded data, using the Locust Swarms algorithm.

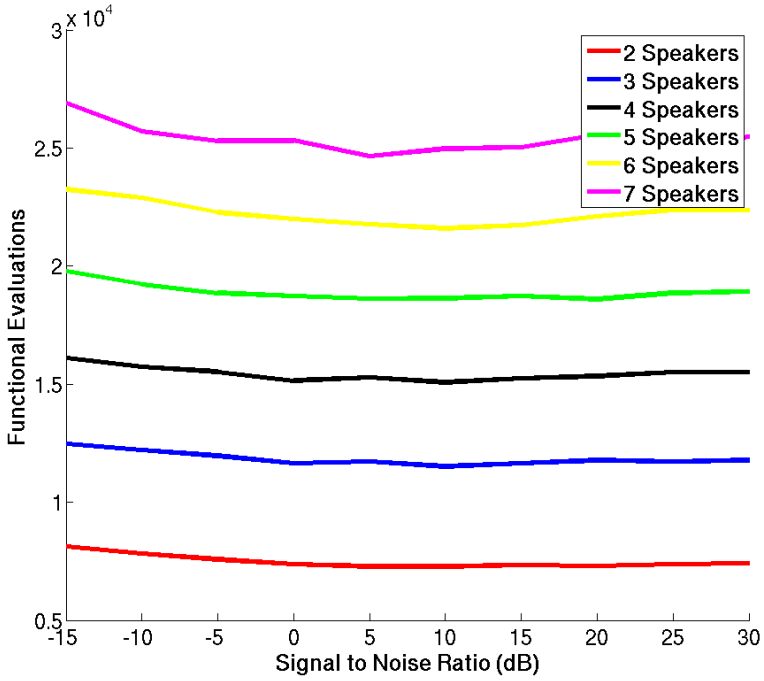


Figure 6.11: Functional Evaluations required to localise sources versus SNR, on the simulated data set using the Locust Swarms algorithm. A range of number of concurrent speakers are shown, showing robustness to noise level and a consistent increase in the number of FEs required for each additional speaker.

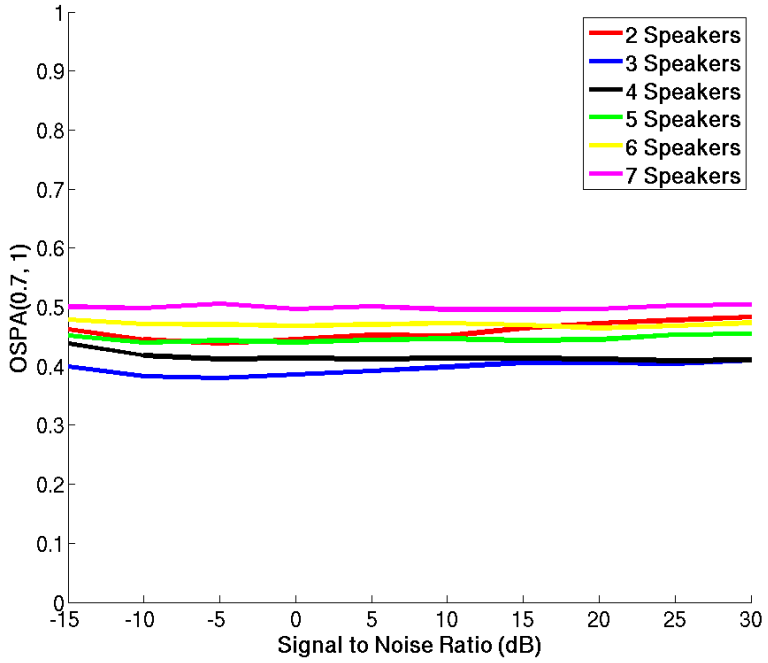


Figure 6.12: Average OSPA score versus SNR for the Locust Swarms algorithm, showing localisation robustness to noise level across a range of different numbers of concurrent speakers.

small but noticeable detrimental effect. This can be readily explained in terms of the peaks caused by acoustic sources being closer in magnitude to the noisy peaks of the SRP objective function in low SNR conditions. Although the algorithm is suitable for operating in noise, less distinct peaks compared to the noisy peaks generally take more effort to find.

Despite the additional effort required, Figure 6.12 demonstrates that if the peaks are still distinct, they will continue to be found with the same level of localisation error regardless of acoustic noise. The only caveat to this is that the noise level which causes peaks to no longer be distinct from noisy peaks is likely to be system dependent. Generally, a system with fewer microphones than used in these experiment might not do so well, and the placement of those microphones should be such that they surround and can cover the entire search area.

6.4 Conclusions

The work in this Chapter has established the suitability of multi-optima PSO algorithms as the basis for multi-speaker acoustic localisation. This was a logical extension of the work carried

out in Chapter 5.

The Chapter compared three different multi-optima approaches, and lead to some interesting conclusions on what is required of a multi-optima algorithm for use with acoustic data. The key observation is that niching - the process of identifying areas of space which might contain an optimum value - is non-trivial on the non-smooth SRP objective function. For sequential niching algorithms, this generally leads to a large amount of optima recorded which are simply peaks in the noise-floor of the function. The sequential nature means that the time taken to reach a peak of interest is unpredictable, and the particular sequential niching method used - the WoSP algorithm - was sensitive to the different characteristics of the SRP over time. This made it hard to tune and unreliable, which marked it as being unsuitable for the purpose of acoustic source localisation.

In contrast, the parallel niching algorithm investigated - Niching PSO - allowed a set of optimal positions to be obtained within a predictable timeframe, albeit with a large amount of clutter not corresponding to true targets. The Niching PSO does have problems with swarm management, meaning the number of results returned was not predictable. This also meant the algorithm was not reliable, however this problem is not insurmountable. As is, the algorithm is unsuitable for reliable localisation because it returns a lot of clutter, but often produces only a very few results corresponding to sources. However, if the algorithm were to be modified, as will be explored in Chapter 7, to return a consistent level of clutter, this has potential use within the context of a tracking system so long as the acoustic targets were found amongst the clutter.

The third multi-optima search technique, the Locust swarms algorithm, was far more successful than the other two. This method effectively set the niching criteria as the successful localisation of a globally optimum value, and then carried on searching whilst excluding any previously discovered optimal positions. This led to a reliable number of outputs - noisy peaks were largely ignored, although because the PSO does not *guarantee* that the search result will be the global optimum, repeated PSO does not guarantee that all results found will not correspond to peaks from noise. Nevertheless, missed observations and spurious results are to be expected, and the Locust algorithm can locate many sources so long as enough iterations to the algorithm are used.

Because the relationship between sources to be found and the number of FEs required is linear, and dependent on the swarm size, the results of Chapter 5 have been extended to conclude that

the computational effort required in contemporary techniques to localise a single source can now be better utilised to find a large number of sources. Because the computational requirements of single source localisation were improved in Chapter 5 by approximately an order of magnitude, an order of magnitude more sources can be localised than was previously possible with the same computational budget. This has been achieved in the context of minimising the localisation error as much as possible, where the OSPA metric has been used to measure this multi-source localisation error.

To conclude, multi-optima PSO techniques have been successfully applied to acoustic data to localise multiple speakers. This has been achieved whilst keeping computational costs low and accuracy to a good level. The experimentation has highlighted some of the difficulties in adapting multi-optima techniques to the acoustic source localisation problem, and has indicated how these techniques might be put to use in a full ASLT system.

Chapter 7
Acoustic Multi-Source Tracking

This Chapter of the thesis sets out to explore how the output of the multi-speaker localisation methods developed in Chapter 6 can be used to perform speaker tracking. The tracking stage of an ASLT system is important as it attempts to take a series of localisation results over time and identify any continuous sources. This means pruning out any clutter - unwanted observations not corresponding to acoustic sources - thereby decreasing the localisation error of true sources over time.

This Chapter recognises that the direct observations of optima of the SRP function across a room are a good fit for filtering using RFS based filters. Furthermore, because the observations map linearly to source positions, the GM-PHD filter is explored as a potential tracking mechanism. The GM-PHD filter is of particular interest because of its low computational complexity compared to SMC (particle filter) based RFS methods, particularly when dealing with an increasing number of speakers. Because the number of speakers which can be found by the previously developed PSO localisation techniques is arbitrary, it lends itself to environments containing a large number of speakers. Previous work has found that tracking can become computationally intractable above 3 speakers, and so the GM-PHD filter is explored as a potential solution to this in conjunction with the localisation power of multi-optima PSO methods.

The localisation methods developed are modified to produce some clutter in order to deal with a potentially increasing number of speakers. The level of clutter expected is then passed to the GM-PHD filter. The filter is then tuned to account for the details of the localisation methods used - this largely means that an idea of noise levels in the observations and the detection probabilities must be known. The ability of the filter to extract targets from the PSO observations is then demonstrated and explored.

7.1 Tracking Multiple Sources

When tracking a single acoustic source, it has been demonstrated that a Bayesian filter such as a Kalman filter (or a non-linear extension thereof) or a particle filter is sufficient to track that source so long as observations can be assigned as being produced by that source. This filtering allows the position of a source in observational noise to be located and subsequent localisations associated over time such that they can be said to have originated from the same source. In the case of multiple sources, it is not necessarily clear which of multiple observations even correspond to a source. The number of sources must be estimated, and observations used to extract multiple source positions.

The problem is not simple, however it has been addressed in the literature. Recently, RFS statistics have been used to produce multi-target tracking algorithms. In particular, the GM-PHD filter - introduced in Section 3.2.3 - is a relatively low complexity system, which gives it the potential to track a large number of sources. Note that some contemporary techniques have non-polynomial complexities given a number of observations [110], and the Cardinalised GM-PHD filter (a modified version of the GM-PHD filter) is cubic in complexity. Whilst cubic complexity is not ideal, it is a great deal more desirable than an NP-hard tracker.

This algorithm has been applied to multiple source tracking using TDOA measurements [111, 112] and shown to be practical. This work is expanded in this Chapter by using the SRP to provide a set of observations which correspond linearly to speaker positions. This removes some complexity - observations directly correspond to source positions and so the unscented transform used in [111] is unnecessary. The SRP is also considered to be a measure which is relatively robust to noise and reverberation, providing extra motivation to use it in place of TDOA measurements.

Given the previously demonstrated ability of PSO based multi-optima search techniques to reliably return sets of observations which correspond well to known source positions, these methods are a natural fit for providing the observations required by a GM-PHD tracker. However, in the previous experiments, the number of sources to be localised was known. To deal with an unknown number of speakers, the tracker must be given information about any new sources which appear. In the case of the Locust Swarms algorithm, without knowing any source positions initially, the algorithm might search for 0 sources and return an empty set of optima, which will lead to no sources ever being tracked.

The Niching PSO algorithm can return a large number of results, most of which counts as clutter. However, it does not do so reliably. Whilst unsuitable as a localisation method which returns peaks with very little clutter, Locust Swarms can be modified to consistently return cluttered results which, importantly, are likely to include the positions of the sources of interest. Because the GM-PHD filter specifically accounts for clutter, Niching PSO is also potentially a good fit for the filtering algorithm.

7.2 Modified Particle Swarm Optimisation Algorithms

Two of the previously studied localisation routines are considered for use with the GM-PHD filter. The Locust Swarms algorithm and the Niching PSO both require slight modifications before their output can be applied to the filtering system. This section details how each algorithm is to be modified.

7.2.1 Locust Swarms

The Locust Swarms algorithm returns only a set number of optimum positions. Therefore, for a tracker to find known sources on subsequent frames, it must request the Locust Swarms localiser to return that many new optimum positions, albeit on the understanding that those new positions might not correspond to the sources already tracked. In order to discover new sources, the tracker must force the localisation routine to identify more sources than are currently tracked. This is particularly true when no speakers have yet been identified, as requesting 0 optimal positions stands no chance of finding any new sources.

For the experiments run for use with the tracker, this was simply achieved by forcing the number of optimal positions to be returned to be 3 greater than the number of sources in the set. Whilst this doesn't use the estimate of the number of sources at each tracker stage, it does ensure that on the set with the smallest number of simultaneous active sources (2 speakers), 5 localised positions were returned by the localiser. This could easily be extended to cope with the situation where the number of targets was truly unknown by requesting a number of observations some constant over the current estimated number, subject to a minimum being returned. For experimental purposes, this was not considered further - the target number of speakers was simply used to generate a constant level of clutter for each combination of speakers.

7.2.2 Niching Particle Swarm Optimisation

Several problems have been identified with using Niching PSO as a localisation method for acoustic sources. The major problem is the tendency for sub-swarms to merge and form larger sub-swarms which cover a high proportion of the search space. This effect frequently results in the method only returning one optimal position, particularly if small swarm sizes are used. The other main problem is the large and varying number of positions returned which do not necessarily correspond to a true acoustic source.

The first problem is tackled by forcing sub-swarms to remain small in terms of radius. This effectively stops the ballooning of the set of sub-swarms into one giant sub-swarm. The second problem is approached by accepting clutter as something that the tracker has to deal with, and altering the convergence criteria so that a large number of observations are returned after a reasonable number of iterations.

7.2.2.1 Sub-swarm Limiting

In order to prevent sub-swarms merging into one giant sub-swarm, the merge step of the Niching PSO algorithm is modified. The change made is simple - two sub-swarms are only allowed to merge if both of their radii are less than a threshold, denoted m_t. In the context of Equation (6.5a), this means that both $(R_{j1} < m_t)$ and $(R_{j2} < m_t)$ must be true before Equation (6.5a) can be used as a condition for merging.

In the experiments on multi-speaker tracking, the threshold was set to $m_t = 0.3$. This threshold is measured in meters, so sub-swarms are effectively made to search areas limited to 30cm around their best encountered position. Sub-swarms are allowed to be larger than this however, as the initial creation of a sub-swarm might pull in a nearest neighbour particle from over 30cm away, particularly if there are very few particles left in the main swarm. The result of this change does indeed prevent the giant sub-swarm problem. However, because most of the sub-swarms are typically not located around a speaker position, they can encounter many noisy peaks. This has the desired effect of producing measurements with clutter, however if the noise floor of the SRP contains many similarly sized peaks, convergence will be very slow for sub-swarms.

7.2.2.2 Niching Criteria

In order to ensure a large number of sub-swarms were created quickly, the niching conditions of the algorithm had to be modified to create sub-swarms relatively easily. This aspect of the algorithm behaviour is controlled by the particle variance threshold, ϵ_t. This threshold was empirically lowered to 0.05 in the multiple-source tracking experiments. This value was found to allow sub-swarms to be created after four or five iterations of the main algorithm loop.

7.2.2.3 Convergence Criteria

Convergence of sub-swarms was found to be slow, taking many iterations of the overall algorithm. This was largely due to swarms which weren't centred over the distinct peaks caused by acoustic sources. The noisy peaks examined by most sub-swarms were all similar in magnitude, and the original implementation of the algorithm is purposefully robust to such noise. Sub-swarms examining these areas would typically encounter a local maximum value, but that location would have many similar peaks locally. This caused regular resetting of the counter used to keep track of how long a sub-swarm's best value had remained unbeaten, resulting in a large number of epochs before a peak was settled on.

This computational cost was made worse by the use of large sub-swarms - necessary in order to keep radii down. Each algorithm iteration therefore required a large number of FEs compared to the single-source case PSO algorithm, where each iteration could easily make do with 10 FEs. With each sub-swarm requiring more algorithm iterations than the equivalent single-source PSO, but with many more particles, the overall number of FEs quickly became uncompetitive when compared to the Locust Swarms method. In the multi-source experiments, the initial main swarm size used was 100 particles, allowing for up to 50 two-particle sub-swarms, assuming no merging. This pushed the number of FEs required for the algorithm to complete into the high tens of thousands, with many swarms not converging and the algorithm completing after reaching a maximum allowed number of iterations.

To counter this issue, the requirements for sub-swarms to converge were significantly relaxed. The number of consecutive unchanged global maximum positions required for each sub-swarm to converge (g_{epochs}) was reduced from 70 to 3. This had the immediate effect of allowing sub-swarms over noisy areas to quickly settle on an arbitrary optimal point in their vicinity. It also continued to allow sub-swarms investigating areas containing peaks caused by sources to

discover and report those peaks, although the peaks reported were likely to be subject to more observational noise, caused by settling on a lesser peak in the vicinity of a true peak before that true peak could be discovered.

7.2.2.4 Computational Cost

The potential advantage of the Niching algorithm over the Locust Swarms algorithm is that peaks are discovered in parallel. By relaxing the sub-swarm convergence criteria, the number of FE to discover a number of optima including clutter was tunable, but in practice was set so that only in the order of thousands were required. Note that because, in the case of using 100 particles, this only requires in the order of tens of algorithm iterations. Whilst parallelisation of the Locust Swarms algorithm with a typical swarm size in the order of tens might yield an order of magnitude practical speed-up, doing the same with the Niching PSO has the potential to yield a 100 times speed up of the localisation.

To add to this, the Niching PSO returns optimal positions without being made aware of how many sources it is required to localise. This gives it a clearly desirable advantage over the Locust Swarms algorithm, which requires a linear increase in the number of FEs required for every potential source to be found. By keeping the number of Niching PSO iterations required for convergence in the order of tens, the algorithm might conceivably be able to localise an almost arbitrary number of sources in the same number of FEs as the Locust Swarms requires to localise one source.

The caveat to this speed advantage in terms of FEs is that when implemented and run under MATLAB, the Niching PSO takes a lot more raw processing time to complete than the Locust Swarms implementation does on the same data set. This was found to be due to the part of the algorithm loop which dealt with swarm management. The MATLAB code used was not particularly well optimised, making repeated use of nested ‘for’ loops and arrays which were not pre-allocated. The task of swarm management is expected to be much more efficient when rewritten to be optimised in MATLAB, or perhaps in a more practical language for online signal processing such as C.

7.3 Gaussian Mixture Probability Hypothesis Density (GM-PHD) Tuning

In the single-source tracking problem, the Kalman filter can be used to filter observations, however it has to be provided with some knowledge of the problem environment. This means that the state update matrix and the observation matrix must be known. The process noise and observation noise covariance matrices must also be known, or at least estimated. Similarly, the successful operation of the GM-PHD filter requires that various parameters be known.

Just like the Kalman filter, the GM-PHD tracker requires (linear) state update equations and process and observation noise covariance estimates. It also requires information concerning the probability that the localisation function will return, amongst the clutter, the true location of a source obscured by some observation noise. This is because it is entirely possible that a true source might be missed, and only clutter found.

The algorithm also requires that the clutter density be known (or at least estimated). This simply means that given the area of a search space, the expected number of clutter observations per unit area must be known. A maximum target velocity must also be provided, which can be readily estimated by considering the maximum speed of a moving speaker ($5\mathrm{ms}^{-1}$), and dividing by the number of audio frames processed per second (2.7). In the experiments run, this parameter was set to 1.8 metres per iteration.

Finally, the algorithm requires the tuning of several threshold values which are not readily derived from the problem environment. These include the weight threshold, which determines when weighted observations are to be considered targets; the merging threshold, which controls when nearby tracked targets are deemed to be observing the same true target; and the extraction threshold, which determines the point where tracked targets are considered to be true targets which should be reported as a final result. In the multi-source experiments, these values were tuned manually to attempt to get the best results from the system. Table 7.1 shows the values of these parameters used in the experiments. The probability of source survival was set to 0.99, that is, it was assumed that active speakers were talking almost continuously.

weight threshold (T)	1×10^{-7}
merging threshold (U)	60
extraction threshold (E)	0.7

Table 7.1: GM-PHD tuning parameters

7.3.1 Problem Model

The movement model is an important factor for acoustic source tracking [55]. In previous work, it was assumed that sources only moved slowly under process noise. Indeed, in the previous Chapter, no movement model was used at all. This was sufficient for testing the localisation power of the multi-optima PSO methods, which did not have to be aware of source movement in order to find peaks of the objective function. However, the model used for speaker movement for the multi-speaker tracking experiments was the simple linear Gaussian model developed in [67].

This model keeps track of a sources' positions and speed, whilst only observing their positions. The state update model specifies that the targets move at a constant velocity (which is a limitation in practice). That velocity is determined when a tracked source is 'birthed' after two observations. It is simply the change that candidate tracked location's position from the previous time step, divided by the time in seconds of that time step. The state is stored as a four element vector, $[x,\, y,\, \dot{x},\, \dot{y}]^{\mathrm{T}}$. In the context of Equation (3.41), the state transition matrix $F_t\zeta$ and the process noise covariance Q_t are given in Equations (7.1) and (7.2) respectively, from [19].

$$F_t\zeta = \begin{bmatrix} \mathbf{I}_2 & \Delta\mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \end{bmatrix} \tag{7.1}$$

$$Q_t = \sigma_\nu^2 \begin{bmatrix} \frac{\Delta^4}{4}\mathbf{I}_2 & \frac{\Delta^3}{2}\mathbf{I}_2 \\ \frac{\Delta^3}{2}\mathbf{I}_2 & \Delta^2\mathbf{I}_2 \end{bmatrix} \tag{7.2}$$

In these equations, Δ is the tracker update period (the length of an audio frame in seconds) and $\mathbf{I}_2$ and $\mathbf{0}_2$ are the two by two identity and zero matrices respectively. σ_ν is the standard deviation of the process noise.

Note that for these experiments, only the two-dimensional position over the horizontal-plane is

tracked for simplicity. This is despite height information being localised, however as before, we assume that speakers cannot be on top of one another, and therefore whilst height is important to localisation, it is not necessarily so important for the final tracked result.

The measurement matrix, H_t is simply a two by two identity matrix, because the measurements obtained by the localisation functions are simply Cartesian positions within the search area.

7.3.2 Birth Model

The target birth model used in [19] uses a Gaussian mixture, with two Gaussian distributions centred over the known source starting positions. This model is not appropriate to the multi-speaker experiments, as the starting positions of the sources are assumed to be unknown. Instead, a mixture model of thirty low variance Gaussian distributions are used to approximate a Uniform distribution over the search space, as described in [113]. This is not an ideal model, however it must suffice for a simple implementation of the algorithm, as this is the form (a Gaussian mixture) required by the initial authors [19].

Future work should consider using a modified GM-PHD which allows the use of a true Uniformly distributed birth model [113], allowing sources to be created from anywhere within the search space with equal probability. It might also be a good idea to model the birth intensities as Gaussian distributions centred over the positions of potential speakers (people) whose positions are estimated from a camera based system. This might allow the improvement of results from both the audio and the video domains by the fusion of sensor information. Importantly, it would also allow a separation to be made between tracking people within a room, and tracking who is speaking.

7.3.3 Clutter Density

The clutter density $\kappa_t(z)$, referred to in Equation (3.46c), is modelled as a Poisson RFS with intensity given by Equation (7.3) [19].

$$\kappa_t(z) = \lambda_c V \mathcal{U}(z) \tag{7.3}$$

In this equation, V represents the search volume, and $\mathcal{U}(z)$ is a Uniform distribution over that

volume. λ_c is the expected number of clutter results per unit volume, and must be changed for each of the different localisation methods used.

For the Locust Swarms algorithm, the clutter was expected to consist of three element every audio block. This was simply divided by the search area to calculate λ_c, and this value was used for all speaker combinations, as 3 extra localisations were demanded in each case, as described in Section 7.2.1. With the simulated room environment having a volume of 120 cubic meters, this led to a value of $\lambda_c = \frac{3}{120} = 0.025$

For the Niching PSO algorithm, tests, 100 particles were used. This implied a maximum of 50 sub-swarms available, each able to return a result. The output was limited to 40 elements as, in practice, 50 elements were never returned as sub-swarms merged together. Because the number of sources was not considered by the localisation routine, the average number of clutter results was taken as approximately $(40 - \bar{r}$, where $\bar{r}$ was the average number of speakers considered rounded to the nearest integer. Cases were tested with between 2 and 7 speakers, leading to an estimate of $\lambda_c = \frac{36}{120} = 0.3$.

7.3.4 Probability of Detection

The probability of the localisation methods successfully detecting any single source had to be determined from the data output by the localisation methods before filtering. What actually counted as a detection was defined as a source being within a 2D radius d_s of the known source locations. This was determined for each source in each set of speakers, and averaged over each of the Monte Carlo trials run, for each algorithm. This test covered around two hundred Monte Carlo trials for each case, each case consisting of around 100 acoustic frames and between 2 and 7 stationary speakers, depending on the experimental run considered. This resulted in the localisation algorithm output being averaged over around $20,000$ trials.

Note that the simulated data sets were used to derive these empirical probabilities, and these data sets consist of continuous speech. This allows the ground truth to be known exactly, and therefore the process noise covariance to be assumed to be 0. However, natural pauses in speech result in no visible peak of the SRP function, and so these probabilities of detection count there being no particle close to the known speaker position as a missed source. Furthermore, the speech signals used were of slightly different lengths, so a large number of frames will have been marked as not detecting a known source, even though that source had finished talking,

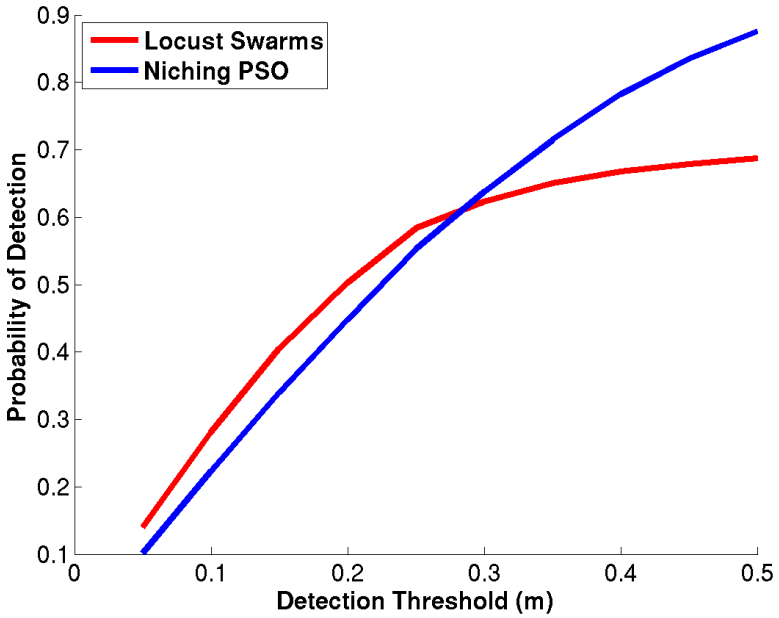


Figure 7.1: Average probability of detecting a source versus source detection radius threshold, d_s.

slightly before the end of the largest speech segment.

Figure 7.1 shows how this probability changes when d_s is varied for both the Locust Swarms algorithm and for the Niching PSO algorithm with parameters.

This graph indicates that the Locust Swarms algorithm detection probability plateaus, and this is largely due to the aforementioned effect of time frames being marked with missed detections when there was no signal to actually detect. In contrast, the Niching PSO algorithm starts of with a lower probability of detection, but does not plateau so quickly. This is due to elements of the clutter returned occasionally being in the vicinity of a known source, even when it is not active, and that clutter element being marked as a detected source.

Whilst these probabilities have problems, they indicate that the Niching PSO generally appears to have a slightly lower probability of source detection than the Locust Swarms algorithm. Furthermore, they do not necessarily reflect on the capability of the algorithms to reliably detect peaks of the objective function. Rather, because speakers are not truly continuously active, these probabilities give an indication to a tracker of how likely it is that a speaker who doesn't stop talking will be detected, despite natural pauses in speech.

Using this graph, the both tracking algorithms can be fed a probability of detection based empirically on how close a detected position has to to the ground truth in order to count as a successful detection. For both algorithms, a value of 0.65 was chosen for this parameter. This corresponds to a detection threshold which is in the middle of the range, and where the Locust Swarms algorithm probability begins to plateau.

7.3.5 Noise Parameters

In order to derive figures for the observational noise covariance, the study of probability of detection was extended to record every (2D) error vector for every observation considered as a detection. The sample covariance of these arrays were then calculated using MATLAB. These covariances were considered to represent the observational covariance in each case, because the simulated data set was used, and so the process noise covariance could be assumed to be zero.

In the GM-PHD model, the expected observation noise is of the form shown in Equation (7.4). This form is simply a diagonal covariance matrix - no cross-covariance terms - whose scale is determined by a single factor, σ_r^2, meaning that the variance in both x and y axes is the same. Because the sample covariance was calculated from a large yet finite set of measurements, the form of the values could not be expected to be a truly diagonal matrix, however the results were close to the expected form. The diagonal components of the calculated matrices in each case were almost the same, and the off-diagonal components were always orders of magnitude smaller.

$$R_t = \sigma_r^2 \mathbf{I}_2 \tag{7.4}$$

Figure 7.2 shows the estimated observation noise variance (σ_r^2) graphed against the detection threshold, d_s, used for the Locust Swarms algorithm. This was calculated by taking the mean of the two diagonal components of the calculated covariance matrices. As these values were always almost the same, this provided a simple way of extracting a value which could be used for the diagonal covariance matrix model given of equation (7.4). The results of the same process on the Niching PSO output are shown in Figure 7.3.

Because the off-diagonal components were calculated numerically, they were not truly zero as expected. The eigenvalue decomposition was taken of each covariance matrix to generate a

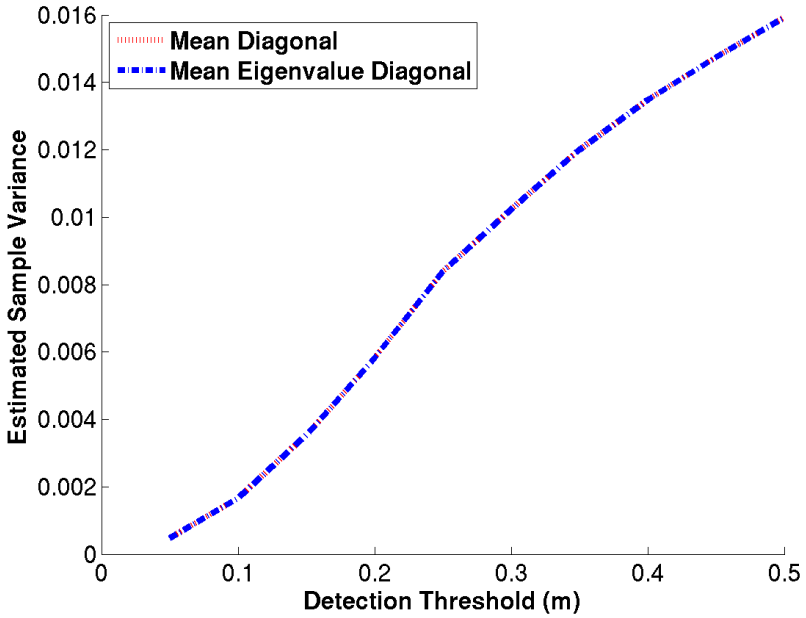


Figure 7.2: Estimated observation noise variance versus detection threshold, for the Locust Swarms algorithm.

truly diagonal matrix, a form of covariance diagonalisation [114]. The mean of the diagonals of these matrices were also calculated, and are plotted alongside the un-diagonalised matrix means in both Figures 7.2 and 7.3.

The purpose of this was simply to show that the simple averaging of the diagonal components of the calculated covariance matrixes was valid, because they do not change appreciably after diagonalisation as they are already approximately of the correct form. Note that these plots are hard to distinguish from each other as they lie directly on top of one another, showing that the measured covariance matrices are not noticeably affected by the diagonalisation process, as expected.

The result of this step simply shows that the sample covariances measured correspond well to the model assumption of a diagonal matrix. Forcing the off-diagonal matrix elements to zero and taking the mean of the resulting diagonal elements created the same variance results as the raw sample variance estimate. This was simply because the difference between the eigenvalue matrix and the original covariance matrix was almost negligible in each case. This also justifies the observation noise model, where there was no particular reason to believe that observational noise should be correlated across axes.

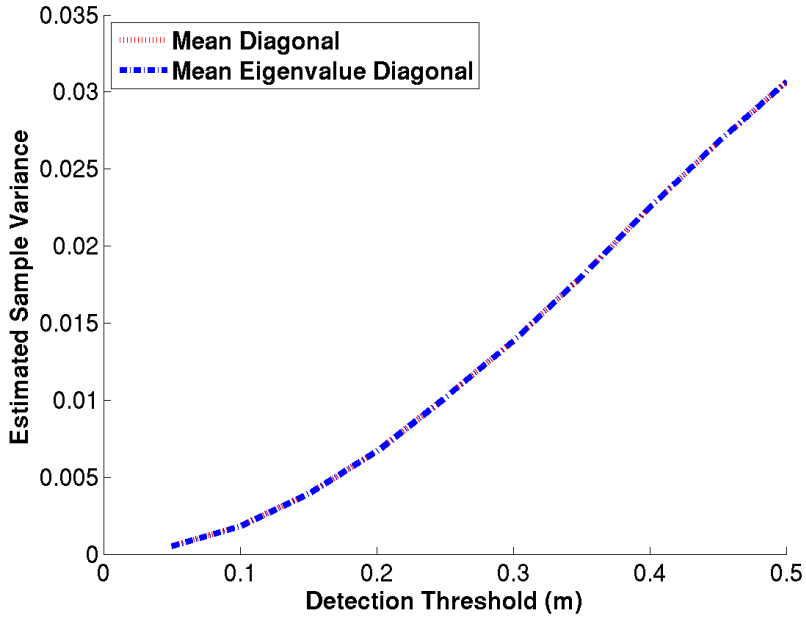


Figure 7.3: Estimated observation noise variance versus detection threshold, for the Niching PSO algorithm.

Finally, it is of interest to note that the standard deviation, $\sqrt{\sigma_r^2}$ in each case is calculated, the result expressed in metres gives an idea of how well the localisation methods perform when their results are considered to correspond to known sources. Figure 7.4 shows the estimated standard deviation achieved for successful localisations for each algorithm.

These figures lie neatly in the order of tens of centimetres for practical detection thresholds of around 30 centimetres, which corresponds to a reasonable radius to consider for a person's personal space. Note that the behaviour is very similar for the two algorithms, up until the point where the Locust Swarms probability of detection starts to plateau. This corresponds to clutter readings in the same area as a known source being counted as part of the variance calculation. These sources are largely unrelated to the source, so they are not restricted by the variance of a Gaussian centred at the source location. As such, they increase the apparent standard deviation of that Gaussian distribution.

7.4 Experimental Results

This section discusses the results obtained when the GM-PHD filter was applied to the observations captured by both the Locust Swarms algorithm and the Niching PSO algorithm. The

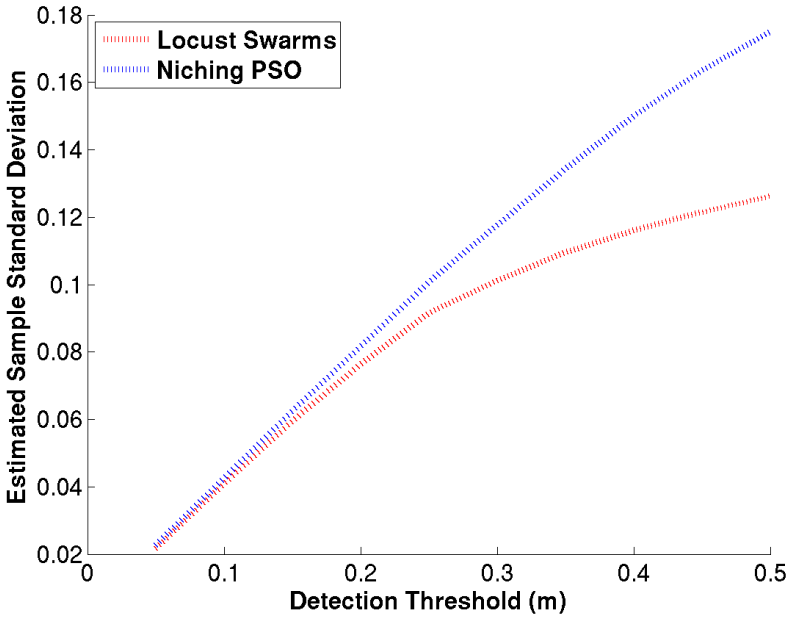


Figure 7.4: Estimated observation standard deviation versus detection threshold, for Locust Swarms and Niching PSO algorithms.

algorithms were run against the observations generated in each Monte Carlo trial of the localisation algorithm, such that the tracker performance metrics could then be averaged out. Each localisation algorithm demanded a slightly different tuning of the filter, due to the different clutter and noise profiles as described.

7.4.1 Experimental Conditions

The localisation algorithms were each run in their modified forms, outputting a set of optima including clutter at each time step of each experimental trial. The data used was the same simulated set of data used in Chapter 6, and the objective was to evaluate how good a fit for the GM-PHD filter the output of each localisation algorithm was. Also included was a set of data consisting of two concurrently active moving speakers. These sources moved past each other at a constant speed (5ms^{-1}, which is actually unrealistically high for a walking person), moving from one corner of the simulated room to another.

Note also that the acoustic noise added to the microphone signals resulted in an SNR of 15dB and that the probability of source detection was set to 0.65, based on the results obtained from the characterisation in Section 7.3.5. Similarly, the observation noise was set to 0.15, and

the process noise standard deviation was set to 0.01, as the ground truth was known exactly, although this might not be appropriate for tracking recorded acoustic sources.

The low clutter data from the Locust Swarms output was expected to perform better than the Niching PSO, due to its good localisation capability coupled with only a small amount of clutter necessary to enable the localisation of new sources. On the other hand, there is high motivation to analyse the performance of the high clutter Niching PSO, as it can be made to generate results with relatively little computational effort.

Both algorithms were run on the previously generated data consisting of between 2 and 7 stationary sources. The filtered output was saved as a set of tracked optimum positions - arrays of 2D position vectors - where each saved position should correspond to an acoustic source, because the filter should have removed spurious localisations. The general ability of the tracker to discern individual sources was of interest, as was the average OSPA metric on the tracker output against the known speaker positions. Note however, that the data sets containing 6 and 7 speakers were too short for the tracker to produce any meaningful output with, and so these results are ignored.

Finally, it was found that the boundary conditions played an important role in the multi-source experiments. In the single-source case, the invisible-walls condition was noted to be well suited to the speaker localisation task, as it minimises unnecessary computations. However, it was found that with multiple speakers, particles were continually being pushed out of bounds. Because they were divided into sub-swarms, this often led to large number of particles becoming trapped outside the search space, resulting in slow convergence and even failure to converge. As such, the wraparound method was used, to ensure that particles were always kept within the search space.

7.4.2 Locust Swarm Results

7.4.2.1 Stationary Targets

The Locust Swarms algorithm output was found to be a highly effective input to the GM-PHD filter for both the relatively simple 2 speaker case and the 5 speaker case. Figure 7.5 shows the graphical output of the filter at the final time step of one of the Monte Carlo trials. Note that the swarm size used for these experiments was 15.

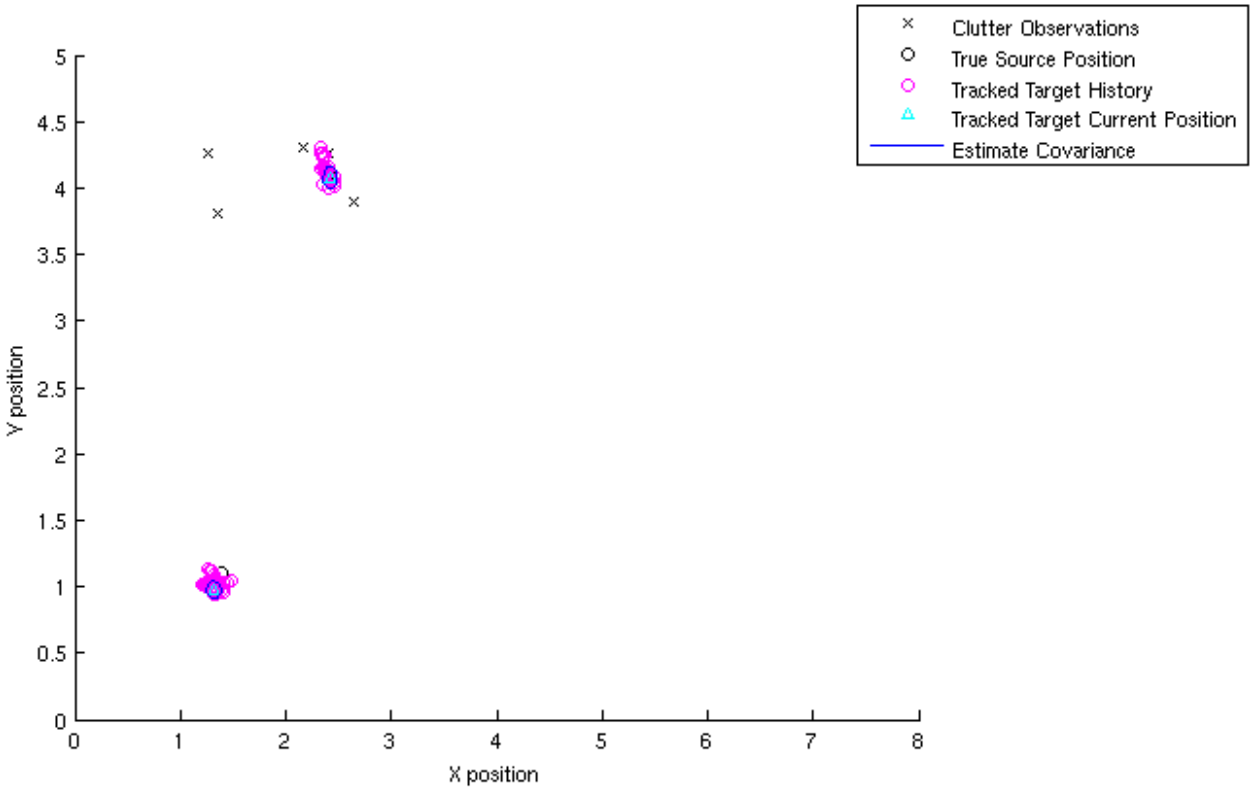


Figure 7.5: Horizontal-plane filter output for 2 speakers from the Locust Swarms algorithm input to the tracker.

This figure displays the input to the filter at that time step as black crosses, and note that they are clustered round a speaker location. These readings represent the output where one of the sources had finished talking, and so the clutter was in the area of the final active source. This trend occurred across all of the trials and time steps, indicating that the clutter output of this algorithm was not distributed uniformly over the search area, as expected by the tracking algorithm. This is fairly intuitive, as the clutter has been purposefully generated by searching for extra optima where none are expected, leading to the swarms best results remaining near its starting position of a previously localised source.

Speaker locations which have been extracted and are believed to have truly originated from a speaker are marked by a small blue triangle surrounded by an ellipse representing the trackers confidence in a location's correctness. True source positions are marked by a black circle, but are obscured by the representation of the algorithm's tracked set of locations from previous time steps, marked as pink circles.

Figure 7.6 shows the evolution of these internally tracked source locations over time for both of the horizontal-plane axes, again in pink circles. As in Figure 7.5, the observations at each time step are marked as black crosses. The relevant dimensions of the true speaker locations are marked as black circles and are more clearly visible than in Figure 7.5. This diagram shows how the tracker takes a number of frames to identify sources.

The algorithm also performed well under the 5 speaker case. Tracking more than 3 speakers was one of the motivations for using the GM-PHD filter, and the ability of the Locust Swarms algorithm to provide good results justifies this choice. Figure 7.7 duplicates Figure 7.5, but for the 5 speaker case. As before, previously tracked locations are highlighted in pink, and cluster round each of the known source locations. Extracted targets at the final time step are again marked with a triangle surrounded by an ellipse.

Figure 7.8 shows the same evolution of tracker state over time for 5 speakers as Figure 7.6 does for 2 speakers, showing the successful detection and tracking of 5 simultaneous speakers.

The performance of the system based on the OSPA metric is of interest, as ultimately the filter should provide an improvement in terms of localisation error. Figure 7.9 shows how the OSPA metric can change over time for an example Monte Carlo trail of the system. The graph shows that at the first time step, both of the OSPA measures start at their maximum of 0.7. As tracks are detected and sources identified, the metric quickly decreases. Note that the move up and

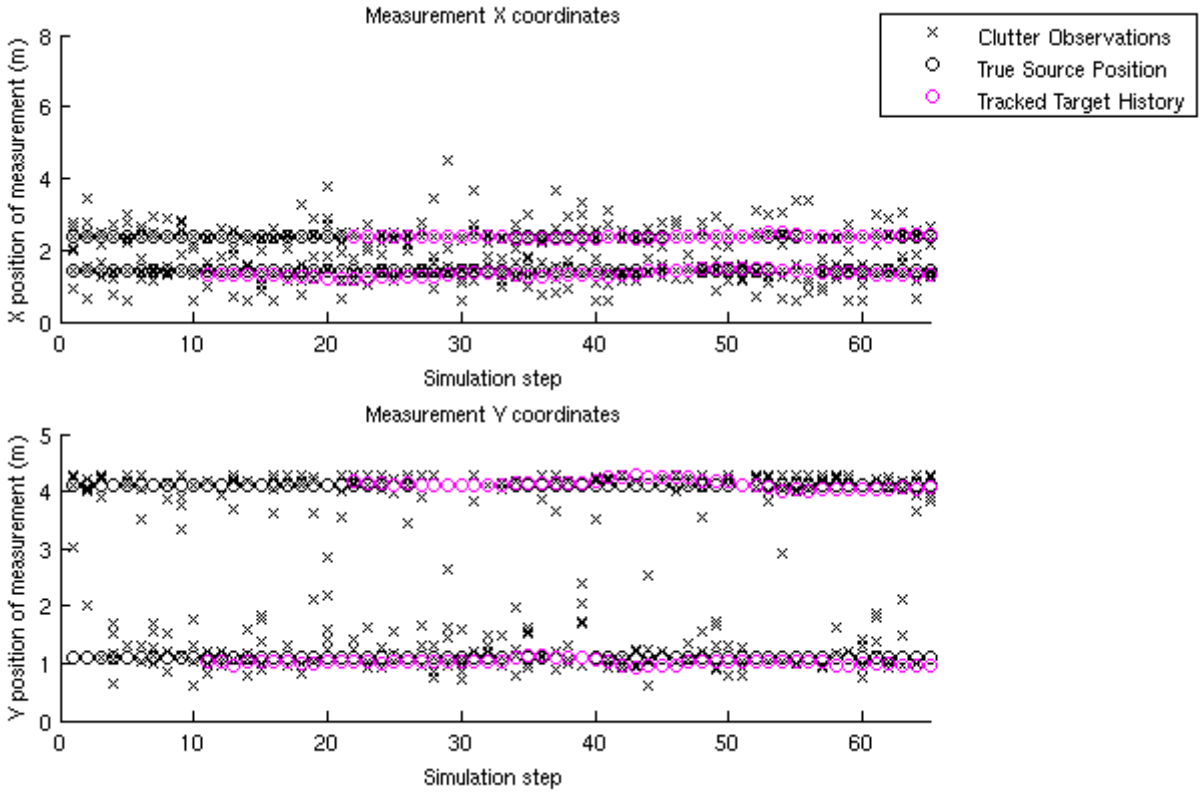


Figure 7.6: Measurements and tracks over time for 2 speakers from the Locust Swarms algorithm input to the tracker.

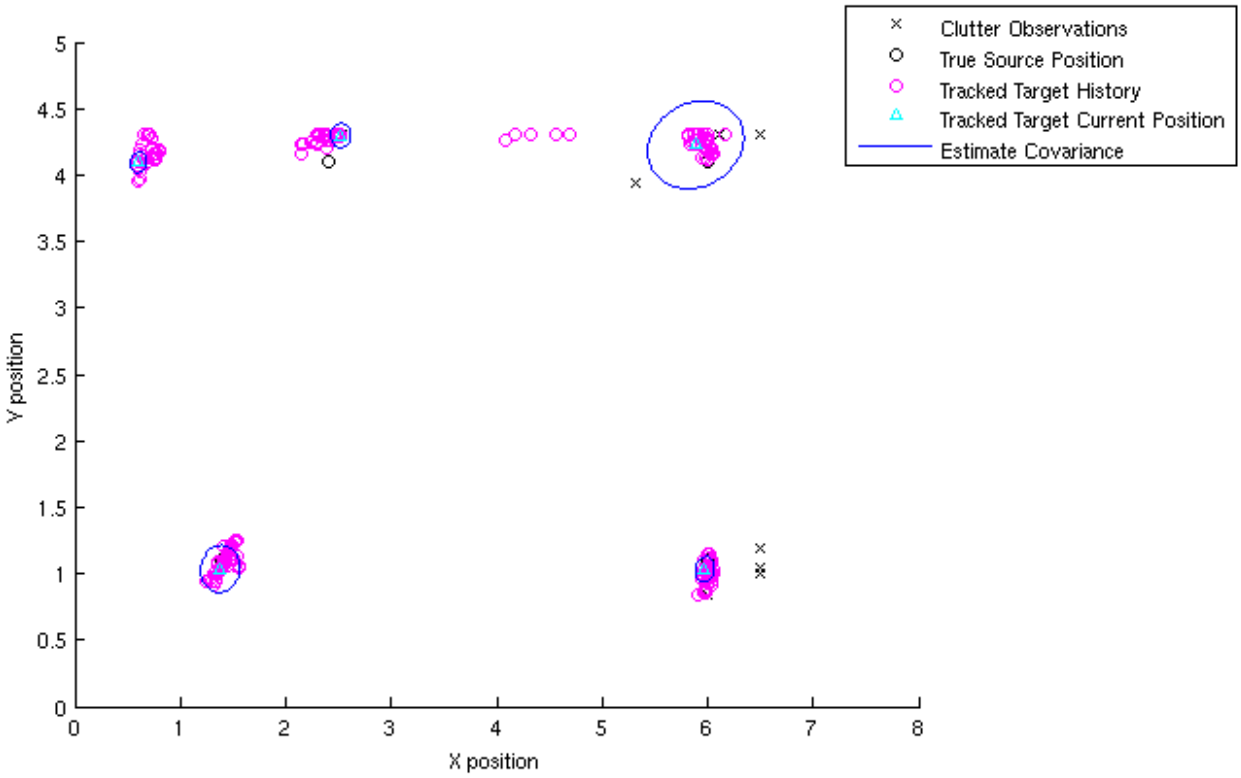


Figure 7.7: Horizontal-plane filter output for 5 speakers from the Locust Swarms algorithm input to the tracker.

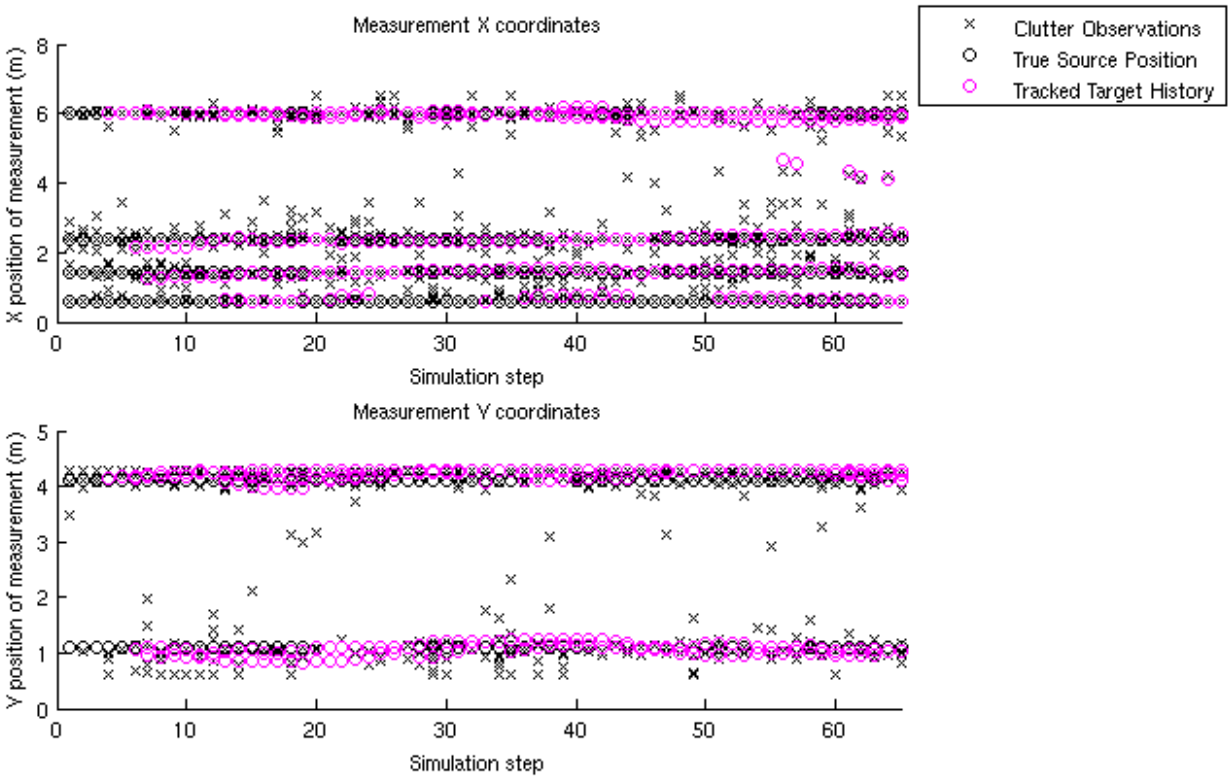


Figure 7.8: Measurements and tracks over time for 5 speakers from the Locust Swarms algorithm input to the tracker.

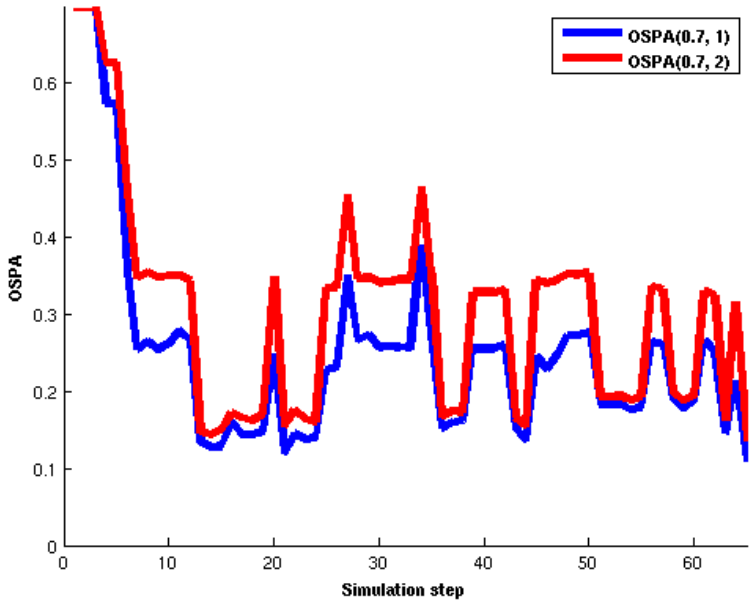


Figure 7.9: OSPA over time for 5 speakers from the Locust Swarms algorithm input to the tracker.

down as different sources change from being considered as true sources and noisy observations. This corresponds in part to gaps between speech sections, but also to the nature of the algorithm, as it discards individual tracked sources which it has less confidence in over time.

The more general effect of filtering on the OSPA metrics is shown in Figure 7.10. This graph shows both OSPA metrics considered after being averaged across each time step of each Monte Carlo trial. This is plotted against the number of speakers in each case, and the result is similar to that shown in Figure 6.8. Note that the trend is roughly the same, but the filtered results consistently score lower than their unfiltered counterparts, as might be expected.

7.4.2.2 Moving Targets

Part of the two-speaker test set included a set of two speakers moving past each other, as described in Section 7.4.1. Figures 7.11 and 7.12 are examples of the tracker output for these trials, which primarily demonstrates the ability of the filter to deal with moving sources, rather than the ability of the localisation technique to return usable results. Figure 7.11 shows the tracker history of the moving sources having followed each speaker as they walk past each other, with the estimates of their positions at the final frame shown.

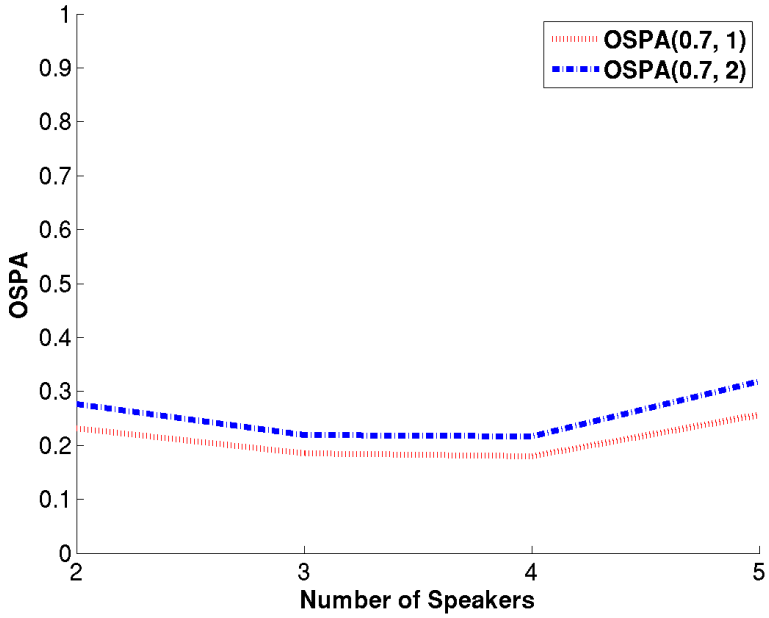


Figure 7.10: Mean OSPA vs number of speakers from the Locust Swarms algorithm inputs to the tracker.

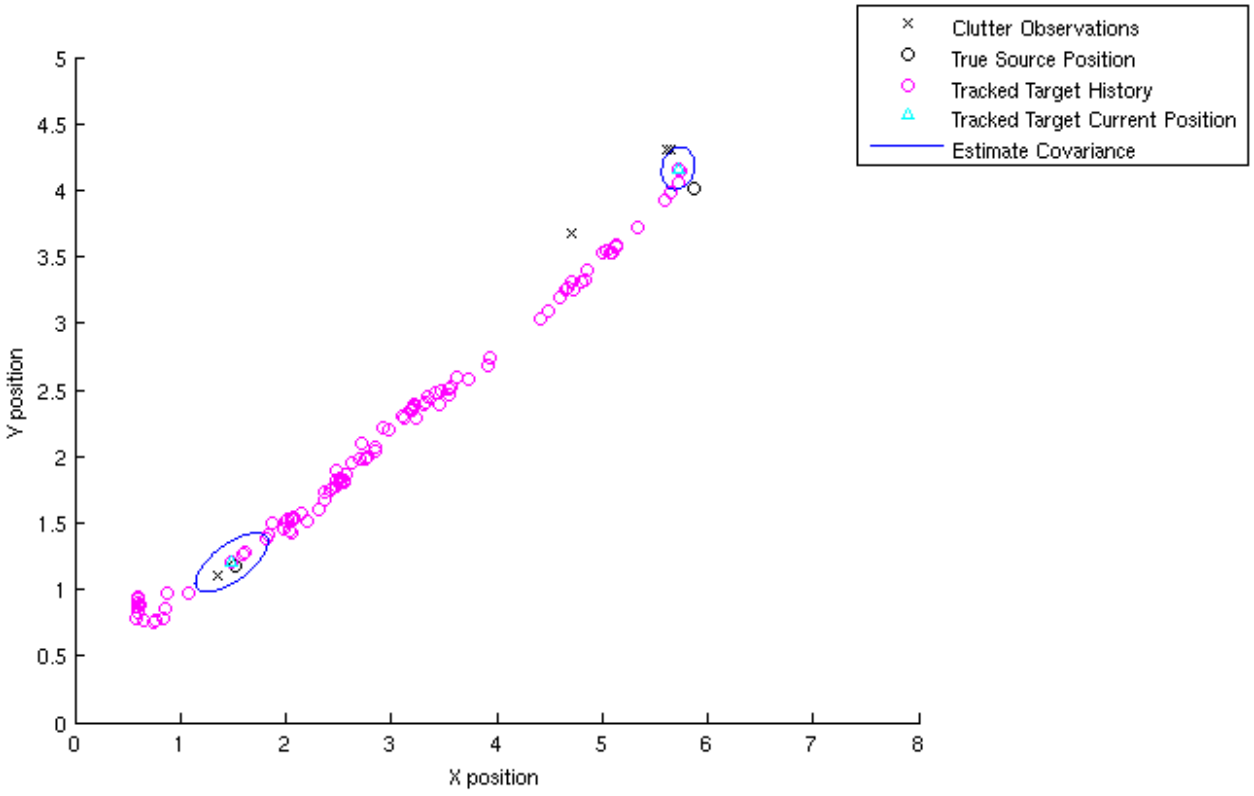


Figure 7.11: Horizontal-plane filter output for 2 moving speakers from the Locust Swarms algorithm input to the tracker.

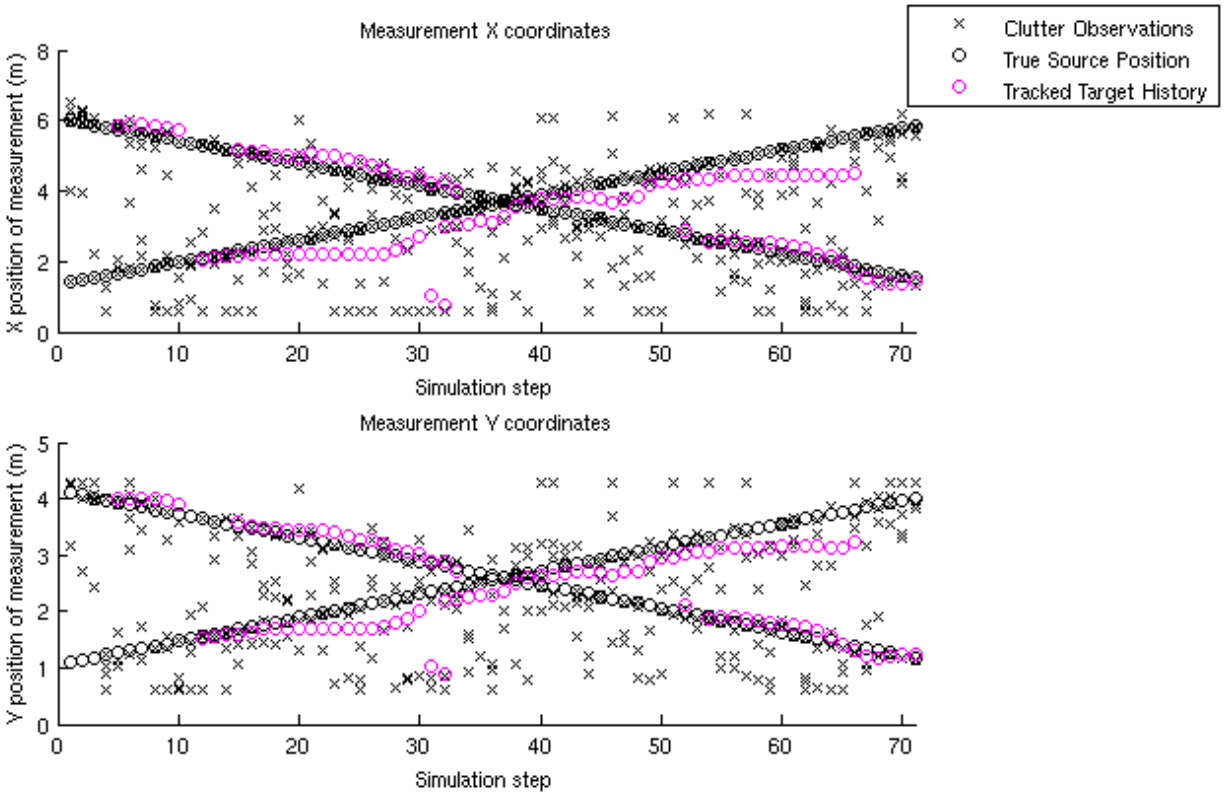


Figure 7.12: Measurements and tracks over time for 2 moving speakers from the Locust Swarms algorithm input to the tracker.

Similarly, Figure 7.12 shows how the tracked positions evolved over time for the trial run. As in the stationary speaker case, there is a delay before the tracker registers the sources. The tracked positions do not line up quite so well as in the stationary speaker case, and there is also a length of time when the trackers meet where the tracked position of one of the speakers is lost. This is not too surprising - because the data used is simulated, it briefly breaks the requirement that speakers cannot overlap each other. When they meet, the tracked states merge, and a new second speaker is born some time later, after they have moved apart.

This experiment shows the importance of the filter set up. Whilst the localisation method detected and returned maxima in exactly the same way as in the stationary speakers case, the adherence of the tracked positions to the known speaker path is visually less strong than that of the non-moving tracked speakers.

Ultimately, this means that for a working system, the filter needs to be carefully tuned. Altering the target spawning parameters of the filter might be one way to do this, as it would quickly

allow merged paths to split again, removing the loss of one of the speaker tracks as they pass each other. This might also result in more spurious detections of speakers, however. Similarly, the target dynamics model used is known to be an important factor for target tracking [55], and the simple linear Gaussian movement model used could be replaced with a more detailed model, such as the Langevin model, in order to (potentially) improve results.

7.4.3 Niching Particle Swarm Optimisation Results

The Niching algorithm combined the GM-PHD filter yielded mixed results. Whilst the 2 speaker performance generally looked good, as shown in Figure 7.13, they were inconsistent across trials and performed worse with a higher number of speakers.

Figure 7.13 presents the filter output over the horizontal-plane after the final acoustic frame has been processed. Note that the clutter is much more uniformly distributed over the horizontal-plane than in the Locust Swarms case. This effectively shows that the Niching PSO algorithm, despite its intentional clutter and weak convergence criteria, can be used to successfully extract speaker positions.

As for the Locust Swarms algorithm, the plot over time of the tracker measurements and detected tracks is shown in Figure 7.14. This shows a similar delay in picking up sources, but also shows incorrect tracks being picked up over time, not corresponding to any true source.

These spurious tracks occur frequently, affecting every Monte Carlo trail to a greater or lesser degree. Coupled with missed sources, this prevents the algorithm from ever scoring well on the OSPA metric, and Figure 7.15 shows how this score changes with the number of speakers. These per-speaker averages are calculated over the results of each audio segment over each Monte Carlo trial, as in Figure 7.10. Note that both scores are consistently barely below the maximum error of 0.7, so whilst the method can extract speakers, it doesn't do so reliably, resulting in a poor average localisation score.

The nature of this relatively poor performance is demonstrated in Figures 7.16 and 7.17. These figures show the tracker output at the final audio frame for the 4 simultaneous speaker scenario. Note that in Figure 7.16, the tracker has detected 6 sources when there are only four speakers, and that whilst some of the tracked positions correspond well to the known speaker locations, many of the ovals surrounding the tracked points are large, indicating a low confidence in their accuracy. Also, whilst some of the tracked sources are approximately centred on known

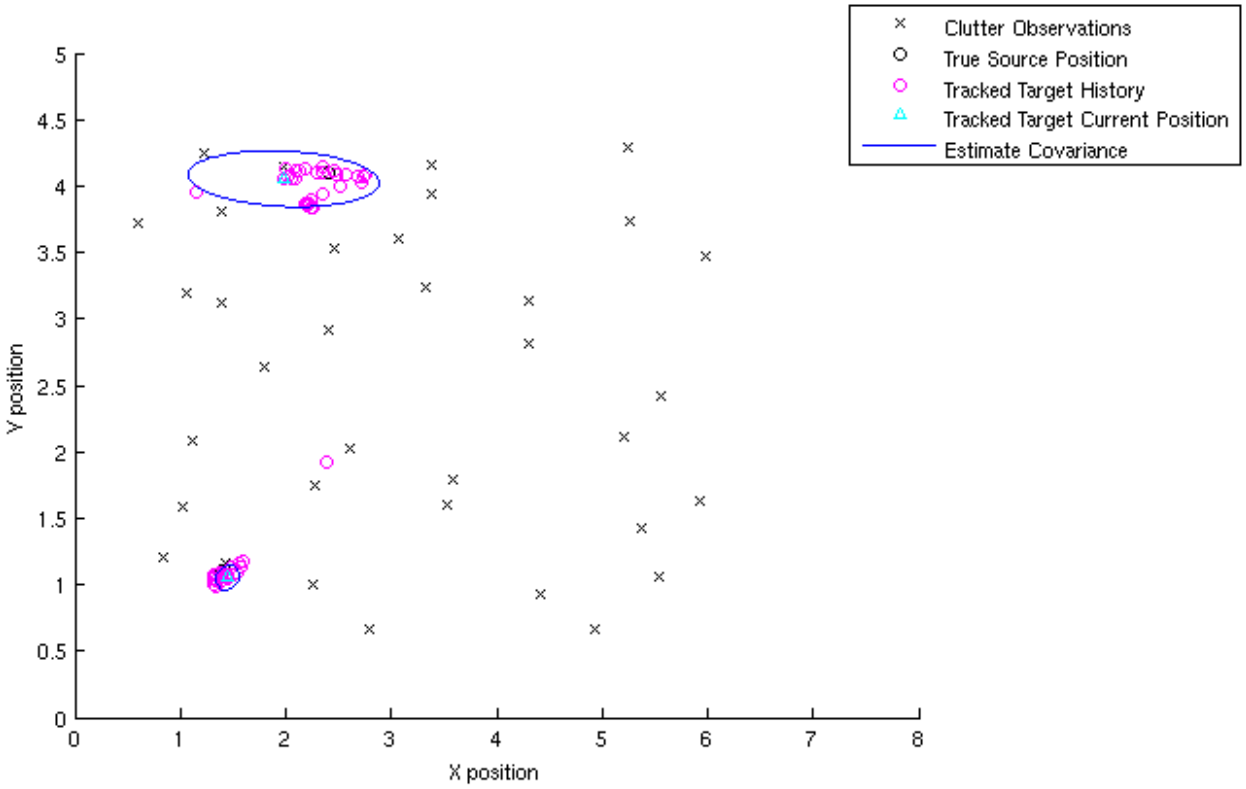


Figure 7.13: Horizontal-plane filter output for 2 speakers from the Niching PSO algorithm input to the tracker.

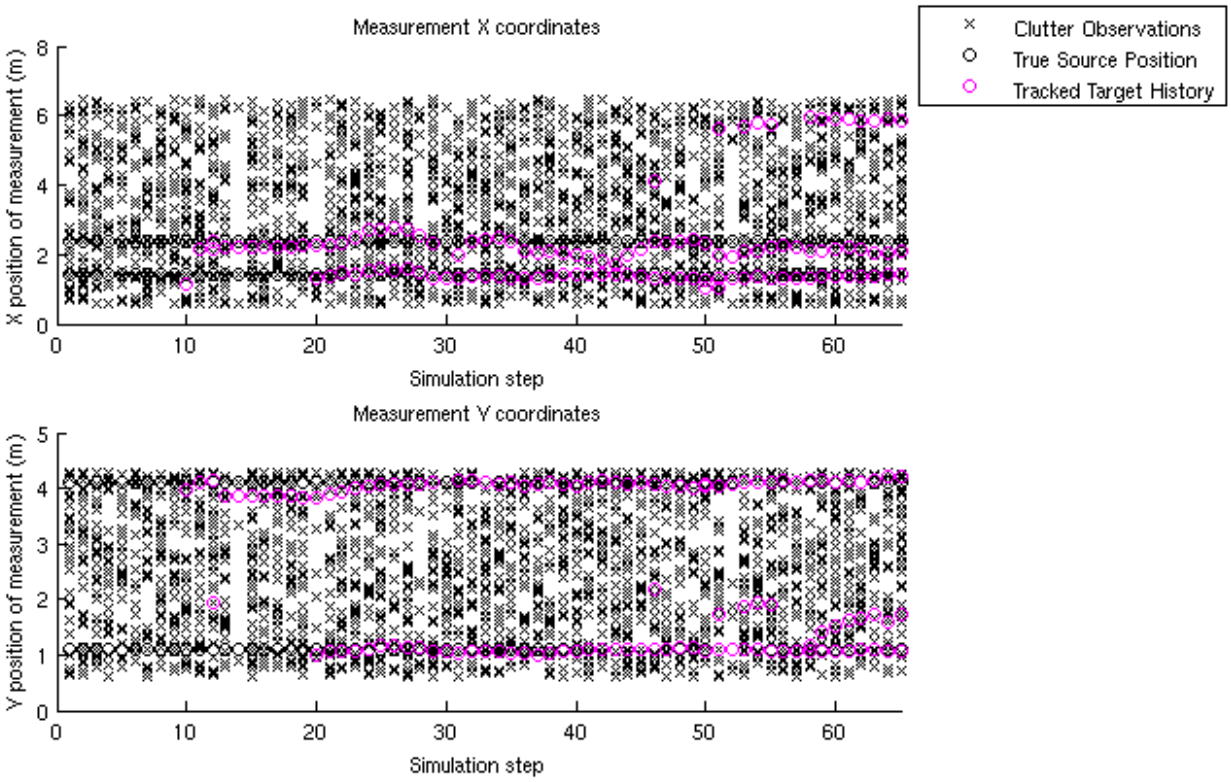


Figure 7.14: Measurements and tracks over time for 2 speakers from the Niching PSO algorithm input to the tracker.

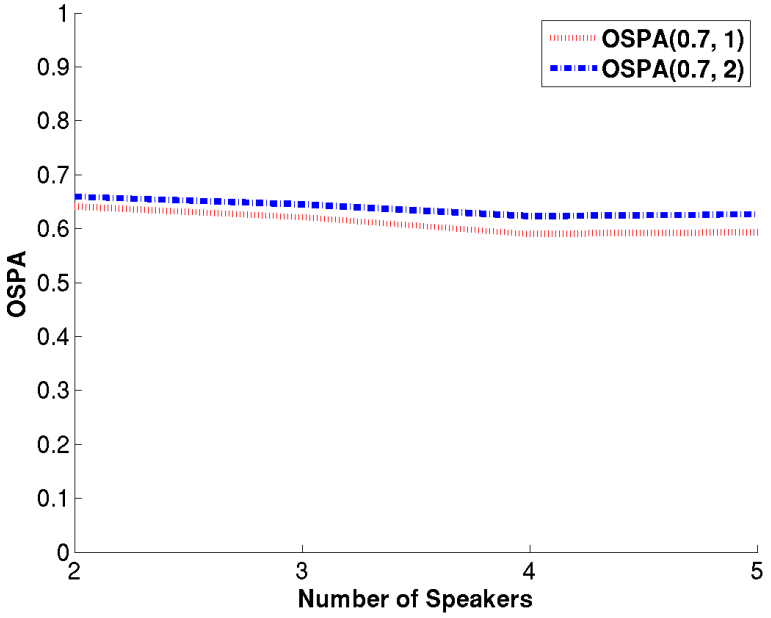


Figure 7.15: Mean OSPA vs number of speakers from the Niching PSO algorithm inputs to the tracker.

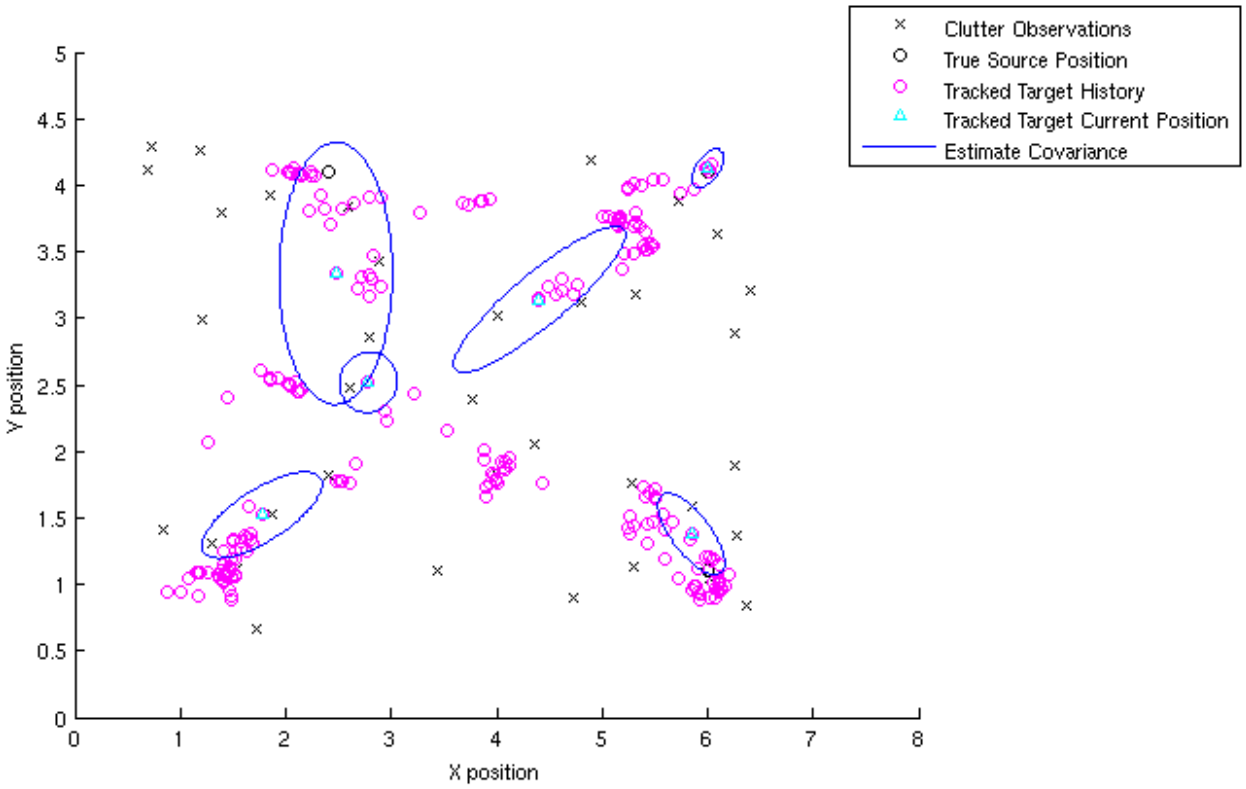


Figure 7.16: Horizontal-plane filter output for 4 speakers from the Niching PSO algorithm input to the tracker.

sources, others are only *close* to targets, giving a large localisation error even if they do truly correspond to known sources.

Finally, Figure 7.17 tracks the same example of the 4 speaker tracker output over time. It can be seen that some sources are correctly picked up, however there are many spurious tracks detected too.

Whilst these results are disappointing, there is unused information in the system, of which a modified filter might make use. It is suggested that future research investigate the possibility of using the SRP output at the localised optima to modify the trackers internal weightings in some way. This is envisaged to be conceptually similar to the use of the SRP as a pseudo-likelihood function for a particle filter approach as described by Lehmann and Williamson [22].

Note, however, that the complexity of the tracking algorithm will be larger in the case of the Niching PSO input than in the case of the Locust Swarms input, due to the increased amount of clutter. Investigation will have to be made into whether or not this increase in complexity

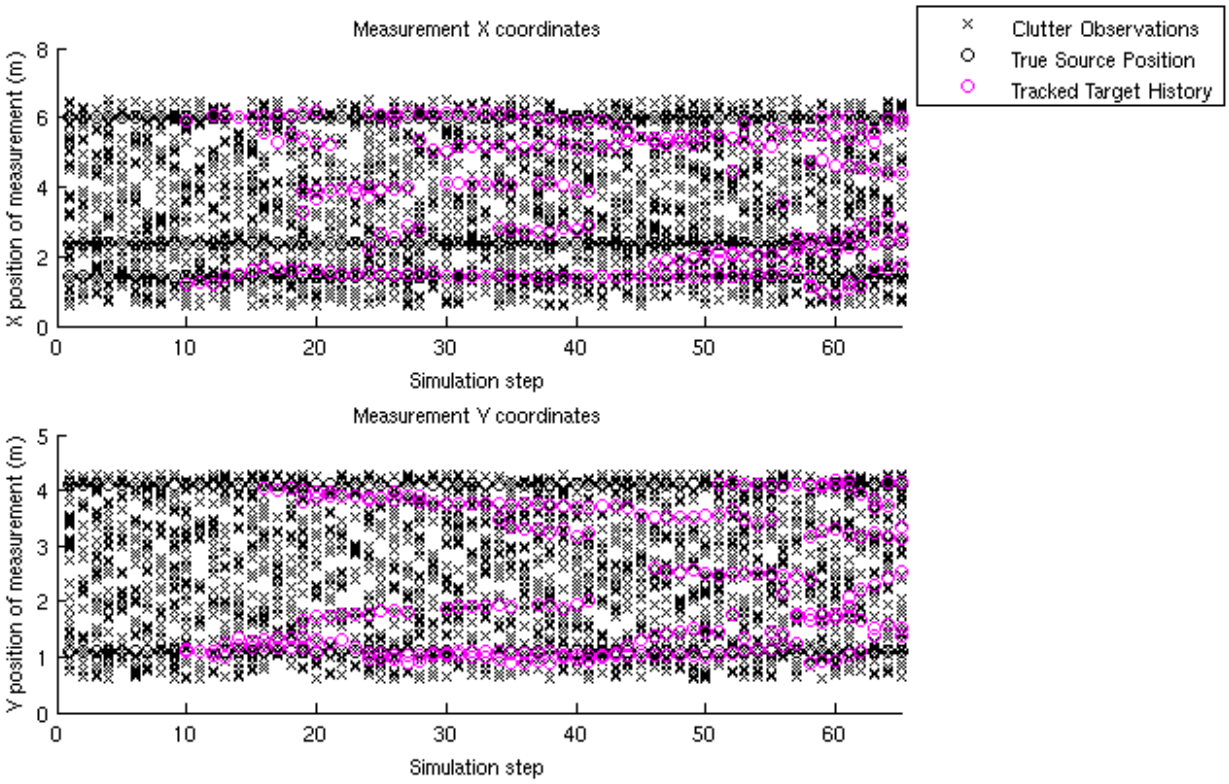


Figure 7.17: Measurements and tracks over time for 4 speakers from the Niching PSO algorithm input to the tracker.

offsets the potential complexity improvements gained by using the Niching PSO algorithm as a localiser compared to the Locust Swarms algorithm.

7.5 Conclusions

This Chapter has contributed the novel combination of computationally efficient multi-optima PSO techniques with the computationally tractable GM-PHD filter to create a system which can track a relatively large number of acoustic sources efficiently. This is in contrast to contemporary techniques, which struggle with both the extraction of an arbitrary number of sources as well as rapidly increasing computational cost for each additional target considered [6, 7].

Two separate multi-optima techniques were modified to fit the requirements of the filter, and tested for their ability to produce output which the tracking algorithm could use to successfully extract target states. The algorithms differed in the number of FEs required to localise a number of sources as well as the amount of clutter observations they produced. The more computationally expensive Locust Swarms algorithm produced a very low level of clutter, and the tracker could use this data to localise many sources. This led to a consistent improvement of the localisation error as measured by the OSPA metric.

In contrast, the lower complexity and high clutter algorithm, Niching PSO, was found to produce usable results in the case of two stationary source. However, the tracker generally detected spurious tracks when used with this localisation algorithm, consistently lowering the OSPA metric and making this method unsuitable for general use. However, the technique still holds potential, and further research directions have been identified which could result in a very efficient tracker for a large number of speakers.

Chapter 8
Conclusions

This Chapter summarises the main findings and contributions of the thesis. Consideration is also given to further research which could be undertaken following on from this work. The primary contributions consist of improved single-source localisation techniques in terms of computational performance; a novel approach to multiple speaker localisation by the use of the SRP as the objective function for a family of existing multi-optima search techniques; and finally the novel application of the GM-PHD multi-target filter to the output of the developed localisation techniques.

8.1 Conclusions

8.1.1 Improved Stochastic Region Contraction

The SRC single-source audio localisation technique was developed to provide an improvement in both localisation accuracy and computational requirements. This was achieved by taking account of the nature of the search space. Specifically, speaker height information was used to speed up the search by effectively limiting the search dimensions in the space around speakers detected in a previous audio frame. This was in contrast to a typical speaker localisation assumption, which forced the search over the vertical axis to evaluate the objective function uniformly over height. This was altered by modelling the likely height of a speaker at any point within a room as a mixture model, consisting of a uniform distribution over the entire vertical axis and a Gaussian distribution centred on an estimated head-height.

The variance of the Gaussian distribution for an area of space within the search volume was increased the further away from a previously detected speaker that space was. This allowed the algorithm to approximate the uniform vertical distribution used by the original SRC algorithm when there was no information available about any potential nearby sources.

This section of work also addressed the problem of non-concurrent multiple speakers, by allowing the head height to be estimated over the entire horizontal-plane by interpolating between

previously detected speakers. This also allowed a video system to be used to help speed up the acoustic search, which was achieved by the same interpolation process. Instead of using a single previously detected height however, the heights of potential speakers within a room extracted from a camera system were used.

8.1.2 Particle Swarm Optimisation for Single Source Localisation

Further computational efficiency gains for the localisation of a single acoustic source by the application of PSO techniques to the SRP objective function. This simple search algorithm was used to provide a localisation routine which was robust to noise. The SRP function generally contains many small peaks simply generated by slight differences in the sums of the underlying correlation measure, as well as larger peaks corresponding to acoustic sources. These small peaks mean that more common gradient based optimisation techniques are not suitable for the task of speaker localisation. The PSO technique was used specifically because it did not use the gradient of the objective function, allowing it to move past these noisy peaks. This provided a system with low localisation error and a significant drop in computational requirements compared to the state of the art.

8.1.3 Multi-Optima Localisation

The PSO search strategy on the SRP objective function was expanded to perform a multi-optima search. The SRP was identified as an excellent candidate for use as a multi-speaker search space, because peaks of the function correspond directly to speaker locations, and multiple speakers produce multiple distinct peaks. Several variants of multi-optima PSO search techniques were explored, resulting in the identification of a localisation technique which could extract an arbitrary number of source positions with a linear increase in the number of FEs required for each additional speaker to be localised.

Also identified were some of the general problems which face a multi-optima localisation technique using the SRP objective function. For searches attempting to identify sources in parallel, the process of niching is made problematic by the presence of noisy optima, which are easily identified by the niching process, but are not relevant as useful localisation results. When niche areas have been identified, it can take a long time for the local searches in those areas to converge, because whilst they do contain small local peaks, those peaks are not very large in

magnitude and swarms regularly find slightly better local peaks in their local search areas.

8.1.4 Multi-Optima Tracking

Two of the localisation techniques developed to find multiple speakers were modified to produce output which could be used by the GM-PHD multi-target filter. This filtering technique and localisation method combination was identified as good match because the direct observations of potential source locations matched the required linear observation framework of the filter. Additionally, the filter is a relatively efficient solution for multi-target tracking, so should be able to cope with tracking a relatively large number of speakers.

One of the localisation techniques combined with the filter produced a consistently reduced localisation error, as measured by the OSPA metric. The other localisation technique was found to be capable of identify speaker locations in conjunction with the filter, despite a necessarily high level of clutter in the array of observations. However, potential research has been identified which could be undertaken to improve these results and produce a highly efficient tracking system.

8.2 Limitations

There are, of course, a number of limitations in this thesis. One of the major limitations is the inability to properly test the ability of the tracker to cope with more than five simultaneous speakers. This was due to the very short speech signals used for the sixth and seventh input signals. Whilst the localisation algorithms produced reasonable output on these signals, they were not long enough for the tracker to pick up on the sources which they represented. Whilst the tracker remains untested with a larger number of speakers, the tests on up to five simultaneous speakers don't suggest that the tracker will suddenly stop working. Further research might aim to identify the limits of both the localisation algorithms and their combination with the GM-PHD filter in terms of number of speakers which can be identified and tracked.

Another major limitation is the limited use of recorded acoustic data. Speakers in four separate positions were recorded and used to create sets of simultaneous speakers. Recordings of further speakers in different positions would have allowed further useful evaluation, especially as the acoustic lab is a highly reverberant environment, making for a challenging test set. Furthermore,

no recordings were made of moving sources. Note however, that moving sources for evaluation purposes have their own difficulty, in that it is hard to know the position of a real moving source at any given time, and therefore audio frame.

Another limitation is that height information was not used in the multiple target localisation algorithms. Whilst height information could have been used to influence the Locust Swarms algorithm, this would only have affected the first iteration of the algorithm, as subsequent iterations do not re-initialise the particle swarm randomly. Height information might be more applicable to the Niching PSO algorithm as it would affect the whole swarm, thereby influencing all of the created sub-swarms which localise peaks of the objective function in parallel.

8.3 Suggestions for Future Work

Future improvements could be made to the systems developed to produce more efficient acoustic source localisation and tracking techniques. Of particular interest is further investigation into the most challenging task considered in this thesis, the case of localising and tracking multiple concurrent speakers. This section details further work which could be undertaken following on from the results of this thesis.

8.3.1 Niching Particle Swarm Optimisation Development

The Niching PSO developed has been shown to be capable of producing results which can be used by a tracking algorithm to extract source locations. However, whilst the computational performance of this algorithm is good, the localisation results are poor. It would be interesting to investigate the modification of the tracking algorithm to take into account the objective function values at the positions returned by the Niching PSO localiser. These might be used to influence the weight of the observations internal to the tracker, similar to the use of the SRP as a pseudo-likelihood function in [11], in the hopes of making source observations easier to identify amongst clutter observations.

8.3.2 Exploitation of Parallelism

Whilst SBF based localisations are costly to evaluate, FEs are largely independent, as each PSO algorithm updates the positions of the whole swarm every iteration. In this case, the

individual FEs at each iteration could easily be performed in parallel. Modern GPUs are capable of performing parallel processing using a large number of simple cores. An example of such a system is the Nvidia CUDA framework.

Parallelisation would be particularly useful for a multi-speaker tracking system, especially for the parallel Niching PSO search, which uses large swarm sizes. Whilst some work [82, 83] has already been done on a GPU based SRP source localisation algorithm, it is suggested that further research focus on implementing the multi-optima PSO localisation techniques on a GPU. This has the potential to provide an online acoustic tracking system which might run comfortably on a suitably equipped modern desktop computer.

8.3.3 Multi-source Tracking Techniques

Investigation should also be made into alternative tracking techniques. For example, the Cardinalised GM-PHD filter is a modified version of the GM-PHD filter which might be evaluated against the standard GM-PHD filter using the localisation algorithms developed. The GM-PHD filter itself could also be evaluated with the use of a uniform birth prior over the search space [113], as opposed to the Gaussian-mixture approximation used in this thesis. Finally, investigation should be made into using a camera system to seed a Gaussian-mixture birth prior based on the position of potential speakers within a room, which could be found using the localisation of people in the visual domain.

References

[1] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the Acoustical Society of America*, vol. 25, no. 5, pp. 975–979, 1953.

[2] Y. Cao, S. Sridharan, and M. Moody, "Speech separation by simulating the cocktail party effect with a neural network controlled wiener filter," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 4, pp. 3261–3264 vol.4, Apr 1997.

[3] I. Peer, B. Rafaely, and Y. Zigel, "Room acoustics parameters affecting speaker recognition degradation under reverberation," in *Hands-Free Speech Communication and Microphone Arrays, 2008. HSCMA 2008*, pp. 136–139, May 2008.

[4] P. Viola and M. Jones, "Robust real-time object detection," in *International Journal of Computer Vision*, 2001.

[5] I. McCowan, "Microphone Arrays: A Tutorial," tech. rep., Queensland University of Technology, Australia, 2001.

[6] X. Zhong, *A Bayesian framework for multiple acoustic source tracking*. PhD thesis, The University of Edinburgh, 2010.

[7] M. Fallon, *Acoustic source tracking using sequential Monte Carlo*. PhD thesis, The University of Cambridge, 2008.

[8] U. Klee, T. Gehrig, and J. McDonough, "Kalman filters for time delay of arrival-based source localization," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 167–167, 1 2006.

[9] S. Ohmori and K. Suyama, "Multiple moving sound source tracking using pso," in *Signal Processing (ICSP), 2012 IEEE 11th International Conference on*, vol. 1, pp. 132–135, Oct 2012.

[10] R. Parisi, P. Croene, and A. Uncini, "Particle swarm localization of acoustic sources in the presence of reverberation," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, pp. 4 pp.–4742, May 2006.

[11] D. B. Ward, E. Lehmann, and R. Williamson, "Particle filtering algorithms for tracking an acoustic source in a reverberant environment," *Speech and Audio Processing, IEEE Transactions on*, vol. 11, pp. 826–836, Nov 2003.

[12] E. Antonacci, D. Riva, A. Sarti, M. Tagliasacchi, and S. Tubaro, "Tracking of two acoustic sources in reverberant environments using a particle swarm optimizer," in *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pp. 567–572, Sept 2007.

[13] H. Do, H. Silverman, and Y. Yu, "A real-time SRP-PHAT source location implementation using Stochastic Region Contraction (SRC) on a large-aperture microphone array," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 1, pp. I–121 –I–124, 4 2007.

[14] H. Do and H. F. Silverman, "A fast microphone array srp-phat source location implementation using coarse-to-fine region contraction(cfrc)," in *Applications of Signal Processing to Audio and Acoustics, 2007 IEEE Workshop on*, pp. 295 –298, oct. 2007.

[15] D. Reid, "An algorithm for tracking multiple targets," *Automatic Control, IEEE Transactions on*, vol. 24, pp. 843–854, Dec 1979.

[16] J. Lim, "The joint probabilistic data association filter (jpdaf) for multi-target tracking," tech. rep., State University of New York at Stony Brook, 2006.

[17] R. Mahler, ""statistics 102" for multisource-multitarget detection and tracking," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 7, pp. 376–389, June 2013.

[18] T. Wood, "Random finite set theory for tracking," tech. rep., University of Bremen, 2010.

[19] B.-N. Vo and W.-K. Ma, "The gaussian mixture probability hypothesis density filter," *Signal Processing, IEEE Transactions on*, vol. 54, pp. 4091–4104, Nov 2006.

[20] M. F. Fallon and S. J. Godsill, "Acoustic source localization and tracking of a time-varying number of speakers," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, pp. 1409 –1415, may 2012.

[21] M. Fallon, "Multi target acoustic source tracking with an unknown and time varying number of targets," in *Hands-Free Speech Communication and Microphone Arrays, 2008. HSCMA 2008*, pp. 77–80, May 2008.

[22] 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 Advances in Signal Processing*, vol. 2006, no. 1, p. 017021, 2006.

[23] X. Zhong, A. Premkumar, and A. S. Madhukumar, "Particle filtering and posterior Cramér-Rao bound for 2-d direction of arrival tracking using an acoustic vector sensor," *Sensors Journal, IEEE*, vol. 12, pp. 363–377, Feb 2012.

[24] Y. Oualil, M. Magimai-Doss, F. Faubel, and D. Klakow, "A probabilistic framework for multiple speaker localization," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 3962–3966, May 2013.

[25] M. I. Mandel, D. P. Ellis, and T. Jebara, "An em algorithm for localizing multiple sound sources in reverberant environments," in *Advances in Neural Information Processing Systems 19* (B. Schölkopf, J. Platt, and T. Hoffman, eds.), pp. 953–960, MIT Press, 2007.

[26] A. Lombard, T. Rosenkranz, H. Buchner, and W. Kellermann, "Multidimensional localization of multiple sound sources using averaged directivity patterns of blind source separation systems," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pp. 233–236, April 2009.

[27] A. Lombard, Y. Zheng, H. Buchner, and W. Kellermann, "Tdoa estimation for multiple sound sources in noisy and reverberant environments using broadband independent component analysis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, pp. 1490–1503, Aug 2011.

[28] A. Lombard, Y. Zheng, and W. Kellermann, "Synthesis of ica-based methods for localization of multiple broadband sound sources," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 157–160, May 2011.

[29] N. Chong, S. Wong, B.-T. Vo, N. Sven, and I. Murray, "Multiple moving speaker tracking via degenerate unmixing estimation technique and cardinality balanced multi-target multi-bernoulli filter (duet-cbmember)," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pp. 1–6, April 2014.

[30] D. Halliday, R. Resnick, and J. Walker, *Fundamentals of Physics*. Wiley, 7th extended ed., 2005.

[31] G. S. Kino, *Acoustic Waves: Devices, Imaging and Analog Signal Processing*. Prentice-Hall, 2000.

[32] P. Buser and M. Imbert, *Audition*. A Bradford book, MIT Press, 1992.

[33] F. Jacobsen, *The diffuse sound field : statistical considerations concerning the reverberant field in the steady state*. Lyngby : Acoustics Laboratory, Technical University of Denmark, 1979.

[34] H. E. Bass, H.-J. Bauer, and L. B. Evans, "Atmospheric absorption of sound: Analytical expressions," *Journal of the Acoustical Society of America*, vol. 52, 1972.

[35] H. Kuttruff, *Room Acoustics, Fifth Edition*. Taylor & Francis, 2009.

[36] L. Kinsler, *Fundamentals of acoustics*. Wiley, 2000.

[37] J. Allen and D. Berkley, "Image method for efficiently simulating small-room acoustics," *Journal of the Acoustical Society of America*, vol. 65, pp. 943–950, 1978.

[38] A. Primavera, S. Cecchi, L. Romoli, P. Peretti, and F. Piazza, "Approximation of real impulse response using IIR structures," in *Proc. 19th "European Signal Processing Conference"*, (Barcelona, Spain), Aug. 2011.

[39] E. A. Lehmann and A. M. Johansson, "Prediction of energy decay in room impulse responses simulated with an image-source model," *The Journal of the Acoustical Society of America*, vol. 124, no. 1, pp. 269–277, 2008.

[40] E. Lehmann and A. Johansson, "Diffuse reverberation model for efficient image-source simulation of room impulse responses," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, pp. 1429 –1439, Aug. 2010.

[41] B. Champagne, S. Bedard, and A. Stephenne, "Performance of time-delay estimation in the presence of room reverberation," *Speech and Audio Processing, IEEE Transactions on*, vol. 4, pp. 148–152, Mar 1996.

[42] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 24, pp. 320 – 327, Aug. 1976.

[43] M. Brandstein and H. Silverman, "A robust method for speech signal time-delay estimation in reverberant rooms," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 1, pp. 375–378 vol.1, 4 1997.

[44] M. Brandstein, J. Adcock, and H. Silverman, "A closed-form location estimator for use with room environment microphone arrays," *Speech and Audio Processing, IEEE Transactions on*, vol. 5, pp. 45–50, Jan 1997.

[45] J. Lewis, *AN-1140 - Microphone Array Beamforming*. Analog Devices.

[46] M. Kajala and M. Hamaldinen, "Broadband beamforming optimization for speech enhancement in noisy environments," in *Applications of Signal Processing to Audio and Acoustics, 1999 IEEE Workshop on*, pp. 19–22, 1999.

[47] H. Cox, R. Zeskind, and T. Kooij, "Practical supergain," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, pp. 393–398, Jun 1986.

[48] S. T. Birchfield, "A Unifying Framework for Acoustic Localization," in *Proceedings of the 12th European Signal Processing Conference (EUSIPCO)*, Sept. 2004.

[49] E. A. Lehmann, *Particle Filtering Methods for Acoustic Source Localisation and Tracking*. PhD thesis, Research School of Information Sciences and Engineering, Department of Telecommunications Engineering, The Australian National University, Canberra, ACT, Australia, July 2004.

[50] M. Berger and H. Silverman, "Microphone array optimization by stochastic region contraction," *Signal Processing, IEEE Transactions on*, vol. 39, no. 11, pp. 2377–2386, 1991.

[51] J. Benesty, "Adaptive eigenvalue decomposition algorithm for passive acoustic source localization," *J. Acoust. Soc. Am*, vol. 107, pp. 384–391, Jan. 2000.

[52] W. Cui, Z. Cao, and J. Wei, "Dual-microphone source location method in 2-d space," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 4, pp. IV–IV, May 2006.

[53] D. Li and S. Levinson, "A bayes-rule based hierarchical system for binaural sound source localization," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 5, pp. V–521–4 vol.5, April 2003.

[54] X. Li and V. Jilkov, "Survey of maneuvering target tracking. part i. dynamic models," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 39, pp. 1333–1364, Oct 2003.

[55] E. A. Lehmann, A. M. Johansson, and S. Nordholm, "Modeling of motion dynamics and its influence on the performance of a particle filter for acoustic speaker tracking," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'07)*, (New Paltz, NY, USA), pp. 98–101, October 2007.

[56] J. Vermaak and A. Blake, "Nonlinear filtering for speaker tracking in noisy and reverberant environments," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 5, pp. 3021–3024, IEEE, 2001.

[57] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[58] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, Feb 1989.

[59] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[60] B. Anderson and J. Moore, *Optimal Filtering*. Dover Books on Electrical Engineering, Dover Publications, 2012.

[61] G. Welch and G. Bishop, "An introduction to the kalman filter," tech. rep., University of North Carolina, Chapel Hill, NC, USA, 1995.

[62] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

[63] M. Wölfel and J. McDonough, *Distant Speech Recognition*. Wiley, 2009.

[64] P. Kaminski, A. E. Bryson, and S. Schmidt, "Discrete square root filtering: A survey of current techniques," *Automatic Control, IEEE Transactions on*, vol. 16, pp. 727–736, Dec 1971.

[65] C. L. Thornton and NASA, *Triangular Covariance Factorizations for Kalman Filtering: National Aeronautics and Space Administration. Technical Memorandum 33-798*. NASA-CR-149 147, University of California, 1976.

[66] M. Mallick, V. Krishnamurthy, and B. ngu Vo, *Integrated Tracking, Classification, and Sensor Management*. Wiley, 2013.

[67] B.-N. Vo, S. Singh, and A. Doucet, "Sequential monte carlo methods for multitarget filtering with random finite sets," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 41, pp. 1224–1245, Oct 2005.

[68] B.-N. Vo, S. Singh, and A. Doucet, "Sequential monte carlo implementation of the phd filter for multi-target tracking," in *Information Fusion, 2003. Proceedings of the Sixth International Conference of*, vol. 2, pp. 792–799, July 2003.

[69] K. Panta, D. Clark, and B.-N. Vo, "Data association and track management for the gaussian mixture probability hypothesis density filter," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 45, pp. 1003–1016, July 2009.

[70] J. Dmochowski, J. Benesty, and S. Affes, "A generalized steered response power method for computationally viable source localization," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, pp. 2510 –2526, 11 2007.

[71] J. H. DiBiase, *A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments Using Microphone Arrays*. PhD thesis, Brown University, 2000.

[72] J. Chen, J. Benesty, and Y. Huang, "Time delay estimation in room acoustic environments: An overview," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, no. 1, p. 026503, 2006.

[73] H. Do and H. Silverman, "Stochastic particle filtering: A fast srp-phat single source localization algorithm," in *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA '09. IEEE Workshop on*, pp. 213–216, Oct 2009.

[74] F. Talantzis, A. Pnevmatikakis, and A. G. Constantinides, *Audio-Visual Person Tracking: A Practical Approach*. Imperial College Press, 2012.

[75] J. Dmochowski, J. Benesty, and S. Affes, "Fast steered response power source localization using inverse mapping of relative delays," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 289 –292, 31 2008-april 4 2008.

[76] R. E. Carlson and F. N. Fritsch, "An algorithm for monotone piecewise bicubic interpolation," *SIAM J. Numer. Anal.*, vol. 26, pp. 230–238, Feb. 1989.

[77] D. T. Lee and B. J. Schachter, "Two algorithms for constructing a Delaunay triangulation," *International Journal of Parallel Programming*, vol. 9, pp. 219–242, 1980. 10.1007/BF00977785.

[78] J. D'Errico, "inpaint_nans." MATLAB Central File Exchange, 2 2012.

[79] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, pp. 469–483, Dec. 1996.

[80] H. Silverman, Y. Yu, J. Sachar, and I. Patterson, W.R., "Performance of real-time source-location estimators for a large-aperture microphone array," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, pp. 593 – 606, 7 2005.

[81] E. D'Arca, A. Hughes, N. Robertson, and J. Hopgood, "Video tracking through occlusions by fast audio source localisation," in *2013 IEEE International Conference on Image Processing*, (Melbourne, Australia), Sept. 2013.

[82] L. G. da Silveira Jr, V. P. Minotto, C. R. Jung, and B. Lee, "A GPU implementation of the srp-phat sound source localization algorithm," in *Proc. 12th International Workshop on Acoustic Echo and Noise Control*, 2010.

[83] V. P. Minotto, C. R. Jung, L. G. da Silveira, and B. Lee, "GPU-based approaches for real-time sound source localization using the srp-phat algorithm," *International Journal of High Performance Computing Applications*, vol. 27, no. 3, pp. 291–306, 2013.

[84] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942–1948 vol.4, Nov 1995.

[85] M. Sami, N. El-Bendary, T.-H. Kim, and A. Hassanien, "Using particle swarm optimization for image regions annotation," in *Future Generation Information Technology* (T.-H. Kim, Y.-H. Lee, and W.-C. Fang, eds.), vol. 7709 of *Lecture Notes in Computer Science*, pp. 241–250, Springer Berlin Heidelberg, 2012.

[86] Z.-H. Liu, W. Qi, Z.-M. He, H.-L. Wang, and Y. Feng, "PSO-based Parameter Estimation of Nonlinear Kinetic Models for b-Mannanase Fermentation," *Chemical & Biochemical Engineering Quarterly*, vol. 2, pp. 195–201, 2008.

[87] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pp. 69–73, May 1998.

[88] L. Cheng, Y. Wang, and S. Li, "A self-adaptive particle swarm optimization based multiple source localization algorithm in binary sensor networks," *International Journal of Distributed Sensor Networks*, 2014.

[89] J. Bansal, P. Singh, M. Saraswat, A. Verma, S. Jadon, and A. Abraham, "Inertia weight strategies in particle swarm optimization," in *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on*, pp. 633–640, Oct 2011.

[90] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 3, pp. –1950 Vol. 3, 1999.

[91] J. Xin, G. Chen, and Y. Hai, "A particle swarm optimizer with multi-stage linearly-decreasing inertia weight," in *Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on*, vol. 1, pp. 505–508, April 2009.

[92] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *Evolutionary Computation, IEEE Transactions on*, vol. 6, pp. 58–73, Feb 2002.

[93] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317 – 325, 2003.

[94] S. Mikki and A. Kishk, "Improved particle swarm optimization technique using hard boundary conditions," *Microwave and Optical Technology Letters*, vol. 46, no. 5, pp. 422–426, 2005.

[95] J. Robinson and Y. Rahmat-Samii, "Particle swarm optimization in electromagnetics," *Antennas and Propagation, IEEE Transactions on*, vol. 52, pp. 397–407, Feb 2004.

[96] S. Mikki and A. Kishk, "Hybrid periodic boundary condition for particle swarm optimization," in *Antennas and Propagation Society International Symposium, 2007 IEEE*, pp. 1581–1584, June 2007.

[97] S. W. Mahfoud, *Niching Methods for Genetic Algorithms*. PhD thesis, University of Illinois, 1995.

[98] R. Brits, A. P. Engelbrecht, and F. V. D. Bergh, "A niching particle swarm optimizer," in *In Proceedings of the Conference on Simulated Evolution And Learning*, pp. 692–696, 2002.

[99] J. Kennedy, "The particle swarm: social adaptation of knowledge," in *Evolutionary Computation, 1997., IEEE International Conference on*, pp. 303–308, Apr 1997.

[100] F. van den Bergh, *An Analysis of Particle Swarm Optimizers*. PhD thesis, University of Pretoria, 2001.

[101] F. van den Bergh and A. Engelbrecht, "A new locally convergent particle swarm optimiser," in *Systems, Man and Cybernetics, 2002 IEEE International Conference on*, vol. 3, pp. 6 pp. vol.3–, Oct 2002.

[102] D. Beasley, D. R. Bull, and R. R. Martin, "A sequential niche technique for multimodal function optimization," *Evol. Comput.*, vol. 1, pp. 101–125, June 1993.

[103] T. Blackwell and J. Branke, "Multi-swarm optimization in dynamic environments," in *Applications of Evolutionary Computing* (G. Raidl, S. Cagnoni, J. Branke, D. Corne, R. Drechsler, Y. Jin, C. Johnson, P. Machado, E. Marchiori, F. Rothlauf, G. Smith, and G. Squillero, eds.), vol. 3005 of *Lecture Notes in Computer Science*, pp. 489–500, Springer Berlin Heidelberg, 2004.

[104] T. Hendtlass, "A particle swarm algorithm for high dimensional, multi-optima problem spaces," in *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, pp. 149–154, June 2005.

[105] T. Hendtlass, "WoSP: a multi-optima particle swarm algorithm," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 1, pp. 727–734 Vol.1, Sept 2005.

[106] S. Chen, "Locust swarms - a new multi-optima search technique," in *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, pp. 1745–1752, May 2009.

[107] N. Hansen, S. Finck, R. Ros, and A. Auger, "Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions," Research Report RR-6829, 2009.

[108] S. Chen and J. Montgomery, "Selection strategies for initial positions and initial velocities in multi-optima particle swarms," in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, (New York, NY, USA), pp. 53–60, ACM, 2011.

[109] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *Signal Processing, IEEE Transactions on*, vol. 56, pp. 3447–3457, Aug 2008.

[110] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Analytic implementations of the cardinalized probability hypothesis density filter," *Signal Processing, IEEE Transactions on*, vol. 55, pp. 3553–3567, July 2007.

[111] N. T. Pham, W. Huang, and S. H. Ong, "Tracking multiple speakers using CPHD filter," in *Proceedings of the 15th international conference on Multimedia*, MULTIMEDIA '07, (New York, NY, USA), pp. 529–532, ACM, 2007.

[112] N. T. Pham, W. Huang, and S. Ong, "Multiple sensor multiple object tracking with gm-phd filter," in *Information Fusion, 2007 10th International Conference on*, pp. 1–7, July 2007.

[113] M. Beard, B.-T. Vo, B.-N. Vo, and S. Arulampalam, "Gaussian mixture PHD and CPHD filtering with partially uniform target birth," in *Information Fusion (FUSION), 2012 15th International Conference on*, pp. 535–541, July 2012.

[114] C. Cheong Took, S. Douglas, and D. Mandic, "On approximate diagonalization of correlation matrices in widely linear signal processing," *Signal Processing, IEEE Transactions on*, vol. 60, pp. 1469–1473, March 2012.

Appendix A
Publications

A.1 Conference papers

- HEIGHT APPROXIMATION FOR AUDIO SOURCE LOCALISATION AND TRACKING
- VIDEO TRACKING THROUGH OCCLUSIONS BY FAST AUDIO SOURCE LOCALISATION

A.2 Papers to be submitted

- PARTICLE SWARM OPTIMISATION FOR SINGLE SOURCE ACOUSTIC LOCALISATION

HEIGHT APPROXIMATION FOR AUDIO SOURCE LOCALISATION AND TRACKING

Ashley Hughes[⋆] *James R. Hopgood*[⋆] *Neil M. Robertson*[†]

[⋆]School of Engineering, The University of Edinburgh [†]Visionlab, Heriot-Watt University, Edinburgh
Email: {a.hughes, james.hopgood}@ed.ac.uk n.m.robertson@hw.ac.uk

ABSTRACT

The stochastic region contraction (SRC) algorithm has been proposed in the literature as a method for acoustic localisation using a microphone array in a noisy and reverberant environment. This technique makes use of the steered response power (SRP), a costly but robust technique for source localisation, and finds the global maximum vastly more efficiently than using a grid search method. We discuss combining this technique with prior information (e.g. in future work we will use a video tracker) to speed up the algorithm by, in some cases, an order of magnitude by limiting the heights to be searched. This gain is derived from simulations and is achieved whilst at the same time not neglecting large search volumes, continuing to allow a change of audio sources to be detected.

Index Terms— Microphones, Acoustic measurements, Optimization methods, Sampling methods

1. INTRODUCTION

Acoustic source localisation has been studied extensively in the literature [1–3]. Systems make use of an array of microphones to sample audio data and commonly use time difference of arrival (TDOA) techniques to estimate the angle of arrival of a sound wave relative to a pair of microphones. These angles can then be used to triangulate the location of an acoustic source [4]. The steered response power (SRP) is a slower method which also has potential for use in multi-speaker detection. This paper relates to previous work by building on a successful technique which uses the SRP to find an audio source quickly by reducing the number of calculations needed to localise a source. The work presented reduces this number further, making the algorithm useful even in relatively low signal to noise ratio (SNR) environments.

The SRP is a useful measure of the acoustic power originating from a particular location in space within a room. It has been shown to be relatively robust to reverberation. The generalised cross correlation with phase transform (GCC-PHAT) [5] from a set of microphones is used by the SRP algorithm to build up a 3-dimensional (3D) map of this power. Since the volume of a room is very large compared to the spatial resolution generally required by source tracking applications and because of the slow nature of the algorithm, the calculation of the SRP across an entire room is computationally expensive. The output is also a 3D array, which makes it costly, although not intractable, to search through. There are various methods [6–8] for finding global maxima of the array, however SRP based audio localisation also has the potential to locate and track multiple speakers more easily than the traditional maximal generalised cross correlation (GCC) TDOA methods [9].

Existing work reduces the time taken to find a maximum within an area by sampling from the search space randomly and then recursively shrinking the search space using the best subset of the results, a technique called stochastic region contraction (SRC) [7]. Rather than assuming the search volume is the whole room, the SRC algorithm [10] assumes that the height of the search volume is restricted to being one metre high and is also offset from the ground [7]. The contribution of this work is a method to extend the implicit assumptions of head height made when using the SRC by assuming that prior information of the expected head height at some positions is available. For example, this information can be estimated from a camera system using Viola-Jones face detection [11, 12]. The contribution in 3 then interpolates and extrapolates to estimate head height across the 2-dimensional (2D) search area, $\mathbb{A}$, and from that, a sampling distribution over height is formed across the room. This allows the number of functional evaluations (FEs) required to find a maximum to be reduced. Because interpolation is used to estimate head height across an area, people missed in the visual domain due to occlusion are still quickly locatable in the audio domain.

This paper describes the SRP, which is the functional of the SRC algorithm, and then goes on to describe the interpolation and probability density function (PDF) used in the proposed height estimation (HE) SRC algorithm. This algorithm is then tested on a recorded data set which had the room set up, for comparison, to be similar to the conditions described in [7]. The paper also proposes a novel approach to multi-source audio localisation. By sampling across every 2D point

The authors would like to give thanks for Scholarships from the James Clerk Maxwell Foundation and from the Maxwell Advanced Technology Fund at the University of Edinburgh.

within a room at a height drawn from this distribution, a 2D SRP map can be made of the search area at relatively low computational cost. This may prove itself to be useful for algorithms to find multiple maxima, corresponding to multiple audio sources, for robust multi-speaker localisation. By increasing the number of samples at each height and averaging, this tends towards the marginalisation of the SRP over height.

2. STOCHASTIC REGION CONTRACTION

A popular method of audio source tracking is extracting and triangulating TDOA values from the maxima of the GCC-PHAT of signals from pairs of microphones in the frequency domain, given by Equation (1)

$$\hat{R}_{x_m x_n}(\tau) = \int_{-\infty}^{\infty} \frac{\hat{G}_{x_m x_n}(f)}{|\hat{G}_{x_m x_n}(f)|} e^{2\pi f \imath \tau} \, df \qquad (1)$$

which is an inverse Fourier transform where $\hat{G}_{x_m x_n}$ is the product of the signals x_m and x_n in the frequency domain.

The SRP makes use of the GCC-PHAT to build an energy map for each point $(x,\, y,\, z)$ in a search area $\mathbb{A}$ using Equation (2)

$$S(x, y, z) = \sum_{n=1}^{M} \sum_{m=n+1}^{M} \hat{R}_{x_n x_m}[\tau_{n\,m}(x, y, z)] \qquad (2)$$

in a system with M microphones. This is the sum over all pairs $(m,\, n)$ of microphones of the corresponding value of the GCC-PHAT for the TDOA τ. The TDOA is defined by Equation (3)

$$\tau_{n\,m}(\mathbf{p}) = (|\mathbf{m} - \mathbf{p}| - |\mathbf{n} - \mathbf{p}|) \,/\, c \qquad (3)$$

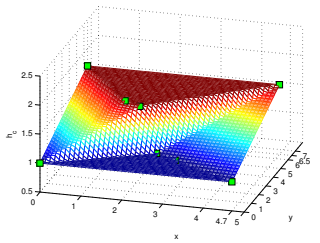
where $\mathbf{p}$ is the vector $(x,\, y,\, z)$ of the point under investigation, c is the speed of sound, and $\mathbf{m}$ and $\mathbf{n}$ are the positions of microphones m and n respectively.

SRC takes samples of the SRP from across the search space and attempts to contract it by using the area given by a set of the highest valued samples [7]. Because these will generally be centred around a peak, caused by a sound source, the search area should quickly shrink. By repeating this, the search space will become an area sufficiently small enough to be considered the point which is the maximum of the SRP function and therefore the source of the sound.

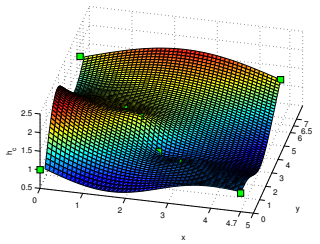
3. INTERPOLATION

To choose head height, existing knowledge of the current positions and heights of people in a room can be used. In an audio-visual (AV) system, this is easy to initialise as video data can be used to make an initial estimation of the heights which should be searched in the audio domain. In addition, existing audio domain search techniques such as the full SRC algorithm can be used to make the first head height estimation. After they have been found initially, the tracked locations of people, both speakers and non-speakers, from both audio and visual sources will allow a good estimate of the height to be used across the room. From a sparse set of people, the head height to be used at every x-y co-ordinate in the SRP map needs to be defined. This means that an assumption about the outer elements of the set and how they relate to the height at the edge of the search area must be made. This work uses the speaker closest to a corner to specify the height at that corner.

When doing interpolation, there is a trade-off between the smoothness of the curve produced and the size of ripples produced. The interpolation should not contain severe ripples as they would lead to large errors in the head height estimation across the room. Ideally, it should be monotonic and one way to achieve this is to use Delaunay triangulation [13] on the set of speakers, which creates a surface which can be evaluated at any 2D point. Figure 1 compares Delaunay triangulation based interpolation to a plate-splines method [14], where the room dimensions are along the x and y axes and the interpolated heights h_c form the set $\mathbf{H}$ across the area of the room. These show that the Delaunay method solves the problem of large ripples, although it leads to a less smooth interpolation. In order to extrapolate correctly, room corners must be pre-allocated nodes. There are several options for choos-



(a) Delaunay triangulation method for estimating head height (h_c) as a function of position $(\mathbf{x}, \mathbf{y})$



(b) Plate-splines method for estimating head height (h_c) as a function of position $(\mathbf{x}, \mathbf{y})$

Fig. 1. Interpolation Method Comparison

ing the height h_{c_j} at each of these j nodes (in a rectangular room, $j = 4$), such as choosing the height to be the same as the height of the nearest speaker, as shown in Equation (4a), where z_i is the height component of r_i, the position of known node i and r_{c_j} is the position of corner j. An alternative is to use Equation (4b), the expected height of a speaker from all known node heights z_i. If it is assumed that there are a limited number of speakers then finding the nearest node to a corner poses no computational problems.

$$h_{c_j} = \arg\min_{z_i}[r_{c_j} - r_i] \tag{4a}$$

$$h_{c_j} = \mathbb{E}\left[z_i\right] \tag{4b}$$

Because the head height, $\mathbf{H}$, is only an estimate, its accuracy varies across the room. To compensate, the head height to be used in the SRC algorithm is drawn from a PDF which ensures that most of the time, samples are taken around head height without being overly restrictive and a small amount of time from less likely areas, so as not to entirely neglect large portions of the search space. The interpolated head height is taken as the mean of a Gaussian distribution whose variance changes depending on its proximity to a known source. This allows the search to concentrate on areas likely to contain people whilst at the same time, not neglecting to check for possible outliers. The height h_{sub} to use at each time step for every 2D point $\mathbf{p_2} = (x_{p_2},\, y_{p_2})$ is then drawn from (5) where $\mathbb{T}$ is the set of known speaker locations.

$$\begin{aligned} \varphi\left(z \mid \mathbf{p_2}\right) &= \alpha_0 \mathcal{N}\left(\mu_h,\, \sigma_h^2\right) + (1-\alpha_0)\,\mathcal{U}\left(0,\, h_r\right) \\ \mu_h &= \mathbf{H}[\mathbf{p_2}] \\ \sigma_h^2 &= \hat{q}(\mathbf{p_2},\, \mathbb{T}) \end{aligned} \tag{5}$$

which mixes the Gaussian with a Uniform distribution across h_r, the entire height of the room.

This can be repeated n times to create an array where $h[n] = h_{sub}$ each time. The resulting SRP value for the point $\mathbf{p_2}$ can either be the maximum value found as in Equation (6a) or the expectation (Equation (6b))

$$SRP_{\mathbf{p_2}} = \max_z[S(x_{p_2}, y_{p_2}, h[n])] \tag{6a}$$

$$SRP_{\mathbf{p_2}} = \mathbb{E}\left[S(x_{p_2}, y_{p_2}, h[n])\right] \tag{6b}$$

of the values, in which case as n increases, $SRP_{\mathbf{p_2}}$ tends towards the marginalisation of the SRP over z, the room height.

Around each person, we can be relatively confident of their height. Further away from them, the decreasing confidence is modelled by increasing the variance of the sampling PDF. The variance at a distance l metres from a speaker is chosen to be modelled by a sigmoid function, q, such as Equation (7a), which is a scaled error function, or Equation (7b).

$$\mathrm{q}(l) = \alpha_1 \operatorname{erf}\left(\alpha_2 l\right) \tag{7a}$$

$$\mathrm{q}(l) = \alpha_1(1 - e^{-l/\alpha_2}) \tag{7b}$$

These are both 0 at the origin and asymptotically approach constants as their arguments tend towards infinity.

These are combined to form a global variance in Equation (8).

$$\begin{aligned} \mathbb{L}_{\mathbf{p},\mathbb{T}} &= \{l : (\exists \mathbf{q} \in \mathbb{T})(l = |\mathbf{p} - \mathbf{q}|)\} \\ \hat{q}(\mathbf{p_2},\, \mathbb{T}) &= \min_{l \in \mathbb{L}_{\mathbf{p},\mathbb{T}}} q(l) \end{aligned} \tag{8}$$

At any point $\mathbf{p}$ in space, the appropriate variance $\hat{q}$ to use will be the sigmoid function q of the minimum of the set of all 2D Euclidian distances $\overline{\mathbf{pq}}$ to known sources, where an element of $\mathbb{T}$ is denoted as $\mathbf{q}$. The minimum is chosen to ensure that the change in variance remains smooth even for overlapping sigmoids from multiple sources.

4. ALGORITHM

The algorithm for finding the global maximum using the estimated head height is given in Algorithm 1, where DT is the Delaunay Triangulation operation.

Initial search for a speech source
while $running$ **do**
  $\hat{\mathbb{T}} = \mathbb{T}$
  for all room corners **do** ▷ Add room corners to $\hat{\mathbb{T}}$
    $\mathbf{n} \leftarrow (x_{\text{corner}},\, y_{\text{corner}},\, z_{\text{nearest member of }\mathbb{T}})$
    $\hat{\mathbb{T}} \leftarrow \hat{\mathbb{T}} \cup \{\mathbf{n}\}$
  end for
  $\hat{\mathbb{H}} \leftarrow \mathrm{DT}(\hat{\mathbb{T}})$ ▷ Delaunay Triangulation of the set
  for all $\mathbf{p_2} = (x_{p_2},\, y_{p_2}) \in \mathbb{A}$ **do** ▷ Whole search area
    $\hat{\mathbb{H}}_0 \leftarrow h_{sub} \sim \varphi\left(z \mid \mathbf{p_2}\right)$ ▷ Choose a height
  end for
  Perform SRC with heights from $\hat{\mathbb{H}}_0$
  $\mathbb{T} = \mathbb{T} \cup \{\text{new speaker positions}\}$
end while

Algorithm 1: HE-SRC Algorithm

5. EXPERIMENTAL RESULTS

The algorithms were run in the environment shown in Figure 2, where the red circles represent each of the 12 microphones (placed along the edges of the room, similar to the panels used in [15]) and the green squares represent the speaker positions. This was a (4.7x6.5)m room, as described in [15] in order to make a direct comparison. A minute of data was recorded for each speaker at 96,000kHz, which gave each around 300 audio windows based on a window size of 160ms. Speakers did not talk at the same time and the two speakers furthest away from the array were at the lower height of 1m, rather than 1.6m, in order to show that this doesn't affect the algorithm. The variant of SRC used was SRC-I, which fixes J_0 - the number of points to be evaluated at the first iteration

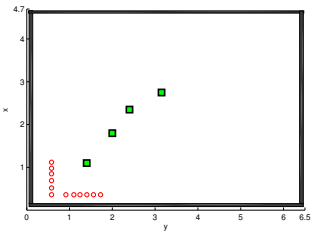


Fig. 2. Room Layout

- to a constant J and then calculates J_i FEs at each iteration of the algorithm, which is decided dynamically [7]. In this variant, a number N of the highest valued samples are used to contract the search region [7]. α_0 was chosen to be 0.95 in order to concentrate the search within head height. Lower values weight the distribution to uniformly draw from across the height of the room, making the search similar to the original SRC algorithm, but with fewer assumptions and therefore slower searches. α_1 was chosen to be 0.5, allowing most of the Gaussian distribution to concentrate on an area 1m tall, similar to the 1m tall Uniform distribution used for height in the original SRC algorithm. Finally, α_2 was generated by choosing the radius l, at which the sigmoid function should be 99% of the way towards α_1, to be 1m, which assumes people have some personal space whilst talking.

Data was evaluated using an Average Location Error (ALE) - the mean of the Euclidian distances of each set of results to their corresponding ground truths. Because the search space was reduced by the height estimation, the number of samples J_i at each stage was lowered to improve overall search times, trading off against accuracy. In the first instance, HE-I, only 350 samples were taken at the first iteration with only the top $N = 30$ used for region contraction. Accuracy decreased as the sound source was further away from the microphone array, implying a lower SNR as in [7], but this may be acceptable in a system whose tracker accounts for noisy state observations and exploiting this may warrant further investigation. For HE-II, J_0 was set to 1000 and N to 60, which brought the accuracy across all sources up whilst keeping the number of FEs low. In HE-III, J_0 was set to 3000 and N to 60, the value as used in [7]. Table 1 shows the results of first (SRC-I) variation of the SRC algorithm from [7] on the data set and compares these configurations with the HE variants. It shows the average number of FEs used within an audio frame and the ALE, where Source 1 is the closest to the microphone array and Source 4 is the furthest.

The results show that with prior information about head height within a room, the SRC can be sped up whilst maintaining accuracy. Because in HE-III the parameters are similar to the SRC-I parameters, the algorithms are expected to perform similarly when there is no known audio source. In this case, the mean of the Gaussian is set to the same offset as that used in the algorithm and the variance is again set to 0.5.

Algorithm	**Source 1**		**Source 2**		**Source 3**		**Source 4**	
	ALE (m)	**# FEs**	**ALE (m)**	**# FEs**	**ALE (m)**	**# FEs**	**ALE (m)**	**# FEs**
SRC-I	0.26	61,1001	0.31	61,1001	0.45	61,001	0.6	61,001
HE-I	0.32	17,156	0.35	21,939	0.44	31,811	0.58	35,053
HE-II	0.12	34,022	0.22	35,136	0.26	41,228	0.5	39,402
HE-III	0.11	40,721	0.15	40,736	0.23	42,900	0.34	44,111

Table 1. Comparison of SRC Methods

Algorithm	**Source 1**		**Source 2**		**Source 3**		**Source 4**	
	ALE (m)	**# FEs**	**ALE (m)**	**# FEs**	**ALE (m)**	**# FEs**	**ALE (m)**	**# FEs**
HE-I	0.45	20,115	0.49	23,849	0.51	36,253	0.6	37,962
HE-II	0.23	35,117	0.24	36,140	0.41	47,281	0.47	48,294
HE-III	0.22	43,783	0.24	43,548	0.32	55,352	0.46	56,667

Table 2. FEs required to find a source with no prior

Table 2 shows the average number of FEs required to find a source using the algorithm without prior information. The results indicate a reduced performance with HE-III, but still within the tractable range of tens of thousands of FEs and close to the performance of SRC-I, as expected. For lower values of J_0 and N, results are improved. In particular, HE-II provides good accuracy and good performance, with or without prior information, so much so that it is suitable as an audio estimator for the initial height information in this situation.

6. CONCLUSIONS

This work contributes a method of speeding up and increasing the accuracy of the SRC algorithm by estimating the height at which to search from prior information, obtainable via standard methods and information from a previous iteration of the algorithm. The key to this technique is to estimate an average head height across an area by interpolating and extrapolating heights of known speakers and forming a probability distribution of head height using this data. This allows a single audio source to be localised quickly whilst still searching across the room to find new source, for example when there is a speaker change. Further work will investigate using the height estimated SRP to locate multiple maxima simultaneously.

7. REFERENCES

[1] A. Badali, J.-M. Valin, F. Michaud, and P. Aarabi, "Evaluating real-time audio localization algorithms for artificial audition in robotics," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, oct. 2009, pp. 2033 –2038.

[2] Weiping Cai, Shikui Wang, and Zhenyang Wu, "Accelerated steered response power method for sound source

localization using orthogonal linear array," *Applied Acoustics*, vol. 71, no. 2, pp. 134 – 139, 2010.

[3] M. F. Fallon and S. J. Godsill, "Acoustic source localization and tracking of a time-varying number of speakers," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 4, pp. 1409 –1415, may 2012.

[4] Matthias Wölfel and John. McDonough, *Distant Speech Recognition*, Wiley, 2009.

[5] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 24, no. 4, pp. 320 – 327, Aug. 1976.

[6] J.P. Dmochowski, J. Benesty, and S. Affes, "A generalized steered response power method for computationally viable source localization," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 8, pp. 2510 –2526, nov. 2007.

[7] Hoang Do, H.F. Silverman, and Ying Yu, "A real-time SRP-PHAT source location implementation using Stochastic Region Contraction (SRC) on a large-aperture microphone array," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, april 2007, vol. 1, pp. I–121 –I–124.

[8] Hoang Do and H.F. Silverman, "Stochastic particle filtering: A fast srp-phat single source localization algorithm," in *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA '09. IEEE Workshop on*, oct. 2009, pp. 213 –216.

[9] Fotios Talantzis, Aristodemos Pnevmatikakis, and Anthony G Constantinides, *Audio-Visual Person Tracking: A Practical Approach*, Imperial College Press, 2012.

[10] M.F. Berger and H.F. Silverman, "Microphone array optimization by stochastic region contraction," *Signal Processing, IEEE Transactions on*, vol. 39, no. 11, pp. 2377–2386, 1991.

[11] Paul Viola and Michael Jones, "Robust real-time object detection," in *International Journal of Computer Vision*, 2001.

[12] T. Gehrig, K. Nickel, H.K. Ekenel, U. Klee, and J. McDonough, "Kalman filters for audio-video source localization," in *Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on*, oct. 2005, pp. 118 – 121.

[13] D. T. Lee and B. J. Schachter, "Two algorithms for constructing a Delaunay triangulation," *International Journal of Parallel Programming*, vol. 9, pp. 219–242, 1980, 10.1007/BF00977785.

[14] John D'Errico, "inpaint_nans," MATLAB Central File Exchange, February 2012.

[15] H.F. Silverman, Ying Yu, J.M. Sachar, and II Patterson, W.R., "Performance of real-time source-location estimators for a large-aperture microphone array," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 593 – 606, july 2005.

VIDEO TRACKING THROUGH OCCLUSIONS BY FAST AUDIO SOURCE LOCALISATION

Eleonora D'Arca, Ashley Hughes, Neil M. Robertson and James Hopgood

Joint Research Institute for Signal and Image Processing,
Heriot-Watt University & University of Edinburgh, UK
visionlab.eps.hw.ac.uk

ABSTRACT

In this paper we present a novel audio-visual speaker detection and localisation algorithm. Audio source position estimates are computed by a novel stochastic region contraction (SRC) audio search algorithm for accurate speaker localisation. This audio search algorithm is aided by available video information (stochastic region contraction with height estimation (SRC-HE)) which estimates head heights over the whole scene and gives a speed improvement of 56% over SRC. We finally combine audio and video data in a Kalman filter (KF) which fuses person-position likelihoods and tracks the speaker. Our system is composed of a single video camera and 16 microphones. We validate the approach on the problem of video occlusion i.e. two people having a conversation have to be detected and localised at a distance (as in surveillance scenarios vs. enclosed meeting rooms). We show video occlusion can be resolved and speakers can be correctly detected/localised in real data. Moreover, SRC-HE based joint audio-video (AV) speaker tracking outperforms the one based on the original SRC by 16% and 4% in terms of multi object tracking precision (MOTP) and multi object tracking accuracy (MOTA). Speaker change detection improves by 11% over SRC.

Index Terms— Video Tracking, Speaker Tracking, Multimodal tracking, Optimization methods, Sampling Methods

1. INTRODUCTION

Solving visual tracking occlusion is inherently challenging when only video information is available. Many existing papers solve the problem by using sophisticated multi-camera 3-dimensional (3D) systems [1] which are still prone to occlusions when the camera fields-of-view do not overlap. Moreover, they are computationally expensive, often requiring GPU/FPGA implementations to function at frame-rate. Thus, supporting tracking with non-visual information, i.e. audio, may compensate for noisy, missing and erroneous video data via speaker detection info, reducing the number of cameras and the computational resources required at the expense of a few microphones. Video and audio "fusion" (or combination) can be achieved in several ways mostly using variations of sampling techniques [2–4]. Existing system architectures work well in very sanitised scenarios e.g. meeting analysis and diarisation [5–9]. They use large sensor networks composed at least of 4 cameras and 16 microphones [3, 4, 6, 8]. Little attention has been focussed on uncontrolled (and larger) areas of interest using smaller and less "invasive" sensor networks. Attention in the literature is principally focussed on general event detection [10–12], rather than on people interactions and behaviour analysis [13, 14]. The novel system we present can localise and recognise a speaker among two people in an ample, reverberant and noisy environment when large video occlusion occur using a small sensor network. To the best of our knowledge this work is similar to the ones from [4, 15]. In contrast, we improve on the state-of-the-art via: *a*) new, high accuracy, fast audio localisation algorithm; *b*) real-time video localisation and tracking using particle filter (PF) [1]; *c*) improved precision and accuracy metrics for multi object tracking (2006 and 2007 CLEAR dataset [16]).

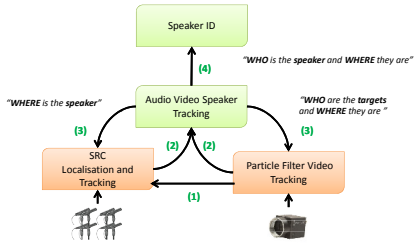


Fig. 1: A schematic of the system presented in this paper. Constituent parts of this diagram are referred to explicitly in the text (e.g. "arrow 1").

2. THEORY

A schematic diagram of our system is shown in Figure 1. In the following sections we describe it in detail.

NR and JH are supported by EC FP7 LOCOBOT (Grant EC/260101). AH is supported by scholarships from the James Clerk Maxwell Foundation and the Maxwell Advanced Technology Fund.

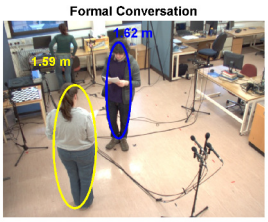


Fig. 2: Video detected height data are novelly used to reduce the search of space for audio source localisation SRC.

2.1. Height detection and video tracking

Full details of the video tracker based on a GPU-accelerated particle filter with ellipsoid models for people can be found in [1]. It is worth noting that we hereby use the video data coming from only 1 camera view. Height measurement is also extracted (Figure 2) to cue the audio localisation algorithm, since it directly corresponds to a good estimate of the speaker's head position.

2.2. Audio source localisation

A popular method of audio source tracking is extracting maximal time difference of arrival (TDOA) values from the generalised cross correlation with phase transform (GCC-PHAT) [17] of signals from a pair of microphones in the frequency domain, given by Equation (1), which is an inverse Fourier transform where $\hat{G}_{x_m x_n}$ is the product of the signals x_m and x_n in the frequency domain.

$$\hat{R}_{x_m x_n}(\tau) = \int_{-\infty}^{\infty} \frac{\hat{G}_{x_m x_n}(f)}{|G_{x_m x_n}(f)|} e^{2\pi f \iota \tau} \, df \tag{1}$$

A method more robust to reverberation, the steered response power (SRP), makes use of the GCC-PHAT to build an energy map using Equation (2) in a system with M microphones. This is the sum over all pairs $(m,\, n)$ of microphones of the corresponding value of the GCC-PHAT for the TDOA τ.

$$S(x, y, z) = \sum_{n=1}^{M} \sum_{m=n+1}^{M} \hat{R}_{x_n \, x_m}[\tau_{n\,m}(x, y, z)] \tag{2}$$

The TDOA is defined by Equation (3), where $\mathbf{p}$ is the vector $x,\, y,\, z$ of the point under investigation, c is the speed of sound, and $\mathbf{m}$ and $\mathbf{n}$ are the positions of microphones m and n respectively.

$$\tau_{n\,m}(\mathbf{p}) = (|\mathbf{m} - \mathbf{p}| - |\mathbf{n} - \mathbf{p}|) \,/\, c \tag{3}$$

Evaluating the SRP across an entire room is computationally costly. In this work we use an enhanced version of the SRC [18] algorithm to localise quicker and better an audio source. This works by sampling the SRP randomly and choosing a subset of the largest samples to form a new region to sample within. This is repeated until the process has discovered a maximum. In order to further improve upon the SRC, instead of sampling uniformly over height, a different sampling distribution is used, centred around a head height. To choose head height, existing knowledge of the current positions and heights of people in a room which is obtained from the camera (Figure 2), is novelly used (SRC-HE). In particular, the height data is updated on each iteration to the height of the last SRP peak found. This reduction of the search space decreases its effective dimensionality, thereby decreasing the computational complexity of SRC.

From a sparse set of people, the head height at every x-y co-ordinate in the SRP map needs to be defined. This is achieved using interpolation and extrapolation. When doing the interpolation, there is a trade-off between the smoothness of the curve produced and the size of ripples produced. The interpolation should not contain severe ripples as they would lead to large errors in the head height estimation across the room. Ideally, it should be monotonic and one way to achieve this is to use Delaunay triangulation [19] on the set of speakers, which creates a surface which can be evaluated at any 2-dimensional (2D) point.

The height h_{sub} to use at each time step for every point $\mathbf{p} = (x,\, y)$ is then drawn from 4, which mixes a Gaussian with a Uniform distribution across h_r, the entire height of the room.

$$\begin{aligned} p\,(z \mid \mathbf{p}) &= \alpha_0 \mathcal{N}\left(\mu_h,\, \sigma_h^2\right) + (1 - \alpha_0)\, \mathcal{U}\,(0,\, h_r) \\ \mu_h &= \mathbf{H}[\mathbf{p}] \\ \sigma_h^2 &= \hat{q}(\mathbf{p},\, \mathbb{T}) \end{aligned} \tag{4}$$

Around each person, we can be relatively confident of their height. Further away from them, the decreasing confidence is modelled by increasing the variance of the sampling probability density function (PDF). The variance at a distance l metres from a speaker is chosen to be modelled by a sigmoid function, q, such as Equation (5), which is a scaled error function. This is 0 at the origin and asymptotically approaches a constant as its argument tends towards infinity.

$$\mathrm{q}(l) = \alpha_1 \operatorname{erf}(\alpha_2 l) \tag{5}$$

These need to be combined to form a global variance. At any point $\mathbf{p}$ in space, the appropriate variance $\hat{q}$ to use will be the sigmoid function q of the minimum of the set of all 2D Euclidian distances $\overline{\mathbf{pq}}$ to known sources, where the set of known source locations is denoted as $\mathbb{T}$ and an element from the set of know sources is denoted as $\mathbf{q}$. This is expressed in Equation (6). The minimum is chosen to ensure that the change in variance remains smooth even for overlapping sigmoids from multiple sources.

$$\mathbb{L}_{\mathbf{p},\mathbb{T}} = \{l : (\exists \mathbf{q} \in \mathbb{T})(l = \overline{\mathbf{pq}})\}$$
$$\hat{q}_{\mathbf{p},\mathbb{T}} = \min_{l \in \mathbb{L}_{\mathbf{p},\mathbb{T}}} q(l) \quad (6)$$

2.3. Joint Audio-Video Speaker Tracking

SRC-HE algorithm allows for direct speaker position calculation, $\mathbf{x}$. Nevertheless, speaker position estimations are characterised by missing and false detections. This is mostly due to speech pauses and room reverberation respectively. Thus, we filter SRC estimated positions $\mathbf{x_a}$ by a KF. We said already that, to speed up SRC searching time, speaker's height computed by the video PF, is input into the audio unit to drive height sampling (arrow 1, Figure 1). Then, after the audio and video data have been aligned, the posteriors of the KF audio tracker and of the PF $\mathbf{x}_a$ and $\mathbf{x}_v$ are fused in a common KF node (arrow 2, Figure 1). As data are gathered simultaneously and used all at once in a centralised fashion, we assume the audio and video *pdfs* to be independent of one another thus, on the basis of the *a priori* local estimates for the state $\mathbf{x}_a(t|t-1)$ and $\mathbf{x}_v(t|t-1)$ predicted by the single-modality trackers at each time step t, we evaluate the joint state estimate $\mathbf{x}_{av}$ as follows (where time dependency has been omitted for clarity):

$$\mathbf{p}(\mathbf{z}_{av} \mid \mathbf{x}) = \mathbf{p}(\mathbf{z}_a \mid \mathbf{x})\mathbf{p}(\mathbf{z}_v \mid \mathbf{x}); \quad (7)$$

this means the joint likelihood is still a Gaussian probability, although no longer normalised, and the *a posteriori* state estimate is given by:

$$\mathbf{x}_{av} = \mathbf{P}_{av}\{\mathbf{P}_a^{-1}\mathbf{x}_a + \mathbf{P}_v^{-1}\mathbf{x}_v\}, \quad (8)$$

where

$$\mathbf{P}_{av} = (\mathbf{P}_a^{-1} + \mathbf{P}_v^{-1})^{-1}. \quad (9)$$

$\mathbf{P}_a^{-1}$ and $\mathbf{P}_v^{-1}$ are the inverse of the audio and video *a posteriori* covariance estimation matrices. $\mathbf{P}_{av}$ is the joint *a posteriori* covariance estimation matrix. Finally, the last joint AV output $\mathbf{x}_{av} = \mathbf{P}_a\mathbf{x}_a + \mathbf{P}_v\mathbf{x}_v$ is fed back into the individual audio and video trackers as the best estimate of the previous time step to improve the single modality estimation (arrow 3, Figure 1). It is important to notice that, as we make the assumption that people speak alternatively, like in a normal conversational mode, to a single audio signal $\mathbf{z}_a$, correspond several video measurements $\mathbf{z}_{v_i}$ at a time, one for each of the N detected targets. By basing the audio-to-video data association step on spatial proximity, i.e. nearest neighbour (NN), speaker segmentation and recognition can also be obtained as long as people are resolved by the AV tracker and its measurements can be considered robust with respect to the speaker motion model. In particular, the speaker identity inferred by the joint tracker is equal to the one of the i-th target if $\mathbf{S}_{av} = \arg\max_i \left\{\mathbf{p}(\mathbf{z}_a, \mathbf{z}_{v_i} \mid \mathbf{x})\right\}, i = 1, ..., N$ (arrow 4, Figure 1). Saying that, once an identity i has been assigned to every target in an image frame, the speaker change detection output by the audio unit is used in order to recover identity (ID) tracking when occlusions occur. In particular, in case audio and video inference about the detected number of targets in the scene is conflicting, or when audio and video data do not both fall within a certain region ($\|\mathbf{x}_a - \mathbf{x}_v\| \leq A$), audio source position is considered to be correct and it is also sent back to the video tracking unit to indirectly re-assign the correct appearance models to the targets, successfully resolving occlusions (arrow 3, Figure 1).

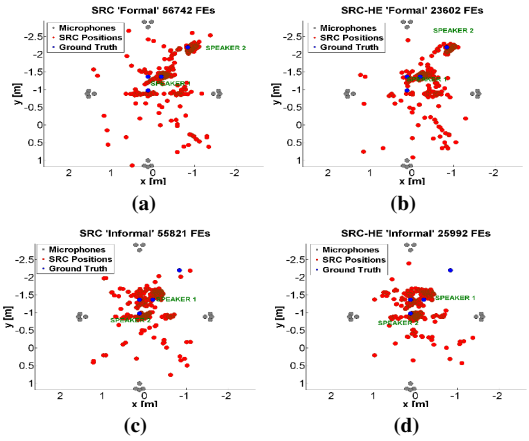


Fig. 3: SRC and SRC-HE raw speaker position detections. Interesting is the number of FEs which on average is reduced by 56% (FEs $56,281$ vs $24,797$) for the SRC-HE implementation. **(a)** and **(b)** show respectively audio source SRC and SRC-HE detections for the *'Formal'* experiment. While **(c)** and **(d)** show them for the *'Informal'* one.

Experiment	System	SSL Accuracy (%)	FEs
'Formal'	**SRC**	62.50	56742
	SRC-HE	69.07	23601
'Informal'	**SRC**	47.30	55821
	SRC-HE	51.22	25992

Table 1: SRC vs SRC-HE performance comparison for the two set of data (*'Formal'* and *'Informal'*). Results are shown for 2 off-line runnings of the two algorithms. SSL accuracy changes by 4% when adding up extracted video height info. More interesting is the 56% change in the number of FEs which has to be calculated, meaning that narrowing down the space of search effectively results in speeding up the localisation task.

3. EXPERIMENTATION AND RESULTS

In this section we show that SRC-HE outperforms original SRC using video data and that our global AV system can maintain and recover speaker ID. We used 1 camera and $4-by-4$ T-shape microphone arrays to record AV data in a typical open office room, whose size is $111.44\ m^2$, where the area considered of interest is $12\ m^2$ (as seen in Figure 4**(a)**). Ground-truth data was hand labelled to $5\ cm$ of accuracy, on a ground plane common to camera and microphones. Audio

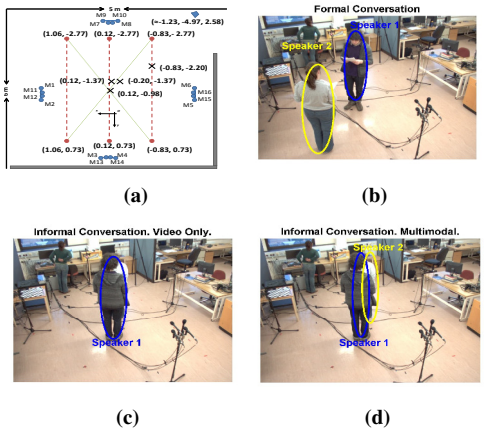


Fig. 4: Real experiments layout **(a)** and *'Formal'* and *'Informal'* visual results. In **(b)** a formal conversation between two people is shown. Video tracker, as well as multimodal tracker, can detect and recognise there are two targets speaking alternatively and their output is the same. **(c)** shows an informal conversation between two people. They are so close the video tracker on its own cannot detect there are two different targets. In **(d)** instead, the AV multimodal tracker is shown to detect the two speakers and successfully recognise their identity.

Experiment	System	MOTP (m)	MOTA (%)	DER (%)
'Formal'	**SRC**	0.35	85	21
	SRC-HE	0.34	90	7
'Informal'	**SRC**	0.20	97	20
	SRC-HE	0.12	100	11.80

Table 2: Experiment results. SRC AV tracker does not incorporate prior video height information while SRC-HE does.

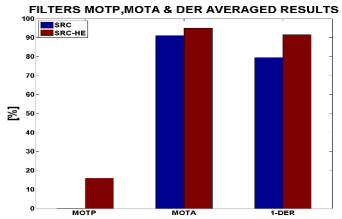


Fig. 5: SRC vs SRC-HE AV tracking averaged over both the experiments and 100 montecarlo runs performance comparison. SRC-HE detection accuracy improvement results in an AV tracker which outperforms SRC based AV tracker precision (MOTP) by 16% and accuracy by 4%. Of interest here, is that DER is also improved by 11%, which make this solution 11% better than SRC in handling large video occlusions. Note that the video tracker on its own instead can not resolve occlusion at all.

signals were sampled by the audio interface with a 24-bit precision resolution at 44.1 kHz, whereas the camera recorded the 640×480 RGB video frames at a 7.5 Hz rate. Moreover, each audio signal was filtered using a $\approx$ 20 ms long Gaussian window to ensure signal stationarity [20]. We made no attempt to reduce normal background noise (desk fans, footsteps, talking etc.) and a large reverberation time ($T_{60} \approx 0.5$ s) was measured. Synchrony of data was insured by processing audio and video streams accordingly to the camera frame rate. Filters were initialised using the video detected position of their correspondent targets and static matrices Q and R [21], whose values were chosen on the basis of an optimisation step. We describe the results in terms of MOTP and MOTA [16]. We also calculate the diarisation error rate (DER), which measures the ability of detecting a change in speaker ID, expressing the speaker error only [22].

Experiments meant to simulate a personal (formal) and intimate (informal) conversation between two people, resulting in an occlusion in the case of informal conversation. Specifically:

'Formal Conversation', considers two people having a 60 s conversation. Throughout all the experiment they are separated by a distance of approximately 104 cm. Results as presented in Figure 4 **(b)**.

'Informal Conversation', considers two people having a 56 s conversation. Throughout all the experiment they are separated by a distance of approximately 40 cm. Results are shown in Figure 4 **(c)** and **(d)** .

Figure 3 demonstrates SRC vs SRC-HE raw speaker position detections for the two set of data (*'Formal'* and *'Informal'*). In Table 1 we enumerate their performance comparison. Results are shown in terms of SSL accuracy and number of FE calculations. In both cases, the results show a significant decrease in the number of FEs as well as an improvement in accuracy. Moreover, video only and SRC-HE based AV tracker outputs are shown in Figures 4 **(c)** and **(d)** for a comparison. Furthermore, in Table 2 we present MOTP, MOTA and detection error rate (DER) of the joint AV trackers based on SRC only and on SRC-HE. At last, their performance comparison is shown in Figure 5. Please note that, when we talk about SRC results we refer to an AV system as in Figure 1 where arrow 1 does not exist (no video cueing).

4. CONCLUSION AND FUTURE WORK

In this paper integrating height information coming from a video PF with a SRC SSL algorithm (SRC-HE), has been proved to speed up by 56% speaker detection based on the original SRC algorithm. Moreover, it has been shown that augmenting video tracking with audio data does solve large occlusion which otherwise would not be solved by the video tracker only. Furthermore, using audio data detected with SRC-HE improves by 16% and 4% AV speaker MOTP and MOTA tracking and by 11% AV speaker change detection, if compared to an AV tracker which uses the original SRC implementation. In future, we would like to carry out a tighter integration between audio and video using updated height information from every frame to investigate further improvements on SRC-HE. Furthermore, we would like to record datasets similar to other existing works to carry out a thorough comparison against state-of-the art joint AV systems in non-meeting rooms.

5. REFERENCES

[1] Wasit Limprasert, Andrew M. Wallace, and Greg Michaelson, "Accelerated people tracking using texture in a camera network," in *VISAPP (2)'12*, 2012, pp. 225–234.

[2] N. Checka, K.W. Wilson, M.R. Siracusa, and T. Darrell, "Multiple person and speaker activity tracking with a particle filter," vol. 5, pp. V–881–4 vol.5, May 2004.

[3] Yeongseon Lee and R. Mersereau, "Data association for people tracking using multiple cameras," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 312008-april4 2008, pp. 2585 –2588.

[4] Huiyu Zhou, M. Taj, and A. Cavallaro, "Target detection and tracking with heterogeneous sensors," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 2, no. 4, pp. 503–513, Aug. 2008.

[5] Kai Nickel, Tobias Gehrig, Hazim Kemal Ekenel, John W. McDonough, and Rainer Stiefelhagen, "An audio-visual particle filter for speaker tracking on the clear'06 evaluation dataset," in *CLEAR*, 2006, pp. 69–80.

[6] D. Gatica-Perez, G. Lathoud, J.-M. Odobez, and I. McCowan, "Audiovisual probabilistic tracking of multiple speakers in meetings," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 2, pp. 601–616, Feb. 2007.

[7] Gerald Friedland, Chuohao Yeo, and Hayley Hung, "Visual speaker localization aided by acoustic models," in *Proceedings of the 17th ACM international conference on Multimedia*, New York, NY, USA, 2009, MM '09, pp. 195–202, ACM.

[8] Shankar T. Shivappa, Bhaskar D. Rao, and Mohan M. Trivedi, "Audio-visual fusion and tracking with multilevel iterative decoding: Framework and experimental evaluation," *J. Sel. Topics Signal Processing*, vol. 4, no. 5, pp. 882–894, 2010.

[9] Xavier Alameda-Pineda, Vasil Khalidov, Radu Horaud, and Florence Forbes, "Finding audio-visual events in informal social gatherings," in *Proceedings of the 13th international conference on multimodal interfaces*, New York, NY, USA, 2011, ICMI '11, pp. 247–254, ACM.

[10] E. Kidron, Y.Y. Schechner, and M. Elad, "Pixels that sound," vol. 1, pp. 88–95 vol. 1, June 2005.

[11] Marco Cristani, Manuele Bicego, and Vittorio Murino, "Audio-visual event recognition in surveillance video sequences," *Multimedia, IEEE Transactions on*, vol. 9, no. 2, pp. 257–267, Feb. 2007.

[12] H. Izadinia, I. Saleemi, and M. Shah, "Multimodal analysis for identification and segmentation of moving-sounding objects," *Multimedia, IEEE Transactions on*, vol. 15, no. 2, pp. 378 –390, feb. 2013.

[13] Maria Andersson, Stavros Ntalampiras, Todor Ganchev, Jrgen Ahlberg Rydell, and Nikos Fakotakis, "Fusion of acoustic and optical sensor data for automatic fight detection in urban environments," in *FUSION 2010*, 2010.

[14] M. Andersson and R. Johansson, "Multiple sensor fusion for effective abnormal behaviour detection in counter-piracy operations," in *Waterside Security Conference (WSS), 2010 International*, nov. 2010, pp. 1 –7.

[15] M. Bregonzio, M. Taj, and A. Cavallaro, "Multi-modal particle filtering tracking using appearance, motion and audio likelihoods," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 16 2007-oct. 19 2007, vol. 5, pp. V –33 –V –36.

[16] Keni Bernardin and Rainer Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *J. Image Video Process.*, vol. 2008, pp. 1:1–1:10, January 2008.

[17] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 24, no. 4, pp. 320–327, January 2003.

[18] Hoang Do, H.F. Silverman, and Ying Yu, "A real-time srp-phat source location implementation using stochastic region contraction(src) on a large-aperture microphone array," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, april 2007, vol. 1, pp. I–121 –I–124.

[19] D. T. Lee and B. J. Schachter, "Two algorithms for constructing a delaunay triangulation," *International Journal of Parallel Programming*, vol. 9, no. 3, pp. 219–242, June 1980.

[20] Maurice Fallon, *Acoustic Source Tracking Using Sequential Monte Carlo*, Ph.D. thesis, Darwin College, University of Cambridge, September 2008.

[21] T. Gehrig, K. Nickel, H.K. Ekenel, U. Klee, and J. McDonough, "Kalman filters for audio-video source localization," pp. 118–121, Oct. 2005.

[22] Jonathan G. Fiscus, Jerome Ajot, Martial Michel, and John S. Garofolo, *The Rich Transcription 2006 Spring Meeting Recognition Evaluation*, NIST, 2006.

PARTICLE SWARM OPTIMISATION FOR SINGLE SOURCE ACOUSTIC LOCALISATION

Ashley Hughes[⋆] *Neil M. Robertson*[†] *James R. Hopgood*[⋆]

[⋆]School of Engineering, The University of Edinburgh [†]Visionlab, Heriot-Watt University, Edinburgh
Email: {a.hughes, james.hopgood}@ed.ac.uk n.m.robertson@hw.ac.uk

ABSTRACT

In this paper, the particle swarm optimisation (PSO) algorithm is evaluated as a method for acoustic source localisation, using the steered response power (SRP) as the objective function. We evaluate both the accuracy and the computational cost of using several variants of the method on both simulated and recorded data. We also introduce our own modifications of the algorithm, which take into account the nature of the problem, namely that between audio frames, a speaker is not likely to change height significantly. We also impose different constraints on the height axis than those imposed on the ground-plane axes. This forces the search to concentrate on a relatively narrow range of heights, as opposed to the initial broad search of the ground-plane axes of the room. These modifications of the search algorithm are then used to demonstrate an improvement in the localisation speed, which is an order of magnitude smaller than previously studied stochastic region contraction (SRC) methods, without sacrificing localisation accuracy.

Index Terms— Microphones, Acoustic measurements, Optimization methods, Sampling methods

1. INTRODUCTION

The steered response power (SRP) has been shown to be a useful measure of the acoustic energy within a room, which can be used for the localisation of acoustic sources within an area suitably equipped with a microphone array. Particle swarm optimisation (PSO) has been used in conjunction with a particle swarm tracker [1, 2] to improve the performance of the tracker in localising audio sources based on the position-dependant audio power. This work further explores the use of PSO to optimise the SRP, evaluating the effectiveness of different swarm control techniques, and introducing new strategies which take the nature of the problem into account.

Whilst SRP based methods of source localisation are relatively robust to reverberation, evaluating the SRP at a point in space is computationally costly, and evaluating every possible region of space up to the available resolution of a system in order to find the maximum SRP value is generally infeasible. Search techniques such as stochastic region contraction (SRC) [3] and its extensions [4] are used to attempt to reduce the number of SRP functional evaluations (FEs), whilst minimising the localisation error.

In this work, PSO is used as an optimisation technique over more general mathematical optimisers, as the SRP surface is typically noisy, which implies many local optima, which must be ignored in the search for the global best value. Whilst swarm dynamics have been used before in the context of a particle filter, this paper uses the technique to provide a single direct estimate of the source position within an audio frame. The localisation results from consecutive frames can then be filtered, for example, using a Kalman filter [5]. This highlights another advantage of SRP based localisation methods over many traditional generalised cross correlation (GCC) methods, which provide a set of arrival angles [6]. These can then can be used to triangulate a source, for example, using linear intersection [7] of the lines which can be drawn between microphone pairs at the angle of arrival determined between those microphones. These angles of arrival vary non-linearly with the source position, and, as such, any filtering used to estimate the position from the set of localisation results must take this into account. This is the approach taken by Klee et al., who make use of an extended Kalman filter (EKF) to track an acoustic source [8]. More complex filtering is possible, such as an combined EKF and particle filter approach [9] which can reduce localisation error in highly reverberant environments. With the direct measurements of source position given by SRP based methods, the linear Kalman filter can be used instead of the EKF. Kalman filters are used in this work due to the direct relationship of the observations to the source position, although this means that observation covariance must be measured *a priori* and the model used must assume a small process noise. Because we seek to evaluate the performance of a source localisation algorithm, rather than a tracker, we make use of audio sources which only move under process noise, allowing us to make use of the relatively simple Kalman filter to show how an improvement in localisation error can be achieved.

The authors would like to give thanks for Scholarships from the James Clerk Maxwell Foundation and from the Maxwell Advanced Technology Fund at the University of Edinburgh.

The remainder of this paper first presents the experimental environments used to evaluate the source localisation methods studied. The SRP is then defined, followed by a description of the PSO algorithm and some of its variants. New variants which are particular to the properties of the acoustic source localisation problem are then introduced, and the framework under which they are evaluated is discussed. Finally, results will be presented which demonstrate the effectiveness and suitability of PSO to the problem.

2. EXPERIMENTAL ENVIRONMENT

The experiments undertaken made use of both a simulated environment and recorded data. In line with previous work [4], the recorded data was made in an audio lab with a microphone setup in the corner, with room dimensions (8.1 x 5.3 x 3)m. The microphones were spaced evenly along the walls, with a minimum separation of 30cm. This layout allows direct comparison with our own previous work, as well as with others' work using SRC [3]. The audio lab used was a highly reverberant environment with concrete walls and ceilings, but with a carpeted floor windows covered by hard cardboard. The room has a measured T_{60} of 0.836s [10].

Furthermore, this environment is replicated using simulated data, which allows experimentation using, for example, different source positions and room impulse response (RIR) parameters. The audio data used for simulation, a set of recordings of speech taken from a microphone close to the speaker, was subjected to simulated reverberation using the image-source model (ISM) [11], where the T_{60} used was set to 0.2, in order to simulate a less harshly reverberant environment than the audio lab. The simulated room dimensions are shown in Figure 1, where the room height was 3m. These simulated audio signals were also combined with a low level of white Gaussian noise after the reverberation simulation had been applied to the source signals. This noise was applied to the signal for every microphone used.

The microphones were sampled at 96kHz and processed in audio frames of 160ms, leading to frame sizes of 15360 samples. In both simulation and the recorded environment, there were four speakers taking turns to speak, such that there was one known speaker position per audio frame, and the objective was to perform speaker localisation for a single source. The sources were spaced such that they gradually moved further away from the microphone array. This meant that for a constant level of speaker volume, the signal to noise ratio (SNR) of the sources would decrease the further away from the microphones, according to the inverse square law for acoustic intensity.

3. AUXILIARY METHODS

The SRP is defined in Equation (1) as the function of a position in space, $(x,\ y,\ z)$ given a set on M microphones, each

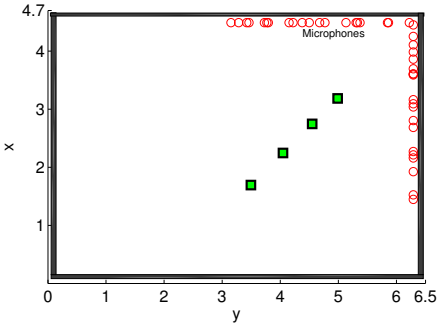


Fig. 1. Room Layout

with a known position. This function uses the generalised cross correlation with phase transform (GCC-PHAT) of signals x_n and x_m, denoted by $\hat{R}_{x_n\,x_m}$, indexed at a target time difference of arrival (TDOA), denoted by $\tau_{n\,m}$.

$$S(x,y,z) = \sum_{n=1}^{M} \sum_{m=n+1}^{M} \hat{R}_{x_n\,x_m}[\tau_{n\,m}(x,y,z)] \qquad (1)$$

For each microphone pair used, the appropriate value of $\tau_{n\,m}$ to use is given by Equation (2), where c is the speed of sound; $\mathbf{m}$ and $\mathbf{n}$ are vectors representing the known positions of microphones m and n respectively, and $\mathbf{p}$ is vector notation for the target location, $(x,\ y,\ z)$. By evaluating the SRP at different positions, a map can be built up of the acoustic power originating from different areas of the room under consideration.

$$\tau_{n\,m}(\mathbf{p}) = (|\mathbf{m}-\mathbf{p}| - |\mathbf{n}-\mathbf{p}|)\,/\,c \qquad (2)$$

The Kalman filter [12] is a minimum mean-square error (MMSE) method used for a tracking the source location given a series of noisy observations of that location. It is important to note that the noise is assumed to be Gaussian and that the state observation and state update functions are assumed to be linear. In this work, we only consider acoustic sources which move under process noise, as in [8], which ensures that the state update function is linear. However, it is possible to change the state update function to take account of moving speakers by changing this assumption, and several methods are explored in [13]. By extracting the coordinates in space which correspond to the maximal SRP value, we also ensure that the observation function is linear, and this allows us to apply the linear Kalman filter.

$$\hat{\mathbf{x}}'_t = \mathbf{A}\hat{\mathbf{x}}_{t-1} \qquad (3a)$$

$$\mathbf{P}'_t = \mathbf{A}\mathbf{P}_{t-1}\mathbf{A}^\intercal + \mathbf{Q} \tag{3b}$$

$$\mathbf{K}_t = \mathbf{P}'_t\mathbf{H}^\intercal\left(\mathbf{H}\mathbf{P}'_t\mathbf{H}^\intercal + \mathbf{R}\right)^{-1} \tag{3c}$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}'_t + \mathbf{K}_t\left(\mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}'_t\right) \tag{3d}$$

$$\mathbf{P}_t = \left(\mathbf{I} - \mathbf{K}_t\mathbf{H}\right)\mathbf{P}'_t \tag{3e}$$

The Kalman filter update steps are given in Equations (3a) to (3e). The new state estimate $\hat{\mathbf{x}}_t$ is derived in two steps, starting with a state estimation expressed in Equations (3a) and (3b), where the state is updated via the state transition matrix $\mathbf{A}$ and the previous state $\hat{\mathbf{x}}_{t-1}$. The state covariance $\mathbf{P}'_t$ is predicted. Then the state estimate is altered based on a noisy observation of the state in the remaining three equations. This makes use of the observation at time t, $\mathbf{z}_t$, which is transformed to the state estimate format by the observation matrix $\mathbf{H}$. The final step is to update the state covariance matrix, $\mathbf{P}_t$.

4. PARTICLE SWARM OPTIMISATION

PSO makes use of a set of particles at positions within the search space which evaluate an objective function and use the results to move the particle locations [14]. At each iteration of the algorithm, both the particles' positions and their velocities are updated based on the global best (in our case, maximal) observed value of the objective function and each particles' own best observed position. Particles move towards these positions, and the algorithm tends to, but is not guaranteed to, converge on the global optimum value of the objective function. The PSO algorithm is detailed in Algorithm 1. The algorithm stops either when a specified maximum number of iterations have been evaluated, or when the global best value changes by less than a specified threshold value g_{thresh} for a specified number of continuous iterations g_{epochs}.

The particle speed and position update rules for the inertial PSO [15] are given in Equations (4a) and (4b). At each time step z, the velocity of the i$^{\text{th}}$ particle, given by V_i, is updated using that particle's historical position $\mathbf{p}_i$ and the global best position, $\mathbf{p}_g$. That particle's position, X_i, is then updated using its new velocity.

$$\begin{aligned} V_i\left(z+1\right) = \omega_p V_i\left(z\right) + c_1 r_1\left[\mathbf{p}_i\left(z\right) - X_i\left(z\right)\right] \\ + c_2 r_2\left[\mathbf{p}_g\left(z\right) - X_i\left(z\right)\right] \end{aligned} \tag{4a}$$

$$X_i\left(z+1\right) = \alpha_p X_i\left(z\right) + \beta_p V_i\left(z+1\right) \tag{4b}$$

α_p and β_p are position control parameters; r_1 and r_2 are random variables with a range of $[0,\, 1]$; c_1 and c_2 are variables which represent the weights applied to the personal best and global best parts of the sum. Increasing c_1 relative to c_2 will force individual particles to move slowly towards the globally best seen position, and concentrate on exploring the area around their own locally seen best objective function evaluation result. ω_t represents an inertia, such that particles

Initialisation
for $i = 1$ to Swarm Size **do**
  Set X_i randomly within the search range
  Set V_i randomly within the permissible velocity range
  Assign the particle's best historical position $\mathbf{p}_i$ to the current (initial) position
  Evaluate the objective function at the particle position
end for
Identify p_g, the swarm's best result
while Stopping Conditions Unmet **do**
  for $i = 1$ to Swarm Size **do**
    Initialise random variables r_1 and r_2
    Update particle velocity and position
    Evaluate the objective function at the new particle position
    Update the particle's best known position, $\mathbf{p}_i$, if the objective function at the current position is higher than that at $\mathbf{p}_i$
  end for
  Identify p_g, the swarm's best result
end while

Algorithm 1: PSO Algorithm

can resist the pull of their local and global attractors. This variable can alter the swarm behaviour, allowing it to perform an aggressive local search if there is very little inertia, or a more wide-ranging search with a larger inertia. The inertial variables can be altered as the algorithm progresses, and there are many strategies [16] for moving from a larger value at the start of the algorithm to a smaller value at the end. The object of this is to allow a wider ranging search over a larger area, and then decreasing the mobility of the particles to perform a more thorough local search as the algorithm starts to converge upon an optimum. In this work, we varied the inertia values linearly between desired starting and end points of c_s and c_e respectively, over a set number of algorithm iterations $t_{\max}$. The inertia ω_t at time step t is shown in Equation (5).

$$\omega_t = \left(\omega_s - \omega_e\right)\left(t_{\max} - t\right)\frac{1}{t_{\max}} + \omega_e \tag{5}$$

5. ALGORITHMS

Many variants of the PSO algorithm have been developed, and several popular variants are applied to the acoustic source localisation problem by using the SRP as the objective function. In this study, we make use of the inertial weight PSO [15]; Trelea's Type 1 and Type 2 PSO [17], and Clerc's Type 1″ PSO [18].

In addition to the standard variants used, we propose two strategies which attempt to adapt the method to acoustic data. As sampling the SRP function around about head height has proven beneficial to localisation [4], we adopt a similar strat-

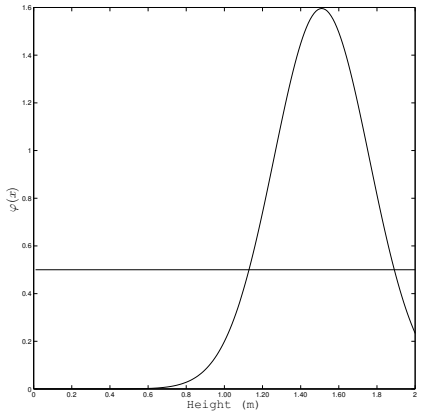


Fig. 2. Mixture model probability density function over height

egy here. When the particles are first initialised, they are spread uniformly across the ground-plane and non-uniformly in the height axis. The heights used to initialise the particles are drawn from a mixture of a Gaussian distribution centred around a height where an acoustic source has been found in a previous audio frame, and a uniform distribution over the entire height of the room. This probability density function (PDF) is shown visually in Figure 2, and the mixture model is defined mathematically in Equation (6). In this model, the parameter α_0 is the mixing parameter, which in our experiments was set to 0.95, which favours the Gaussian at head-height.

Head height is represented by μ_h; the height of the room by h_r and the variance of the Normal distribution by σ_h^2. The uniform distribution covers the whole range of the height of the room, and the variance of the Gaussian is dependant on the position of the particle on the ground-plane, $\mathbf{p_2}$, thus, the entire PDF is conditional on this parameter. The value for the variance is chosen such that, as the particle gets closer to, $\mathbf{p_1}$, the previous discovered position of a source on the ground-plane, the variance approaches 0.

$$\begin{aligned} \varphi\left(z \mid \mathbf{p_2}\right) &= \alpha_0 \mathcal{N}\left(\mu_h,\, \sigma_h^2\right) + (1-\alpha_0)\,\mathcal{U}\left(0,\, h_r\right) \\ \sigma_h^2 &= q\left(|\mathbf{p_2} - \mathbf{p_1}|\right) \end{aligned} \tag{6}$$

As a particle's position on the ground-plane becomes further away from the discovered position, we can be less confident that there will still be a speaker at that height, and so we use a sigmoid function to gradually increase the variance as the distance l between the particle and the previous speaker position increases, as shown in Equation (7). This equation is a scaled error function, with coefficient α_2 which determines how quickly the variance changes with l. Coefficient α_1 determines the maximal value of the function, and is chosen such that 99% of the Normal PDF lies over a range half a meter below and half a meter above head height. As such, over most of the room, the proposed PDF approximates a typical implicit assumption made by source localisation strategies that only a subset of the possible heights in a room need be explored [3], as speakers are rarely to be found on the floor. Nevertheless, the mixture model ensures that particles can start exploring these unlikely areas, as it is not impossible that they occur. The parameter α_2 is chosen such that at $l = 0.5$m, the function reaches its maximum, which gives a speaker in the same position between audio frames a radius of personal space, such that under normal circumstances, they are the only person within their own personal space. This means that only their approximate height should be explored first, and as the boundary of personal space is approached, it is more likely that another speaker could be present, and so the algorithm should aim to return to a less biased and more thorough search.

$$\mathrm{q}(l) = \alpha_1 \operatorname{erf}\left(\alpha_2 l\right) \tag{7}$$

With this strategy, we hope to largely ignore spurious interference sources at different heights, whilst at the same time allowing the search to cover the vertical range of the search space to allow for speakers at different heights. In contrast to [4], where particle heights are chosen from the distribution at each algorithm iteration, this scheme only chooses heights from the proposed distribution at the initialisation of the algorithm on each audio frame. The effect of this is to allow a source present in two consecutive audio frames to be discovered relatively quickly in the second frame because one dimension of the search requires less exploration, whilst still allowing a potential change of speaker to be detected.

Secondly, we also recognise that the search dimensions of a room's length and breadth are generally larger than the vertical dimension, and the search over height can conceivably be made faster by quickly moving towards a more intensive local search over height from an initial broad search of the entire vertical range of a room. In contrast, it is desirable to spend a relatively long time exploring the ground-plane broadly, before allowing the inertial components to let the search focus more closely on an increasingly small area. To this end, the inertial coefficients are modified in Equation (4a) to act differently on each dimension of the problem. The scalar ω_t is replaced by a vector as shown in Equation (8), where each search dimension is given its own inertia value. The Hadamard product is used to apply a separate inertia weight to each of the x, y, and z axes, as shown in Equation (9). The height axis is given a lower starting inertial weight than the other two, and in this way, particles should be kept at an appropriate height, and it is hoped that this should lead to a faster convergence. In our experiments, $\boldsymbol{\omega}_t$ continues to decrease linearly according to Equation (5), where the scalar ω_t

is replaced by the vector $\boldsymbol{\omega}_t$. Each component is given its own start and end value, and we choose $\omega_{s,\,x} = \omega_{s,\,y} = 0.09$; $\omega_{s,\,z} = 0.04$ and $\omega_{e,\,x} = \omega_{e,\,y} = \omega_{e,\,z} = 0.01$, all over $t_{\max} = 50$. Note that the absolute scale of the PSO variables is determined by the scale of the problem at hand - assigning particles too high a speed means that they will jump from one end of the room to the other, but this is complicated by the time scale being in sub-second discrete steps. Importantly, the relative initial inertia for the vertical direction is set much lower than that of the two ground-plane axes, allowing the search to become more local over height in fewer iterations.

$$\boldsymbol{\omega}_t = \begin{pmatrix} \omega_{x,\,t} \\ \omega_{y,\,t} \\ \omega_{z,\,t} \end{pmatrix} \tag{8}$$

$$\begin{aligned} V_i\left(z+1\right) = \boldsymbol{\omega}_p \circ V_i\left(z\right) + c_1 r_1 \left[\mathbf{p}_i\left(z\right) - X_i\left(z\right)\right] \\ + c_2 r_2 \left[\mathbf{p}_g\left(z\right) - X_i\left(z\right)\right] \end{aligned} \tag{9}$$

To evaluate the effectiveness of the algorithms studied, the performance in terms of computational complexity and localisation error were studied over a range of different swarm sizes. The localisation results were filtered using a Kalman filter, with parameters chosen to represent a very small process noise covariance, and relatively large observation noise covariance. A relatively large observation noise covariance was chosen primarily to deal with spurious observations not corresponding to a speaker, which typically (although not exclusively) occur frames containing pauses between speech segments. The number of iterations (epochs) of the PSO algorithm was measured for each variant, as was the average location error (ALE), which is simple the root mean square (RMS) error, given the localisation results and the known source locations. The total number of FEs required by each algorithm was directly related to the number of epochs used. Because the swarm size did not change between epochs in an audio frame, the number of FEs required is approximately the number of epochs required for the frame, multiplied by the swarm size used. This must be corrected by subtracting the number of FEs *not* evaluated each epoch due to the boundary conditions imposed.

6. BOUNDARY CONDITIONS

In optimisation tasks such as the speaker localisation problem, it doesn't make sense to try and evaluate the objective function beyond some bounds on the function range. In the localisation case, this simply means that there should be hard limits on the particle positions, which correspond to the walls of the room being studied. There exist several strategies for dealing with particles whose speeds would otherwise take them out of these bounds [19, 20], and we investigate some of these to determine if any is particularly appropriate to the problem.

A likely candidate strategy is to simply not evaluate the objective function for any particle which escapes the boundaries of the room, with the hope that they will quickly be pulled back in to the room. This has the advantage of potentially reducing the number of FEs used overall. Other strategies include simply saturating the positions of any escaping particles at the boundary locations; making particles wrap around the bounds along any dimension which they attempt to escape from, and bouncing particles against the boundaries such that their speed is conserved, but the direction of their velocity is reversed along the dimension of the bound encountered.

7. SNR

The algorithms must be usable over a wide range of SNR conditions, and as such, must be robust to fairly high levels of acoustic noise. To evaluate this, the recorded data set consists of speakers who are each progressively further away from the microphone array. This is intended to effectively decrease the SNR with each successive speaker, due to the inverse square law. This effect of SNR change is more robustly tested using the simulated data set, as the acoustic power of the noise signal added to the microphone signals of the simulated data set can be varied to consider a wide range of precisely known SNRs.

8. EXPERIMENTAL RESULTS

Table 1 summarises the important PSO parameters used in our experimental work. Each algorithm was tested on both the recorded and simulated data sets over several minutes of audio consisting of speech with natural pauses, for example between paragraphs. Note that in these frames, the target position is not acoustically active and so the result of the optimisation algorithm within these frames is effectively noise. Each algorithm was trialled over the same data 100 times, resulting in thousands of audio frames processed per algorithm. Each algorithm therefore underwent a Monte Carlo simulation, so that average values for the metrics for each algorithm variation can then be taken as indicative of the performance of that variant.

Figure 3 shows that the number of PSO epochs does not vary significantly with swarm size when the modified algorithm variants are run under the simulated environment. In contrast, there is a decrease in the required number of epochs for two of the original PSO algorithms - both of the Trealea variants. Figure 5 shows a similar trend for the recorded data, although the decreasing epochs requirement is much less pronounced. Figure 4 shows the linear scaling of FEs required against swarm size, which gives an indication of the computational complexity of acPSO based techniques in in terms

Parameter	Value
Maximum Iterations	500
c_1	0.002
c_2	0.02
$t_{\max}$	50
g_{thresh}	1×10^{-55}
g_{epochs}	70
$\omega_{s,\,z}$	0.04
$\omega_{s,\,x},\ \omega_{s,\,y}$	0.09
$\omega_{e,\,x},\ \omega_{e,\,y},\ \omega_{e,\,z}$	0.01

Table 1. Key PSO Parameters

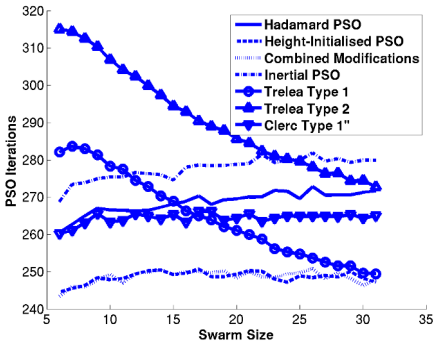


Fig. 3. PSO Epochs vs. Swarm Size, Simulated Environment

which can be compared directly to, for example, SRC, which uses between tens and hundreds of FEs [3].

In both cases, the Hadamard product inertial weight shows a decrease in the number of epochs required for convergence, although this effect is less pronounced on the recorded data set. When the two new variants are used together, there is a further reduction in epochs, however this is no more significant than simply using the height-initialised variant on its own.

Figure 6 shows that adjusting the initial height of the particles leads to a slight increase in the localisation error on the simulated data set, however, the localisation error achieved on the recorded data set shows no significant deviation from that of the original Inertial PSO. As with the epochs metric, the combination of the new PSO strategies shows no improvement over using the height-initialised variant on its own. This suggests that using them together represents a trade-off between any potential advantages of the height-initialised variant and the Hadamard variant. Rather than a cumulative effect of increasingly improved computational performance, there is

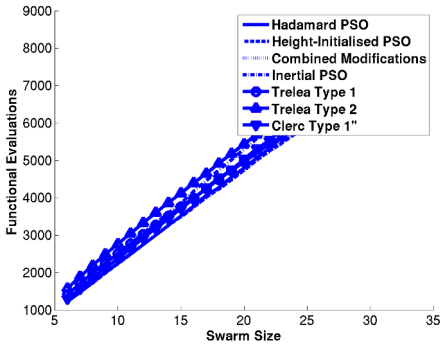


Fig. 4. FEs vs. Swarm Size, Simulated Environment

Fig. 5. PSO Epochs vs. Swarm Size, Recorded Environment

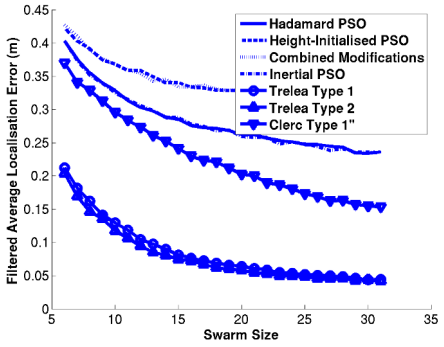


Fig. 6. Filtered ALE vs. Swarm Size, Simulated Environment

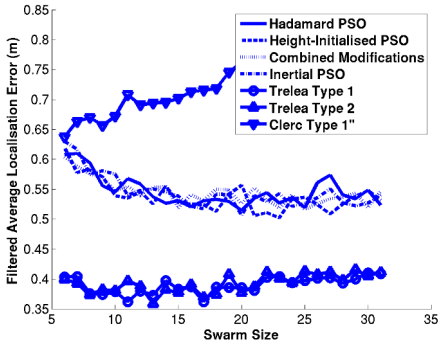


Fig. 7. Filtered ALE vs. Swarm Size, Recorded Environment

a limit to what can be achieved.

These experiments were repeated for each speaker in each data set, allowing the effect of different source positions and a rough change in SNRs to be investigated. The results were consistent across speakers, in that continuously increasing the swarm size has diminishing returns in both accuracy and time to converge. The lack of significant change across sources at different SNRs in the recorded data set prompted the more thorough investigation into the robustness of the algorithm to acoustic noise using a noise power sweep on the simulated data.

Figure 8 shows the effect of SNR on the epochs required for the Hadamard variant of the algorithm at various swarm sizes. Whilst there is a sharp decline in the required Epochs as the SNR increases, this trend only represents a small decline of the number of FEs required, as shown in Figure 9. This pattern repeats itself over each of the PSO variants considered.

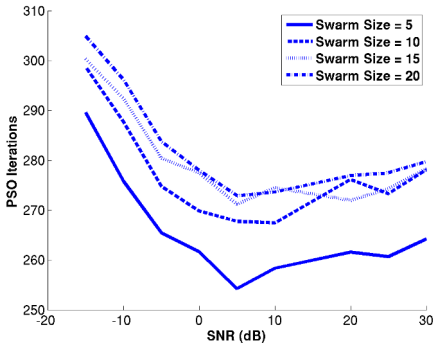


Fig. 8. PSO Epochs vs SNR

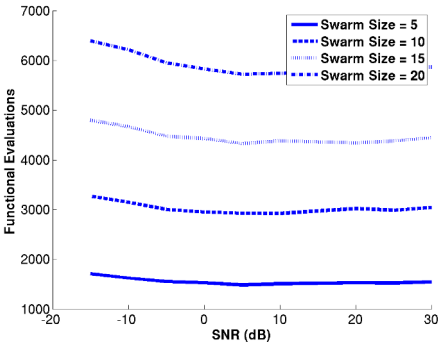


Fig. 9. Functional Evaluations vs SNR

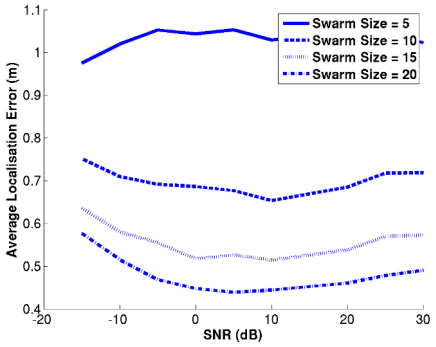


Fig. 10. ALE vs SNR

Similarly, the raw localisation error is graphed against changing SNR in Figure 10. Note that this metric is presented as calculated without the application of the Kalman filter, to demonstrate how the raw observations change with SNR, rather than a filter's ability to deal with increasingly noisy observations. There is a clear trend in all but the smallest of swarm sizes, of an increase in localisation error for negative SNRs, which is to be expected. This suggests that the underlying objective function, the SRP, is itself robust to acoustic noise, and consistently produces a peak at a speaker location which can be localised by the PSO technique. So long as such a peak is consistently created by the objective function, then PSO techniques should generally be able to find them. This is not unreasonable, as the technique was chosen based on it's ability to work with non-smooth objective functions which would otherwise be hard to optimise using gradient based methods.

9. CONCLUSIONS

The performance of PSO techniques was investigated when applied to the audio source localisation problem, and it was found that the method produces results which have an acceptable margin of error and converge using a low number of algorithm epochs. The number of FEs used in each case was also recorded for direct comparison with existing techniques, and it was found that this metric varied approximately linearly with the swarm size used. We also found that an improvement in the computational complexity metric can be obtained by adapting the methods to the specifics of the problem. These methods allow an acceptable level of localisation error to be maintained whilst minimising the swarm size required, and therefore bringing the required number of FE down to the order of several thousand.

Our results are in agreement with the general rule for particle swarms that large swarm sizes are unnecessary, as there is no large gain in speed or accuracy. The number of FEs is directly proportional to the swarm size, and for small to medium sized swarms the number of FEs is in the order of thousands, which is a significant improvement over the SRC method. SRC methods, particularly the height-estimated version, can achieve similar accuracy using in the order of tens of thousands of FEs, and this work represents an order of magnitude improvement in the search speed without sacrificing accuracy.

10. REFERENCES

[1] R. Parisi, P. Croene, and A Uncini, "Particle swarm localization of acoustic sources in the presence of reverberation," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, May 2006, pp. 4 pp.–4742.

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

[3] Hoang Do, H.F. Silverman, and Ying Yu, "A real-time SRP-PHAT source location implementation using Stochastic Region Contraction (SRC) on a large-aperture microphone array," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, 4 2007, vol. 1, pp. I–121 –I–124.

[4] Ashley Hughes, James R Hopgood, and Neil Robertson, "Height approximation for audio source localisation and tracking," in *21st European Signal Processing Conference 2013 (EUSIPCO 2013)*, Marrakech, Morocco, Sept. 2013.

[5] Greg Welch and Gary Bishop, "An introduction to the kalman filter," Tech. Rep., University of North Carolina, Chapel Hill, NC, USA, 1995.

[6] Matthias Wölfel and John. McDonough, *Distant Speech Recognition*, Wiley, 2009.

[7] M.S. Brandstein, J.E. Adcock, and H.F. Silverman, "A closed-form location estimator for use with room environment microphone arrays," *Speech and Audio Processing, IEEE Transactions on*, vol. 5, no. 1, pp. 45–50, Jan 1997.

[8] U. Klee, T. Gehrig, and J. McDonough, "Kalman filters for time delay of arrival-based source localization," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 167–167, 1 2006.

[9] Xionghu Zhong and J.R. Hopgood, "Nonconcurrent multiple speakers tracking based on extended kalman

particle filter," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 31 2008-april 4 2008, pp. 293 –296.

[10] Xionghu Zhong, *A Bayesian framework for multiple acoustic source tracking*, Ph.D. thesis, The University of Edinburgh, 2010.

[11] E.A. Lehmann and A.M. Johansson, "Diffuse reverberation model for efficient image-source simulation of room impulse responses," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 6, pp. 1429 –1439, Aug. 2010.

[12] Rudolph Emil Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[13] Eric A. Lehmann, Anders M. Johansson, and Sven Nordholm, "Modeling of motion dynamics and its influence on the performance of a particle filter for acoustic speaker tracking," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA'07)*, New Paltz, NY, USA, October 2007, pp. 98–101.

[14] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, Nov 1995, vol. 4, pp. 1942–1948 vol.4.

[15] Yuhui Shi and R. Eberhart, "A modified particle swarm optimizer," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, May 1998, pp. 69–73.

[16] J.C. Bansal, P.K. Singh, M. Saraswat, A. Verma, S.S. Jadon, and A. Abraham, "Inertia weight strategies in particle swarm optimization," in *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on*, Oct 2011, pp. 633–640.

[17] Ioan Cristian Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317 – 325, 2003.

[18] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 1, pp. 58–73, Feb 2002.

[19] Said Mikki and Ahmed Kishk, "Improved particle swarm optimization technique using hard boundary conditions," *Microwave and Optical Technology Letters*, vol. 46, no. 5, pp. 422–426, 2005.

[20] S.M. Mikki and A.A. Kishk, "Hybrid periodic boundary condition for particle swarm optimization," in *Antennas and Propagation Society International Symposium, 2007 IEEE*, June 2007, pp. 1581–1584.