Speech-based Localization of Multiple Persons for an Interface Robot

Gradje Klaassen

Speech-based Localization of Multiple Persons for an Interface Robot

Gradje Klaassen

Master of Science Thesis in Artificial Intelligence,
Intelligent Autonomous Systems
University of Amsterdam

Supervised by Dr. Ir. Ben J.A. Kröse

Approved

Date *Amsterdam*

Acknowledgements

I thank my supervisor Ben J.A. Kröse for his help and support in this project. I thank Albert van Breemen of Philips Research for providing the iCat microphone system for the experiments. Last but not least, I thank Wojciech Zajdel for his guidance and advice and for having an approximately infinite amount of patience with me.

Contents

Chapter 1

Introduction

Figure 1.1: Philips iCat interface robot. The two microphones are mounted on the sides of the bottom panel.

Robots are conveniently controlled by a human operator with spoken commands, since voice is a natural communication medium for humans. In order to successfully carry out a command, a robot needs to known which of the possibly many people gave the command and where this person is. In this thesis we present a particle filter-based algorithm for localization of multiple speakers, in an environment where there is only one person speaking at a time. The algorithm incorporates person-specific voice features in order to distinguish between the speakers. The voice features are supported by location estimates: *azimuth* angle measurements obtained by a pair of microphones. We test our approach using the microphone system of the Philips iCat interface robot.

1.1 Problem Description

As a part of our speaker localization problem we address the problem of distinguishing between multiple speakers in a noisy and reverberant environment using a pair of microphones. Our approach is based on tracking azimuth angle measurements for every person. When the

speakers' paths cross, the azimuth features will not have enough resolution to distinguish between the speakers. In order to disambiguate such difficult cases, we attempt to combine azimuth cues with speaker-specific voice features, where we will focus on extracting formant frequencies. We also assume that only one of the speakers is active at a time.

We consider a probabilistic framework, where every speaker is described with a latent state variable that includes the "true" azimuth angle and formant frequencies. Given a segment of the input signals, we update our beliefs about states of all speakers. The beliefs become a basis associating the input segments with one of the speakers. Our implementation applies a sample based version of the joint probabilistic data association filter to compute the interesting association probabilities.

1.2 Motivation

Natural interaction between human and robot is one of the research goals of artificial intelligence. One aspect of natural interaction for the robot is to interpret in an intelligent manner various audio signals, and more specific to detect and process the human voice. Whenever voice-controlled robots operate in a human-crowded environment, ambiguity problems emerge such as determining the (active) speaker in the crowded environment. Possible support for ambiguity-resolution techniques can be provided by sound source localization algorithms. Thus, for various tasks it is desirable that the robot knows the location of the speaker.

Furthermore, if the robot is able to detect the direction of the sound source (the speakers voice), it can turn towards the speaker and associate the voice with additional visual information. As a consequence, this enables the robot to track the active speaker and focus its attention towards this person, establishing a more natural human-robot interaction.

This issue becomes more important, as current robots are moving out of the factory floor into environments inhabited by human. Examples are museum or exhibition robots [60], [1], care-for-elderly robots [42], office [2] and entertainment robots [11]. In their role as guide, companion or servant these systems have to interact with the humans they encounter. Therefore, in robot perception and human-robot social interaction, it is becoming more important to incorporate a robust speaker tracking module.

1.3 The Human Auditory System

For a robot, it is difficult to match the hearing capabilities of a human. The human sophisticating audio sensing mechanism takes various physical effects into account like the acoustic shadowing created by the head and the reflections of the sound by the two ridges running along the edges of the outer ears [61]. This ability enables humans to locate sound sources in three dimensional space.

Several methods have been developed in order to mirror the human sound source location capability. Typically, one relies on time-difference of arrival measurements between the signals from a pair of microphones. These measurements indicate the azimuth angle between the geometrical centre of the microphone pair. Thus, with the use of only two microphones we cannot compute the distance between the microphones and the sound source, we can only obtain an estimate of the relative position of the sound source. Therefore localization

with two microphones is incomplete: three-dimensional localization of the sound source is not possible. Further difficulties arise from imprecise readings when the sound source is in the same axis as the microphone pair and the robot is unable to distinguish if the sounds are coming from the front or the back.

In order to compensate for the high level of complexity of the human auditory system, one can increase the sensing resolution by using multiple microphone pairs [47]. With the use of an array of eight microphones one is able recover the three-dimensional location of the sound source [8]. From each microphone pair that is part of the microphone array we can recover a vector that indicates the sound source location. Thus from the centre of each microphone pair a vector can be formulated. A three-dimensional source location can be recovered from the intersection of these vectors. However, usually more than one intersection point is simultaneously present, due to noise and calibration errors. To tackle this problem in this over-determined system, an outlier detector algorithm is proposed by [31] The incorrect vectors are detected and discarded, leaving only the vectors that are regarded as the most reliable ones giving one intersection point.

Another approach to locate sound sources proposed by [44] is to include a learning model. Through self-organization by perceptual cycle (iteration of the sensing and moving processes) the robot can learn to rotate its head to the sound direction. The rationale of this approach is based on the human abilities to locate sound sources by repeatedly gathering information about -and interacting with- the environment. Thus, no explicit supervision from the environment is assumed to acquire smoother and quicker motor control abilities. Neural networks that consist of a visual estimation module and an auditory estimation module are responsible for the learning process.

An alternative to the signal-processing-motivated algorithms for acoustic source localization is provided by the work of Auditory Scene Analysis (ASA) or Computational Auditory Scene Analysis (CASA) [7]. Inspired by the human ability to analyse the auditory environment, the localization of acoustic sources and the perception of speech simultaneously with just two receivers, CASA research aims to resemble the human auditory system with physiologically oriented models. Such models should approximate the human ability to interpret sound mixtures as the combination of distinct acoustic sources [21].

Even from this brief overview, it is obvious that the research [1] is inspired by the human sound source location capability, together with human sophisticating mechanism for voice recognition allowing for accurate speaker identification.

1.4 Overview Thesis

The rest of the thesis is organized as follows: chapter two discusses several methods for tracking a single speaker. The content of chapter three is dedicated to methods for extracting voice features. In chapter four, extended versions of the methods presented in chapter two are discussed. The methods presented in chapter four are designed for tracking multiple speakers and typically require additional features. Therefore we present a multiple target tracking algorithm that incorporates voice features in addition to the azimuth features. Chapter five is reserved for the description of the experiments and discussion of the results. The final chapter six presents the conclusions of the experiments and future work.

[1] That is, the research with the aim to create robust auditory systems for robots.

Chapter 2

Localization Of A Single Source

This chapter discusses basic methods to track a single speaker. First, different measuring techniques are discussed that form the basis for the *localization function*. Since the experiments were conducted with the restriction of using only one microphone pair, the measuring techniques are discussed in the light of this restriction. Next, two different probabilistic approaches to track the sound source location are discussed and compared. The presented principles for tracking a single source will be extended in order to track multiple sources which will be described in chapter four. Finally, single source tracking experiments conducted with various tracking algorithms are discussed.

2.1 Localization Measurements

As mentioned in the previous chapter, it is not possible to obtain the distance between the speaker and the robot when the only available sensory information is provided by two microphones. With this restriction, tracking the speaker results in determining the relative position of the speaker with respect to the microphones. The basis for a simple method to locate a sound source is to extract the *time-difference of arrival* (TDOA) between the two signals that are received by a pair of microphones. A change in location (that is if the movement is around the pair of microphones) usually corresponds to a change in the TDOA between the two signals. The TDOA can be transformed to an arriving angle, which corresponds to relative position of the sound source to the microphone pair [6]. Our localization function employs the above described concepts for sound source localization.

2.1.1 Azimuth Angle

The azimuth angle θ specifies the direction of the sound wave when it strikes the microphone pair. Consider Fig. 2.1A (that represents an extremely simplified scheme of a human listener) where the sound arrives at the right "ear" before the left "ear". From Fig. 2.1A we see that:

$$d = a\,\theta + a\,sin\theta, \quad 0 \leq \theta \leq \pi/2, \tag{2.1}$$

where a denotes the radius of the spherical head. Consider the simpler scheme in Fig. 2.1B where the distance is approximated as $d = 2a\,sin\theta$. If we divide the distance d by the speed of sound $c = 342m/s$, then we obtain the formula for *interaural time difference* which provides us with the time shift τ:

$$\tau = \frac{d}{c} \approx \frac{2a}{c}(sin\theta), \quad 0 \leq \theta \leq \pi/2. \tag{2.2}$$

Thus, in order to estimate the direction θ of the incoming sound, the localization function first extracts the time shift τ between the two received signals, second transforms the time shift into the corresponding azimuth angle according to equation 2.2.

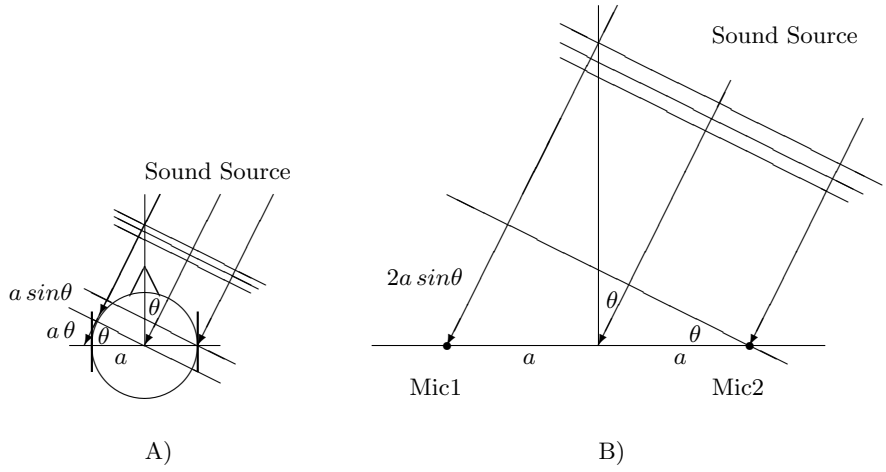

Figure 2.1: A) Directional cues for human listener. B) Simplified scheme for finding the direction of incoming sound wave with two microphones.

2.1.2 Time Difference Of Arrival Estimation

In order to select and evaluate the potential true source locations we first choose the appropriate time shifts (TDOAs) for the localization function. Each possible "true" sound source candidate should correspond to the TDOA that would have been observed, if the sound source was active at that potential location. The TDOA candidates for "virtual steering" are evaluated by applying *cross-correlation* to the signal. In order to give a complete description of the evaluation we first present a mathematical model of the involved signals. In addition the next paragraph aims to illustrate the challenges involved in extracting the "true" TDOA from "real life" recordings.

Signal Model Consider a spatially separated pair of microphones (M_l, M_r) and let $s_l(t)$ and $s_r(t)$ denote the recorded sensor signals in a noisy and reverberant room. If we assume a single source, then the discrete-time signal can be modelled mathematically as:

$$s_l(t) = h_l(t) \odot s(t) + n_l(t), \tag{2.3}$$
$$s_r(t) = h_r(t) \odot s(t) + n_r(t),, \tag{2.4}$$

where $h_l(t)$ denotes the complete acoustic impulse response between the sound source and microphone M_l, the additive term n_l denotes the noise (assumed to be uncorrelated with the source signal and n_r) and $\odot$ denotes the convolution operator.

Surfaces present in the room can scatter the signal producing "images" of the true source. Aside from the true sound and its "images" there is usually a noise component present in

the signal. This will lead to multi-valued solutions: spurious maxima in the cross-correlation function. Consider the following (real life [1]) scenario. Aside from striking the ears of the listener, the sound waves strike the walls, floor, ceiling and the objects present in the room. Depending on the material of the walls etc. the original sound is reflected and intermixed with the (new) direct sound. A part of this mixture is again reflected and intermixed and so on. From this scenario its clear that a deformed sound image arises, compared to the original (direct) sound that originates from the sound source. In order to estimate the TDOA between two signals, one typically aims to compute the relative time shift τ_{lr} of the direct-path time differences of arrival τ_l and τ_r of the recorded sensor signals. Thus the impulse responses from the sound source to sensors have to be separated in direct path and multipath terms. We define τ_{lr} as $\tau_{lr} = \tau_l - \tau_r$ and rewrite formulae (2.3) and (2.3) as:

$$s_l(t) = \alpha h_l^{mod}(t) \odot s(t - \tau_l) + n_l(t), \tag{2.5}$$

$$s_r(t) = \alpha h_r^{mod}(t) \odot s(t - \tau_r - \tau_{lr}) + n_r(t), \tag{2.6}$$

with α denoting the attenuation factor and $h_l^{mod}(t)$ denoting the modified impulse response. This response is a modified (scaled) version of the original impulse response $h_l(t)$ without the direct-path time shift τ_l (similarly for $h_r^{mod}(t)$).

2.1.3 Cross-Correlation

Cross-correlation is a function that is commonly used to compare two signals. In the context of estimating the relative position of a sound source, cross-correlation is used to estimate the "true" TDOA between the two signals. The cross-correlation method shifts the signals relative to each other with some time lag. A property of cross-correlation is that the function is maximized when the signals are identical. Therefore, the time shift that maximizes the cross-correlation between the signals corresponds to the "true" TDOA between two signals[2]. The (un-normalized) cross-correlation between signals s_l and s_r at time shift τ is defined as:

$$R_{l,r}(\tau) = \sum_{t=-\infty}^{+\infty} s_l(t) s_r(t - \tau). \tag{2.7}$$

Usually the signals have a finite length N. Therefore it is convenient to use following form:

$$R_{l,r}(\tau) = \begin{cases} \sum_{t=1}^{N-\tau} s_l(t)\, s_r(t + \tau), & 0 \le \tau \le N - 1, \\ \sum_{t=1}^{N-\tau} s_l(t)\, s_r(t - \tau), & N - 1 \le \tau \le 0, \end{cases} \tag{2.8}$$

where again τ denotes the time shift. Searching for the maximum in the cross-correlation function can be restricted by searching in the range of potential τ values defined in the interval $[-\tau_{max}\,, \tau_{max}]$. The term τ_{max} denotes the upper bound range of time shifts that are evaluated. One would typically choose τ_{max} to be the maximum physical possible time shift, which is dependent on the physical configuration:

$$\tau_{max} = \frac{d}{c} F_s, \tag{2.9}$$

[1] Its nearly impossible to find circumstances for a space that is free of reflection in real life. One would have to stand on the top of a mountain with a sharpened peak in order to come close to experiencing a space that is free of reverberation.

[2] With the assumption that the signals are not corrupted by reverberation and noise.

where d denotes the distance between the microphones, c denotes the speed of sound and F_s denotes the sampling frequency. However, allowing time shifts outside the interval $[-\tau_{max}\,,\,\tau_{max}]$ for possible TDOA candidates could be useful as a confidence measure for the found maximum [4]. Basically, if the (global) maximum of the cross-correlation function is found inside the interval $[-\tau_{max}\,,\,\tau_{max}]$ while the cross-correlation is computed over the interval $[-(\tau_{max}+n)\,,\,\tau_{max}+n]$ where n denotes a positive integer, then this maximum is regarded as reliable. If on the other hand, the (global) maximum is not found inside the interval $[-\tau_{max}\,,\,\tau_{max}]$, then the (local) maximum inside this interval is not regarded as highly reliable.

2.1.4 Crosspower Spectrum Phase

Equations (2.3) and (2.4) assume that the true sound source is intermixed with reverberant components and corrupted with some additional noise. Since the manifestation of these components is usually easier to detect in the frequency domain, an alternative approach for TDOA estimation considers the signals s_l and s_r in the frequency domain. This approach allows to remove noise artefacts that are typically present in real life situations.

In order to deal with the reverberation and noise effects pre-processing is applied to the signal. *A priori* knowledge of these components (for each frame) would increase the accurateness of the time delay estimate. However, in real life situations the noise and reverberation effects have a dynamic character: windows and doors that are opened and closed influence the amount of noise and reverberation in the signal. Therefore acquiring accurate knowledge about the statistics of the involved artefacts becomes a challenging task.

A method that extracts the TDOA from the signals by using and modifying information in the frequency domain is usually referred to as the *generalized cross-correlation*(GCC). This technique computes the TDOA as the inverse Fourier Transform of the received signal crosspower spectrum scaled by a weighting function. This method pre-filters the signal before computing the correlation [48], [49].

Definition GCC Let $s(t)$ denote the data (in the time domain) received at time t and let $\mathcal{F}\{.\}$ denote the Fourier transform and let $S(\omega) = \mathcal{F}\{s(t)\}$ represent the data received at time t in the frequency domain. The cross-correlation between the signals s_l and s_r according to the GCC method is defined as:

$$R_{l,r}(\tau) = \int_{-\infty}^{\infty} G(\omega) S_l(\omega) S_r^*(\omega) \exp^{(j\omega\tau)}\, d\omega, \tag{2.10}$$

where S_l denotes the Fourier transformed signal received by microphone M_l and S_r^* denotes the *conjugate* of the Fourier transformed signal received by microphone M_r. The conjugate of S_r is defined as:

$$conj(S_r) = real(S_r) - j\ imag(S_r), \tag{2.11}$$

where $G(\omega)$ is a weighting term (responsible for the pre-filtering).

PHAT weighting function The *phase transform* (PHAT) weighting function was introduced in [36] and is defined as:

$$G(\omega) = |S_l(\omega) S_r^*(\omega)|^{-1}. \tag{2.12}$$

This weighting function places equal importance on each frequency, by dividing the spectrum by its magnitude. By applying this *whitening* filter the magnitude of the crosspower spectrum is flattened, preserving only information about the phase differences between the signals s_l and s_r. In other words, applying the PHAT function results in a constant energy concentrated on the correct TDOA caused by high coherence between the two signals at the corresponding lag.

A linear phase shift in the crosspower spectrum corresponds to a time shift in the cross-correlation [3]. Aiming to find the relative time shift τ_{lr} (see also equation (2.6)) such that the two compared signals have the maximum coherence [3], we can write the crosspower spectrum of the two measured signals $s_l(t)$ and $s_r(t)$ as:

$$C_{s_l s_r}(\omega) = \left[\int_{-\infty}^{\infty} s_l(t) \exp^{-j\omega t}\, dt\right]^* \int_{-\infty}^{\infty} s_l(t-\tau_{lr}) \exp^{-j\omega t}\, dt = \exp^{-j\omega\tau_{lr}} P_l(\omega), \quad (2.13)$$

where $P_l(\omega)$ denotes the power spectrum of s_l. Since the measurements are usually not recorded in a noiseless and non-reverberant room, noise and reverberation components are present in the signal. Therefore the cross-correlation peak is sharpened by "whitening" the signal:

$$\frac{C_{s_l s_r}(\omega)}{|C_{s_l s_r}(\omega)|} = \exp^{-j\omega\tau_{lr}}\,. \quad (2.14)$$

Thus, the cross power spectrum is normalized to remove all magnitude information, preserving only phase information, where the energy is concentrated on the main phase. Assuming that noise and reverberant signals are not dominant over the true sound source, the dominant phase of the signal corresponds to the true sound source.

Complexity Analysis Cross-correlation Applying cross-correlation in the frequency domain yields an advantage in the computational load: computing cross-correlation in the time domain implies using convolution whereas applying cross-correlation in the frequency domain implies multiplication: $R_{s_l s_r} = s_l \odot s_r = \mathcal{F}^{-1}(S_l S_r^*)$. Thus applying cross-correlation in the frequency domain yields a complexity reduction [54]. If the windowed segment contains N samples the complexity of computing the cross-correlation using equation 2.7 is $\mathcal{O}(N^2)$, whereas computing the cross-correlation using equation 2.10 is $\mathcal{O}(Nlog_2N)$. Note that the complexity of computing the N-point FFT of the windowed segment for M microphones is $\mathcal{O}(MNlog_2N)$.

Alternative weighting functions In a sense, the PHAT weighting function is an extreme choice, since it whitens the input signal since there usually is no a priori knowledge about the statistics of the involved signals. The PHAT localization algorithm is robust and reliable in realistic reverberant acoustic conditions [10], however the algorithm does not take the *signal to noise ratio* (SNR) conditions into account. If the interference is dominant over the desired signal, then the phase information obtained from the PHAT function will be unreliable. Methods for estimating the noise spectra and incorporating them in the weighting functions are discussed in [10], [5]. Aiming to locate the speaker, one should aim for detecting the voice in the spectrum. Therefore the weighting function should emphasize the regions that are likely to contain voice components (that is: where the SNR is maximum). A proposed

[3] Ideally the relation between the two signals would be: $s_l(t) = s_r(t-\tau_{lr})$ this would be the case if the measurements were recorded in a non-reverberated ($h_l^{mod}(t) = \delta(t)$) and noiseless environment ($n_l(t) = 0$))

method by [61] introduces a noise masking weight and is defined as:

$$w(\omega) = max\bigg(0.1, \frac{S(\omega) - \alpha N(\omega)}{S(\omega)}\bigg), \tag{2.15}$$

where $S(\omega)$ denotes the spectrum of the signal and $N(\omega)$ the estimated noise spectrum (based on the time average of the previous $S(\omega)$). The coefficient $\alpha < 1$ is responsible for a more conservative noise estimate. Such a noise masking weight becomes near zero in the regions that are dominated by noise, whereas the noise masking weight becomes close to 1 in the regions where the (desired) signal is dominant over the noise. A second part of the weighting function is introduced in order to increase the contribution of the tonal regions of the spectrum (where the SNR is maximum). The enhanced weighting function is defined as:

$$w_e(\omega) = \begin{cases} w(\omega), & S(\omega) \leq N(\omega), \\ w(\omega)\big(\frac{S(\omega)}{N(\omega)}\big)^{\gamma}, & S(\omega) \geq N(\omega), \end{cases} \tag{2.16}$$

where the exponent $0 < \gamma < 1$ increases the weight in the regions where the signal is far dominant over the noise. The suggested values for weighting function parameters are $\alpha = 0.4$ and $\gamma = 0.3$. The resulting weighted cross-correlation function is defined as:

$$R_{l,r}(\tau) = \int_{-\infty}^{\infty} \frac{w_e^2(\omega) S_l(\omega) S_r^*(\omega)}{|S_l(\omega) S_r^*(\omega)|} \exp^{(j\omega\tau)} d\omega. \tag{2.17}$$

Similar to this approach is a pitch-based TDOA estimator introduced by [9] where knowledge about the *pitch* characteristics of speech are exploited [4]. The voice has a narrow bandwidth and is usually concentrated in lower regions of the frequency spectrum. Furthermore, the *voiced* speech segments in the signal is rich of harmonics[5]. Noise is usually not of harmonic nature. This approach attempts to design a GCC weighting function based on estimated periodicity of harmonic intervals. The excitation spectrum of human speech can be modelled as a function of a *fundamental frequency* ω_0 (also referred to as pitch, see section D.3 for further details), and harmonic dependent voiced/unvoiced mixtures. The regions in the spectrum that exhibit a distinct periodic nature are less influenced by noise and reverberation components and should therefore receive appropriate emphasis in the GCC weighting function. Note that the above mentioned approaches require additional knowledge about the involved sound sources in the signal, that needs to be estimated. The robustness of these approaches is directly related to the estimation performance.

2.2 Probabalistic Approach

As stated in section 2.1 the localization function obtains the sound source location estimates based on TDOA measurements. Reverberation and noise can cause spurious maxima in the localization function. Even though the GCC method improves the robustness of the localization function, it remains a challenging task to accurately track a moving sound source in the presence of a strong multipath and noise sources. Furthermore the inverse map from TDOA to location is *non-linear* and *ill-posed* [20], [66] (see appendix C for details). Therefore

[4] Pitch is briefly discussed in section D.3

[5] This property is briefly discussed in section D.1

it is difficult to accurately track the azimuth with a deterministic approach. Typically a state-space approach overcomes the drawback of using only current sensor data to locate the true sound source [64]. The key to this approach is that the true sound source follows a dynamical motion model between consecutive measurements, whereas there is no temporal consistency in the spurious peaks. The involved dynamics are complex and typically are not completely known. Furthermore it is difficult to describe the involved dynamics with a deterministic model. Therefore it is convenient to use a probabilistic model to describe the motion of the speaker.

2.2.1 Inference Methods

Estimating the unknown process x based on sensory data y presented in the Bayesian framework implies solving the Bayesian filtering problem [19] . The *posterior* distribution $p(x_0, x_1, \ldots, x_t|y_0, y_1, \ldots, y_t)$ includes all relevant information on $\{x_0, x_1, \ldots, x_t\}$ at time t in the Bayesian framework. Typically, in signal processing applications one is particularly interested in the so-called filtering distribution $p(x_t|y_0, y_1, \ldots, y_t)$ which is a marginal of the *posterior* distribution. Estimating recursively in time the distribution $p(x_t|y_0, y_1, \ldots, y_t)$ is also known as the Bayesian filtering problem [18]. Except under specific assumptions: linear Gaussian state space models, it is impractical to evaluate these distributions analytically. However, in real life these assumptions about the state-space are rather strong and hinder the robustness of the tracking mechanism. Discarding these simplified assumptions usually results in an intractable problem. The next section aims to provide some background for these problems. In addition the section aims to illustrate the typical intractable nature of the Bayesian filtering problem since no closed-form solutions exists to the complex high dimensional integrals that are typically involved.

2.2.2 Background

With respect to our tracking problem we are interested in the *belief* that the speaker is at some location θ. In other words we want to obtain the posterior distribution of the state space (represented in azimuth angles) conditioned on the sensor data [41]. We can denote this belief by:

$$Bel(\theta_t) = p(\theta_t|\theta_t^z, \ldots, \theta_0^z). \tag{2.18}$$

Where θ_t represents the state of the speaker at time t and θ_t^z denotes the location obtained by transforming the measurement (TDOA estimate) into the corresponding azimuth angle with the use of equation (2.2) at time t. Bayes filter relies on the assumption that object moves according to a Markovian motion model [6]. In addition the filter relies on the assumption that the measurements originate from an environment where past and future measurements are conditionally independent given knowledge of the current state. With these assumptions we can compute the posterior at time t with the use of a Bayes filter. Bayes filter estimates the belief recursively. To derive a recursive update equation we transform equation (2.18) by applying Bayes rule:

$$Bel(\theta_t) = \frac{p(\theta_t^z|\theta_t, \theta_{t-1}^z, \ldots, \theta_0^z)\, p(\theta_t|\theta_{t-1}^z, \ldots, \theta_0^z)}{p(\theta_t^z|\theta_{t-1}^z, \ldots, \theta_0^z)}. \tag{2.19}$$

The denominator is a normalization constant relative to θ_t, so we can write equation (2.19) as:

$$Bel(\theta_t) = \eta\, p(\theta_t^z|\theta_t, \theta_{t-1}^z, \ldots, \theta_0^z)\, p(\theta_t|\theta_{t-1}^z, \ldots, \theta_0^z), \tag{2.20}$$

[6] In the experiments we use a first order Markov motion model, for further details see appendix A

where:

$$\eta = p(\theta_t^z|\theta_{t-1}^z, \ldots, \theta_0^z)^{-1}. \tag{2.21}$$

With the assumption that the measurement θ_t^z depends only on the current state θ_t , equation (2.20) can be simplified further, giving:

$$Bel(\theta_t) = \eta\, p(\theta_t^z|\theta_t)\, p(\theta_t|\theta_{t-1}^z, \ldots, \theta_0^z). \tag{2.22}$$

The prediction term can be expanded by integrating over the state at time $t-1$:

$$p(\theta_t|\theta_{t-1}^z, \ldots, \theta_0^z) = \int p(\theta_t|\theta_{t-1}, \theta_{t-1}^z \ldots, \theta_0^z)\, p(\theta_{t-1}|\theta_{t-1}^z, \ldots, \theta_0^z)\, d\theta_{t-1}. \tag{2.23}$$

Giving:

$$Bel(\theta_t) = \eta\, p(\theta_t^z|\theta_t) \int p(\theta_t|\theta_{t-1}, \theta_{t-1}^z \ldots, \theta_0^z)\, p(\theta_{t-1}|\theta_{t-1}^z, \ldots, \theta_0^z)\, d\theta_{t-1}. \tag{2.24}$$

By applying the previous assumption again, equation (2.24) can be simplified further to:

$$Bel(\theta_t) = \eta\, p(\theta_t^z|\theta_t) \int p(\theta_t|\theta_{t-1})\, p(\theta_{t-1}|\theta_{t-1}^z, \ldots, \theta_0^z)\, d\theta_{t-1}. \tag{2.25}$$

The equation (2.25) is recursive of nature: the rightmost term is $Bel(\theta_{t-1})$. In order to compute $Bel(\theta_t)$ two conditional densities need to be known: $p(\theta_t|\theta_{t-1})$ which is the *motion model* (state transition density) describing the behaviour of the system, and $p(\theta_t^z|\theta_t)$ which is the *sensor model* (measurement density) describing the perception of the desired features in the real world.

In the next two sections we discuss two different filtering methods within the state-space framework: the Kalman filter and the particle filter. The first method assumes that the models used for the dynamics and measurement are linear and Gaussian. Under these assumptions the Bayesian filtering equations lead to Kalman filter equations. The second method - particle filter does not make these assumptions about the state-space, and provides a numerical solution to Bayesian filtering problem by approximating the filtering distribution.

2.2.3 Kalman Filter

The Kalman filter is a set of equations that provides an estimate of the state of a dynamic system. The filter provides a recursive solution to the linear filtering problem [65], [15]. We cannot compute the distance to the sound source with the use of only two microphones. Therefore we cannot compute the exact location in the Cartesian coordinate system. However, we can compute the relative x- and y coordinates with use of the (measured) azimuth angle θ^z (which represents the relative location) with distance 1. Thus, we transform the estimated TDOA into the azimuth angle θ^z, followed by a transform from the angle into relative x^z and y^z coordinates. These coordinates will be referred to as the measurements in this section.

$$x^z = cos(\theta^z), \tag{2.26}$$

$$y^z = sin(\theta^z). \tag{2.27}$$

Suppose we want to estimate the state x of a discrete-time controlled process. This process is described with following linear stochastic difference equation:

$$x_t = Ax_{t-1} + Bu_{t-1} + w_{t-1}. \tag{2.28}$$

With measurement equation:

$$x_t^z = Hx_t + v_t. \tag{2.29}$$

The random variables w_t and v_t represent respectively the process and measurement noise. Both w_t and v_t are supposed to be independent of each other, white and normally distributed.

$$p(w_t) = \mathcal{N}(w_t|0, Q), \tag{2.30}$$
$$p(v_t) = \mathcal{N}(v_t|0, R). \tag{2.31}$$

Therefore, the process and measurement noise are respectively sampled from process noise covariance matrix Q and measurement noise covariance matrix R. In practice the amount of noise usually varies within time, however here we assume it is constant, thus Q and R are assumed to be constant.

The matrix A describes the evolution of the system, the previous step is related to the current step without considering a driving force or process noise. Again, in practice matrix A might change with each time step but here we assume it is constant. Matrix B describes the influence of the optional control input u on state x. The matrix H relates the state x_t to the measurement x_t^z. In practice H might change with each time step but here we assume it is constant.

If we have knowledge of the process at time step $t-1$, then we can compute the *a priori* state estimate at time step t denoted by $\hat{x}_t^-$. With each measurement we can compute the *a posteriori* state estimate denoted by $\hat{x}_t$. The *a priori* e_t^- and *a posteriori* e_t estimate errors are defined as:

$$e_t^- = x_t - \hat{x}_t^-, \tag{2.32}$$
$$e_t = x_t - \hat{x}_t. \tag{2.33}$$

The a priori estimate error covariance becomes:

$$P_t^- = E[e_t^- e_t^{-T}]. \tag{2.34}$$

The a posteriori estimate error covariance becomes:

$$P_t = E[e_t e_t^T]. \tag{2.35}$$

The main objective of applying a Kalman filter is to find the best possible estimate of state x at time t.

We compute the *a priori* estimate of the state $\hat{x}_t$ with the use of equation (2.28):

$$\hat{x}_t^- = A_t \hat{x}_{t-1}. \tag{2.36}$$

Aiming to obtain an updated estimate of $\hat{x}_t$ based on the measurement x_t^z, we are interested in an equation that computes an *a posteriori* state estimate $\hat{x}_t$ as a linear combination of an *a priori* state estimate $\hat{x}_t^-$ and a weighted difference between an actual measurement x_t^z and a measurement prediction $H\hat{x}_t^-$:

$$\hat{x}_t = \hat{x}_t^- + K(x_t^z - H\hat{x}_t^-). \tag{2.37}$$

The difference between $(x_t^z - H\hat{x}_t^-)$ reflects the amount of agreement between the predicted $H\hat{x}_t^-$ and the actual measurement x_t^z. This difference is commonly referred to as measurement *innovation*, or the *residual*. The weighting matrix K is chosen to be the blending factor

or gain. We aim to obtain K such that it optimises the updated estimate.

The optimisation means minimizing the estimated error variance for the estimated state. Thus, the individual terms along the diagonal of the *a posteriori* error covariance matrix P_t should be minimized. The optimisation is accomplished by substituting $\hat{x}_t = \hat{x}_t^- + K(x_t^z - H\hat{x}_t^-)$ into $e_t = x_t - \hat{x}_t$. The result of this substitution is substituted in $P_t = E[ee^T]$. The next step is computing the expectation, then differentiating the trace of P_t with respect to K. Setting the result of these steps equal to zero, and then solving for K. One popular form of K that minimizes $P_t = E[ee^T]$ is:

$$K_t = P_t^- H^T (HP_t^- H^T + R)^{-1}. \tag{2.38}$$

The relationship between the *a priori*- and *a posteriori* error covariance matrices can be expressed by:

$$P_t = (I - K_t H)P_t^-. \tag{2.39}$$

From equation (2.38) we see that if the error covariance R approaches zero, then the gain K weights the residual more heavily. In other words, if $\lim_{R\to 0} K = H^{-1}$ than the measurements are regarded as more reliable. The residual is weighted less heavily, if the *a priori* estimate covariance matrix approaches zero. Note that: $\lim_{P\to 0} K = 0$. Thus for such cases the actual measurements are not regarded as highly reliable and more trust is placed in the predicted measurements.

The Kalman filter estimates the state of a dynamic process by using feedback control. First a prediction step is performed which results in an *a priori* estimate of the state. This prediction step or time update step (*motion model*) is responsible for projecting the current state and error covariance estimates forward in time. These first estimates are corrected in the update step or measurement step (*sensor model*). The feedback comes in the form of the measurement update equations. Updating the *a priori* estimate with a new measurement results in an *a posteriori* estimate, an improved estimate.

Note that with the Kalman filter, the azimuth angle is not tracked directly. To obtain a new estimate of the azimuth angle θ, the measured azimuth angle based on the TDOA measurement, is first transformed into relative x^z and y^z coordinates. The *motion model* and *sensor model* are computed for the relative x and y coordinates. These new computed estimates are then transformed back into a new estimate of the azimuth angle θ.

2.2.4 Particle Filter

The Kalman filter provides us with the linear minimum variance estimates of the states. However, with the assumption that the involved noises are Gaussian and that the functions for state- and measurement evolution are linear. Unfortunately this modelling is not appropriate for the speaker tracking problem, since vital information is lost.

Kalman filtering, based on Gaussian densities which, being uni-modal, cannot represent simultaneously alternative hypotheses about the true sound source location. Furthermore, as stated earlier, the transformation from the TDOA estimates to source locations are non-linear. As a consequence the state-space is non-Gaussian and non-linear [62]. As stated before, with these properties of the state-space, typically no-closed form solutions exist for computing the filtering distributions. Computing the filtering distribution in non-linear and

non-Gaussian estimation problems typically results as an intractable problem. Therefore typically these estimation problems are solved by approximating the filtering distribution [18].

Particle filtering is a method to construct a sample-based representation of the filtering distribution. Like the Kalman filter the particle filter is recursive in nature and can be divided roughly into two stages: *prediction* and *update*. In the prediction stage (*motion model*) the particles (or samples, which can be thought of as possible instantiations of the variable of interest[7]) are propagated according to the dynamic model (for further details we refer the reader to appendix A). In the update stage (*sensor model*) the particles are (re-)evaluated based on the latest sensory information available. This stage assigns a weight to each particle, enabling to define the contribution for each particle to the overall estimate of the variable. Thus in the prediction step the dynamic model is used to simulate the effect of the possible actions (movements) on the filtering distribution. The sensory information (measurement) is used in the update stage to update the particle weights to accurately describe the filtering distribution of the (changing) azimuth angle. In the earliest studies, the particle filter algorithm consisted of only the prediction and the update stage. To avoid the degeneracy of the particle set where the mass of the filtering distribution tends concentrate towards a unique particle with a very high weight, a resampling step has been added to enforce the particles in multiple areas with high likelihood. Table 2.1 describes a generic particle filtering algorithm for sound source localization based on TDOA measurements [64].

Form an initial set of particles $\{\boldsymbol{\alpha}_{0,k} \quad k = 1 : M\}$ and give them uniform weights $\{w_{0,k} = 1/M \quad k = 1 : M\}$. Then, as each new frame of data is received:

1. Resample the particles from the previous frame $\{\boldsymbol{\alpha}_{t-1,k}\}$ according to their weights $\{w_{t-1,k}\}$ to form the resampled set of *un-weighted* particles $\{\hat{\boldsymbol{\alpha}}_{t-1,k} \quad k = 1 : M\}$
2. Predict a new set of particles $\{\boldsymbol{\alpha}_{t,k}\}$ by propagating the resampled set $\{\hat{\boldsymbol{\alpha}}_{t-1,k}\}$ according to the source dynamical model (see appendix A)
3. Transform the raw data into localization measurements through application of the localization function $\boldsymbol{f}$:

$$\theta_t^z = \boldsymbol{f}(\theta, TDOA_t)$$

4. Form the likelihood function (see appendix B):

$$p(\theta_t^z|\theta_{\boldsymbol{\alpha}}) = F(\theta_t^z|\theta_{\boldsymbol{\alpha}})$$

5. Weight the new particles according to the likelihood function:

$$w_{t,k} = p(\theta_t^z|\theta_{t,k})$$

and normalize so that $\sum_k w_{t,k} = 1$

6. Compute the current source location estimate $\hat{\theta}_t$ as the weighted sum of the particle locations $\theta_{\boldsymbol{\alpha}}$:

$$\hat{\theta}_t = \sum_{k=1}^{M} w_{t,k}\theta_{t,k}$$

7. Store the particles and their respective weights $\{\boldsymbol{\alpha}_{t,k}, w_{t,k} \quad k = 1 : M\}$

Table 2.1: Particle filtering algorithm for sound source localization

[7]We note that we will use the terms *particles* and *samples* interchangeably throughout this thesis.

In Table 2.1 each particle $\boldsymbol{\alpha}$ is a four dimensional vector denoting the position and speed in the usual Cartesian coordinates. Thus the kth particle expresses the state on the speaker at time t with $\boldsymbol{\alpha}_{t,k} = [y_{t,k}, \dot{y}_{t,k}, x_{t,k}, \dot{x}_{t,k}]$, for further details we the reader to appendix A. The term $\theta_{t,k}$ denotes the localization parameter corresponding to state. Similar to the Kalman filter the azimuth angle is not tracked directly. Like Kalman filtering the *motion model* equations are computed with the use of the relative x- and y coordinates, however unlike Kalman filtering each coordinate pair is transformed to its corresponding relative position: the azimuth angle. The relative positions in conjunction with the measured azimuth angles θ^z obtained from the localization function $\boldsymbol{f}(\theta|TDOA)$ are used in order to compute the *sensor model* equations.

Resampling A disadvantage of approximating the filtering distribution with samples is the depletion of the sample population [53]. The particles that explore possible movements of the source which have drifted far from the actual movement or the measured movement will have near-zero weights. These particles do not substantially contribute to the overall estimate of the filtering distribution. The process that eliminates the particles with *near-zero-weight* is referred to as *resampling*. Several methods for resampling have been proposed, in general the idea of these methods is to eliminate the near-zero-weight particles and duplicate the particles with high weights, with the requirement that the filtering distribution constructed after resampling is similar to that of the filtering distribution before resampling.

A Resampling Method A method for resampling is discussed here in more detail. First a measure is needed to estimate the number of *near-zero-weight* particles. Lui *et al* [38] refer to coefficient of variation cv_t^2 and is defined as:

$$cv_t^2 = \frac{var(w_{t,k})}{E^2(w_{t,k})} = \frac{1}{M}\sum_{k=1}^{M}(M\, w_k - 1)^2. \tag{2.40}$$

With equation (2.40) the effective sampling size ESS_t can be computed and is defined as:

$$ESS_t = \frac{M}{1 + cv_t^2}. \tag{2.41}$$

If all the particles contribute substantially to the constructed filtering distribution, then this will yield a high value for the ESS. On the other hand if the amount of particles in the sample set with *near-zero-weigh*t increases (resulting in an increase of the variance of the weights) the value for the ESS will drop. The effective sampling size is compared to a threshold, this is usually a percentage of the number of particles M. If the ESS is below that threshold, then the particle set is resampled. The effective sampling size provides an indication for the average substantial contribution of the particles to the constructed filtering distribution.

Select with Replacement With the use of ESS the degeneracy of the particle set can be evaluated. The next step is to avoid a degenerated particle set composed of only few particles with high weights and the remaining particles with very small weights. A resampling method should maintain multiple hypotheses on the position of the source and in the long run keep only those particles that represent a location that is likely given the sequence of measurements. Here a standard resampling method *select with replacement* is briefly discussed. This method resamples the particles according to their weights. A cumulative sum is used to determine the "survival" probability for each particle proportional to its weight. Table 2.2 presents a formal description of the *select with replacement* algorithm.

Input : AN ARRAY WITH THE PARTICLE WEIGHTS $\{w_k,\, k = 1, \ldots, M\}$

Require :

$\sum_{k=1}^{M} w_k = 1$

$Q = \texttt{cumsum}(w)$; COMPUTE THE RUNNING TOTALS $Q_j = \sum_{l=0}^{j} w_l$

$t = \texttt{rand}(M + 1)$; t IS AN ARRAY OF $M + 1$ RANDOM NUMBERS

$T = \texttt{sort}(t)$; SORT THEM ($\mathcal{O}((n \log n))$ TIME)

$T(M + 1) = 1$; $k = 1$; $j = 1$; ARRAYS START AT 1

```
while (k ≤ M) do

    if T[k] < Q[j]
      Index[k] = j;
      k = k + 1;
    else
      j = j + 1;
    end

end
```

Return : `Index`

Table 2.2: Select with replacement algorithm

2.3 Experiments: Tracking A Single Speaker

This section discusses the experiments on tracking a single speaker. Various algorithms for tracking were tested. We start with a brief discussion of the recording conditions. The recording conditions hold for all the evaluated algorithms. The remaining part of this section is organized as follows: first the results obtained from tracking a single speaker with a Kalman filter are discussed. Finally the results obtained from tracking a single speaker with a generic particle filter are discussed.

2.3.1 Recording Conditions

Room Conditions We used a stereo recording obtained with the Philips iCat interface robot, with a frequency sampling rate of 48kHz. The recording was taken in an rectangle-shaped office room [2.8 $\times$ 4.5m] with moderate noise conditions (PC running in the background - approximately one meter from the microphone pair). However, due to the size and the material of the surfaces (solid walls and wooden floor) of the room, we expect that the amount of reverberation will be considerable. The microphones were placed in the right side of the room, 1.5m above the ground level. The distance between the microphones was 0.3m.

Trajectory The trajectory of the speaker describes approximately the first half of an arc. The speaker starts in front of the microphone pair. The initial distance between the speaker and microphone pair is rather small - approximately 0.5m. After a few seconds the speaker starts moving away from the microphone pair, moving to the left side of the room. Then halfway of the recording, the speaker is in the left corner of the room. When the speaker is in the left corner, the azimuth angle between the microphone pair and the speaker corresponds to approximately -90 degrees. The speaker remains stationary once it reached the azimuth location of approximately -90 degrees. In the figures we express the azimuth angles in radians. Note that an azimuth angle of approximately 0.78 radians corresponds to the azimuth angle of 90 degrees. The distance between the microphone pair and the speaker in

the last part of the recording is approximately 3m. Thus the tracking algorithm must first detect and locate the speaker, then keep track of the speaker when it starts moving away from the microphone pair. And finally the tracking algorithm must notice that the speaker stops moving and maintains in position.

Segments For each tracking algorithm that was tested, the recording was cut into segments. The recording was analysed with non-overlapping segments, where a Hamming windowing function was applied to each segment. The Hamming windowing function ensures that the edge effects at the beginning and end of the segment are reduced. We assume a quasi-static location for the speaker within each segment. Each segment corresponds to a 25ms interval, we do not expect that the speaker will not move substantially within such intervals.

Data Rate As mentioned, the recorded signal is sampled with a frequency rate of 48kHz where each sample is represented with 16 bits. Thus each segment potentially contains frequencies up to 24kHz. This approximately corresponds to the frequency spectrum, for which the human hearing mechanism is sensitive to [8]. This relative high frequency sampling rate increases the granularity, in the sense that the original speech signal is approximated with a relative high precision. This precision issue becomes important when the amount of coherence between the recorded signals from the left and the right microphone is measured. In order to distinguish between the various potential sound sources, the original signal must be approximated as closely as possible. Note that with the GCC function the spectral information is removed, leaving only phase information. However the same precision issue arises here since we aim to detect the true sound source by extracting the dominant phase from the signal.

2.3.2 Kalman Filter Based Tracking

As stated in section 2.2.1 the Kalman filter is typically unable to deal with non-linear state space and non-Gaussian noise. Note that the transformation from TDOA to azimuth, and the transformation from azimuth to relative Cartesian coordinates is non-linear. In addition, the measurements obtained from the GCC method frequently exhibit a multi-modal distribution, due to the presence of a strong multipath. Due to these properties it will be a challenging task to track a speaker with a Kalman filter.

Parameters We choose the evolution noise as $Q = 0$ and the measurement noise as $R = 0.5$. We assume that, if the speaker is walking, it is at an approximately constant (slow) pace. In addition we assume that the x- and y coordinates are independent of each other. Note that our aim is to track the ratio between the x- and y coordinate. For each coordinate we choose the evolution matrix A, matrix B from the state evolution equation (2.28), measurement matrix H and initial state covariance matrix P as follows:

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} .5 & .1 \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad P = \begin{pmatrix} 10 & 0 \\ 0 & 1 \end{pmatrix}.$$

Results Figure 2.2 presents the result of tracking a speaker with a Kalman filter. The Kalman filter was initiated with the above described parameter setting. The measured azimuth angles are indicated with dots, the expected trajectory is indicated with the bold line.

[8] Typically the perceived frequency spectrum by the human ears is approximated with $20 - 20000$Hz.

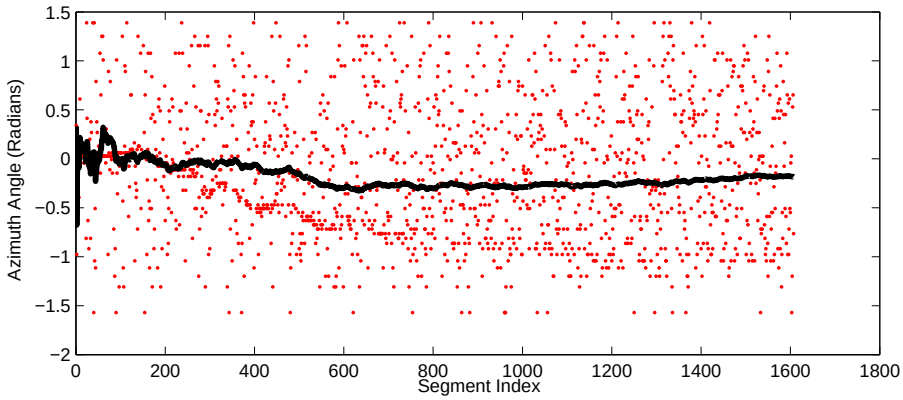

Figure 2.2: Tracking of a single person with a Kalman filter-based tracking algorithm. The measured azimuth angles are overlaid with the estimated expected location of the speaker.

The Kalman filter was initialised with the "true" initial position and velocity, that is:

$$\hat{x}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad \hat{y}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

This corresponds to the speaker at azimuth angle $\theta = 0$ radians. Despite this proper initialisation the Kalman filter has difficulty keeping track of the speaker and eventually the Kalman filter fails in tracking the speaker. When the speaker starts moving from the microphone pair, the measurements become less reliable. Whenever the distance increases between the speaker and the microphone pair, the less dominant the "true" sound source becomes compared to the noise and reverberation components in the recorded signal. As a consequence, the measured azimuth angles are distributed more uniformly across the potential range of azimuth angles. Note that when the speaker is close to the microphone pair, the measured azimuth angles are more concentrated in the area that corresponds to the true azimuth angle. This concentration in measurements is gradually lost when the speaker moves away from the microphone pair (see also Fig. 2.9). With this gradual change in distribution, the Kalman filter gradually loses the target. As stated earlier the Kalman filter assumes that the state space is non-linear and non-Gaussian. Therefore it can only provide a linear minimum variance estimate on the speakers location. This however, is not sufficient to accurately track a speaker with the use of a non-linear motion model in the presence of a strong multipath[9].

Additional Experiments Since we were not satisfied with the tracking result, we conducted additional tests with various parameter configurations. The Kalman filter was initiated with various values for measurement noise R, the initial state covariance matrix P and matrix B from the state evolution equation (2.28). However, none of the various configurations succeeded in tracking the speaker accurately. Figure 2.3 presents the various results obtained from the Kalman filter with various parameter configurations. The top panel presents the result where the measurement noise was chosen as $R = 1$ and the matrices P and B as follows:

$$P = \begin{pmatrix} 10 & 0 \\ 0 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} .05 & .1 \end{pmatrix}.$$

[9]Note that for the motion model we transform the TDOA to an azimuth angle. The azimuth angle is transformed to relative x- and y coordinates. These transforms are non-linear

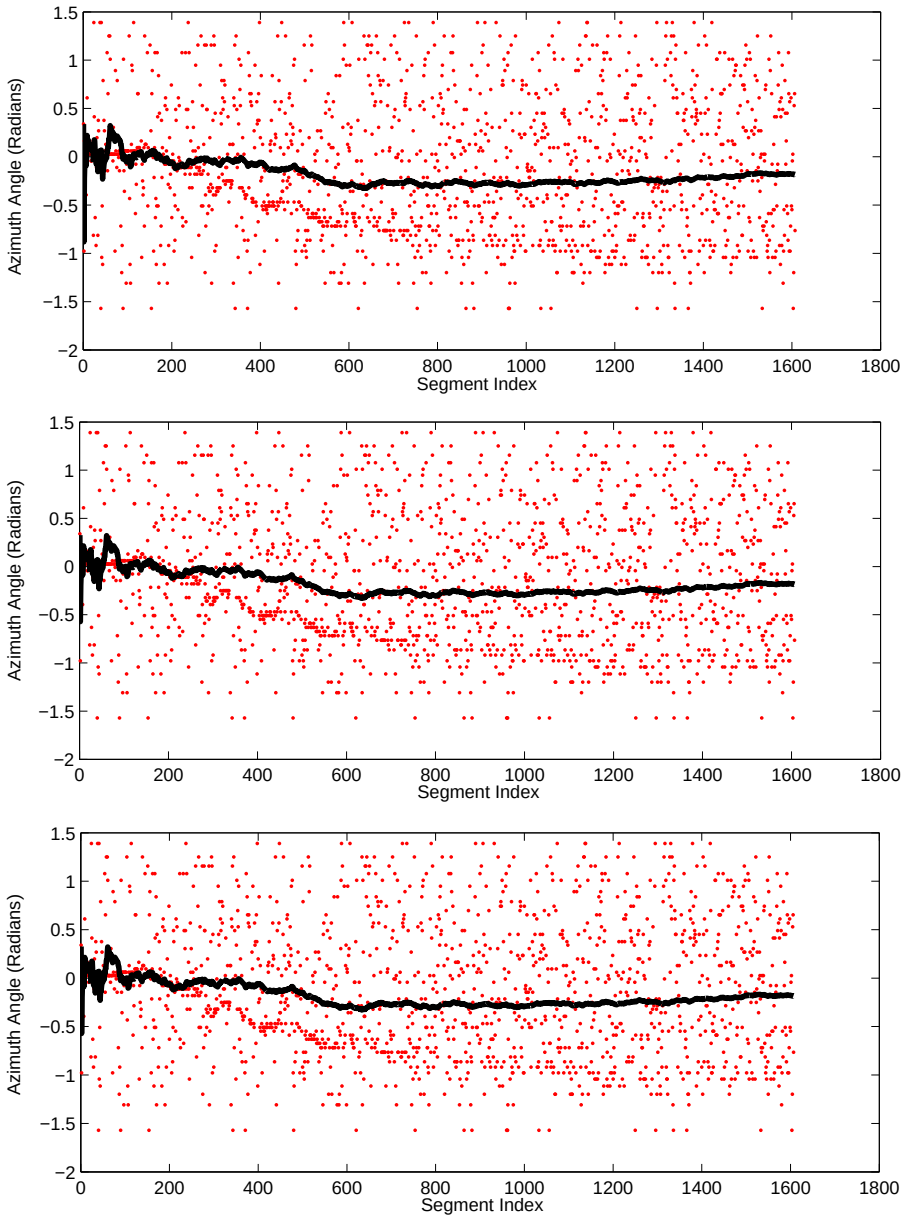

Figure 2.3: Various results obtained from the Kalman filter-based tracking algorithm where the filter was initiated with various parameter configurations (see paragraph *Additional Experiments* in section 2.3.2 for details). The measured azimuth angles are overlaid with the estimated expected location of the speaker.

The middle panel presents the result where the measurement noise was chosen as $R = 10$ and the matrices P and B as follows:

$$P = \begin{pmatrix} 100 & 0 \\ 0 & 10 \end{pmatrix}, \qquad B = \begin{pmatrix} .5 & .1 \end{pmatrix}.$$

The bottom panel presents the result where the measurement noise was chosen as $R = 10$ and the matrices P and B as follows:

$$P = \begin{pmatrix} 100 & 0 \\ 0 & 10 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 1 \end{pmatrix}.$$

For all the above described parameter configurations we chose the process noise Q as follows $Q = 0$.

2.3.3 Particle Filter Based Tracking

Parameters We set the model parameters as follows: The sensor variance for azimuth angle measurements was chosen as $\sigma^2 = 0.1$. The number of potential azimuth locations for the likelihood function was chosen as $K = 5$. The parameters for the motion model (the Langevin) process were $v_x = 0.5ms^{-1}$, and $\beta_x = 10s^{-1}$. These values correspond to a human (slowly) walking in a room. For the particle filter we used $M = 1000$ samples. Since each sample is a four-dimensional vector, running the particle filter with 1000 samples becomes a rather time consuming process. Thus the state-space is four-dimensional, therefore increasing the number of samples will introduce a substantial computational load.

Results In contrast to the Kalman filter based tracking algorithm, the particle filter based tracking algorithm is not initialised with the true azimuth location. The initial particle set is uniformly distributed over the potential range of azimuth angles $[-90\,,\,90]$ degrees. Figure 2.4 presents the result obtained from the particle filter based tracking algorithm.

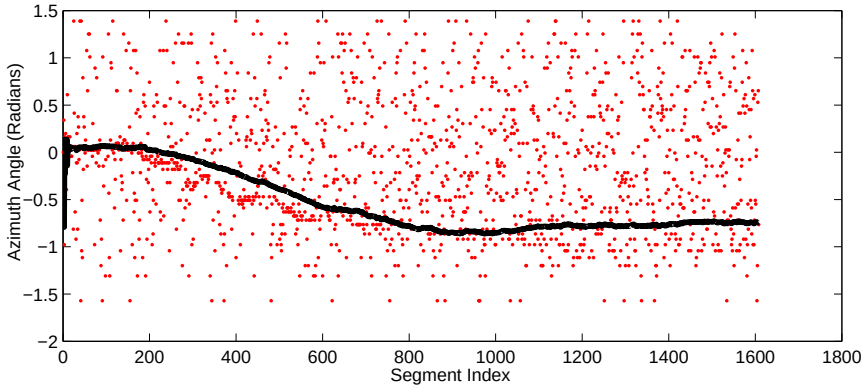

Figure 2.4: Tracking of a single person with a generic particle filter.

The measured azimuth angles are represented with the dots, the expected trajectory is represented by the bold line. Note that here, only the azimuth angle is plotted corresponding to the lag that maximizes the GCC function. Although this result is much better compared to the result obtained from the Kalman based tracking algorithm. The tracker is still unable to track the speaker accurately. Consider Fig. 2.5, the three presented results were obtained by running the algorithm with the same configuration as described above.

By choosing the number of samples large enough, the particle filters‘ solution will closely approximate the posterior distribution. We have to choose this number on experimental basis. However, the larger the number of samples, the slower the algorithm will become. Therefore, real-time implementation is compromised. Furthermore, choosing the largest number

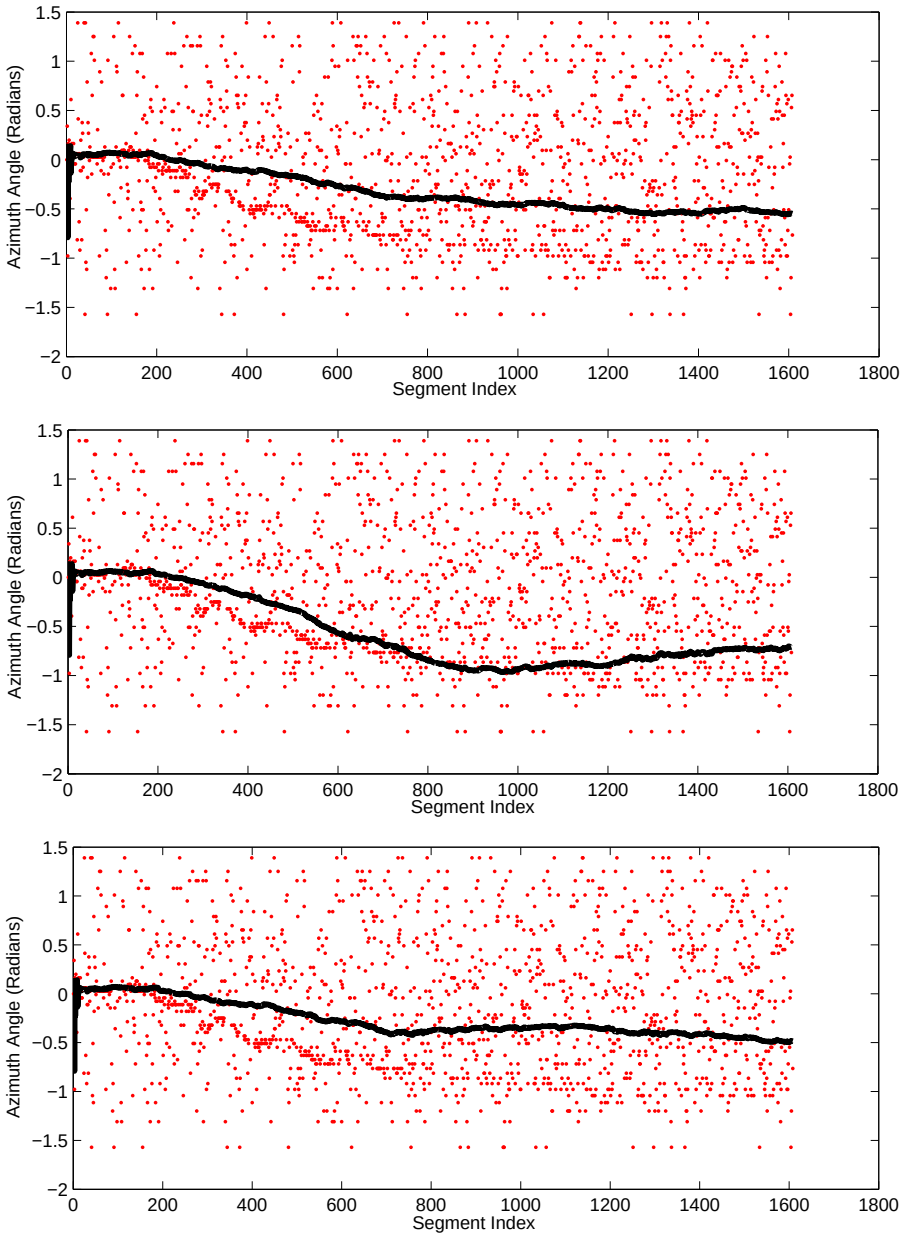

Figure 2.5: Results obtained from various runs with the particle filter. The measured azimuth angles are overlaid with the estimated expected location of the speaker.

possible [10] for the sample population, will still not guarantee that each run will converge to (approximately) the same solution. Therefore we extend the current algorithm such that the accurateness and the speed of convergences to an approximately constant solution is increased.

[10]That is, the largest number possible in real-time implementation.

2.4 Tracking Algorithm Extensions

This section discusses two extensions for the tracking algorithm. The first extension was applied to both the Kalman- and particle filter based tracking algorithm. Due to design issues the second extension was applied only to the particle filter based tracking algorithm.

2.4.1 Extension 1: Low-Pass Filter

As stated in section the inverse map from TDOA to azimuth angle exhibits ill-posed behaviour. No straightforward methods exists to suppress ill-posed behaviour. However in practice, aiming for suppression one typically includes all available a priori knowledge in the algorithm. Since our aim is to track the speaker, an obvious choice would be applying a Low-Pass (LP) filter. Typically, human speech utterances have a potential range in the frequency spectrum up to 8kHz. Thus human speech utterances do not substantially contribute to the energy in the frequency spectrum (of the recorded signal) above 8kHz. Therefore noise artefacts become more dominant in the higher regions of the signals frequency spectrum. Various potential sound sources "present" in the signal will lead to multi-modal sensor distribution. Therefore we want to exclude all the frequencies above 8kHz in the GCC function. We assume that by suppressing the regions from where potential undesired dominant sources may originate, we can suppress the ill-posed nature of the inverse map. Figure 2.6 shows the scheme of the tracking algorithm extended with a LP filter.

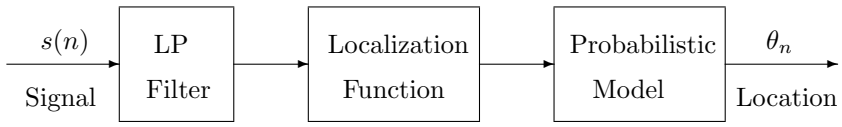

Figure 2.6: Schematic representation of extended tracking algorithm #1

Both the Kalman- and particle filter based tracking algorithms are applied to the LP filtered signal. The results are presented respectively in Fig. 2.7 and Fig. 2.8. From Fig.

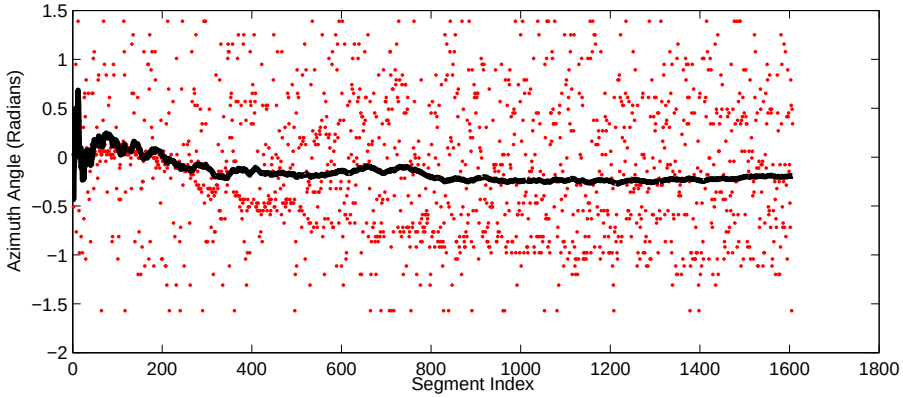

Figure 2.7: Tracking of a single person with a Kalman filter in conjunction with a LP filter.

2.7 we see that the Kalman filter is still not able to track the speaker accurately. From Fig. 2.8 we see that the particle filer in conjunction with the LP filter is able to track

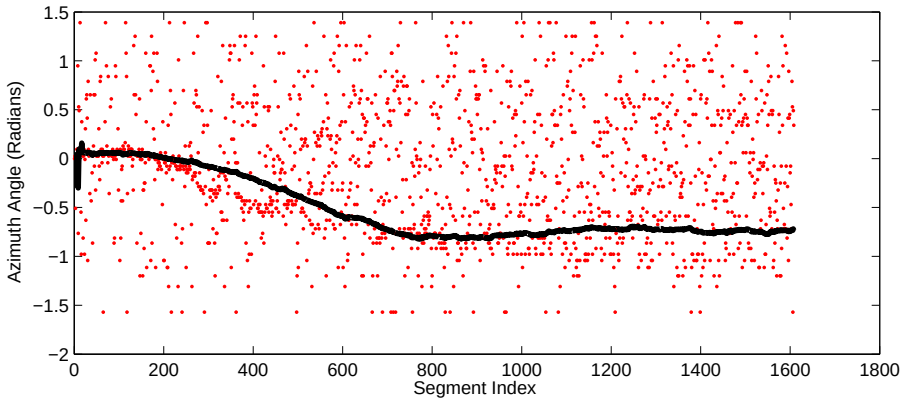

Figure 2.8: Tracking of a single person with a particle filter in conjunction with a LP filter.

the speaker accurately. By pre-processing the raw data for the localization function, the localization function is able to provide more accurate information about the speakers location to the probabilistic model that is used for inference. Figure 2.9 presents a comparison of the measured azimuth angles obtained from the localization function with- against without pre-processing the signal with a LP-filter.

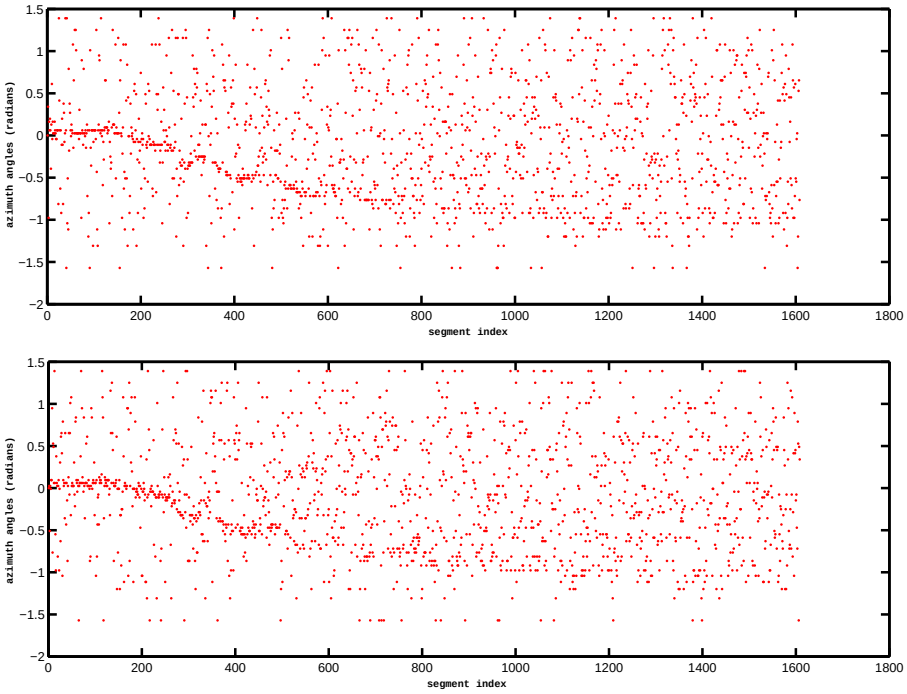

Figure 2.9: Top panel: the measured azimuth angles with the signal containing frequency components up to 24kHz. Bottom panel: the measured azimuth angles with the signal containing frequency components up to 8kHz.

Note that there is a visible improvement by pre-processing in the first part of the signal. We assume that pre-processing with a LP-filter is particularly effective when the speaker is close to the microphone pair. Therefore we assume we can enhance the signal by applying the LP filter for the localization function, as long as the speaker is dominant over the noise artefacts. Figure 2.10 presents a comparison of the TDOA estimates obtained from the GCC function with- against without applying the LP-filter. The top panel presents the TDOA estimates extracted from analysed segment at index $n = 52$ with the GCC method. The bottom panel presents the TDOA estimates extracted from the same segment, obtained by the GCC function in conjunction with a LP-filter. In Fig. 2.10 the horizontal axis represent the range of lags that are used for the TDOA estimation. The vertical axis shows for each lag the corresponding coherence energy obtained from the GCC function. The true TDOA corresponds to a lag of approximately 0ms.

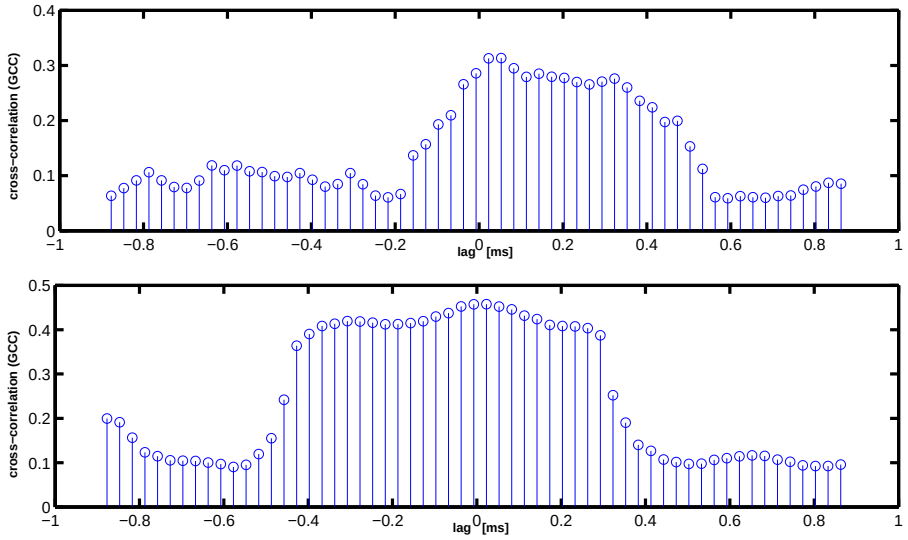

Figure 2.10: Comparison of TDOA measurements obtained from the GCC function in conjunction with- (bottom panel) against without (top panel) an additional LP-filter.

From Fig. 2.10 we see that by pre-processing the input data, the GCC function provides us with more accurate TDOA estimates. Note that, applying a LP-filter with the cut-off frequency set to 8kHz to the recorded signal, where the original signal was approximated with a frequency sampling rate of 48kHz, does not yield the same information on the original signal, if we recorded the original signal with a frequency sampling rate of 16kHz.

With respect to the precision issue, sampling with a high frequency rate, approximates the original signal more closely compared to sampling with a lower frequency sampling rate. Therefore, a larger number of samples per segment, allows for a time shift analysis on a finer scale. Note that, in the experiment the GCC analyses the segment by computing the cross-correlation for values in the range of $[-0.88\,,\,0.88]$ ms.

Figure 2.11 presents a comparison of the GCC analysis on the 25ms segment at index $n = 61$, with- against without down-sampling. The top panel presents the GCC analysis on the segment where the recorded signal was down sampled to 16kHz. The bottom panel presents the GCC analysis on the same segment where the recorded signal was pre-processed with a

LP-filter. Both segments were analysed with discrete time shift steps of 0.02ms. Figure 2.12 presents a similar comparison (LP-filtering against down-sampling the signal) of the GCC analysis on the same 25ms segment ($n = 61$).

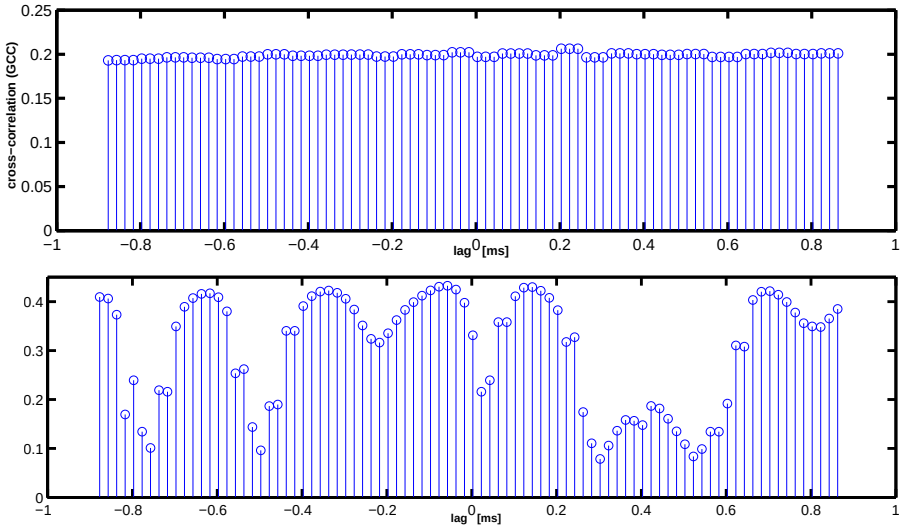

Figure 2.11: Comparison of TDOA measurements obtained from the GCC function in conjunction with a LP-filter (bottom panel) against measurements from the GCC function with the recorded signal down sampled to 16kHz (top panel). Both segments were analysed with discrete time shift steps of 0.02ms.

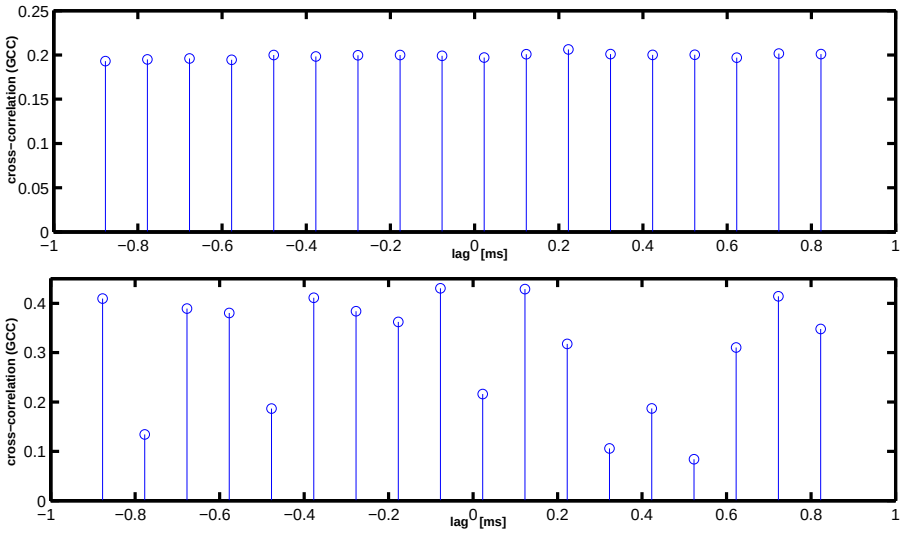

Figure 2.12: Comparison of TDOA measurements obtained from the GCC function in conjunction with a LP-filter (bottom panel) against measurements from the GCC function with the recorded signal down sampled to 16kHz (top panel). Both segments were analysed with discrete time shift steps of 0.1ms.

However in Fig. 2.12 the segment was analysed with a larger discrete time shift of 0.1ms. In Fig. 2.11 and Fig. 2.12 the horizontal axis represent the range of lags that are used for the TDOA estimation. The vertical axis shows for each lag the corresponding coherence energy obtained from the GCC function.

Note that since the "true" TDOA is approximately 0ms, the best result was obtained by the GCC method in conjunction with a LP-filter, where the 25ms segment was analysed with a high(er) granularity scale. Thus, in order to approximate the "true" TDOA as closely as possible with the lag that maximizes the GCC function, one should choose an appropriate frequency sampling rate for approximating the original signal and an appropriate granularity scale for the GCC analysis.

2.4.2 Extension 2: Threshold- And Adaptive Sigma Function

The GCC will typically fail to detect the true TDOA whenever reverberation and noise artefacts become dominant over the true sound source. We assume that whenever the true sound source is active, it is dominant over the undesired sound sources. Thus we assume that, if the analysed segment contains an active speaker, then this will increase the SNR. Note that here we assume that the noise level remains approximately constant. In addition we assume that the volume level of the speech remains approximately constant. With these assumptions, the energy in the analysed segment typically indicates whether the speaker is active or not. Thus, we consider the analysed segments with high energy levels as more reliable, due to an increase in SNR. Therefore, excluding the segments that contain relative low energies, will reduce the amount of false TDOA estimates. In order to exclude the low-level energy segments, we employ a threshold function and a threshold measure. Therefore we compute the energy E for each segment and compare it to the threshold value. For segment s at index n we compute the E_n as follows:

$$E_n = \frac{\sum_{g=1}^{N} |\mathcal{F}\{s_n(g)\}|}{2}, \tag{2.42}$$

where $\mathcal{F}\{.\}$ denotes the Fourier transform and where N denotes the number of samples in segment s.

The value for the energy threshold should be chosen such that: the segments where the true sound source is not the dominant sound source are excluded. On the other hand the energy threshold should pass those segments where the true sound source is the dominant sound source. Note that when the distance increases between the speaker and the microphone pair, the amount of energy in the segment that originates from the speaker decreases. Therefore, a dynamic threshold would be ideal. However, we do not know the distance [11]. Therefore an appropriate threshold value should also allow those segments for analysis where, despite the relative large distance between the speaker and the microphone pair, the speaker is the dominant sound source. The measure for the threshold was determined experimentally, the best value was found for $energyThreshold = 80$. Thus the segments, containing energies levels below 80 are discarded.

Note that the likelihood function for the particle filter (see appendix B) is designed to take notice that occasionally the true sound source may be silent. During the experiments we

[11] With the assumption that the speaker remains frequently active throughout the recorded signal. Then a possible dynamic threshold function could be implemented based on time-averaging information.

noticed that, aside from the occasional "silent" segments, the segments that contained the beginning or an ending of a speech utterance gave rise to false TDOA estimates. These inhaling and exhaling alike sounds of speech sounds are typically not well articulated. Therefore we assume that these inhaling and exhaling alike sounds radiate more divergent from the mouth, with the "direct sounds" striking the microphone pair in a less concentrated fashion. Due to this property we assume that, for such segments, the GCC method is frequently unable to detect the speaker as the dominant sound source. Typically the energy levels for these segments are low, therefore a substantial amount of these segments are discarded by the threshold function. Figure 2.13 presents the result of the extended tracking algorithm.

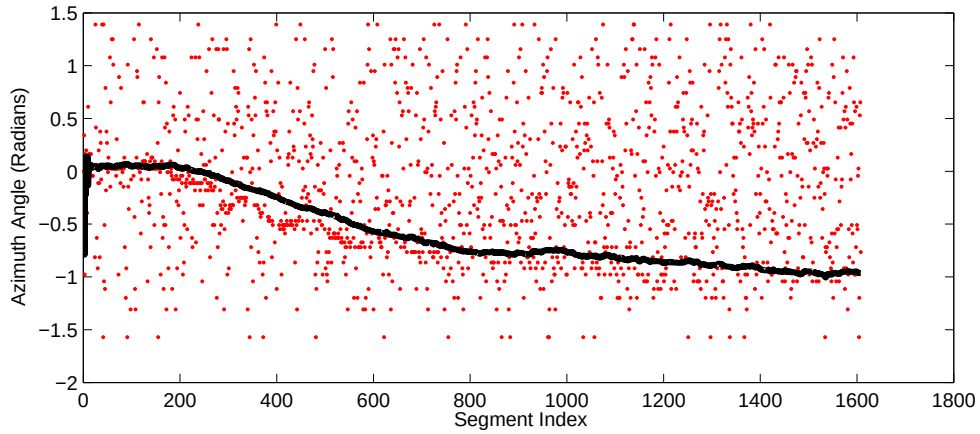

Figure 2.13: Tracking of a single person with a particle filter in conjunction with an additional threshold function with $energyThreshold = 80$. The measured azimuth angles are overlaid with the estimated expected location of the speaker.

From Fig. 2.13 we see an improved tracking result compared to the result presented in Fig. 2.5. We note that we have tested the extended algorithm various times, the result presented in Fig. 2.13 indicates the typical result of these experiments. However, we see that the speaker is not accurately tracked in the middle part of the recording, corresponding to approximately the segment sequence $[400, 1000]$. Furthermore in the latter part of Fig. 2.13 there is a slight overshoot in the expected location. The temporary loss of the speaker, could partially be explained by the temporary loss of actual measurement information due to the threshold function. For each segment that did not reach the energy threshold, no measurement information was incorporated in order to compute the filtering distribution.

Since we do not use the measurement information to compute the filtering distribution for the "discarded" segments, we update the particles propagated from the motion model with uniform weights. Thus effectively, only the prior distribution is used to approximate the posterior distribution for such segments. To compensate for this temporary loss of information, we propose a modified version of the current likelihood function. This modified likelihood function should effectively incorporate all available knowledge. By including additional information, a more accurate particle weight update is accomplished. Hence we increase the accuracy of the estimated location and the quality of the predictive distribution. Thus the particle population will be generated from a more accurate predictive distribution. We assume that this property enables the particle population to keep a more accurate track of the speaker, during the occasional loss of measurement information. To incorporate effectively

all the available information in the likelihood function we introduce the *adaptive sigma*.

The Adaptive Sigma As mentioned section 2.1.2 from the GCC method we obtain for each TDOA candidate a value that indicates the amount of coherence between the left- and right signal. Thus for each TDOA candidate we obtain a coherence energy unit. In order to incorporate additional information in the likelihood function extracted from the GCC method, we make two additional assumptions.

Assumption 1 The manifestation of a dominant true sound source in the GCC function, typically results in a region of more concentrated coherence energy with a sharpened peak, that corresponds to the true sound source.

Assumption 2 An active true sound source, typically causes an increase in the coherence energy values obtained from the GCC function.

From the first assumption we can derive that, an active speaker increases the variance in the measurements obtained from the GCC function. From the second- in conjunction with the first assumption, we can derive that an active speaker, typically causes a relative and absolute increase in the differences of measured coherence energy. Since the window length is 25ms of analysis segments, an active speaker will typically cause a change in observed coherence energy patterns, consistent with the above stated assumptions for a consecutive sequence of segments. Based on these assumptions along with the resampling property of the particle filter, we modify the likelihood function of the particle filter in order to increase the accuracy of the tracking algorithm. We consider the more sharpened peaks, as more likely to be caused by the true sound source. Therefore, we want to enforce a more dense particle population in the area that corresponds to these sharpened peaks. We can accomplish this by effectively controlling the value for the ESS (see paragraph *A Resampling Method* in section 2.2.4).

Whenever the value for the ESS drops below the resampling threshold value, the particles will be enforced into the high likelihood areas. Note that simply setting the resampling threshold to a low value will not increase the robustness of the tracking algorithm. Setting the resampling threshold too low, will increase the probability that the particles are enforced into regions that do not correspond to the true location. We can effectively decrease the ESS value whenever the GCC analysis on a segment is consistent with the above assumptions, by increasing the variance in the particle weights values. Therefore we relate the variance in the measurements obtained from the GCC function, to the variance used in the likelihood function to evaluate the potential source locations. In practice, we relate the coherence energy values corresponding to the K potential TDOA candidates to the variance used to evaluate the corresponding K potential source locations in the likelihood function. Figure 2.14 presents a schematic representation of particle filter based tracking algorithm extended with a threshold function in conjunction with the adaptive sigma algorithm.

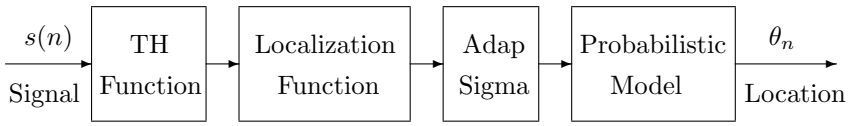

Figure 2.14: Schematic representation of extended tracking algorithm #2

Table 2.3 presents a formal description of the adaptive sigma algorithm that is incorporated in the extended particle filter-based tracking algorithm.

COMPUTE ADAPTIVE SIGMA ACCORDING TO COHERENCE ENERGY:

`for` $k = 1 : K$

$$\sigma_{adap}(k) = \frac{1}{ce(k)^{P_f}}$$

`end`

NORMALIZE & SCALE

$$T = \sum_{k=1}^{K} \sigma_{adap}(k)$$

`for` $k = 1 : K$

$$\sigma_{adap}(k) = \frac{\sigma_{adap}(k)}{T}$$

$$\sigma_{adap}(k) = \frac{\sigma_{adap}(k) \cdot K}{C}$$

`end`

Table 2.3: Adaptive sigma algorithm

In table 2.3 $\sigma_{adap}(k)$ denotes the adaptive sigma for potential source location θ_k, $ce(k)$ denotes the coherence energy for TDOA estimate k that corresponds to the source location θ_k. The variable P_f denotes the power factor, allowing for an increase or decrease in the variance in the K coherence energy values. The variable C denotes the scaling factor, in order to control the range of the variance values. Thus for each analysed segment, the variance σ adapts to the changes in differences in the coherence energy obtained from the GCC function. Thus, each of the K potential source locations is evaluated with an individual variance, that reflects the degree of certainty for each location estimate based on information obtained from the GCC function.

Parameters The extended algorithm was tested with the following parameter setting: the number of potential TDOA candidates was chosen as $K = 5$, the power factor was chosen as $P_f = 2$. The scaling factor was chosen as $C = 10$, this value is based on the sensor variance value $\sigma = 0.1$ that was used in the previous experiments. Note that if the coherence energies for all the K potential TDOA candidates are approximately equal then this will yield:

$$\sigma_{adap}(k) = \frac{0.2 \cdot 5}{10} = 0.1,$$

for each of the K potential source locations. All other involved parameters are set to the same values as used in the previous experiments (see section 2.3.3 for details). Figure 2.15 presents the result of the extended tracking algorithm (with the above described parameter setting).

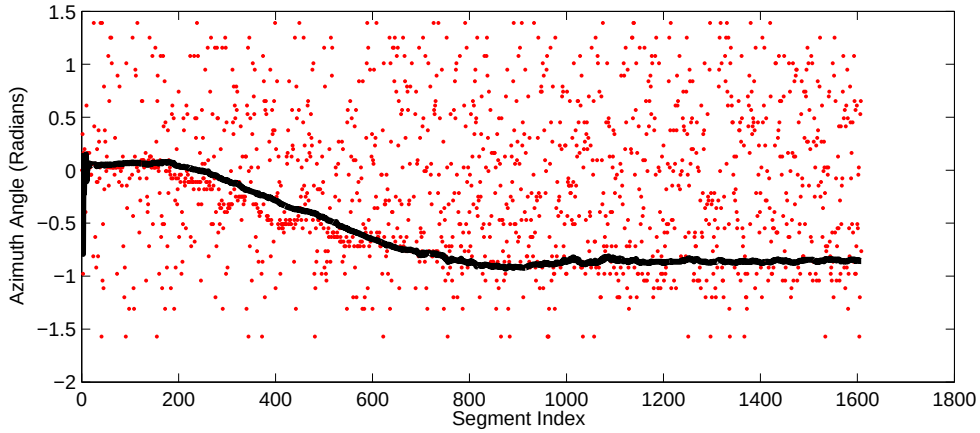

Figure 2.15: Tracking of a single person with an additional threshold- and adaptive sigma function (with $energyThreshold = 80$ and $P_f = 2$). The measured azimuth angles are overlaid with the estimated expected location of the speaker.

Note that the improvement is clearly visible in segment sequence $[400, 1000]$. Furthermore there is no overshoot in the expected location in the final part of Fig. 2.15. Thus the extended algorithm is able to track the speaker accurately, despite the occasional loss of measurement information.

The value for the power factor P_f plays an important role, the best results were obtained with $P_f = 2$. We cannot provide any analytical explanation for this value, however we assume that overemphasising the differences in coherence energy will eventually lead to a (undesired) Gaussian approximation of the sensor distribution for a substantial part of the segments. Note that occasionally it might happen that the true sound source does not correspond to the global maximum peak in the measurements obtained from the GCC function. In these cases we want to maintain a *mixture-of-Gaussians* to approximate the sensor distribution. On the other hand, if the differences in the coherence energy are not emphasised enough, then the effect of the adaptive sigma function that initially was aimed for will be lost. Figure 2.16 illustrates the differences in emphases by running the algorithm with different values for the power factor P_f.

The number of potential locations K should be chosen appropriate. Ideally, this number should be such that we obtain a consistent inclusion of the true source location in the obtained potential locations, whenever the true sound source is "present" in the segment. In addition, with respect to the adaptive sigma algorithm, the value for K should be chosen large enough to effectively reflect the difference between the maximum in the GCC measurements and the remaining $K-1$ TDOA candidates. Note that with respect to the normalization step that is part of the adaptive sigma algorithm, the value for K plays an important role. On the other hand, choosing K too large, will include too many undesired locations and thereby compromising the robustness of the tracking algorithm. Furthermore, setting the value for K too high, will unnecessary increase the computational load.

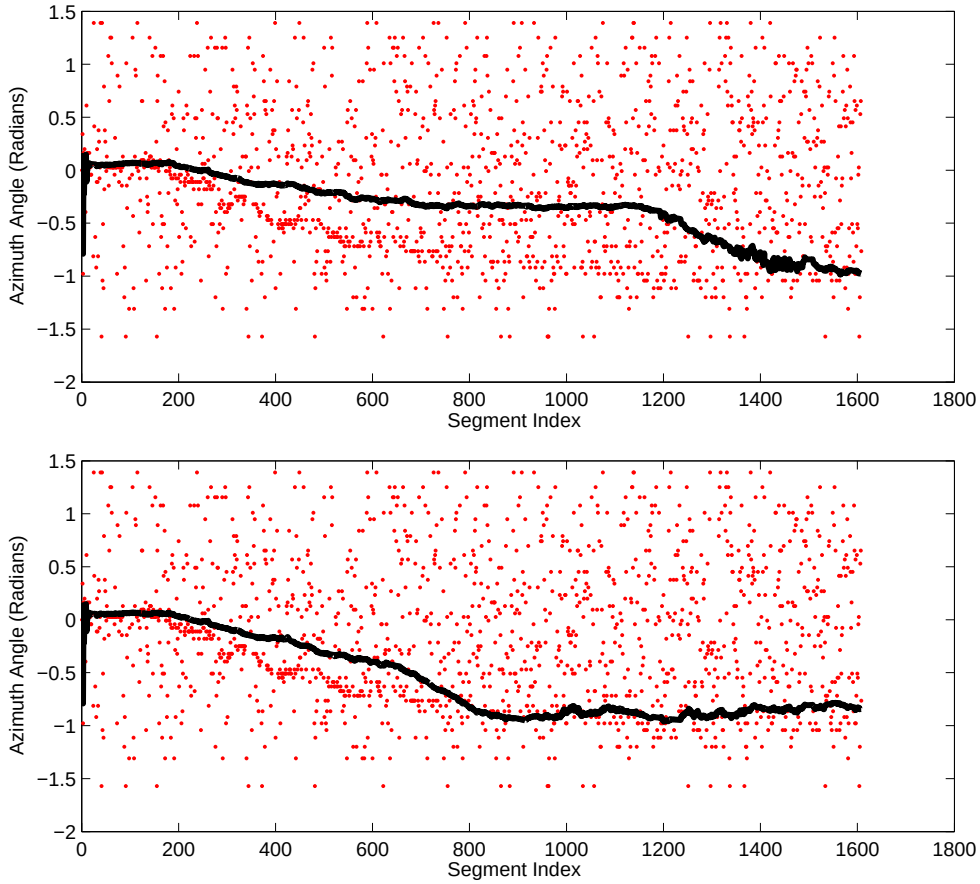

Figure 2.16: Results obtained from the particle filter with an additional threshold-($energyThreshold = 80$) and adaptive sigma function with various parameter settings for the power factor Pf (top panel: $P_f = 1.5$ and bottom panel: $P_f = 3$). The measured azimuth angles are overlaid with the estimated expected location of the speaker.

The best value: $K = 5$ was found experimentally, however Fig. 2.17, Fig. 2.18 and Fig. 2.19 aim to provide some analytical explanation. In Fig. 2.17, Fig. 2.18 and Fig. 2.19 the horizontal axis represent the range of lags that are used for the TDOA estimation. The vertical axis shows for each lag the corresponding coherence energy obtained from the GCC function. In addition, the presented figures aim to illustrate the differences in coherence energy distribution, since this property is at the core of the two assumptions and thus at the core of the adaptive sigma algorithm.

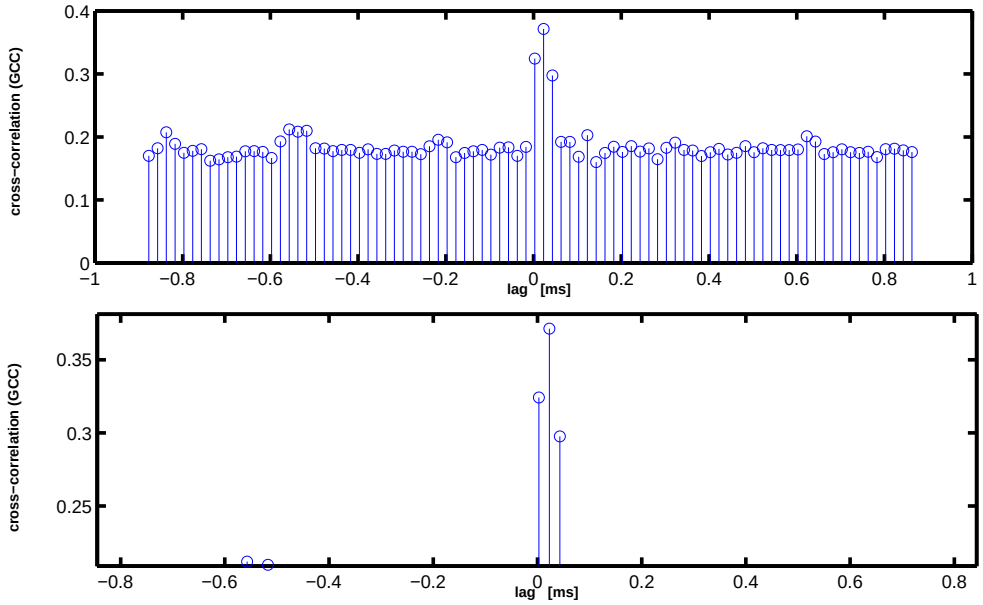

Figure 2.17: The top panel presents the results obtained from the GCC function where the analysed segment contained a clear speech utterance. The bottom panel is a blow-up of the results containing the K potential TDOA candidates that maximize the GCC function.

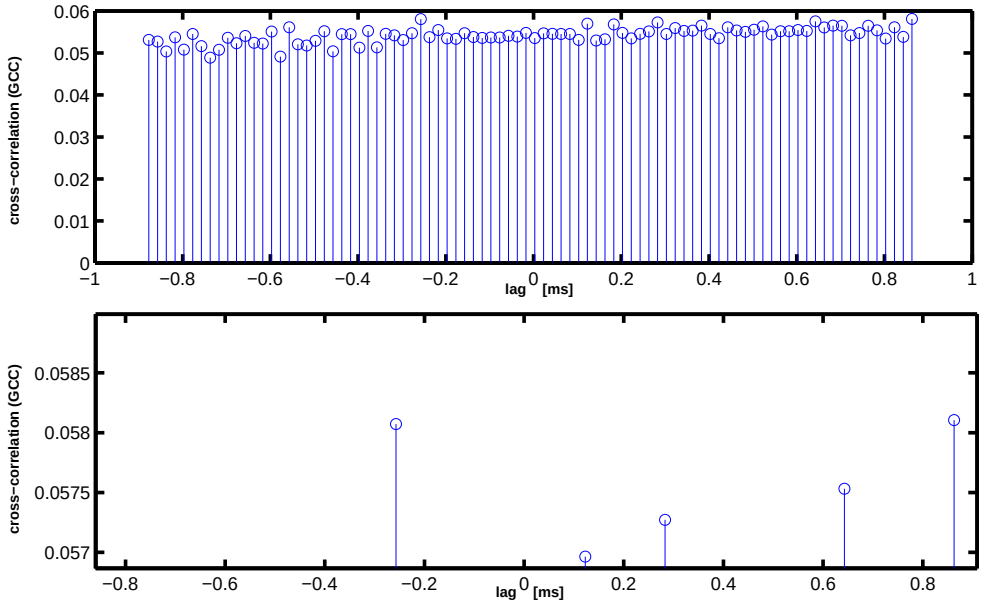

Figure 2.18: The top panel presents the results obtained from the GCC function where the analysed segment contained mainly noise. The bottom panel is a blow-up of the results containing the K potential TDOA candidates that maximize the GCC function.

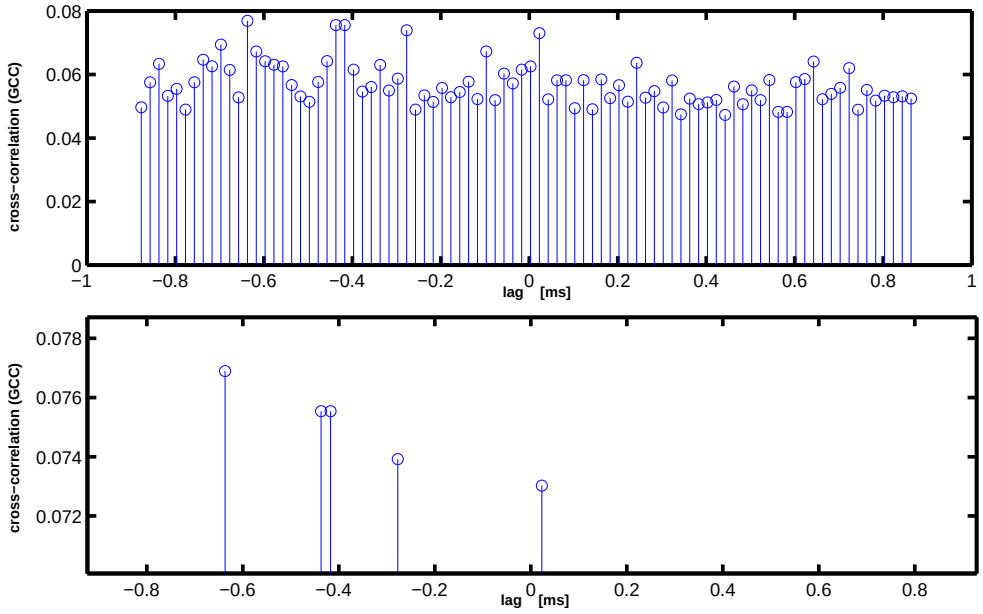

Figure 2.19: The top panel presents the results obtained from the GCC function where the analysed segment contained a very weak speech utterance. The bottom panel is a blow-up of the results containing the K potential TDOA candidates that maximize the GCC function.

We have presented Fig. 2.17, Fig. 2.18 and Fig. 2.19 where the true TDOA approximately corresponds to 0 ms. We see from Fig. 2.17, that $K = 3$ would be the appropriate choice [12], however this choice is not always appropriate. Consider Fig. 2.18 that presents the GCC analysis on a "noisy" segment. By choosing $K = 5$, the true TDOA is included and thus the corresponding true source location is included in set of potential locations. We see from Fig. 2.18 that the differences in coherence energy are rather small. Therefore, the differences in values for the variances used for evaluating the potential locations by the likelihood function will be small. According to table 2.3 in conjunction with the above described parameter setting, each adaptive sigma will approximately be $\sigma_{adap} = 0.1$. Thus, by choosing $K = 5$, the stochastic drift is suppressed for a (potential) particle population towards false azimuth locations[13].

A similar analysis holds for the analysed segment presented in Fig. 2.19. Note that the differences in coherence energy distribution play an important role in the adaptive sigma algorithm. For the segments where the true sound source is not the dominant source, typically the K coherence energy values obtained from the GCC function that correspond to the K potential locations can be approximated with a uniform distribution. Whereas the segments where the true sound source is the dominant source, typically exhibit a Gaussian distribution on the coherence energy values. Occasionally it might happen that the most sharpened peak in the GCC measurements corresponds to a "false" source location. However the effect of these scenarios will be negligible due to their spurious character in conjunction with a dense particle population in the "true" source location.

[12] Actually, for this segment $K = 1$ would be the most appropriate choice.
[13] For such TDOA measurements as presented in Fig. 2.18.

2.5 Additional Trajectories Experiments

This section presents additional experiments with alternative trajectories. All the experiments were conducted with the same recording conditions as presented in section 2.3.1 [14] All the experiments were conducted with a particle filter based tracking algorithm extended with an additional LP filter in conjunction with the adaptive sigma function. However no threshold function was used.

Thus, the parameters for the sensor variance were set according to the adaptive sigma algorithm. Since these parameter values are similar to the (best found) values in the previous experiments we refer to paragraph *Parameters* in section 2.4.2 for details. The parameters for the motion model (the Langevin[15]) process were $v_x = 0.1ms^{-1}$, and $\beta_x = 10s^{-1}$. For the particle filter we used $M = 1000$ samples

First we present a speaker tracking experiment where the speaker is located in one of the outer corners (≈ -0.9 radians) of the room and remains stationary. The distance between the microphone pair and the speaker is approximately 3m. Thus in this experiment the task of the tracking algorithm is to locate the sound source and maintain this location estimate throughout the signal without picking up a clutter trail. The result of this experiment is presented in Fig.2.20.

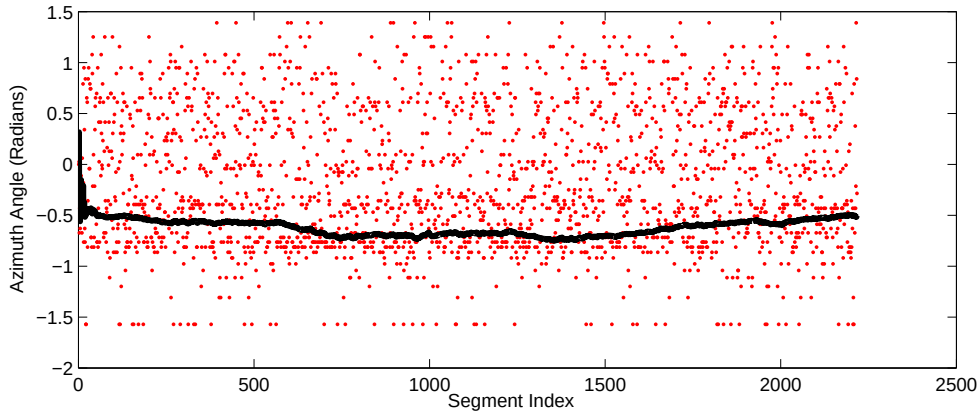

Figure 2.20: Tracking of a stationary single speaker with a particle filter in conjunction with an additional LP-filter and adaptive sigma function. The speaker remained in the most outer corner of the room (≈ -0.9 radians) throughout the recording. The measured azimuth angles are overlaid with the estimated expected location of the speaker.

From Fig.2.20 we see that the tracker has an accurate location estimate for the most part of the signal. From the latter part of Fig.2.20 we see that the tracker has difficulty in maintaining a fixed expected location for the speaker. We believe that is partly due to the change in measurement distribution, notice the less dense measurement concentration on the true azimuth angle in the final part of Fig.2.20. Note that the initial offshoot is caused by the initial distribution of the particle population. The particles are uniformly distributed across the potential azimuth range ($[-\pi/2, \ldots, \pi/2]$).

[14] Obvious the trajectory description does not hold for the additional experiments.
[15] For details see appendix A.

Next we present experiments were the speaker moves from the left side (≈ 0.78 radians) to the right side (≈ -0.78 radians) of the room with an approximate constant pace. We present two results Fig.2.5 and Fig.2.5 obtained from two different recordings.

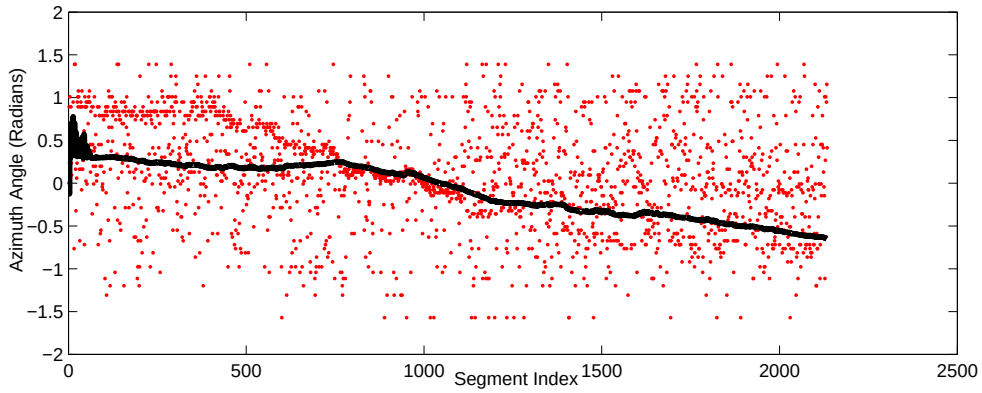

Figure 2.21: Result for the first recording (*Recording 1*) obtained from the particle filter with an additional LP-filter in conjunction with the adaptive sigma function. The measured azimuth angles are overlaid with the estimated expected location of the speaker.

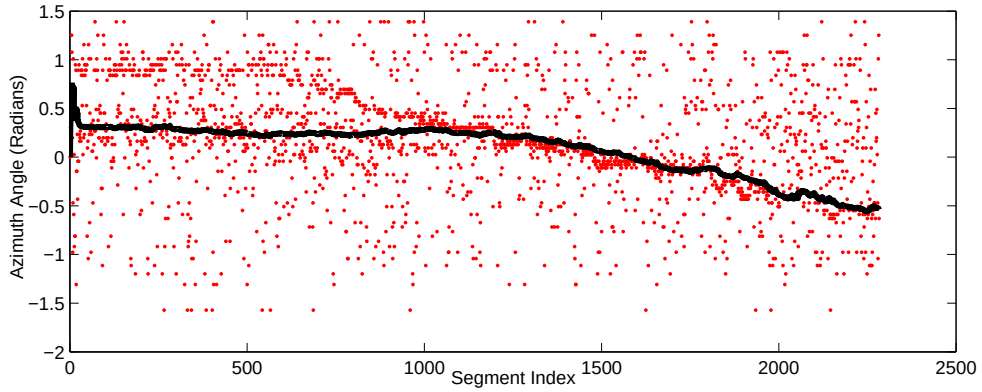

Figure 2.22: Result for the second recording (*Recording 2*) obtained from the particle filter with an additional LP-filter in conjunction with the adaptive sigma function. The measured azimuth angles are overlaid with the estimated expected location of the speaker.

From Fig. 2.5 and Fig. 2.5 we see that the tracker has difficulty in finding the speaker. Initially the tracker locks onto a clutter trail before locking onto the speaker. When the tracker is locked onto the speaker it maintains an accurate location estimate throughout the recording. We assume that we can overcome this problem (the initial offshoot and the tracking of the clutter trail) by altering the initial location of the particle population. Therefore we conducted an additional experiment with the second recording (*Recording 2*), where the particles are initiated across the azimuth range ($[0.6, \ldots, \pi/2]$). The result is presented in Fig.2.23.

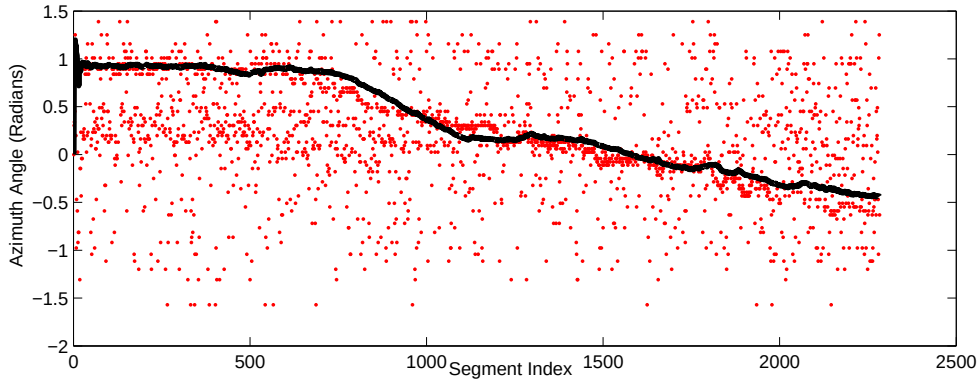

Figure 2.23: Result for the second recording (*Recording 2*), obtained from the particle filter with an additional LP-filter in conjunction with an adaptive sigma function where the particles are initiated across the azimuth range ($[0.6, \ldots, \pi/2]$). The measured azimuth angles are overlaid with the estimated expected location of the speaker.

From Fig.2.23 we see that the tracker is able to successfully track the speaker throughout the recording. We note that the microphone pair is the most sensitive to measurements that originate from the azimuth angle of 0 radians. Therefore we assume that the clutter (in this sensitive direction) in conjunction with the initial (uniform) distribution of the particle population across the potential azimuth range ($[-\pi/2, \ldots, \pi/2]$) causes the tracker to lock on the clutter trail in Fig. and Fig. .

2.6 Concluding Remarks

Concluding, *extension 1* pre-processes the signal with a LP-filter, excluding the spectral region from which potential noise artefacts may arise. Thereby suppressing the ill-posed nature of the inverse map from TDOA estimate to azimuth location. Thus, the typical multi-valued solutions from the localization function are suppressed. Assuming that the suppressed peaks do not correspond to the "true" source location, the inference method is provided with more accurate azimuth measurements that correspond to the location of the speaker. The proposed *extension 2* is an alternative attempt to suppress the ill-posed nature of the tracking problem. The typical multi-valued solutions obtained from the localization function are suppressed by excluding the segments that are considered as unreliable. In addition, information on the coherence energy distribution is effectively incorporated in the inference method, with an effective biased evaluation of the potential locations as result. Both extensions aim to provide more accurate and consistent sequences of location estimates, with respect to the true azimuth angle, for the inference method. With respect to the particle filter based tracking algorithm, the improvements will eventually lead to an increase of effective predictive particles and thus decreasing the amount of poor predictive particles [16]. With effective prediction, we obtain a more dense particle population in the area that corresponds to the true source location. As a consequence less samples are needed to accurately approximate the posterior distribution. Both extensions were tested several times, with every run yielding approximately the result as presented in the corresponding Fig. 2.8 (extension 1) and Fig. 2.15 (extension 2).

[16] With the assumption that the predictive particles are generated from a well-behaved (Langevin) motion model.

Chapter 3

Voice Feature Extraction

The extracted azimuth features within the multiple target tracking framework will have enough discriminative power to uniquely locate the speakers, provided that the speakers remain spatially separate. When the speakers paths cross, the azimuth features will not have enough resolution to distinguish between the people. Thus an additional feature is needed in order to disambiguate such difficult cases. The additional feature should provide such (approximately) consistent information for the robot, that it is able to recognize earlier stored patterns of the individual speakers voice. Though, in practice this is a rather challenging task, we attempt to uniquely identify the speakers on basis of vowel extracted information.

Speaker identity is correlated with the physiological and behavioural characteristics of the speaker. These characteristics exist both in the spectral envelope (vocal tract characteristics) and in the supra-segmental features (voice source characteristics and dynamic features spanning several segments). The most common are the features related to the spectral envelope. These features usually provide enough information to uniquely identify a particular vowel. In addition we assume that each speaker produces unique sounding vowels. Therefore, we attempt to build a vowel profile for each speaker during their conversation. This 'vowel profile' could then aid the tracking module whenever critical scenarios emerge such as the crossing of the speakers paths.

This chapter is organized as follows: first, human speech production is briefly discussed and finally several speech analysis techniques are discussed. We refer the reader to appendix D for a more elaborate discussion of the human speech production. The reason for presenting an elaborate discussion in appendix D is threefold. First, the techniques for speech analysis and synthesis are based on our understanding of human speech production. Therefore discussing human speech production should contribute to the comprehension of these techniques. Second, section D.1 aims to illustrate the motivation for choosing vowel features over consonant features [1]. Finally, by describing the sophisticated nature of the human speech organs, the reader should get an impression of the discrepancy between the actual and the modelled speech production. We feel that this discrepancy accounts for some of the problems we encountered during the experiments. To be more specific, whenever two speakers fall into the same category [2], a general approximation of the speech signal will usually not have enough resolution to uniquely identify a speaker.

[1] Obviously, incorporating both features should increase the robustness of the tracking module. However, due to limited time, only one additional feature extraction algorithm was implemented and tested.

[2] Category in this specific context is based on: age, gender etc.

3.1 Formants

A *phoneme* can be distinguished from another by its frequency spectrum (for further details see section D.1). Each spectrum contains peaks where certain resonance frequencies appear. The shapes of the resonance peaks are determined by the configuration of the *vocal tract*. The vocal tract consists of the pharynx, the mouth, and the nasal cavity. The configuration of the vocal tract depends on the movement of organs, such as the tongue, the lips, and the soft palate. Although the vibration of the vocal cords determines the pitch and intensity of the speech sounds. It is the specific character and combination of the resonances of the vocal tract that are responsible for distinguishing one phoneme from another [17].

Considering the vowel sounds, the peaks that occur in their spectra independent of the pitch, are referred to as *formants* [3]. Each formant corresponds to one or more resonances in the vocal tract. A formant is described by its position in the frequency domain expressed in Hz, along with the height of the resonance or amplitude expressed in dB, and its bandwidth describing the breadth of resonance region also expressed in Hz. As stated earlier, distinction between phonemes is possible through their unique spectra. For humans, perceiving the character of the vowel sound is mainly determined by the first and second formant. However, one would notice a considerable change in the vowel sound, if one of these formants is filtered out. Typically, no phonetic association would be produced, if one was to perceive the formants singularly.

It is custom to use a phonetic alphabet to indicate the various phonemes. Various phonetic alphabets have been developed by speech science researchers. Throughout this thesis we will use consistently the single-letter ARPAbet symbols for phonetic transcription[4]. In addition the different vowels will be denoted in this thesis by their corresponding phonetic transcription placed between slashes. For example the vowel contained in the word 'call' is denoted with the symbol /c/.

3.2 Speech Analysis

In general, speech analysis attempts to reconstruct the configuration of the vocal tract along with the excitation source. The assumption is made that this specific combination was responsible for the observed speech segment.

Probably the most popular method for extracting spectral information from speech is *linear prediction* (LP) based speech analysis. LP analysis relies on the assumption that a speech signal over a short time interval can be approximated by specifying three types of parameters. The first parameter type specifies the excitation force, this can be either a quasi-periodic train of impulses for voiced speech or random (white) noise for unvoiced speech. The second parameter type specifies, if used the frequency of the periodic wave. The third parameter type specifies the coefficients of the filter used to mimic the vocal tract response.

Typically, extracting voice features involves pitch extraction. For our initial voice-feature extraction algorithm we have implemented an autocorrelation-based pitch extraction method.

[3] The name formants is derived from 'to form' since they form the shape of the spectral envelope.

[4] There are two versions of the ARPAbet: one with single-letter symbols and one that uses all uppercase symbols. The latter version uses some double letter designators. The name ARPAbet originates from its developers, United States Advanced Research Projects Agency (ARPA).

However we found the results obtained from this algorithm not satisfying. We note that the speakers involved in the experiments were both male and thus have similar pitch values. The obtained pitch values for each speaker were too close to each other and inconsistent to provide discriminative power. Therefore we did not incorporate pitch extraction our voice feature extraction algorithm. We briefly discuss pitch extraction and the autocorrelation-based pitch extraction method in section D.3.

3.2.1 Linear Prediction Coefficients

The above mentioned parameters form also the basis for the *linear prediction coefficients* (LPC) method for speech compression. Recorded speech segments are characterized according to the three parameter types of the model. In addition these extracted parameters are also useful for speech synthesis. The most common are the features specified by the second parameter, since they are related to the specific shape of the spectral envelope. In the simplest case these features are taken to be the LPC coefficients. More elaborate approaches use LPC-cepstral coefficients together with their regressions coefficients [40]. A comprehensive survey of speaker identification techniques is provided in [23].

The main underlying assumption of the LPC method is that speech is produced by a buzzer at the end of the tube. The glottis is responsible for producing this buzz, which is characterized by its pitch and intensity. The tube is formed by the vocal tract and is characterized by its resonances i.e. the formants. Thus the speech is modelled as if it is generated by a particular source and system (filter). This type of analysis is called source-filter separation. In practice, the LPC method analyses the speech by estimating the formant locations, and in addition remove their effects. This process is called inverse filtering and the remaining signal is called the residue. Figure 3.1 shows speech as the output of a linear system model.

Figure 3.1: Speech model as a linear time-varying (LTV) system

With the assumptions that speech is the output of a linear system model and that the linear system is slowly time-varying, the system function [5] can be described as:

$$H(z) = \frac{G}{1 + \sum_{k=1}^{P} \alpha_k z^{-k}}, \tag{3.1}$$

where G denotes the gain of the filter, P denotes the order of the LPC model and α denotes the filter coefficients. Central to the idea of linear prediction is to approximate each sample of the speech signal as a linear combination of past samples. Thus, for such a system the speech output is related to the input by the difference equation where each sample of the signal is expressed as a linear combination of the P previous samples:

$$s(n) = -\sum_{k=1}^{P} \alpha_k s(n-k) + Gu(n), \tag{3.2}$$

[5] For short time intervals.

where $u(n)$ is the input or excitation that is either white noise or quasi periodic train of impulses. The aim is to find the set of prediction coefficients a_k, since these coefficients characterize the formants. Thus, the mean-squared prediction error between the speech signal $s(n)$ and the predicted signal based on a linear combination of past samples must be minimized. The short-time prediction error is then given by:

$$E = \sum_n e^2(n) = \sum_n \left\{ s(n) - \hat{s}(n) \right\}^2, \tag{3.3}$$

where $\hat{s}(n)$ is the prediction of speech signal $s(n)$ by the sum of P weighted samples of the speech signal and is defined as:

$$\hat{s}(n) = -\sum_{k=1}^{P} a_k s(n-k). \tag{3.4}$$

Thus, aiming to find the formant frequencies from the speech spectrum we need to know the modelled system and the frequencies of its resonances. LP techniques make the assumption that the prediction coefficients are identical to the speech model parameters. With this assumption in conjunction with equation (3.3) we see that the output of the prediction error filter is:

$$e(n) = -\sum_{k=1}^{P} \alpha_k s(n-k) + Gu(n) + \sum_{k=1}^{P} a_k s(n-k) \equiv Gu(n), \tag{3.5}$$

and the corresponding system function of the prediction error filter is:

$$A(z) = 1 + \sum_{k=1}^{P} a_k z^{-k}. \tag{3.6}$$

From equation (3.5) we see that the prediction error or LP residual, mostly contains information about the excitation force. If one was to pass the speech signal trough this inverse filter with the optimal values for the prediction coefficients, then this would yield an estimate for the excitation source. Estimating the formants involves computing the roots of the inverse of the prediction error filter. These roots correspond to the poles of the estimated speech spectrum. The location of these poles should be such that the original speech signal is modelled as closely as possible. Each pole roughly corresponds to the vocal tract pole i.e. the formant frequency [6].

Minimizing equation (3.5) in practice involves the computation of a matrix of coefficients values and the solution to a set of linear equations. Several methods for minimizing are available: autocorrelation, covariance and recursive lattice formulation. In the experiments we have used the MATLAB LPC implementation. The LPC algorithm provided by MATLAB uses the autocorrelation method of *autoregressive* (AR) to find the filter coefficients. Thus, in order to find the formant frequencies from the filter we need to find the locations of the resonance regions that characterize the filter. This implies considering the filter coefficients as a polynomial and solving for the roots of the polynomial. We do not further elaborate on this method since we regard the details as outside the scope of this thesis.

[6] Note that the speech spectrum depends on the specific vocal-tract configuration and displays information on the resonant structure of the vocal-tract.

Pre-Emphasis The spectrum of a voiced speech segment has a natural spectral tilt, with the frequencies below $1kHz$ having greater energy compared to the higher frequencies. This energy distribution is due to glottal waveform and the radiation load from the lips (for further details see appendix D). In order to model the speech segment as closely as possible according to the speech model described in section 3.2.1 it is desirable to attenuate the lower frequencies. Therefore it is customary to pass the speech signal through a *pre-emphasis* filter to boost the signal spectrum by $6\,dB$/octave. As a result the energy in the speech spectrum is re-distributed, such that the contribution of the glottal waveform and radiation load effect from the lower frequencies is approximately removed. The system function of the pre-emphasis filter is given by:

$$P(z) = 1 - \beta z^{-1} \tag{3.7}$$

After pre-emphasising, equation 3.2 to model the speech signal becomes:

$$s(n) = -\sum_{k=1}^{P} \alpha_k s(n-k) + Gu(n) + \beta\, Gu(n-1). \tag{3.8}$$

Where β denotes the pre-emphasis factor. Figure 3.2 presents the spectra of the spoken vowel /@/ (as in the word 'hat') with and without applying pre-emphasis. From Fig. 3.2 we

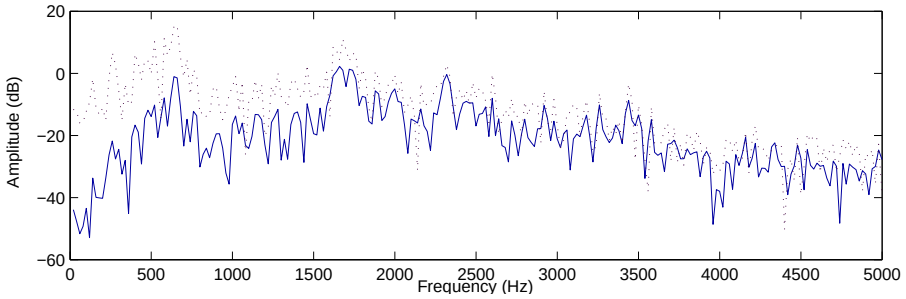

Figure 3.2: Spectra of spoken vowel /@/. The bottom spectrum corresponds to the pre-emphasized speech segment (solid line). The top spectrum corresponds to the speech segment without pre-emphasis (dotted line).

see that lower part of the spectrum is suppressed such that the regions of resonance in the spectrum approximately correspond to the regions of resonances of the vocal-tract.

3.2.2 The Adaptive Band-Pass Filterbank

An alternative to the LPC method as described in section 3.2.1 to estimate the formant frequencies from the spectral envelope of the observed speech segment is a method that involves the adaptive band-pass filterbank. The most part of this section describes the work of [43]. The author of [43] contributed to the work of [51] by adding additional adaptive components to the adaptive filterbank. We have modified the adaptive filterbank in order to construct the DFT [7] component in our formant frequency extraction algorithm.

The adaptive filterbank consists of individual formant filters and adaptive energy detectors. In addition the adaptive filterbank is extended with moving average decision maker,

[7] The DFT algorithm will be described in section 3.3.3.

that contributes to the adaptive character of the filterbank. The formant filters are designed in the complex domain because it yields several advantages over time- and frequency domain. Designing unity gain and zero-lag filters is easier in the complex domain furthermore the amount of aliasing of the signal is decreased when the signal is converted to complex domain [43]. The latter advantage yields an increase of accuracy of the spectral estimation technique that is used for formant frequency estimation. Thus in order to track the formants the speech signal is converted into its complex representation. The complex form of the discrete real-time signal $s_r(n)$ can be represented by the following form:

$$s_c(n) = s_r(n) + j\, s_h(n), \tag{3.9}$$

where $s_h(n)$ denotes the Hilbert transform of $s_r(n)$ and is defined as follows:

$$s_h(n) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{s_r(\tau)}{n-\tau} d\tau, \tag{3.10}$$

where τ denotes the lag over which is integrated. From equation (3.10) we see that ideal Hilbert transforms cannot be implemented in real-time. Therefore an approximate Hilbert transform is used for conversion. In practice the techniques that convert the real-time discrete signal into its complex representation involve a *finite impulse response* (FIR) filter for producing the imaginary part combined with a delayed version of the real-time signal. The real part of the delayed signal [8] is added back to the Hilbert transform to obtain the complex representation of the signal.

Formant Filters The filterbank divides the signal into bands where the formant estimates are tracked individually. One can think of this as analysing different "copies" of the segment, in each copy different regions of resonances in the spectral envelope are emphasized and suppressed. Each "copy" emphasizes one of the formants (by placing a pole on the corresponding formant frequency) while the remaining formants are suppressed (by placing zeros on the remaining formant frequency estimates). Thus each channel of the filterbank analyses a different "copy" of the segment.

In order to emphasize and suppress different resonance regions, each channel of the filterbank consists of an *all-zero filter* (AZF) cascaded with a single pole *dynamic tracking filter* (DTF). This combination is responsible for the individual formant tracking and is called a *formant filter*. The zeros and pole of each filter are varied adaptively. Therefore, the filters allow suppression of interference from the neighbouring formant frequencies and/or from spectral noise sources, while tracking an individual formant frequency as it may vary within the windowed frame (segment).

AZF Each formant filter sets its zero locations to the values of previous values estimated from the other formant filters. The all-zero filter is adaptive in the sense that the zero locations are adjusted at each sample index n. At each sample index n, each formant filter estimates the formant frequencies from the previous segment. The transfer function of the f^{th} AZF at segment index n is:

$$H_{AZF_f}(n,z) = K_f(n) \prod_{i=1}^{\phi_{max}} (1 - r_z \exp^{j2\pi\, \phi^i_{n-1,i}}\, z^{-1}), \qquad i \neq f, \tag{3.11}$$

[8] The real-time signal is delayed to account for the delay in implementing the approximate Hilbert transform.

where r_z denotes the radius of the zeros of the Z-plane, $\phi^i_{n-1,i}$ denotes the formant frequency estimate of the i^{th} filter at segment index $n-1$ and ϕ_{max} denotes the maximum formant frequency that is estimated by the adaptive filterbank. The gain K_f ensures that the AZF has unity gain and zero phase lag at the estimated formant frequency and is defined for the f^{th} component at segment index n as:

$$K_f(n) = \frac{1}{\prod_{i=1}^{\phi_{max}} (1 - r_z \exp^{j2\pi\, \phi^i_{n-1,i} - \phi^f_{n-1,f}})}, \qquad i \neq f. \tag{3.12}$$

The suggested value for parameter r_z is $r_z = 0.98$ [43].

DTF The filter responsible for tracking an individual formant frequency is a single-pole dynamic tracking filter. The DTF is dynamic in the sense that for each sample shift, the initial pole location is set to the previous estimate of the formant frequency. Thus, at current segment n, each formant filter estimates its corresponding formant frequency based on its previous estimate. The transfer function of the f^{th} DTF at segment index n is:

$$H_{DTF_f}(n) = \frac{1 - r_p}{1 - r_p \exp^{j2\pi\, \phi^f_{n-1,f}}\, z^{-1}}, \tag{3.13}$$

where r_p denotes the radius of the pole and $\phi^f_{n-1,f}$ denotes the formant frequency estimate of the f^{th} filter at segment index $n-1$. The suggested value for parameter r_p is $r_p = 0.90$ [43].

Adaptive Energy Detector As described in section (3.1) a formant is characterized by its location, bandwidth and amplitude. The amplitude along with the formants bandwidth give a rough indication of the amount of energy present in a formant band. For each formant band the *root-mean-square* (RMS) energy is computed after the signal is filtered [9]. The computed RMS energies from the formant bands serve as an indicator whether the speech segment is voiced or not. In order for the observed speech signal to be voiced, the RMS energies computed from the formant bands have to be above certain energy thresholds. Similar to formant frequency estimation the energy threshold levels for each formant band are updated after each segment shift. The equation for the energy threshold level update is given by:

$$ET_f(n) = ET_f(n-1) - \big(0.002 * \big(ET_f(n-1) - \mathcal{E}_{n,f}\big)\big), \tag{3.14}$$

where $ET_f(n)$ denotes the energy threshold at segment index n for the f^{th} formant frequency. The energy threshold for previous segment is denoted with $ET_f(n-1)$ for the f^{th} formant frequency and $\mathcal{E}_{n,f}$ denotes the measured RMS energy for formant f at segment n. All values are in dB. Equation (3.14) is insensitive to (isolated) abrupt changes. However if the changes appear to be consistent of character, the energy threshold level adapts gradually to these changes. The segment is analysed with the following initial values for the energy threshold levels:

$$\begin{aligned} InitialF_1EnergyThresholdLevel &= -35dB \\ InitialF_2EnergyThresholdLevel &= -40dB \\ InitialF_3EnergyThresholdLevel &= -45dB \end{aligned}$$

[9] To obtain the RMS value of an n-element vector x we use: $RMS(x) = \texttt{norm}(x)/\texttt{sqrt}(n)$.

Note that these initial values are calibrated for speech signals whose energy levels have been normalised to have a mean of 0 dB [43][10].

Moving Average Decision Maker If the RMS energy level remain approximately constant throughout the analysis by the filterbank, then the formant frequency estimates are assigned to the segment. If the segment is not voiced or contains inconsistent energy levels due to local maxima, then the current estimate of the formant frequency decays toward the moving average value for the corresponding formant frequency. The update equation for the moving average value for each formant frequency f is given by:

$$\phi_f^{MA}(n) = \frac{1}{n} \sum_{k=1}^{n} \phi_{n,f}. \tag{3.15}$$

The energies from the formant bands are calculated independently of each other. Thus, this may result in a scenario where some of the formant band energies are above their energy thresholds, while other are below their energy thresholds. As a consequence, the formants that have an energy level below their energy threshold are not spectrally estimated and decay towards their average value according to:

$$\phi_{n,f} = \phi_{n-1,f} - \big(0.002 * \big(\phi_{n-1,f} - \phi_f^{MA}(n-1)\big)\big), \tag{3.16}$$

where $\phi_{n,f}$ denotes the formant frequency estimate at segment index n for the f^{th} formant frequency and $\phi_f^{MA}(n-1)$ denotes the previous value of the moving average at segment index $n-1$ for the f^{th} formant frequency.

3.3 Formant Frequency Extraction Algorithm

The formant frequency extraction algorithm is based on two components: LPC analysis for the initial formant frequency estimates and a formant band filter for adjusting and verifying the initial estimates. This section gives an overview of the formant frequency extraction algorithm that is used in the experiments.

3.3.1 Energy Comparison

In order to decrease the computational load and the amount of false positives each segment is first analysed on basis of its energy distribution. If the segment contains a clear vowel utterance and thus is not intermixed with undesired noise artefacts that contains high frequency components, then this segments should mainly contain energy in the $0 - 4000$Hz region. Therefore we apply a LP filter with a cut-off frequency at 4000Hz to the segment. If the amount of energy in the LP filtered segment is approximately equal to the amount of energy of the "original" segment, then we consider the segment as a potential "vowel segment". For each frame we compute the energy $\mathcal{E}$ of segment g and the energy $\mathcal{E}_{lp}$ of the low-pass filtered segment g_{lp} as follows:

$$\mathcal{E} = \frac{\sum_{n=1}^{N} |\mathcal{F}\{g(n)\}|}{2}, \tag{3.17}$$

$$\mathcal{E}_{lp} = \frac{\sum_{n=1}^{N} |\mathcal{F}\{g_{lp}(n)\}|}{2}. \tag{3.18}$$

[10] After pre-emphasizing the segment, the RMS energy levels are normalized in order to have a mean of 0 dB.

Typically, these energies will not be exactly equal, since the original segment potentially contains energy in the frequencies up to 24kHz whereas the low-pass filtered segment only contains energy in the frequencies up to 4kHz. Therefore we allow those segments for LPC analysis that satisfy the following equation:

$$(\mathcal{E} * energyRatio) < \mathcal{E}_{lp}, \tag{3.19}$$

where $energyRatio < 1$ and is responsible for controlling the quality of the segments that are allowed for LPC analysis. Note that choosing high value for $energyRatio$ allows only those segments for LPC analysis where approximately all the energy is concentrated in the lower regions of the spectrum. Whereas choosing a low value for $energyRatio$ will allow segments that have a more normal energy distribution.

The value for $energyRatio$ was experimentally chosen as follows $energyRatio = 0.45$. This rather low value is a compromise between allowing as many possible segments for LPC analysis and excluding the high potential false positive segments. Note that with setting the $energyRatio$ to a high value, we exclude potential "positive" segments.

Consider the following example: a segment containing a spoken vowel along with some high frequency noise artefact. However the speech and noise components are not intermixed in the frequency domain (the different components manifest on different sides of the frequency spectrum). Therefore the formant frequency estimation of the speech segment obtained from the LPC analysis will not be affected by this high frequency noise artefact. Thus this segment should be allowed for LPC analysis, by choosing the value for $energyRatio$ too high this segment will be excluded for further analysis. Therefore we choose the value for $energyRatio$ rather low. Furthermore, we are confident that the formant frequency extraction algorithm is robust enough to eventually exclude the false positive segments for building the speakers vowel profile.

3.3.2 Vowel Identification

The sections 3.2.1 and 3.2.2 described methods for voice feature extraction. These methods provide us with values that give an indication of the (speech) spectrum. However these methods do not identify the (speech) segment. In order to identify the (speech) segment we use two tables that contain information on average vowel utterances. The first table (table 3.1) based on research from Peterson and Barney [50] conducted in 1952 contains information on average formant frequencies for ten different vowels. The second table (table 3.2) based on research from Dunn [17] conducted in 1961 contains information on average (maximum) bandwidths for these average formant frequencies. With these tables we can construct an

Vowels	/i/	/I/	/E/	/@/	/a/	/c/	/U/	/u/	/A/	/R/
$\mathcal{F}_1$	270	390	530	660	730	570	440	300	640	490
$\mathcal{F}_2$	2290	1990	1840	1720	1090	840	1020	870	1190	1350
$\mathcal{F}_3$	3010	2550	2480	2410	2440	2410	2240	2240	2390	1690

Table 3.1: Average formant frequencies (Hz) for male.

VOWELS	/i/	/I/	/E/	/@/	/a/	/c/	/U/	/u/	/A/	/R/
$\mathcal{B}_1$	80	100	120	140	160	120	120	100	140	80
$\mathcal{B}_2$	120	120	140	200	80	200	200	140	140	120
$\mathcal{B}_3$	300	300	300	300	300	240	200	200	300	120

Table 3.2: Average (maximum) formant bandwidths (Hz) for male.

(average) template for each of these vowels. We use the average formant frequency values as the mean and the average bandwidths as the covariance. We have implemented and tested several distance functions for vowel identification. We obtained the best results with the *Gaussian curve membership function* (*gaussmf*). Therefore, in order to identify the vowel on basis of the estimated formant frequencies we use a distance measure based on the *gaussmf* that is provided by MATLAB. We have no analytical explanation for this choice other than the experimentally-based motivation. However we note that the speakers used in the experiments are not native English speakers. This factor in conjunction with the specific values used for mean and covariance might be an influence in the difference in results obtained from the various distance functions. The distance function used for vowel identification at segment index n takes the following form:

$$d(\gamma|\phi_n) = \prod_{f=1}^{\phi_{max}} \exp^{\frac{-(\phi_{n,f}-m_f^\gamma)^2}{2(\sigma_f^\gamma)^2}}, \tag{3.20}$$

where $d(\gamma|\phi_n)$ denotes the distance between the measured formants $\phi_n = [\phi_{n,1}, \phi_{n,2}, \phi_{n,3}]$ correspond to the average formant frequencies $\gamma = [m_1^\gamma, m_2^\gamma, m_3^\gamma]$ that represent a particular vowel γ. With:

$$\gamma\, \epsilon\, \{/i/, /I/, /E/, /@/, /a/, /c/, /U/, /u/, /A/, /R/\}.$$

The term m_f^γ denotes the mean that corresponds to its entry in the average formant frequency table. The term σ_f^γ denotes the covariance that corresponds to its entry in the average formant bandwidth table. The term $\phi_{n,f}$ denotes measured formant frequency with formant index f from segment n and where ϕ_{max} denotes the maximum formant frequency that is used for identification. In the experiments we use $\phi_{max} = 3$, we note that a vowel can be uniquely identified with the first three formant frequencies.

For each vowel γ we compute $d(\gamma_n|\phi_n)$ and take $max\, d(\gamma|\phi_n)$ to find the most likely vowel to be present in segment n. however we only assign a particular vowel to the segment n if $max\, d(\gamma|\phi_n) > vT$. Where vT denotes a (vowel assignment) threshold and was experimentally chosen as $vT = 0.8$.

3.3.3 Dynamic Formant Tracker

As described in section 3.2.1 with the MATLAB LPC method we can estimate the areas of resonance frequencies from the speech spectrum. If the segment passed the first *energy comparison* test (section 3.3.1) and if the measured formant frequencies obtained from the LPC analysis passed the *vowel detection and identification* test ($max\, d(\gamma|\phi_n) > vT$, section 3.3.2) then the measured vowel is assigned to one of the speakers profile.

During the experiments we noticed that the *vowel detection and identification* test allowed for a considerable amount of false positives. Note that the *energy comparison* test does not guarantee that all false positive examples are excluded for the *vowel detection and identification* test. Occasionally due to coloured noise [11] a segment that does not contain a vowel may pass the *energy comparison* test. This may result in an even worse scenario where the segment may be classified as a segment containing a vowel. Thus here we obtain a false positive. By simply setting the vowel (assignment) threshold vT to a very high value will introduce an additional problem: very few entries for the speakers vowel profile. This will yield (unreliable) vowel profiles that do not represent accurately the speakers intrinsic voice characteristics.

Thus we need a method that allows positives examples into the profile and rejects all false positives examples, without increasing the vowel (assignment) threshold. Typically, these false positives segments do not contain consistent formant frequency information throughout the segment due to the spurious nature of the noise artefacts. This property inspired us to analyse the segment further with an dynamic approach. We apply a dynamic LPC analysis on each segment that passed both the *energy comparison-* and the *vowel detection and identification* test.

In order to avoid the entrance of false positive vowel examples into the speaker profiles we use a sub-band filter based on the filterbank presented in section 3.2.2. We use the filter in order to verify the initial formant frequency estimates by analysing their consistency throughout the segment. For the remaining part of the thesis we shall refer to this procedure as the *dynamic formant tracker* (DFT) algorithm. Table 3.3 presents an overview of the DFT algorithm.

For each segment:

1. Transform the windowed signal to its complex representation.
2. Initiate each formant filter with the corresponding formant frequency estimates obtained from the first LPC analysis.
3. Initiate a 20ms LPC window starting from the first sample of the segment.
4. Compute for the 20ms LPC window the formant frequencies.
5. Shift the 20ms LPC window one sample within the segment and initiate the formant filters with the corresponding formant frequencies estimates obtained from the formant filters in the previous step.
6. Repeat the two previous steps while the last sample of the 20ms LPC window is within the segment.

Table 3.3: Dynamic Formant Tracker Algorithm

The values obtained from the last 20ms LPC window are used to verify the initial formant frequency estimates. The initial estimates are verified if the formant frequencies obtained from the DFT algorithm yield the same vowel identification according to formula 3.20 in conjunction with the vowel (assignment) threshold vT.

[11] Note that, here we use the term coloured noise in order to indicate the noise in the lower end of the frequency spectrum.

The Fig.3.3 and Fig. 3.4 present the results of the (diverse) LPC analyses on the vowel /u/ (as in the word 'boot'). Despite the fact that the initial values are adjusted, the DFT verifies these initial formant frequency estimates.

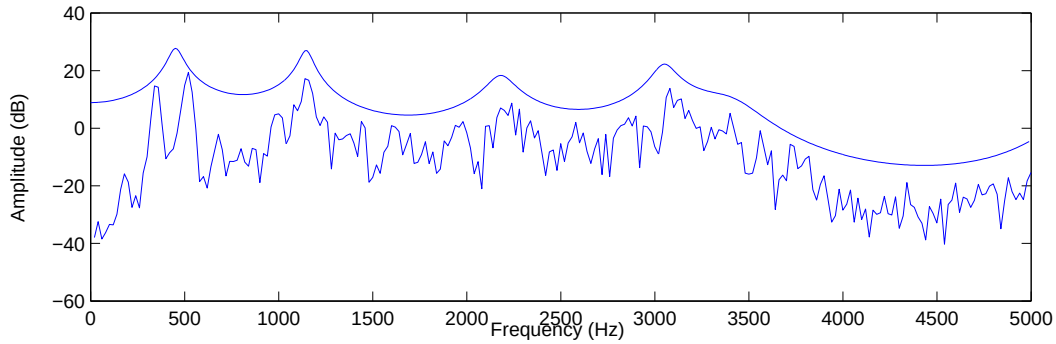

Figure 3.3: Analysis on the (spoken) vowel /u/. The bottom spectrum corresponds to the pre-emphasized speech segment. The top spectrum corresponds to the LPC approximation of the speech spectrum.

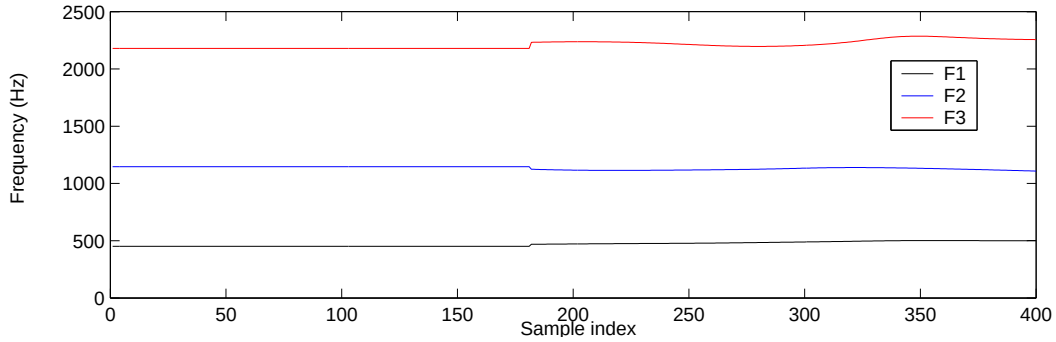

Figure 3.4: Analysis on the (spoken) vowel /u/ by the DFT. Each line presents the result of an individual formant filter on tracking its corresponding formant frequency.

To illustrate the verifying task of the dynamic formant tracker further, we present Fig. 3.5 and Fig. 3.6. Both Fig. 3.5 and Fig. 3.6 present the (diverse) LPC analyses on a segment that contains mainly (coloured) noise. Despite the fact that the segment does not contain a vowel, it passed the *energy comparison* test and *vowel detection and identification* test. Since the dynamic formant tracker cannot verify the presence of consistent (high) energy levels in the resonance regions, the initial found formant frequency estimates are not verified.

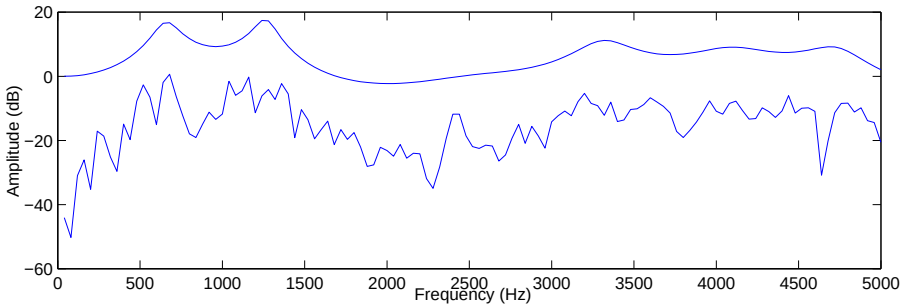

Figure 3.5: Analysed segment that contains mainly (coloured) noise. The bottom- and top spectrum correspond to respectively the pre-emphasized- and the LPC approximated signal.

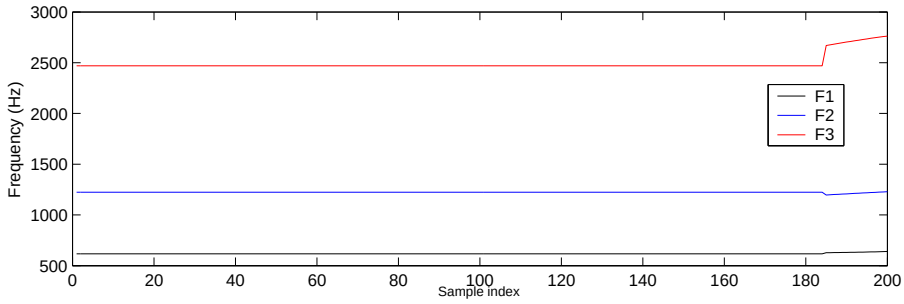

Figure 3.6: Analysis on the segment that contains mainly (coloured) noise by the DFT. Each line presents the result of an individual formant filter on tracking its corresponding formant.

3.3.4 Expanding Window LPC Analysis

The variation in duration of vowel utterances is considerable. Typically a duration may vary from 40 to 400ms. Therefore it is difficult to define the ideal window length that would capture each vowel utterance perfectly. For the initial testing we used a window length of 50ms, this in order to ensure that we could potentially capture the shortest vowel duration as a whole for analysis [12].

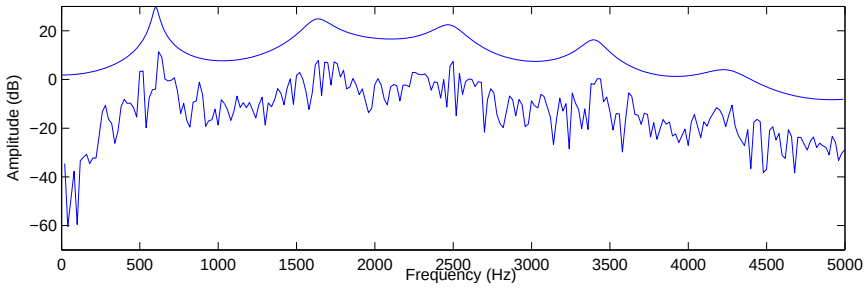

Figure 3.7: Analysis on vowel /@/. The bottom spectrum corresponds to the pre-emphasized speech segment. Top spectrum: LPC approximation of the speech segment.

[12]Note that the Hamming window reduces the potential edge effects. Therefore the beginning and the end of the (possible) speech utterances (that are typically present in the segment) are suppressed.

During the experiments we noticed that on a regular basis that segments containing (perfect) vowels passed through the *energy comparison* test, however they did pass *vowel detection and identification* test. Figure 3.7 presents a typical output of the LPC analysis of such a segment. In Fig. 3.7 we present the spectrum of the pre-emphasized speech signal and the LPC approximation of the speech spectrum. It stroke us as odd that such segments did not pass *vowel detection and identification* test since we heard a (perfect) vowel. Furthermore when we analysed the plots of the LPC approximation of the speech spectrum, the regions of resonance seemed to be correct.

Figure 3.7 presents the analysis on the vowel /@/ (as in the word 'cat') with average formant frequencies for male at $m_1^{@} = 660$Hz, $m_2^{@} = 1720$Hz and $m_3^{@} = 2410$Hz. These values approximately correspond to the peaks in the LPC spectrum, however the computed vowel distance (as described in section 3.3.2) did not reach the vowel (assignment) threshold vT. The reason for this is that the peak estimation algorithm found a local maximum at approximately 2000Hz. This value is assigned to the to third formant frequency estimate. The estimated formant frequencies at segment index n corresponding to Fig. 3.7 based on the first LPC analysis are: $\phi_{n,1} = 601$Hz , $\phi_{n,2} = 1628$Hz and $\phi_{n,3} = 1973Hz$. The first two formant frequencies fit the /@/ vowel template reasonably. However due to third formant frequency estimate the segment is not identified as vowel /@/. Thus a local maximum is the cause for the non-classification.

Figure 3.8 presents again the LPC analysis on the segment, now with two arrows indicating the local maximum and the frequency region in the LPC approximated spectrum that should have been suppressed. Since our aim is to collect as many vowel based information as possible for each speaker. This in order to build an appropriate and reliable profile for each speaker. Therefore we lose valuable information by rejecting this segment.

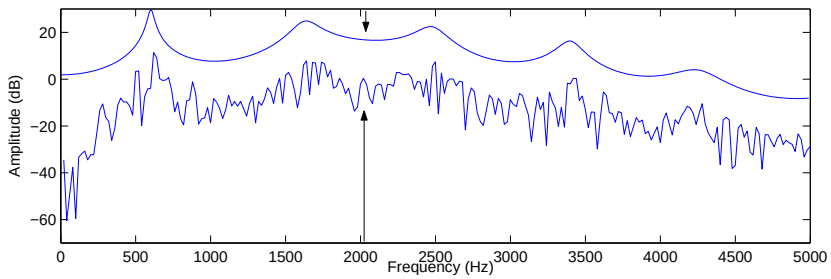

Figure 3.8: Analysis on a speech segment (vowel /@/). The bottom spectrum corresponds to the pre-emphasized speech segment. The top spectrum corresponds to the LPC approximation of the speech spectrum. The arrows indicates the local maximum responsible for the wrong formant frequency estimate.

We assume that this local maximum is spurious of nature. Therefore we assume that this wrong estimate would manifest itself if we conducted a dynamic LPC analysis within the 50ms segment. Therefore we propose the *centre expanding window* (CEW) LPC analysis. This method places a 20ms window in the centre of the 50ms segment. This window is analysed with the LPC method (as described in section 3.2.1) and the formant frequencies estimates are stored. Next we expand the window length with one sample on each side of the window. The previous procedure is repeated until the window length reaches the original segment length of 50ms. The rationale for this approach is that, if the initial formant frequencies values are

correct then they should be consistent of nature. Thus, the (correct) formant values found in each of the different window lengths should not differ much from each other. Whereas a wrong formant frequency estimate due to some temporal noise artefacts would manifest by a (larger) variation in the found values with the CEW method. Figure 3.9 presents the results of the CEW LPC analysis on the original 50ms segment.

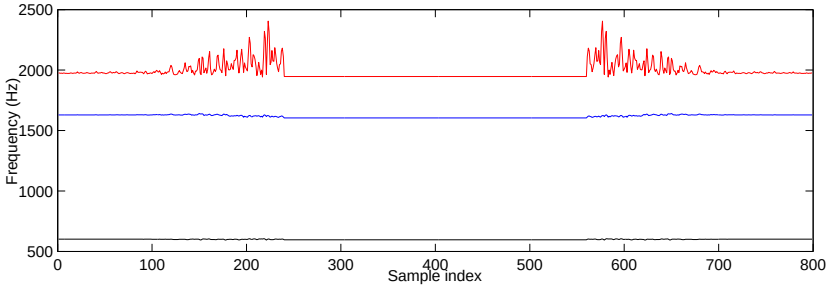

Figure 3.9: Result obtained from the Centre Expanding Window LPC analysis on the first three formant frequency estimates

From Fig. 3.9 we see that the third formant frequency estimate suffers from the largest variation in values. Thus we have a strong indication that the original estimate for the third formant frequency is wrong. Therefore we want to replace this value with an appropriate one, in order to provide the formant band filter with proper initial values. With use of the average formant frequency values and a modified version of formula 3.20 we can replace the wrong estimate with an appropriate average value. First we select the wrong estimate that is to be replaced. For each estimated formant frequency $\phi_{n,f}$:

$$ReplaceValue_{\phi_{n,f}} = [min(\phi_{n,f})/max(\phi_{n,f})], \tag{3.21}$$

where $ReplaceValue_{\phi_{n,f}}$ indicates for each estimated formant frequency $\phi_{n,f}$ the necessity for replacement. In order to find its corresponding formant index we simply compute:

$$\phi_f^R = argmin[ReplaceValue_{\phi_{n,1}}, \ldots, ReplaceValue_{\phi_{n,\phi_{max}}}]. \tag{3.22}$$

Thus ϕ_f^R denotes the formant with the highest variance in measurements obtained from the CEW analysis. If the corresponding $ReplaceValue_{\phi_{n,f}}$ for ϕ_f^R is below some (replace) threshold rT then it should be replaced. This threshold rT ensures that only those formant frequency estimates with a large variance (in their corresponding CEW LPC analysis results) are considered for replacement. The value for this threshold was experimentally chosen as $rT = 0.9$. Second we re-compute the distance between the estimated formant frequencies and the average formant frequencies similar to the procedure as described in section 3.3.2. However without the formant frequency that is to be replaced, giving:

$$d(\gamma|\phi_n) = \prod_{f=1}^{\phi_{max}} \exp^{\frac{-(\phi_{n,f}-m_f^\gamma)^2}{2(\sigma_f^\gamma)^2}}, \quad f \neq \phi_f^R. \tag{3.23}$$

Thus, for the two remaining formant frequency estimates we determine the best fit for one of the average vowel templates with the use of formula 3.23. As described in section 3.3.2 we compute $d(\gamma|\phi_n)$ for each vowel template and take $argmax_\gamma\, d(\gamma|\phi_n)$ to find the most likely vowel to be present in segment n. If $max_\gamma\, d(\gamma|\phi_n)$ is above the vT then we replace the wrong

formant frequency estimate with the average value corresponding to the "missing" formant frequency. We take the average value from table 3.1 where the selected formant frequency entry corresponds to the "missing" formant frequency. Thus the formant band filter is initiated with the two formant frequency estimates and with an average formant frequency value. We now simply follow the procedure for formant frequency extraction as presented in section 3.3. That is starting from initiating formant filters with the "new" values. Figure 3.10 presents various results of the LPC analysis after the wrong formant frequency is replaced with a corresponding average value based on the CEW LPC analysis [13].

From Fig. 3.10 we see that the spurious local maximum is no longer present. Note the much steeper descend in the LPC approximation of the spectrum in the 2000Hz region. Thus by analysing the segment with the CEW LPC analysis we can detect a possible local maximum that causes a false formant frequency estimate. In addition with the use of table 3.1 we can replace this estimate with an appropriate average value. This enables us to initiate the formant band filter with proper values such that we can recover formant frequency information from the segment.

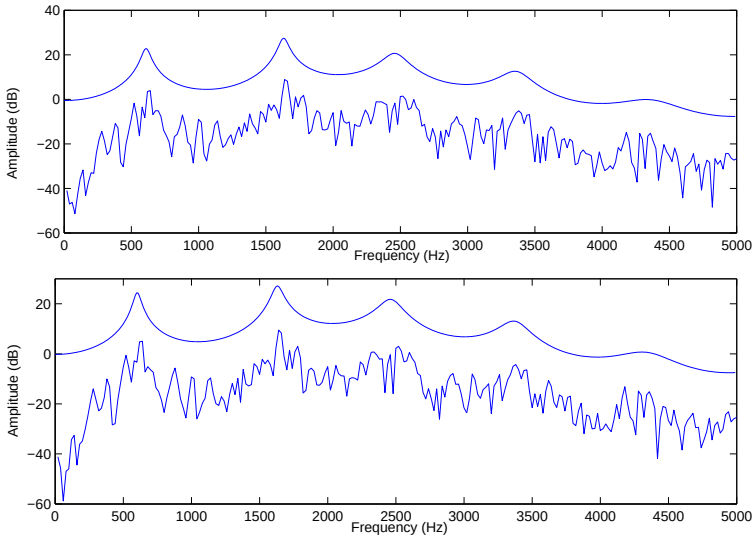

Figure 3.10: Various results obtained from the formant frequency extraction algorithm after the wrong formant frequency estimate is replaced according to the CEW analysis with its corresponding average value. Both figures present analysis on a speech segment (vowel /@/). In each figure, the bottom spectrum corresponds to the pre-emphasized speech segment. The top spectrum corresponds to the LPC approximation of the speech spectrum.

3.3.5 Summary

Since there are several procedures involved in the formant frequency extraction algorithm we feel that by providing a brief overview in conjunction with a schema of the algorithm will increase the comprehension of the algorithm. Figure 3.11 presents a schematic overview of the formant frequency extraction algorithm.

[13]The LPC analysis is applied on windowed data centred around the current segment n with a shifts of 1ms. This procedure will be discussed in more detail in section 3.6.

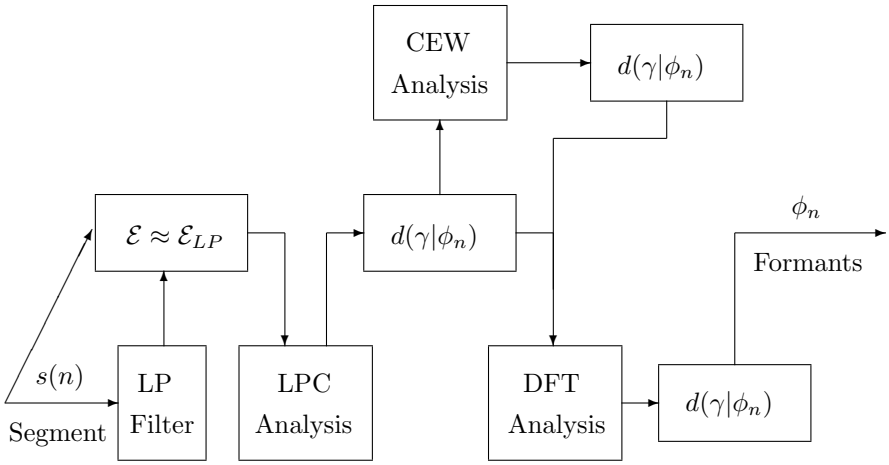

Figure 3.11: Schematic representation of formant frequency extraction algorithm.

Thus each segment is first analysed on its energy distribution. If this distribution satisfies $(\mathcal{E} * energyRatio) < \mathcal{E}_{lp}$ then LPC analysis is applied on the segment. The results obtained from the LPC analysis on the segment have to satisfy $max\, d(\gamma|\phi_n) > vT$. If not, then CEW LPC analysis is applied on the segment. If these results do not satisfy $max\, d(\gamma|\phi_n) > vT$ then the algorithm stops and no vowel is detected in the segment. If the results obtained from the (CEW) LPC analysis satisfy $max\, d(\gamma|\phi_n) > vT$ then these values are used to initiate the formant band filter (DFT analysis). If the results from the DFT analysis satisfy $max\, d(\gamma|\phi_n) > vT$ then the extracted formant frequencies are used to build the vowel profile for one of the speakers.

3.4 Algorithm Extensions

We tested our formant frequency extraction algorithm and found the results satisfying, in the sense that the individual vowel detection and classification were correct. However, when we used the vowel profiles for the data association problem we noticed that the profiles solely based on formant frequencies did not have enough discriminative power to robustly distinguish between the two speakers.

We believe that the reasons for this are three-fold. First the speakers are not native English speakers, therefore they are hindered to speak in a natural fashion. This issue is important since, in order to build reliable profiles we need reliable speech utterances that remain approximately constant for each speaker (see also section D.2). We assume that in our experiments the unique character of each voice is compromised since the speech utterances produced by (our not-native) speakers exhibit a relative wide range of values. We note that typically non-native speakers are identified (among other inferences) by their incorrect placement of stress and timing[14]. Second we assume that, since both speakers are male they (naturally) produce similar sounding vowels. We note that similar (physical configured) speech organs produce similar sounding speech[15]. Third we assume that whenever the speakers were further away from the microphone pair the accuracy of the measurements decreased. Note that, if

[14]If these speakers were to speak in a non-native language, for further details we refer to section D.2
[15]Some background on this subject is provided in appendix D.

the distance is increased between the microphone pair and the speaker then the potential influence of noise artefacts increases.

3.4.1 RMS Energy

In order to increase the discriminative power of the speakers profiles, we added information to the formant profiles from an other dimension: the RMS energy estimates from each formant band $\mathcal{E}_n = [\mathcal{E}_{n,1}, \mathcal{E}_{n,2}, \mathcal{E}_{n,3}]$. As described in section 3.2.2 the RMS energy of each formant band is computed. Thus we extract formant frequency information in conjunction with their corresponding RMS energy values from each segment that is used to construct a formant profile. We found an improvement in correct vowel/speaker association with the use of the additional information in the RMS energy dimension.

To illustrate the difference between the basic and the extended formant frequency extraction algorithm we present Fig. 3.12 and Fig. 3.13.

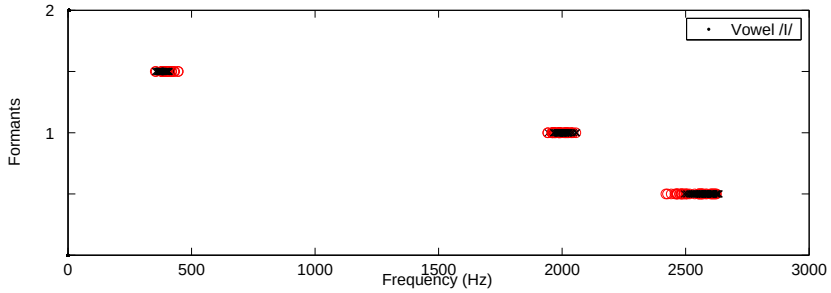

Figure 3.12: Results obtained from the formant frequency extraction algorithm for the vowel /I/. The results that correspond to the first speaker are indicated with circles. The results that correspond to the second speaker are indicated with crosses.

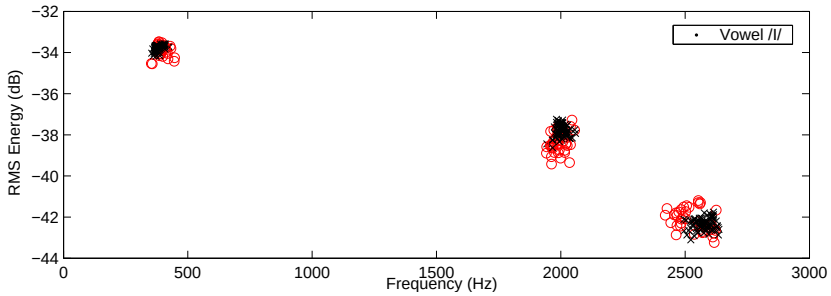

Figure 3.13: Results obtained from the extended formant frequency extraction algorithm for the vowel /I/. The results that correspond to the first speaker are indicated with circles. The results that correspond to the second speaker are indicated with crosses.

From Fig. 3.12 and Fig. 3.13 we see that the results corresponding to the different speakers are easier to distinguish from each other by adding the RMS energy information. Therefore we assume that the additional information enables the extended algorithm to accomplish a more robust classification. Despite the improvement we were still not satisfied with the amount of correct vowel/speaker assignments. Therefore we searched for additional

features that we could extract from the segments that are used to construct the formant profiles.

3.4.2 Re-estimation Of Formants

We aim to extract additional information from the segments such that the discriminative power of the profiles increases. We focused on ratio information that is based on the distances between the formant frequencies. The inspiration for extracting this additional feature came partly from the CEW LPC analysis experiments. During these experiments we noticed that typically the third formant frequency was replaced. We assume that this is mainly caused by its relative low energy. This property increases the probability that the estimate for the third formant frequency is more likely to be corrupted by possible noise artefacts, compared to the estimate for the first formant frequency. Thus we assume that the estimate for the first formant frequency is the most reliable estimate of all the formant frequency estimates.

Procedure 1: Re-estimate Each Formant Frequency With Average Distance Ratio Since we have a table of average formant frequencies (table 3.1) we also have information on average distances between the formant frequencies for each vowel. This allows us to re-estimate the third formant frequency from the first formant frequency estimate in conjunction with table 3.1.

Suppose we have detected the vowel γ_n with the formant frequencies $\phi_n = [\phi_{n,1}, \phi_{n,2}, \phi_{n,3}]$. We re-estimate the third formant frequency by simply computing the *average ratio* $\mathcal{R}_m^{1,3} = m_3^\gamma / m_1^\gamma$ that corresponds to the average values for the detected vowel γ_n and multiplying this ratio $\mathcal{R}_m^{1,3}$ with the first formant frequency estimate $\phi_{n,1}$. Obliviously a similar procedure allows us to re-estimate the third formant frequency with the second formant frequency as reference. We repeat these procedures for the first- and second formant frequency. In addition we can follow similar procedures to re-estimate the RMS energy $\mathcal{E}_n$ values. For re-estimating the RMS energy values we take the initial values (see section 3.2.2) as the reference. As a result we obtain two additional estimates of each formant. Thus in total we have acquired three estimates of each formant [16].

Note that by using average ratios for re-estimating the individual formant frequencies we implicitly assume that each speaker produces formant frequencies with similar ratios. This is a rather strong assumption since we initially assumed that each speaker produces a unique set of formant frequencies for each vowel. With this assumption it is likely that each speaker produces these formant frequencies with unique distances between them. Furthermore, one should be careful with re-estimating the formant frequencies for each speaker with (the same) average values (ratios) since this could result in more similar formant profiles for each speaker.

Procedure 2: Re-estimate Each Formant Frequencies With Measured Distance Ratio In order to emphasize the unique ratios for each speaker we introduce a variant of the above described procedure. This alternative procedure re-estimates each formant frequency with the use of the (unique) *measurement ratio* in conjunction with the average ratio.

Suppose we have detected the vowel γ_n with the formant frequencies $\phi_n = [\phi_{n,1}, \phi_{n,2}, \phi_{n,3}]$. First we re-estimate the initial third formant frequency estimate by computing the measurement ratio $\mathcal{R}_\phi^{1,3} = \phi_{n,3}/\phi_{n,1}$ and multiplying this with the estimated third formant frequency

[16] Note that we obtained the first estimate from our formant frequency extraction algorithm.

$\phi_{n,3}$. Second we scale the result of the multiplication back by dividing it by the corresponding average ratio $\mathcal{R}_m^{1,3}$. Note that if

$$\frac{\mathcal{R}_\phi^{1,3}}{\mathcal{R}_m^{1,3}} \approx 1$$

the effect of this procedure on the initial estimate will be negligible. However if the measurement ratio $\mathcal{R}_\phi^{1,3}$ differs substantially from the average ratio $\mathcal{R}_m^{1,3}$ then the effect will become considerable. Similar to the first described procedure we can repeat these steps for the other formant frequency estimates and the RMS energy values. Thus we obtain another two additional estimates of each formant.

Constructing Vowel Templates With The Re-estimated Formant Frequencies We are provided with two different methods to re-estimate the formant frequencies. The first method aims to suppress the influence of noise artefacts on the formant frequency estimations. The second method aims to emphasize the unique distances between the estimated formant frequencies for each speaker.

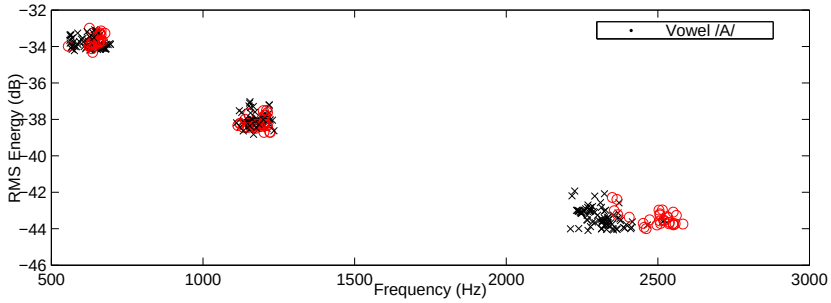

Figure 3.14: Results obtained from the formant frequency extraction algorithm for the vowel /A/.The first speaker is indicated with circles the second speaker is indicated with crosses

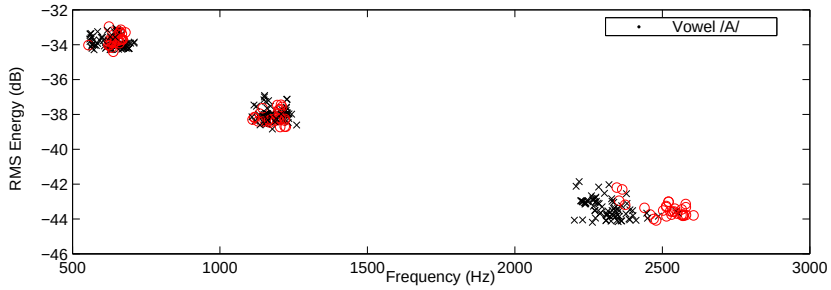

Figure 3.15: Results obtained from the formant frequency extraction algorithm extended with the re-estimation procedure for the vowel /A/. The first speaker is indicated with circles the second speaker is indicated with crosses

From the formant frequency extraction algorithm in conjunction with the re-estimation procedures we have obtained five different estimates of each formant. That is, five different estimates for each formant frequency. In addition, from each segment that is used to build the formant profile, we have obtained five different estimates for each RMS energy value.

Aiming to compress these values into one value, that is three formant frequencies and their corresponding RMS energy values, we use a Kalman filter in order to obtain this single value. For the remaining part of this thesis we refer to the combined formant profile with the corresponding RMS energy profile as the vowel templates.

The Kalman filter is initiated with the estimated formant frequencies and RMS energy values obtained from the formant frequency extraction algorithm. We update the initial estimates with the re-estimated values by applying the Kalman filter. Thus after the filtering process we are provided with a single estimate for each formant.

The Kalman filtered estimates are used to build the vowel templates. Figure 3.14 and Fig. 3.15 present a comparison of the obtained formant frequency estimates for the vowel /A/ from the different methods. Figure 3.14 presents the results obtained from the formant frequency extraction algorithm. Figure 3.15 presents the results obtained from the same algorithm however extended with the re-estimation procedures. At first glance there does not seem to be a substantial difference between the obtained results from the different methods. However we detect a substantial difference when we take a closer look at the second formant frequency estimates. Therefore we present Fig. 3.16 and Fig. 3.17.

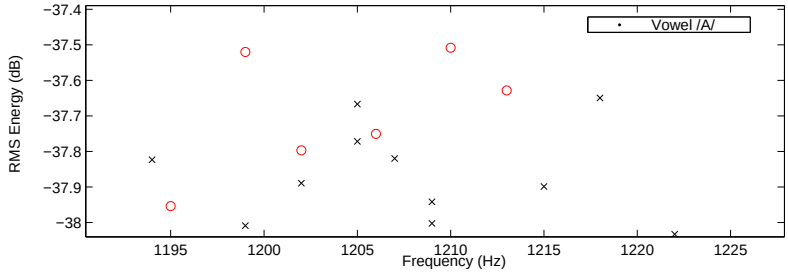

Figure 3.16: Results obtained from the formant frequency extraction algorithm for second formant frequency estimates of the vowel /A/. The formant frequency estimates that correspond to the first speaker are indicated with circles. The formant frequency estimates that correspond to the second speaker are indicated with crosses.

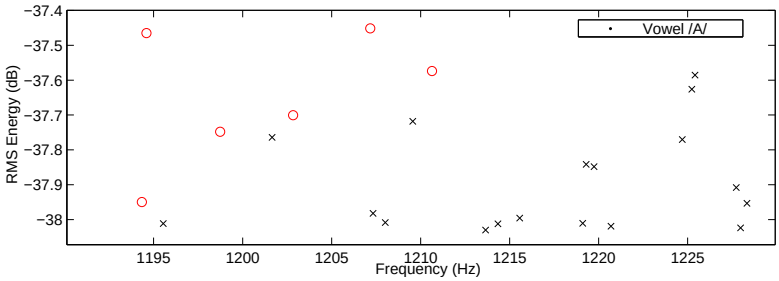

Figure 3.17: Results obtained from the extended formant frequency extraction algorithm extended with the re-estimation procedure for second formant frequency estimates of the vowel /A/. The initial estimated values are re-estimated with the ratio-procedures. The results that correspond to the first speaker are indicated with circles. The results that correspond to the second speaker are indicated with crosses.

From Fig. 3.16 and Fig. 3.17 we see an improvement. The results presented in Fig. 3.17 (obtained by applying additional re-estimation procedures) can be approximately linear separated. In order to determine whether or not the re-estimation procedure methods are a consistent improvement we tested the different methods on recorded data.

3.5 Comparative Experiments

We tested the different methods on recorded data that will also be used to test the tracking algorithm. We split the recording into two parts in order to obtain a part that is used to build the vowel templates. The remaining part is used to evaluate the vowel templates. Thus the first part of the recorded data is used build vowel templates for each speaker. As described in the beginning of the chapter we use a Kalman filter to build the vowel templates for each speaker. The second part is used to evaluate these constructed vowel templates. Since we have extended the measurement with an additional voice feature we have to extend our association event model. We evaluate the vowel templates with the following distribution functions:

$$p(\phi_n|f^i) = \prod_{f=1}^{\phi_{max}} \mathcal{N}(\phi_{n,f}|\boldsymbol{f}^i_{n-1,f}(\gamma_n), R_f + \boldsymbol{R}^i_{n-1,f}(\gamma_n)), \tag{3.24}$$

$$p(\mathcal{E}_n|e^i) = \prod_{e=1}^{\phi_{max}} \mathcal{N}(\mathcal{E}_{n,e}|\boldsymbol{e}^i_{n-1,e}(\gamma_n), R_e + \boldsymbol{R}^i_{n-1,e}(\gamma_n)), \tag{3.25}$$

where $p(\phi_n|\boldsymbol{f}^i)$ denotes the probability that the measured formant frequencies ϕ_n originated from the ith speaker conditioned on its corresponding vowel template $\boldsymbol{f}^i$. With the second formula we compute the probability $p(\mathcal{E}_n|\boldsymbol{e}^i)$ that the measured RMS energies $\mathcal{E}_n$ originate from the ith speaker conditioned on its corresponding RMS energy template $\boldsymbol{e}^i$. The term $\boldsymbol{f}(\gamma)$ indicates a subset of entries from $\boldsymbol{f}$ that correspond to the vowel γ, similarly for $\boldsymbol{e}(\gamma), \boldsymbol{R}_f(\gamma)$ and $\boldsymbol{R}_e(\gamma)$. The term ϕ_{max} denotes the number of formants (and their corresponding RMS energies) and was chosen as $\phi_{max} = 3$ (see also section 3.3.2).

We approximate the distribution on the measured quantities $\mathcal{Q}_n = [\phi_n, \mathcal{E}_n]$ conditioned on the vowel templates $\mathcal{T}^i = [\boldsymbol{f}^i, \boldsymbol{e}^i]$ corresponding to the ith speaker with the following factorial formula:

$$p(\mathcal{Q}_n|\mathcal{T}^i) \approx p(\phi_n|\boldsymbol{f}^i)p(\mathcal{E}_n|\boldsymbol{e}^i). \tag{3.26}$$

We note that formula 3.26 is an approximation since in general the measured quantities ϕ_n and $\mathcal{E}_n$ will not be independent given the vowel template $\mathcal{T}^i$. Thus for each speaker we compute $p(\mathcal{Q}_n|\mathcal{T}^i)$ we assign the measured quantities to the one speakers according to $argmax_i\, p(\mathcal{Q}_n|\mathcal{T}^i)$.

However we do not wish to assign the measured quantities to one of the speakers if the probability $p(\mathcal{Q}_n|\mathcal{T}^i)$ is not convincing. Therefore we use a threshold function to decide whether or not we assign the measured quantities to one of the speakers. Thus we assign the measured quantities to one of the speakers if and only if:

$$max\big(\alpha\, p(\mathcal{Q}_n|\mathcal{T}^i)\big) > aT, \tag{3.27}$$

where α denotes a normalization constant ensuring that the probabilities sum up to one. The term aT denotes the assignment threshold.

Method 1	$aT = 0.65$	$aT = 0.85$
Correctly classified	68 (41%)	47 (28%)
Incorrectly classified	64 (39%)	19 (12%)
Not classified	33 (20%)	99 (60%)
Method 2	$aT = 0.65$	$aT = 0.85$
Correctly classified	69 (42%)	47 (28%)
Incorrectly classified	63 (38%)	17 (10%)
Not classified	33 (20%)	101 (62%)
Method 3	$aT = 0.65$	$aT = 0.85$
Correctly classified	73 (44%)	48 (29%)
Incorrectly classified	64 (39%)	14 (8%)
Not classified	28 (17%)	103(63%)

Table 3.4: Results obtained from the different methods.

Table 3.4 presents the results obtained from the different methods. In total there were 165 segments extracted from the second part of the signal. The first number in each classification column indicates the number of segments followed by its percentage. The term *Not classified* indicates that $max\, p(\mathcal{Q}_n|\mathcal{T}^i)$ did not reach the assignment threshold aT. The first method (*Method 1*) incorporates only the measured formant frequency information for classification. Thus *Method 1* computes the probability that measured quantities originate from the nth speaker as $p(\mathcal{Q}_n|\mathcal{T}^i) \approx p(\phi_n|\boldsymbol{f}^i)$. The second method (*Method 2*) incorporates the measured formant frequencies in conjunction with the measured RMS energy values in order to compute the probability that measured quantities originate from the ith speaker. The third method (*Method 3*) incorporates the measured formant frequencies in conjunction with the measured RMS energy values in order to compute the probability that measured quantities originate from the ith speaker. The values in both dimensions are re-estimated according to the procedures as described in section 3.4.2.

From table 3.4 we see that the third method yields the best results. We cannot provide any (additional) analytical explanation for this result other than the presented analysis and assumptions [17] in section 3.4.2. Since *Method 3* provides us with the best results we use this method for our multiple target tracking algorithm.

[17]The assumptions that form the basis for the re-estimating procedures.

3.6 Multi-segment Formant Extraction

Due to the computational load of tracking the speakers with many samples, we do not use overlapping segments. We simply use non-overlapping Hamming windowed segments of $25ms$ for tracking. A drawback of this implementation is that we occasionally obtain unfortunate windowed data.

For instance: two consecutive segments containing both one half of a vowel utterance. Therefore these segments do not contain optimal vowel information. Typically no vowel information is extracted from these segments since the result obtained from the LPC analysis typically yields: $max\, p(\gamma_n|\phi_n) < vT$. Here we lose again valuable information.

In order to overcome this problem we modify the formant frequency extraction algorithm. First we set the vowel (assignment) threshold to a lower value, experimentally chosen as $vT = 0.6$. This value should not be chosen too low since we do not wish to lose (valuable) processing time and only wish to investigate potentially promising segments. If the segment reached this (lower) threshold then we apply a more extensive LPC analysis centred around the current segment, as presented in table 3.5.

For each segment:

1. Select the first windowed data for LPC analysis by taking the middle sample from the current segment as the last sample for the LPC window (the current segment is shifted back by a half window size).
2. Estimate the formant frequencies from the segment according to the scheme presented in Fig. 3.19.
3. Shift the LPC window 1ms.
4. Repeat the two previous steps until the middle sample of the current segment is first sample of the LPC window (the current segment is shifted forward by a half window size).

Table 3.5: Multi-segment shift algorithm

In practice the algorithm of table 3.5 applies additional LPC analysis on multiple 25ms windowed data segments that are centred around the middle sample from the current segment. A schematic representation of the multi-segment shifts is presented in Fig. 3.18. On each windowed data the same LPC analysis is applied as described in the sections 3.2.1 and 3.3.3. However for each method we use a different value for the vowel threshold. For the first LPC analysis we choose this value as $vT_1 = 0.6$. The segments eventually have to be verified by the formant band filter where the vowel threshold is set back to the (initial) value $vT_2 = 0.8$. Figure 3.19 presents a schematic overview of the extended formant frequency extraction algorithm for each windowed data.

Note that we can choose alternative values for the discrete time shifts. By choosing this value as 1ms we potentially obtain 50 formant frequency estimates ϕ_n for the detected vowel γ_n. In practice this number is much lower since a substantial amount of the windowed data did not reach the vT_2. We further note that, if this procedure is applied to two (or more) consecutive segments we obtain duplicate information. However this is easily avoided by appropriate implementation.

Thus we increase the amount of formant frequency estimates from the signal with the formant frequency extraction algorithm as presented in Fig. 3.11. However we choose a lower value for the vowel threshold vT. The segments that reach the this lower threshold are further anlysed by the multi-segment shift algorithm as presented in table 3.5.

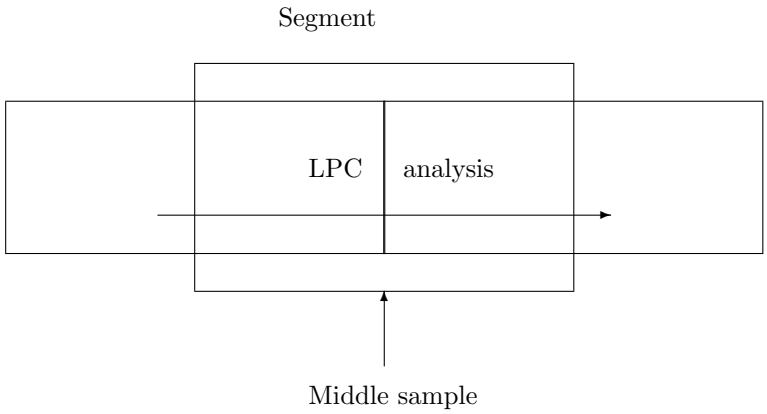

Figure 3.18: Schematic representation of the LPC window shifts within the segment.

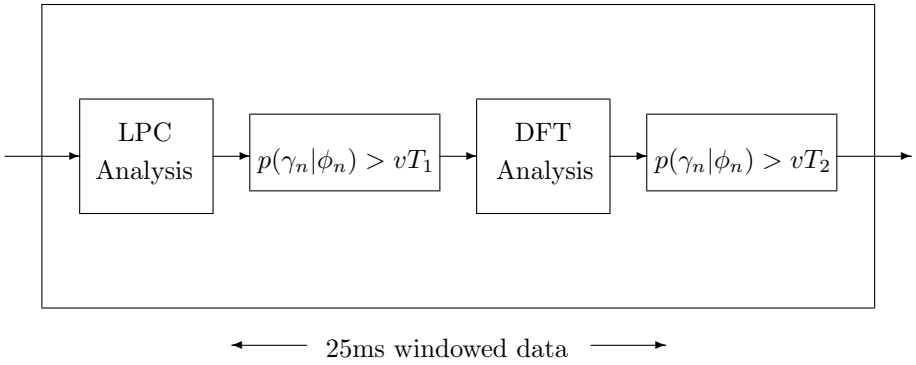

Figure 3.19: Schematic representation of formant frequency extraction for each windowed data.

Chapter 4

Multiple Target Tracking

Tracking multiple moving targets in general requires estimating the joint probabilistic distribution of the target states [29], [32]. As stated in chapter two, in practice computing the filtering distribution of the state of a single target is typically intractable[1]. Obviously, computing the filtering distribution of the states of multiple targets becomes more difficult, since the size of the state space grows exponentially with the number of targets. Furthermore, when dealing with multiple targets the data association problem arises.

We assume that the measurements originate from the targets or from clutter. The fundamental problem is to cast each measurement to one of the targets or to clutter. Typically, the origin of the measurement must be inferred, since the sensor usually does not provide any identification of the measurement source. Clutter, also referred to as a false alarm, is described with a special model whose (spatio-temporal) statistical properties differ from the targets. These differences are used in order to extract target tracks from clutter. Therefore, the main difficulty with multiple target tracking is assigning each received measurement to one of the target models. In general, methods for multiple target tracking deal with estimating the state of an unknown number of targets and in general the true target models are unknown. Thus, apart from the estimation problem the additional data association problem must be solved and these problems have to be solved jointly. We consider now the problem of tracking the locations of multiple speakers based on the azimuth angle measurements in conjunction with measured voice-features.

Brief Overview Of Existing Of Data Association Methods Several methods exists for the data association in multiple target tracking. A simple method is *nearest neighbour* (NN), where only the closest observation to any predicted state is used to perform the measurement update step. The *nearest neighbour standard filter* (NNSF) is a method based on this principle, where global optimisation is accomplished by minimizing the total observation-to-track statistical distance. The *probabilistic multiple hypothesis tracking* (PMHT) method explores simultaneously several alternative associations (hypotheses). To select the most likely hypothesis, the PMHT can use a maximum-likelihood method in conjunction with the *expectation maximization* (EM) method in order to estimate the parameters for association probabilities and state estimates. Another method that provides a solution for the data association problem is *joint probabilistic data association filter* (JPDAF)) [57], which is an

[1]The intractability depends (among others) on the assumptions about the state-space. However "real life" tracking problems are typically intractable due to the non-linear- and non-Gaussian state-space.

extension of the *probabilistic data association filter* (PDAF) in order to track multiple targets. The states of the targets are estimated by summing over the entire association hypothesis weighted by the probabilities from the likelihood.

We have implemented a sample-based version of the JPDAF method in order to track multiple targets. Therefore the JPDAF is discussed in more detail in the upcoming sections. First the background of the JPDAF is discussed, followed by a description of the general framework of the JPDAF. Next we discuss a generic sample-based version of the JPDAF as proposed by [55]. Finally, we present our model and (sample-based JPDAF) filtering algorithm for combining azimuth features with voice features.

4.1 Background

The PDAF could be described as an extension of the Kalman filter [52]. As described in section 2.2.3, the Kalman filter provides a method how to update the state of a single target given a measurement, that is: the measurement contains one observed feature at time step t. The PDAF like the Kalman filter is developed in a Bayesian framework and provides a method how to update the state of a single target given a measurement at time step t that contains more than one observed feature. The fundamental idea of the PDAF is to combine the *innovations* for each measured feature j at time step t. The combined *innovation* v is computed as the weighted sum of the individual measurement *innovation* v_j and can be generally expressed as:

$$v = \sum_{j=1}^{m_t} \beta_j v_j, \tag{4.1}$$

where m_t denotes the number of measured features at time t. Each β_j denotes the probability of the association event λ_j, that the jth observed feature within the measurement originated from the target. Thus, the measured features are weighted by the PDAF by assigning an association probability to each feature, in order to update the target state estimate. To illustrate that the PDAF is based on the Kalman equations, consider the following derivation for computing the posterior target state estimate with the PDAF. Within the PDAF framework the optimal estimate of target state x is defined as:

$$\hat{x}_t = \sum_{j=1}^{m_t} \beta_j \hat{x}_{t,j}. \tag{4.2}$$

The term $\hat{x}_{t,j}$ can be defined as a modified version of equation (2.37):

$$\hat{x}_{t,j} = \hat{x}_t^- + K_t \, \tilde{x}_{t,j}^z. \tag{4.3}$$

The matrix $K_t = P_t^- H^T (H P_t^- H^T + R)^{-1}$ denotes the Kalman gain as in equation (2.38). The term $\tilde{x}_{t,j}^z$ reflects the amount of agreement between the predicted state and the jth observed feature, and can be defined as:

$$\tilde{x}_{t,j}^z = (x_{t,j}^z - H\hat{x}_t^-). \tag{4.4}$$

Thus, the evaluation of each $\hat{x}_{t,j}$ is the Kalman filter solution for updating the predicted state $\hat{x}_t^-$ with measured feature j. The probability that none of the observed features is target-originated is denoted by β_0. All the possible association events cover all the possible

interpretations of the received data, thus:

$$\sum_{j=0}^{n} \beta_j = 1.$$

From the current target state estimate along with its uncertainty a notion, of a *validation gate* can be derived. With the assumption that the measurements are distributed according to a Gaussian distribution centred on the predicted state, a *validation gate* can be thought of as an ellipsoidal volume in the measurement space, such that the probability of a target-originated measurement appearing outside the *validation gate* is negligible. To compute each association probability β_j the PDAF assumes that the target-originated is the only persistent one in the environment. Thus, the PDAF assumes that all the measurements are caused by the target or by clutter. This assumption does not hold when multiple targets are present. Thus, a problem arises whenever a target-originated measurement falls within another target *validation gate*. This problem cannot be solved by simply running separate PDAFs for each target, because this could lead to scenario where multiple trackers are locked onto the same target.

The JPDAF [2] avoids the above mentioned scenario by sharing information among the separate PDAF trackers. Thus, information is shared to obtain more accurate association probabilities with an *exclusion principle* as the essential result. A common accepted notion of the exclusion principle postulates that two trackers cannot use the same measurement and thereby preventing two trackers locking onto the same target. In the particle filtering literature the concept of mutual exclusion is referred to as treating the joint event as a nuisance parameter [39]. The joint event is integrated out, with the result that the effects of all the joint events on the targets distribution are included. This concept is also used in the Kalman filtering framework under the name JPDAF. Thus, the conjunction of possible target-measured feature pairings forms the basis in the JPDAF framework. Such a conjunction can be thought of as a partition in the association space.

4.2 Association Space

The association space consists of all possible association events, that is: the associations between the measured features $\{0, \ldots, m_t\}$ and all possible sources $\{0, \ldots, T\}$ at time t. The event space can be partitioned such that every measured feature has a unique identified source [16]. The source can be either clutter or one of the targets. Since each target produces at most one measurement, no more than one measured feature can have the same source within such a partition. The joint association events are defined as such a partition of the event space, where each measurement is uniquely associated with a source. Let Θ_Λ denote a particular joint association event. Where the subscript Λ denotes the index of the joint association event. Λ is a list of m_t elements: $\lambda_1, \ldots, \lambda_{m_t}$. The element $\lambda_j = (ji)$ of Λ assigns measured feature j to source i. The index $i = 0$ is dedicated to model false alarm. Thus, if $\lambda_j = (j, 0)$ then the jth measured feature is associated with clutter. Usually no kinematic model is associated with false alarms. Thus, Θ_Λ is a set of pairs $(j, i) \epsilon \{0, \ldots, m_t\} \times \{0, \ldots, T\}$. These joint association events are of particular interest because their probabilities can be computed. In order to provide a solution (that is based on the PDAF) for the data association problem, the probabilities of all of the possible events for each target have to be computed. Such an event is composed of joint association events

[2] Originally developed to track manoeuvring targets based on aircrafts radar returns

and is referred here as a marginal event. Thus, a marginal event is a union of all valid joint association events where a particular target is the source of a particular measured feature in each of those joint association events. The marginal event where measured feature j originated from target i is defined as:

$$\Omega_{ji} = \cup_{\Lambda|\lambda_j=(j,i)} \Theta_\Lambda. \tag{4.5}$$

The term marginal is used because of its analogy with forming the marginal *pdf* for one variable of a multi-variable joint *pdf*.

4.3 JPDAF Framework

A key notion in the JPDAF is the marginal event. Since a marginal event is a conjunction of joint association events the entire surveillance region is used as the validation gate for each target. In order to achieve efficiency only feasible joint events are considered. A feasible joint association event satisfies two criteria:

Criterion 1 Each measured feature originates from a target or from clutter

Criterion 2 Each target produces zero or at most one feature at each time

The first criterion expresses that the associations in a joint association event is exclusive and exhaustive: $\sum_{i=0}^{T} \beta_{ji} = 1$. The second criterion implies that the number of measurements m_t may differ from the number of targets T. Furthermore, the second criterion implies that the association variables λ_j are dependent for $j \,\epsilon\, \{1, \ldots, m_t\}$.

The JPDAF provides a method to compute the probability of a marginal event. The probability for each marginal event is computed in order to prevent that multiple trackers lock onto the same target. Let $X^t = \{x_t^1, \ldots, x_t^T\}$ denote the states of the targets at time t. Let $Z(t) = \{z_{t,1}, \ldots, z_{t,m_t}\}$ denote a measurement at time t that contains the m_t measured features. Let Z^t denote the sequence of measurements up to time t. To model the event when no feature is found for a target the notation $z_{t,0}$ is used. The posterior probability β_{ji} that feature j is caused by target i at time t is computed by the JPDAF according to:

$$\beta_{ji} = \sum_{\Theta \epsilon \Omega_{ji}} p(\Theta|Z^t). \tag{4.6}$$

The probability of an individual joint association event conditioned on the measurement at time t can be derived with the use of Bayes‘ rule and the assumption that the estimation problem is Markov [56]. Thus we can compute $p(\Theta|Z^t)$ as follows:

$$\begin{aligned} p(\Theta|Z^t) &= p(\Theta|Z(t), Z^{t-1}) \qquad (4.7)\\ &= \int p(\Theta, X^t|Z(t), Z^{t-1})\, dX^t \\ &= \int p(\Theta|Z(t), Z^{t-1}, X^t)\, p(X^t|Z(t), Z^{t-1})\, dX^t \\ &\overset{Markov!}{=} \int p(\Theta|Z(t), X^t)\, p(X^t|Z^{t-1})\, dX^t \\ &\overset{Bayes!}{=} \int \eta\, p(Z(t)|\Theta, X^t)\, p(\Theta|X^t)\, p(X^t|Z^{t-1})\, dX^t \qquad (4.8) \end{aligned}$$

where $p(\Theta|X^t)$ denotes the probability of the assignment Θ conditioned on the current states of the targets. Here we make the rather strong assumption that all the assignments have the same likelihood so that this term can be approximated with a constant [56]. The term $p(Z(t)|\Theta, X^t)$ denotes the probability the measurement conditioned on the state of the targets and the specific assignment of the observed features and the targets. In order to compute this probability, the possibility has to be considered that a feature is not caused by any of the targets. The probability that an observed feature is caused by false alarm is denoted by γ and the number of false alarms present in an association event Θ is given by $(m_t - |\Theta|)$. The probability assigned to all false alarms in $Z(t)$ given Θ is denoted by $\gamma^{(m_t-|\Theta|)}$. All the remaining observed features are uniquely assigned to a target. With the assumption that the features are detected independent of each other, we can define

$$p(Z(t)|\Theta, X^t) = \gamma^{(m_t-|\Theta|)} \prod_{(j,i)\epsilon\Theta} p(z_{t,j}|x_t^i). \tag{4.9}$$

With this definition equation (4.7) becomes:

$$p(\Theta|Z^t) = \int_x \eta\, \gamma^{(m_t-|\Theta|)} \prod_{(j,i)\epsilon\Theta} p(z_{t,j}|x_t^i)\, p(x_t^i|Z^{t-1}). \tag{4.10}$$

In the prediction stage JPDAFs apply recursive Bayesian filtering to update the beliefs $p(x_t^i)$ about the individual states of the targets. By applying equation (2.23) the term for the predicted state for target i becomes:

$$p(x_t^i|Z^{t-1}) = \int p(x_t^i|x_{t-1}^i)\, p(x_{t-1}^i|Z^{t-1})\, dx_{t-1}^i. \tag{4.11}$$

In the update stage, the state of target i is corrected whenever new sensory information is available. By applying equation (2.20) the term for the updated state for target i becomes:

$$p(x_t^i|Z^t) = \eta\, p(Z(t)|x_t^i)\, p(x_t^i|Z^{t-1}). \tag{4.12}$$

The obtained features in measurement $Z(t)$ have to be associated with the targets. Since these assignments are typically unknown, the single features are integrated according to the assignment probabilities β_{ji}:

$$p(x_t^i|Z^t) = \eta \sum_{j=0}^{m_t} \beta_{ji}\, p(z_{t,j}|x_t^i)\, p(x_t^i|x_{t-1}^i), \tag{4.13}$$

where again, η is a normalization factor. Thus, we need to define the *motion model* $p(x_t^i|x_{t-1}^i)$ and the *sensor model* $p(z_{t,j}|x_t^i)$ in order to compute the belief that target i is in state x at time t.

4.4 Sample-based JPDAF

As pointed out in chapter 2, particle filtering is able to deal with multi-modality caused by noise and reverberation components. If one was to track multiple targets with a single particle filter, then each particle gives a hypothesis on the state of one of the targets. This would yield an *a posteriori* distribution of the target states, given the measurements, that is represented by a *mixture-of-Gaussian*. Where each mode of this distribution corresponds

to one of the targets. However, several problems arise with this approach. The likelihood evaluation is only possible, given the prior probabilities of all possible associations between the measurements and the targets. Even if this evaluation is possible [3] one has to deal with the occlusion problem [4].

Consider the following scenario, where occlusion can lead to the loss of one the targets. The weights that represent the state of the target that is occluded will decrease, since the (potential) measurements of the occluded target are suppressed. This increases the probability that the occluded target particle weights will be discarded during the resampling step. Thus, tracking multiple targets with a single particle filter is only feasible if all the targets are sensed at every point in time and if the measurement errors are small. Therefore, the targets are tracked independently with particle filters and to deal with the data association problem the likelihood of the measurements are evaluated with the use of the JPDAF.

Sample-based representations of the individual beliefs of the states of the targets As described in chapter 2, for tracking a single target the particle filtering method can be used to construct a sample-based representation of the filtering distribution. For tracking multiple targets, a set of particle filters connected with statistical data association is used to construct a sample-based representation of the filtering distribution [56]. As described above, the assignment probabilities are evaluated according to the probabilities of each possible association. Using JPDAF with respect to the likelihood of the measurements, the assignment probabilities β_{ji} have to be considered in the update step of the particle filter. A particle filter version of equation (4.10) is used to compute β_{ji}:

$$p(\Theta|Z^t) = \eta\,\gamma^{(m_t-|\Theta|)} \prod_{(j,i)\epsilon\Theta} \frac{1}{M}\sum_{k=1}^{M} p(z_{t,j}|x^i_{t,k}). \tag{4.14}$$

where $x^i_{t,k}$ denotes the state of target i at time t according to particle k [5]. Given the assignment probabilities, the weights of the particles can be computed according to:

$$w^i_{t,k} = \alpha \sum_{j=0}^{m_t} \beta_{ji}\, p(z_{t,j}|x^i_{t,k}). \tag{4.15}$$

where α is a normalization factor to ensure that the weights sum up to one.

Thus, each individual target is tracked with a particle filter. The prediction step of the Bayesian filtering is realized by propagating the sample set for each target according to the *motion model* $p(x^i_t|x^i_{t-1})$, similar to the single target particle filter based tracking algorithm. The update step is realized by integrating the measurement $Z(t)$ into the sample sets that are obtained in the prediction step. The likelihood function is formed with the use of the JPDAF. As stated, the assignment probabilities are evaluated according to the possible associations. The particle weights are evaluated, given these assignment probabilities. Thus, a probabilistic exclusion principle [6] is accomplished by making the particle filters dependent through the evaluation of the assignment probabilities [33].

[3] With a naive approach, one could just choose a uniform distribution

[4] Following from Gaussian motion distribution and a mixture-of-Gaussian sensor distribution, the filtered distribution takes the form of (an intractable) mixture-of-Gaussian

[5] Note that $x^i_{t,k}$ is sampled from predictive distribution.

[6] In order to ensure that each measured feature belongs to at most one target, an extra term is added to the sensor model

4.5 Model For Combining Azimuth Features With Voice Features

Our algorithm operates on fixed-length segments representing 25ms-long windows from the input signals. We will use $n = 1, 2, \ldots$ as a segment index. From every segment we will compute a set of features $\lambda_n = [\theta_n^z, \gamma_n, \mathcal{Q}_n]$, where θ_n are azimuth angle measurements, γ_n is a discrete vowel indicator and $\mathcal{Q}_n$ denotes the measured voice features: $\mathcal{Q}_n = [\phi_n, \mathcal{E}_n]$. The term ϕ_n is a vector of vowel formant frequencies and the term $\mathcal{E}_n$ is a vector with the corresponding RMS energies for each formant frequency. The collected voice features are stored in templates (vowel templates) for each speaker. These templates are build (updated) with a Kalman based filter. Since we assume that each speaker produces vowels that are approximately constant, we do not use a typical motion transition model. For the evolution matrix A from equation 2.28 we simply choose $A = I$. Furthermore, we discard matrix B from equation 2.28. In the rest of this section, we present the details of computing these features. We note that the presented model is based on the work of [35].

4.5.1 Overview

Our primary goal is association of the measured features λ_n with one of the speakers. For this purpose, we describe the ith speaker with a state variable $\boldsymbol{s}_n^i$, which summarizes the persons' location and voice properties during the nth segment. Since the state cannot be directly observed, we will consider it as a hidden (latent) random variable with a prior distribution $p(\boldsymbol{s}_0^i)$. We will also define a *sensor model*, which describes a probabilistic dependency of measurements on the state $p(\lambda_n|\boldsymbol{s}_n^i)$. Under such a framework will compute posterior state distribution $p(\boldsymbol{s}_n^i|\lambda_{1:n})$ given measurements using Bayesian filtering [27]. On the basis of this distribution we associate segments with speakers.

The state of ith person during the nth segment is described by $\boldsymbol{s}_n^i = [\boldsymbol{\alpha}_n^i, \mathcal{T}^i]$, where $\boldsymbol{\alpha}_n^i = [y_n^i, \dot{y}_n^i, x_n^i, \dot{x}_n^i]$ is a 4-dimensional vector denoting the position and speed in the usual Cartesian coordinates, and $\mathcal{T}^i$ is a the "vowel template" for the ith person. Each vowel template $\mathcal{T}^i$ contains its corresponding "formant profile" $\boldsymbol{f}^i$ and its corresponding "RMS energy profile" $\boldsymbol{e}^i$. Each profile is a collection of respectively 30 characteristic formant frequencies $\boldsymbol{f}^i = [f^i_{/\mathtt{i}/}, \ldots, f^i_{/\mathtt{R}/}]$ and their 30 corresponding RMS energies $\boldsymbol{e}^i = [e^i_{/\mathtt{i}/}, \ldots, e^i_{/\mathtt{R}/}]$ of the detected vowels. Each f^i_γ represents the three formants for the vowel γ and each e^i_γ represents the three RMS energies for the vowel γ. We assume that the profiles are constant, therefore we did not use subscript n with $\boldsymbol{f}^i$ and $\boldsymbol{e}^i$.

We set the center $(0, 0)$ of the coordinate system in the middle of the microphone pair. Note, that we cannot measure the distance between the speaker and microphones. Thus, in the (x, y) coordinates, we will be effectively estimating the ratio x/y from the azimuth data. Our choice for (x, y) coordinates, follows from the fact that we can now apply a well-behaved Langevin motion model for the speakers.

4.5.2 Prior

The prior state distribution summarizes our knowledge about states before the measurements become available. However for the formant- and RMS energy profile we set the prior to first available measurement (for each corresponding vowel). We will factorise this distribution

into a product of Gaussian (Normal) density functions

$$p(\boldsymbol{s}_0^i) = \mathcal{N}(\boldsymbol{\alpha}^i|\boldsymbol{m}_\alpha, \boldsymbol{R}_\alpha)\mathcal{N}(\boldsymbol{f}^i|\boldsymbol{m}_f, \boldsymbol{R}_f)\mathcal{N}(\boldsymbol{e}^i|\boldsymbol{m}_e, \boldsymbol{R}_e) \tag{4.16}$$

We assume that a-priori every person is standing still at the front of the robot, $\boldsymbol{m}_\alpha = [1, 0, 0, 0]$. In the experiments we have chosen a density, with diagonal covariance $\boldsymbol{R}_\alpha = \boldsymbol{I}_{4\times 4}$[7].

4.5.3 Langevin Motion Model

For the sake of completeness we briefly discuss the Langevin motion model. The location of person may change continuously. For simplicity, we assume a quasi-static location within each segment. Our segments correspond to 25ms intervals, and we do not expect the speakers to move substantially within such intervals. The motion between the segments is described as a stochastic Langevin process [62]

$$x_n = x_{n-1} + \delta\dot{x}_n, \tag{4.17}$$

$$\dot{x}_n = \mu\,\dot{x}_{n-1} + \beta\nu, \tag{4.18}$$

where $\nu \sim \mathcal{N}(0, \sigma_\nu)$ is a stochastic velocity disturbance, μ and β are coefficients (see experiments section 5.2) and δ denotes the time gap between segments. Identical equations hold for the y coordinate. We refer to appendix A for further details on the Langevin motion model. We note that the equations (4.17) and (4.18) are rewritten versions of the equations that are presented in appendix A.

4.5.4 Sensor Model

Sensor model defines a probabilistic dependency between the state of a person $\boldsymbol{s}_n^i$ and the measured quantities $\lambda_n = (\theta_n^z, \gamma_n, \mathcal{Q}_n)$. This model is the same for every person, so we omit the superscript i. The model takes the form

$$p(\lambda_n|\boldsymbol{s}_n) = p(\gamma_n)\,p(\mathcal{Q}_n|\gamma_n, \mathcal{T}^i\,p(\theta_n^z|\boldsymbol{\alpha}_n), \tag{4.19}$$

$$p(\theta_n^z|\boldsymbol{\alpha}_n) = \frac{1/K}{\sqrt{2\pi}\sigma}\sum_{k=1}^{K}\exp\frac{(\tan(\theta_{n,k}) - x_n/y_n)^2}{\sigma^2}, \tag{4.20}$$

$$p(\mathcal{Q}_n|\gamma_n, \mathcal{T}) = \mathcal{N}(\mathcal{Q}_n|\mathcal{T}(\gamma_n), \boldsymbol{R}_s(\gamma_n)) \quad \text{iff} \quad \gamma_n \neq 0, \tag{4.21}$$

where the $p(\gamma_n)$ is chosen uniform. For the position measurements, we use a mixture of Gaussian, each centred at one of the measured hypothetical azimuth angles. The number of potential locations is denoted with K and was chosen as $K = 5$ (see also section 2.3.3). The constant $1/K$ ensures a proper normalization of the mixture. For simplicity, we assume Gaussian density for the measured formants given the "true" formant profile (similar for the corresponding RMS energies). The term $\mathcal{T}(\gamma_n)$ denotes entries from $\mathcal{T}$ corresponding to vowel γ, and $\boldsymbol{R}_s$ is a (diagonal) sensor noise variance. When there was no vowel detected, i.e. $\gamma_n = 0$, we use a uniform density.

[7] In the experiments, typically the speakers are initiated in the opposite sides of the room. Thus typically the second speaker is initiated with $\boldsymbol{m}_\alpha = [-1, 0, 0, 0]$.

4.6 Filtering

In this section we describe a procedure that updates state distributions $p(\boldsymbol{s}_n^i|\lambda_{1:n})$ from a sequence measurements $\lambda_{1:n}$. These distributions represent our knowledge about the motion and formant profile of each speaker. Given a new measurement, we can compute association probabilities to find the speaker that is the most likely source of the measurement.

Similar to the single target tracking problem as described in chapter 2 we have formulated our multiple target tracking problem as a stochastic time-series process with noisy observations [27]. The interesting distributions can be computed with a recursive Bayesian filtering procedure. However, our task is an instance of a more general class of probabilistic multi-target tracking problems. As mentioned in the beginning of this chapter exact filtering for these problems is typically intractable since one has to couple state estimation with measurement-target association. Therefore we apply a well-established approximate approach within the Bayesian framework, for dealing with association uncertainty: JPDAFs.

Here, we apply the JPDAF scheme together with sample-based representation of the motion component. This component will be estimated using particle filtering as described in section 2.2.4. The particle weights will be modified in order to account for measurement-target association uncertainty, as proposed in [56]. On the other hand, the formant component will be represented with a Gaussian family. Since both the prior and the sensor models are Gaussian, this component can be seen as a linear Gaussian system with data association uncertainty [27].

4.6.1 Representation

For simplicity, the filtered distribution on the state of the ith speaker after the $n$the segment will be approximated as $p(\boldsymbol{s}_n^i|\lambda_{1:n}) \approx p(\boldsymbol{\alpha}_n^i|\lambda_{1:n})p(\mathcal{T}^i|\lambda_{1:n})$. The factorial formula is an approximation, since in general the motion component $\boldsymbol{\alpha}_n^i$ and the vowel template $\mathcal{T}$ containing the formant profile $\boldsymbol{f}^i$ and RMS energy profile $\boldsymbol{e}^i$ will not be independent given the measurements.

Given the highly non-linear and multi-modal sensor model for location measurements (4.20), we choose a particle-based representation of the motion component

$$p(\boldsymbol{\alpha}_n^i|\lambda_{1:n}) \approx \sum_{k=1}^{M} \delta(\boldsymbol{\alpha}_n^i - \boldsymbol{\alpha}_{n,k}^i), \tag{4.22}$$

where M is the number of particles (per object), $\boldsymbol{\alpha}_{n,k}^i$ is the kth particle, and $\delta()$ is a delta function.

The distribution of vowel templates will be approximately represented with a Gaussian density function

$$p(\mathcal{T}^i|\lambda_{1:n}) \approx \mathcal{N}(\mathcal{T}^i|\boldsymbol{m}_{n,T}^i, \boldsymbol{R}_{n,T}^i), \tag{4.23}$$

where $\boldsymbol{m}_{n,T}^i$ and $\boldsymbol{R}_{n,T}^i$ are the mean and covariance. The prior (4.16) and sensor (4.21) models assume diagonal covariances, therefore $\boldsymbol{R}_{n,T}^i$ will also be diagonal.

4.6.2 Algorithm

The filtering algorithm for our problem comprises three basic steps, for every segment: 1) predict the states from past data, 2) compute the association probabilities, 3) update the

states with the current measurement. Below we describe these steps in detail.

Prediction

Assume that at $n-1$ there are I speakers, and their state distributions are parameterised by $\boldsymbol{\alpha}^i_{n-1,k}$, $\boldsymbol{m}^i_{n-1}$ and $\boldsymbol{R}^i_{n-1}$. Predictive distribution for the motion component, follows from the standard particle filtering scheme (see table 2.1), where we obtain predictive particles $\hat{\boldsymbol{\alpha}}^i_{n,k}$ by sampling from the motion model conditioned on $\boldsymbol{\alpha}^i_{n-1,k}$. The formant- and RMS energy profiles are assumed constant, so we just use the current distribution:

$$p(\boldsymbol{s}^i_n|\lambda_{1:n-1}) = \sum_{k=1}^{M} \delta(\boldsymbol{\alpha}^i_n - \hat{\boldsymbol{\alpha}}^i_{n,k})\mathcal{N}(\boldsymbol{f}^i|\boldsymbol{m}^i_{n-1,f}, \boldsymbol{R}^i_{n-1,f})\mathcal{N}(\boldsymbol{e}^i|\boldsymbol{m}^i_{n-1,e}, \boldsymbol{R}^i_{n-1,e}). \quad (4.24)$$

We also predict the state of a new $(I+1)$th speaker, by setting the predictive distribution equal to the prior (4.16).

Association Events

Let $\beta_n = i$, denote the event that the ith speaker was the source of the measurement $\lambda_n = (\theta_n, \gamma_n, \mathcal{Q}_n)$. The event $\beta_n = I+1$ corresponds to a new speaker. We can compute association probabilities as

$$\beta^i_n = p(\beta_n = i) = \eta\, \vartheta_i \sum_{k=1}^{M} p(\theta_n|\hat{\boldsymbol{\alpha}}^i_{n,k}), \quad (4.25)$$

$$\vartheta_i = p(\mathcal{Q}_n|\mathcal{T}^i, R + \boldsymbol{R}^i_{n-1}(\gamma_n)), \quad (4.26)$$

where η is a constant, ensuring that $\sum_{i=1}^{I+1} \beta^i_n = 1$. The term ϑ_i indicates how the measured quantities $\mathcal{Q}_n$ (containing the formant frequency and RMS energy estimates) fit to the ith vowel template $\mathcal{T}^i$ (see also section 3.5). If there was no vowel detected $\gamma_n = 0$ we set $\eta_i = 1$.

The association probabilities allow to find the most likely speaker by taking $\text{argmax}_i p(\beta_n = i)$. We can also decide whether there is a new speaker in the environment by evaluating $p(\beta_n = I+1)$. In our implementation we decided the introduce a new speaker only if $p(\beta_n = I+1) > 0.9$.

Update

The particle filter updates the sampled-based distribution of the motion component by weighting the predictive particles. The weight of particle $\hat{\boldsymbol{\alpha}}^i_{n,k}$ is

$$w^i_{n,k} = \eta \beta^i_n p(\theta_n|\hat{\boldsymbol{\alpha}}^i_{n,k}), \quad (4.27)$$

where η is a constant ensuring that $\sum_{k=1}^{M} w^i_{n,k} = 1$. Note the term β^i_n which accounts for association uncertainty. After computing the weights, we obtain a new set of particles $\boldsymbol{\alpha}^i_{n,k}$ with the standard particle resampling as described in table 2.2.

If there was no vowel detected in the formant predictive densities do not require the update step, and are propagated unchanged. Otherwise, we update only the template of the most likely speaker $j = \text{argmax}_i p(\beta_n = i)$. Once the association has been resolved, the update of the Gaussian formant density is identical to the update step in standard linear Gaussian models [27]. We note, that updating only the most likely source is a simplification since it does not take the association ambiguity into account.

Chapter 5

Experiments

This chapter describes the experiments, where we track multiple speakers. The experiments were conducted with the aim to see whether we could successfully track the speakers whenever their paths cross. As mentioned in section 1.1, the azimuth features will not have enough resolution to distinguish between the speakers whenever they are close to each other. In order to disambiguate such difficult cases, we attempt to collect as many as possible speaker-specific voice features. We collect these voice features while the speakers are spatially separated [1].

We demonstrate our approach using a collection of stereo recordings obtained with the Philips iCat interface robot. We consider various two-speaker configurations, where the recorded signals contain sentences produced by the speakers in turns. Both speakers were male. In total, each of our signals was approximately 45s long. Allowing for a potential detection of 450 to 675 phonemes, for details see section D.1. The recordings were taken in the same rectangle-shaped office room where the single target tracking experiments were conducted. Therefore we refer to section 2.3.1 for recording condition details [2].

5.1 TDOA Estimation

First, we show how estimation of the vowel formants improves the TDOA measurements. Typically, before further processing the GCC function applies a low-pass filter (8kHz cut-off). This filter removes high-frequency noise present in the signal, and preserves the speech components, which are located in the lower-end of the spectrum. When the vowel formants are available, we can now use the knowledge about vowel spectral location to more precisely select the filter cut-off. In Fig. 5.1 and Fig. 5.2 we show the cross-correlation vs. TDOA plot. In Fig. 5.1 and Fig. 5.2 the horizontal axis represent the range of lags that are used for the TDOA estimation. The vertical axis shows for each lag the corresponding coherence energy obtained from the GCC function. In each top panel we have used a fixed low-pass filter. In each bottom panel, we present the same plot however obtained with a low-pass filter, where the cut-off was selected to be the third formant frequency. In this way we could extend the range of discarded frequencies and remove many spurious peaks in the correlation function.

[1] Spatially separated in the sense that there is a convincing margin between the two estimated azimuth angles that correspond to the different speakers.

[2] Note that the trajectory information presented in paragraph of section 2.3.1 does not hold for the multiple target tracking experiments.

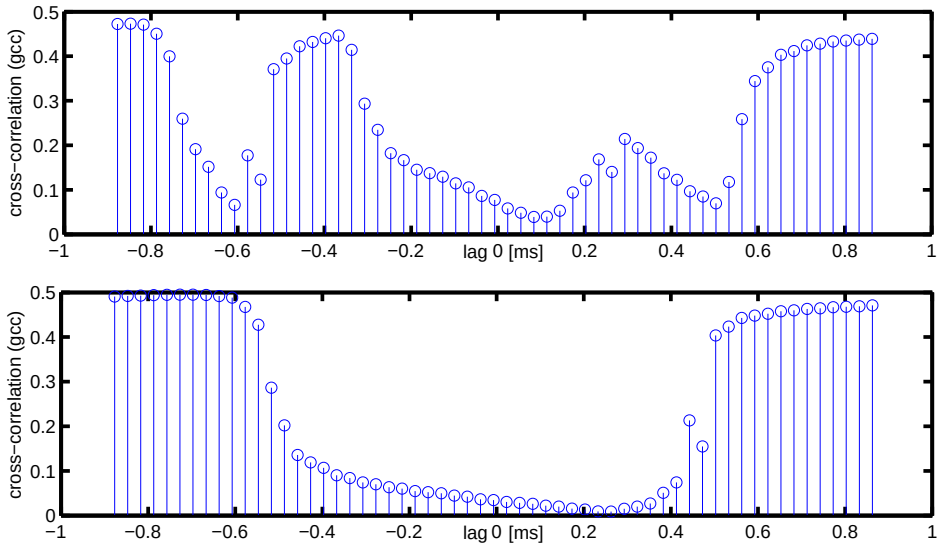

Figure 5.1: GCC analysis on a segment that contains the vowel /a/. Comparison of TDOA measurements obtained from regular GCC algorithm against measurements obtained using vowel-specific frequency range. The top panel presents the GCC algorithm results obtained with a fixed low-pass filter. The bottom panel presents the GCC algorithm results obtained with a low-pass filter, where the cut-off was selected to be the third formant frequency.

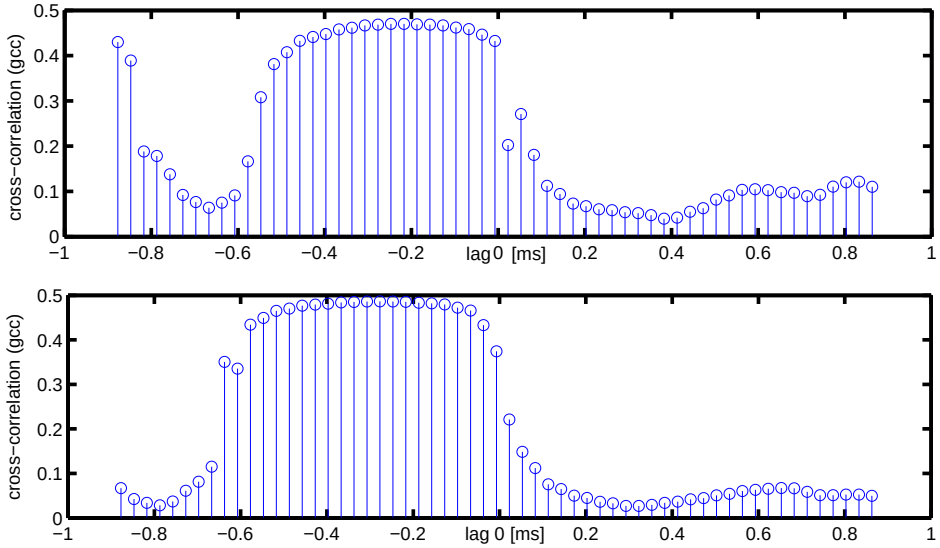

Figure 5.2: GCC analysis on a segment that contains the vowel /i/. Comparison of TDOA measurements obtained from regular GCC algorithm against measurements obtained using vowel-specific frequency range. The top panel presents the GCC algorithm results obtained with a fixed low-pass filter. The bottom panel presents the GCC algorithm results obtained with a low-pass filter, where the cut-off was selected to be the third formant frequency.

5.2 Parameters

We set the model parameters as follows: The sensor variance for formant measurements $\boldsymbol{R}_s$ is a diagonal 60×60 identity matrix, $\boldsymbol{R}_s = \boldsymbol{I}_{60\times 60}$. The parameters for the motion model (the Langevin) process were $\alpha = 0.8$, and $\beta = 0.6$. These values correspond to a human (slowly) walking in a room. For the particle filter we used $M = 100$ samples per object.

Each speaker is tracked is tracked with a particle filter in conjunction with the adaptive sigma function. However we did not apply the threshold function (see section 2.4.2) in the multiple target tracking experiments. We feel that with by applying the LP filter [3] to the signal in conjunction with the adaptive sigma function yields robust tracking results. In addition we feel that the non-dynamic character of the threshold function is a major drawback. Note that the threshold function typically needs different calibrations for different signals. Furthermore excluding measurements whenever the speakers are further away from the microphone pair is far from ideal. For the parameter setting of the adaptive sigma function we refer the reader to section 2.4.2.

In addition we found that by replacing $R_f + \boldsymbol{R}^i_{n-1,f}(\gamma_n))$ in equation (3.24) with R_f in conjunction with replacing $R_e + \boldsymbol{R}^i_{n-1,e}(\gamma_n))$ in equation (3.25) with R_e yielded more robust association assignments. Thus in our experiments we used a modified version of equation (4.26) giving: $\vartheta_i = p(\mathcal{Q}_n|\mathcal{T}^i, \boldsymbol{R})$. We believe that this is due to differences in the number of received vowel examples for each object. These differences result in different covariances for the found formant frequencies for each speaker. Since the vowel profiles are learned online, in a very limited amount of time, the differences in covariance can become considerable. By using the same covariance for each speaker, we guarantee that the speakers who talk frequently will not be favoured over the speakers who talk infrequently.

5.3 The Crossing Of Paths Experiments

In this experiments we consider two speakers who are initially positioned on the opposite sides of the microphone pair. In order to build profiles that can be compared, the speakers were requested to speak short sentences that contained the vowels mentioned in section 3.3.2. The speakers gradually move toward each other, while increasing their distance from the microphones.

As stated earlier we build the vowel templates for each speaker while they are spatially separated. In practice, if a vowel is detected then we compute the TDOA with the LP filtered segment where the cut-off frequency was set to the third formant frequency[4]. If the estimated location obtained from the localization function approximately corresponds to one of the speakers expected location, then we update the corresponding vowel template with the detected vowel. In order not to pollute the carefully build vowel templates we do not update any vowel templates when the speakers are close to each other. This in order to ensure we do not falsely assign a vowel to one of the vowel templates. Thus if the speakers are close to each other and if we detect a vowel, then we use the vowel templates in conjunction with the measured vowel in order to disambiguate. However the measured vowel is not used to update any vowel template. In Fig. 5.3 we present two results obtained from the tracking

[3] Note that we yield even more robust results by applying the LP filter to signal with the cut-off frequency set to the third formant frequency whenever we detect a vowel.

[4] Instead of the fixed cut-off frequency of 8kHz, see also section 5.1

algorithm, we present the azimuth measurements (as thin dots) and the expected locations of the target speakers (in bold line).

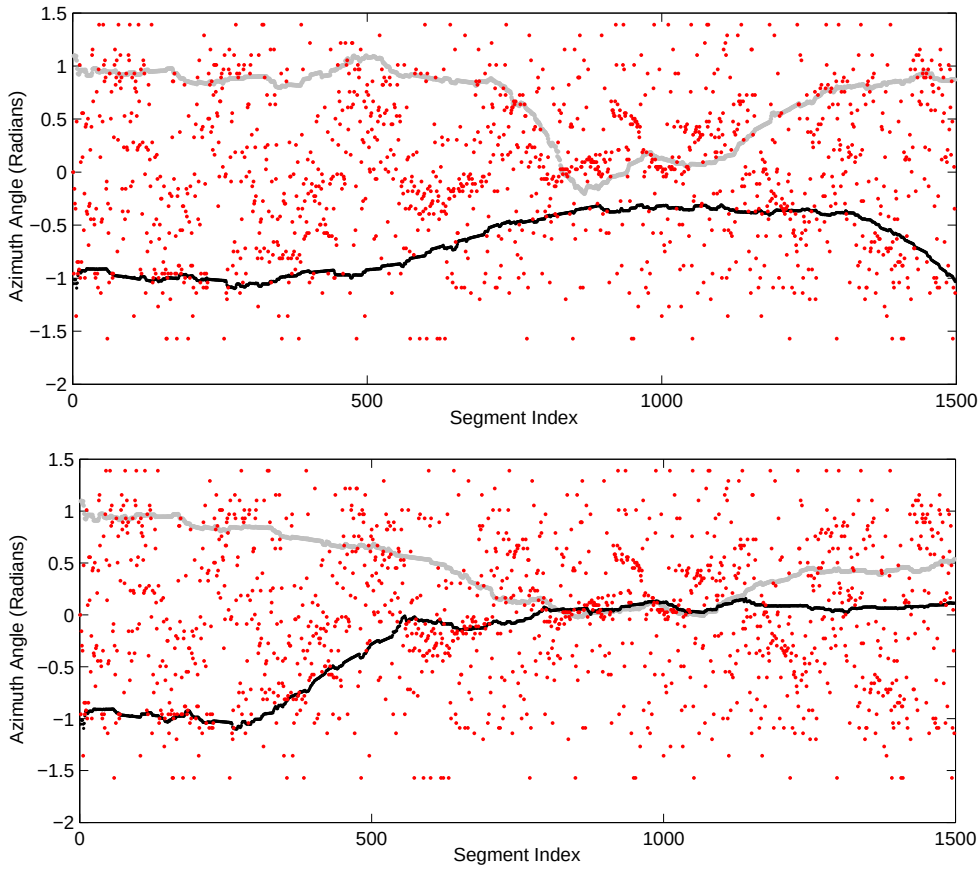

Figure 5.3: Results obtained from various runs of the filtering algorithm. The horizontal axis represents segments form the input signals. The vertical axis shows location as azimuth angles (in radians). Each dot represents a measured angle. Two bold lines give the estimated speakers' locations

From Fig. 5.3 we see that the tracking algorithm fails in maintaining the speakers in the correct paths. After analysing the data (obtained measurements for both azimuth and formant frequency estimation) we noticed that our formant frequency extraction algorithm did not detect any vowels when the speakers crossed each other paths. Therefore the tracking algorithm relies solely on azimuth information for the data association. An other problem is that the speakers were speaking (relative) long sentences. Thus the tracking algorithm does not incorporate (actual) measurement information for updating the state of the silent speaker. Obviously, the longer the speaker is silent the less accurate the corresponding estimated location becomes. This in conjunction with the strong clutter yielded poor tracking results.

In order to overcome these problems we requested the speakers to speak with shorter sentences. And in order to increase the probability that the formant frequency extraction algorithm will detect vowels and correctly associate these with the speakers when they cross each other paths, the speakers were requested to articulate their speech utterances more strongly. The result of one of these (additional) experiments is presented in Fig. 5.4.

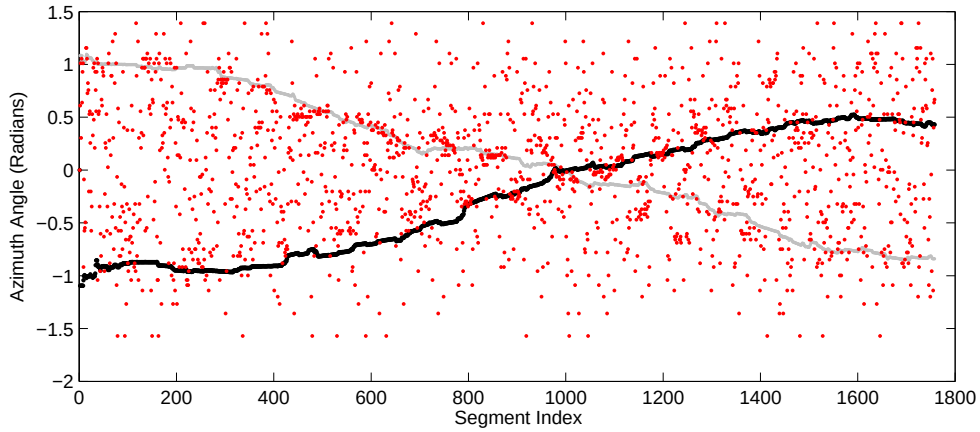

Figure 5.4: Tracking two speakers, the speakers were requested to speak with shorter sentences ($1-2$s). The horizontal axis represents segments form the input signals. The vertical axis shows location as azimuth angles (in radians). Each dot represents a measured angle. Two bold lines give the estimated speakers' locations.

Despite the strong clutter and the fact the both speakers are male, the tracker can estimate the location quite reliable for the most of the sequence. In the later part, when the targets move far away from the microphones, the trackers looses one of its targets.

The presented result in Fig. 5.4 gives a good indication of the "true" paths of the speakers. However not each run yielded the same accurate result. Various results obtained form the tracking algorithm are presented in Fig. 5.5.

Typically, the speakers are accurately tracked in the first part of the signal. Furthermore, the data association problem is solved correctly when the speakers path cross. However after the crossing, typically the tracking algorithm fails in maintaining an accurate location for each of the speakers. We believe that this is mainly caused by an increase of the speakers' pace. From Fig. 5.5 we see that after the crossing, the patterns in azimuth estimates change. This could indicate an increase in velocity for each of the speakers. We note that the speakers are remaining in position for a short period when they are in front of the microphones. This is followed by a relative rapid movement towards an outer corner by each of the speakers. Therefore we believe that this relative rapid change in velocity in conjunction with the strong clutter causes the loss of (one of) the targets.

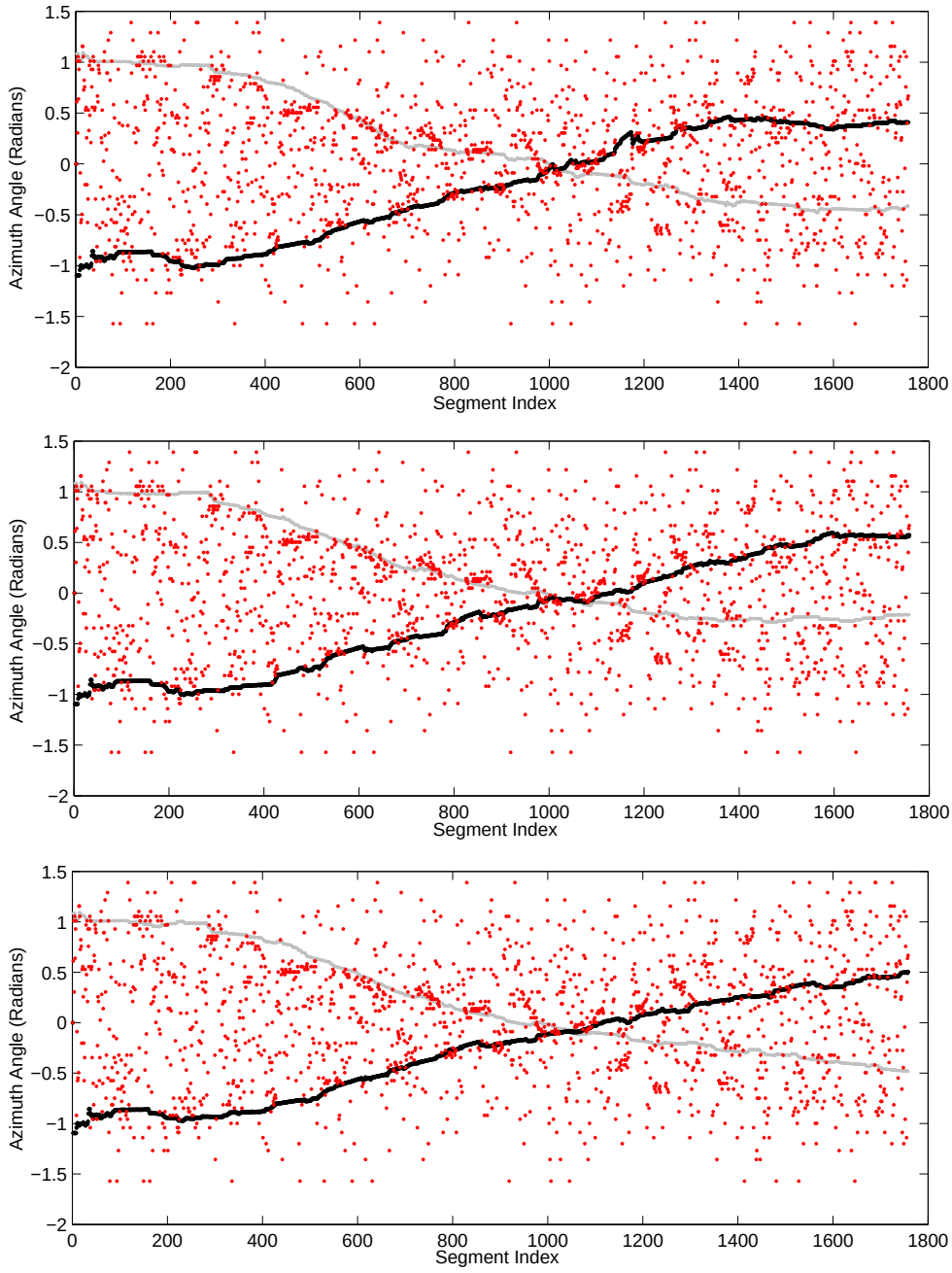

Figure 5.5: Results obtained from various runs of the filtering algorithm for tracking two speakers. The speakers were requested to speak with shorter sentences (1−2s). The horizontal axis represents segments form the input signals. The vertical axis shows location as azimuth angles (in radians). Each dot represents a measured angle. Two bold lines give the estimated speakers' locations

5.4 Comparative Experiments: Excluding Voice Features

In order to make a sensible comparison between the tracking the speakers with and without incorporating voice features we have conducted additional experiments. The filtering algorithm that is used in the additional experiments excluded all voice feature-based information. Thus no vowel templates were constructed and exploited in order to disambiguate in the "crossing scenarios". Moreover no vowel specific information was incorporated in the GCC method as described in section 5.1. Since we used a parameter setting that is equivalent to the parameter setting in the previous experiments we refer to section 5.2 for details [5].

Various results from the additional experiments are presented in Fig. 5.6. From Fig. 5.6 we see that none of the results is correct, clearly the filtering algorithm that incorporates voice feature-based information outperforms the filtering algorithm that excludes the voice feature-based information. Note that the middle panel presents a result where the tracking of the crossing of the speakers is actually correct. However in the latter part the filtering algorithm loses one of the speakers. Furthermore the filtering algorithm is not able to perform a correct and consistent speaker tracking in the crossing part (and the subsequent parts) of the recording.

We believe that, apart from insufficient information to tackle the data association problem, the filtering algorithm suffers from (more) strong clutter. We believe that the TDOA estimates improve by incorporating voice feature-based information in the GCC method (for further motivation see section 5.1). Therefore we believe that the filtering algorithm that tracks the speakers without incorporating voice feature-based information is more prone to pick up a clutter trail, compared to the filtering algorithm that incorporates voice feature-based information.

5.5 Data Associaton: A Closer Look

This section presents a closer look at the critical part of the tracking: the crossing of the speakers' path. The experiment from which various results are presented in Fig. 5.4 and Fig. 5.5, serves as a platform for the analysis in this section. Figure 5.7 presents the evolution of the particle population for each speaker in the critical part. From the figure we see that when the speakers cross each others path the azimuth angle does not provide enough resolution to distinguish between the speakers. However with the additional vowel-based information we can successfully keep track of the speakers.

Figure 5.8 presents a sequence of consecutive analysed segments: 1012, 1013 and 1014, where both speakers were located in an azimuth angle of approximately 0 radians. The segments present the particle population for each speaker. Each particle population is expressed in azimuth angles (in radians) with their respective weights. No vowel was detected in the segment presented in the top panel. Since the particle populations from both speakers overlap each other, the azimuth angle measurements do not provide enough discriminative power. In the next two segments (the middle- and bottom panel respectively) a vowel was detected. Therefore the obtained azimuth estimates can be assigned to one of the speakers more exclusively with the aid of the constructed speakers' vowel templates.

[5] Since all voice feature-based information was excluded in the experiments, the parameters for constructing the vowel templates do not apply here.

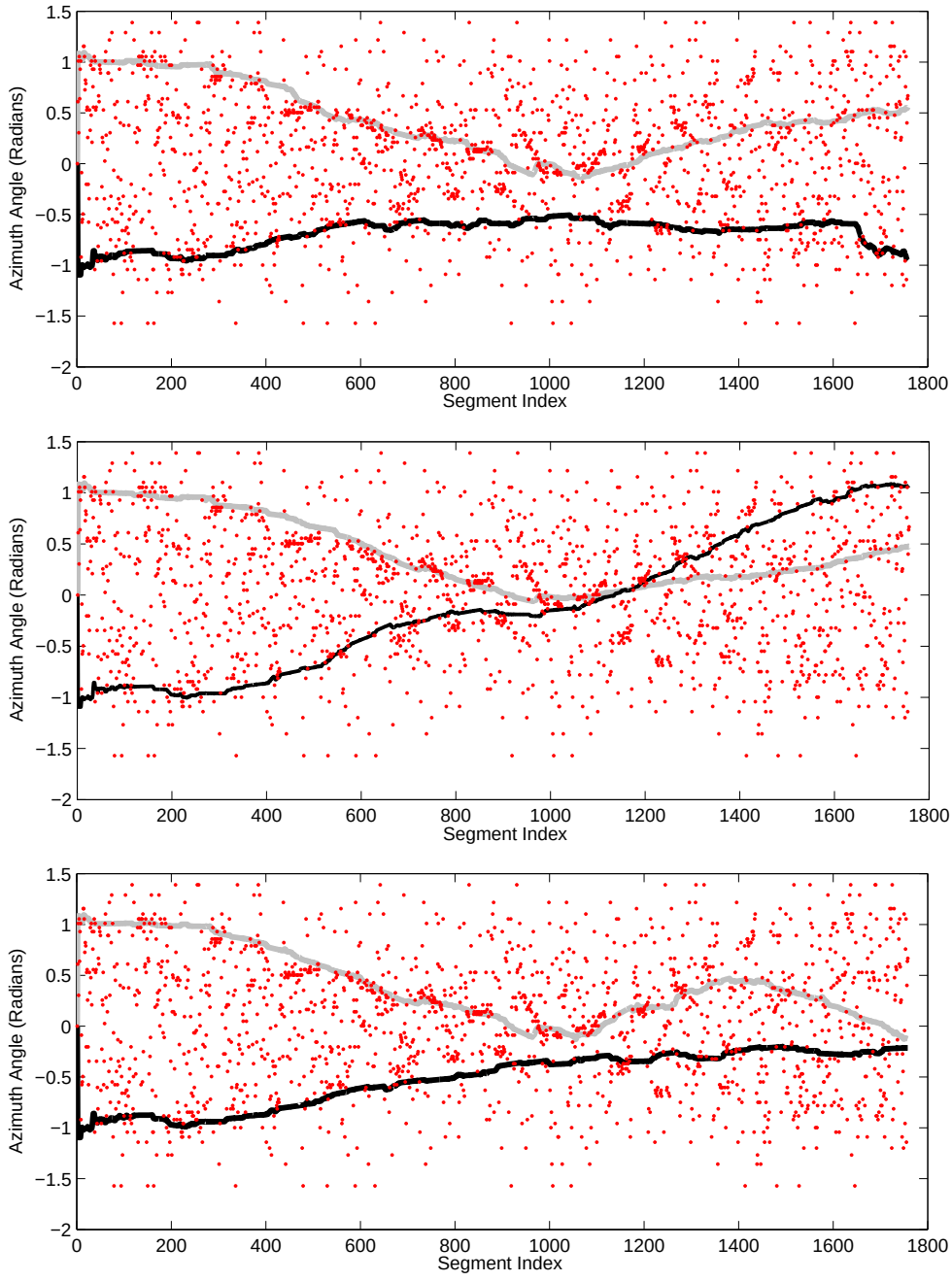

Figure 5.6: Results obtained from various runs of the filtering algorithm for tracking two speakers. The filtering algorithm excluded all vowel specific information. Thus only azimuth features are used to tackle data association problem. The horizontal axis represents segments form the input signals. The vertical axis shows location as azimuth angles (in radians). Each dot represents a measured angle. Two bold lines give the estimated speakers' locations.

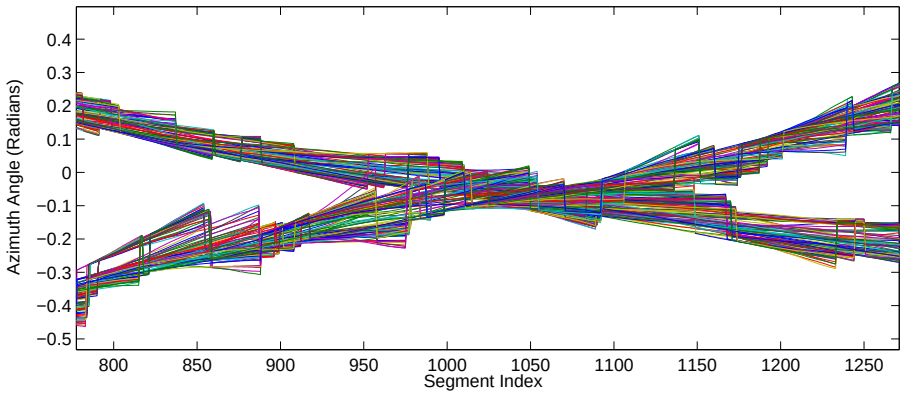

Figure 5.7: The corresponding particle populations for each speaker are plotted as they evolve in time. The vertical axis shows for each speaker the particle population in azimuth angles.

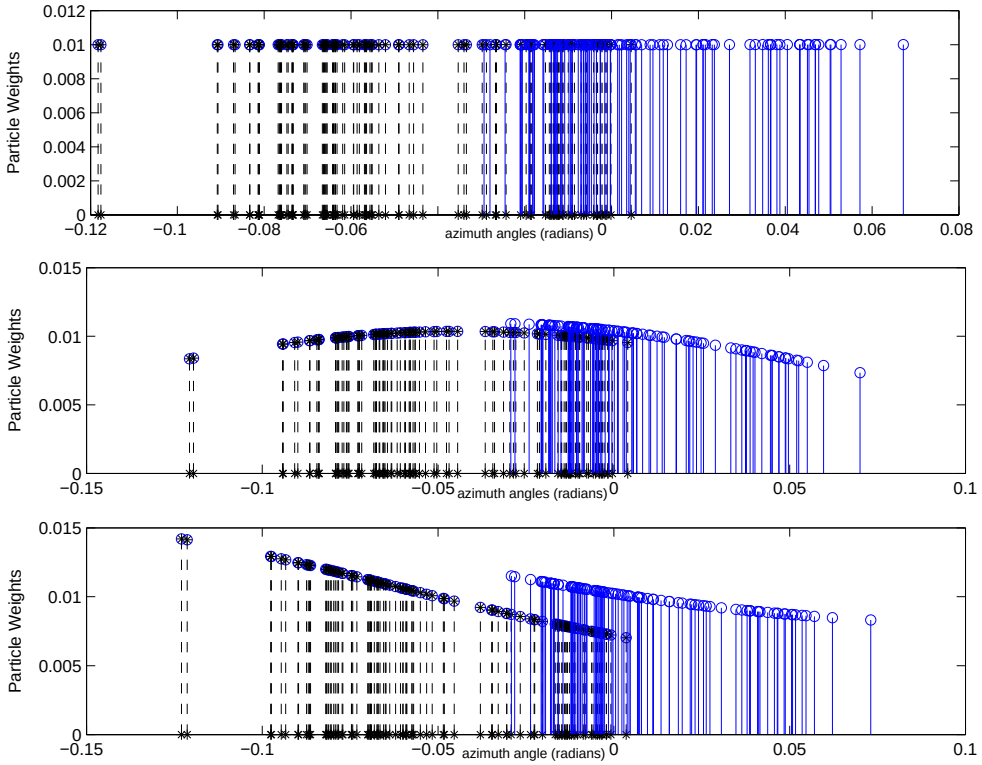

Figure 5.8: The figure shows the stochastic drift of each particle population. The horizontal axis shows the particle population for each speaker (expressed in azimuth angles). The vertical axis shows the corresponding particle weights. The first speaker is indicated with dashed lines and asterisks. The second speaker is indicated with solid lines and circles. The figure presents the results from segment 1012 (top panel) to segment 1014 (bottom panel). At segment 1013 and 1014 a vowel was detected and assigned correctly to one of the speakers.

5.6 Additional Trajectories Experiments

This section presents various additional experiments with alternative trajectories. All the experiments were conducted with the same recording conditions as presented in section 2.3.1 [6] All the experiments were conducted with a particle filter based tracking algorithm extended with an additional LP filter in conjunction with the adaptive sigma function. However no threshold function was used. For the parameter setting we refer to section 5.2 for details.

First we discuss an experiment where one speaker is in front of the microphone pair (an azimuth angle of ≈ 0 radians) and remains in this location, the other speaker is in right corner of the room (an azimuth angle of ≈ -0.9 radians) and walks towards the stationary speaker. The result is presented in Fig. 5.9.

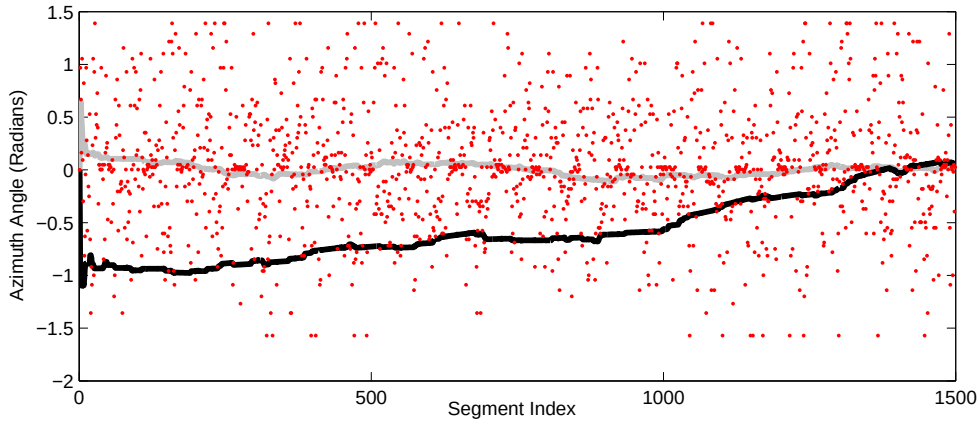

Figure 5.9: Tracking two speakers, the first speaker remains stationary throughout the recording. The second speaker gradually moves towards the stationary speaker. The horizontal axis represents segments form the input signals. The vertical axis shows location as azimuth angles (in radians). Each dot represents a measured angle. Two bold lines give the estimated speakers' locations.

From Fig. 5.9 we see that the tracker is able to accurately track both speakers for the most part of the recording. However the tracking in the final part is not accurate since the speakers are standing next to each other. Therefore they should be located in a slightly different azimuth angle.

Next we present an experiment where the speakers are in the opposite side of the room and move towards each other. However they do not cross each others path. When both speakers are close to each other (an azimuth angle of ≈ 0 radians) they move back towards their initial location. Two results of different runs are presented in Fig.5.10.

[6] Obvious the trajectory description does not hold for the additional experiments.

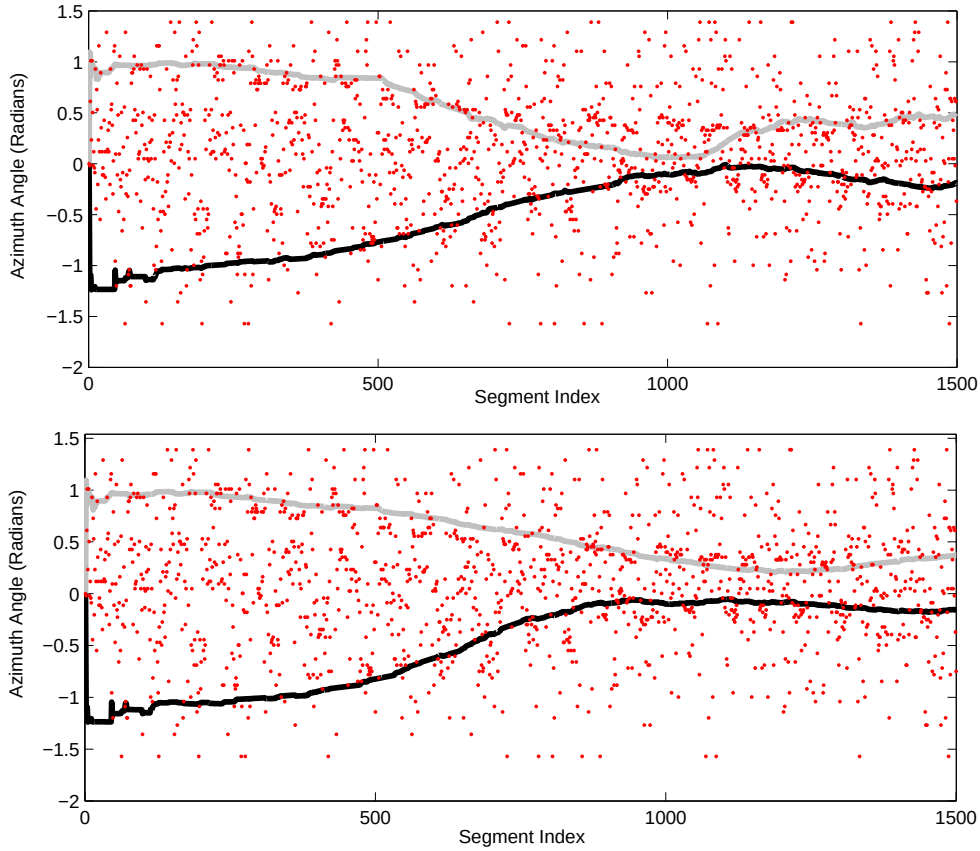

Figure 5.10: Results obtained from various runs of the tracking algorithm. The two speakers move towards each other (from the outer corners of the rooms), however they do not cross each others path. The horizontal axis represents segments form the input signals. The vertical axis shows location as azimuth angles (in radians). Each dot represents a measured angle. Two bold lines give the estimated speakers' locations

From Fig.5.10 we see that the tracker has difficulty in the latter part of the recording. Both speaker location estimates are not very accurate. We assume that our motion model has difficulty with providing accurate predictive particles in a strong cluttered environment. In addition we assume that our sensor model has difficulty in detecting a (strong) change in direction in this strong cluttered environment.

Chapter 6

Conclusions And Future Work

6.1 Concluding Remarks

We have presented a system that allows a static robot to keep track of multiple speakers in its neighbourhood. Our approach relies on two microphones, which provide azimuth angle cues about locations of the speakers around the robot. The azimuth measurements are combined with a vowel template containing a "formant profile" and "RMS energy profile" of each person. The templates represent intrinsic speaker properties, which are learned on-line. Although the presented test involved a limited number or tracked speakers, they already indicate the benefits of using vowel-intrinsic features.

We have formulated our tracking problem in a probabilistic framework where we can apply approximate filtering in order to estimate the state of each speaker. We have computed the interesting distributions with the use of a well-established approach for dealing with the data association uncertainty: JPDAFs. Since our (azimuth) tracking problem is in an environment where the state-space is non-linear and non-Gaussian we have implemented a sample-based JPDAF. By approximating the speakers location with samples we are able to deal with the non-linear and multi-modal sensor model. Note that in our tracking problem the reverberation was the main cause for the multi-modal measurement distribution.

For our tracking problem the main involved challenge emerged whenever the speakers were no longer spatially separate [1]. In order to tackle the involved data association problem we combined the azimuth features with voice features. The extracted voice features (vowel formant frequencies with their corresponding RMS energies) are stored in the vowel templates. Each vowel template corresponds to a speaker and is as mentioned constructed online.

Dealing with the reverberation was another challenge in our tracking problem. We note that we were not equipped with omni-directional microphones. Therefore the microphone pair was only (highly) sensitive in one direction[2]. Due to the above mentioned challenges the presented tracking algorithm only allows for multiple speaker tracking in a limited amount of scenarios. In our experiments the speakers were requested to cross each others path when they were close to the microphones and right in front of the microphones (approx. $1m$). This

[1] That is spatially separate in the azimuth space.

[2] In our experiments the most sensitive direction was set to an azimuth angle of 0 radians. Therefore the best results (TDOA measurements) were obtained when the speakers were in the azimuth angle of approximately 0 radians.

way we can optimally benefit from the sensing resolution of the microphone pair. We believe that we can not solve the data association problem where the speakers cross each other path further away from the microphone pair or if the crossing occurred in a non-sensitive direction. Whenever the speakers were not in this sensitive direction the influence of noise artefacts increased. We believe that we can improve the results of our tracking algorithm by increasing the sensing resolution with omni-directional microphones. Further progress can be accomplished by using a microphone array of eight or more microphones as proposed by [46].

In order to improve the azimuth estimates we have contributed the adaptive sigma algorithm. By relating the amount of coherence energy from the potential TDOA candidates obtained from the GCC function, to the variance centred on each corresponding potential azimuth angle, we obtained more robust location estimates. An elegant property of particle filtering is the incorporation of a multi-modal sensor model for localization measurements. We note that this property in conjunction with the assumptions presented in section 2.4.2 allowed us to develop the adaptive sigma function.

The transformation from TDOA to azimuth angle is an inverse problem. These inverse problems typically exhibit ill-posed behaviour. The ill-posed nature of our tracking problem manifested itself whenever the solution became multi-valued. Note that the measurements originated from the true sound source and/or by noise artefacts (reverberation). In order to suppress the multi-valued solution we have contributed a refinement in the GCC method where we exploit the vowel characteristics of speech. We simply apply a LP filter to the signal before we apply cross-correlation to the signal. We set the cut-off frequency of the LP filter to the third formant frequency (plus an offset of 100Hz) whenever we detected a vowel. Typically by applying this modified LP filter we obtained an extended range of discarded frequencies and thereby removing many spurious peaks in the correlation function. Thus we tackle the ill-posed nature of the inverse problem by suppressing the multi-valued solution. The multi-valued solution is suppressed by dynamically modifying the cut-off frequency of the LP filter according to specific vowel information, based on its spectral location.

In constructing the vowel templates we found another challenge. Note that in order to make a sensible comparison between the speakers, we need vowel templates that are characteristic for the corresponding speakers. This issue becomes more important whenever the different speakers have similar voices (as in our experiments). Due to several involved challenges as measurement noise and *prosodic features*[3], speakers with similar voices will have a potential overlap in their estimated vowel formant characteristics. These formant characteristics include formant- frequencies, spectral magnitude and bandwidths.

We feel that in order to construct a characteristic vowel template, we need (several) reliable measurements. Thus this implies that for each vowel that is to be compared, we need for each speaker several reliable vowel examples (measurements). Therefore we instructed the speakers to speak with short sentences (in order to obtain a sufficient amount of vowel examples) and to more or less repeat each other (in order to construct approximately equally balanced vowel templates) [4]. In order to ensure that only reliable measurements are used to construct the vowel templates, we have contributed a formant frequency extraction algorithm

[3]We refer the reader to section D.2 for details.

[4]We note that in the experiments the involved conversations were rather artificial due to the above mentioned reasons. However by extending our vowel template (increasing the number of vowels that can be compared) or by extracting additional voice features the experiment conversations could gradually shift towards more natural conversations.

that consists of established methods in conjunction with a contributed method. To increase the amount of vowel examples for the templates we contribute the expanding window LPC analysis and the multi-segment formant extraction. In addition, aiming to increase the discriminative power of the vowel templates, we contributed a refinement to the voice feature estimation methods: the formant re-estimation procedures.

We believe that with the above described (potential) extensions for both hard- and software we can equip the interface robot with a speaker tracking mechanism allowing for tracking the speakers in more difficult and elaborate scenarios.

6.2 Future Work

The presented ideas can be extended to more elaborate scenarios, where the audio cues are used jointly with visual sensors. As an example, robot could use the audio signals to steer the camera toward the current speaker. Alternatively, in limited closed areas, the audio feature could help keep track of a person who disappears from robots field of view [13].

The tracking problem can be roughly divided into two sub-problems. Estimating the azimuth angle and estimating the formant frequencies. Both problems are approached in a straight forward manner. In order to create a more robust tracking module, one can incorporate additional sub-modules that are closer to human auditory mechanisms. With respect to tracking the azimuth angle one can incorporate a module that exploits information on the precedence effect (PE). The PE is referred to as the observation that two sounds occurring in rapid succession are perceived as a single auditory object localized near the leading sound [37]. A substantial amount of our azimuth tracking results suffered from noise artefacts caused by reverberation. We believe that we could suppress the influence of these noise artefacts further by appropriate modelling of the TDOA data with respect to the PE.

During the experiments we noticed that the reverberation also has a considerable influence on the formant frequency estimates. Since the input for the formant frequency extraction algorithm is one-channel data, the channel choice is important. If one is estimating the formant frequencies for the speaker that is closer to the left microphone than to the right microphone pair, one should analyse the data coming from the left channel. Note that the closer the sound source is to the microphone the less the influence of potential corruption caused by various noise artefacts becomes. Therefore by appropriate switching between the channels, the amount of corrupted input data for the formant frequency extraction algorithm can be reduced.

Appropriate switching can be achieved by analysing the signals on *interaural intensity differences* (IID) [45]. As mentioned in the first chapter the human auditory system finds spatial information (amongst other inferences) by acoustic shadowing. Note that if the sound source is to the left side of the human listener, then the right ear receives auditory information that is "shadowed" by the head. Therefore the human auditory system perceives a difference in loudness. Inferences from this difference contribute to the ability to localize sound sources. Note that a module that incorporates information based on IID is complementary to the GCC method. With such a module one cannot determine a specific direction (azimuth angle). However such a module could be a potential aid for making the appropriate channel choice for the formant frequency extraction algorithm. The GCC occasionally fails in estimating the correct TDOA (and therefore we obtain an incorrect azimuth angle) and

thereby providing us with a (potential) incorrect channel choice. A complementary module that incorporates IID based information allows for more robust channel choices.

In addition the azimuth tracking mechanism could also benefit from such a module. By simply regarding the TDOA estimates that are not consistent with the IID information as less reliable or even excluding them (as described in section 2.4.2) for further analysis.

The second sub-problem, formant frequency estimation is approached with straight forward signal processing methods. Though in practice, we obtained acceptable results, it is highly unlikely that this approach approximates the sophisticated human vowel detection and identification process. A more sophisticated method for speech recognition can be accomplished with the use of dynamic Bayesian networks (DBNs) [67]. DBNs can be used to represent complex stochastic processes. In the context of speech modelling, the DBNs provide us a convenient method that maintain an explicit representation of the lips, tongue, jaw and other speech articulators as they change over time. With such sophisticated modelling we can expect that the generated speech is more accurately modelled. Furthermore the DBNs are able to model correlations among multiple acoustic features and thereby enabling us to extract more robust voice features.

In order to distinguish speakers from each other on basis of speech utterances the formant frequency extraction algorithm could be extended with the extraction of additional features. An obvious choice would be extracting consonant based information (formant frequency estimation), a less obvious choice is extracting rhythm patterns from the speech [25]. These rhythm patterns are based on silence cues and could complement the formant frequencies patterns estimated from vowel and consonant utterances for speaker identification.

As stressed in earlier chapters each speaker produces unique sounding speech. Elaborating on the unique character of each speaker we believe that each speaker produces speech from a unique vocabulary. We believe that typically each speaker uses a (different) biased set of words. These biased set of words will especially manifest in natural communication between speakers. Therefore we believe that by incorporating probabilistic models with respect to choice of words can improve the discriminative power of the speakers profiles. We strongly note that such a module should be complementary of nature. By no means should the speakers intrinsic vocabulary features solely form the basis for data association.

Finally, we feel that both sub-problems could benefit from the CASA research by exploiting their *localization models* and *sound source seperation models*. Where the latter model allows for scenarios where the speakers (occasionally) speak simultaneously. This issue becomes important whenever the tracking experiments gradually move towards more "real life" scenarios[5].

[5]Note that in our experiments we requested the speakers to speak in turns.

Appendix A

Langevin Model

There are several models that describe the (typical) dynamics of a person moving in a room [30]. The Langevin model can be used to represent the time-varying locations of a speaker. The Langevin equations are commonly used in the research field physics and chemistry to describe dynamic processes with stochastic excitation forces [26] [63]. The Langevin model is reasonably simple, but has been shown to work well in practice [64]. In the context of the moving speaker, the velocity is prescribed as a random process, with properties fulfilling the theoretical dynamic hypothesis. The source state is assumed to follow a first-order Markov process specified by:

$$p(\boldsymbol{\alpha_t}|\boldsymbol{\alpha_{t-1}}) = p(x_t|x_{t-1},\dot{x}_t)p(\dot{x}_t|\dot{x}_{t-1},i_t) \times p(y_t|y_{t-1},\dot{y}_t)p(\dot{y}_t|\dot{y}_{t-1},i_t)p(i_t|i_{t-1}), \qquad \text{(A.1)}$$

where $\boldsymbol{\alpha_t}$ denotes the source state at discrete time t and is defined as: $\boldsymbol{\alpha_t} \doteq (x_t, \dot{x}_t, y_t, \dot{y}_t, i_t)$ and where (x_t, y_t) and $(\dot{x}_t, \dot{y}_t)$ denotes the source position in the Cartesian coordinate system and velocity respectively. The variable i_t denotes whether the source is in motion ($i_t = 1$) or stationary ($i_t = 0$). Thus, the Cartesian coordinates in the source motion model are assumed to be independent. This is rather a strong assumption, but was found to work well in practice [62]. The model that is used for the dynamics is based on that of source subjected to excitation and frictional forces. When the source is in motion the excitation force is assumed to be an *i.i.d.* zero-mean Gaussian random variable with variance σ_x^2. The frictional force causes the motion to seize whenever the excitation force is removed. In the x -coordinate the motion is given by:

$$\begin{aligned} x_t &= x_{t-1} + \delta T\, \dot{x}_t, && \text{(A.2)} \\ \dot{x}_t &= a_t\, \dot{x}_{t-1} + b_t\, F_x, && \text{(A.3)} \\ a_t &= e^{-\beta_x \delta T}, && \text{(A.4)} \\ b_t &= v_x \sqrt{1 - a_t^2}, && \text{(A.5)} \end{aligned}$$

where F_x denotes the excitation force and is given by $F_k \sim \mathcal{N}(0, \sigma_x^2)$, δT denotes the discrete time step separating two location estimates and is given by $\delta T = L/F_s$ with L being the frame length in samples and F_s denoting the sampling frequency. The term v_x denotes an additional source velocity parameter and β_x denotes the frictional force, the suggested values for these parameters by [62] are $\sigma_x = 5ms^{-2}$, $v_x = 1ms^{-1}$ and $\beta_x = 10s^{-1}$. The dynamics and parameters in the other Cartesian dimension is assumed to be identical.

Appendix B

Likelihood Function For Particle Filter

The uncertainty of the targets state is represented by M particles, as mentioned in section 2.2.4 the particles are hypotheses about the evolution of the targets state. Thus, the k^{th} particle consists of a hypothesis for the state sequence $\boldsymbol{\alpha}_{1:t,k}$, and an associated weight, $w_{t,k}$, consistent with the history of measurements: $z_{1:t} = 1, \ldots, t$. Ideally, the particles are sampled from the posterior. However, one is typically unable to draw the particles from the posterior distribution (see also section 2.2.1). Therefore, the state sequence is extended at each iteration, with the use of a proposal distribution: $q(\boldsymbol{\alpha}_t|\boldsymbol{\alpha}_{t-1}, z_t)$. Typically, this proposal distribution takes a convenient form, in the sense that each particle k is sampled from a distribution that is based on only the hypothesis on the current state, giving:

$$\boldsymbol{\alpha}_{t,k} \sim q(\boldsymbol{\alpha}_{t,k}|\boldsymbol{\alpha}_{t-1,k}, z_t). \tag{B.1}$$

To compensate for the disparity between the proposal and posterior distribution the particle weights are updated according to:

$$w_{t,k} \propto w_{t-1,k} \frac{p(z_t|\boldsymbol{\alpha}_{t,k})\, p(\boldsymbol{\alpha}_{t,k}|\boldsymbol{\alpha}_{t-1,k})}{q(\boldsymbol{\alpha}_{t,k}|\boldsymbol{\alpha}_{t-1,k}, z_t)}. \tag{B.2}$$

Updating the particle weights according to equation(B.2) leads to a considerable computational load. Note that a trade-off issue arises here. If one assumes that particles sampled from such computational expensive proposals produce more accurate results compared to particles sampled from relative simple proposals, then by sampling from such sophisticated proposals, one can do with fewer particles. This will reduce the overall computational cost. Usually, the choice for the proposal depends on the application/environment. However, the prior is commonly used as the proposal, in order to simplify equation(B.2) considerably. Thus, with:

$$q(\boldsymbol{\alpha}_{t,k}|\boldsymbol{\alpha}_{t-1,k}, z_t) \doteq p(\boldsymbol{\alpha}_{t,k}|\boldsymbol{\alpha}_{t-1,k}), \tag{B.3}$$

equation(B.2) becomes:

$$w_{t,k} \propto w_{t-1,k}\, p(z_t|\boldsymbol{\alpha}_{t,k}). \tag{B.4}$$

Thus, we need to form the likelihood function $p(z_t|\boldsymbol{\alpha}_{t,k})$ in order to update the particle weights.

This paragraph discusses the likelihood function $p(\theta^z|\theta_\alpha) = F(\theta^z|\theta_\alpha)$ that is used in the experiments, and therefore considers the particles and measurements in the context of azimuth angles. For a given source state α, the aim is to develop a likelihood model based on the TDOA measurements. The likelihood function should be chosen with respect to fact that peaks in the localization function correspond to likely source locations. Furthermore, peak positions may also have slight errors due to sensor calibration errors. Thus we assume that one of the peaks in the localization function is caused by the true sound source corrupted by some additive Gaussian noise. However, the likelihood function should also take notice of the possibility that there occasionally is no peak in the localization function corresponding to the true sound source (such as when the source is silent). Therefore we make an additional assumption that occasionally none of the peaks in the localization function is caused by the true sound source. With these assumptions we can form the Gaussian likelihood function:

$$F(\theta^z, \theta_\alpha) = \sum_{p=1}^{K} q_p \mathcal{N}(\theta^{z(p)}|\theta_\alpha, \sigma^2) + q_0, \tag{B.5}$$

where $p = 1 \dots K$ corresponds to the potential source locations obtained from the localization function $\boldsymbol{f}(\theta, TDOA)$ and where $q_p\,(< 1)$ denotes the prior probability that location $\theta^{z(p)}$ is the true source location. Without prior knowledge of the likely source locations, a typical choice would be:

$$q_p = \frac{1 - q_0}{K}, \quad p = 1, \dots, K, \tag{B.6}$$

where $q_0\,(< 1)$ denotes the prior probability that none of the potential locations is due to the true source location. The peaks in the localization function are treated as being equally likely with the Gaussian likelihood. One alternative is, treating larger peaks as more likely to originate from the true sound source. However, one should be cautious in order not to exclude or suppress the true sound source peak with this implicit weighting.

Appendix C

Non-linear And Ill-posed problems

In a famous paper by Jacques Hadamard published in 1902 discusses the notion of a well-posed problem [28]. The same author argued in an earlier paper published in 1901 that well-posed problems are physically important problems are both solvable and uniquely solvable. In the same paper he gave examples of problems that are not well posed and claimed that they are not physically important.

Nowadays, the definition of a well-posed problem is that it is uniquely solvable and is such that the solution depends in a continuous way on the data. This in contrast to ill-posed problems where the solution depends in a discontinuous way on the data. When the solution becomes unstable then small changes in the data will have large effects on the estimate. Thus, small errors such as rounding off errors, measurement errors, or errors caused by noise can cause large deviations. Today, the notion that ill-posed problems are not important is not a broad shared view. In fact, [34] claims that any measurement, except the trivial ones, gives rise to an inverse problem that is ill-posed.

Definition: Well-posed Problem Let H_1 and H_2 be two normed sets. Let T be some linear operator from H_1 to H_2. Problem $\mathcal{P}$: given T and $y \, \epsilon \, H_2$, find $f \, \epsilon \, H_1$ such that $Tf = y$

The problem $\mathcal{P}$ is well-posed if its solution:

- exists
- is unique
- is stable ($||f - f_t|| \leq C||y - y_t||$) - the solution depends continuously on the data

Definition: Ill-posed problem Problem $\mathcal{P}$ is ill-posed if its solution violates one of the above requirements [12].

Inverse Map From TDOA Estimate To Azimuth Angle With Hadamards notion of ill-posedness, many inverse problems exhibit such behaviour, as their results are sensitive to noise in the data. Consider the inverse map from TDOA to azimuth: the received data may originate from the "true" sound source, or by any of its "images" induced by the scattering

surfaces [1]. Thus, an ill-posed element is introduced, with the solution becoming multi-valued [2]. Furthermore, the third constraint is violated. Consider the following scenario: suppose at time t our TDOA estimate is 0.4ms and our system configuration is as follows: $d = 2a = 30$cm with $cs = 342$ m/s, where d denotes the distance between the microphones and where cs denotes the sound wave propagation. From equation (2.2) we can write:

$$\theta = arcsin\left(\frac{\tau c}{2a}\right) \tag{C.1}$$

With the above specified parameter settings, the measured azimuth angle θ^z becomes: 0.4735 (in radians). Suppose now that we have made an calibration error, with respect to the distance parameter. In addition suppose that the actual distance between the microphones is 27cm and suppose that our estimated TDOA corresponds to the true TDOA. Thus, we have a calibration error of 10% and thus the actual azimuth angle becomes: 0.5313. Consequently, the calibration error of 10% produced an error of 12.2080% with respect to our estimated azimuth angle θ^z. The effect of the ill-posed nature of the inverse map increases with the distance between the speaker and the microphone pair, as we can see from Fig. C.1.

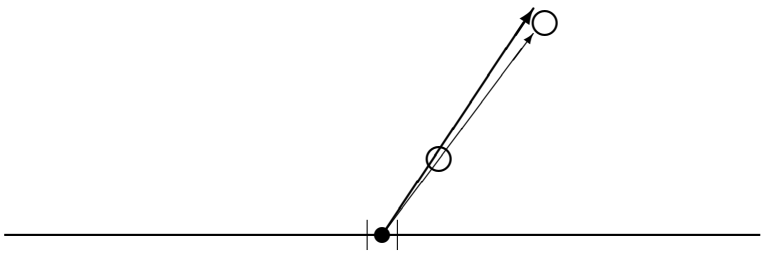

Figure C.1: Illustration Ill-posed nature Azimuth transform

In Fig. C.1 the circles represent the active speaker at different locations. The dot on the baseline represents the geometrical centre of the microphone pair. The thick and longer vector denotes the estimated azimuth direction. Where the thinner and shorter vector denotes the actual azimuth direction. Despite the estimation error, the estimated azimuth direction will still strike the speaker [3] as long as the distance between the speaker and microphone pair remains rather small. However, if the active speaker will move backwards in the actual azimuth direction, then up to some point, the estimated azimuth direction will not strike the speaker.

No straight forward solutions exist in solving these inverse problems that exhibit ill-posed behaviour. However, recent work in developing algorithms for these problems, include all available a priori knowledge in the formulation [20]. The idea is that improved modelling, by including a priori information, will suppress the potential sources causing ill-posed behaviour. Returning to our inverse problem of estimating source location θ from TDOA estimates. We attempted to suppress to effects of noise in the data, by including specific speech extracted information in our source localization algorithm. Therefore, we included vowel formant frequencies information in the GCC method, aiming to reduce the typical multi-valued nature of the solution (see also sections 2.1.4 and 5.1).

[1] As decribed in section 2.1.2

[2] Note, that the first constraint is violated.

[3] Note, that in order to establish a more natural human-robot interaction as mentioned in chapter 1, one of our objectives is to turn the robots head towards the active speaker.

Appendix D

Human Speech Production

Speech sounds production originates with vibration of the vocal cords or by constricting the air flow. The air forced from the lungs is submitted to a filtering process in the vocal tract, and finally radiated through the lips or the nose. The principle parts of the vocal tract consists are the larynx and vocal cords, pharynx, tongue, lips, teeth, nasal- and oral cavity [17]. The oral cavity or mouth, is the part of the vocal tract that vary the most in size and shape. This flexibility makes the oral cavity probably the most important single part of the vocal tract. The flexibility is due to the adjustable nature of the organs that are part of the oral cavity. The tongue can move its tip and edges independently. Furthermore, the tongue can be moved forward and backward, up and down. In addition to the freely movable tongue, adjusting the relative positions of the soft palate, the lips, the teeth and cheeks, change the size and shape of the oral cavity and therefore its acoustics. The produced sounds are radiated through the opening of the mouth, its size and shape are controlled by the lips. When the radiated sound contains wavelengths that are rather large compared to mouth opening, the radiation efficiency is decreased. If the wavelengths approaches the size of the opening, then the radiation efficiency is increased. Thus the mouth radiates effectively at higher frequencies, where a rise of 6 dB per octave is a good approximation of this effect.

D.1 Phonemes

One of the phenomenal abilities of the human sensing system is recognizing the sounds of language. Humans can recognize over thirty *phonemes* per second. Although this number is based on rate of 400 words per minute and assuming that in each word five phonemes are present; still in normal conversation the ability is required to recognize ten to fifteen phonemes per second. Phonemes, or the articulation of individual speech sounds, can be roughly divided into two groups: vowels and consonants. Since vowel sounds are produced with the vocal cords in vibration, they are referred to as always being voiced. Consonants sounds however, are not always voiced. Despite the fact that consonants are more independent of language and dialect compared to vowel sounds. Due to this property the consonant is a desirable voice feature to extract, if one was to detect sounds of language without having *a priori* knowledge of the language being spoken. However speech recognition tends to focus more on vowel extracted information. Typically the production consonants involves very rapid, sometimes subtle changes in sound. Therefore, the consonants are more difficult to analyse and describe acoustically compared to the vowel sounds. Phonemes classified as being vowel sounds are more or less steady in duration. Furthermore, they are rich(er) of harmonics. This

allows for more robust distinction between (coloured) noise and vowel sounds, compared to distinguishing between (coloured) noise and consonants.

D.2 Prosodic Features Of Speech

Conveying meaning, emphasis and emotion without actually changing the phonemes are called *prosodic features*. These acoustic patterns of prosodic features manifest in systematic changes in duration, pitch, intensity, rhythm, accent and spectral patterns of individual phonemes. Their importance, in the context of communicating information, is language dependent. In general, prosodic features give an indication of the state of the speaker. For example: by adding stress to speech (increasing the pitch and loudness of the voice) one tends to indicate anger, whereas increasing the rate of speaking tends to indicate excitement. If one stresses a voiced syllable, then this will give to a glottal source spectrum with an increased high frequency content relative to the lower frequency content. Thus, stressed vowels will contain higher magnitudes for the higher-frequency formants compared to the lower-frequency formants, than unstressed vowels.

D.3 Pitch Extraction

In practice, extracting the pitch[1] or fundamental frequency (f_0) typically involves estimating the lowest frequency, or *partial*, that relates well to most of the other partials. If the waveform is periodic, such as a vowel utterance, these partials are harmonically related [2]. Typically the frequency of the lowest partial of the waveform is referred to as the f_0 [24].

There are different theories on how the human auditory system perceives pitch. In general the perception of pitch becomes more distinct whenever the relation between the partials becomes more harmonic. A spectrum that is rich of harmonics tends to reinforce the perception of pitch. Various approaches are known for pitch extraction such as Cepstrum analysis, component frequency ratios and by auditory modelling techniques [14]. A basic approach in extracting the pitch from speech is to find the waveform that represents the change in air pressure over time. From this waveform the f_0 is estimated. Estimating the f_0 in the time-domain one attempts to find the period of the waveform (assuming that the waveform is periodic). The period of the waveform is inversely related to the fundamental frequency.

Aiming to find the period of the waveform one can apply autocorrelation to the waveform. At zero lag the correlation is maximum. If the waveform is periodic and the lag is shifted towards half of the period of the waveform, then correlation reaches a minimum. If the lag is further shifted to the length of one period the correlation reaches again a maximum. This maximum (after the zero lag maximum) indicates the period of the waveform. Note that problems arise whenever the waveform is quasi periodic and harmonically complex of nature with the autocorrelation method. Typically the first (smaller) maximum does not correspond to the period of the waveform.

[1] Here we use the terms pitch and fundamental frequency interchangeably despite the (subtle) differences in psychoacoustical and physical meaning. Psycho-acousticians use the term pitch to denote the perceived fundamental frequency of a sound (independent to whether or not this sound is actually present in the waveform). For example speech that is transmitted through phone lines typically are band limited to approximately $300 - 3000$Hz. Therefore the fundamental frequency of the waveform is at least 300Hz. However if the transmitted waveform contains a typical male voice, then the perceived pitch will be lower than the f_0 of 300Hz.

[2] Thus, the frequency of most of the partials are related to the f_0 by a small whole-number ratio.

Bibliography

[1] Arras, K.O., Philippsen, R., Tomatis, N., de Battista, M., Schilt, M. and Siegwart, R. A Navigation Framework for Multiple Mobile Robots and its Application at the *Expo.02 Exhibition", in Proceedings of the IEEE International Conference on Robotics and Automation*, (2003).

[2] H. Asoh, N. Vlassis, Y. Motomura, F. Asano, I. Hara, S. Hayamizu, K. Itou, T. Kurita, T. Matsui, R. Bunschoten, and Ben Kröse. Jijo-2: An office robot that communicates and learns. *IEEE Intelligent Systems*, 16(5):46–55, Sep/Oct 2001.

[3] N.P Basse, S. Zoletnik, M. Saffman, W. Svendsen, G. Kocsis and M. Endler Two-point Correlation Measurements of Density Fluctuations in the W7-as Stellarator *12th International Stellarator Workshop*, Madison, 1999

[4] D. Bechler and K. Kroschel Confidence Scoring of Time Difference of Arrival Estimation for Speaker Localization with Microphones Arrays *13. Konferenz "Elektronische Sprachsignalverarbeitung (ESSV)"*, Sept. 2002, Dresden, Germany

[5] E. Ben-Reuven and Y. Singer Discriminative Binaural Sound Localization *In S. Becker, S. Thrun and K. Obermayer (Eds.), Advances in Neural Information Processing Systems 15* (pp. 1229-1236). Cambridge, MA: MIT Press. 2002

[6] S. T. Birchfield and D.K. Gillmor Acoustic Source Direction By Hemisphere Sampling In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001.

[7] M. Bodden Binaural Modelling and Auditory Scene Analysis *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 1995

[8] M. Brandstein, H. Silverman A Practical Methodology for Speech Source Localization With Microphone Arrays *Computer, Speech and Language*, 11(2):91-126, 1997.

[9] M. S. Brandstein A pitch-based approach to time-delay estimation of reverberant speech *Proc. IEEE ASSP Workshop Appls. Signal Processing Audio Acoustics*, 1997

[10] M. Brandstein, H. Silverman. A robust method for speech signal time-delay estimation in reverberant rooms. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1996.

[11] A. J. N. van Breemen, K. Crucq, B.J.A Kröse, M. Nuttin, J.M. Porta, and E. Demeester. A user-interface robot for ambient intelligent environments. P. Fiorini, editor, *Proceedings of the 1st International Workshop on Advances in Service Robotics, ASER'03*, pages 132–139, Bardolino, Italy, 2003. Fraunhofer IRB Verlag.

[12] S. Canu and C. Soon Ong Learning and Regularization from Interpolation to Approximation *Lecture Notes*, WWW: asi.insa-rouen.fr/∼scanu

[13] Y. Chen and Y. Rui Real-Time Speaker Tracking Using Particle Filter Sensor Fusion *Proceedings of the IEEE*, vol. 92, no. 3, 2004

[14] P. Cosi, S. Pasquin and E. Zovato Auditory Modelling Techniques for Robust Pitch Extraction and Noise Reduction *Proc. ICSLP-98, International Conference on Spoken Language Processing*, Sydney, Australia, Volume 7, pp. 2807-2810, 30 Nov. - 4 Dec., 1998.

[15] J. van Dam, A. Dev, L. Dorst, F.C.A. Groen, L.O. Hertzberger, B.J.A. Kröse, J. Lagerberg and A. Visser Organisation and Design of Autonomous Systems *Reader* Faculty of Science (FdNWI), University of Amsterdam, 2000

[16] S. J. Davey and S. B. Colegrove A Unified Joint Probabilistic Data Association Filter with Multiple Models *DSTO-TR-1184*, July 2001

[17] J. R. Deller, Jr., J. G. Proakis and J. H. Hansen Discrete-Time Processing of Speech Signals *New York : Macmillan Publishing Company* 1993

[18] A. Doucet, N. de Freitas, and N. Gordon (eds.), *Sequential monte carlo methods in practice*, Springer-Verlag, 2001.

[19] A. Doucet On Sequential Simulation-Based Methods for Bayesian Filtering *Eng. Dept., Cambridge Univ., Cambridge, U.K., Tech. Rep. CUED/F-INFENG/TR 310* 1998

[20] R. Duraiswami, D. N. Zotkin, L. S. Davis Active speech source localization by a dual coarse-to-fine search *Proc. IEEE ICASSP*, vol. 5, pp. 3309-3312. 2001

[21] D. Ellis Computational Auditory Scene Analysis Exploiting Speech-recognition Knowledge *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 1997

[22] J. Fritsch, M. Kleinehagenbrock, S. Lang, T. Btz, G. A. Fink, and G. Sagerer, Multi-modal anchoring for human-robot-interaction. *Robotics and Autonomous Systems, Special issue on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems*, 43(2–3):133–147, 2003.

[23] S. Furui. Research on individuality features in speech waves and automatic speaker recognition techniques. *Speech Communication*, 5(2):183–197, 1986.

[24] D. Gerhard Pitch Extraction and Fundamental Frequency: History and Current Techniques *Technical Report*TR-CS 2003-06, 2003

[25] D. Gerhard Silence as a Cue to Rhythm in the Analysis of Speech and Song *Journal of the Canadian Acoustical Association*, 31:3 p22-23, 2003

[26] B. G. de Grooth A simple model for Brownian motion leading to the Langevin equation *Am. J. Phys.*, 67 (12), 1999.

[27] Z. Ghahramani. Learning dynamic Bayesian networks. C.L. Giles and M. Gori, editors, *Adaptive Processing of Temporal Information*, Lecture Notes in Artificial Intelligence, pages 168–197. Springer-Verlag, 1998.

[28] J. Hadamard Sur les problmes aux drives partielles et leur signification physique *Princeton University Bulletin*, 49–52. 1902

[29] C. Hue, J. P. Le Cadre and P. Perez Sequential Monte Carlo Methods for Multiple Target Tracking and Data Fusion *IEEE Transactions on Signal Processing*, vol. 50, no. 2, 2002

[30] M. Isard and A. Blake Condensation-Conditional density propagation for visual tracking *Int. J. Computer Vision*, vol.29, no. 1, pp. 5-28, 1998

[31] E. E. Jan and J. Flanagan Sound Source Localization in Reverberant Environments using an Outlier Elimination Algorithm *Proceedings of the International Conference on Spoken Language Processing*, 1996 vol.3, pp.1321-1324.

[32] R. Karlsson and F. Gustafsson Monte Carlo data association for multiple target tracking *IEE Target tracking: Algorithms and applications*, The Netherlands, Oct 2001.

[33] Z. Khan, T. Balch and F. Dellaert An MCMC-based Particle Filter for Tracking Multiple Interacting Targets *Technical Report number GIT-GVU-03-35*, October 2003

[34] Christer Kiselman Ill-Posed Problems - IPP - Illa stllda problem *Lecture Notes*, WWW: http://www.math.uu.se/∼kiselman, 2003

[35] G. Klaassen, W. Zajdel and B. J.A. Kröse Speech-based Localization of Multiple Persons for an Interface Robot *IEEE Symposium on Computational Intelligence in Robotics and Automation*, Finland, 2005.

[36] C.H. Knapp and G.C. Carter The generalized correlation method for estimation of time delay *Transactions on Acoustics, Speech and Signal Processing*, 24(4):320-327, 1976

[37] R.Y. Litovsky, C.C.Lane, C.A. Atencio and B. Delgutte. Physiological measures of the precedence effect and spatial release from masking in the cat inferior colliculus. *Breebaart, Houtsma, Kohlrausch, Prijs and Schoonhoven (Eds.)*, Shaker Publishing, 221-228. (2001).

[38] J. S. Lui, R. Chen and T. Logvinenko A theoretical framework for sequential importance sampling and resampling In A. Doucet, N. de Freitas, and N. Gordon editors, *Sequential monte carlo methods in practice*, Springer-Verlag, 2001.

[39] S. Maskell, M. Briers and R. Wright Fast Mutual Exclusion *Proceedings of SPIE Conference on Signal Processing of Small Targets*, 2004.

[40] T. Matsui and S. Furui. A text-independent speaker recognition method robust against utterance variations. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1992.

[41] A. Milstein, J. N. Sanchez and E. T. Williamson Robust Global Localization Using Clustered Particle Filtering *Proceedings of AAAI-2002*, 581-586, 2002

[42] A. J. Davison, M. Montemerlo, J. Pineau, N. Roy, S. Thrun and V. Verma, Experiences with a Mobile Robotic Guide for the Elderly *Proc. of the AAAI National Conf. on Artificial Intelligence*, 2002

[43] K. Mustafa. Robust formant tracking for continuous speech with speaker variability in noisy environments. *MAScThesis* McMcMaster University - Dept. of Electrical and Computer Engineering, 2003.

[44] H. Nakashima, T. Mukai and N. Ohnishi Self-organisation of a Sound Source Localization Robot by Perceptual Cycle *Proceedings of the 9th International Conference on Neural Information Processing*, ICONIP '02, 2002.

[45] K. Nakadai, H. G. Okuno and H. Kitano Real-time sound source localization and separation for robot audition *Proc. of 7th International Conference on Spoken Language Processing (ICSLP-2002)*, pp.193-196, Denver, USA, Sep. 2002.

[46] P. Svaizer, M. Matassoni and M. Omologo Acoustic Source Localization in a Three-Dimensional Space using Crosspower-Spectrum Phase *Proc. of ICASSP 97*, Munich, Germany, April 1997

[47] M. Omologo, P. Svaizer. Talker Localization and Speech Enhancement in a Noisy Environment using a Microphone Array based Acquisition System. EUROSPEECH 1993.

[48] M. Omologo and P. Svaizer Acoustic Event Localization using a Crosspower-Spectrum Phase based Technique *Proceedings IEEE ICASSP*, Adelaide 1994 vol.2, pp.273-276.

[49] M. Omologo and P. Svaizer Use of the Crosspower-Spectrum Phase in Acoustic Event Localization *IEEE Transactions on Speech and Audio Processing*, vol.5, no. 12, pp. 288-292.

[50] G. E. Peterson and H. L. Barney Control methods used in a study of the vowels *J. Acoust. Soc. Am.*, vol. 24, no. 2 pp. 175-184, 1952

[51] A. Rao and R. Kumaersan. On decomposing speech into modulated components. *IEEE Transactions on Speech and Audio Processing*, 8:240-254, 2000.

[52] C. Rasmussen and G. D. Hager Joint Probabilistic Techniques for Tracking Objects Using Multiple Visual Cues *IEEE International Conference on Intelligent Robots and Systems*, IROS-98, Victoria, BC, 1998.

[53] I. Rekleitis, G. Dudek nd E. Milios Probabilistic cooperative localization and mapping in practice *International Conference on Intelligent Robots and Systems*, pp.685-691, 1993

[54] R. Yong and D. Florencio New Direct Approaches To Robust Sound Source Localization *Proc. IEEE Int'l Conf. Multimedia and Expo*, Baltimore, July 2003

[55] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. People Tracking with a Mobile Robot Using Sample-based Joint Probabilistic Data Association Filters. *Int. Journal of Robotics Research*, 22 (2), 2003.

[56] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. *IEEE Int. Conf. on Robotics and Automation*, 2001.

[57] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association.* Academic Press, 1988.

[58] K. N. Stevens The quantal nature of speech: evidence from articulatory-acoustic data Human Communication: A Unified View, McGraw-Hill, 1972

[59] A. Swain, W.H. Abdulla. Estimation of LPC Parameters of Speech Signals in Noisy Environment. *IEEE TENCON*, 2004.

[60] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C.R. Rosenberg, N. Roy, J. Schulte and D. Schulz, MINERVA: A Tour-Guide Robot that Learns, {KI} - Kunstliche Intelligenz, (1999) 14-26

[61] J-M. Valin, F. Michaud, J. Rouat and D. Letourneau Robust Sound Source Localization Using a Microphone Array on a Mobile Robot *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.

[62] J. Vermaak and A. Blake. Nonlinear filtering for speaker tracking in noisy and reverberant environments. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2001.

[63] D. Violeau, S. Piccon and J.P. Chabard Two Attempts of Turbulence Modelling in Smoothed Particle Hydronamics *Proceeding of Flow Modeling and Turbulence Measurement*, Tokyo, 2001

[64] D. B. Ward, E. A. Lehmann and R. C. Williamson Particle Filtering Algorithms for Acoustic Source Localization *IEEE Transactions on Speech and Audio Processing*,vol. 11, no. 6, November 2003

[65] G. Welch and G. Bishop An Introduction to the Kalman Filter *Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press, Addison-Wesley, Los Angeles, CA, USA (August 12–17), SIGGRAPH, 2001

[66] D. N. Zotkin and R. Duraiswami Accelerated Speech Source Localization via a Hierarchical Search of Steered Response Power IEEE Transactions on Speech and Audio Processing, vol. 12(5), pp. 499-508, 2004

[67] G. Zweig and S. Russel Speech Recognition with Dynamic Bayesian Networks In *Proceedings of the The fifteenth national/tenth conference on Arificial Intelligence/ Innovative applications of artificial intelligence*, 173-180, 1998.