Beamforming

Sebastian Lindberg
910214-2792
selind@kth.se

Måns Åberg
930717-3113
mansab@kth.se

Abstract

In this paper the topic of beamforming, or space-time filtering, is explored and a basic model of the problem is presented. The performance of a linear array is derived for three different beamforming methods and discussed. An off the shelf solution is finally tested and evaluated for use in speech recognition.

1 Introduction

Separating mixed audio signals is a problem we encounter every day. Whether holding a conversation in a crowded room or having a heated argument with more than one person, we must be able to separate one voice from a cacophony of surrounding sounds. Just how difficult the problem is becomes clear when a machine tries to do it. Nature solved this by introducing multiple channels; our two ears are used to determine the direction of a sound and focus on it. Beamforming is the attempt to give the same ability to the machine.

When recording sound over a distance a number of problems can arise that make voice recognition more difficult. The desired signal can be degraded by reverb, echos or other sources. Some noise types, like echos, can be filtered out with temporal filters, but other signals, which may have the same frequency content as the desired signal, can not. By not only sampling the signal in time, but also in space, spacial filtering can be utilized.

A common method for performing spacial filtering is placing multiple microphones at known locations. The propagating sound wave from one source will reach the different sensors at slightly different times, and the delay at each microphone will be unique to the exact location of that source. By inverting the delays and summing together the recorded signals, sound coming from the desired source is added constructively while other sources are suppressed.

This report will focus on the basic theory of beamforming and study the properties of the linear microphone array. It will also discuss three different methods of performing beamforming. Finally an off the shelf solution will be evaluated using the Microsoft Kinect hardware and software development kit.

2 Beamforming

The problem description is illustrated in fig. 1 which depicts a two-dimensional space with multiple sound sources $s(t)$. These signals are modeled as point sources positioned at coordinates p. The desired signal is placed in p_d and denoted $s_{p_d}(t)$, while the other L interfering sources are labeled $s_{n,l}(p,t)$.

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

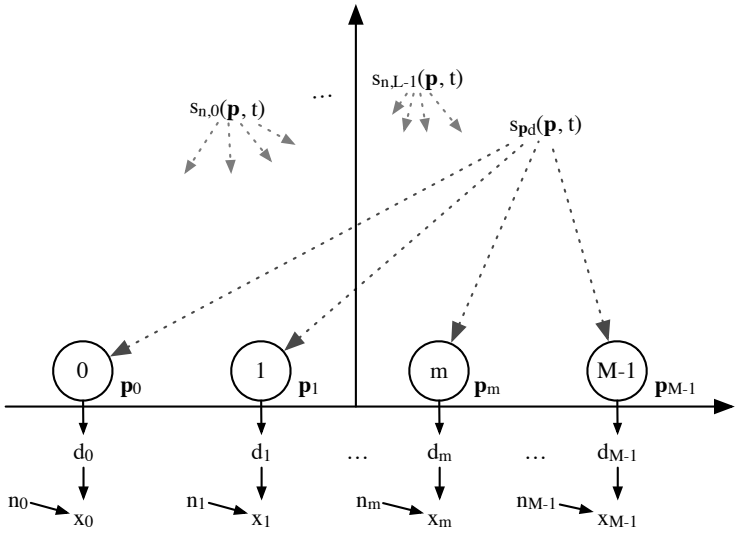

Figure 1: A model of a typical system where beamforming is applicable. The desired sound source $s_{\mathbf{p}_d}$ is placed in $\mathbf{p}_d$ and surrounded by L noise sources $s_{n,l}$. M microphones placed in $\mathbf{p}_0, \ldots, \mathbf{p}_{M-1}$ record the sound from both the desired source as well as the interfering ones, and the result is the sampled signal $x_m(k)$. It is modeled as the sum of the desired signal d_m and the interference n_m.

The signals are recorded by M microphones, placed in $p_0, \ldots, p_{M-1}$. In microphone m the sampled contribution from the desired signal can be written

$$\begin{aligned} d_m(k) &= s_{p_k}(kT_s - \tau_{d,m}) \\ \tau_{d,m} &= \frac{||p_d - p_m||}{c} \end{aligned} \tag{1}$$

where c is the speed of sound and $\tau_{d,m}$ is a delay dependent on the distance between the microphone and the source.

The noise term $n_m(k)$ will not be handled in much detail in this report, but it consists of the interfering noise sources $s_{n,l}$ as well as noise caused by the recording setup. We can finally write the signal recorded by each sensor as

$$x_m(k) = d_m(k) + n_m(k).$$

Next we introduce a FIR filter to each of the sampled signals x_m. It is realized by selecting a set of N weights,

$$\mathbf{w}_m(k) = [w_{0,m}(k), w_{1,m}(k), \ldots, w_{N-1,m}(k)]^T.$$

The output of the beamformer can now be written as

$$y(k) = \sum_{m=0}^{M-1} \mathbf{w}_m(k)^T \mathbf{x}_m(k).$$

The problem is thus to select weights $w_{n,m}(k)$ such that the desired signal is preserved while the noise is attenuated. Using this very generalized model this is not an easy task, which is why we will introduce some simplifications.

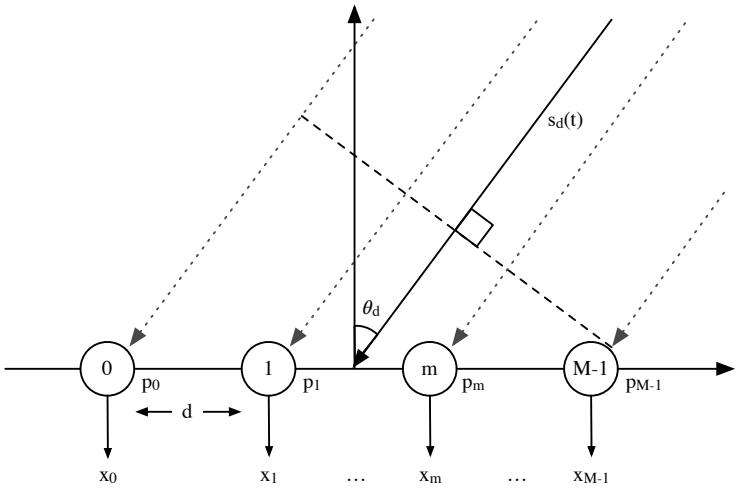

Figure 2: A common simplification is to model the incoming signal as a planar wave.

2.1 Planar Wave Model

To understand how the different parameters affect the performance of the microphone array we will study a monochromatic source of a single frequency ω_0. This type of narrowband beamforming is not generally applicable to sound, which spans a broader band of the spectrum, but it is useful for visualization purposes. We define the desired signal as

$$s_d(t) = S \exp(i\omega_0 t), \tag{2}$$

where S is the amplitude of the complex wave. A further simplification of the model is to treat $s_d(t)$ as a planar wave with incident angle θ_d. This approximation holds as long as the source is far enough away that the difference between the angles of arrival at each microphone is small. Finally we fixate the geometry of the microphone array to a one dimensional array. Note that other configurations are possible with slight modifications to the following equations. The system is depicted in fig. 2.

Since the distance to the signal source no longer matter we alter the receiving model in (1) to

$$\begin{aligned} d_m(k) &= s_d(kT_s - \tau_m)) \\ \tau_m &= \underbrace{\left(\frac{M-1}{2} - m\right)\frac{d}{c}}_{\beta_m} \cos\theta_d \end{aligned} \tag{3}$$

where τ_m is the delay of the signal relative to the origin. Combining (2) with (3) and disregarding noise we arrive at the sampled signal at microphone m,

$$x_m(k) = \underbrace{d_m(k)}_{\text{no noise}} = S \exp(i(\omega_0 k T_s - \tau_m).$$

Transforming the signal to the discrete time frequency domain yields

$$\begin{aligned} X_m(\omega) &= \sum_{k=-\infty}^{\infty} x_m(k) \exp(-i\omega k T_s) \\ &= \exp(-i\omega\tau_m) \underbrace{\frac{2\pi S}{T_s}\delta((\omega - \omega_0)T_s)}_{S_d(\omega)}. \end{aligned}$$

We can identify the exponential term in X_m as the component holding information about which direction the wave arrived from. We introduce the *steering vector* which for a given direction of arrival (DOA) θ and frequency ω is defined as

$$\mathbf{v}(\theta, \omega) = [\exp(i\omega\tau_0), \exp(i\omega\tau_1) \ldots, \exp(i\omega\tau_{M-1})]^H,$$

where H is the Hermitian operator. [1] The steering vector describes how the original signal arrived at, or was *steered* to, the sensor array and allows us to isolate the geometry of the system.

Next we assume that the source is stationary, so that the weights $\mathbf{w}_m(k) := \mathbf{w}_m$.[2] The discrete time Fourier transform (DTFT) of the weights is given by

$$W_m(\omega) = \sum_{n=0}^{N-1} w_{n,m} \exp(-i\omega n T_s).$$

On this form we can see how each of the N weights $w_{n,m}$ scale a phase shift of the signal x_m. We can collect the weights for all of the microphones (in the frequency domain) in one vector

$$\mathbf{w}(\omega) = [W_0(\omega), \ldots, W_{M-1}(\omega)]^H. \tag{4}$$

Finally the output of the beamformer is given by

$$Y(\omega) = \mathbf{w}^H(\omega)\mathbf{v}(\theta_d, \omega_0)\frac{2\pi S}{T_s}\delta((\omega - \omega_0)T_s)$$

and we can identify the transfer function of the system as

$$G(\omega) = \mathbf{w}_f^H(\omega)\mathbf{v}(\theta_d, \omega_0).$$

Recovering the original signal is thereby a matter of identifying the steering vector $\mathbf{v}(\mathbf{k}, \omega)$ and canceling it using the weights $\mathbf{w}_f(\omega)$. Next we will do this using the delay and sum beamformer.

2.2 Delay and Sum

The delay and sum (DS) beamformer attempts to invert the delay at each microphone and estimates the desired signal s_d bu constructively summing together the offset channels. The method may at first glance seem optimal but is in fact a fairly naive approach to the problem. It is however useful in illustrating the properties of the beamformer array.

By defining the weights as the complex conjugate of the steering vector and dividing by the number of sensors,

$$\mathbf{w}(\omega) = \frac{1}{M}\mathbf{v}^*(\theta_d, \omega),$$

we get the desired transfer function

$$G(\omega, \theta) = \frac{1}{M}\mathbf{v}^H(\theta_d, \omega)\mathbf{v}(\theta, \omega) = 1.$$

Using these weights we can rewrite the transfer function as

[1] The Hermitian operator transposes the complex conjugate of the vector.

[2] This is done purely to simplify the derivation and does not generally hold, unless the desired source only ever moves slightly.

Table 1: Parameter values used unless otherwise stated.

Parameter	Value	Description
c	340 m/s	Speed of sound
d	5 cm	Distance between microphones
M	4	Number of microphones
θ_d	$\pi/2$	Look direction
f_0	2 kHz	Signal frequency

$$\begin{aligned} G(\omega,\theta) &= \frac{1}{M}\mathbf{v}_d^H(\theta_d,\omega)\mathbf{v}(\theta,\omega) \\ &= \frac{1}{M}\sum_{m=0}^{M-1}\exp\left(i\omega\beta_m(\cos\theta_d-\cos\theta)\right) \end{aligned}.$$

This is a truncated geometric series which can be simplified to

$$\begin{aligned} G(\omega,\theta) &= \frac{1}{M}\frac{\sin(\omega M\tau_b/2)}{\sin(\omega\tau_b/2)} \\ \tau_b &= \frac{d}{c}(\cos\theta_d-\cos\theta). \end{aligned}$$

Studying this expression we can see that for some combinations of angles and dimensions the beamformer will cancel the signal entirely. The first such null of the response will occur when $\omega M\tau_b/s=\pi$, which can be written as

$$\cos\theta=\cos\theta_d-\underbrace{\frac{2\pi c}{wMd}}_{\frac{c}{fMd}}.$$

Here we can clearly see how the different parameters influence the beam width at a particular frequency ω and look angle θ_d. Increasing the aperture Md of the array narrows the main lobe, improving the array's directivity. The null position is also dependant on the frequency, limiting how small the array can be made given a certain lowest frequency of the signal.

2.2.1 Performance

The performance of the beamformer array is effectively visualized through the beam power, defined as

$$P(\omega|\theta)=|G(\omega,\theta)|^2$$

that has been calculated for an array of four microphones ($M=4$) for different parameter values in fig. 3 and fig. 4. The parameter values used to generate the figures are, unless otherwise stated, presented in table 1.

It is worth noting that, especially for higher frequencies and larger apertures, the beamformer transfer function can equal 1 outside of the main lobe. This is due to spacial aliasing and happens when $d\leq\frac{c}{2f}$. These additional lobes makes attenuation of noise in some frequencies impossible.

2.3 Least Squares Beamformer

The DS beamformer design is flawed in that it only considers the transfer function $G(\omega,\theta)$ in the look direction θ_d. Another approach to designing the beamformer is through the least-squares (LS) method.

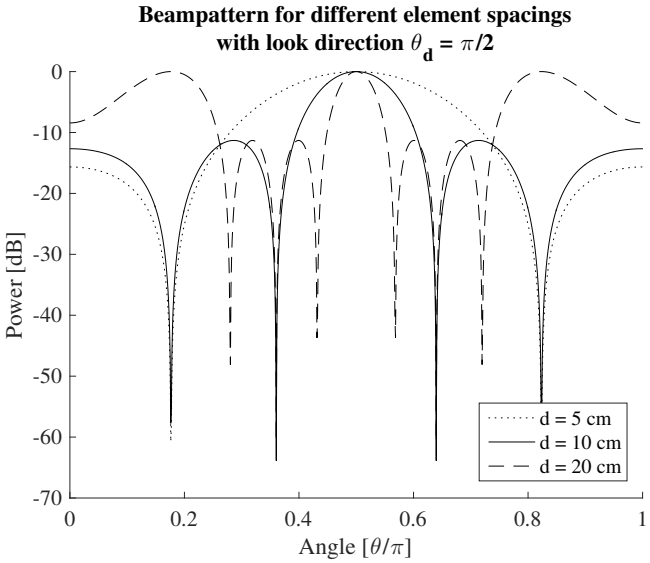

Figure 3: The response of the DS beamformer for different distances d between the microphone elements. The main lobe width decreases with larger distances.

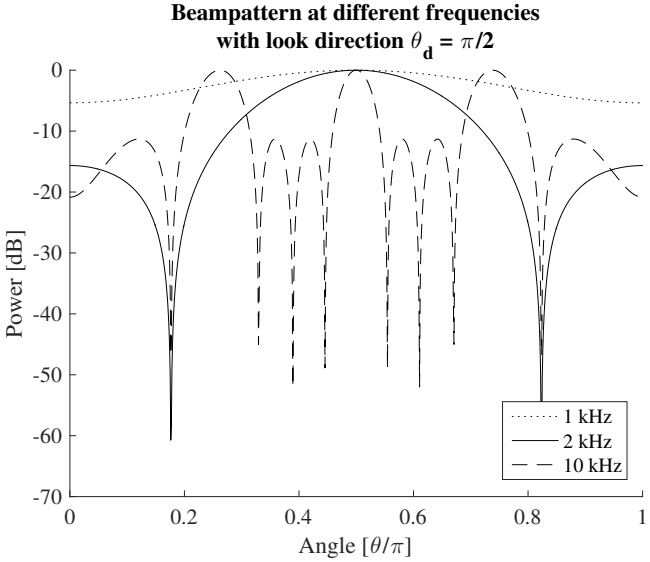

Figure 4: The response of the DS beamformer for different frequencies. The main lobe width decreases with higher frequencies. For low frequencies the response is nearly flat, resulting in poor directivity. The dimensions and geometry of the microphone array clearly limit the operating range of the DS beamformer.

We start with the notion of a desired transfer function $G_d(\omega, \theta)$. This allows for a more descriptive model of the beamformer, as the single look direction is replaced by a function. This can typically be defined as non-zero in a small range of angles,

$$G_d(\omega, \theta) = \begin{cases} 1 & \text{if } \theta_1 \leq \theta \leq \theta_2 \\ 0 & \text{otherwise} \end{cases}.$$

This model thereby allows for some uncertainty in the DOA. The error function we wish to minimise can now be formulated as the difference between the desired transfer function and the actual one. The square error is defined as the integral over all angles θ, such that

$$\varepsilon^2 = \int_0^\pi |G(\omega,\theta) - G_d(\omega,\theta)|^2 d\theta.$$

Using the definition of the weights in (4) we can expand the error term,

$$\begin{aligned}
\varepsilon^2 &= \mathbf{w}^T(\omega)\mathbf{A}\mathbf{w}(\omega) - 2\mathbf{w}^T(\omega)\mathbf{b} + \int_0^\pi |G_d(\omega,\theta)|^2 d\theta \\
\mathbf{A} &= \int_0^\pi \mathbf{v}(\theta,\omega)\mathbf{v}^H(\theta,\omega) d\theta \\
\mathbf{b} &= \int_0^\pi \mathrm{Re}[\mathbf{v}(\theta,\omega)G_d(\omega,\theta)] d\theta.
\end{aligned}$$

By taking the derivative of ϵ^2 with respect to $\mathbf{w}$ and setting the result to zero we obtain the least-square solution

$$\mathbf{w}_{\mathrm{LS}} = \mathbf{A}^{-1}\mathbf{b}.$$

The matrix $\mathbf{A}$ is only dependent on the geometry of the array, just like in the DS case, but $\mathbf{b}$ now include information about the desired transfer function $G_d(\omega,\theta)$.

$$\mathbf{A} = \begin{bmatrix} \int_0^\pi 1 \, d\theta & \cdots & \int_0^\pi \exp(i\omega(\beta_{M-1} - \beta_0)\cos\theta) \, d\theta \\ \int_0^\pi \exp(i\omega(\beta_0 - \beta_1)\cos\theta) \, d\theta & \cdots & \int_0^\pi \exp(i\omega(\beta_{M-1} - \beta_1)\cos\theta) \, d\theta \\ \vdots & \ddots & \vdots \\ \int_0^\pi \exp(i\omega(\beta_0 - \beta_{M-1})\cos\theta) d\theta & \cdots & \int_0^\pi 1 \, d\theta \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} \int_{\theta_1}^{\theta_2} \cos(\omega\beta_0\cos\theta) \, d\theta \\ \int_{\theta_1}^{\theta_2} \cos(\omega\beta_1\cos\theta) \, d\theta \\ \vdots \\ \int_{\theta_1}^{\theta_2} \cos(\omega\beta_{M-1}\cos\theta) \, d\theta \end{bmatrix},$$

where β_m is defined in (3).

2.3.1 Performance

The narrowband LS beamformer allows for controlling the width of the main lobe. This can be seen in fig. 5 where the shape of the main lobe has been designed at different frequencies. Towards the higher end of the spectrum the beamformer looses its ability to affect the beampattern due to physical limitations. The wavelength is simply too small in relation to the distance between the microphones.

As can be seen in fig. 6 the beam width of the narrow band beamformer is not constant, but varies with frequency. This is not ideal for audio signals, which span a large part of the frequency spectrum. The problem is most obvious when the DOA is not estimated correctly in which case the beamformer will act as a low pass filter on the desired signal. The attenuation of noise and interference will also not be the same for all frequencies which introduces further distortions. This is handled by a broadband beamformer.

2.4 Broadband Beamformer

A broadband beamformer transforms the signals from the microphones into the frequency domain and performs narrowband beamforming on each frequency. That way the beampattern can be kept constant over the entire spectrum.

2.4.1 Performance

While in theory very simple, the method works well for lessening the strong dependency on frequency of the beampattern, as can be seen in fig. 7.

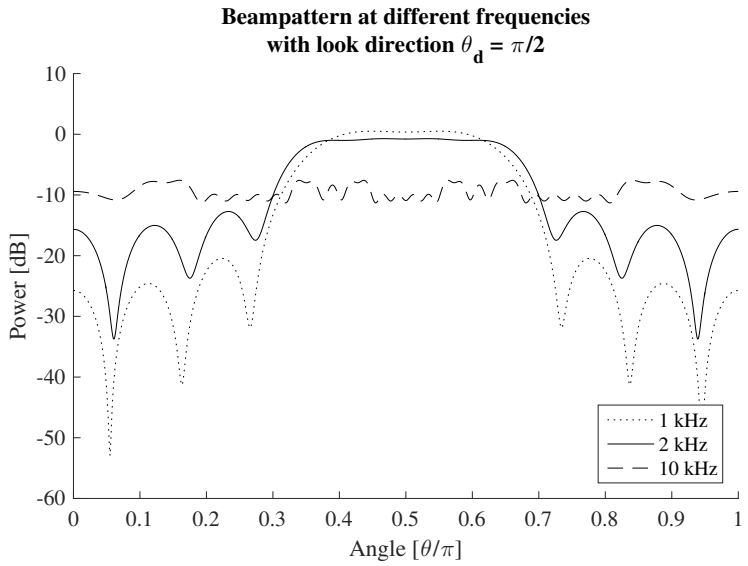

Figure 5: The beampattern for an LS beamformer, designed at three different frequencies.

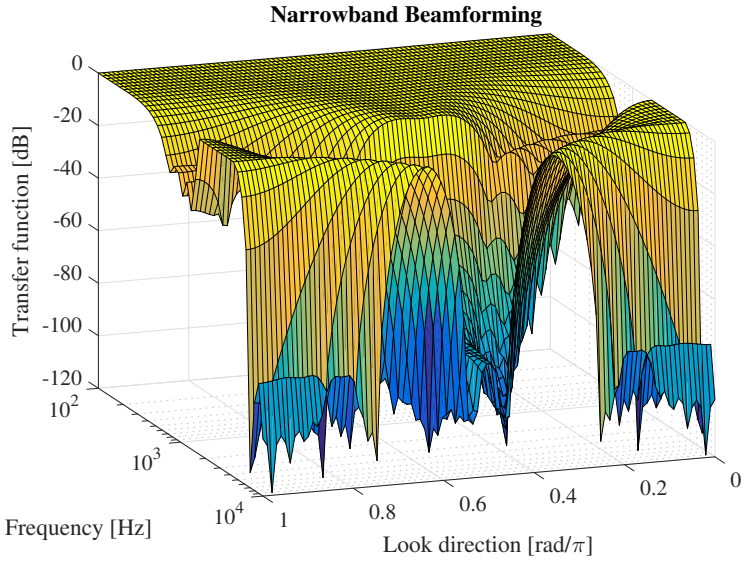

Figure 6: The beampattern across the spectrum for a narrowband beamformer. The shape of the main lobe, in this case designed at 1.5 kHz, varies greatly with frequency.

3 Kinect Beamforming

The Kinect sensor has, apart from its cameras, an array consisting of four microphones each using 24-bit ADC and sampling at 16kHz. It also provides signal processing such as noise suppression and echo cancellation. The good thing about having this array of microphone is the ability to utilize beamforming- or "spatial filtering".

The SDK for the Kinect does not provide the distance to an audio source, but rather its angle and where it is in the XY-plane in relation to the center of the sensor. A beamforming technique that the Kinect SDK provides is Adaptive Beamforming where the Kinect selects the beam. You can utilize Automatic Beamforming where the system chooses the beam or you can set the Kinect to let an application handle the beamforming, treating the sensor as a 4-channel microphone.

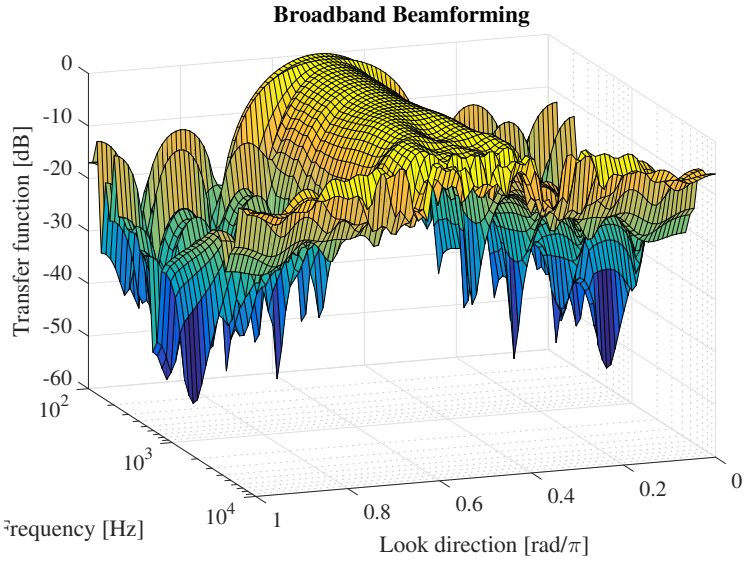

Figure 7: The beampattern across the spectrum for a broadband beamformer. The shape of the main lobe is, compared to the narrowband beamformer response in fig. 6, mostly constant across the spectrum.

As for the algorithm used in the Kinect, the only documentation found on this says that it uses an MVDR (Minimum Variance Distortionless Response) (Thomas, Ahrens, & Tashev, n.d.) that is also known as the Capon beamformer. It is said to be providing higher selectivity and quality of the sound than the DSBF (Delay-And-Sum-Beamforming) (Li, Banerjee, Popescu, & Skubic, 2013). The power is given by:

$$\hat{P}_{\text{Capon}}(\theta) = \frac{1}{v^H R^{-1} v} \tag{5}$$

3.1 Method for testing

Since the Kinect already provides noise suppression the testing of the sensor will be a comparison between using only a single microphone on the Kinect and using beamforming. In practice this would mean;

- The sensor is either placed close to a noisy sound-source, i.e a computer with loud fans, or noise is applied to the sound that is going to be recorded by the Kinect.
- A sound source, one-channel output, is placed at a distance from the sensor where the recorded sound is greater than the noise and that is not right in front of the sensor, but rather to the side so that the effect of beamforming will be more prominent.
- At least one utterance from the tidigits database is played back for the Kinect to record using both a single microphone and beamforming with the microphone array.
- The recordings are then tested with the HTK toolkit to see how well the it recognises both.

3.2 Experiments

The set of data that was used was a subset from the Tidigits database. One speaker that was male, and one that was female. Five digit sequences were chosen from each speaker accounting for a total of ten soundfiles. The data was used in a set of experiments for recording the digit sequences by the Kinect sensor.

3.2.1 Experiment 1

Each digit sequence was played back through a speaker that was connected to the same computer that was recording via the Kinect. The Kinect was placed close to the stationary computer in order to capture the noise coming form the fans. Each sequence of digits was then played back to the Kinect from two angles;

- 90° from the center/to the side of the Kinect.
- 0° from the center/in front of the Kinect.

For each test, the Kinect recorded all ten sequences with and without its built-in beamforming, called 'optimbeam'.

For this first experiment with noise this accounted for a total of 40 recordings.

3.2.2 Experiment 2

The same set of data was used for the second experiment, meaning ten different digit sequences where five were male, and five female. For this experiment, however, the Kinect was to test the ability to source out the stronger of two sounds. Two speakers were used for this, one where the digit sequences was played from, and the other with a recording of someone saying "bla bla bla..." continuously. The speakers were placed roughly 45° from the center of the Kinect, but at different distances to ensure that the speaker playing back the digit sequence was louder. This was done as well by using both the beamforming found in the Kinect and only using a single microphone in the array, accounting for 20 recordings in total.

3.2.3 Experiment 3

When all of the data, 60 recordings, was at hand, the only thing left was to try and see whether or not the beamforming yielded better results than a single microphone. This was done in two ways:

- Analysing the SNR using matlabs snr(x) function.
- Training and testing a G-HMM and GMM-HMM model to recognise digits in the sequences.

The SNR was computed for each digit sequence, with and without beamforming, and stored in a vector such as there were three values; one for the test with noise with speaker in front, one for the test with noise with speaker to the side and one for the test with two channels. To compare the SNR they were then plotted, for each sequence of digits, beamforming vs single microphone.

The training was done with the same model that was used in the third lab of the course, but with this subset of the tidigits database. First the G-HMM model was trained with MFCC_0_D_A features followed by testing it with various datasets such as;

- All the recordings that used beamforming.
- All the recordings that used a single microphone.
- Recordings only using beamforming/single mic and noise.
- Recordings only using beamforming/single mic and two channels.
- Recordings only using beamforming/single mic from the side and in front of the Kinect respectively.

Same thing was done training the GMM-HMM model to see if it changed the outcome of the results.

3.3 Results

First, the SNR given by all but one comparison showed a greater SNR for the beamforming compared to a single mic. All of the other comparisons follows the same trend as the comparison shown in fig. 8

The second experiment did not show any different results. As shown in fig. 9, where each color corresponds to a test that was done on the G-HMM model. There was no difference in the results between the G-HMM and GMM-HMM models except for overall better accuracy.

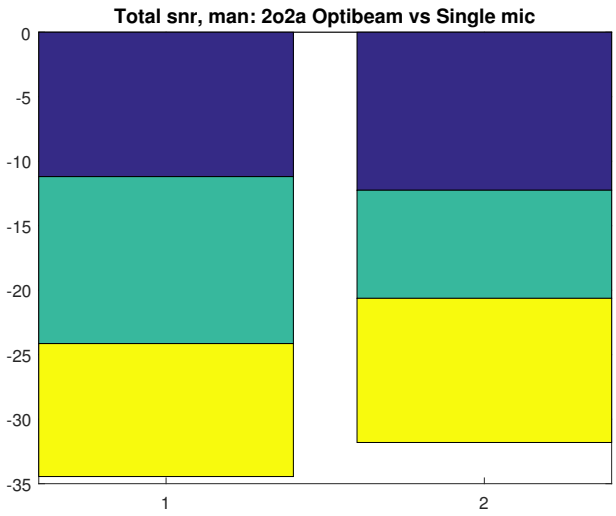

Figure 8:

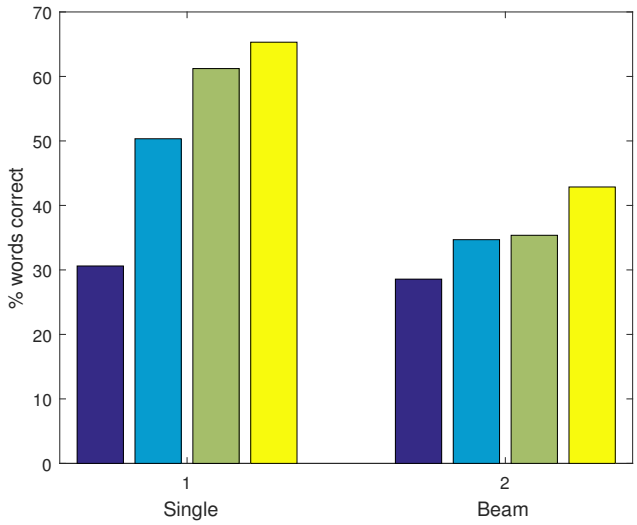

Figure 9: Correctly guessed words.

The first test was the best one for beamforming, the test with two channels, since it almost performed as well as the single microphone. The other three staples corresponds to the noise-tests recorded from the side, the front and both of them combined.

3.4 Discussion

In theory beamforming would yield better results than using just a single microphone. One thing to keep in mind is that the tests, and especially the one with placing the speaker to the side, might not even utilize the beamforming properly. As mentioned by another group doing experiments with the angle of a sounds using beamforming, the Kinect's built-in algorithm does not function properly if the angle exceeds $50°$. This test from the side might not have used the built-in algorithm properly at all then which could partly explain the results.

The beamforming can also act as a low pass filter at times, and since the playback of the digit sequences were through a speaker that had more feedback in the lower frequencies this perhaps affected the outcome of the results.

3.5 Conclusion and future work

In theory, beamforming helps to improve ASR, and while we cannot conclude this from the evaluation of the Kinect it still holds due to flaws in the evaluation.

For future work it would be more interesting to see if an implementation of a beamforming algorithm in, for example, Matlab would hold up against a single microphone. Also evaluating different algorithms to see their strengths and weaknesses in practice, and even see how they could be optimized.

Evaluating the Kinect V2 could also prove to be interesting in order to see how much better it performs given the improved specifications that it has in terms of hardware. If the built-in software is the same in both versions is unknown for this report, but given that the algorithm is the same, how much would the hardware affect the results is probably an interesting topic to investigate.

References

[1] Wolfgand Herbordt. (2004) Sound capture for human/machine interfaces : practical aspects of microphone array signal processing. *ISBN 3-540-23954-5* Berlin: Springer.

[2] Benesty, J. & Chen, J. & Huang, Y. (2008) Microphone Array Signal Processing. *ISBN 978-3-540-78611-5* Berlin: Springer.

[3] Li, Y., Banerjee, T., Popescu, M. & Skubic, M. (2013) Improvement of acoustic fall detection using Kinect depth sensing. Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, *EMBS, 6736–6739. doi:10.1109/EMBC.2013.6611102*

[4] Thomas, M. R. P., Ahrens, J., & Tashev, I. J. (n.d.) Beamformer Design Using Measured Microphone Directivity Patterns: Robustness to Modelling Error, 0(1), 1–4.