Distributed Robust Min Variance Beamforming

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

by

Alexander Brill

Supervised by

Prof. Danny Dolev

Dr. Danny Bickson

School of Engineering and Computer Science
The Hebrew University of Jerusalem
Israel

September 8, 2009

Acknowledgments

I wish to express my deep appreciation to Prof. Danny Dolev for his support, valuable ideas and excellent guidance throughout the research process.
I would also like to express my thanks to Dr. Danny Bickson for his professionalism, open mindness and his encouragement during this period.

Abstract

Recent work by Lorentz and Boyd present a robust minimal variance beamformer (RMVB) that is robust to the transmitter angle of arrival. In the current work we extend the RMVB construction in two directions. First, we propose a distributed algorithm that be computed jointly by the receiving antennas. Second, we extend the RMVB model to allow uncertainties in both phase and amplitude of the received signal as well as the angle of arrival.

Finally, we show that our distributed RMVB model reaches at least the same accuracy in calculating the beamformer as the regular one, besides being more efficient.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Distributed beamforming is a subject of recent interest as it is applicable in sensor networks and other distributed systems, where cooperation among terminals is needed. A system with transmitter cooperation or transmit–beamforming includes a number of antennas transmitting similar narrowband signals, each using a different complex weight. The beamforming weights are determined so that the different transmitted waves add constructively in the direction of the receiver. In a system with receiver cooperation or receive beamforming, the different weights are applied to the signals received at the different antennas/terminals, so that the weighted signals coherently add for waves received from a desired physical direction, where the transmitter is located. Transmit and receive beamforming are mathematically very similar and their optimal weights are identical. When applied to sensor networks, receive beamforming is more challenging because it requires the collection of signals received at different terminals.

The common method of calculating the weights in a centralized system is called Minimum Variance Beamforming (MVB) or Capon's method [1]. In this method, the weights minimize the SNR with a constraint that the overall gain of the desired signal is fixed. The MVB method is not robust to inaccuracies of the CSI (channel state information); In particular, the MVB solution is not robust to uncertainty regarding the angle of arrival in the receive beamforming setting or the desired angle of the beam in the transmit beamforming case. A significant improvement to the MVB method, named Robust Minimum Variance Beamforming, was suggested by *Lorenz & Boyd* [1]. This (centralized) algorithm takes into account the uncertainty in the channel, as well as in the desired direction of the beam. Our work extends the RMVB method to a distributed settings, where the beamformer is computed jointly by the receiving antennas.

1.2 Thesis Outline

The thesis is organized as the following. Chapter 2 introduces the related work that was done in this area. There is a review of several essays that contribute to the understanding of the field. The problem settings, Capon and RMVB algorithms are described in Chapter 3. Chapter 4 discusses our construction of the solution. The experimental results are shown in Chapter 5. Finally, Chapter 6 concludes the research and discusses future work.

Chapter 2

Related Work

2.1 Background

In wireless sensor/relay networks, each node serves as a distributed transmitter trying to send a message to a common base-station receiver. The applications for such networks may vary and among others there are acoustics, radar, sonar, communications, medical imaging and data processing etc.

By focusing the transmission of each node in the direction of the base-station the systems enjoys a proportional SNR increase. Thus, if a single transmitter sends a message with power P and the receiver achieves an SNR of p, an array of N such transmitter will achieve N*p SNR with the same total gain P. Meaning that each transmitter will now have to use only P/N power. Such an effect happens if the signals of the transmitters are adapted to constructively interfere at the receiver. This enhanced energy efficiency can be crucial for a network design. Adaptive antenna arrays also cancel the interference by null-steering.

Keeping this in mind, it's obvious that beamforming can also be used in networks of limited transmit power sensors. A solution for increasing communication distance can be simply adding more transmitters. In addition to these cases, beamforming can be used in networks, such as low carrier frequencies communication networks, where in some cases directional antennas are hard to implement.

2.2 The Challenge

The challenge in implementing beamforming is the requirement of phase and frequency synchronization. In a distributed network, in comparison to a centralized one, each transmitter has its' own carrier RF signal which comes from a separate local oscillator circuit. It means that the signals are not synchronized a priori. Centralized antenna arrays make several assumptions, which are not met in a distributed system. The base-band signals are synchronized and the carrier frequency and phase, which coherent the signals, are the same for all the nodes. It is easily accomplished thanks to common circuits that generate the signals and the a priori known delays between the circuits. Capon's implementation suffers from the distributed network characteristics. The assumption of a point source, as in Capon's solution, is not always practical. Opposite examples are cellular mobile communications, radar and sonar, where the source of interest is scattered around some nominal position. It can also be easily shown, and our results show it, that whenever the knowledge of the signal vector is imprecise, which is common as a result of imprecise angle of arrival estimation or inaccurate array response knowledge, Capon's beamformer performance decreases.

Distributed beamforming network, faces several more difficulties. The channel phase response complicates the phase coherence of the signals. Also, one must remember that it is common for a network to have certain constrains one its' nodes, such as complexity, gain and communication with other nodes as well as with the receiver.

Trying to design a network that solves the synchronization problem by estimating the channel directly and then sending the estimation to the receiver, requires a large overhead since all the channels paths between each transmitter must be calculated. Thus, the existing beamforming schemes sum up to networks where the receiver and the transmitter have no knowledge about the channel realization.

In master-slave implementations, which are described later, there is an unknown propagation delay between the transmitters. Finally, another effect which would be considered in some realizations is channel phase responses change in time. This might happen, for example, due to Doppler effects from moving elements.

2.3 Master-Slave Implementation

The authors of [2] and [3] suggest a master-slave architecture, where the master transmitter is in charge of the signal synchronization of other slave sensors. There is no communication with the remote base station. Instead, there is only a need in a cheap local communication with the master. Thus, a centralized antenna array is emulated. Their research shows that as time goes by, even once the slave transmitters are synchronized, phase noise in oscillators gets them out of sync. That's why the master sensor has to resynchronize its slaves periodically. The simulation results show that even using a non-optimal PLL(phase-locked loop) and having phase error on the order of $60°$, the achieved SNR can reach 70% of the maximum.

2.4 Feedback Channel

The beamforming schemes in [4], [5] and [6] offer a structure consisting of several transmitters, receiver and a limited (usually in bandwidth) feedback channel.

In [4], the setup is restricted to sending binary messages to a receiver (sending +1 or -1) and accepting a one bit feedback channel (+1 or -1). This simplified setup serves the authors to prove good upper and lower bound to the convergence time and to show it is linear in the number of sensors in the network. The authors strictly state that the settings are not representative of a real wireless channel but the combinatorial techniques is worth presenting. The algorithm is as follows. Given N sensors, sensor i sends the message $c_i[j] \in \{+1, -1\}$ at round j. The channel is represented by a vector $c \in \{+1, -1\}^N$. The transmission of the message through the channel is multiplying both vectors element by element, which result at vector $Y[j]$. The gain at the receiver is naturally $|Y[j]|$.

The beamforming algorithm uses randomization. Each node starts with an arbitrary signal (+1 or -1). At every round, each node flips a coin with some certain (later changing) probability to land on heads. The node will toggle its signal according to the flip and send it. The receiver, after accepting $Y[j]$, will compare it to $Y[j-1]$ and return +1 (-1) if $Y[j] > Y[j-1]$ ($Y[j] \leq Y[j-1]$). Nodes that toggled their signal and received +1, change their defaults. Others remain with the defaults. It is shown that although the time required satisfying the convergence condition (which is: $Y[j] >= \beta N$ for $\beta < 1$, β is some percentage of the maximum gain) is linear at N, the upper and lower bounds are loose.

The distributed synchronization procedure in [6] is related to several similar researches. The authors show new theoretical results and observations on a known prototype. It is made of master-slave architecture in order to synchronize the frequency and carrier signals and a one bit feedback channel with the base station. The feedback control algorithm is similar to the one described in [4] except of two differences. A phase rotation angle is randomized at each round, instead of a single bit. Also after a successful round the phase is added to the node's default value, instead of being toggled. The authors build an analytical model for the convergence of the algorithm and show three interesting results: 1) the algorithm is reaching perfect coherence almost surely for any initial phase angle; 2) the expected received signal is a non-decreasing function of N; 3) the time needed for convergence is at most linear at N.

In addition, an improved algorithm that takes special care of channel phase change in time is presented.

[5] is based on the feedback structure, which is described in [6], but it also presents a new research of the process. The algorithm for improving the SNR by randomizing a change in phase is being analyzed together with its' performance. The authors are interested in the reasons behind the effectiveness of the proposed scheme. In order to solve this, they establish equivalence between a local random search algorithm and the magnitude function $Mag(.)$, which measures the received signal magnitude. Naturally, $Mag(.)$ needs to be maximized in order to enhance the SNR. This comparison lets them study the convergence of the distributed beamforming scheme.

The research is built in several steps. First, after the equivalence has been constructed, it's necessary to show that $Mag(.)$ has no local maxima other than the global ones. Otherwise the local search algorithm can stuck on some local maximum if the random phase delta is too small. This is a necessary condition for the convergence. Then, they show that $Mag(.)$ has multiple global maxima. This denotes the fast convergence issue. In addition, the hitting time proof shows that it scales linearly with the number of transmitters.

2.5 GRCB

In [7] a new adaptive beamforming algorithm called Gaussian Robust Capon Beamformer (GRCB) is proposed. It is based on a conventional Capon. The algorithm is used in a communication environment, where there is a high angular spread (more than 2°) on signal of interest. As was mentioned earlier, Capon is sensitive to variants and inaccuracy in the signal. The proposed GRCB uses diagonal loading approach to efficiently solve the problem of response errors and to improve the signal of interest isolation.

In general, it recalculates a new response vector from the knowledge of area of uncertainty and adds the optimal diagonal loading according to the angular spread.

The algorithm is as follows: A channel estimation algorithm determines the signal of interest angular spread (α) and the Gaussian uncertainty area of the pdf (probability density function). A mean value, which is selected from the pdf, will be a new estimation for the response vector a_0. The data covariance matrix R is decomposed by SVD to UAU^*, where A is a diagonal matrix with data covariance eigen-values. A new data covariance matrix $R_d = R + \alpha * A$ is constructed and the beamformer w is calculated: $w = \frac{R^{-1} * a_0}{a_0^* R^{-1} * a_0}$. It should be noted that the execution time of the algorithm is up to 8 times faster than the conventional case.

Chapter 3

Problem Statement

3.1 The beamforming problem

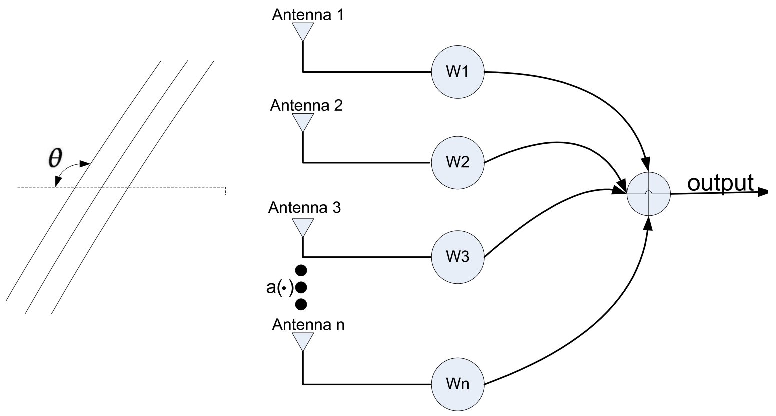

Figure 3.1: Diagram of the antenna array used.

3.2 Robust Min Variance Beamforming (RMVB)

In this section we describe the RMVB method of [1].
The minimum variance beamformer (MVB) is chosen as the optimal solution of:

$$\begin{array}{ll} \text{minimize} & w^* R_y w \,, \\ \text{subject to} & w^* a(\theta) = 1 \;. \end{array} \tag{3.1}$$

The RMVB beamformer deals with the uncertainty of a, presenting a more general form of (3.1) would be:

$$\begin{array}{ll} \min\limits_{w} & w^* R_y w\,, \\ \text{subject to} & \Re\{w^* a\} \geq 1 \qquad \forall a \in \mathcal{E}\,. \end{array} \tag{3.2}$$

$\mathcal{E}$ is an ellipsoid that covers the possible range of values of a. The use of $\mathcal{E}$ is due to imprecise knowledge of the array manifold a, phase uncertainty at the receivers or other factors. The optimal solution of (3.2) is called: the Robust Minimum Variance Beamforming (RMVB).

The n-dimensional ellipsoid is defined as:

$$\mathcal{E} = \{Au + c \;\mid\; \|u\|_2 \;\leqslant 1\;\}\,,$$

where: $A \in R^{n\times n}$ and $c \in R^n$. $\mathcal{E}$ describes an ellipsoid whose center is c and whose principle semiaxes are the vectors of A.

It is more convenient to convert w and $a(\cdot)$ to their real representation

$$x = \begin{bmatrix} \Re\{w\} \\ \Im\{w\} \end{bmatrix} \quad \text{and} \quad z = \begin{bmatrix} \Re\{a\} \\ \Im\{a\} \end{bmatrix}\,.$$

The quadratic form $w^* R_y w$ is expressed as $x^T R x$, where

$$R = \begin{bmatrix} \Re\{R_y\} & -\Im\{R_y\} \\ \Im\{R_y\} & \Re\{R_y\} \end{bmatrix}\,.$$

The RMVB can then be expressed as

$$\begin{array}{ll} \min\limits_{x} & x^T R x\,, \\ \text{subject to} & \|A^T x\|_2 \leqslant c^T x - 1\,. \end{array} \tag{3.3}$$

By imposing of the additional constraint that $c^T x \geq 1$, RMVB becomes an optimal solution to

$$\begin{array}{rc} \min\limits_{x} & x^T R x\,, \\ \text{subject to} & \|A^T x\|_2^2 = (c^T x - 1)^2\,. \end{array} \tag{3.4}$$

The conforming Lagrangian is

$$L(x,\lambda) = x^T R x + \lambda(\|A^T x\|_2^2 - (c^T x - 1)^2)\,, \tag{3.5}$$

where $\lambda \geq 0$ is the Lagrange multiplier. For finding the optimality conditions we compute the partial derivatives for x and λ

$$\frac{\partial L(x,\lambda)}{\partial x} = (R + \lambda Q)x + \lambda c = 0\,, \tag{3.6}$$

$$\frac{\partial L(x,\lambda)}{\partial \lambda} = x^T Q x + 2c^T x - 1 = 0\,, \tag{3.7}$$

where $Q = AA^T - cc^T$. The equations are reduced to a secular equation:

$$f(\lambda) = \lambda^2 \sum_{i=1}^{n} \frac{\overline{c_i}^2 \gamma_i}{(1+\lambda\gamma_i)^2} - 2\lambda \sum_{i=1}^{n} \frac{\overline{c_i}^2}{1+\lambda\gamma_i} - 1\,.$$

γ are the generalized eigenvalues of Q and R and are the roots of the equation $\det(Q - \lambda R) = 0$. Then the optimal value λ^*, satisfying $f(\lambda^*) = 0$ is computed. Finally the RMVB is computed according to

$$x = -\lambda^*(R + \lambda^* Q)^{-1} c\,.$$

Given R, strictly feasible A and c:

1. *Calculate* $Q \leftarrow AA^T - cc^T$
2. *Change coordinates*
   (a) compute Cholesky factorization $LL^T = R$
   (b) compute $L^{-\frac{1}{2}}$
   (c) $\widetilde{Q} \leftarrow L^{-\frac{1}{2}} Q (L^{-\frac{1}{2}})^T$
3. *Eigenvalue/eigenvector computation.*
   (a) compute $V \Gamma V^T = \widetilde{Q}$
4. *Change coordinates*
   (a) $\bar{c} \leftarrow V^T R^{-\frac{1}{2}} c$
5. *Secular equation solution.*
   (a) compute initial feasible point $\hat{\lambda}$
   (b) find $\lambda^* > \hat{\lambda}$ for which $f(\lambda) = 0$
6. *Compute* $x^* \leftarrow (R + \lambda^* Q)^{-1} c$

Figure 3.2: Robust min variance beamforming algorithm

Chapter 4

Our Construction

We first describe a centralized solution to the RMVB problem, and then show how to distribute it. We observe that the RMVB formulation (3.3) is a *second order cone program* (SOCP), a convex optimization problem of the form

$$\min_x f^T x$$

$$\text{such that} \quad \|A_i x + b_i\|_2 \le c_i^T x + d_i, \quad i = 1, \ldots, m\,,$$

$$A \in \mathbb{R}^{n\times n} \quad f, x, c \in \mathbb{R}^n \quad d \in \mathbb{R}.$$

Now, unlike [1] we propose to use an interior point method (Newton method) for solving the SOCP problem. The Newton method is given in Table 4.1.

Given		feasible starting point z_0 and tolerance $\epsilon > 0$, $k = 1$
Repeat	1	Compute the Newton step and decrement $\Delta z = -f''(z)^{-1} f'(z), \qquad \mu^2 = f'(z)^T \Delta z$
	2	Stopping criterion. Quit if $\frac{1}{2}\mu^2 \le \epsilon$
	3	Line search. Choose step size t by backtracking line search.
	4	Update. $z_k := z_{k-1} + t\Delta z, \quad k = k + 1$

Table 4.1: The Newton method [8, §9.5.2]

We start by constructing the Lagrangian (3.5), and compute the first order optimality conditions by differentiating it and setting the partial derivatives equal to zero (3.6),(3.7). We define a new variable $z \triangleq [x^T \ \ \lambda]^T$. For computing the search direction Δz we need to compute the gradient $f'(z)$ and Hessian $f''(z)$ which are given below:

$$f'(z) = \begin{pmatrix} (R+\lambda Q)x + \lambda c \\ \frac{1}{2}x^TQx + c^Tx - \frac{1}{2} \end{pmatrix}, \qquad f''(z) = \begin{pmatrix} R+\lambda Q & Qx+c \\ (Qx+c)^T & \mathbf{0} \end{pmatrix},$$

We start from the initial point $z_0 = c$, which is the center of the ellipsoid $\mathcal{E}$ and thus conforming to the constraints. In each Newton step we compute the search direction z, we use a backtracking line search to compute a step size $t \in (0, 1]$, and update the current point to $z_{k-1} + t\Delta z$. We stop when $\frac{1}{2}\mu^2 < \epsilon$.

The main computational bottleneck of the Newton method shown above is the Newton step. In each Newton step, we solve the following linear system of equations.

$$f''(z)\Delta z = -f'(z) \ , \tag{4.1}$$

This can be done naively using Gaussian elimination in $O((2n+1)^3)$ operations since $z \in \mathbb{R}^{2n+1}$. We propose the *preconditioned conjugate gradient* (PCG) method to speed up this computation. The PCG method [9],[10, §6.6], [11, chap. 2], [12, chap. 5] is given in figure [4.1]. The input to the PCG method is a matrix which is positive definite. Since the matrix $f''(z)$ is not positive definite, we have used the following preconditioning (denoting $H \triangleq f''(\Delta z), g \triangleq f'(z)$)

$$H^T H\Delta z = -H^T g \ , \tag{4.2}$$

Now we are able to use the PCG method. PCG makes all matrix and vector arithmetics distributedly. It includes vector inner product, matrix by vector product, matrix by scalar product, vector substraction, etc'. We can see that each step takes $O(n^3)$ products or additions.

Input: A, b, x_0. A is a real, symmetric, positive-definite matrix.

1. $r_0 := b - Ax_0$
2. $P_0 := r_0$
3. $k := 0$
4. repeat:
   (a) $\alpha_k := \frac{r_k^T r_k}{P_k^T A P_k}$
   (b) $x_{k+1} := x_k + \alpha_k P_k$
   (c) $r_{k+1} := r_k - \alpha_k P_k$
   (d) **if** r_{k+1} is sufficiently small **then** exit loop **end if**
   (e) $\beta_k := \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
   (f) $P_{k+1} := r_{k+1} + \beta_k P_k$
   (g) $k := k + 1$

Output x_k.

Figure 4.1: Conjugate Gradient algorithm for solving $Ax = b$. The input vector x_0 can be an approximate initial solution or $\mathbf{0} \in \mathbb{R}^n$.

Chapter 5

Experimental results

In this Section we present experimental results to illustrate our extended model. Following the numerical example [1, §1.4], we use a uniform linear antennas array with 10 elements, centered at the origin. Matlab code of our example is available on [13]. The spacing between the elements is half of a wavelength and the response of each element is isotropic and has unit norm. The coupling between elements is ignored. The response of the array $a : R \to C^{10}$ to a plane wave propagating at an angle θ is given by

$$a(\theta) = [e^{j\frac{-9\phi}{2}}, e^{j\frac{-7\phi}{2}}, e^{j\frac{-5\phi}{2}}, \ldots, e^{j\frac{5\phi}{2}}, e^{j\frac{7\phi}{2}}, e^{j\frac{9\phi}{2}}]\,,$$

where $\phi = \cos(\theta)$ and θ is the angle of arrival. The array response is derived from the *array factor* formula: $AF = \sum_{i=1}^{N} e^{j(i-1)\psi}$ where $\psi = kd\cos(\theta)$, d is the spacing between the elements, θ is the angle of arrival and $k = \frac{2\pi}{\text{wavelength}}$.

Three signals impinge upon the array: a desired signal s_d with SNR 20 dB at each element and two uncorrelated interfering signals s_{int1} and s_{int2} with SNR 40 dB and 20 dB respectively. The angles of arrival of the interfering signals are $\theta_{int1} = 30°$ and $\theta_{int} = 75°$. The estimated covariance is then

$$ER = Eyy^* = \sigma_d^2 a_d a_d^* + \sigma_{int1}^2 a_{int1} a_{int1}^* + \sigma_{int2}^2 a_{int2} a_{int2}^* + \sigma_n^2 I\,,$$

where

$$\frac{\sigma_d^2}{\sigma_n^2} = 10^2, \quad \frac{\sigma_{int1}^2}{\sigma_n^2} = 10^4, \quad \frac{\sigma_{int2}^2}{\sigma_n^2} = 10^2 \text{ and } Evv^* = \sigma_n^2 I\,.$$

We assume that the nominal angle of arrival θ is 45°.

5.1 Angle of arrival uncertainty

We border the actual array response into an ellipsoid $\mathcal{E}(c, P)$ whose center and configuration matrix are computed from N = 64 equally spaced samples of the array response at angles between 40° and 50°. The ellipsoid is calculated the following way. Given nominal angle of arrival θ_{nom}, array output $a(\cdot)$ and number of equally-spaced samples N, we first calculate the center c:

$$c = \tfrac{1}{N} \sum_{i=1}^{N} a(\theta_i)\,,$$

and then the ellipsoid P itself

$$P = \tfrac{1}{N} \sum_{i=1}^{N} (a(\theta_i) - c)(a(\theta_i) - c)^*,$$

where

$$\theta_i = \theta_{nom} + (-\tfrac{1}{2} + \tfrac{i-1}{N-1})\Delta_\theta\,.$$

and $\Delta_\theta = 10°$.

Figure 5.1 shows the beamformer response using three tested algorithms: Capon method, RMVB using secular equations and RMVB our Newton method. As expected, the graph shows that Capon method has a good performance at the exact angle of arrival, but small variation on the angle result in degraded performance. In contrary, the RMVB beamformer has a very good performance under the uncertainty of AOA (±5°), which is the input the problem. Furthermore, in this region the distributed beamformer has no visible performance loss.

5.2 Phase uncertainty

We use the following model to express phase uncertainty:

$$a(\theta, \rho) = [e^{j(\frac{-9\phi}{2}+\rho)}, e^{j(\frac{-7\phi}{2}+\rho)}, e^{j(\frac{-5\phi}{2}+\rho)}, ^{j(\frac{5\phi}{2}+\rho)}, e^{j(\frac{7\phi}{2}+\rho)}, e^{j(\frac{9\phi}{2}+\rho)}]\,,$$

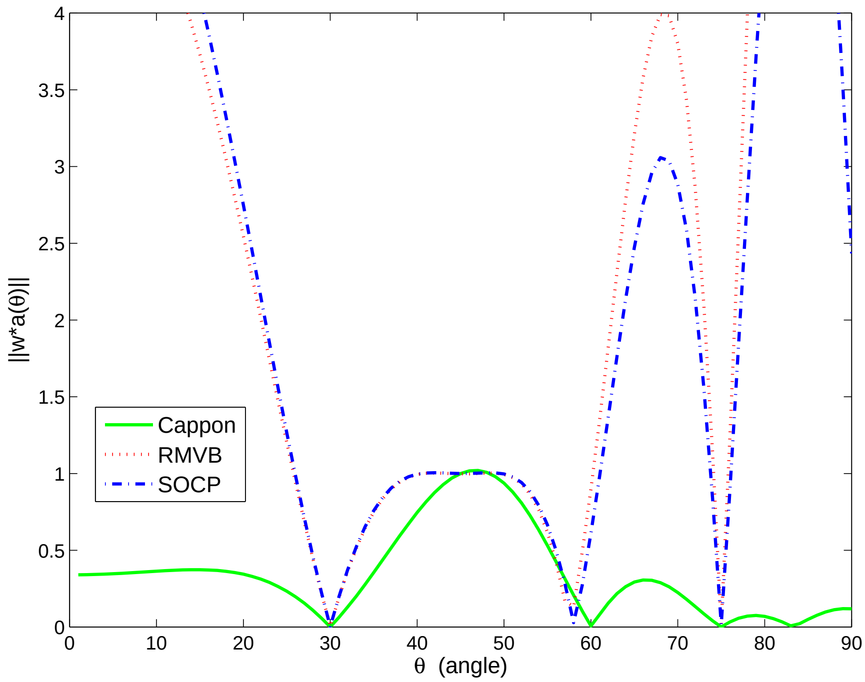

Figure 5.1: Comparison of the different methods facing angle of arrival uncertainty.

where c is the weighting constant defined accordingly to the antenna position in the array, $\phi = \cos(\theta)$ where θ is the assumed angle of arrival, and ρ is the phase uncertainty. We border the actual array response into an ellipsoid $\mathcal{E}(c, P)$ whose center and configuration matrix are computed from $\mathrm{N} = 64$ equally spaced samples of phase uncertainty on the range $[0, 0.2]$. Figure 5.2 plots the array response to different phases, where the x-axis represent the phase, and the y-axis the beamformer output. Clearly the allowed phases gives the desired response which is close to one, on the range $[0, 0.2]$.

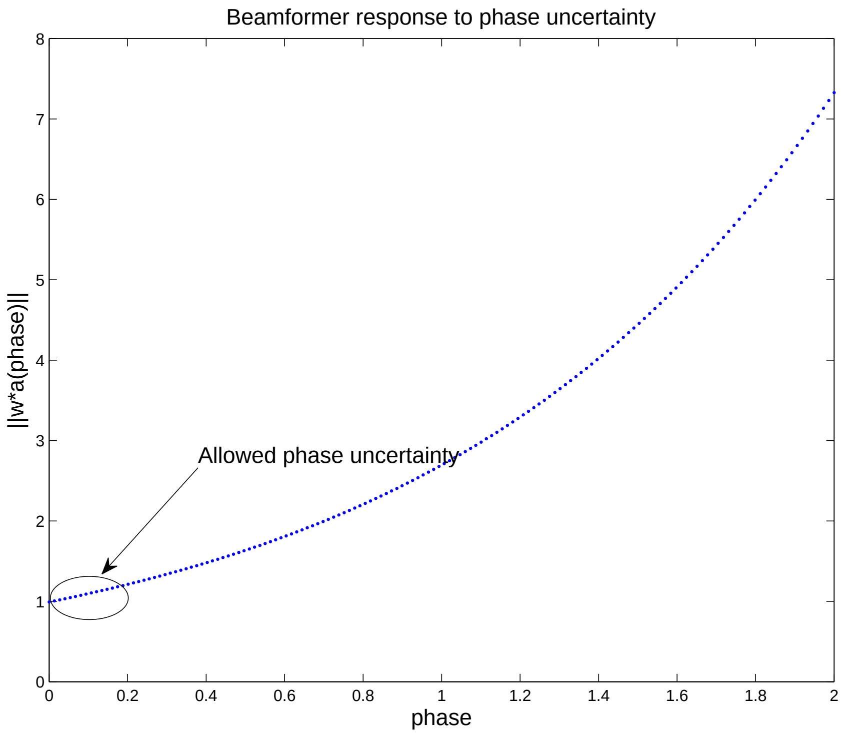

Figure 5.2: RMVB response to phase uncertainty

Chapter 6

Conclusions and Future Work

Our research showed an alternative algorithm for the RMVB problem using a distributed system solution.We also showed, that the algorithm can be robust for angle of arrival uncertainty and for phase uncertainty, which is usually the case in a real life multi-transmitter system.

The distributed solution will improve the speed and decrease the gain of the beamformer calculation and can contribute to embed such low gain systems in cell phone hardware, sensor networks, etc'.

In the future, it may be valuable to investigate new synchronization techniques for improving the phase synchronization step, which will let the beamforming system be not centralized. In addition, we will work on robustness needed as a result of combining several system uncertainties in a single setup.

Bibliography

[1] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, March 2004.

[2] L. Lorentz and S. Boyd, "Robust min variance beamforming," in *Robust Adaptive Beamforming*, P. Stoica and J. Li, Eds. ISBN 0-471-67850-3: Wiley, 2006, ch. 1, pp. 1–47.

[3] R. Mudumbai, G. Barriac, and U. Madhow, "On the feasibility of distributed beamforming in wireless networks," *IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 6, NO. 4*, 2007.

[4] G. Barriac, R. Mudumbai, and U. Madhow, "Distributed beamforming for information transfer in sensor networks." Proc. of the Third International Symposium on Information Processing in Sensor Networks, 2004, pp. 81–88.

[5] M. Johnson, M. Mitzenmacher, and K. Ramchandran, "Distributed beamforming with binary signaling." Information Theory, 2008. ISIT 2008. IEEE International Symposium on, July 2008.

[6] C. Lin, V. V. Veeravalli, and S. Meyn, "Distributed beamforming with feedback: Convergence analysis," arxiv.org/abs/0806.3023, Tech. Rep., July 2008.

[7] R. Mudumbai, B. Wild, U. Madhow, and K. Ramchandran, "Distributed beamforming using 1 bit feedback: from concept to realization." Allerton06, 2006.

[8] H. Monjardin, D. H. Covarrubias, and R. F. Nunez, "A new proposal capon beamformer for angular spreads on distributed sources in a cellular environment," *Progress In Electromagnetics Research C*, vol. 6, pp. 167–177, 2009.

[9] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Pittsburgh, PA, USA, Tech. Rep., 1994. [Online]. Available: http://portal.acm.org/citation.cfm?id=865018

[10] J. Demmel, *Applied Numerical Linear Algebra.* SIAM, 1997.

[11] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations.* SIAM, 1995.

[12] J. Nocedal and S. J. Wright, *Numerical Optimization.* Springer, 1999.

[13] Gaussian Belief Propagation implementation in matlab [online] `http://www.cs.huji.ac.il/labs/danss/p2p/gabp/`.