

ABSTRACT

Title of dissertation: Optimal Space-Time-Frequency
Design of Microphone Networks

Yenming Mark Lai, Doctor of Philosophy, 2014

Dissertation directed by: Professor Radu V. Balan
Department of Mathematics

Consider a sensing system using a large number of N microphones placed in multiple dimensions to monitor an acoustic field. Using all the microphones at once is impractical because of the amount of data generated. Instead, we choose a subset of D microphones to be active. Specifically, we wish to find the D set of microphones that minimizes the largest interference gain at multiple frequencies while monitoring a target of interest. A direct, combinatorial approach – testing all N choose D subsets of microphones – is impractical because of problem size. Instead, we use a convex optimization technique that induces sparsity through a l_1 -penalty to determine which subset of microphones to use. Our work investigates not only the optimal placement (space) of microphones but also how to process the output of each microphone (time/frequency). We explore this problem for both single and multi-frequency sources, optimizing both microphone weights and positions *simultaneously*. In addition, we explore this problem for random sources where the output of each of the N microphones is processed by an individual multirate filterbank. The N processed filterbank outputs are then combined to form one final signal. In this case, we fix all the analysis filters and optimize over all the synthesis filters. We show how to convert the continuous frequency problem to a discrete frequency approximation that is computationally tractable. In this random source/multirate filterbank case, we once again optimize over space-time-frequency *simultaneously*.

Optimal Space-Time-Frequency Design of Microphone Networks

by

Yenming Mark Lai

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2014

Advisory Committee:

Professor Radu V. Balan, Chair/Advisor

Professor John J. Benedetto

Professor Howard Elman

Professor Patrick Fitzpatrick

Professor James Yorke

© Copyright by
Yenming Mark Lai
2014

To April and Richard Lai, my loving parents,

Acknowledgments

I would first like to thank my advisor Radu Balan. It has been an incredible experience being under your tutelage for these last three years. I appreciate the countless hours you have invested in me and am so very grateful to have had such a kind and intelligent advisor. I carry your intellectual banner onto Texas!

Thanks to John Gilbert for encouraging me to start my graduate school journey six years ago. Thank you for believing in me.

Thanks to John Benedetto for his excellent coaching throughout these past six years. We know who is boss.

Thanks to Howard Elman for mid-class stretch breaks and giving me a chance to redeem myself on your final. Your coaching has helped me more than you can imagine.

Thanks to Jim Yorke for always hearing me out. You are a role model in scholarship and leadership, and it is difficult to walk away from a conversation with you without learning something new.

Thanks to Giovanni Forni for his patient office-hours tutoring in Complex Analysis my very first semester. You are a brilliant teacher and scholar.

Thanks to Bob Warner for cutting a clear path through the analysis jungle for uninitiated new comers.

Thanks to Mike Fitzpatrick for always lending me his ear. I will miss seeing your open door on the third floor.

Thanks to Sennur Ulukus for her helpful advice and modeling clear and excellent teaching. I still remember 667 and the pack of cookies you brought to our makeup class!

Thanks to Rama Chellappa for great stories and insights. Your work sets the bar in scholarship.

Thanks to Min Wu for a tremendous DSP class. I learned so much.

Thanks to Jeff Henrikson for his fantastic technical support. My thesis would have no computational meat without your ninja skills. Sorry about flipping the power switch!

Thanks to Eitan Tadmor for graciously providing CSCAMM resources. My large scale computations would not have been possible without the CSCAMM cluster.

Thanks to Michael Grant for providing commercial grade optimization software for free to academics. Your toolbox is helping the research community press into unknown territory.

Thanks to Dan Mote for taking the time to meet with a graduate student. I would wrestle with choices A and B and somehow you always suggested C, the superior option.

Thanks to Stephen Trachtenberg for his leadership insights. It is such a treat to sit in front of a master.

Thanks for Dave Mount for helpful coaching and great tennis practices. I learned how to volley playing with you!

Thanks to Razi Haimi-Cohen for his excellent mentoring at Bell Labs and giving me my first taste of research. I respect and appreciate your high standards, work ethic, and integrity.

Thank to Terri Rogers for her support and advice at Bell Labs. I appreciate your always open door!

Thanks to Heiko Claussen for a super fast computer and always helpful suggestions. You are both a great teammate and leader.

Thanks to Justinian Rosca for great vision and helping me to see the bigger picture. We need to play tennis again!

Thanks to Lex McCusker for your NSF I-Corp mentorship. You have fundamentally altered how I see both research and entrepreneurship and broadened my horizons. I learned so much working side-by-side with you.

Thanks to Karl Ginter for knowing everything about everything. I lean heavily on your wisdom and think so much of your leadership. Thanks for showing me the first few steps in how to hunt.

Thanks to Nate Strawn for helping me make decisions at crucial crossroads. Your advice was invaluable and always right.

Thanks to Stan Reeves and Chris Rozell for excellent coaching during my job hunt. I respect the way you have balanced both career and family.

Thanks to a special group of professors at Rice University who inspired my love for mathematics and engineering. Behnaam Aazhang, you are both smart and savvy and also a great leader. Rich Baraniuk, your enthusiasm is infectious and your teaching and research lights so many fires! Don Johnson, your class first taught me what it means to think really hard and those early lessons have carried me far.

Thanks to a group of guys who helped me hack it through Rice – Shay Harnoy, Patrick MacAlpine, Juan Navarro, and Gary Printy. I am sad to see BCS come to close. I am so proud to have known and worked with each of you when we were all clueless undergraduates and look forward to seeing the impacts each of you will have in your respective professions. (Shay - Fortune 500 CEO, Patmac - brilliant technical lead, Juan - Federal Reserve executive, Gary Printy, the guy who pulls strings around town!) Also, Shay, is TomNod going to pay for our dinner sometime?

Thanks to Jim Tour for modeling scholarship, faith, and boldness and Shireen Tour for providing warmth and hospitality week after week. Your lives are powerful testimonies.

Thanks to Robert Newsome for his steadfast friendship and adventuresome spirit. This thesis would not have been written without your help and hospitality.

Thanks to Sarah Berkey for her support and friendship. I am grateful our lives intersected in Maryland.

Thanks to Ryan Penley and Maryland Cru for the incredible encouragement over these last six years. I can't wait to see what is in store for the next six years.

Thanks to the guys who were crazy enough to slog it out in the gym and on the track with me: Alex Cloninger, James Murphy, Ryan Hunter, Avinash Sahu, Kevin Galloway, Daniel Yin, Chao-wei Chen, Mike Fahey, Pat Brown, Stefan Zavalin, Casey Allred, and Mike Casto. You each have a little bit of iron in your souls.

Thanks to Mike Casto and Kyungjin Yoo for running countless tennis drills with me. I am working hard to get up to a 4.5 rating in Texas!

Thanks to the MATH bible study group: Josh Ballew, Victoria Taroudaki, Bryant Angelos, Clare Wickman, Rui Can, Jonathan Fernandez, Christiana Sabett, Asia Wyatt, Franck Olivier Ndjakou Njeunje. Studying and praying with you has been such nourishment.

Thanks to Kevin Offner and the Intervarsity Graduate Fellowship group. I have learned so much about reconciling faith and work in my day to day life.

Thanks to my housemates and friends: Kevin Galloway (captain worthy of respect), Matt Videon (athlete, friend, and hunter of White Buffalo), Eli Tung (thoughtful friend and super chef), Matt Buttner (County, country, and a good guy), Cain Jeffries (good music and good chicken!), Sebastian Van Neste (leader, comrade, and always striving for the best), Andrew Flavin (thinker, diplomat, and brother), Jim Spatz (friend and roommate of the year), Troy Stevens (fun train), Esrael Seyoum (Pad Thai and man of faith), Ruben Amalalo (Are sure Ghanaians are not Asians?), Sheldon Neuberger (John Mayer, expert computer scientist, and a chill guy), Jadon Ramsing (soccer star with a kind heart), Jeremy Perry (amazing voice and caring teacher), Emmanuel Yeboah (extraordinary work ethic and built like a rock), and Matt Shriver (faithful in everything he does). 5010 will always have a special place in my heart because of you. Great times, great food, and great stories to tell the kids! When's the next housecleaning?

Thanks to Kevin Moore for teaching me how to teach. I deeply respect your practice and am thankful for your friendship.

Thanks to Terrence Long for helping me get through MATH631. I had no business being in that class and would not have made it through without your patient tutoring.

Thanks to AJ Hergenroeder for his steadfast friendship and constant encouragement. I love your intensity and spirit brother.

Thanks for Austin Bratton for always checking in. You are a faithful friend.

Thanks for Justin Lo for his help small and large. You are always watching out for me and am very grateful for your friendship.

Thanks to Jim James McTuttle for helping me get back to Texas. I am grateful for your friendship and the way it challenges me.

Thanks to Jonathan Lugo for all the on-the-fly coding coaching. It has been wonderful to chat with both an expert and friend.

Thanks to Joaquin Guadarrama. You are a friend in stormy times and would not have made it through Lanier without your help. You have my vote, literally and figuratively.

Thanks to Gilbert Wang for cooking a feast for my going away party. You are going to make a woman very happy one day, and without a doubt, you are solid.

Thanks to William Mantzel for giving me both backroom insights into academia and thoughtful suggestions.

Thanks to Bill Bill Hodges for helping me on a very important drive from DC to Nashville and constant encouragement through the years. Have you looked at the Comp210 assignment yet?

A special thanks to Alverda McCoy. You are wise, kind, and gracious and mom to so many students. I am so grateful to have been under your leadership for five years. I will dearly miss our regular chats and your wise counsel. Thank you for your support, time, and prayers. AMSC is so

blessed to have you on board.

Thanks to Konstantina Trivisa for taking a chance and admitting a graduate student mid-year. You have incredible vision and energy and a warm heart. I will miss hearing your voice in the hallways, and I anticipate big things in your future!

Thanks to Angel and Chang Ho for their steadfast support and gracious hospitality. Indianapolis has a special place in my heart, especially Angel's refrigerator and cupboard.

Thanks to Mike and Jenny Holberg for their support and prayers. It has been a joy to see your family grow and grow up!

Thanks to my mom April Lai for her faithful prayers and encouragement through these years. You are a source a strength.

Thanks to my dad Richard Lai for always modeling diligence, patience, and integrity. You have repeatedly earned my respect.

Finally, thanks to Elizabeth Matthews for her constant support and encouragement. You hold up half the sky. I hope one day to dedicate a book to you.

List of Tables

3.1	Number of separated minima of criterion J for $D = 5$ active microphones out of N and a neighborhood radius R shows the vast majority of combinations are nonoptimal locally.	28
3.2	Comparison λ -method vs. global minimum.	29
3.3	WTIG for the λ -method and SA	31
4.1	Worst interference gains[dB] for: Beamformer, MultiFreq λ -method, SA Multi-Freq λ -method, SingleFreq λ -method, SA SingleFreq λ -method.	48

List of Figures

3.1	Sample Scenario: $L = 193$ sources (1 target and 192 interferences), $N = 200$ microphones. The 200 microphones are placed on the walls of a 10 m by 8 m room. To monitor the target located at (3m, 4m), we wish to use only V microphones instead of all N so as not to overwhelm our central processing unit with data. If we set $V = 5$, there are $\binom{N}{V} = \binom{100}{5} \approx 7.5 \times 10^7$ possible sets of 5 microphones to use.	14
3.2	Gain pattern of a 32 element microphone array at 1000 hz found by λ -method. The worst target to interference gain is 1.5 dB. The array is down-selected from 1000 uniform distributed microphones.	30
3.3	Worst target to interference gain for a number of active microphones and different target locations. 32 microphones are sufficient to achieve performance similar to 1000 microphones. The sufficient number of microphones is dependent on the target location.	31
3.4	Gain pattern of a 32 element microphone array at 1000 hz found by Simulated Annealing. The worst target to interference gain is 0.1 dB. The search was initialized by the result of the λ -method	32
4.1	Sample Scenario: $L = 193$ sources (1 target and 192 interferences), $N = 200$ microphones. The 200 microphones are placed on the walls of a 10 m by 8 m room. To monitor the target located at (3m, 4m), we wish to use only V microphones instead of all N so as not to overwhelm our central processing unit with data. If we set $V = 5$, there are $\binom{N}{V} = \binom{100}{5} \approx 7.5 \times 10^7$ possible sets of 5 microphones to use. The L sources are assumed to be broadband, that is each source consists of the F frequencies f_1, f_2, \dots, f_F . We allow the N filter weights to vary for each of the F frequencies and hence our optimization problem returns a set of $F \cdot N$ filter weights.	37
4.2	Gain map for classical beamformer that maximizes SNR at 1000 Hz	49
4.3	Gain map for beamformer found by multi-frequency λ -method at 1000 Hz	50
4.4	Comparison of mean and variance of maximum gain for varying number of randomly placed interferences at 1000 Hz	50
4.5	Ordered weights found by multi-frequency λ -method	50
5.1	Single channel of multi-rate filterbank	52

5.2	Filterbank for microphone n . Notice that D , the downsampling and upsampling factor, is not necessarily equal to C , the number of channels (subbands).	54
6.1	A single acoustic source x_r propagates to N microphones, and the output of each the N microphones is processed by an individual multi-rate, finite-impulse response (FIR) filterbank. The subscript r of x_r indexes the $I + 1$ sources which consist of I interference sources, whose overall gain we wish to minimize, and 1 target source, whose gain we wish to be exactly 1 so that our system perfectly reconstructs the target. We index the target source by setting $r = 0$, and hence the target source is denoted by x_0 . The propagation from source x_r to filterbank n is modeled with transfer function $P_{r,n}$. The prefilter $I_{0,n}$ inverts the propagation from the target source, x_0 , to filterbank n . The cascade of the propagation filter $P_{r,n}$ and target inversion filter $I_{0,n}$ is denoted as $H_{r,n}$. The output of the N filterbanks are summed to give a processed signal y_r	59
6.2	FIR multi-rate filter bank for n -th microphone	63
6.3	A single acoustic source x propagates to N microphones, and the output of each the N microphones is processed by an individual multi-rate, finite-impulse response (FIR) filterbank. The propagation from source x to filterbank n is modeled with transfer function P_n . The prefilter $I_{0,n}$ inverts the propagation from the target source, x_0 , to filterbank n . The cascade of the propagation filter P_n and target inversion filter $I_{0,n}$ is denoted as H_n . The output of the N filterbanks are summed to give a processed signal y_r	67
8.1	Room environment.	104
8.2	Performance on optimization grid consisting of 1240 interferences in db.	105
8.3	Maximum magnitude of each subchannel's synthesis filter for the λ value that produced the desired number of subchannels.	106
8.4	Maximum magnitude of each subchannel's synthesis filter after debiasing.	107
8.5	Locations of active low frequency subchannels.	108
8.6	Locations of active high frequency subchannels.	109
8.7	The synthesis filters frequency responses of a filterbank whose low frequency subchannel is only active.	109
8.8	The synthesis filters frequency responses of a filterbank whose high frequency subchannel is only active.	110
8.9	The synthesis filters frequency responses of a filterbank whose subchannels are both active.	110
8.10	The synthesis filters frequency responses of a filterbank whose subchannels are both inactive.	111
8.11	Performance on evaluation grid consisting of 4960 interferences in db	111

List of Abbreviations

DFT	Discrete Fourier Transform
DTFT	Discrete Time Fourier Transform
WSS	Wide Sense Stationary
WSCS	Wide Sense Cyclostationary

Chapter 1: Overview

In this work, we explore the design of audio systems to monitor targets in complex environments. We propose algorithms to not only choose the placement of microphones but also how to process each of the microphones sampled signals to monitor a target while attenuating other interfering sources.

This work is primarily motivated by industrial needs. For example, engineering managers are interested in monitoring specific bearings on a wind turbine, car manufacturers interested in the sound of a specific piston, and train conductors interested in detecting aberrant sounds in a specific wheel set.

These monitoring problems would be trivial if any of the following conditions applied:

- Microphones can be placed adjacent to the target of interest.
- The target source is generated in a quiet or interference-free environment.
- The interference sources' location and signature are known.

Each of the three previous use cases violate some if not all these conditions.

We will assume our environment contains either a large number of interferences with known locations or a small number of interferences with unknown locations. In addition, we will assume we are limited to using only N microphones because of bandwidth constraints.

Furthermore, to make our problem computationally tractable, we will discretize possible

microphone locations so that there is a finite set of possible microphone locations. Choosing a fixed number of microphone locations from a set of possible microphone locations is by definition a combinatorial problem and for even a moderate size problem, the number of possibilities can be overwhelming. For example, there are approximately 10^{60} possibilities when choosing 32 microphone locations from a possible set of 1000 microphone locations. In comparison, there are approximately 10^{50} atoms that make up planet Earth.

To make reasonable comparisons in our simulations, we conduct exhaustive searches on small problem setups to discover the global optimum for each setup and compare our algorithm, a convex optimization scheme, against these global “bests”.

This work consists of three parts, with each subsequent part building on previous parts’ ideas and algorithms. The content of each part, however, is self-contained and can be read individually without consulting other parts.

1.1 Part I: Processing Single Frequency Source Using Delay-Scale-Sum Beamformer

In part I, we explore the case where the acoustic source consists of a single frequency. We propose an algorithm to find not only the microphone locations but also the corresponding beamforming weights. In other words, we both place the microphones and offer algorithms to process the sampled output of each the microphones.

Our processing scheme is a basic delay-scale-sum beamformer. Our system takes the output of each of the N microphones, applies a chosen delay and amplitude scaling to each of the microphone’s output and then sums up the N processed signals to give a final output. In the frequency domain, this delay and scaling beamforming weight can be represented as simply a scaled complex exponential.

In practice, the processing system works as follows. Each of the N microphones samples

the continuous time signal at the appropriate sampling rate (\geq Nyquist). A Discrete Fourier Transform (DFT) of sufficient length to achieve the needed frequency resolution is then taken on each of the N streams of discrete samples. If the original source signals consisted only of a pure tone and the correct sampling rate and DFT length were chosen, the DFT transform should produce an output of DFT coefficients with only one non-zero entry. For each of the N sets of DFT coefficients, the system multiplies the computed beamforming weight at the non-zero frequency bin. We note that the beamforming weights can vary for each of the N processing streams.

The output of the N processing streams is then summed to give the final result.

1.2 Part II: Processing Broadband Source Using Delay-Scale-Sum Beamformer

Part II is a direct extension of Part I. Here, we explore the broadband case, where signals are assumed to be a sum of F narrowband signals. Our algorithm again finds the optimal placement of microphones but now also computes beamforming weights *for each of the F frequencies of interest*.

If the original source signals consisted only of a sum of F pure tones and the correct sampling rate and DFT length were chosen, the DFT transform should produce an output of DFT coefficients with only F non-zero entries. Our algorithm computes beamforming weights for each of the F non-zero entries for each of the N processing streams.

1.3 Part III: Processing Broadband Source Using Multi-rate Filterbanks

Part III generalizes the microphone placement and processing scheme of Part I and Part II. In this part's setup, we now have N multi-rate filterbanks, each processing the corresponding output of one of the N microphones. Each of the N filterbanks decomposes the discrete input into C subchannels, and hence our system uses a total of $N \cdot C$ subchannels. The previous constraint

of Part I and Part II of using only N microphones because of bandwidth limitations can now be generalized to the constraint of using only $N \cdot C$ subchannels. We can now use the output of between N and $N \cdot C$ microphones and still not violate our bandwidth constraint by selecting which subchannels to process in each filterbank. By using N microphones and each of the N microphones C subchannels, we fulfill our bandwidth constraint of $N \cdot C$ subchannels. By using $N \cdot C$ microphones but only choosing to use one subchannel of each of the microphones, we also fulfill our our bandwidth constraint of $N \cdot C \cdot 1$ subchannels.

In other words, instead of choosing the placement of N microphones out of a set of P possible microphone locations, we choose a subset of $N \cdot C$ subchannels to use out of a possible $P \cdot C$ subchannels. This type of processing scheme would be appropriate when deploying microphones is relatively inexpensive but the transfer of the collected data of each of the microphone is expensive.

In multi-rate filterbanks, each subchannel is processed by both an analysis and a synthesis filter. We fix the analysis filters to reduce the computational complexity and instead compute only the synthesis filters for each of the chosen $N \cdot C$ subchannels. Computing the filters for the multi-rate filterbanks generalizes computing beamforming weights in Part I and Part II.

The DFT and IDFT implementation of Part I and Part II can be interpreted as the analysis and synthesis filtering respectively. The choice of beamforming weights of Part I and Part II now correspond to the choice of the synthesis filters in Part III.

Chapter 2: Notation

We denote both the label of and the frequency response of the filter by capitalization of a letter. We denote the impulse response by the lowercase of the same letter. Hence a filter F has transfer function F and impulse response f . Summation over all possible indices t is indicated by placing the index variable t below the summation, that is

$$\sum_t = \sum_{t=-\infty}^{\infty}$$

The complex number $\sqrt{-1}$ is denoted by the letter j , in contrast to the letter i . Hence, we write Euler's formula as $e^{j\theta} = \cos(\theta) + j \sin(\theta)$. The complex conjugate of complex number is denoted by a bar over the number. Hence we have, for all real a and b ,

$$\overline{a + jb} = a - jb$$

Parameters that are continuous are placed in parenthesis whereas parameters that are discrete are placed in brackets. Hence, the function $f(t) = \sin(t), 0 \leq t < 2\pi$ has a continuous parameter t whereas $f[t] = 3t, t = 0, 1, 2, \dots, 5$ has a discrete parameter t . We denote the z -transform of a discrete signal $r[k]$ as $R(z) = \sum_k r[k]z^{-k}$.

The formula for the sum of the first D terms of a geometric series with ratio $r \neq 1$

$$\sum_{k=0}^{D-1} r^k = \frac{1 - r^D}{1 - r}$$

yields the useful identity

$$\sum_{k=0}^{D-1} e^{-j2\pi \frac{n}{D}k} = \begin{cases} D & \text{if } n \text{ is an integer multiple of } D; \\ 0 & \text{otherwise.} \end{cases} \quad (2.1)$$

In addition, we will let W_D be the D -th root of unity, that is

$$W_D = e^{-\frac{2\pi j}{D}} \quad (2.2)$$

The delta sequence with a single parameter m , δ_m , is defined as follows

$$\delta_m = \begin{cases} 1 & \text{if } m = 0; \\ 0 & \text{otherwise.} \end{cases} \quad (2.3)$$

The Kronecker delta symbol with two parameters m, n , $\delta_{m,n}$ is defined as follows

$$\delta_{m,n} = \begin{cases} 1 & \text{if } m = n; \\ 0 & \text{otherwise.} \end{cases} \quad (2.4)$$

Matrices and in turn vectors will be denoted in bold face. Hence, \mathbf{A} refers to a matrix \mathbf{A} .

The i -th, j -th entry, the entry of the i -th row and j -th column, of a matrix is denoted as $(\mathbf{A})_{i,j}$.

Indices of matrixes will begin with 0. Hence $(\mathbf{A})_{2,0}$ refers to the entry in the third row and first

column of the matrix \mathbf{A} . $\overline{\mathbf{A}}$ denotes the matrix \mathbf{A} but with each entry complex conjugated. Hence,

we have

$$(\mathbf{A})_{i,j} = \overline{(\overline{\mathbf{A}})_{i,j}}$$

\mathbf{A}^T refers to the transpose of the matrix \mathbf{A} . Hence, we have

$$(\mathbf{A})_{i,j} = (\mathbf{A}^T)_{j,i}$$

\mathbf{A}^* refers to the conjugate transpose of the matrix \mathbf{A} where \mathbf{A}^* is obtained by taking the transpose of \mathbf{A} and then taking the complex conjugate of each entry. Hence we have

$$\mathbf{A}^* = \overline{\mathbf{A}^T}$$

The z -transform of a discrete function $x[t]$, $X(z)$, is defined as follows

$$X(z) = \sum_t x[t]z^{-t}$$

The Discrete Time Fourier Transform (DTFT) of a discrete function $x[t]$, $\mathbb{X}(w)$, is defined as follows

$$\mathbb{X}(w) = \sum_t x[t]e^{-jw t}$$

We note that if we set $z = e^{jw}$ in the z -transform, we have the DTFT, that is

$$X(e^{jw}) = \sum_t x[t](e^{jw})^{-t} = \mathbb{X}(w) \quad (2.5)$$

Using (2.5), we have

$$\mathbb{X}(w + 2\pi) = \sum_t x[t](e^{j(w+2\pi)})^{-t} = \sum_t x[t](e^{jw})^{-t} \underbrace{e^{-j2\pi t}}_1 = \mathbb{X}(w) \quad (2.6)$$

In other words, the DTFT is 2π periodic. If we again set $z = e^{jw}$ in the z -transform, we have the following relationship

$$X(zW_D^d) = X(e^{j(w-2\pi\frac{d}{D})}) = X(e^{j(w-w_{D,d})}) = \mathbb{X}(w - w_{D,d}) \quad (2.7)$$

using the shortened notation

$$w_{D,d} = 2\pi \frac{d}{D} \quad (2.8)$$

Equation(2.7) connects the z -transform notation and the DTFT notation when $z = e^{jw}$. Without loss of generality, we will assume that the z -transform series converge in a region around the unit circle e^{jw} except possibly for a finite number of points. We will use the z -transform notation exclusively in this paper, with the assumption $z = e^{jw}$.

Convolution between two discrete functions f and g , evaluated at time t , is defined as follows

$$(f * g)[t] = \sum_{\tau} f[t - \tau]g[\tau]$$

A change of variable shows that convolution is commutative, that is

$$(f * g)[t] = (g * f)[t]$$

A change of variable also shows that if $a[t] = (f * g)[t]$, then

$$A(z) = F(z)H(z)$$

$$\mathbb{A}(w) = \mathbb{F}(w)\mathbb{H}(w)$$

In other words, convolution in the time domain becomes a product in the z -domain and Fourier domain.

The Hilbert-Schmidt norm for a matrix \mathbf{A} , size $M \times N$, is denoted as $\|A\|_{HS}^2$ and is defined as follows:

$$\|\mathbf{A}\|_{HS}^2 = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |(\mathbf{A})_{m,n}|^2 \quad (2.9)$$

The Hilbert-Schmidt inner product for matrices \mathbf{A} and \mathbf{B} , both of size $M \times N$, is denoted as $\langle \mathbf{A}, \mathbf{B} \rangle_{HS}$ and is defined as follows:

$$\langle \mathbf{A}, \mathbf{B} \rangle_{HS} = \text{Tr}(\mathbf{A}\mathbf{B}^*) \quad (2.10)$$

Part I

Processing Single Frequency Source Using Delay-Scale-Sum Beamformer

Chapter 3: Processing Single Frequency Source Using Delay-Scale-Sum Beamformer

3.1 Overview

The main goal of this work is to adaptively employ a large set of microphone sensors distributed in multiple dimensions to scan an acoustic field. Processing data from a large set of sensors will necessarily involve intelligent definition of suitable subsets of sensors active at various times. This paper presents a novel method for optimal beam pattern design for large scale sensor arrays using convex and non-convex optimization techniques to define optimal subsets of sensors capable to select a target location while suppressing a large number of interferences. The first of two optimization techniques we present, uses a LASSO-type approach to convexify the corresponding combinatorial optimization problem. The second approach employs simulated annealing to search for optimal solutions with a fixed size subset of active sensors. Our numerical simulations show that for scenarios of practical interest, the convex optimization solution is almost optimal.

3.2 Introduction

Consider a large scale microphone array having N microphones that monitors a surveillance area. Here N could be in the order of thousands. Data could be collected from all microphones, processed locally, and further communicated to a central processing unit. For example, for $N = 1000$ microphones and a data sampling rate of 100000 samples per second, the bandwidth requirements for a system performing exhaustive sampling is 1Gsamples/sec, an overwhelming amount of data for even current hardware systems. Alternatively, the central processing unit could implement a policy of sparse spatial sampling where only a subset of microphones is polled for data at any time. This may be desirable for practical reasons: cost (in power consumption, communication or processing bandwidth) is simply too high for an exhaustive strategy. This part advocates the need for advanced, dynamic control and signal processing of what data should be manipulated in place of the full extent of the data, and it introduces novel strategies and formal analysis of what can be achieved when only a sparse subset of microphones is sampled simultaneously.

Assume a surveillance area that consists of a set of point sources that we wish to distinguish from one another. Our task is to sample any given source location using a maximum of V microphones of the very large array ($V \ll N$). We then process the raw data from the V microphones to estimate the point source at that location. Our system can then scan the room by using different V -size subsets of microphones for each possible source location at different times. By doing so, the central processing unit is processing data from V microphones at a time and not all N microphones at a time. Figure 3.1 provides a possible setup scenario. We will thus design and analyze sparse spatial sampling strategies with near-optimal performance. Specifically, we develop algorithms that minimize the p -norm of interference gains with $p = 1, 2, \infty$ while reconstructing the

target source perfectly. Conceptually, the system works in two steps. First, the discovery step estimates locations of all point-like sources. Second, the system separates source signals, one source at a time.

Below we define the problem in precise terms. Throughout this part, we assume the following hypothesis:

H1. The total number of microphones N and their locations (x, y, z -coordinates) are known.

H2. The total number of sources L and their locations (x, y, z -coordinates) are known.

H3. The maximum number of microphones active at any time is fixed in advance and denoted by V . In general $V \ll L$.

H4. Signals are narrow band and operate on the same frequency f .

In practice, the number of sources and their locations are often unknown. By assuming a large number of known sources and locations, our system can search each possible source location to detect whether or not a real source is present.

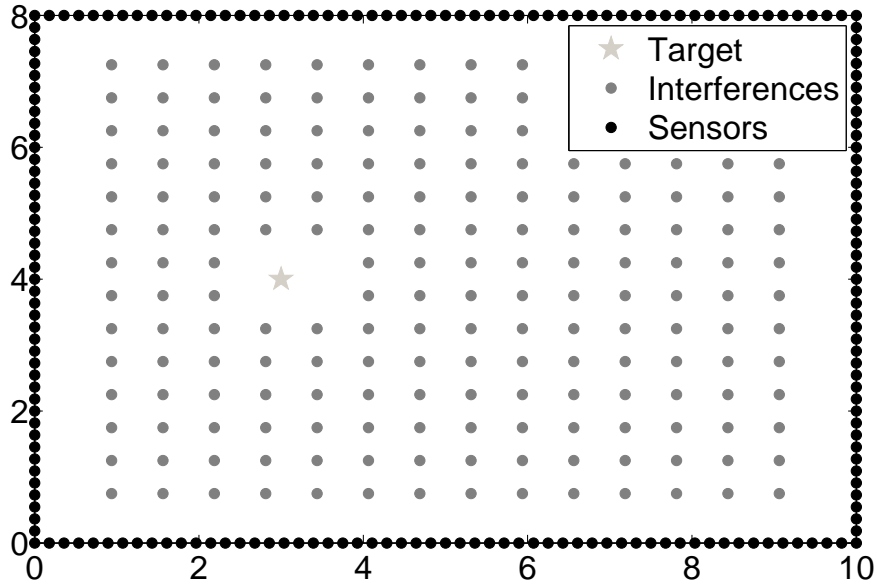


Fig. 3.1. Sample Scenario: $L = 193$ sources (1 target and 192 interferences), $N = 200$ microphones. The 200 microphones are placed on the walls of a 10 m by 8 m room. To monitor the target located at (3m, 4m), we wish to use only V microphones instead of all N so as not to overwhelm our central processing unit with data. If we set $V = 5$, there are $\binom{N}{V} = \binom{200}{5} \approx 7.5 \times 10^7$ possible sets of 5 microphones to use.

3.3 Problem formulation

Consider the setup described earlier and sketched in Figure 3.1. We assume that both the N microphone and L source locations are known. We use the direct-path model to model the propagation from source to microphone, and hence the transfer function between source l and microphone n at frequency f is given by

$$H_{l,n}(f) = \frac{e^{-2\pi i f c \|r_l - r_n\|}}{\|r_l - r_n\|}, \quad 0 \leq l \leq L - 1, 0 \leq n \leq N - 1 \quad (3.1)$$

where r_n, r_l denote the position vectors of microphone n and source l respectively, and c is the wave propagation speed, which in our case is the speed of sound, 343 m/sec.

Let $x_n(\tau)$ denote the continuous time-domain signal received by microphone n . The short-time Fourier transform (STFT) of x_n , $X_n(f, t)$, is related to the STFT of source signals $S_l(f, t)$ as follows

$$X_n(f, t) = \sum_{l=0}^{L-1} H_{l,n}(f) S_l(f, t) + \nu_n(f, t) \quad (3.2)$$

where $\nu_n(f, t)$ is the microphone n noise, f denotes the frequency, and t denotes the time frame index. In other words, for fixed frequency f and time frame index t , the signal received by microphone n , in the frequency domain, is a sum of L signals and the noise of microphone n , ν_n . Each of the L signals received by the microphone originates from one of the L sources, S_l , and the propagation of the source signal S_l to microphone n is modeled by multiplying the source signal S_l by transfer function $H_{l,n}$.

Our system estimates *one source signal at a time*. Hence, one of the sources is declared the target, and the remaining sources are declared interferences. Our system can then estimate all the L sources by repeating this process L times – choosing a target source, declaring the rest of the sources as interferences, and then estimating the target source.

Our system is a delay-scale-sum beamformer. In other words, the input to each microphone, x_n , is delayed and scaled by some variable amount, and then all the delayed and scaled microphone signals are summed up to give a single final output signal, y . By choosing an appropriate delay and scaling for each microphone, we wish this final output signal to contain an undistorted target source signal and attenuated versions of the other $L - 1$ interference source signals.

The delay and scaling that microphone n performs can be modeled in the frequency domain as multiplication by a weighted complex exponential or equivalently multiplication by a complex number, $w_n(f)$. **A key observation is that if $w_n(f) = 0$, then microphone n is inactive at frequency f .**

For simplicity of notation, we assume the zero indexed source is the target and the remaining

$L - 1$ sources are interferences. The STFT of the final output y is given by

$$\begin{aligned} Y(f, t) &= \sum_{n=0}^{N-1} w_n(f) X_n(f, t) \\ &= \sum_{l=0}^{L-1} \left(\sum_{n=0}^{N-1} w_n(f) H_{l,n}(f) \right) S_l(f, t) + \sum_{n=0}^{N-1} w_n(f) \nu_n(f, t) \end{aligned} \quad (3.3)$$

We observe from (3.3) that the response factor of source l , S_l , is given by

$$R_l(f) = \sum_{n=0}^{N-1} w_n(f) H_{l,n}(f). \quad (3.4)$$

We note that $R_l(f)$ is a complex scalar that shows how our delay-scale-sum beamformer processes source l . By taking the modulus of $R_l(f)$, we get the gain $K_l(f)$, that is

$$K_l(f) = |R_l(f)| \quad (3.5)$$

The gain $K_l(f)$ is a real non-negative scalar that measures how much our system attenuates or amplifies a given source l . If $0 \leq K_l(f) < 1$, the system attenuates the source l and conversely if $K_l(f) > 1$, the system amplifies source l . If $K_l(f) = 1$, the system does not distort the amplitude of source l but may possibly introduce a phase shift to source l .

Ideally, for fixed frequency f , we wish to choose N beamforming weights such that $R_l(f) = 0$ for $l = 1, 2, \dots, L - 1$ and $R_0(f) = 1$. In other words, our system of N microphones completely cancels out all $L - 1$ interferences while perfectly reconstructing the target source, source 0. This is achievable in the case where there are more microphones than sources, that is $N > L$.

In general, for fixed frequency f , we wish to choose N beamforming weights such that p -norm of the $L - 1$ interference responses, $R_l(f)$ for $l = 1, 2, \dots, L - 1$, is small while still achieving perfect reconstruction of the target source, that is $R_0(f) = 1$. We denote the column vector of interference responses, size $(L - 1) \times 1$, as $\mathbf{R}_{L,(L-1) \times 1}(f)$ where

$$\mathbf{R}_{I,(L-1)\times 1}(f) = [R_1(f), R_2(f), \dots, R_{L-1}(f)]^T \quad (3.6)$$

In our setup, each of the L sources propagates to each of the N microphones where the L sources consist of 1 target and $L - 1$ interferences. For the l -th source, there are N transfer functions modeling the propagation from the l -th source to the N microphones. We denote the row vector of transfer functions for source l , size $1 \times N$, as $\mathbf{H}_{l,1\times N}(f)$ where

$$\mathbf{H}_{l,1\times N}(f) = [H_{l,1}(f), H_{l,2}(f), \dots, H_{l,N-1}(f)] \quad (3.7)$$

Since we have a total of N microphones, there are a total of $L \times N$ transfer functions. We denote the source transfer function matrix, size $L \times N$, as $\mathbf{H}_{L\times N}(f)$ where

$$\mathbf{H}_{L\times N}(f) = \begin{bmatrix} \mathbf{H}_{0,1\times N}(f) \\ \mathbf{H}_{1,1\times N}(f) \\ \mathbf{H}_{2,1\times N}(f) \\ \vdots \\ \mathbf{H}_{L-1,1\times N}(f) \end{bmatrix} \quad (3.8)$$

If we consider only transfer functions corresponding to the $L - 1$ from interferences, we can delete the first row of (3.8) to get the interference transfer function matrix, size $(L - 1) \times N$, $\mathbf{H}_{I,(L-1)\times N}(f)$ where

$$\mathbf{H}_{I,(L-1)\times N}(f) = \begin{bmatrix} \mathbf{H}_{1,1\times N}(f) \\ \mathbf{H}_{2,1\times N}(f) \\ \vdots \\ \mathbf{H}_{L-1,1\times N}(f) \end{bmatrix} \quad (3.9)$$

We can denote the column vector of all beamforming weights, size $N \times 1$, as $\mathbf{w}_{N \times 1}(f)$ where

$$\mathbf{w}_{N \times 1}(f) = [w_0(f), w_1(f), \dots, w_{N-1}(f)]^T \quad (3.10)$$

We observe from the definition of response factor, (3.4), that all $L - 1$ interference response factors, $\mathbf{R}_{I, (L-1) \times 1}(f)$, can be expressed as a matrix-vector multiply between interference transfer function matrix $\mathbf{H}_{I, (L-1) \times N}(f)$ and weight column vector of $\mathbf{w}_{N \times 1}(f)$, that is

$$\mathbf{R}_{I, (L-1) \times 1}(f) = |\mathbf{H}_{I, (L-1) \times N}(f) \mathbf{w}_{N \times 1}(f)| \quad (3.11)$$

In addition, the target perfect reconstruction requirement, that is $R_0(f) = 1$, can be compactly rewritten as $\langle \mathbf{w}_{N \times 1}(f), \bar{\mathbf{H}}_{0, 1 \times N}(f) \rangle = 1$, where the bar denotes complex conjugation, to compensate for the complex conjugation in the inner product.

Hypothesis H3 limits the number of simultaneously active microphones. Microphone n is considered active if for fixed frequency f , $w_n(f) \neq 0$. Thus an appropriate measure of the number of active microphones for fixed frequency f is the pseudo-norm $\|\mathbf{w}_{N \times 1}(f)\|_0$ defined by

$$\|\mathbf{w}_{N \times 1}(f)\|_0 = |\{n ; w_n(f) \neq 0\}| \quad (3.12)$$

where $|U|$ represents the cardinal of set U . We can now state our optimization problem as follows:

Optimal V -Sparse Beampattern Design Problem for Fixed Frequency f

$$\begin{aligned} & \min_{\mathbf{w}_{N \times 1}(f)} && \|\mathbf{H}_{I, (L-1) \times N}(f) \mathbf{w}_{N \times 1}(f)\|_p \\ & \text{subject to} && \\ & && \langle \mathbf{w}_{N \times 1}(f), \bar{\mathbf{H}}_{0, 1 \times N}(f) \rangle = 1 \\ & && \|\mathbf{w}_{N \times 1}(f)\|_0 = V \end{aligned} \quad (3.13)$$

Note that this is a non-convex optimization problem due to the l^0 pseudo-norm constraint $\|w\|_0 \leq V$.

3.4 Optimization Strategies

We study two optimization strategies for problem (4.5), which is non-convex and combinatorial by nature. First, we present a solution we call the λ -method, which solves a regularized version the problem. Second, we present a simulated annealing based method, which has global search capability.

3.4.1 The λ -method

This method is inspired by LASSO regularization [1], a regression technique that minimizes the sum of squares of residual errors subject to the l^1 norm of the coefficients being less than a constant. Furthermore, similar to the sparse signal and model estimation approach in [2], the l^0 pseudo-norm (3.12) is replaced by the l^1 norm $\|\mathbf{w}_{N \times 1}(f)\|_1 = \sum_{n=1}^N |w_n(f)|$ which is then incorporated into the optimization criterion using a Lagrange multiplier λ . The optimization problem (4.5) is first replaced by the following *convex* optimization problem which is the λ -method:

$$\begin{aligned} \min_{\mathbf{w}_{N \times 1}(f)} \quad & \|\mathbf{H}_{I,(L-1) \times N}(f) \mathbf{w}_{N \times 1}(f)\|_\infty + \lambda \|\mathbf{w}_{N \times 1}(f)\|_1 \\ \text{subject to} \quad & \end{aligned} \tag{3.14}$$

$$\langle \mathbf{w}_{N \times 1}(f), \bar{\mathbf{H}}_{0,1 \times N} \rangle = 1$$

As λ increases, the l_1 penalty term forces more and more of the weights $w_n(f)$ to have very small magnitude, which means that the corresponding microphones are very near inactive. For very large λ , the optimization problem finds a solution with exactly one non-trivial weight and hence exactly one active microphone so that the target perfect reconstruction constraint is fulfilled. As

λ decreases, the penalty term becomes less expensive and more microphones become active. At the lower limit of $\lambda = 0$, all microphones are allowed to be active. We note, however, in general, there is no need for more than $\min(N, L)$ active microphones.

We note that λ is a *data-dependent parameter* that needs to be tuned to discover the final subset of microphones that are to be used. An additional step is needed to find the corresponding microphone weights.

Bisection method to discover λ that produces V microphones

We discover the λ that produces exactly V microphones using the bisection method. The algorithm follows:

(1) Choose an upper bound b_{upper} for λ . We note that b_{upper} is user-chosen and data-dependent. The lower bound b_{lower} for λ is automatically set to 0.

(2) Set $\lambda = (b_{upper} + b_{lower})/2$.

(3) Solve the optimization problem (3.14) with the λ calculated in step (2).

(4) We measure the number of active microphones returned by the solution of step (3) as follows. We note that the the solution of (3.14) is a N -length vector of beamforming weights, i.e. complex numbers. First, we measure how strongly active microphone n is by measuring the magnitude of its beamforming weight w_n . Hence, let $g_n = |w_n(f)|$, $1 \leq n \leq N$ be the measure of activity of microphone n . Sort the sequence $(g_n)_{1 \leq n \leq N}$ so that its values are monotonically decreasing, that is $g_{n_1} \geq g_{n_2} \geq \dots \geq g_{n_N}$. In other words, sort the microphones in terms of how active they are, with the most active microphones first. If $g_{n_{V_\lambda+1}}$ is 3 orders of magnitude smaller than g_{n_1} , we say that V_λ microphones are active. In other words, if the activity of a microphone is much smaller than the activity of the most active microphone, we consider the microphone inactive.

(5) If $V_\lambda > V$, set $b_{lower} = \lambda$ and go to step (2). In other words, increase λ since there

are too many active microphones. If $V_\lambda < V$, set $b_{upper} = \lambda$ and go to step (2). In other words, decrease λ since there are too few active microphones. If $V_\lambda = V$, we are done. The set of active of microphones, \mathcal{A} , is then

$$\mathcal{A} = \{n_1, n_2, \dots, n_{V_\lambda}\} \quad (3.15)$$

Remark The above algorithm assumes monotonicity in the number of active microphones with respect to the tuning parameter λ , that is as λ increases, the number of active microphones is non-increasing. In practice, we find the assumption holds globally but not always locally. While, in general, as λ increases, the number of active microphones decreases. we find that small changes in λ sometimes leads to non-monotonic behavior. For example, a small increase in λ might increase the number of active microphones from 4 to 6 or similarly a small decrease in λ might decrease the number of active microphones from 6 to 4.

To handle these cases, we introduce these additional checks in the above algorithm:

(1) We keep track of the number of active microphones discovered by the previous λ iteration and also the previous λ value. Before entering step (5), we check if the current λ iteration produces a number of active microphones that exhibits non-monotonic behavior. We define non-monotonic behavior as an increase in the number of active microphones if λ was previously increased and, similarly, a decrease in the number of active microphones if λ was previously decreased. If non-monotonic behavior has occurred, we immediately return to step (2), and instead of computing the average of b_{lower} and b_{upper} , we choose a random number between b_{lower} and b_{upper} .

(2) To guarantee convergence of the algorithm, we also check the following two termination conditions:

- In step (2), if the difference between the new λ and the previous λ exceeds a user-defined

tolerance, terminate the algorithm.

- In step (5), if the number of iterations exceeds a user-defined maximum number of iterations, terminate the algorithm. An iteration is defined as the completion of one round of steps (2), (3), and (4).

Discovery of final weights using only V microphones

The final weights are obtained by solving a second time (3.14) restricted to the submatrix indexed by \mathcal{A} with either $\lambda = 0$, if the submatrix is well-conditioned, or the same λ otherwise. This step is known as “debiasing”. This step is needed because of the weight vector found by the final step (3), while containing V large weights also contains $N - V$ weights that are small but are still non-zero.

3.4.2 Simulated Annealing (SA)

Our second approach to solve (4.5) uses simulated annealing (SA). We note that the constraint on the number of active microphones makes (4.5) a combinatorial optimization problem. Simulated annealing is a simple randomized technique for iterative improvement introduced in [3]. SA repeatedly traverses a Markov chain by sampling the search space from a neighborhood of the current point with a probability proportional to $\exp(\frac{f(x)}{T})$ where T is a temperature parameter and f is the objective function to be minimized. The idea is that when T is large or, metaphorically speaking, when the “the iron is hot” and therefore pliable, all points in the space have approximately equal probability of being selected, and as T decreases or the “the iron cools” and is less pliable, only points in the space that are close to the current search point have approximately equal probability of being selected. The temperature functions follows an user-determined schedule of heating, cooling, and the re-heating.

If the new point discovered is better, the point is set as the current search point. A key idea

of SA is that if the new point discovered is worse, the algorithm will accept it as the current search point with some non-zero probability in order to avoid local minima.

In our case, we wish to find the subset of V microphones out of N and their corresponding weights that minimize the problem given by (4.5). In other words, we wish to find the subset of V beamforming weights out of N that minimizes the p -norm of the interference gains while maintaining target unity gain, that is achieves target perfect reconstruction. The objective function value f is then the p -norm of the $L - 1$ interference gains. If the optimization program fails for any reason, the objective function value is set to infinity.

SA generates a new random candidate solution point j in the neighborhood of the current point i , where j 's distance from the current search point (the difference of the microphone indices modulo N) is proportional to the temperature parameter. The candidate replaces the current search point if the candidate's objective function value improves, i.e. $f(j) < f(i)$. This is also done if the candidate's objective function value is worse, but only randomly with a probability given by:

$$p = \exp\left(\frac{f(j) - f(i)}{T}\right) \quad (3.16)$$

Otherwise, the candidate is rejected.

We index our microphones from 0 to $N - 1$, and our initial point is the set of V active microphone indices discovered by the λ -method. By choosing our initial point in this manner, we guarantee that SA will find a solution no worse than the solution discovered by the λ -method.

We perturb these V indices in proportion to the current temperature T . If $v \geq N$ or $v < 0$, we set $v = v \pmod{N}$. By doing so, we make all the new indices valid microphone indices but also introduce the notion that the microphones are arranged in a circular manner, that is the first microphone, microphone 0, and the last microphone, microphone $N - 1$, are only a distance 1 apart.

In our experiments, we use the following SA parameters. The *initial temperature* is set to

100, and the *annealing schedule* decreases the temperature by 5% each iteration. We limit the length of the Markov walk to a *total length* of 1200 iterations. After every 100 iterations, the annealing is restarted by setting the temperature back to its initial value of 100. Further, we rerun SA ten times.

3.5 Relation to Prior Work

In this section we compare and relate our approach to four problems in the literature: the beam pattern design, co-array design, grid-based beamforming, and compressive sampling.

Beampattern Design Lebret and Boyd [4] showed that given arbitrary microphone locations, finding the set of microphone weights that minimizes the maximum interference gain could be formulated as a convex optimization problem. They modeled sources as point sources, complex exponentials decaying as a function of distance. Specifically, the problem could be showed equivalent with a Second Order Cone Programming (SOCP) problem and thus efficiently solved by interior point methods [5].

Our approach is similar in spirit to that of Ling et al. [6]. The authors followed the approach of [4] but added an additional l^1 norm penalty to the weights. The l^1 penalty sparsifies the microphone weights and therefore microphone locations. Specifically, the l^1 penalty on weights causes many of the beamforming weights to be close to zero in magnitude. Our work differs in the following three aspects. First, we minimize p -norm of interference gains instead of the side lobes at different incident angles since we are interested in a two dimensional beam pattern. Second, our simulations consider microphone locations on the perimeter of a rectangular room, rather than only on a straight line. Third, we do not add an l^2 penalty on the weights to protect against large gains. Instead, after optimizing over a large set of microphones with a l^1 penalty, we select the microphone locations with the largest weight values and then reoptimize over this subset of

locations. During the reoptimization, we use the same l^1 penalty.

The performance of the proposed λ -method is compared against the performance of an exhaustive search, showing how close to optimal the λ -method achieves. The search space of possible microphone configurations is also analyzed by counting the number of local minima.

Co-array Design A common approach to choose microphone locations for a variety of problems is to examine the co-arrays formed by the microphone locations [7, 8]. Recently, Pal and Vaidyanathan [9] proposed an alternate method to find microphone positions with desired co-array properties through the use of nested uniform arrays. However, the theory of co-prime arrays deals principally with the one dimensional case. The optimization criteria we consider in the present paper can deal implicitly with two or three dimensional array designs.

Grid-based Beamforming Brandstein and Ward modeled an acoustic enclosure as a rectangular grid of point sources in [10]. Grids were labeled either as sources or interferences based on prior knowledge. Microphone weights were calculated to maximize the optimization criterion given by the ratio between source gains and interference gains. This prior work did not analyze the relationship between microphone placement and the optimization criterion. The authors extended their algorithm [11] to address room reverberation by allowing for sources to lie outside the room, an idea drawn from the “image” model where reflections of sources off of walls are modeled as virtual sources lying outside the room [12].

Compressive Sampling The widespread compressive sampling problem (see [13, 14]) is to minimize the l^0 pseudo-norm of a vector \mathbf{x} subject to a linear constraint $\mathbf{Ax} = \mathbf{b}$ in the absence of noise, or an inequality $\|\mathbf{Ax} - \mathbf{b}\|_p \leq \varepsilon$ in the presence of noise. Using Lagrange multipliers, this problem becomes

$$\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|_p + \lambda \|\mathbf{x}\|_0 \quad (3.17)$$

where $p \geq 1$.

Problem (4.5) can be brought to this form if we make the *additional* assumption that a specific microphone is active. For simplicity of notation, assume that we know $|w_N(f)| > 0$, that is microphone N , the last microphone is active. We know that at least one microphone is active since if no microphones were active the target unit gain constraint would not be fulfilled. Next, we solve for $w_N(f)$ from the target unit gain constraint $\langle \mathbf{w}_{N \times 1}(f), \bar{\mathbf{H}}_{0,1 \times N}(f) \rangle = 1$ and substitute back into $\mathbf{H}_{L,(L-1) \times N}(f) \mathbf{w}_{N \times 1}(f)$. If we denote $\tilde{\mathbf{w}}_{(N-1) \times 1}(f)$ as the $(N-1) \times 1$ column vector which consists of the first $N-1$ components of $\mathbf{w}_{N \times 1}(f)$, that is

$$\tilde{\mathbf{w}}_{(N-1) \times 1}(f) = \begin{bmatrix} w_0(f) \\ w_1(f) \\ \vdots \\ w_{N-1}(f) \end{bmatrix},$$

and $\mathbf{A}_{L \times (N-1)}(f)$ the $L \times (N-1)$ matrix whose l, n entry is given by

$$(\mathbf{A}_{L \times (N-1)}(f))_{l,n} = H_{l,n}(f) - \frac{H_{l,N}(f)H_{0,n}(f)}{H_{0,N}(f)},$$

and $\mathbf{b}_{L \times 1}(f)$ the $L \times 1$ column vector whose l -th entry is given by

$$(\mathbf{b}_{L \times 1}(f))_l = -\frac{H_{l,N}(f)}{H_{0,N}(f)},$$

then (4.5) becomes

$$\min_{\tilde{\mathbf{w}}_{(N-1) \times 1}(f): \|\tilde{\mathbf{w}}_{(N-1) \times 1}(f)\|_0 = V-1} \|\mathbf{A}_{L \times (N-1)}(f) \tilde{\mathbf{w}}_{(N-1) \times 1}(f) - \mathbf{b}_{L \times 1}(f)\|_p \quad (3.18)$$

which can be converted into a problem similar in form to (3.17):

$$\min_{\tilde{\mathbf{w}}_{(N-1) \times 1}(f)} \|\mathbf{A}_{L \times (N-1)}(f) \tilde{\mathbf{w}}_{(N-1) \times 1}(f) - \mathbf{b}_{L \times 1}(f)\|_p + \lambda \|\tilde{\mathbf{w}}_{(N-1) \times 1}(f)\|_0. \quad (3.19)$$

A different formulation of (3.19) is to minimize the number of sensors that achieve a given gain level τ , that is

$$\begin{aligned}
& \min && \|\tilde{\mathbf{w}}_{(N-1) \times 1}(f)\|_0 \\
& \tilde{\mathbf{w}}_{(N-1) \times 1}(f) && \\
& \text{subject to} && \\
& \|\mathbf{A}_{L \times (N-1)}(f)\tilde{\mathbf{w}}_{(N-1) \times 1}(f) - \mathbf{b}_{L \times 1}(f)\|_p \leq \tau
\end{aligned} \tag{3.20}$$

The convex relation is then given by

$$\begin{aligned}
& \min && \|\tilde{\mathbf{w}}_{(N-1) \times 1}(f)\|_1 \\
& \tilde{\mathbf{w}}_{(N-1) \times 1}(f) && \\
& \text{subject to} && \\
& \|\mathbf{A}_{L \times (N-1)}(f)\tilde{\mathbf{w}}_{(N-1) \times 1}(f) - \mathbf{b}_{L \times 1}(f)\|_p \leq \tau
\end{aligned} \tag{3.21}$$

We note that existing literature such as [15] gives guarantees on the quality of solution found by optimization problems (3.21) if $\mathbf{A}_{L \times (N-1)}$ satisfies the Restricted Isometry Property (RIP) [16]. However in our problem setup, $\mathbf{A}_{L \times (N-1)}$ does not satisfy RIP for reasonable numbers of available microphones (e.g. $N = 1000$) since the microphones are closely spaced and hence columns of $\mathbf{A}_{L \times (N-1)}$ have high similarity.

3.6 Experimental Results

Below we present two types of results. First we do a comprehensive analysis of the combinatorial optimization problem (4.5) for a tractable size. Second, we study a large scale setup. All reported gain results are in [dB] defined as $10 \log_{10} \|Hw\|_\infty$.

I). Consider the rectangular room similar to that of Figure 3.1 of size (10m \times 8m) but with the target located at (1m, 4m) and 6200 interferences operating at $f = 1\text{kHz}$. The total number of

Table 3.1. Number of separated minima of criterion J for $D = 5$ active microphones out of N and a neighborhood radius R shows the vast majority of combinations are nonoptimal locally.

N \ R	R										$\binom{N}{5}$
	1	2	3	4	5	6	7	8	9	10	
10	13	7	3	2	2	1	1	1	1	1	252
12	13	2	1	1	1	1	1	1	1	1	792
14	29	9	5	1	1	1	1	1	1	1	2002
16	112	27	10	4	4	3	2	2	2	2	4368
18	184	49	15	10	8	7	1	1	1	1	8568
20	288	56	24	13	6	6	2	2	2	2	15504

microphones N runs from 10 to 20 but only $V = 5$ are active microphones. For each combination $\pi = (\pi_1, \pi_2, \pi_3, \pi_4, \pi_5)$ of 5 microphones out of N we solve the reduced optimization problem

$$J(\pi) = \min_{\tilde{w}} \|\tilde{H}\tilde{w}\|_{\infty} \quad , \quad \text{subject to } \langle \tilde{w}, \tilde{H}_0 \rangle = 1 \quad (3.22)$$

where \tilde{H} is the 6200×5 submatrix of H corresponding to the combination π of active microphones. Then we analyze the number of separated (local) minima across the set of $\binom{N}{5}$ combinations. We use the l^1 modulo N for distance. We analyze neighborhoods of radius R , with R from 1 to 10. Specifically a combination ρ is a separated minimum if $J(\rho) \leq J(\pi)$ for all combinations π so that $d(\rho, \pi) \leq R$. Table 3.1 summarizes the number of separated minima for a given R . Next we apply the λ -method where we tune λ to achieve 5 significant nonzero weights. Due to axial symmetry in cases $N=12$ and $N=14$ the 5^{th} and 6^{th} largest weights are equal. In the other cases, 5^{th} largest weight was at least 1000 times larger than the 6^{th} weight. The optimal criterion found by the λ -method is less than 0.75dB higher than the global minimum (see Table 3.2).

II). We perform simulations to show the performance and properties of the λ -method, where we aim to separate sounds from a known target location and unknown interference locations. To account for the unknown locations we consider a uniform grid of 100×80 interferences. We refer to this grid as the *optimization grid*. All experiments assume a rectangular room of dimension

Table 3.2. Comparison λ -method vs. global minimum.

N	10	12	14	16	18	20
λ -method [dB]	-0.86	0.94	0.55	-1.39	-0.95	-1.55
Global optim [dB]	-1.31	0.54	0.54	-1.6	-1.68	-1.62

10m x 8m and $N = 1000$ microphones. We disregard interferences that are closer than 0.5m to the source or the walls of the room and report the worst target to interference gain (WTIG) as performance measure.

First, we use the λ -method to select a subset of size $V = 32$ microphones from the original 1000 that achieves the lowest WTIG for a 1kHz target signal at position $(x, y) = (3, 4)$ for a single path reflection model on a fine 1000×800 *evaluation grid*. This is achieved as discussed in Section 4.4. The resulting gain beam pattern is illustrated in Fig. 3.2. Here, the WTIG is found is 1.5dB. The disregarded regions close to the walls and the target are indicated by white coloring. Note that both inferred microphone locations are beam pattern are symmetric. Also, the beam pattern shows ray-like regions of higher gains “emitting” from pairs of microphones. These rays overlap in a 2D design to archive a high gain at the target and to suppress signals at other locations in the room.

Next, we analyze performance as a function of the number of selected microphones V and the target location. Figure 3.3 illustrates the WTIG of a 1kHz target signal at positions (1,4) and (3,4) for $D = 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1000$ active microphones and $\lambda = 0.1$. This shows that a small subset of e.g., 32 microphones can be sufficient to achieve performance similar to the solution of all 1000 microphones. However, it also shows that the necessary number of microphones, for close to optimum performance, is dependent on the target location. That is, while sub-selecting 32 microphones results in a WTIG increase over the 1000 microphone solution of 0.1dB for position (1,4) on the evaluation grid, it results in an WTIG increase of 4.8dB for position

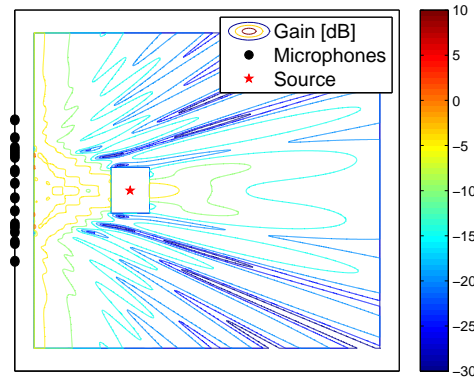


Fig. 3.2. Gain pattern of a 32 element microphone array at 1000 hz found by λ -method. The worst target to interference gain is 1.5 dB. The array is down-selected from 1000 uniform distributed microphones.

(3,4).

Finally, we evaluate the quality of the result from our λ -method by comparing it with the results from the SA method. This is necessary due to the non-convex nature of the optimization problem (4.5), which suggests that our approach could converge to a sub-optimal local minimum. Both, the λ -method and SA method use the optimization grid to find the microphone locations and weights. SA is initialized at the final microphone locations of the λ -method to reduce processing time. The result of this comparison is illustrated in Table 3.3. The small improvement from SA illustrates the quality of the λ -method result. In Fig. 3.4, the gain pattern is no longer symmetric but achieves a better WTIG of 1.5 dB.

Table 3.3 and Fig. 3.3 also show that the microphone selection and weights obtained using the optimization grid perform sub-optimally on the finer evaluation grid. That is, while Table 3.3 reports a WTIG of -12.71 dB for position (1,4), where the evaluation is based on the optimization grid, the same position only achieves a WTIG of -4.79 dB when the evaluation is based on the finer evaluation grid. Note that it is computationally infeasible to use the fine evaluation grid for optimization with the λ -method.

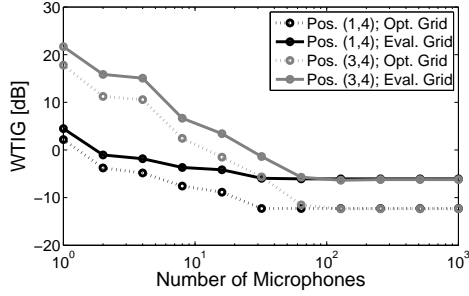


Fig. 3.3. Worst target to interference gain for a number of active microphones and different target locations. 32 microphones are sufficient to achieve performance similar to 1000 microphones. The sufficient number of microphones is dependent on the target location.

Table 3.3. WTIG for the λ -method and SA

	Method	Pos. (1,4)	Pos. (3,2,5)	Pos. (3,4)
Opt. Grid (100 × 80)	λ -method	-12.71	-7.35	-5.73
	SA	-13.00	-7.66	-5.73
Eval. Grid (1000 × 800)	λ -method	-4.79	0.30	1.5
	SA	-5.34	-0.14	0.1

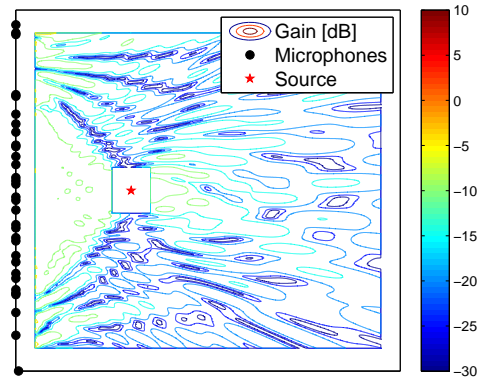


Fig. 3.4. Gain pattern of a 32 element microphone array at 1000 hz found by Simulated Annealing.

The worst target to interference gain is 0.1 dB. The search was initialized by the result of the λ -method

3.7 Conclusions

Our problem is acoustic scene understanding, which subsumes a number of high level tasks such as localization of acoustic sources, tracking of features of the sources, and even source separation. We aim to utilize a very large number of microphones distributed spatially in the volume of interest. This paper discusses optimal beam patterns for very large microphone arrays. It shows that only a sparse active subset of microphones is sufficient at any given time. The equivalent problem with a small set of microphones would be element spacing. The paper formalizes the problem of defining active subsets of microphones capable to zoom into a target location while suppressing a large number of interferences. It shows that near optimal subsets are solutions of a convex approximation of the originally stated non-convex criterion using a LASSO inspired approach. Future work will show how solutions representing multiple active subsets of a very large multidimensional microphone array could be adaptively employed in order to locate and separate sources of interest.

Part II

Processing Broadband Source Using Delay-Scale-Sum Beamformer

Chapter 4: Processing Broadband Source Using Delay-Scale-Sum Beamformer

4.1 Overview

Consider a sensing system using a large number of N microphones placed in multiple dimensions to monitor a broadband acoustic field. Using all the microphones at once is impractical because of the amount of data generated. Instead, we choose a subset of V microphones to be active. Specifically, we wish to find the set of V microphones that minimizes the p -norm of the gain at multiple frequencies while monitoring a target of interest with $p \in \{1, 2, \infty\}$. A direct, combinatorial approach – testing all N choose V subsets of microphones – is impractical because of the problem size. Instead, we use a convex optimization technique that induces sparsity through a l_1 -penalty to determine which subset of microphones to use. We test the robustness of the our solution through simulated annealing and compare its performance against a classical beamformer which maximizes SNR. Since switching from a subset of D microphones to another subset of D microphones at every sample is possible, we construct a space-time-frequency sampling scheme that achieves near optimal performance.

4.2 Introduction

Consider a large scale microphone array having N microphones that monitors a surveillance area. Using all microphones simultaneously may be unreasonable in terms of power consumption and data processing. For example, for $N = 10000$ microphones and a data sampling rate of 100000 samples per second, the bandwidth requirement is 1Gsamples/sec. Instead, we could poll only a subset of D microphones at any one given time. The N choose D number of choices of microphones allows for a myriad of microphone configurations, and the task is then to choose a subset that achieves our objective.

Alternatively, our problem is to place D microphones in a constrained region of space. We convert this non-convex optimization problem into a combinatorial problem by discretizing the possible set of microphone locations. In this context our approach can be seen as an optimal test design.

Assume the surveillance area consists of a set of point-like sources. We seek designs that minimize the p -norm of interference gains across frequencies from a potentially very large number of locations while still maintaining target unit gain at each frequency. Throughout this paper, we assume the following hypotheses:

H1. Microphone locations, that is their $(x, y, z$ -coordinates), are known. Their locations however can be arbitrary.

H2. The number of real interferences, their locations, and their spectral powers are unknown.

H3. The maximum number of microphones active at any time, V , is fixed in advance.

H4. All sources, both target and interferences, are broadband that is each source consist of F frequencies f_0, f_1, \dots, f_{F-1}

When microphones have local computational processing power, we make an additional hypothesis:

H5. Microphones can band-pass signals and they can switch data transmission on a sample-by-sample basis.

Under H5, the system can sample selectively the space-time-frequency domain.

Our system divides the surveillance area into a large number of small, non-overlapping target areas. For each fixed target area, we find a subset of microphones that minimizes the p -norm of broadband gains of a large number of virtual interferences with $p \in \{1, 2, \infty\}$. By doing so, the system is robust to a wide variety of unknown interference configurations. In addition, we note that the system can choose a different set of microphone locations for each of the three different p -norms.

After finding a set of V microphones to monitor each of the individual target areas, the system begins to iteratively scan each target area by polling the predetermined set of V microphones for each of the corresponding target areas. We note that the V microphones need not be unique for each target area but rather the same microphone can be reused across different sets of V microphone configurations as target areas vary. By polling sets of $numactivemics$ microphone for each target area, we can measure the signal strength in each target area and hence discover and monitor the actual targets.

Our goal then is to find for a fixed target area the optimal set of V microphones and their corresponding broadband weights and that minimizes the p -norm of the broadband gain of the interferences while maintain target unity gain at each frequency with $p \in \{1, 2, \infty\}$

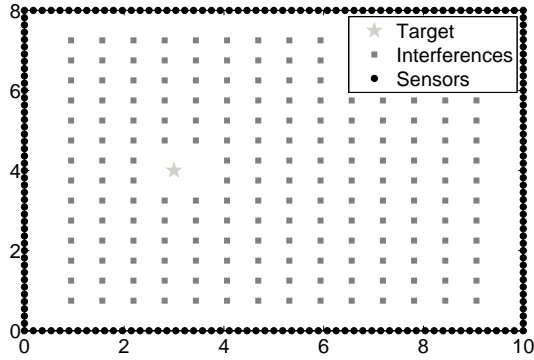


Fig. 4.1. Sample Scenario: $L = 193$ sources (1 target and 192 interferences), $N = 200$ microphones. The 200 microphones are placed on the walls of a 10 m by 8 m room. To monitor the target located at (3m, 4m), we wish to use only V microphones instead of all N so as not to overwhelm our central processing unit with data. If we set $V = 5$, there are $\binom{N}{V} = \binom{100}{5} \approx 7.5 \times 10^7$ possible sets of 5 microphones to use. The L sources are assumed to be broadband, that is each source consists of the F frequencies f_1, f_2, \dots, f_F . We allow the N filter weights to vary for each of the F frequencies and hence our optimization problem returns a set of $F \cdot N$ filter weights.

4.3 Problem formulation

Consider the setup described earlier and sketched in Figure 4.1. We assume we know the microphone locations and then fix a target area. Since the number, location, and power distribution of each of the interferences is unknown, we assume a large number of virtual interferences, say L . To account for possible reflections off of walls, we include virtual interferences outside our surveillance area. Our objective is to minimize the p -norm of the interference broadband gains with $p \in \{1, 2, \infty\}$.

Assume the zero indexed source is the target, and the remaining L sources are interferences. For these “virtual” $L+1$ sources we use the direct-path model, where the transfer function between

source l and microphone n at frequency f_i is given by

$$H_{l,n}(f_i) = \frac{e^{-2\pi i f_i c \|r_l - r_n\|}}{\|r_l - r_n\|}, 0 \leq l \leq L, 1 \leq n \leq N \quad (4.1)$$

where r_n, r_l denote the position vectors of microphone n and source l respectively, and c is the speed of sound, which in our case is the speed of sound, 343 m/sec. The direct path model assumes that the signal amplitude decays in proportion to distance and the frequency phase varies cyclically in proportion to both the propagation speed and frequency.

Let $w_n(f_i)$ denote the processing weight for microphone n at frequency f_i . In other words, we assume that each microphone applies a delay and scaling in the time domain to the signal for each frequency of interest of f_i . This can be accomplished by splitting the signal into frequency bands via bandpass filters and operating on each bandlimited signal individually. A key observation is that if $w_n(f_i) = 0$, then microphone n is inactive at frequency f_i .

For a fixed interference l and frequency f_i , the the system processing *response* to source l , $R_l(f_i)$, of our delay-scale-sum system is given by

$$R_l(f_i) = \sum_{n=0}^{N-1} H_{l,n}(f_i) w_n(f_i). \quad (4.2)$$

We note that for a fixed frequency f_i each of the N microphones receives a “copy” of source l that has been distorted by some transfer function $H_{l,n}$ that is a function of the distance between the microphone n 's and source l 's location. Each microphone n 's copy is processed by some weight w_n and all N processed, distorted copies are summed up to give the system processed source l .

$R_l(f_i)$ is then a complex scalar that is exactly the delay-scale-sum system's effect on source l for frequency f_i . By taking the modulus of $R_l(f_i)$, we get the *gain* $K_l(f_i)$, that is

$$K_l(f_i) = |R_l(f_i)| \quad (4.3)$$

The gain $K_l(f_i)$ is a real non-negative scalar that measures how much our system attenuates or amplifies a given source l at frequency f_i . If $0 \leq K_l(f_i) < 1$, the system attenuates the source l and conversely if $K_l(f_i) > 1$, the system amplifies source l . If $K_l(f_i) = 1$, the system does not distort the amplitude of source l but may possibly introduce a phase shift to source l at frequency f_i .

Assume that we have F distinct frequencies of interest, f_0, f_1, \dots, f_{F-1} . Given F sets of N microphone weights, each interference l has F different gains, one for each frequency. Note that for each of the F frequencies, we use a different set of N microphone weights to calculate the interference gains at that frequency. Our objective is to minimize the p -norm of *sum of gains across F frequencies* for interference sources while still achieving target unit response at each frequency with $p \in \{1, 2, \infty\}$. We call the sum of gains of an interference across F frequencies an interference's *broadband gain*. Hence, the broadgain for source l for frequencies of interest f_0, f_1, \dots, f_{F-1} , $K_l(f_0, f_1, \dots, f_{F-1})$, a real non-negative scalar, is given by

$$K_l(f_0, f_1, \dots, f_{F-1}) = \sum_{i=0}^{F-1} \left| \sum_{n=0}^{N-1} H_{l,n}(f_i) w_n(f_i) \right| \quad (4.4)$$

Assumption H3, however, limits us to V simultaneously active microphones.

In this multi-frequency setup, when H5 is not satisfied, microphone n becomes active if $w_n(f_i) \neq 0$ for any i . Thus, the number of non-zeros in the N -dimensional vector $\{\max_{0 \leq i \leq F-1} |w_n(f_i)|\}$ is an appropriate measure of the number of active microphones. In other words, if a fixed microphone n has any non-zero weight for any of its F filter weights, the microphone is considered active. Let $\|\mathbf{w}\|_0$ denote the pseudo-norm which counts the number of non-zeros in the vector \mathbf{w} . We can now state our optimization problem:

$$\begin{array}{l}
\min_{(w_n(f_i))_{i=0, n=0}^{F-1, N-1}} \quad \left\| \left\{ \sum_{i=0}^{F-1} \left| \sum_{n=0}^{N-1} H_{l,n}(f_i) w_n(f_i) \right| \right\}_{1 \leq l \leq L} \right\|_p \\
\text{subject to} \quad \sum_{n=0}^{N-1} H_{0,n}(f_i) w_n(f_i) = 1 \text{ for } i = 0, 1, \dots, F-1 \\
\left\| \left\{ \max_{0 \leq i \leq F-1} |w_n(f_i)| \right\}_{0 \leq n \leq N-1} \right\|_0 \leq V
\end{array} \tag{4.5}$$

Note that this is a non-convex optimization problem due to the l^0 pseudo-norm constraint $\left\| \left\{ \max_{0 \leq i \leq F-1} |w_n(f_i)| \right\}_{0 \leq n \leq N-1} \right\|_0 \leq V$. When hypothesis H5 is satisfied, we can solve the optimization problem (4.5) independently for each frequency, and then implement an adaptive space-time-frequency sampling.

4.4 Convex Optimization Strategy

Our method is inspired by LASSO regularization [1], a regression technique that minimizes the sum of squares of residual errors subject to the l^1 norm of the coefficients being less than a constant. Similar to the sparse signal and model estimation approach in [2], the l^0 pseudo-norm is replaced by the l^1 norm $\|w\|_1 = \sum_{n=0}^{N-1} |w_n(f)|$ which is then incorporated into the optimization criterion using a Lagrange multiplier λ . The optimization problem (4.5) is then replaced by the following convex optimization problem which we call the λ -method:

$$\begin{array}{l}
\min_{(w_n(f_i))_{i=0, n=0}^{F-1, N-1}} \quad \max_{1 \leq l \leq L} \sum_{i=0}^{F-1} \left| \sum_{n=0}^{N-1} H_{l,n}(f_i) w_n(f_i) \right| + \lambda \sum_{n=0}^{N-1} \max_{0 \leq i \leq F-1} |w_n(f_i)| \\
\text{subject to} \quad \sum_{n=0}^{N-1} H_{0,n}(f_i) w_n(f_i) = 1 \text{ for } i = 0, 1, \dots, F-1
\end{array} \tag{4.6}$$

For very large λ , the penalty term forces many of the microphones to become inactive. Specifically, let $\mathbf{W}_{N \times F}$ be the N by F matrix of microphone weights produced by the optimization where a row represents the F filter weights of an individual microphone. With a large λ penalty, many of the rows of $\mathbf{W}_{N \times F}$ contain only values very close to zero. If the n th row only contains such small values, microphone n is inactive. As λ decreases, the penalty term becomes

less expensive and more microphones become active. At the limit, $\lambda = 0$, all microphones are allowed to be active. We fine tune λ until we get V non-zero components. Specifically, this is accomplished when $g_{n(V+1)}$ is significantly smaller than $g_{n(1)}$ (e.g. by 3 orders of magnitude), where $g_n = \max_{0 \leq i \leq F-1} |w_n(f_i)|$ and $g_{n(i)}$ is the i -th largest element in magnitude of the set $\{g_n\}$, $n = 0, 1, \dots, N-1$. We then solve (4.6) using this λ . The final weights are obtained by solving a second time (4.6) restricted to V microphones and $\lambda = 0$, commonly known as the debiasing step [1].

4.5 Varying Microphone Support Over Frequencies

When we optimize over multiple frequencies (4.6), the support of the chosen microphones remains fixed over the frequencies. However, we can also run (4.6) for each individual frequency of interest f_i , that is we solve the optimization problem

$$\begin{aligned} & \min_{(w_n(f_i))_{n=0}^{N-1}} \max_{1 \leq l \leq L} \left| \sum_{n=0}^{N-1} H_{l,n}(f_i) w_n(f_i) \right| + \lambda_i \sum_{n=0}^{N-1} |w_n(f_i)| \\ & \text{subject to } \sum_{n=0}^{N-1} H_{0,n}(f_i) w_n(f_i) = 1 \end{aligned} \quad (4.7)$$

We have to run (4.7) a total of F times, once for each frequency of interest f_i . A run of (4.7) for frequency f_i means to iteratively choose λ_i until we discover a λ_i that produces V active microphones. We note that every time we test a new candidate λ_i , the optimization problem (4.7) needs to be solved .

Each run of (4.7) for fixed f_i produces a different set of V microphones and hence the support of the microphones can vary over frequencies. If the frequencies are chosen with uniform spacing, we can use non-uniform sampling to reconstruct the signals of interest at each frequency. Specifically assume we have a total of F frequency bands (e.g. $F = 4$ as in the results below). Then each microphone has an additional F -channel filter bank each processing disjoint frequency

bands. The filter bank outputs are downsampled by F and the corresponding sample streams are sent according the transmission policy obtained in the optimization step. The central controller reconstructs the broadband signal by putting together the disjoint frequency bands.

Another method of allowing the microphone supports to vary over frequency is to solve the following optimization problem

$$\begin{aligned}
& \min_{(w_n(f_i))_{i=0, n=0}^{F-1, N-1}} \max_{1 \leq l \leq L} \sum_{i=0}^{F-1} \left| \sum_{n=0}^{N-1} H_{l,n}(f_i) w_n(f_i) \right| + \sum_{i=0}^{F-1} \lambda_i \sum_{n=0}^{N-1} |w_n(f_i)| \\
& \text{subject to } \sum_{n=0}^{N-1} H_{0,n}(f_i) w_n(f_i) = 1 \text{ for } i = 0, 1, \dots, F-1
\end{aligned} \tag{4.8}$$

Every time (4.8) is solved, F different λ_i must be chosen. The question arises then how to iteratively choose the F differnt λ_i so that V active microphones are chosen for each frequency.

We also note that (4.8) differs from (4.7) in that the objective function includes the broadband gain.

4.5.1 Varying λ_1 and λ_2 For Two Frequencies

We are interested in the pair of lambda values that produces *exactly the same number of significant microphones weights for each of the two frequencies* where one lambda is a scalar multiply of the ell-one norm of microphone weights for the first frequency and the other lambda is similarly a scalar multiply of the ell-one norm of microphones weights for the second frequency within the objective function. In other words, we have a fixed number of microphones for each frequency but the microphone positions and weights are allowed to vary for each frequency.

For example, if we had 100 possible microphone locations, we would have then have 200 total microphones weights we wish to optimize, 100 weights for each of the two frequencies. We want to find the pair of lambda values that would produce 32 significant weights in the 100 weights corresponding to frequency 1 and 32 significant weights in the 100 weights corresponding

to frequency 2.

A weight is considered significant if its magnitude is greater than 0.1% of the largest magnitude of the weights for that frequency.

We note that the optimization problem consists of an objective function and a constraint. The objective function is the sum of three terms: the p-norm of the sum of the interference gain across the two frequencies, the first lambda multiplied against the ell-one norm of the microphone weights corresponding to frequency one, and the second lambda multiplied against the ell-one norm of the microphone weights corresponding to frequency two. In addition, we have the constraint that the microphone weights produce unit gain for the target at both frequencies.

We also note by increasing the lambda value corresponding to a single frequency, the number of significant weights for that frequency decreases and similarly by decreasing the lambda value corresponding to a single frequency, the number of significant weights for that frequency increases. What is unknown is the effect of changing the lambda for a given frequency on the number of significant weights for the other frequency.

To gain intuition on the effect of varying one lambda on the other frequency's number of significant weights, we run the following experiment.

We fix our possible microphones locations, the target, the interferences, and the two frequencies.

We then vary two different lambdas values within our objective function. For example, if we wish to test the lambda values 0,1, and 100 we would have 3^2 lambda pairs to test. For each pair of lambdas we test, we collect four values: the number of significant weights for each of the two frequencies and the ell-one norm of the microphone weights for each of the two frequencies.

We specify the number of significant microphone weights for each frequency we are interested in. In our case, 5,8, and 32 microphones weights out of 100.

For example, for 32 significant microphones, we plot the contour lines for 32 microphones for each of the two frequencies on the same plot with the the x-axis and y-axis representing the variation of lambda values tested We have then plotted two different contour lines, one for frequency. We find the intersection of the two contour lines and for that intersection point, a specific pair of lambdas, we look up the corresponding ell-one norm of weights of each of these two frequencies. This represents two different ell-one weights. One is the ell-one norm of the weights for frequency one and the other is the ell-one norm of the weights for frequency two.

We note that quantization (round-off error) is involved in finding this intersection point since not all possible lambda values between smallest lambda and largest lambda, an uncountably infinite number of points, are tested. The intersection point is chosen to be the pair of lambda values that are closest in distance to the intersection of the two contour lines where the contour lines are they themselves interpolated curves.

We then plot the contour lines for two different ell-one weights for both frequencies on the same plot with the the x-axis and y-axis representing the variation of lambda values tested. This represents 4 different contour lines.

4.6 Simulated Annealing

We test the robustness of the solution found by convex optimization through simulated annealing (SA). Simulated annealing is a simple randomized technique for iterative improvement introduced in [3]. SA will probabilistically accept worse transitions in order to avoid local minima. In our case, SA minimizes the objective function given by the largest gain for all interference positions for a fixed-size subset of microphones over locations of the microphones in the subset. More precisely, given a fixed number of D microphones, we run the same convex optimization problem as the λ -method (e.g., find the filter weights that minimize the maximum valid interfer-

ence over a coarse grid while maintaining a target gain of unity) but with λ set to 0. The objective function value f is then the largest gain of the valid interferences. If the optimization program fails for any reason, the objective function value is set to infinity. The *initial temperature* is set to 100, and the *annealing schedule* decreases the temperature by 5% each iteration. We limit the length of the Markov walk to a *total length* of 1200 iterations. The initial search point is the point produced by the λ found through an iterative binary search that produces $D = 32$ microphones. We assume that the target location, room size, and the frequency of interest are fixed.

4.7 Relation to Prior Work

In this section, we review prior work in relation to our work.

The author of [17] derives a probabilistic expression for the peak side lobe level for a linear array with randomly placed elements. The elements' weights are uniform in magnitude but their phases are chosen to maximize the main lobe in the direction normal to the array. In our case, we allow weights to vary in both magnitude and phase.

In [18], the authors address the problem of finding the sparsest linear array given a desired beam pattern and error tolerance. The elements' phases are fixed and their magnitudes' are allowed to vary. They furthermore assume symmetry in their linear array so that their optimization variables are real. Analogous to our setup, they formulate an l_0 quasi norm objective to count the number of active array elements by counting the number of non-zero weight values. The number of possible of active array elements available, N in our case, is large but finite to make the problem computationally tractable. They convert the objective to a l_p norm objective with $0 < p < 1$ and solve the optimization problem using a simplex search.

In [19], the author assumes a symmetric linear array setup and formulates an optimization problem to find the real weights that minimize peak side lobe level for fixed sensor locations. He

then proposes another optimization problem that finds both sensor weights and sparse locations simultaneously. To minimize the number of sensors, he formulates a l_0 quasi norm objective to count the number of active array elements by counting the number of non-zero weight. The optimization problem also includes a user-specified constraint on the maximum allowable peak side lobe value. He solves this combinatorially difficult problem for small values using a branch and bound method. In our case, we work with 2-D arrays and optimize over complex weights. In addition, we fix the number of active microphones rather than trying to minimize the number of active microphones and seek to minimize the maximum interference gain in the case $p = \infty$. We relax the l_0 quasi norm to an l_1 norm to discover microphone locations.

In [20], the authors propose two greedy deletion algorithms and a l_1 minimization algorithm to find a sparse linear array containing the desired number of elements. First is the “minimum increase” algorithm, where for a given array, an element is removed and the performance of the remaining elements is measured. This procedure is repeated for each element of the array. The element which affects the performance the least is then deleted. This procedure is then repeated iteratively until the desired number of array elements remain. Second, is the “smallest coefficient” algorithm where the array element with the smallest weight is deleted. This is also repeated iteratively until the desired number of array elements remain. The third algorithm relaxes an l_0 pseudo-norm constraint to a l_1 norm to get a sparse solution. The algorithm then chooses the elements with the V largest weight magnitudes where V is the desired number of active elements. Finally, the algorithm finds new weights for the chosen V algorithms, a debiasing step. Our algorithm is similar in spirit to the third algorithm but we instead we rerun the optimization problem, tuning a sparsity parameter λ using a bisection method, until the l_1 norm minimization problem produces the desired number of V microphones.

In [21], the authors uses the theory of difference sets to place elements. In addition, he

proposes the idea of “spatial hopping” wherein one part of the array form one desired beampattern and the other part of the array forms another desired beampattern. This idea is analogous to the idea of using different subset of microphones to focus in targets at different locations.

The authors of [22] solve the problem of finding the maximally sparse linear array that matches a given reference beampattern within a specified tolerance by converting the problem to finding the maximally sparse linear array that matches a given reference beampattern with the highest a-posteriori probability . They solve the converted problem via Bayesian Compressive sampling. Their algorithm is dependent on an initial estimate of the error variance, σ_0^2 .

4.8 Experimental Results

We run experiments by optimizing over a simpler model and then evaluating over a more sophisticated model. Our optimization model is as follows: The room size is 10 m by 8 m. The target of interest is located at (3 m,4 m). There are 1000 possible microphone locations along the perimeter of the rectangular room. We optimize over four frequencies of 250, 500, 750,and 1000 Hz. There are 6200 virtual interferences, and a direct path model is used to calculate the transfer functions. We do not place these interferences within 0.5 m of the perimeter of the room or the target. The evaluation model differs from the optimization model in two ways: There is a denser set of 620000 interferences, and we include reflections for each of them. We run five types of experiments to compare the performance of the λ -method. First, we run the optimization problem (4.6), simultaneously optimizing over the four frequencies. This setup fixes the support of the microphone setup across all 4 frequencies. Second, we run the optimization problem (4.6) again four times, once at each individual frequency. The support of the chosen microphones are then allowed to vary over frequencies. Third, we randomly perturb the set of microphones found by the multi-frequency optimization of experiment (1) using simulated annealing (SA) to see by how much we can improve the solution. Fourth, we again use SA to perturb the set of microphones

found by the single-frequency optimization of experiment (2). Fifth, we test the performance of the beamformer that maximizes the signal to noise ratio (SNR). This beamformer can be shown to be the set of microphones that lie closest to the target.

Table (1) shows the worst interference gain in dB for the five setups using the evaluation model. The results show our multi-frequency λ -method (column 3) outperforming the beamformer that maximizes SNR (column 2) for every frequency. The single-frequency λ -method performs better than multi-frequency λ -method since microphone locations are allowed to vary across frequencies. Simulated annealing sometimes but not always finds better performing solutions when measured with the evaluation model. By algorithm construction, simulated annealing finds a solution at least as good as the initial point when measured on the optimization model. Figs. 4.2 and 4.3 show the beam patterns for both the λ -method and the maximum-SNR beamformer at 1000 Hz along with the placement of the microphones. Results are in dB, with unit target gain (0 dB). Fig. 4.4 compares the expected value and variance of the maximum gain among a varying number of randomly placed interferences in the surveillance area. Finally, Fig. 4.5 shows the sharp drop in filter weights produced by the multi-frequency λ -method.

f[Hz]	BF	MF λ	SA MF λ	SF λ	SA SF λ
250	15.6	9.2	3.9	-1.19	-3.1
500	14.5	7.6	4.3	3.3	2.1
750	12.5	3.4	4.1	-0.9	-0.2
1000	10.4	2.4	5.8	1.5	0.1

Table 4.1. Worst interference gains[dB] for: Beamformer, MultiFreq λ -method, SA MultiFreq λ -method, SingleFreq λ -method, SA SingleFreq λ -method.

4.9 Conclusions

We aim to utilize a very large number of available microphones by using customized subsets of microphones to monitor specific areas of interest. This selective sampling of microphones then produces reasonable amounts of data to be processed. An equivalent problem to our microphone subset selection is microphone spacing. Our optimization criterion finds microphones that suppress a large number of interferences across multiple frequencies while monitoring a target location. We allow the subset of microphones we choose to have different weights for different frequencies of interest. We show that our multi-frequency LASSO-inspired convex optimization technique can find subsets of microphones that give reasonable performance on evaluation models that contain large numbers of virtual interferences and reflections even though the optimization criterion assumes many fewer virtual interferences and no reflections. If frequencies of interest are uniformly spaced, we can achieve even better performance by allowing the active microphone subset to change over frequencies and then reusing space-time-frequency sampling to recover our signal of interest.

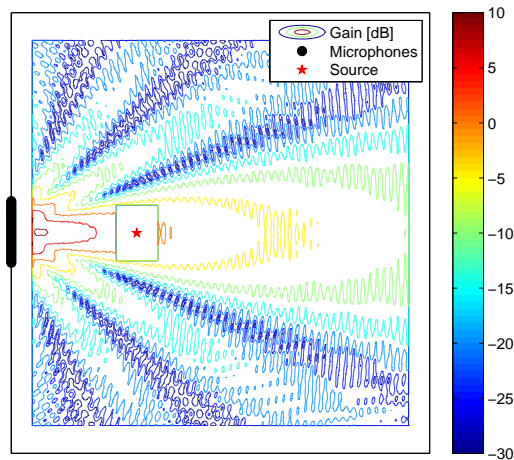


Fig. 4.2. Gain map for classical beamformer that maximizes SNR at 1000 Hz

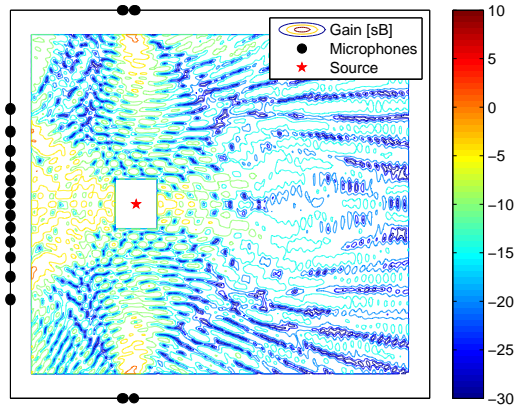


Fig. 4.3. Gain map for beamformer found by multi-frequency λ -method at 1000 Hz

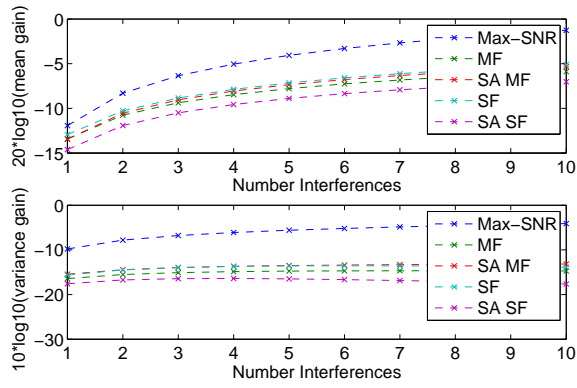


Fig. 4.4. Comparison of mean and variance of maximum gain for varying number of randomly placed interferences at 1000 Hz

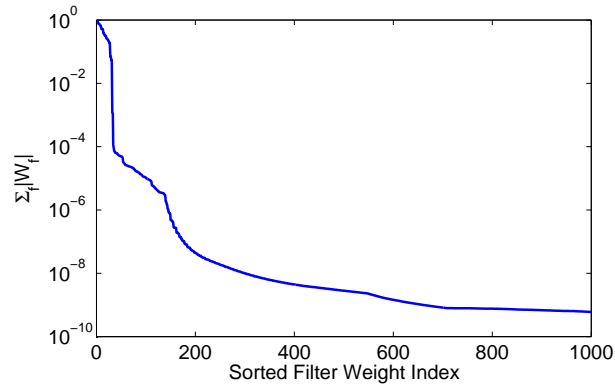


Fig. 4.5. Ordered weights found by multi-frequency λ -method

Part III

Processing Broadband Source Using Multi-rate Filterbanks

Chapter 5: Multirate Filterbank Review

We note that this chapter's derivations are known in literature. In particular, we refer the reader to [23].

5.1 Filtering Followed by Downsampling

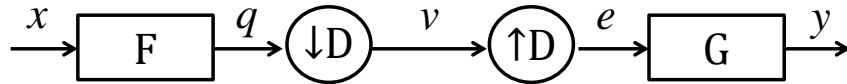


Fig. 5.1. Single channel of multi-rate filterbank

In the first half of Fig. 5.1, the input signal x is first filtered by filter F and then downsampled by integer D . The output after downsampling is denoted by v . The output at time t' after filtering but before downsampling is given by the convolution $q[t'] = \sum_t f[t' - t]x[t]$. If the input to the down sampler is q , then the output of the downsampler is $v[m] = q[mD]$. Combining these two relationships yields the input/output relationship

$$v[m] = q[mD] = \sum_k f[mD - k]x[k] \quad (5.1)$$

Reference [23] in Section 4.1 shows that the z -transform of (5.1) is given by

$$V(z^D) = \frac{1}{D} \sum_{d=0}^{D-1} F(zW_D^d)X(zW_D^d) \quad (5.2)$$

5.2 Upsampling Followed by Filtering

In the second half of Fig. 5.1, the signal v is upsampled by integer D and then filtered by filter G . We denote the signal after upsampling but before filtering as $e[\tau]$ and the output after filtering as $y[t]$. When upsampling by integer D , we have

$$e[\tau] = \begin{cases} v[\frac{\tau}{D}] & \text{if } \tau \bmod D = 0; \\ 0 & \text{otherwise.} \end{cases} \quad (5.3)$$

The output y is a convolution of the filter G and e so by the previous relationship we can derive the input/output relationship as follows

$$\begin{aligned} y[t] &= (g * e)[t] \\ &= \sum_{\tau} g[t - \tau]e[\tau] \\ &= \sum_k g[t - kD]e[kD] \\ &= \sum_k g[t - kD]v[k] \end{aligned} \quad (5.4)$$

In the third line, we let $\tau = kD$ with integer k since $e[\tau]$ is only non-zero when τ is an integer multiple of D . [23] in §4.1 shows that the z -transform of (5.4) is given by

$$Y(z) = G(z)V(z^D) \quad (5.5)$$

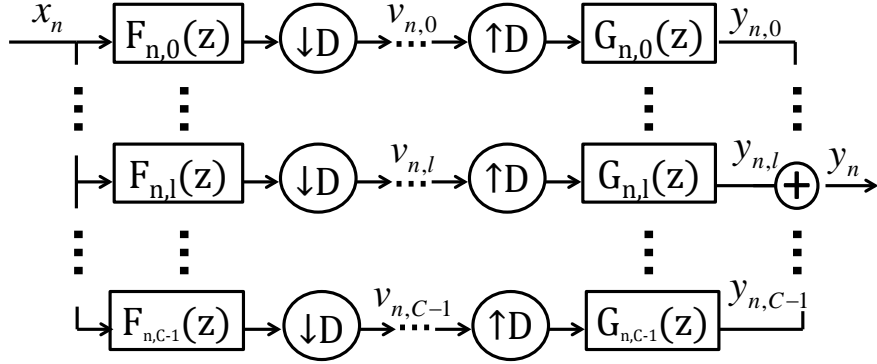


Fig. 5.2. Filterbank for microphone n . Notice that D , the downsampling and upsampling factor, is not necessarily equal to C , the number of channels (subbands).

5.3 Filterbank Processing

The output of the n -th's microphone filterbank, y_n , is the sum of each of the C channels processing result. Hence, we have

$$y_n[t] = \sum_{l=0}^{C-1} y_{n,l}[t] \quad (5.6)$$

where $y_{n,l}$ is the processed result of the l -th channel of the n -th microphone's filterbank.

The z -transform of (5.6) is simply

$$Y_n(z) = \sum_{l=0}^{C-1} Y_{n,l}(z) \quad (5.7)$$

Using (5.4), we can expand (5.6) as follows

$$y_n[t] = \sum_{l=0}^{C-1} \sum_k g_{n,l}[t - kD] v_{n,l}[k] \quad (5.8)$$

where $g_{n,l}$ is the synthesis filter of the l -th channel of the n -th microphone and $v_{n,l}$ is the

channel signal after decimation of the l -th channel of the n -th microphone.

From (5.7), (5.2), and (5.5), we can express the output of the n -th microphone as follows

$$Y_n(z) = \frac{1}{D} \sum_{l=0}^{C-1} G_{n,l}(z) \sum_{d=0}^{D-1} F_{n,l}(zW_D^d) X(zW_D^d) \quad (5.9)$$

(5.9) can also be equivalently be expressed as a scaled product of a row vector, matrix, and column vector, that is

$$Y_n(z) = \frac{1}{D} \mathbf{G}_{n,0,D,1 \times C}(z) \mathbf{F}_{n,C \times D}(z) \mathbf{X}_{n,1 \times D}^T(z) \quad (5.10)$$

where

$$\mathbf{G}_{n,d,D,1 \times C}(z) = [G_{n,0}(zW_D^d), G_{n,1}(zW_D^d), \dots, G_{n,C-1}(zW_D^d)] \quad (5.11)$$

is a row vector, size 1 by C , containing all C synthesis filters for the n -th filterbank,

$$\mathbf{F}_{n,C \times D}(z) = \begin{bmatrix} F_{n,0}(z) & F_{n,0}(zW_D) & \dots & F_{n,0}(zW_D^{D-1}) \\ F_{n,1}(z) & F_{n,1}(zW_D) & \dots & F_{n,1}(zW_D^{D-1}) \\ \vdots & \vdots & \ddots & \vdots \\ F_{n,C-1}(z) & F_{n,C-1}(zW_D) & \dots & F_{n,C-1}(zW_D^{D-1}) \end{bmatrix} \quad (5.12)$$

is a C by D *analysis modulation matrix* ([24], §4.1) for the n -th filterbank, and

$$\mathbf{X}_{n,1 \times D}^T(z) = [X_n(z), X_n(zW_D), \dots, X_n(zW_D^{D-1})]^T \quad (5.13)$$

is a column vector, size D by 1, containing aliased versions of $X_n(z)$, the z -transform of the n -th filterbank. We also define the *synthesis modulation matrix*, size D by C , for the n -th

filterbank as follows

$$\mathbf{G}_{n,D \times C}(z) = \begin{bmatrix} G_{n,0}(z) & G_{n,1}(z) & \dots & G_{n,C-1}(z) \\ G_{n,0}(zW_D) & G_{n,1}(zW_D) & \dots & G_{n,C-1}(zW_D) \\ \vdots & \vdots & \ddots & \vdots \\ G_{n,0}(zW_D^{D-1}) & G_{n,1}(zW_D^{D-1}) & \dots & G_{n,C-1}(zW_D^{D-1}) \end{bmatrix} \quad (5.14)$$

If we expand out the matrix vector multiply $\mathbf{F}_{n,C \times D}(z)\mathbf{X}_{n,1 \times D}^T(z)$ as follows

$$\mathbf{F}_{n,C \times D}(z) \mathbf{X}_{n,1 \times D}^T(z) = X_n(z) \begin{bmatrix} F_{n,0}(z) \\ F_{n,1}(z) \\ \vdots \\ F_{n,C-1}(z) \end{bmatrix} + X_n(zW_D) \begin{bmatrix} F_{n,0}(zW_D) \\ F_{n,1}(zW_D) \\ \vdots \\ F_{n,C-1}(zW_D) \end{bmatrix} + \dots + X_n(zW_D^{d-1}) \begin{bmatrix} F_{n,0}(zW_D^{d-1}) \\ F_{n,1}(zW_D^{d-1}) \\ \vdots \\ F_{n,C-1}(zW_D^{d-1}) \end{bmatrix} \quad (5.15)$$

and observe that $W_D^d = e^{-j2\pi \frac{d}{D}}$ is D periodic in d , we notice the following useful equality from the commutative property of addition

$$\mathbf{F}_{n,C \times D}(zW_D^d)\mathbf{X}_{n,1 \times D}^T(zW_D^d) = \mathbf{F}_{n,C \times D}(z)\mathbf{X}_{n,1 \times D}^T(z) \quad (5.16)$$

for $d = 0, 1, \dots, D - 1$. Using (5.16), the input/output scalar equality of (5.10) can be written in vector form as follows:

$$\boxed{\mathbf{Y}_{1 \times D}^T = \frac{1}{D} \mathbf{G}_{n,D \times C}(z) \mathbf{F}_{n,C \times D}(z) \mathbf{X}_{n,1 \times D}^T(z)} \quad (5.17)$$

where

$$\mathbf{Y}_{n,1 \times D}^T(z) = [Y_n(z), Y_n(zW_D), \dots, Y_n(zW_D^{D-1})]^T \quad (5.18)$$

is a column vector, size D by 1, containing aliased versions of $Y_n(z)$

Chapter 6: Processing Random Source Using Plurality of Multirate Filterbanks

In this chapter, Section 6.1 builds on top of the results of Chapter 5. Section 6.2 presents the second order description of cyclostationary processes. Sections 6.3 and 6.4 describes how cyclostationary processes propagate through a plurality of multirate filterbanks, which is one of our main contributions. We note that the single multirate filterbank description for cyclostationary processes has been derived in [25].

6.1 Problem Setup

Assume that we have N microphones and that the output of each of the N microphones, after pre-filtering, is processed by an individual filterbank. Each filterbank is implemented as a multi-rate, finite-impulse response (FIR) filterbank and each filterbank has the same setup as shown in Fig. 5.2 with D , the downsampling and upsampling rate, and C , the number of channels fixed for all N filterbanks. The analysis and synthesis filters for each subchannel of each filterbank, however, will be allowed to vary. Each of the N filterbanks receives a signal propagating from source x_r . See Fig. 6.1 for the overall system processing setup for a single source x_r .

The n -th filterbank receives a sampled input originating from acoustic source x_r . Each microphone samples at the same uniform rate, and this rate is sufficient to recover all $I + 1$ sources,

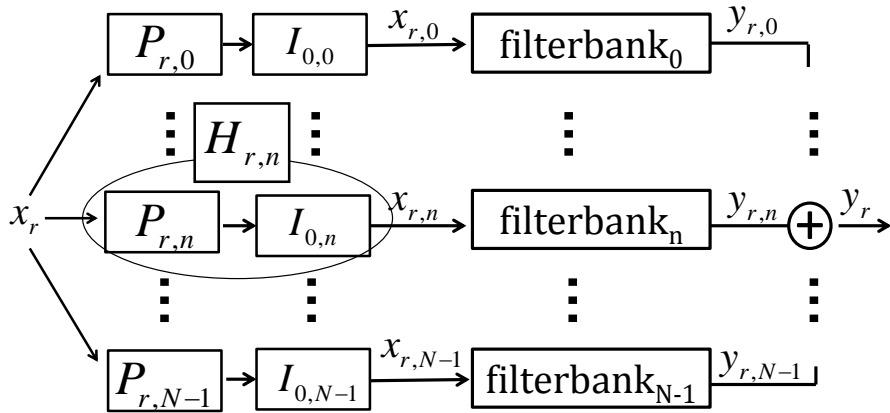


Fig. 6.1. A single acoustic source x_r propagates to N microphones, and the output of each the N microphones is processed by an individual multi-rate, finite-impulse response (FIR) filterbank. The subscript r of x_r indexes the $I + 1$ sources which consist of I interference sources, whose overall gain we wish to minimize, and 1 target source, whose gain we wish to be exactly 1 so that our system perfectly reconstructs the target. We index the target source by setting $r = 0$, and hence the target source is denoted by x_0 . The propagation from source x_r to filterbank n is modeled with transfer function $P_{r,n}$. The prefilter $I_{0,n}$ inverts the propagation from the target source, x_0 , to filterbank n . The cascade of the propagation filter $P_{r,n}$ and target inversion filter $I_{0,n}$ is denoted as $H_{r,n}$. The output of the N filterbanks are summed to give a processed signal y_r

each of which is assumed to be bandlimited. The sampled input is given by $x_{r,n}[k] = x_{r,n}(kT_s)$ where k is an integer time index and T_s is the sampling period. The input signal for filterbank n , $x_{r,n}$, is then fed into C subchannels. The l -th subchannel for the n -th filterbank consists of a FIR analysis filter $F_{n,l}(z)$, a down-sampler of integer D , an up-sampler (zero interpolator) of integer D , and FIR synthesis filter $G_{n,l}(z)$. The outputs of the C subchannels are combined to give an

output signal $y_{r,n}$ or in the z -domain,

$$Y_{r,n}(z) = \sum_{d=0}^{D-1} \left(\frac{1}{D} \sum_{l=0}^{C-1} G_{n,l}(z) F_{n,l}(zW_D^d) \right) X_{r,n}(zW_D^d) \quad (6.1)$$

where the above equation is simply a rearrangement of terms of (5.9). $G_{n,l}$ is the transfer function of the synthesis filter of filterbank n 's subchannel l and $F_{n,l}$ is the transfer function of the analysis filter of filterbank n 's subchannel l . In short, $y_{r,n}$ is the processed output of filterbank n given an input signal propagating from source x_r .

Assume we have $I + 1$ acoustic point sources in the environment. We denote the r -th source as x_r . We set the 0-th source, x_0 , to be the target of interest and denote all the remaining I sources as interferences. We wish to minimize the overall gain of these I interferences.

Let $X_{r,n}(z)$ be the signal received by the n -th filterbank from source x_r , and let $P_{r,n}$ be the transfer function that models the propagation from source x_r to the n -th filterbank. Each filterbank is pre-filtered with a target inversion filter, that is a filter $I_{0,n}$ that inverts the propagation effect on the target's phase from the target x_0 to filterbank n . We denote the cascade of the propagation and target inversion filter from source x_r to microphone n as $H_{r,n}$, that is

$$H_{r,n}(z) = P_{r,n}(z)I_{0,n}(z) \quad (6.2)$$

We then have

$$X_{r,n}(z) = H_{r,n}(z)X_r(z) \quad (6.3)$$

Note that the target inversion pre-filter does not vary with source x_r but only varies with filterbank n .

Remark Ideally, source x_0 , the target source, enters to each of the N filterbanks with only a amplitude scaling and with its *original* phase. If we assume that the propagation $P_{0,n}$ from target

source x_0 to microphone n can be inverted perfectly by prefilter $I_{0,n}$, then we have

$$H_{0,n}(z) = P_{0,n}(z)I_{0,n}(z) = \alpha_{0,n}z^{-\Delta_{H_n}}, \quad (6.4)$$

where $\alpha_{0,n}$ is a real scalar representing the amplitude change from propagation and Δ_{H_n} represents a processing delay.

The processed output of each of the N filterbanks, $Y_{r,n}(z)$, are summed to give the final output signal $Y_r(z)$, the system output from system input $X_r(z)$, that is

$$Y_r(z) = \sum_{n=0}^{N-1} Y_{r,n}(z) \quad (6.5)$$

By substituting the right hand side of (6.1) for $Y_{r,n}(z)$ in (6.5) and then substituting the right hand side of (6.3) evaluated at zW_D^d for $X_{r,n}(zW_D^d)$ in the resulting equation, we have, after rearranging terms,

$$Y_r(z) = \sum_{d=0}^{D-1} \left(\frac{1}{D} \sum_{n=0}^{N-1} H_{r,n}(zW_D^d) \sum_{l=0}^{C-1} G_{n,l}(z)F_{n,l}(zW_D^d) \right) X_r(zW_D^d) \quad (6.6)$$

We want our system to perfectly reconstruct the target source, x_0 , after some processing delay Δ , that is we want, in the z -domain,

$$Y_0(z) = z^{-\Delta} X_0(z) \quad (6.7)$$

By substituting the right hand side of (6.6) with $r = 0$, the target index, into the left hand side of (6.7), we get

$$\sum_{d=0}^{D-1} \left(\frac{1}{D} \sum_{n=0}^{N-1} H_{0,n}(zW_D^d) \sum_{l=0}^{C-1} G_{n,l}(z)F_{n,l}(zW_D^d) \right) X_0(zW_D^d) = z^{-\Delta} X_0(z) \quad (6.8)$$

For the equality of (6.8) to hold, the following D conditions need to hold:

$$\sum_{n=0}^{N-1} H_{0,n}(zW_D^d) \sum_{l=0}^{C-1} G_{n,l}(z) F_{n,l}(zW_D^d) = \begin{cases} Dz^{-\Delta} & \text{if } d = 0; \\ 0 & \text{if } 1 \leq d \leq D-1 \end{cases} \quad (6.9)$$

where Δ is some positive integer, denoting the overall delay of the system.

In matrix-vector form, the D target perfect reconstruction of conditions of (6.9) can be written as

$$\underbrace{\begin{bmatrix} \dot{\mathbf{F}}_{0,1 \times C}(z) & \dot{\mathbf{F}}_{1,1 \times C}(z) & \cdots & \dot{\mathbf{F}}_{N-1,1 \times C}(z) \\ \dot{\mathbf{F}}_{0,1 \times C}(zW_D) & \dot{\mathbf{F}}_{1,1 \times C}(zW_D) & \cdots & \dot{\mathbf{F}}_{N-1,1 \times C}(zW_D) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{\mathbf{F}}_{0,1 \times C}(zW_D^{D-1}) & \dot{\mathbf{F}}_{1,1 \times C}(zW_D^{D-1}) & \cdots & \dot{\mathbf{F}}_{N-1,1 \times C}(zW_D^{D-1}) \end{bmatrix}}_{\dot{\mathbf{F}}_{D \times N \cdot C}} \underbrace{\begin{bmatrix} \mathbf{G}_{0,1 \times C}^T(z) \\ \mathbf{G}_{1,1 \times C}^T(z) \\ \vdots \\ \mathbf{G}_{N-1,1 \times C}^T(z) \end{bmatrix}}_{\mathbf{G}_{1 \times N \cdot C}^T} = \begin{bmatrix} Dz^{-\Delta} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (6.10)$$

where the matrix $\dot{\mathbf{F}}_{D \times N \cdot C}$, size D by $N \cdot C$, consists of $D \cdot N$ row vectors, size 1 by C , defined as

$$\dot{\mathbf{F}}_{n,1 \times C}(z) = H_{0,n}(z) \cdot [F_{n,0}(z), F_{n,1}(z), \cdots, F_{n,C-1}(z)] \quad (6.11)$$

and the column vector $\mathbf{G}_{1 \times N \cdot C}^T$, size $N \cdot C$ by 1, consists of N column vectors, size C by 1, defined as

$$\mathbf{G}_{n,1 \times C}^T(z) = [G_{n,0}(z), G_{n,1}(z), \cdots, G_{n,C-1}(z)]^T \quad (6.12)$$

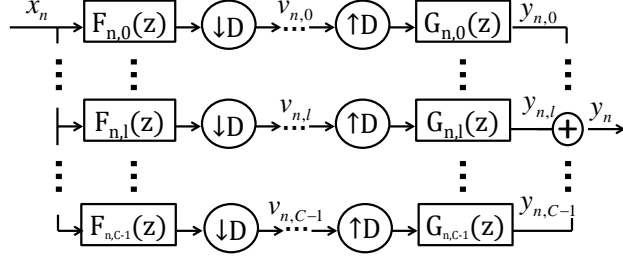


Fig. 6.2. FIR multi-rate filter bank for n -th microphone

6.2 Cyclostationary Processes

The output of each microphone is processed by a corresponding multirate filterbank, Fig. 6.2, and hence it is useful to introduce cyclostationary processes when analyzing such filterbank's statistical properties.

Definition 6.2.1. The *correlation function* of two zero mean discrete random processes x_{n_1}, x_{n_2} , evaluated at time t_1, t_2 , is defined as

$$R_{x_{n_1}, x_{n_2}}[t_1, t_2] = E[x_{n_1}[t_1] \overline{x_{n_2}[t_2]}] \quad (6.13)$$

Definition 6.2.2. Two zero mean discrete random processes x_{n_1}, x_{n_2} are said to be jointly *wide-sense stationary (WSS)* if their correlation function depends only on the difference of their time evaluation, $t_1 - t_2$, that is

$$R_{x_{n_1}, x_{n_2}}[t_1, t_2] = R_{x_{n_1}, x_{n_2}}[t_1 - t_2, 0] \text{ for all } t_1, t_2 \quad (6.14)$$

To shorten notation for the correlation of jointly WSS signals, we will define the following equivalence:

$$R_{x_{n_1}, x_{n_2}}[t_1 - t_2, 0] \equiv R_{x_{n_1}, x_{n_2}}[t_1 - t_2] \quad (6.15)$$

Definition 6.2.3. The *spectral density* of two zero mean discrete random processes x_{n_1}, x_{n_2} which are jointly WSS is the z -transform, evaluated at $e^{j\omega}$, of their correlation function, that is

$$S_{x_{n_1}, x_{n_2}}(e^{j\omega}) = \sum_{\tau} R_{x_{n_1}, x_{n_2}}[\tau] e^{-j\omega\tau} \quad (6.16)$$

Definition 6.2.4. Two zero mean discrete random processes x_{n_1}, x_{n_2} are said to be jointly *wide-sense cyclostationary (WSCS)* with period D if their correlation function is periodic, that is

$$R_{x_{n_1}, x_{n_2}}[t_1 + D, t_2 + D] = R_{x_{n_1}, x_{n_2}}[t_1, t_2] \text{ for all } t_1, t_2 \quad (6.17)$$

Remark Assume x_{n_1}, x_{n_2} are jointly WSS, then we have

$$R_{x_{n_1}, x_{n_2}}[t_1 + P_0, t_2 + P_0] = R_{x_{n_1}, x_{n_2}}[t_1 - t_2] = R_{x_{n_1}, x_{n_2}}[t_1, t_2] \quad (6.18)$$

(6.18) shows that jointly WSS is a special case of jointly WSCS since jointly WSS implies jointly WSCS with period 1.

Remark Assume x_{n_1}, x_{n_2} are jointly WSCS with period D and fix an integer u_0 . Then their correlation function, defined as a function of integer k ,

$$R_{x_{n_1}, x_{n_2}}[u_0 + k, k] \quad (6.19)$$

is D periodic with respect to input parameter k .

Definition 6.2.5. The *cyclic correlation function* for period D and integer d with $0 \leq d \leq D - 1$ is defined as

$$R_{x_{n_1}, x_{n_2}}^{d/D}[u] = \frac{1}{D} \sum_{k=0}^{D-1} R_{x_{n_1}, x_{n_2}}[k+u, k] e^{-j2\pi \frac{d}{D}k} \quad (6.20)$$

The value d/D is known as the *cyclic frequency*.

Remark From equation (6.17), we observe that for fixed u , $R_{x_{n_1}, x_{n_2}}[k+u, k]$ is a periodic sequence in k with period D if x_{n_1}, x_{n_2} are jointly WSCS with period D . Hence, $R_{x_{n_1}, x_{n_2}}^{d/D}[u]$ are the Fourier coefficients of $R_{x_{n_1}, x_{n_2}}[k+u, k]$, that is

$$R_{x_{n_1}, x_{n_2}}[k+u, k] = \sum_{d=0}^{D-1} R_{x_{n_1}, x_{n_2}}^{d/D}[u] W_D^{-kd} \quad (6.21)$$

(6.21) can be verified by substituting in (6.20) into the right hand side and using (2.1).

Remark If x_{n_1}, x_{n_2} are jointly WSS, then the application of (2.1) for any $D \geq 1$ shows that

$$R_{x_{n_1}, x_{n_2}}^{d/D}[u] = \begin{cases} R_{x_{n_1}, x_{n_2}}[u] & \text{if } d = 0 \text{ mod } D; \\ 0 & \text{if } 0 < d \leq D - 1. \end{cases} \quad (6.22)$$

Definition 6.2.6. The *cyclic spectral density* for period D and integer d is the z -transform, evaluated at e^{jw} , of the cyclic correlation function for period D and integer d :

$$S_{x_{n_1}, x_{n_2}}^{d/D}(e^{jw}) = \sum_u R_{x_{n_1}, x_{n_2}}^{d/D}[u] e^{-jwu} \quad (6.23)$$

where $w \in [0, 2\pi)$.

Remark If x_{n_1}, x_{n_2} are jointly WSS, we observe from (6.22) that the cyclic spectral density for $d = 0$ is exactly the spectral density, that is

$$S_{x_{n_1}, x_{n_2}}^0(e^{jw}) = S_{x_{n_1}, x_{n_2}}(e^{jw}) \quad (6.24)$$

6.3 System Input/Output Relationship

The following section derives an input/output relationship for the system given in Fig. 6.3. We note that Fig. 6.3 is identical to that of Fig. 6.1 except the subscript r has been dropped from the source x since in the following derivations we are assuming a fixed source x_r . We assume that the source x is white noise and hence WSS and with uncorrelated time samples and the output y is the sum of N multirate filterbanks each processing a signal propagating from the source x .

In the following, we drop the subscript r from our notation

Remark We will show in the following section that when the input signal x is WSS, then x_{n_1}, x_{n_2} , the input to the n_1 and n_2 filterbank respectively, are jointly WSS. Furthermore, $v_{n_1,p}, v_{n_2,q}$, the p and q subchannel signal of the n_1 and n_2 filterbank respectively are also jointly WSS. However, y_{n_1}, y_{n_2} , the output of the n_1 and n_2 filterbank respectively, are jointly WSCS with period D .

6.3.1 Filterbank Input Cyclic Spectral Density

Lemma 6.3.1. Assume the fixed source x is WSS with spectral density $S_{x,x}(e^{jw})$ and is processed with the scheme depicted in Fig. 6.3. Then x_{n_1} and x_{n_2} are jointly WSS with cyclic spectral density given as follows:

$$S_{x_{n_1}, x_{n_2}}^{d/D}(e^{jw}) = \begin{cases} S_{x,x}(e^{jw})H_{n_1}(e^{jw})\overline{H_{n_2}(e^{jw})} & \text{if } d = 0 \text{ mod } D; \\ 0 & \text{if } d \neq 0 \text{ mod } D. \end{cases} \quad (6.25)$$

where D is any given period, $D \geq 1$.

In the special case when the input signal x is white noise, that is x is WSS and decorrelated, with variance σ_x^2 , then the cyclic spectral density simplifies to the following:

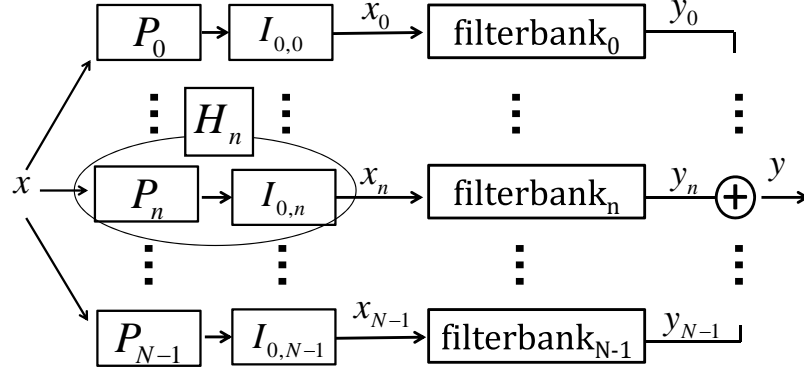


Fig. 6.3. A single acoustic source x propagates to N microphones, and the output of each the N microphones is processed by an individual multi-rate, finite-impulse response (FIR) filterbank. The propagation from source x to filterbank n is modeled with transfer function P_n . The prefilter $I_{0,n}$ inverts the propagation from the target source, x_0 , to filterbank n . The cascade of the propagation filter P_n and target inversion filter $I_{0,n}$ is denoted as H_n . The output of the N filterbanks are summed to give a processed signal y_r

$$S_{x_{n_1}, x_{n_2}}^{d/D}(e^{jw}) = \begin{cases} \sigma_x^2 H_{n_1}(e^{jw}) \overline{H_{n_2}(e^{jw})} & \text{if } d = 0 \bmod D; \\ 0 & \text{if } d \neq 0 \bmod D. \end{cases} \quad (6.26)$$

where as before D is any given period, $D \geq 1$.

Proof. The inputs x_{n_1} and x_{n_2} to the n_1 -th filterbank and n_2 -th filterbank respectively from a single WSS source x are jointly WSS, that is

$$R_{x_{n_1}, x_{n_2}}[t_1, t_2] = R_{x_{n_1}, x_{n_2}}[t_1 - t_2] \quad (6.27)$$

A simple proof of this statement follows:

$$\begin{aligned}
E[x_{n_1}[t_1]\overline{x_{n_2}[t_2]}] &= E\left[\sum_k x[t_1 - k]h_{n_1}[k] \sum_l \overline{x[t_2 - l]h_{n_2}[l]}\right] \\
&= \sum_k \sum_l h_{n_1}[k]\overline{h_{n_2}[l]}E[x[t_1 - k]\overline{x[t_2 - l]}] \\
&= \underbrace{\sum_k \sum_l h_{n_1}[k]\overline{h_{n_2}[l]}R_{x,x}[t_1 - t_2 - k + l]}_{\text{function}[t_1-t_2]}
\end{aligned}$$

where the last line follows because the source x is assumed to be WSS.

To show our desired result, we first note that (6.22) implies for $d \neq 0 \pmod D$, $S_{x_{n_1}, x_{n_2}}^{d/D}(e^{jw}) = 0$ since x_{n_1} and x_{n_2} are jointly WSS. Hence, we now calculate the cyclic spectral density between x_{n_1} and x_{n_2} for period D and $d = 0$.

$$\begin{aligned}
S_{x_{n_1}, x_{n_2}}^0(e^{jw}) &= \sum_u R_{x_{n_1}, x_{n_2}}^0[u - 0]e^{-jwu} \\
&= \sum_u E\left[\underbrace{\left(\sum_k h_{n_1}[u - k]x[k]\right)}_{(x*h_{n_1})[u]} \underbrace{\left(\sum_l \overline{h_{n_2}[-l]x[l]}\right)}_{(x*h_{n_2})[0]}\right]e^{-jwu} \\
&= \sum_k \sum_l \sum_u h_{n_1}[u - k]\overline{h_{n_2}[-l]} \underbrace{E[x[k]\overline{x[l]}}_{R_{x,x}[k-l]} e^{-jw(u-k)} e^{-jw(k-l)} e^{-jwl} \\
&= \sum_l \sum_k \underbrace{\sum_u h_{n_1}[u - k]e^{-jw(u-k)}}_{H_{n_1}(e^{jw})} R_{x,x}[k - l] e^{-jw(k-l)} \overline{h_{n_2}[-l]} e^{-jwl} \\
&= H_{n_1}(e^{jw}) \sum_l \underbrace{\sum_k R_{x,x}[k - l] e^{-jw(k-l)} \overline{h_{n_2}[-l]}}_{S_{x,x}^0(e^{jw})} e^{-jwl} \\
&= H_{n_1}(e^{jw}) \overline{H_{n_2}(e^{jw})} S_{x,x}^0(e^{jw}) \tag{6.28}
\end{aligned}$$

Hence, we have proven (6.25). If the source signal x is white noise with variance σ_x^2 , then we have

$$R_{x,x}[k + u, k] = R_{x,x}[u] = \sigma_x^2 \delta_u \tag{6.29}$$

Using the definition of cyclic spectral density, (6.23), (6.22), and the correlation of white noise, (6.29), we then have

$$S_{x,x}^0(e^{jw}) = \sum_u R_{x,x}^0 e^{-jwu} = \sum_u R_{x,x}[u] e^{-jwu} = \sigma_x^2 \quad (6.30)$$

Substituting (6.30) into (6.28), we then have the desired

$$S_{x_{n_1}, x_{n_2}}^0(e^{jw}) = \sigma_x^2 H_{n_1}(e^{jw}) \overline{H_{n_2}(e^{jw})} \quad (6.31)$$

which proves (6.26). □

6.3.2 Downsampled Subchannel Signal Spectral Density

We now derive the spectral density of subchannel signals $v_{n_1,p}$ and $v_{n_2,q}$ where $v_{n,k}$ is the signal of the k -th subchannel of the n -th filterbanks after being processed by the analysis filter $F_{n,k}$ and then downsampled by D . As before, $n = 0, 1, \dots, N - 1$ indexes the filterbanks, and $k = 0, 1, \dots, C - 1$ indexes the subchannels. See Fig. 5.2.

Lemma 6.3.2. The signals $v_{n_1,p}$ and $v_{n_2,q}$ are jointly WSS with spectral density given by

$$S_{v_{n_1,p}, v_{n_2,q}}(e^{jDw}) = \frac{1}{D} \sum_{m=0}^{D-1} F_{n_1,p}(e^{j(w-w_{D,m})}) \cdot \sum_{n'=0}^{D-1} \overline{F_{n_2,q}(e^{j(w-w_{D,n'})})} S_{x_{n_1}, x_{n_2}}^{(n'-m)/D}(e^{j(w-w_{D,m})}) \quad (6.32)$$

where as in (2.8), $w_{D,d} = \frac{2\pi}{D}d$.

Remark The statement of this lemma holds true even under the weaker assumption that x_{n_1}, x_{n_2} are jointly WSCS with period D . The following proof remains the same under the weaker assumption.

Proof. We note that this derivation parallels the derivation in Appendix I of [25]. First we show that the downsampled subchannel signals $v_{n_1,p}, v_{n_2,q}$ are jointly WSS so that their spectral density exists. Using (5.1), we can derive a relationship between the correlation of the downsampled channels $v_{n_1,p}, v_{n_2,q}$ and the correlation of filter input signals x_{n_1}, x_{n_2} . See Fig. (5.2) for the setup of a single filterbank.

$$\begin{aligned}
R_{v_{n_1,p}, v_{n_2,q}}[m, n] &= E[v_{n_1,p}[m], \overline{v_{n_2,q}}[n]] \\
&= E \left[\sum_k f_{n_1,p}[k] x_{n_1}[Dm - k] \sum_{k'} \overline{f_{n_2,q}[k']} x_{n_2}[Dn - k'] \right] \\
&= \sum_k \sum_{k'} f_{n_1,p}[k] \overline{f_{n_2,q}[k']} R_{x_{n_1}, x_{n_2}}[Dm - k, Dn - k'] \\
&= \underbrace{\sum_k \sum_{k'} f_{n_1,p}[k] \overline{f_{n_2,q}[k']} R_{x_{n_1}, x_{n_2}}[D(m - n) - k, -k']}_{\text{function } [m - n]} \quad (6.33)
\end{aligned}$$

where we subtracted $-Dn$ from both arguments in the second line since x_{n_1}, x_{n_2} are jointly WSS, (6.27), and therefore jointly WSCS with period 1, (6.18). The last line shows that $R_{v_{n_1,p}, v_{n_2,q}}[m, n]$ is a function of only $m - n$ and therefore $v_{n_1,p}, v_{n_2,q}$ are jointly WSS.

We now begin the derivation of (6.32) with the definition of the spectral density of $v_{n_1,p}$ and $v_{n_2,q}$, (6.16), evaluating the spectral density at the argument e^{jDw} :

$$S_{v_{n_1,p}, v_{n_2,q}}(e^{jDw}) = \sum_{\tau} R_{v_{n_1,p}, v_{n_2,q}}[\tau] e^{-jDw\tau} \quad (6.34)$$

We now substitute in (6.33) into (6.34) by letting $\tau = m - n$, yielding

$$S_{v_{n_1,p}, v_{n_2,q}}(e^{jDw}) = \sum_{\tau} \sum_k \sum_{k'} f_{n_1,p}[k] \overline{f_{n_2,q}[k']} R_{x_{n_1}, x_{n_2}}[D\tau - k, -k'] e^{-jDw\tau}$$

We factor out common terms in the summations (e.g. push summation onto only the terms that vary with the summation index), add and subtract k' in the first argument of $R_{x_{n_1}, x_{n_2}}$, and let

$\tau' = D\tau + k' - k$ to get the following:

$$\begin{aligned}
&= \sum_k \sum_{k'} f_{n_1,p}[k] \overline{f_{n_2,q}[k']} \sum_{\tau} R_{x_{n_1},x_{n_2}}[-k' + \underbrace{D\tau + k' - k}_{\tau'}, -k'] e^{-jDw\tau} \\
&= \sum_k \sum_{k'} f_{n_1,p}[k] \overline{f_{n_2,q}[k']} \sum_{\tau': \tau' - k' + k = 0 \pmod{D}} R_{x_{n_1},x_{n_2}}[-k' + \tau', -k'] e^{-jw(\tau' - k' + k)}
\end{aligned}$$

We next insert the following identity that can be verified using the formula for the sum of terms of a geometric series.

$$\frac{1}{D} \sum_{m=0}^{D-1} W_D^{-m(\tau' - k' + k)} = \begin{cases} 1 & \text{if } \tau' - k' + k \text{ is a multiple of } D; \\ 0 & \text{otherwise.} \end{cases}$$

where, as before, $W_D = e^{-j2\pi/D}$. By inserting this identity, the summation over τ' :

$\tau' - k' + k = 0 \pmod{D}$ can be simplified to be a summation over all τ'

Hence, we have

$$= \sum_k \sum_{k'} f_{n_1,p}[k] \overline{f_{n_2,q}[k']} \sum_{\tau'} R_{x_{n_1},x_{n_2}}[-k' + \tau', -k'] \left(\frac{1}{D} \sum_{m=0}^{D-1} W_D^{-m(\tau' - k' + k)} \right) e^{-jw(\tau' - k' + k)}$$

We now use equation (6.21) to expand $R_{x_{n_1},x_{n_2}}[-k' + \tau', -k']$ into its Fourier Series representation, rearrange terms, and use the definition of the Discrete Time Fourier Transform to get the following:

$$\begin{aligned}
&= \frac{1}{D} \sum_{m=0}^{D-1} \sum_{n=0}^{D-1} \sum_k \sum_{k'} f_{n_1,p}[k] \overline{f_{n_2,q}[k']} \sum_{\tau'} R_{x_{n_1},x_{n_2}}^{n/D}[\tau'] W_D^{nk'} \cdot W_D^{-m(\tau' - k' + k)} e^{-jw(\tau' - k' + k)} \\
&= \frac{1}{D} \sum_{m=0}^{D-1} \sum_{n=0}^{D-1} \sum_k \sum_{k'} f_{n_1,p}[k] e^{-jwk} W_D^{-mk} \overline{f_{n_2,q}[k']} e^{jk'k'} W_D^{(m+n)k'} \sum_{\tau'} R_{x_{n_1},x_{n_2}}^{n/D}[\tau'] e^{-jw\tau'} W_D^{-m\tau'} \\
&= \frac{1}{D} \sum_{m=0}^{D-1} \sum_{n=0}^{D-1} F_{n_1,p}(e^{j(w-w_D,m)}) \cdot \overline{F_{n_2,q}(e^{j(w-w_D,m+n)})} S_{x_{n_1},x_{n_2}}^{n/D}(e^{j(w-w_D,m)})
\end{aligned}$$

where as in (2.8), $w_{D,d} = \frac{2\pi}{D}d$. We again factor out common terms in the summation and

let $n' = m + n$ to get

$$= \frac{1}{D} \sum_{m=0}^{D-1} F_{n_1,p}(e^{j(w-w_{D,m})}) \cdot \sum_{n'=m}^{m+D-1} \overline{F_{n_2,q}}(e^{j(w-w_{D,n'})}) S_{x_{n_1},x_{n_2}}^{(n'-m)/D}(e^{j(w-w_{D,m})})$$

For fixed m with $0 \leq m \leq D-1$, the inner sum over n' has $(m+D-1) - (D) + 1 = m$ terms whose n' index is greater than or equal to D . Each of these “overflow” indices can be shifted by $-D$ to make them each less than m because of the following two relationships, each of which can be verified by direct substitution:

$$\begin{aligned} S_{x_{n_1},x_{n_2}}^{(n'-m)/D}(e^{j(w-w_{D,m})}) &= S_{x_{n_1},x_{n_2}}^{(n'-D-m)/D}(e^{j(w-w_{D,m})}) \\ \overline{F_{n_2,l}}(e^{j(w-w_{D,n'})}) &= \overline{F_{n_2,l}}(e^{j(w-w_{D,n'-D})}) \end{aligned}$$

After shifting the “overflow” indices, the summation over n' runs from 0 to $D-1$ instead of from m to $m+D-1$. We then have the desired result:

$$S_{v_{n_1,p},v_{n_2,q}}(e^{jDw}) = \frac{1}{D} \sum_{m=0}^{D-1} F_{n_1,p}(e^{j(w-w_{D,m})}) \cdot \sum_{n'=0}^{D-1} \overline{F_{n_2,q}}(e^{j(w-w_{D,n'})}) S_{x_{n_1},x_{n_2}}^{(n'-m)/D}(e^{j(w-w_{D,m})})$$

□

6.3.3 Filterbank Input and Downsampled Subchannel Signal Matrix Relationship

We first define the *cyclic spectral density matrix of filterbank input signals* x_{n_1} and x_{n_2} of size $D \times D$ as follows

$$(\mathbf{S}_{x_{n_1},x_{n_2},D \times D}(e^{jw}))_{m,n} = S_{x_{n_1},x_{n_2}}^{(n-m)/D}(e^{j(w-w_{D,m})}) \quad (6.35)$$

where $m, n = 0, \dots, D - 1$ and $w_{D,m} = \frac{2\pi}{D}m$, as in (2.8). Note that when x_{n_1}, x_{n_2} are jointly WSS, $\mathbf{S}_{x_{n_1}, x_{n_2}, D \times D}(e^{jw})$ is a diagonal matrix. Then (6.25) implies we can decompose the matrix $\mathbf{S}_{x_{n_1}, x_{n_2}, D \times D}(e^{jw})$ as follows

$$\mathbf{S}_{x_{n_1}, x_{n_2}, D \times D}(e^{jw}) = \mathbf{H}_{n_1, D \times D}(e^{jw}) \mathbf{S}_{x, x, D \times D}(e^{jw}) \mathbf{H}_{n_2, D \times D}^*(e^{jw}) \quad (6.36)$$

where

$$\mathbf{H}_{n_k, D \times D}(e^{jw}) = \text{diag}(H_{n_k}(e^{jw}), H_{n_k}(e^{j(w-w_{D,1})}), \dots, H_{n_k}(e^{j(w-w_{D,D-1})})) \quad (6.37)$$

for $k = 0, 1, \dots, N - 1$ and

$$\mathbf{S}_{x_{n_1}, x_{n_2}, D \times D}(e^{jw}) = \text{diag}(S_{x,x}(e^{jw}), S_{x,x}(e^{j(w-w_{D,1})}), \dots, S_{x,x}(e^{j(w-w_{D,D-1})})) \quad (6.38)$$

If x is white noise with variance σ_x^2 , then (6.36) simplifies to

$$\mathbf{S}_{x_{n_1}, x_{n_2}, D \times D}(e^{jw}) = \sigma_x^2 \mathbf{H}_{n_1, D \times D}(e^{jw}) \mathbf{H}_{n_2, D \times D}^*(e^{jw}) \quad (6.39)$$

We observe that the scalar spectral density of downsampled subchannel signals, (6.32), can then be written as product of a row vector, matrix, and column vector, that is

$$S_{v_{n_1,p}, v_{n_2,q}}(e^{jDw}) = \frac{1}{D} \mathbf{F}_{n_1,p,1 \times D}(e^{jw}) \mathbf{S}_{x_{n_1}, x_{n_2}, D \times D}(e^{jw}) \mathbf{F}_{n_2,q,1 \times D}^*(e^{jw}) \quad (6.40)$$

where

$$\mathbf{F}_{n,r,1 \times D}(e^{jw}) = [F_{n,r}(e^{jw}), F_{n,r}(e^{j(w-w_{D,1})}), \dots, F_{n,r}(e^{j(w-w_{D,D-1})})] \quad (6.41)$$

Notice that in (6.32), the subchannel indexes p and q are both fixed. Each of the two filterbanks n_1, n_2 has C subchannels and therefore both p, q can vary between 0 and $C - 1$, that is $0 \leq p, q \leq C - 1$. To capture all $C \times C$ spectral density relationships between different downsampled subchannels of filterbanks n_1 and n_2 , we define the *spectral density matrix of downsampled subchannel signals*, size $C \times C$, as follows:

$$(\mathbf{S}_{v_{n_1}, v_{n_2}, C \times C}(e^{jDw}))_{pq} = S_{v_{n_1}, p, v_{n_2}, q}(e^{jDw}) \quad (6.42)$$

Then using (6.40), the modulation matrix of the analysis filters, (5.12), and (6.36), we have

$$\begin{aligned} \mathbf{S}_{v_{n_1}, v_{n_2}, C \times C}(e^{jDw}) &= \frac{1}{D} \mathbf{F}_{n_1, C \times D}(e^{jw}) \mathbf{S}_{x_{n_1}, x_{n_2}, D \times D}(e^{jw}) \mathbf{F}_{n_2, C \times D}^*(e^{jw}) \\ &= \frac{1}{D} \mathbf{F}_{n_1, C \times D}(e^{jw}) \mathbf{H}_{n_1, D \times D}(e^{jw}) \mathbf{S}_{x, x, D \times D}(e^{jw}) \mathbf{H}_{n_2, D \times D}^*(e^{jw}) \mathbf{F}_{n_2, C \times D}^*(e^{jw}) \end{aligned} \quad (6.43)$$

6.3.4 Filterbank Output Cyclic Correlation

We now derive the following cyclic correlation between two filterbanks' output signals y_{n_1} and y_{n_2} where y_n is the output of the n -th filterbank in terms of correlation between downsampled subchannel signals and $v_{n,l}$ is the l -th subchannel signal of the n -th filterbank after downsampling.

Lemma 6.3.3. For any integer u and $0 \leq d \leq D - 1$,

$$R_{y_{n_1}, y_{n_2}}^{d/D}[u] = \frac{1}{D} \sum_{p,q=0}^{C-1} \sum_r \sum_\tau g_{n_1,p}[r+u-D\tau] \cdot \bar{g}_{n_2,q}[r] R_{v_{n_1}, p, v_{n_2}, q}[\tau] e^{-j2\pi \frac{d}{D} r} \quad (6.44)$$

Proof. First, we derive an expression for the correlation between two filterbanks' outputs in terms of their downsampled subchannel signals where y_n is the multirate output of the n -th filterbank, $n = 0, 1, \dots, N-1$, and $v_{n,l}$ is the l -th subchannel signal after downsampling of the n -th filterbank, $l = 0, 1, \dots, C - 1$. We use (6.13), (5.8), and (6.33) to get the following:

$$\begin{aligned}
R_{y_{n_1}, y_{n_2}}[n, k] &= E[y_{n_1}[n]\overline{y_{n_2}[k]}] \\
&= E\left[\sum_{p=0}^{C-1}\sum_{m'}g_{n_1,p}[n-Dm']v_{n_1,p}[m']\sum_{q=0}^{C-1}\sum_m\bar{g}_{n_2,q}[k-Dm]\bar{v}_{n_2,q}[m]\right] \\
&= \sum_{p,q=0}^{C-1}\sum_{m'}\sum_m g_{n_1,p}[n-Dm']\cdot\bar{g}_{n_2,q}[k-Dm]R_{v_{n_1,p},v_{n_2,q}}[m'-m] \quad (6.45)
\end{aligned}$$

We can verify that the correlation of processed filterbank outputs are cyclostationary with period D , that is

$$R_{y_{n_1}, y_{n_2}}[n + D, k + D] = R_{y_{n_1}, y_{n_2}}[n, k] \quad (6.46)$$

using two changes of variables ($m'' = m' - 1, m''' = m - 1$) in (6.45). If we let $n = k + u$ and $\tau = m' - m$ in (6.45), we have

$$R_{y_{n_1}, y_{n_2}}[k + u, k] = \sum_{p,q=0}^{C-1}\sum_{\tau}\sum_m g_{n_1,p}[k + u - D(\tau + m)]\cdot\bar{g}_{n_2,q}[k - Dm]R_{v_{n_1,p},v_{n_2,q}}[\tau] \quad (6.47)$$

We substitute (6.47) into the definition of the cyclic correlation function, (6.20), rearrange terms, and then multiply by $e^{j2\pi\frac{d}{D}Dm} = 1$ for integers d and m to get

$$\begin{aligned}
R_{y_{n_1}, y_{n_2}}^{d/D}[u] &= \frac{1}{D}\sum_{k=0}^{D-1}\sum_{p,q=0}^{C-1}\sum_{\tau}\sum_m g_{n_1,p}[k + u - D(\tau + m)]\cdot\bar{g}_{n_2,q}[k - Dm]R_{v_{n_1,p},v_{n_2,q}}[\tau]e^{-j2\pi\frac{d}{D}k} \\
R_{y_{n_1}, y_{n_2}}^{d/D}[u] &= \frac{1}{D}\sum_{p,q=0}^{C-1}\sum_{\tau}R_{v_{n_1,p},v_{n_2,q}}[\tau]\cdot \\
&\quad \left(\sum_{k=0}^{D-1}\sum_m g_{n_1,p}[k - Dm + u - D\tau]\cdot\bar{g}_{n_2,q}[k - Dm]e^{-j2\pi\frac{d}{D}(k-Dm)}\right)
\end{aligned}$$

We observe that $\sum_{k=0}^{D-1}\sum_m f[k - Dm] = \sum_r f[r]$, and hence we let $r = k - Dm$ to combine the summations over k and m into one summation over r to get the desired cyclic cross correlation.

$$R_{y_{n_1}, y_{n_2}}^{d/D}[u] = \frac{1}{D} \sum_{p,q=0}^{C-1} \sum_r \sum_\tau g_{n_1,p}[r+u-D\tau] \cdot \bar{g}_{n_2,q}[r] R_{v_{n_1,p}, v_{n_2,q}}[\tau] e^{-j2\pi \frac{d}{D} r}$$

□

6.3.5 Filterbank Output Cyclic Spectral Density

We now derive the filterbank output cyclic spectral density in terms of the spectral density of downsampled subchannel signals. We begin with the definition of cyclic spectral density (6.23), substitute in (6.44), factor out common terms in the summations, multiply by $e^{jw(r-r+D\tau-D\tau)} = 1$, and use the definition of the Discrete Time Fourier Transform to get the following:

$$\begin{aligned} S_{y_{n_1}, y_{n_2}}^{d/D}(e^{jw}) &= \sum_u R_{y_{n_1}, y_{n_2}}^{d/D}[u] e^{-jwu} \\ &= \frac{1}{D} \sum_{p,q=0}^{C-1} \sum_u \sum_r \sum_\tau g_{n_1,p}[r+u-D\tau] \cdot \bar{g}_{n_2,q}[r] R_{v_{n_1,p}, v_{n_2,q}}[\tau] e^{-j2\pi \frac{d}{D} r} e^{-jwu} \\ &= \frac{1}{D} \sum_{p,q=0}^{C-1} \sum_r \bar{g}_{n_2,q}[r] e^{j(w-2\pi \frac{d}{D})r} \sum_\tau R_{v_{n_1,p}, v_{n_2,q}}[\tau] e^{-jwD\tau} \\ &\quad \cdot \sum_u g_{n_1,p}(r+u-D\tau) e^{-jw(r+u-D\tau)} \\ &= \frac{1}{D} \sum_{p,q=0}^{C-1} \bar{G}_{n_2,q}(e^{j(w-w_{D,d})}) S_{v_{n_1,p}, v_{n_2,q}}(e^{jDw}) G_{n_1,p}(e^{jw}) \end{aligned} \quad (6.48)$$

where, as before in (2.8), $w_{D,d} = 2\pi \frac{d}{D}$.

6.3.6 Downsampled Subchannel Signal and Filterbank Output Matrix Relationship

We observe that (6.48) can be written as a product of a row vector, matrix, and column vector, that is

$$S_{y_{n_1}, y_{n_2}}^{d/D}(e^{jw}) = \frac{1}{D} \mathbf{G}_{n_1,0,D,1 \times C}(e^{jw}) \mathbf{S}_{v_{n_1}, v_{n_2}, C \times C}(e^{jDw}) \mathbf{G}_{n_2,d,D,1 \times C}^*(e^{jw}) \quad (6.49)$$

where

$$\mathbf{G}_{n,d,D,1 \times C}(e^{jw}) = [G_{n,0}(e^{j(w-w_{D,d})}), G_{n,1}(e^{j(w-w_{D,d})}), \dots, G_{n,C-1}(e^{j(w-w_{D,d})})]$$

We define the *cyclic spectral density matrix of output signals* y_{n_1}, y_{n_2} , size $D \times D$, as

$$(\mathbf{S}_{y_{n_1}, y_{n_2}, D \times D}(e^{jw}))_{m,n} = S_{y_{n_1}, y_{n_2}}^{(n-m)/D}(e^{j(w-w_{D,m})}) \quad (6.50)$$

where, as before in (2.8), $w_{D,m} = 2\pi \frac{m}{D}$. Then, using (6.49) and the modulation matrix of synthesis filters, (5.14), we have

$$\mathbf{S}_{y_{n_1}, y_{n_2}, D \times D}(e^{jw}) = \frac{1}{D} \mathbf{G}_{n_1, D \times C}(e^{jw}) \mathbf{S}_{v_{n_1}, v_{n_2}, C \times C}(e^{jDw}) \mathbf{G}_{n_2, D \times C}^*(e^{jw}) \quad (6.51)$$

6.3.7 Matrix Relationship Between Filterbank Inputs and Filterbank Outputs

Combining the previous matrix relationships of (6.36), (6.43), and (6.51) gives the following statistical input/output relationship of two different filterbanks n_1, n_2 .

$$\begin{aligned} \mathbf{S}_{y_{n_1}, y_{n_2}, D \times D}(e^{jw}) &= \frac{1}{D^2} \cdot (\mathbf{G}_{n_1, D \times C}(e^{jw}) \mathbf{F}_{n_1, C \times D}(e^{jw}) \mathbf{H}_{n_1, D \times D}(e^{jw})) \mathbf{S}_{x,x, D \times D}(e^{jw}) \\ &\quad (\mathbf{H}_{n_2, D \times D}^*(e^{jw}) \mathbf{F}_{n_2, C \times D}^*(e^{jw}) \mathbf{G}_{n_2, D \times C}^*(e^{jw})) \end{aligned} \quad (6.52)$$

To shorten notation, we define the following relationship:

$$\mathbf{Q}_{n, D \times D}(e^{jw}) = \mathbf{G}_{n, D \times C}(e^{jw}) \mathbf{F}_{n, C \times D}(e^{jw}) \mathbf{H}_{n, D \times D}(e^{jw}) \quad (6.53)$$

where synthesis modulation matrix $\mathbf{G}_{n, D \times C}(e^{jw})$ is given by (5.14), analysis modulation matrix $\mathbf{F}_{n, C \times D}(e^{jw})$ is given by (5.12), and propagation and target inversion cascade matrix $\mathbf{H}_{n, D \times D}(e^{jw})$, a diagonal matrix, is given by (6.37)

Hence, we can write (6.52) as simply

$$\mathbf{S}_{y_{n_1}, y_{n_2}, D \times D}(e^{jw}) = \frac{1}{D^2} \mathbf{Q}_{n_1, D \times D}(e^{jw}) \mathbf{S}_{x, x, D \times D}(e^{jw}) \mathbf{Q}_{n_2, D \times D}^*(e^{jw}) \quad (6.54)$$

In the case when x is white noise with variance σ_x^2 , then (6.54) simplifies to

$$\mathbf{S}_{y_{n_1}, y_{n_2}, D \times D}(e^{jw}) = \frac{\sigma_x^2}{D^2} \mathbf{Q}_{n_1, D \times D}(e^{jw}) \mathbf{Q}_{n_2, D \times D}^*(e^{jw}) \quad (6.55)$$

6.4 Time Averaged Energy of Output

We now show that the *time averaged energy* of the system output y , the sum of processing from N filterbanks, from a single source x is related to the trace of the cyclic spectral density of filterbank outputs $\mathbf{S}_{y_{n_1}, y_{n_2}, D \times D}$. We measure the time averaged energy of the output, σ_y^2 , as follows:

$$\sigma_y^2 = \lim_{\substack{T_1 \rightarrow \infty \\ T_2 \rightarrow \infty}} \frac{1}{T_1 + T_2 + 1} \sum_{t=-T_1}^{T_2} E[y[t] \overline{y[t]}] \quad (6.56)$$

The following theorem is used to measure the gain of a single WSS source being processed by a plurality of multirate filterbanks.

Theorem 6.4.1. Assume source x is white noise with variance σ_x^2 . Then the output y after processing through the system given by Fig. 6.1, has time averaged energy given as follows:

$$\sigma_y^2 = \frac{\sigma_x^2}{2\pi D^2} \sum_{d_1=0}^{D-1} \sum_{d_2=0}^{D-1} \int_{-\frac{\pi}{D}}^{\frac{\pi}{D}} \left| \left(\sum_{n=0}^{N-1} \mathbf{Q}_{n, D \times D}(e^{jw}) \right)_{d_1, d_2} \right|^2 dw \quad (6.57)$$

where $\mathbf{Q}_{n, D \times D}(e^{jw}) = \mathbf{G}_{n, D \times C}(e^{jw}) \mathbf{F}_{n, C \times D}(e^{jw}) \mathbf{H}_{n, D \times D}(e^{jw})$ as defined in (6.53).

To simplify the notation of (6.57), we define the 2-norm of a $D \times D$ matrix valued function $\mathbf{Q}_{D \times D}(e^{jw})$ as

$$\|\mathbf{Q}_{D \times D}(e^{jw})\|_2^2 \equiv \sum_{d_1=0}^{D-1} \sum_{d_2=0}^{D-1} \int_{-\frac{\pi}{D}}^{\frac{\pi}{D}} |(\mathbf{Q}_{D \times D}(e^{jw}))_{d_1, d_2}|^2 dw \quad (6.58)$$

With the above definition, (6.57) becomes

$$\sigma_y^2 = \frac{\sigma_x^2}{2\pi D^2} \left\| \sum_{n=0}^{N-1} \mathbf{Q}_{n, D \times D} \right\|_2^2 \quad (6.59)$$

Proof. In Fig. 6.1, for a given interference x , the output y is a sum of N outputs from each microphone's filter bank, that is

$$y[t] = \sum_{n=0}^{N-1} y_n[t] \quad (6.60)$$

where y_n is the output of the n -th microphone's filter bank.

Using (6.60) and (6.46), we can verify that y is WSCS with period D . Hence, we can measure the energy of the output just over one period D as the following calculation shows:

$$\begin{aligned}
\sigma_y^2 &= \lim_{\substack{T_1 \rightarrow \infty \\ T_2 \rightarrow \infty}} \frac{1}{T_1 + T_2 + 1} \sum_{t=-T_1}^{T_2} E[y[t]]^2 \\
&= \lim_{N_p \rightarrow \infty} \frac{1}{D(2N_p + 1)} \sum_{p=-N_p}^{N_p} \sum_{l=0}^{D-1} E[y[l + pD] \overline{y[l + pD]}] \\
&= \frac{1}{D} \sum_{l=0}^{D-1} \lim_{N_p \rightarrow \infty} \frac{1}{(2N_p + 1)} \sum_{p=-N_p}^{N_p} E[y[l + pD] \overline{y[l + pD]}] \\
&= \frac{1}{D} \sum_{l=0}^{D-1} \lim_{N_p \rightarrow \infty} \frac{1}{(2N_p + 1)} (2N_p + 1) E[y[l] \overline{y[l]}] \\
&= \frac{1}{D} \sum_{l=0}^{D-1} E[y[l] \overline{y[l]}] \\
&= \frac{1}{D} \sum_{l=0}^{D-1} R_y[l, l] \tag{6.61}
\end{aligned}$$

Expanding the correlation of $y, R_y[l, l]$, shows that we are interested in the correlation of each of the N filterbank's outputs.

$$\begin{aligned}
R_y[l, l] &= E[y[l] \overline{y[l]}] \\
&= E\left[\sum_{n_1=0}^{N-1} y_{n_1}[l] \sum_{n_2=0}^{N-1} \overline{y_{n_2}[l]} \right] \\
&= \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} R_{y_{n_1}, y_{n_2}}[l, l] \tag{6.62}
\end{aligned}$$

Substituting (6.62) into (6.61) and changing the order of the finite summations, we have

$$\sigma_y^2 = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \frac{1}{D} \sum_{l=0}^{D-1} R_{y_{n_1}, y_{n_2}}[l, l] \tag{6.63}$$

We observe that the weighted inner sum is simply the time averaged energy of the cross correlation between filter bank outputs y_{n_1}, y_{n_2} (see (6.61)), that is

$$\sigma_y^2 = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \sigma_{y_{n_1}, y_{n_2}}^2 \tag{6.64}$$

The following derivation relates $\sigma_{y_{n_1}, y_{n_2}}^2$ to the trace of the cyclic spectral density matrix

$\mathbf{S}_{y_{n_1}, y_{n_2}}$.

$$\begin{aligned}
\sigma_{y_{n_1}, y_{n_2}}^2 &= \frac{1}{D} \sum_{l=0}^{D-1} R_{y_{n_1}, y_{n_2}}[l, l] \\
&= \frac{1}{D} \sum_{l=0}^{D-1} \sum_u R_{y_{n_1}, y_{n_2}}[l+u, l] \cdot \underbrace{\frac{1}{2\pi} \int_{-2\pi+\frac{\pi}{D}}^{\frac{\pi}{D}} e^{-jwu} dw}_{1 \text{ if } u=0 \text{ and } 0 \text{ otherwise}} \\
&= \frac{1}{2\pi} \int_{-2\pi+\frac{\pi}{D}}^{\frac{\pi}{D}} \sum_u \underbrace{\frac{1}{D} \sum_{l=0}^{D-1} R_{y_{n_1}, y_{n_2}}[l+u, l]}_{R_{y_{n_1}, y_{n_2}}^0[u]} e^{-jwu} dw \\
&= \frac{1}{2\pi} \int_{-2\pi+\frac{\pi}{D}}^{\frac{\pi}{D}} \underbrace{\sum_u R_{y_{n_1}, y_{n_2}}^0[u] e^{-jwu}}_{S_{y_{n_1}, y_{n_2}}^0(e^{jw})} dw \\
&= \frac{1}{2\pi} \sum_{r=0}^{D-1} \int_{-\frac{\pi}{D}}^{\frac{\pi}{D}} S_{y_{n_1}, y_{n_2}}^0(e^{j(w-w_{D,r})}) dw \\
&= \frac{1}{2\pi} \int_{-\frac{\pi}{D}}^{\frac{\pi}{D}} \sum_{r=0}^{D-1} S_{y_{n_1}, y_{n_2}}^0(e^{j(w-w_{D,r})}) dw \\
&= \frac{1}{2\pi} \int_{-\frac{\pi}{D}}^{\frac{\pi}{D}} \text{Tr}(\mathbf{S}_{y_{n_1}, y_{n_2}, D \times D}(e^{jw})) dw \tag{6.65}
\end{aligned}$$

The third and fourth line follow from the definition of cyclic correlation, (6.20), and cyclic spectral density, (6.23), respectively. The fifth line breaks apart the integration into D intervals each of length $\frac{2\pi}{D}$, and the last line follows the definition of $\mathbf{S}_{y_{n_1}, y_{n_2}, D \times D}(e^{jw})$, (6.50). Substituting (6.65) into (6.64) yields:

$$\sigma_y^2 = \frac{1}{2\pi} \int_{-\frac{\pi}{D}}^{\frac{\pi}{D}} \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \text{Tr}(\mathbf{S}_{y_{n_1}, y_{n_2}, D \times D}(e^{jw})) dw \tag{6.66}$$

Using (6.55), the definition of the Hilbert-Schmidt inner product, (2.10), and the definition of the Hilbert-Schmidt norm, (2.9), we can simplify the integrand of (6.66) as follows:

$$\begin{aligned}
\sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \text{Tr}(\mathbf{S}_{y_{n_1}, y_{n_2}, D \times D}(e^{jw})) &= \frac{\sigma_x^2}{D^2} \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \text{Tr}(\mathbf{Q}_{n_1, D \times D}(e^{jw}) \mathbf{Q}_{n_2, D \times D}^*(e^{jw})) \\
&= \frac{\sigma_x^2}{D^2} \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \langle \mathbf{Q}_{n_1, D \times D}(e^{jw}), \mathbf{Q}_{n_2, D \times D}^*(e^{jw}) \rangle_{HS} \\
&= \frac{\sigma_x^2}{D^2} \left\langle \sum_{n_1=0}^{N-1} \mathbf{Q}_{n_1, D \times D}(e^{jw}), \sum_{n_2=0}^{N-1} \mathbf{Q}_{n_2, D \times D}^*(e^{jw}) \right\rangle_{HS} \\
&= \frac{\sigma_x^2}{D^2} \left\| \sum_{n=0}^{N-1} \mathbf{Q}_{n, D \times D}(e^{jw}) \right\|_{HS}^2 \\
&= \frac{\sigma_x^2}{D^2} \sum_{d_1=0}^{D-1} \sum_{d_2=0}^{D-1} \left| \left(\sum_{n=0}^{N-1} \mathbf{Q}_{n, D \times D}(e^{jw}) \right)_{d_1, d_2} \right|^2 \quad (6.67)
\end{aligned}$$

Substituting (6.67) into (6.66) yields the desired

$$\sigma_y^2 = \frac{\sigma_x^2}{2\pi D^2} \sum_{d_1=0}^{D-1} \sum_{d_2=0}^{D-1} \int_{-\frac{\pi}{D}}^{\frac{\pi}{D}} \left| \left(\sum_{n=0}^{N-1} \mathbf{Q}_{n, D \times D}(e^{jw}) \right)_{d_1, d_2} \right|^2 dw$$

□

Chapter 7: Optimization Over Synthesis Filters

Using a Sparse Number of Subchannels

7.1 Overview

Our overall objective is to minimize the p -norm of interference source gains while both reconstructing perfectly the target source and using a sparse number of subchannels. In our problem setup, there are two types of sources, I interferences, whose gains we wish to attenuate and a single target, whose gain from system processing we wish to be exactly equal to 1. In other words, our system will process the target source with no distortion. Specifically, our desired optimization problem reads

$$\begin{aligned}
 \min_{\mathbf{G}_{n,D \times C}(e^{jw})} & \quad \left\| \left\{ \left\| \sum_{n=0}^{N-1} \mathbf{G}_{n,D \times C}(e^{jw}) \mathbf{F}_{n,C \times D}(e^{jw}) \mathbf{H}_{n,D \times D}^{(r)}(e^{jw}) \right\|_2^2 \right\}_{1 \leq r \leq L} \right\|_p \\
 \text{subject to} & \quad \sum_{n=0}^{N-1} \mathbf{G}_{n,D \times C}(e^{jw}) \mathbf{F}_{n,C \times D}(e^{jw}) \mathbf{H}_{n,D \times D}^{(0)}(e^{jw}) = D e^{jw} \text{diag} \left[\left(e^{-\frac{2\pi j d}{D}} \right)_{d=0}^{D-1} \right] \\
 & \quad \sum_{n=0}^{N-1} \left\| \left\{ \max_{-\pi \leq w \leq \pi} |(\mathbf{G}_{n,D \times C}(e^{jw}))_{0,l}| \right\}_{0 \leq l \leq C-1} \right\|_0 \leq V_S
 \end{aligned} \tag{7.1}$$

where $\mathbf{H}_{n,D \times D}^{(r)}(e^{jw})$ refers to the product of two frequency domain objects: the propaga-

tion of source r to microphone n and the target inversion filter specific to microphone n , D is the decimation factor, C is the number of subchannels per filterbank, and V_S is the desired number of active subchannels. Unfortunately, this is not a convex optimization problem.

We assume our set of $N \cdot C$ analysis filters, denoted by F , are fixed and known where N is the number of microphones and C is the number of subchannels for each microphone, that is

$$F = (F_{n,l}(e^{jw}))_{\substack{0 \leq n \leq N-1 \\ 0 \leq l \leq C-1 \\ w \in [0, 2\pi)}} \quad (7.2)$$

is a known quantity.

We further assume that the locations of the sources and microphones are known and hence the cascade of the source propagation and target inversion filters, denoted by H , is also known. There are $I + 1$ sources with I interferences and a single target source and N microphones and hence H consists of $(I + 1) \cdot N$ transfer functions. We define H as follows

$$H = (H_{r,n}(e^{jw}))_{\substack{0 \leq r \leq I \\ 0 \leq n \leq N-1 \\ w \in [0, 2\pi)}} \quad (7.3)$$

Our optimization is then over the $N \cdot C$ *synthesis filters*, denoted by G , where G is defined as follows

$$G = (G_{n,l}(e^{jw}))_{\substack{0 \leq n \leq N-1 \\ 0 \leq l \leq C-1 \\ w \in [0, 2\pi)}} \quad (7.4)$$

To make finding our unknown G computationally tractable, we will discretize the continuous variable w , $0 \leq w < 2\pi$, with N_f equally spaced points, that is we let

$$w = \frac{2\pi f}{N_f} \text{ with } f = 0, 1, \dots, N_f - 1 \quad (7.5)$$

To simplify our computations, we will also make the assumption that N_f , the number of discretized frequencies, is an even multiple of D , the downsampling factor, that is

$$N_f = M \cdot D \quad (7.6)$$

with M a positive even integer. In addition, to vectorize our computations, we will combine the indexes n and l over s (for subchannel) by letting

$$s = l + nC \text{ and } S = N \cdot C \quad (7.7)$$

Hence, the discretized version of unknown synthesis filters G , G_{compute} , consists of $S \cdot N_f$ unknown complex numbers, that is

$$G_{\text{compute}} = \left(G_{n,l} \left(e^{j \left(\frac{2\pi}{N_f} f \right)} \right) \right)_{\substack{0 \leq s \leq S-1 \\ 0 \leq f \leq N_f-1}} \quad (7.8)$$

By penalizing non-zero subchannel synthesis tap coefficients in our optimization problem with an absolute value penalty term, in the spirit of the LASSO algorithm given in [1], we will force certain subchannels to be considered inactive and create a sparse set of active subchannels. A subchannel will be considered inactive if its synthesis tap coefficients are zero or close to zero.

We will measure gain using the time-averaged energy given by (6.57). Notice that (6.57) assumes a fixed source x , and hence we will need to compute (6.57) for each of our $I + 1$ sources x_r .

In this chapter, we first review the prior work before detailing our problem solution. Next, we discuss the objective function which consists of two terms, the p -norm of interference source gains and the sparse subchannel penalty term. Afterwards, we discuss the target source perfect reconstruction constraint and then combine the results in the final section to give the full optimization problem.

7.2 Prior Work

Our work optimizes both the placement of microphones and the design of the multirate filterbanks processing the microphone's inputs *simultaneously*. To the best of our knowledge, this is a novel contribution. There has been extensive work, however, in both the areas of sensor placement and optimization of multirate filterbanks.

7.2.1 Sensor Placement

There has been extensive work in the sensor placement problem using a variety of strategies. In [26], the authors use simulated annealing to simultaneously optimize both weights and sensor locations on a linear array. In [27], the authors address a parameter estimation problem that includes sensor location using convex optimization. They relax a binary variable of a sensor being off, 0, or on, 1, by letting the variable instead be in the range of $[0, 1]$. In [28], the authors relax the binary variable of a sensor being off, 0, or on, 1, by converting the unknown vector to a matrix of 0s and 1s which belongs to the class of Steifel matrices. The relaxation is to a 1-d sphere and they jump to multiple dimensions using a greedy algorithm. The authors here are interested optimizing an objective criteria involving the KullbackLeibler divergence. [29] minimizes the side lobe produced by a sparse linear array using a greedy deletion algorithm.

[30] maximizes the Fisher Information in their optimization criteria. When formulating an optimization problem to choose sensors, they convert a matrix equality to a matrix inequality in their convex relaxation.

[31] address the problem of recovering K target sources emanating from a N possible locations using N sensors where $K \ll N$. They convert the problem to one of selecting rows that minimize the correlation coefficient and propose a genetic algorithm that is data-dependent.

[32] addresses the the following problem: Given a desired beampattern response and error tolerance, what is the sparsest sensor setup that will achieve the desired beampattern within the given error tolerance? The authors address this problem using reweighted $l - 1$ minimization and assuming the use of regularly spaced rectangular array, symmetric subarrays, and weights with conjugate symmetry. In [22], the authors address the same problem using a regularization approach inspired by Bayesian statistics. [33] also addresses the same problem. The author constrain microphone distances and guess a configuration and the perturb the configuration seeking a better solution. [34] address the same problem for a beampattern produced by a transmission array and propose a greedy addition algorithm.

In contrast, our work addresses the following problem: Given a fixed number of sensors, what is the best possible beampattern achievable?

7.2.2 Filterbank Optimization

There has also been extensive work in the optimization of filterbanks. [35], a work published in 1980, was an early forerunner in the field. The author optimizes a quadrature mirror filterbank to meet a user-given frequency response criteria. He minimizes both the ripple energy and out of band energy using a search algorithm whose success is highly dependent on both the starting point and step size. Similar to our proposed problem setup but for a single filterbank, [36] fixes the analysis filters and optimizes the synthesis filters to achieve the best possible reconstruction given a user-specified integer time delay. They convert their problem to a H_∞ problem to take advantage of existing software. In [37], the authors design a multi-dimensional perfect reconstruction filterbank where both the analysis and synthesis filter are FIR filters of equal length. They directly embed this non-linear and non-convex constraint into their optimization problem where the objective function measures the difference between a desired analysis filterbank and the

optimized analysis filterbank. The author of [38] designs perfect reconstruction and near perfect reconstruction filterbanks using a quadratic constrained least-squares optimization problem. The author solves a relaxed problem by converting the quadratic constraints to linear constraints. The authors of [39] design a cosine modulated filterbanks with a linear phase prototype filter using convex optimization by converting a non-convex set to a convex set. More recently, Davidson in [40] provides a novice-friendly tutorial of convex optimization techniques when designing FIR filters. Finally, [25] optimizes a single FIR filterbank to minimize the time-averaged mean-squared error when the high-frequency subband is dropped. We build off their techniques to derive an input/output relationship for our system consisting of a plurality of filterbanks.

7.3 Objective Function

Our objective function is the sum of two terms, one measuring the p -norm of the gain of the I interferences, $J_{I,p}$, and the other penalizing active subchannels, J_S , that is those subchannels with non-zero synthesis filters responses. Typical p -norms of interest would be $p = 1, 2, \infty$. We adjust the severity of our active subchannel penalty by *choosing* a non-negative constant λ to weight our active subchannel term J_S . The larger a λ chosen, the more severe the active subchannel penalty and the fewer number of active subchannels our optimization problem will recover. Conversely, the smaller a λ chosen, the less severe the active subchannel penalty and the greater number active subchannels our optimization problem will recover. By choosing λ equal to 0, we eliminate the active subchannel penalty altogether and allow the use of all $N \cdot C$ channels.

Our optimization is over $N \cdot C$ synthesis filter responses where N is the number of microphones and C is the number of subchannels for each microphone. We denote these set of filter responses by G defined in (7.4). Hence our objective function J can be written as

$$J(G) = J_{I,p}(G) + \lambda \cdot J_S(G) \quad (7.9)$$

G as stated above is a function of continuous variable w and to make $J(G)$ computationally tractable, we will discretize G using G_{compute} , defined in (7.8), and update both $J_{I,p}$ and J_S accordingly. Hence, what we will compute for our objective function $J(G)$ will be the approximation $J_{\text{compute}}(G_{\text{compute}})$, that is,

$$\begin{aligned} J(G) &\approx J_{\text{compute}}(G_{\text{compute}}) \\ &= J_{I,p,\text{compute}}(G_{\text{compute}}) + \lambda \cdot J_{S,\text{compute}}(G_{\text{compute}}) \end{aligned} \quad (7.10)$$

7.3.1 p -norm of Interference Gains Term

We let the p -norm of the interference gains be the p -norm of the interference sources' time-averaged energies, that is

$$J_{I,p}(G) = \|(\sigma_{y_r}^2)_r\|_p \quad (7.11)$$

where the time-averaged energy $\sigma_{y_r}^2$ is given by (6.57). We observe that (6.57) is a function of $\mathbf{Q}_{n,D \times D}$ which varies with n , the microphone index. The definition of $\mathbf{Q}_{n,D \times D}$, (6.53), further shows that for fixed n , $\mathbf{Q}_{n,D \times D}$ varies with $\mathbf{H}_{n,D \times D}$, the cascade of propagation and target and inversion filter responses, $\mathbf{F}_{n,C \times D}$, the analysis filter responses, and $\mathbf{G}_{n,D \times C}$, the synthesis filter responses. As stated above, we assume that both H and F , the set of target and inversion filter responses and analysis filter responses for all n respectively, are fixed and known and hence in our setup, the time-averaged energy only varies with synthesis filter responses for each n , that is G .

For the case $p = \infty$, $J_{I,p}(G)$ then becomes

$$J_{I,\infty}(G) = \max_r \sigma_{y_r}^2 \quad (7.12)$$

7.3.2 Discretization of $\sigma_{y_r}^2$

In the following derivations, we will discretize the value $\sigma_{y_r}^2$ over frequency w with a set of finite, uniformly spaced points to give $\sigma_{y_r,\text{compute}}^2$, a computationally tractable term. We will further assume all our sources have equal variance σ_x^2 to further simplify computations to give $\hat{\sigma}_{y_r,\text{compute}}^2$

To begin, we use (6.57) to expand $\sigma_{y_r}^2$ and observe that the d_1, d_2 term of the sum of N $\mathbf{Q}_{n,D \times D}$ matrices is equal to summing each of the d_1, d_2 terms of the same set of N matrices to get the following:

$$\begin{aligned} \sigma_{y_r}^2 &= \frac{\sigma_x^2}{2\pi D^2} \sum_{d_1=0}^{D-1} \sum_{d_2=0}^{D-1} \int_{-\frac{\pi}{D}}^{\frac{\pi}{D}} \left| \sum_{n=0}^{N-1} \mathbf{Q}_{n,D \times D}(e^{jw}) \right|_{d_1, d_2}^2 dw \\ &= \frac{\sigma_x^2}{2\pi D^2} \sum_{d_1=0}^{D-1} \sum_{d_2=0}^{D-1} \int_{-\frac{\pi}{D}}^{\frac{\pi}{D}} \left| \sum_{n=0}^{N-1} (\mathbf{Q}_{n,D \times D}(e^{jw}))_{d_1, d_2} \right|^2 dw \end{aligned} \quad (7.13)$$

Next, by observing that $\mathbf{H}_{n,D \times D}(e^{jw})$ is a diagonal matrix in the definition of $\mathbf{Q}_{n,D \times D}(e^{jw})$, (6.53), we can write the scalar expression $(\mathbf{Q}_{n,D \times D}(e^{jw}))_{d_1, d_2}$ as follows

$$(\mathbf{Q}_{n,D \times D}(e^{jw}))_{d_1, d_2} = \sum_{l=0}^{C-1} G_{n,l}(e^{j(w-w_{D,d_1})}) F_{n,l}(e^{j(w-w_{D,d_2})}) H_{r,n}(e^{j(w-w_{D,d_2})}) \quad (7.14)$$

Substituting (7.14) into (7.13) yields,

$$\begin{aligned}
\sigma_{y_r}^2 &= \frac{\sigma_x^2}{2\pi D^2} \sum_{d_1=0}^{D-1} \sum_{d_2=0}^{D-1} \cdot \\
&\int_{-\frac{\pi}{D}}^{\frac{\pi}{D}} \left| \sum_{n=0}^{N-1} \sum_{l=0}^{C-1} G_{n,l}(e^{j(w-w_{D,d_1})}) F_{n,l}(e^{j(w-w_{D,d_2})}) H_{r,n}(e^{j(w-w_{D,d_2})}) \right|^2 dw
\end{aligned} \tag{7.15}$$

We can then approximate the integral in (7.15) as follows:

$$\begin{aligned}
&\int_{-\frac{\pi}{D}}^{\frac{\pi}{D}} \left| \sum_{n=0}^{N-1} \sum_{l=0}^{C-1} G_{n,l}(e^{j(w-w_{D,d_1})}) F_{n,l}(e^{j(w-w_{D,d_2})}) H_{r,n}(e^{j(w-w_{D,d_2})}) \right|^2 dw \\
&= \int_{-\frac{\pi}{D}}^{\frac{\pi}{D}} \left| \sum_{n=0}^{N-1} \sum_{l=0}^{C-1} G_{n,l}(e^{j(w-\frac{2\pi d_1}{D})}) F_{n,l}(e^{j(w-\frac{2\pi d_2}{D})}) H_{r,n}(e^{j(w-\frac{2\pi d_2}{D})}) \right|^2 dw \\
&\approx \frac{2\pi}{N_f} \sum_{f=-\frac{N_F}{2D}}^{\frac{N_F}{2D}-1} \left| \sum_{n=0}^{N-1} \sum_{l=0}^{C-1} G_{n,l}(e^{j2\pi(\frac{f}{N_f}-\frac{d_1}{D})}) F_{n,l}(e^{j2\pi(\frac{f}{N_f}-\frac{d_2}{D})}) H_{r,n}(e^{j2\pi(\frac{f}{N_f}-\frac{d_2}{D})}) \right|^2
\end{aligned} \tag{7.16}$$

Substituting (7.16) for the integral in (7.15), yields $\sigma_{y_r, \text{compute}}^2$, an approximation of $\sigma_{y_r}^2$

that is computationally tractable, that is

$$\begin{aligned}
\sigma_{y_r}^2 &\approx \sigma_{y_r, \text{compute}}^2 \\
&= \frac{\sigma_{x_r}^2}{D^2 N_f} \sum_{d_1=0}^{D-1} \sum_{d_2=0}^{D-1} \cdot \\
&\sum_{f=-\frac{N_F}{2D}}^{\frac{N_F}{2D}-1} \left| \sum_{n=0}^{N-1} \sum_{l=0}^{C-1} G_{n,l}(e^{j2\pi(\frac{f}{N_f}-\frac{d_1}{D})}) F_{n,l}(e^{j2\pi(\frac{f}{N_f}-\frac{d_2}{D})}) H_{r,n}(e^{j2\pi(\frac{f}{N_f}-\frac{d_2}{D})}) \right|^2
\end{aligned} \tag{7.17}$$

If we assume that each source x_r has the same variance, that is $\sigma_{x_r}^2 = \sigma_x^2$ for all r , we can treat the leading coefficient as a constant and (7.17) becomes

$$\hat{\sigma}_{y_r, \text{compute}}^2 = \frac{\sigma_x^2}{D^2 N_f} \cdot \sum_{d_1=0}^{D-1} \sum_{d_2=0}^{D-1} \sum_{f=-\frac{N_F}{2D}}^{\frac{N_F}{2D}-1} \left| \sum_{n=0}^{N-1} \sum_{l=0}^{C-1} G_{n,l}(e^{j2\pi(\frac{f}{N_f}-\frac{d_1}{D})}) F_{n,l}(e^{j2\pi(\frac{f}{N_f}-\frac{d_2}{D})}) H_{r,n}(e^{j2\pi(\frac{f}{N_f}-\frac{d_2}{D})}) \right|^2 \quad (7.18)$$

To simplify notation, we define the following equality:

$$\dot{F}_{n,l,r}(e^{j2\pi(\frac{f}{N_f}-\frac{d_2}{D})}) \equiv F_{n,l}(e^{j2\pi(\frac{f}{N_f}-\frac{d_2}{D})}) H_{r,n}(e^{j2\pi(\frac{f}{N_f}-\frac{d_2}{D})}) \quad (7.19)$$

We note that the value of (7.19) is known by assumption. We also combine the summations over n and l inside the magnitude squared into a single summation over s (for subchannel) by letting

$$s = l + nC \text{ and } S = N \cdot C \quad (7.20)$$

Hence, (7.18) becomes

$$\hat{\sigma}_{y_r, \text{compute}}^2 = \frac{\sigma_x^2}{D^2 N_f} \sum_{d_1=0}^{D-1} \sum_{d_2=0}^{D-1} \sum_{f=-\frac{N_F}{2D}}^{\frac{N_F}{2D}-1} \left| \sum_{s=0}^{S-1} G_s(e^{j2\pi(\frac{f}{N_f}-\frac{d_1}{D})}) \dot{F}_{s,r}(e^{j2\pi(\frac{f}{N_f}-\frac{d_2}{D})}) \right|^2 \quad (7.21)$$

To efficiently compute (7.21), we would like to rewrite it as a product of a row vector, matrix, and column vector where the row and column vector contain the unknown discretized frequency responses of all S synthesis filters. To begin, we first expand the magnitude squared of (7.21) and rearrange the finite summations to get

$$\hat{\sigma}_{y_r, \text{compute}}^2 = \frac{\sigma_{x^2}}{D^2 N_f} \sum_{d_1=0}^{D-1} \sum_{f=-\frac{N_F}{2D}}^{\frac{N_F}{2D}-1} \sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} G_{s_1}(e^{j2\pi(\frac{f}{N_f}-\frac{d_1}{D})}) \overline{G}_{s_2}(e^{j2\pi(\frac{f}{N_f}-\frac{d_1}{D})}) \cdot \underbrace{\left(\sum_{d_2=0}^{D-1} \dot{F}_{s_1,r}(e^{j2\pi(\frac{f}{N_f}-\frac{d_2}{D})}) \overline{\dot{F}}_{s_2,r}(e^{j2\pi(\frac{f}{N_f}-\frac{d_2}{D})}) \right)}_{\Phi_r(s_1, s_2, f)} \quad (7.22)$$

We choose the symbol Φ since it represents capital “F” in Greek. In order to reduce the summations over both d_1 and f to a single summation over f , we make the assumption that N_f is an even multiple of D , that is

$$N_f = M \cdot D \quad (7.23)$$

with M a positive even integer. Rewriting the arguments to G_{s_1} and \overline{G}_{s_2} , we then have

$$\hat{\sigma}_{y_r, \text{compute}}^2 = \frac{\sigma_{x^2}}{D^2 N_f} \sum_{d_1=0}^{D-1} \sum_{f=-\frac{N_F}{2D}}^{\frac{N_F}{2D}-1} \sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} G_{s_1}(e^{j2\pi(\frac{f-Md_1}{N_f})}) \overline{G}_{s_2}(e^{j2\pi(\frac{f-Md_1}{N_f})}) \Phi_r(s_1, s_2, f) \quad (7.24)$$

A calculation in the appendix, (A.1), shows that $\Phi_r(s_1, s_2, f)$ is M -periodic in f , that is

$$\Phi_r(s_1, s_2, f - M) = \Phi_r(s_1, s_2, f) \quad (7.25)$$

Hence, we then have

$$\hat{\sigma}_{y_r, \text{compute}}^2 = \frac{\sigma_{x^2}}{D^2 N_f} \sum_{d_1=0}^{D-1} \sum_{f=-\frac{N_F}{2D}}^{\frac{N_F}{2D}-1} \sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} G_{s_1}(e^{j2\pi(\frac{f-Md_1}{N_f})}) \overline{G}_{s_2}(e^{j2\pi(\frac{f-Md_1}{N_f})}) \Phi_r(s_1, s_2, f - Md_1) \quad (7.26)$$

By letting $f = -\frac{N_F}{2D}, \dots, \frac{N_F}{2D} - 1$ and $d_1 = 0, 1, \dots, D - 1$ in the relationship $\hat{f} = (f - Md_1) \bmod N_f$, we have $\hat{f} = 0, 1, \dots, N_f - 1$. In addition, we observe that the z -transform is 2π

periodic in w for $z = e^{jw}$ which means we can re-index $f - Md_1 \bmod N_f$ in the arguments of G_{s_1} and \overline{G}_{s_2} . Finally, since $N_f = M \cdot D$ by assumption, $\Phi_r(s_1, s_2, f)$ is also N_f periodic in f which means we can also re-index $f - Md_1 \bmod N_f$ in the arguments of Φ . We can then combine the summations over d_1 and f into one summation over f in (7.26) as follows

$$\hat{\sigma}_{y_r, \text{compute}}^2 = \frac{\sigma_{x^2}}{D^2 N_f} \cdot \sum_{f=0}^{N_f-1} \sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} G_{s_1}(e^{j2\pi(\frac{f}{N_f})}) \overline{G}_{s_2}(e^{j2\pi(\frac{f}{N_f})}) \Phi_r(s_1, s_2, f) \quad (7.27)$$

If we make the assumption that the analysis and the synthesis filters' FIR coefficients are real, the term $G_{s_1}(e^{j2\pi(\frac{f}{N_f})}) \overline{G}_{s_2}(e^{j2\pi(\frac{f}{N_f})}) \Phi_r(s_1, s_2, f)$ is conjugate symmetric in continuous variable f . Hence we can write (7.27) as follows:

$$\begin{aligned} \hat{\sigma}_{y_r, \text{compute}}^2 &= \frac{\sigma_{x^2}}{D^2 N_f} \left[2 \sum_{f=1}^{\frac{N_f}{2}-1} \sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} G_{s_1}(e^{j2\pi(\frac{f}{N_f})}) \overline{G}_{s_2}(e^{j2\pi(\frac{f}{N_f})}) \Phi_r(s_1, s_2, f) \right. \\ &\quad + \sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} G_{s_1}(e^{j2\pi(\frac{0}{N_f})}) \overline{G}_{s_2}(e^{j2\pi(\frac{0}{N_f})}) \Phi_r(s_1, s_2, 0) \\ &\quad \left. + \sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} G_{s_1}(e^{j2\pi(\frac{N_f}{N_f})}) \overline{G}_{s_2}(e^{j2\pi(\frac{N_f}{N_f})}) \Phi_r(s_1, s_2, \frac{N_f}{2}) \right] \quad (7.28) \end{aligned}$$

For a fixed f , any of the three summations over s_1 and s_2 of (7.27) can be expressed as product of a row vector, matrix, and column vector, that is

$$\begin{aligned} &\sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} G_{s_1}(e^{j2\pi(\frac{f}{N_f})}) \overline{G}_{s_2}(e^{j2\pi(\frac{f}{N_f})}) \Phi_r(s_1, s_2, f) \\ &= \mathbf{G}_{1 \times S}(e^{j2\pi(\frac{f}{N_f})}) \mathbf{\Phi}_{r, S \times S}(f) \mathbf{G}_{1 \times S}^*(e^{j2\pi(\frac{f}{N_f})}) \quad (7.29) \end{aligned}$$

where

$$\mathbf{G}_{1 \times S}(e^{j2\pi(\frac{f}{N_f})}) = [\mathbf{G}_{0,1 \times C}(e^{j2\pi(\frac{f}{N_f})}), \mathbf{G}_{1,1 \times C}(e^{j2\pi(\frac{f}{N_f})}), \dots, \mathbf{G}_{N-1,1 \times C}(e^{j2\pi(\frac{f}{N_f})})] \quad (7.30)$$

is a row vector, size $1 \times S$, containing all S synthesis filters' responses at discretized frequency f . The entries of the square matrix $\Phi_{r,S \times S}(f)$, size $S \times S$, is given as

$$(\Phi_{r,S \times S}(f))_{s_1, s_2} = \Phi_r(s_1, s_2, f) = \sum_{d_2=0}^{D-1} \dot{F}_{s_1, r}(e^{j2\pi(\frac{f}{N_f} - \frac{d_2}{D})}) \overline{\dot{F}_{s_2, r}(e^{j2\pi(\frac{f}{N_f} - \frac{d_2}{D})})} \quad (7.31)$$

In addition, $\Phi_{r,S \times S}(f)$ can be expressed as the product of the analysis matrix $\dot{\mathbf{F}}_{r,S \times D}(e^{j2\pi(\frac{f}{N_f})})$ and its conjugate that is

$$\Phi_{r,S \times S}(f) = \dot{\mathbf{F}}_{r,S \times D}(e^{j2\pi(\frac{f}{N_f})}) \dot{\mathbf{F}}_{r,S \times D}^*(e^{j2\pi(\frac{f}{N_f})}) \quad (7.32)$$

where the matrix $\dot{\mathbf{F}}_{r,S \times D}(e^{j2\pi(\frac{f}{N_f})})$, size S by D , is defined as

$$\dot{\mathbf{F}}_{r,S \times D}(e^{j2\pi(\frac{f}{N_f})}) = \begin{bmatrix} \dot{\mathbf{F}}_{r,0,1 \times C}(e^{j2\pi(\frac{f}{N_f})}) & \dot{\mathbf{F}}_{r,1,1 \times C}(e^{j2\pi(\frac{f}{N_f})}) & \dots & \dot{\mathbf{F}}_{r,N-1,1 \times C}(e^{j2\pi(\frac{f}{N_f})}) \\ \dot{\mathbf{F}}_{r,0,1 \times C}(e^{j2\pi(\frac{f}{N_f} - \frac{1}{D})}) & \dot{\mathbf{F}}_{r,1,1 \times C}(e^{j2\pi(\frac{f}{N_f} - \frac{1}{D})}) & \dots & \dot{\mathbf{F}}_{r,N-1,1 \times C}(e^{j2\pi(\frac{f}{N_f} - \frac{1}{D})}) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{\mathbf{F}}_{r,0,1 \times C}(e^{j2\pi(\frac{f}{N_f} - \frac{D-1}{D})}) & \dot{\mathbf{F}}_{r,1,1 \times C}(e^{j2\pi(\frac{f}{N_f} - \frac{D-1}{D})}) & \dots & \dot{\mathbf{F}}_{r,N-1,1 \times C}(e^{j2\pi(\frac{f}{N_f} - \frac{D-1}{D})}) \end{bmatrix}^T \quad (7.33)$$

and consists of $D \cdot N$ column vectors, size C by 1, defined in row vector notation as

$$\dot{\mathbf{F}}_{r,n,1 \times C}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})}) = H_{r,n}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})}) \cdot [F_{n,0}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})}), F_{n,1}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})}), \dots, F_{n,C-1}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})})] \quad (7.34)$$

with $d \in \{0, 1, \dots, D-1\}$. Hence, we can further rewrite (7.29) as follows

$$\begin{aligned}
& \sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} G_{s_1}(e^{j2\pi(\frac{f}{N_f})}) \overline{G_{s_2}}(e^{j2\pi(\frac{f}{N_f})}) \Phi_r(s_1, s_2, f) \\
&= \mathbf{G}_{1 \times S}(e^{j2\pi(\frac{f}{N_f})}) \dot{\mathbf{\Phi}}_{r, S \times S}(f) \mathbf{G}_{1 \times S}^*(e^{j2\pi(\frac{f}{N_f})}) \\
&= \mathbf{G}_{1 \times S}(e^{j2\pi(\frac{f}{N_f})}) \dot{\mathbf{F}}_{r, S \times D}(e^{j2\pi(\frac{f}{N_f})}) \dot{\mathbf{F}}_{r, S \times D}^*(e^{j2\pi(\frac{f}{N_f})}) \mathbf{G}_{1 \times S}^*(e^{j2\pi(\frac{f}{N_f})}) \\
&= \|\dot{\mathbf{F}}_{r, S \times D}^*(e^{j2\pi(\frac{f}{N_f})}) \mathbf{G}_{1 \times S}^*(e^{j2\pi(\frac{f}{N_f})})\|^2
\end{aligned} \tag{7.35}$$

We can then express the right hand side of (7.28) as a product of a block diagonal matrix and column vector, that is,

$$\hat{\sigma}_{y_r, \text{compute}}^2 = \frac{\sigma_{x^2}}{D^2 N_f} \cdot 2 \|\dot{\mathbf{F}}_{r, S(\frac{N_f}{2}+1) \times D(\frac{N_f}{2}+1)}^*(e^{j2\pi(\frac{f}{N_f})}) \mathbf{G}_{1 \times S(\frac{N_f}{2}+1)}^*(e^{j2\pi(\frac{f}{N_f})})\|^2 \tag{7.36}$$

and by transpose and conjugation we have

$$\hat{\sigma}_{y_r, \text{compute}}^2 = \frac{\sigma_{x^2}}{D^2 N_f} \cdot 2 \|\mathbf{G}_{1 \times S(\frac{N_f}{2}+1)}(e^{j2\pi(\frac{f}{N_f})}) \dot{\mathbf{F}}_{r, S(\frac{N_f}{2}+1) \times D(\frac{N_f}{2}+1)}(e^{j2\pi(\frac{f}{N_f})})\|^2 \tag{7.37}$$

where

$$\begin{aligned}
& \dot{\mathbf{F}}_{r, S(\frac{N_f}{2}+1) \times D(\frac{N_f}{2}+1)}(e^{j2\pi(\frac{f}{N_f})}) = \\
& \text{diag} \left(\frac{1}{\sqrt{2}} \dot{\mathbf{F}}_{r, S \times D}(e^{j2\pi(0)}), \dot{\mathbf{F}}_{r, S \times D}(e^{j2\pi(\frac{1}{N_f})}), \dots, \dot{\mathbf{F}}_{r, S \times D}(e^{j2\pi(\frac{N_f}{2}-1)}), \frac{1}{\sqrt{2}} \dot{\mathbf{F}}_{r, S \times D}(e^{j2\pi(\frac{N_f}{2})}) \right)
\end{aligned} \tag{7.38}$$

and

$$\mathbf{G}_{1 \times S(\frac{N_f}{2}+1)}(e^{j2\pi(\frac{f}{N_f})}) = \left[\mathbf{G}_{1 \times S}(e^{j2\pi(0)}) \quad \mathbf{G}_{1 \times S}(e^{j2\pi(\frac{1}{N_f})}) \quad \dots \quad \mathbf{G}_{1 \times S}(e^{j2\pi(\frac{N_f}{2})}) \right] \tag{7.39}$$

7.3.3 Computationally Tractable Approximation of $J_I(G)$

Using (7.37), we then have the computationally tractable approximation of (7.11), that is

$$\begin{aligned}
 J_{I,p}(G) &= \|(\sigma_{y_r}^2)_r\|_p \\
 &\approx \|(\hat{\sigma}_{y_r, \text{compute}}^2)_r\|_p \\
 &= J_{I,p, \text{compute}}(G_{\text{compute}})
 \end{aligned} \tag{7.40}$$

7.3.4 Sparse Subchannels Penalty Term

In the following section, we will derive a computationally tractable and efficient sparse subchannel penalty term $J_{S, \text{compute}}(G_{\text{compute}})$. The derivation begins by defining $J_{S, \text{sgn}}(G)$, which counts the number of active subchannels by seeing whether each subchannel's synthesis filter frequency response is non-zero or not. Next, the continuous frequency variable w is discretized along a finite, uniformly spaced set of points to give the computationally tractable term $J_{S, \text{sgn}, \text{compute}}(G_{\text{compute}})$. Finally, an $L-1$ like penalty is substituted to not only increase computational efficiency but also to induce sparse solutions to give the desired $J_{S, \text{compute}}(G_{\text{compute}})$.

Our goal is to find an objective function that not only minimizes the gain (time-averaged energy) of interference sources but also encourages sparse *subchannels*, that is only a few of the $N * C$ subchannels are active. As before, N is the number of microphones, and C is the number of subchannels of each microphone. We say a subchannel is inactive if its synthesis filter frequency response is zero or very small in magnitude. We measure the number of active subchannels as follows:

$$J_{S, \text{sgn}}(G) = \sum_{n=0}^{N-1} \sum_{l=0}^{C-1} \text{sgn}^2 \left(\max_{0 \leq w < \pi} |G_{n,l}(e^{jw})| \right) \tag{7.41}$$

where $\text{sgn}^2(x)$ is 1 if $x < 0$, 0 if $x = 0$, and 1 if $x > 0$. In other words, a channel is considered active if any portion of its frequency response is non-zero. We note that $0 \leq w < \pi$ rather than $0 \leq w < 2\pi$ since we assume our filter taps are real and hence the frequency response is conjugate symmetric. (7.41) counts the number of active subchannels and since sgn is applied to the max of absolute values, the value of (7.41) lies in the appropriate range of 0 to $N \cdot C$.

To make (7.41) computationally tractable, we discretize the continuous variable w using N_f points as in (7.16) so (7.41) becomes

$$\begin{aligned} J_{S,\text{sgn}}(G) &\approx J_{S,\text{sgn,compute}}(G_{\text{compute}}) \\ &= \sum_{n=0}^{N-1} \sum_{l=0}^{C-1} \text{sgn}^2 \left(\max_{f \in \{0,1,\dots,N_f/2\}} |G_{n,l}(e^{j2\pi \frac{f}{N_f}})| \right) \end{aligned} \quad (7.42)$$

We combine the summations over n and l into a single summation over s using (7.20) as before. In addition, similar to the spirit of Compressive Sampling, we replace the sgn^2 function by the absolute value. Hence (7.42) becomes

$$\boxed{J_{S,\text{abs,compute}}(G_{\text{compute}}) = \sum_{s=0}^{S-1} \left(\max_{f \in \{0,1,\dots,N_f/2\}} |G_s(e^{j2\pi \frac{f}{N_f}})| \right)} \quad (7.43)$$

In the objective function that we will compute, (7.10), we will use (7.43) and hence we will shorten its notation as follows:

$$J_{S,\text{compute}}(G_{\text{compute}}) = J_{S,\text{abs,compute}}(G_{\text{compute}}) \quad (7.44)$$

7.4 Target Source Perfect Reconstruction Condition

The target perfect reconstruction condition, TPR , is given by (6.9). To make (6.9) computationally tractable, we discretize the continuous variable w using N_f points as in (7.16) so (6.9) becomes for $f = 0, 1, \dots, N_f - 1$

$$\begin{aligned} TPR_{\text{compute}}(f, d) &= \sum_{n=0}^{N-1} H_{0,n}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})}) \sum_{l=0}^{C-1} G_{n,l}(e^{j2\pi(\frac{f}{N_f})}) F_{n,l}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})}) \\ &= \begin{cases} D e^{j2\pi(\frac{f}{N_f})(-\Delta)} & \text{if } d = 0; \\ 0 & \text{if } 1 \leq d \leq D - 1 \end{cases} \end{aligned} \quad (7.45)$$

Since we have D constraints for each of our N_f discretized frequencies, the TPR condition now has a total of $N_f \cdot D$ constraints.

In matrix-vector form and analogous to (6.10), the D target perfect reconstruction of conditions of (7.45) for a fixed $f \in \{0, 1, \dots, N_f - 1\}$ can be written as

$$TPR_{\text{compute}}(f) = \dot{\mathbf{F}}_{S \times D}^T(e^{j2\pi(\frac{f}{N_f})}) \mathbf{G}_{1 \times S}^T(e^{j2\pi(\frac{f}{N_f})}) = D e^{j2\pi(\frac{f}{N_f})(-\Delta)} \cdot \mathbf{e}_{0,1 \times D}^T \quad (7.46)$$

where $S = N \cdot C$, the number of subchannels. The matrix $\dot{\mathbf{F}}_{S \times D}^T(f)$, size D by S , is defined as

$$\dot{\mathbf{F}}_{S \times D}^T(e^{j2\pi(\frac{f}{N_f})}) = \begin{bmatrix} \dot{\mathbf{F}}_{0,1 \times C}(e^{j2\pi(\frac{f}{N_f})}) & \dot{\mathbf{F}}_{1,1 \times C}(e^{j2\pi(\frac{f}{N_f})}) & \cdots & \dot{\mathbf{F}}_{N-1,1 \times C}(e^{j2\pi(\frac{f}{N_f})}) \\ \dot{\mathbf{F}}_{0,1 \times C}(e^{j2\pi(\frac{f}{N_f} - \frac{1}{D})}) & \dot{\mathbf{F}}_{1,1 \times C}(e^{j2\pi(\frac{f}{N_f} - \frac{1}{D})}) & \cdots & \dot{\mathbf{F}}_{N-1,1 \times C}(e^{j2\pi(\frac{f}{N_f} - \frac{1}{D})}) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{\mathbf{F}}_{0,1 \times C}(e^{j2\pi(\frac{f}{N_f} - \frac{D-1}{D})}) & \dot{\mathbf{F}}_{1,1 \times C}(e^{j2\pi(\frac{f}{N_f} - \frac{D-1}{D})}) & \cdots & \dot{\mathbf{F}}_{N-1,1 \times C}(e^{j2\pi(\frac{f}{N_f} - \frac{D-1}{D})}) \end{bmatrix} \quad (7.47)$$

and consists of $D \cdot N$ row vectors, size 1 by C , defined as

$$\begin{aligned} \mathbf{F}_{n,1 \times C}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})}) = \\ H_{0,n}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})}) \cdot [F_{n,0}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})}), F_{n,1}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})}), \dots, F_{n,C-1}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})})] \end{aligned} \quad (7.48)$$

with $d \in \{0, 1, \dots, D - 1\}$.

The column vector $\mathbf{G}_{1 \times S}^T(e^{j2\pi(\frac{f}{N_f})})$, size S by 1, is defined as

$$\mathbf{G}_{1 \times S}^T(e^{j2\pi(\frac{f}{N_f})}) = \begin{bmatrix} \mathbf{G}_{0,1 \times C}^T(e^{j2\pi(\frac{f}{N_f})}) \\ \mathbf{G}_{1,1 \times C}^T(e^{j2\pi(\frac{f}{N_f})}) \\ \vdots \\ \mathbf{G}_{N-1,1 \times C}^T(e^{j2\pi(\frac{f}{N_f})}) \end{bmatrix} \quad (7.49)$$

and consists of N column vectors, size C by 1, defined as

$$\mathbf{G}_{n,1 \times C}^T(e^{j2\pi(\frac{f}{N_f})}) = [G_{n,0}(e^{j2\pi(\frac{f}{N_f})}), G_{n,1}(e^{j2\pi(\frac{f}{N_f})}), \dots, G_{n,C-1}(e^{j2\pi(\frac{f}{N_f})})]^T \quad (7.50)$$

The column vector $\mathbf{e}_{k,1 \times D}^T$, size $D \times 1$, is defined as the $D \times 1$ zero vector but with the k -th entry set to 1, that is

$$\mathbf{e}_{k,1 \times D}^T = [0, \dots, 0, \underbrace{1}_{k\text{-th entry of } D \text{ size vector}}, 0, \dots, 0]^T \quad (7.51)$$

We note that if we use the same set of C analysis filters for each filterbank that is

$$\begin{aligned} [F_{n_1,0}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})}), F_{n_1,1}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})}), \dots, F_{n_1,C-1}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})})] \\ = [F_{n_2,0}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})}), F_{n_2,1}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})}), \dots, F_{n_2,C-1}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})})] \end{aligned} \quad (7.52)$$

for all $0 \leq n_1, n_2 \leq N - 1$, $0 \leq d \leq D - 1$, and $0 \leq f \leq N_f - 1$, and the target inversion pre-filter removes the effect on phase from propagation perfectly, that is

$$H_{0,n}(e^{j2\pi(\frac{f}{N_f} - \frac{d}{D})}) = \alpha_{0,n} \quad (7.53)$$

for all $0 \leq n \leq N - 1$, $0 \leq d \leq D - 1$, and $0 \leq f \leq N_f - 1$, then (7.47) is of rank = $\min(D, C)$ since every C columns are scalar multiples of the previous C columns. Since we need to fulfill D constraints, these two additional assumptions imply that $D \leq C$.

If we assume that all filter taps are real, then we can almost halve the number of constraints using the conjugate symmetry of filter responses and the 2π periodicity in w in the z -transform for $z = e^{jw}$. First if (7.46) holds for $0 \leq f \leq N_f - 1$, then the conjugate of the entire equation also holds, that is

$$\overline{\dot{\mathbf{F}}_{S \times D}^T(e^{j2\pi(\frac{f}{N_f})}) \mathbf{G}_{1 \times S}^T(e^{j2\pi(\frac{f}{N_f})})} = \overline{D \cdot \mathbf{e}_{0,1 \times D}^T} = D \cdot \mathbf{e}_{0,1 \times D}^T \quad (7.54)$$

where the last equality follows because $\mathbf{e}_{0,1 \times D}^T$ contains all real entries. Next, we have

$$\begin{aligned} \overline{\dot{\mathbf{F}}_{S \times D}^T(e^{j2\pi(\frac{f}{N_f})}) \mathbf{G}_{1 \times S}^T(e^{j2\pi(\frac{f}{N_f})})} &= D \cdot \mathbf{e}_{0,1 \times D}^T \\ \dot{\mathbf{F}}_{S \times D}^T(e^{j2\pi(\frac{-f}{N_f})}) \mathbf{G}_{1 \times S}^T(e^{j2\pi(\frac{-f}{N_f})}) &= D \cdot \mathbf{e}_{0,1 \times D}^T \\ \dot{\mathbf{F}}_{S \times D}^T(e^{j2\pi(\frac{-f \bmod N_f}{N_f})}) \mathbf{G}_{1 \times S}^T(e^{j2\pi(\frac{-f \bmod N_f}{N_f})}) &= D \cdot \mathbf{e}_{0,1 \times D}^T \end{aligned}$$

where the second line follows from conjugate symmetry and the third line follows from 2π periodicity in w . In summary, we have shown that if $TPR_{\text{compute}}(f)$ holds, so does $TPR_{\text{compute}}(-f \bmod N_f)$. Hence, we can express the constraints of (7.46) for all f as a product of a block-diagonal matrix-vector multiply, that is

$$\dot{\mathbf{F}}_{\left(\frac{N_f}{2}+1\right)S \times \left(\frac{N_f}{2}+1\right)D}^T \mathbf{G}_{1 \times \left(\frac{N_f}{2}+1\right)S}^T = D \cdot \dot{\mathbf{E}}_{0,1 \times \left(\frac{N_f}{2}+1\right)D}^T \quad (7.55)$$

and by transposition we have

$$\mathbf{G}_{1 \times \left(\frac{N_f}{2}+1\right)S} \dot{\mathbf{F}}_{\left(\frac{N_f}{2}+1\right)S \times \left(\frac{N_f}{2}+1\right)D} = D \cdot \dot{\mathbf{E}}_{0,1 \times \left(\frac{N_f}{2}+1\right)D} \quad (7.56)$$

where the block diagonal matrix $\dot{\mathbf{F}}_{\left(\frac{N_f}{2}+1\right)S \times \left(\frac{N_f}{2}+1\right)D}$, size $\left(\frac{N_f}{2} + 1\right)S \times \left(\frac{N_f}{2} + 1\right)D$, is given by

$$\begin{aligned} & \dot{\mathbf{F}}_{\left(\frac{N_f}{2}+1\right)S \times \left(\frac{N_f}{2}+1\right)D} = \\ & \text{diag}\left(\frac{1}{\sqrt{2}}\dot{\mathbf{F}}_{S \times D}(e^{j2\pi\left(\frac{0}{N_f}\right)}), \dot{\mathbf{F}}_{S \times D}(e^{j2\pi\left(\frac{1}{N_f}\right)}), \dots, \dot{\mathbf{F}}_{S \times D}(e^{j2\pi\left(\frac{N_f-1}{N_f}\right)}), \frac{1}{\sqrt{2}}\dot{\mathbf{F}}_{S \times D}(e^{j2\pi\left(\frac{N_f}{N_f}\right)})\right) \end{aligned} \quad (7.57)$$

the row vector of unknowns $\mathbf{G}_{1 \times \left(\frac{N_f}{2}+1\right)S}$, size $1 \times \left(\frac{N_f}{2} + 1\right)S$, is given by

$$\mathbf{G}_{1 \times \left(\frac{N_f}{2}+1\right)S} = [\mathbf{G}_{1 \times S}(e^{j2\pi\left(\frac{0}{N_f}\right)}), \mathbf{G}_{1 \times S}(e^{j2\pi\left(\frac{1}{N_f}\right)}), \dots, \mathbf{G}_{1 \times S}(e^{j2\pi\left(\frac{N_f}{N_f}\right)})], \quad (7.58)$$

and the row vector of constraints $\mathbf{E}_{0,1 \times \left(\frac{N_f}{2}+1\right)D}$, size $1 \times \left(\frac{N_f}{2} + 1\right)D$, is given by

$$\begin{aligned} & \dot{\mathbf{E}}_{0,1 \times \left(\frac{N_f}{2}+1\right)D} = \\ & \underbrace{\left[\frac{1}{\sqrt{2}}e^{j2\pi\left(\frac{0}{N_f}\right)(-\Delta)}\mathbf{e}_{0,1 \times D}, e^{j2\pi\left(\frac{1}{N_f}\right)(-\Delta)}\mathbf{e}_{0,1 \times D}, \dots, \right]}_{f=0} \underbrace{\left[e^{j2\pi\left(\frac{N_f-1}{N_f}\right)(-\Delta)}\mathbf{e}_{0,1 \times D}, \frac{1}{\sqrt{2}}e^{j2\pi\left(\frac{N_f}{N_f}\right)(-\Delta)}\mathbf{e}_{0,1 \times D}\right]}_{f=\frac{N_f}{2}} \end{aligned} \quad (7.59)$$

As before, S represents the total number of subchannels and is equal to the product the number of filterbanks N and the number of subchannels per filterbank C , that is $S = N \cdot C$.

In addition, by conjugating (7.55), we make the terms consistent with (7.35) where the unknown synthesis filters are a column vector, and hence we have

$$TPR_{\text{compute}} = \dot{\mathbf{F}}^*_{\left(\frac{N_f}{2}+1\right)S \times \left(\frac{N_f}{2}+1\right)D} \mathbf{G}^*_{1 \times \left(\frac{N_f}{2}+1\right)S} = D \cdot \dot{\mathbf{E}}^*_{0,1 \times \left(\frac{N_f}{2}+1\right)D} \quad (7.60)$$

7.5 Optimization Problem

We can now combine the results of the previous sections to give an optimization problem that is computationally tractable.

$$\begin{aligned} & \underset{G_{\text{compute}}}{\text{minimize}} && J_{\text{compute}}(G_{\text{compute}}) = J_{I,p,\text{compute}}(G_{\text{compute}}) + \lambda J_{S,\text{compute}}(G_{\text{compute}}) \\ & \text{subject to} && TPR_{\text{compute}} \end{aligned} \quad (7.61)$$

where G_{compute} is given by (7.8), $J_{I,p,\text{compute}}$ is given by (7.40), $J_{S,\text{compute}}$ is given by (7.44), and TPR_{compute} is given by (7.56).

We note again that λ is a user chosen constant that denotes the severity of the sparse sub-channel penalty and the problem (7.61) is convex.

Chapter 8: Numerical Results When Optimizing Over Synthesis Filters

In this chapter, we detail our numerical simulations. Our simulations rely heavily on Michael Grant's CVX Toolbox, a package for specifying and solving convex programs [41], and its integration with the professional solver MOSEK.

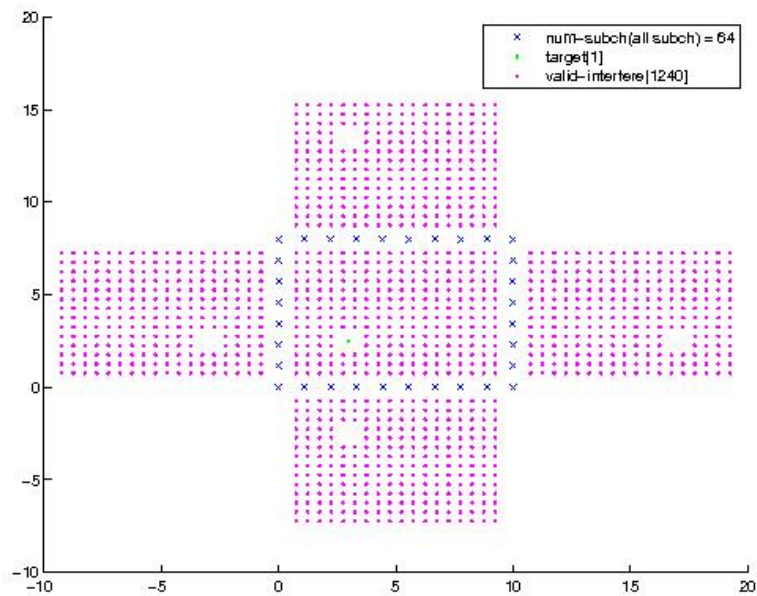


Fig. 8.1. Room environment.

The rectangular room shown in Fig. 8.1 has dimensions 10 meters by 8 meters and 32 microphones are placed uniformly along its 4 walls. The lower left hand corner of the room is

defined as the origin, coordinate (0,0), and our target source is located at (3, 2.5). There are 1240 virtual interferences in our room. Four-fifths of the interferences lie outside the room and model reflections off the four walls. We model our room environment using a large number of virtual interferences since we do not apriori where the actual sources lie.

Each microphone is processed by a two channel filterbank, with each channel being decimated and then later upsampled by a factor of 2. Hence we have $32 * 2 = 64$ subchannels. We fix the two analysis filters of each of the filterbanks to be Haar filters. We wish to design the synthesis filters that will minimize the maximum gain of the interferences and hence let $p = \infty$. In addition, we wish to perfectly reconstruct the target source and use only 20 out of the possible 64 subchannels. In other words, only 20 out of the 64 synthesis filters are allowed to be active and have non-zero frequency responses. We note that 64 choose 20 is approximately $2 * 10^{16}$, and hence we have an overwhelming number of subsets of subchannels to choose from.

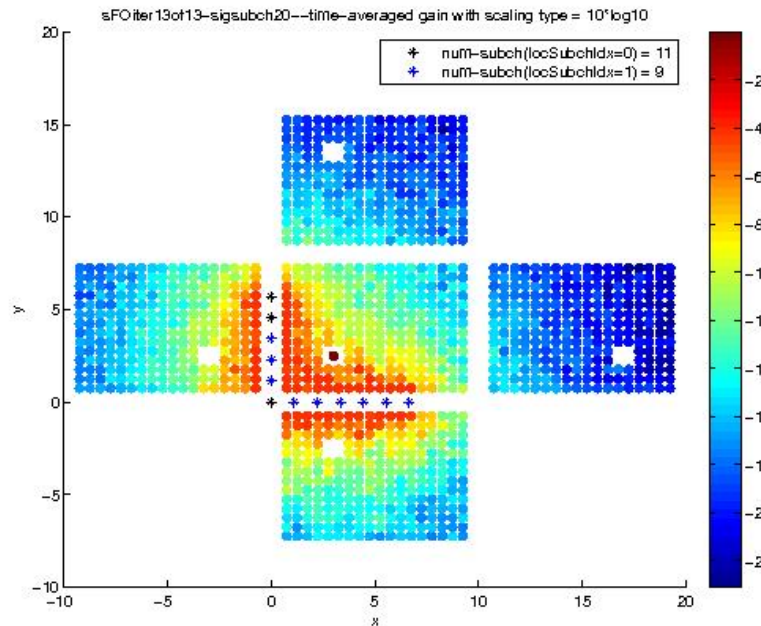


Fig. 8.2. Performance on optimization grid consisting of 1240 interferences in db.

We run our optimization problem iteratively, tuning the parameter λ at each iteration until

we find exactly 20 active subchannels. λ , a non-negative scalar, is found through a bisection algorithm since as λ increases, the number of active subchannel decreases, and similarly as λ decreases, the number of active subchannels increase. A subchannel is considered inactive if the maximum magnitude of its synthesis filter is less than $\frac{1}{1000}$ of the maximum magnitude of all the synthesis filters. Fig. 8.3 shows the maximum magnitude of each subchannel’s synthesis filter after our iterations completed. Notice that 20 of the 64 subchannel have non-trivial synthesis filters .

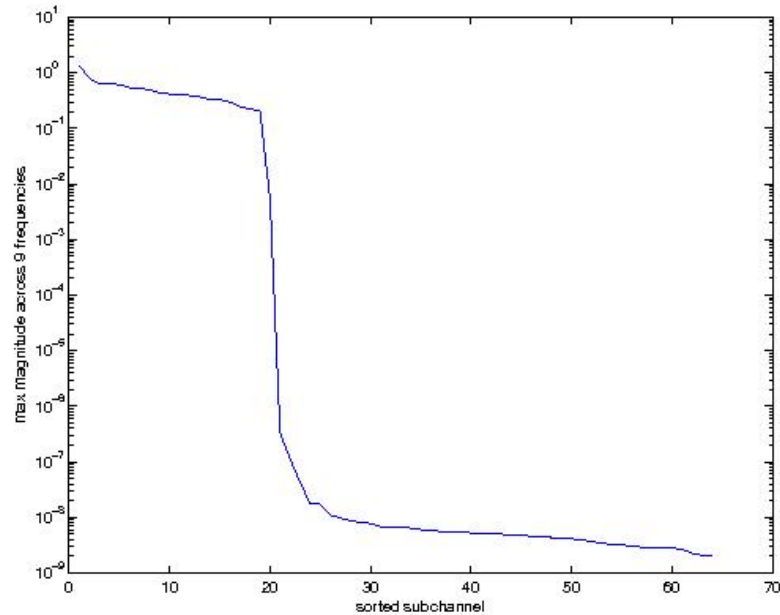


Fig. 8.3. Maximum magnitude of each subchannel’s synthesis filter for the λ value that produced the desired number of subchannels.

Our optimization problem finds a sparse number of synthesis filters and thus a sparse number of active subchannels. However, the frequency response of the inactive synthesis filters are not exactly zero.

Hence, we run our optimization routine one more time, optimizing the synthesis filters of only the previously discovered active subchannels and setting our sparsity parameters λ to zero.

This is known as the *debiasing* step. In some sense, this step redistributes the “crumbs” of energy in the inactive subchannels’ synthesis filters to the active subchannels’ synthesis filters. Fig. 8.4 shows each subchannel’s maximum synthesis filter magnitude after this debiasing step.

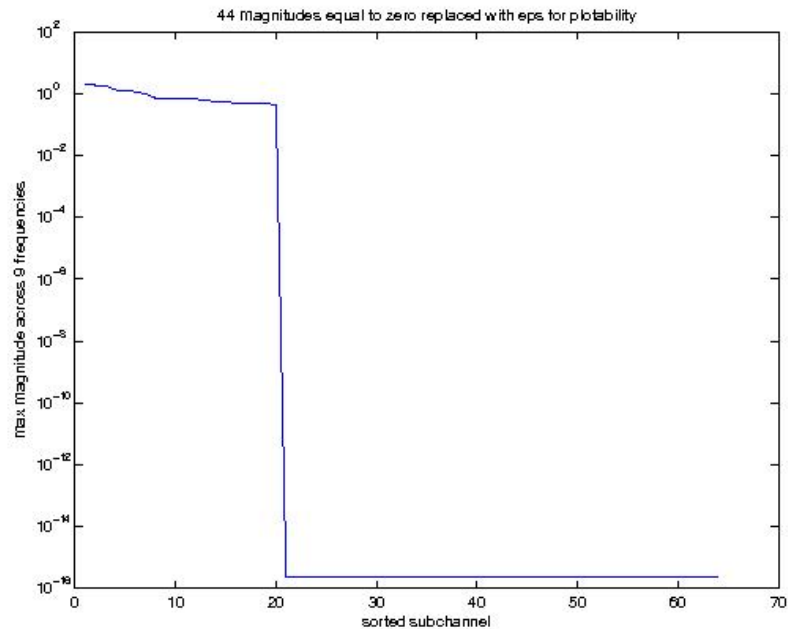


Fig. 8.4. Maximum magnitude of each subchannel’s synthesis filter after debiasing.

Our optimization routine returns a setup that uses slightly more low-frequency subchannels than it does high-frequency subchannels. Furthermore, the setup even occasionally uses only a single subchannel of a filterbank and not the other subchannel. See Fig. 8.7, 8.8, 8.9, and 8.10 for sample synthesis filter frequency responses of the 64 subchannels, 20 of which are active. Fig. 8.5 shows the locations of the active low-frequency subchannels, and Fig. 8.6 shows the locations of the active high-frequency subchannels. We optimized over 16 discrete frequencies, 9 of which are unique since we assume all our filter taps are real and therefore frequency responses are conjugate symmetric. Notice that the total number of active subchannels sums to the desired number of active subchannels, 20.

Fig. 8.2 shows how the 20 synthesis filters we found performed on the set of sources we

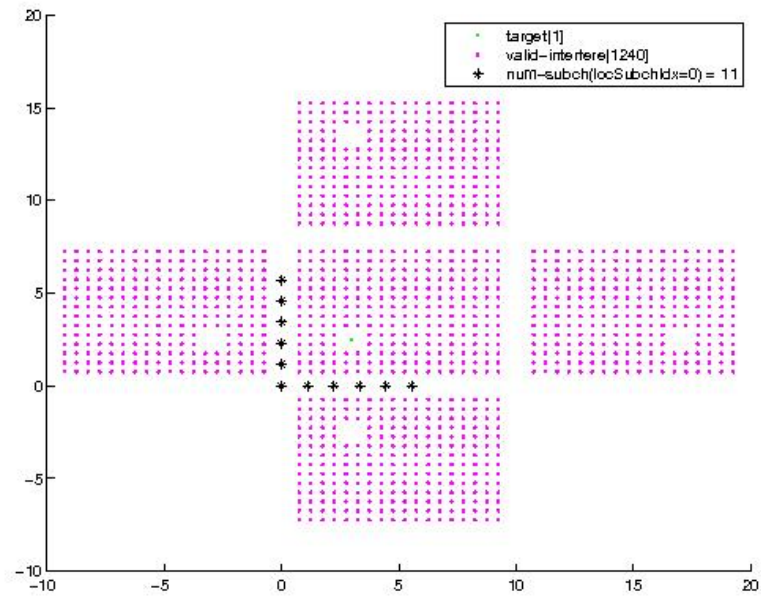


Fig. 8.5. Locations of active low frequency subchannels.

optimized over. The target gain is 0 db because an optimization constraint was target perfect reconstruction. The worst interference gain is -8.27 db. While this result is impressive, a more realistic measure of performance is to test our system's filters in a room with a denser set of interferences, as in Fig. 8.11 where the worst interference gain is 1.84 db. Not surprisingly, the performance is worst near the microphones. However, the target gain is again 0 db, and the gain map decays very smoothly.

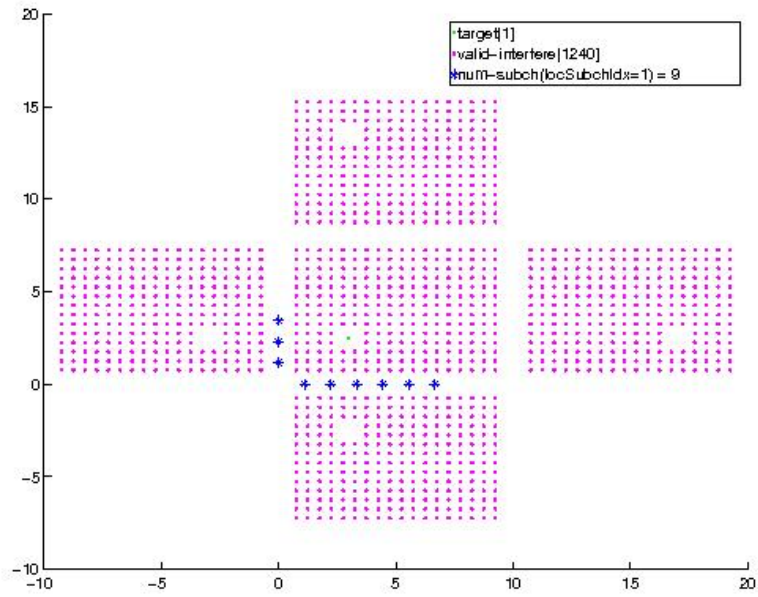


Fig. 8.6. Locations of active high frequency subchannels.

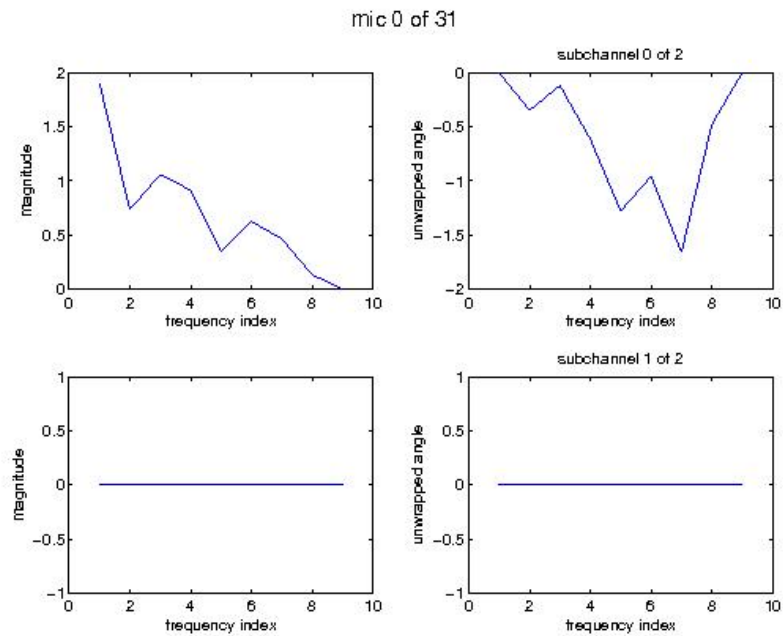


Fig. 8.7. The synthesis filters frequency responses of a filterbank whose low frequency subchannel is only active.

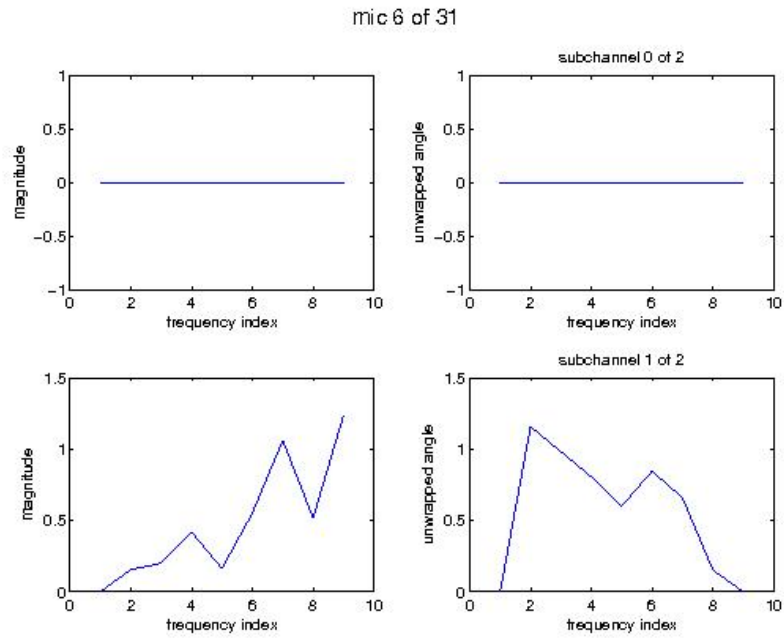


Fig. 8.8. The synthesis filters frequency responses of a filterbank whose high frequency subchannel is only active.

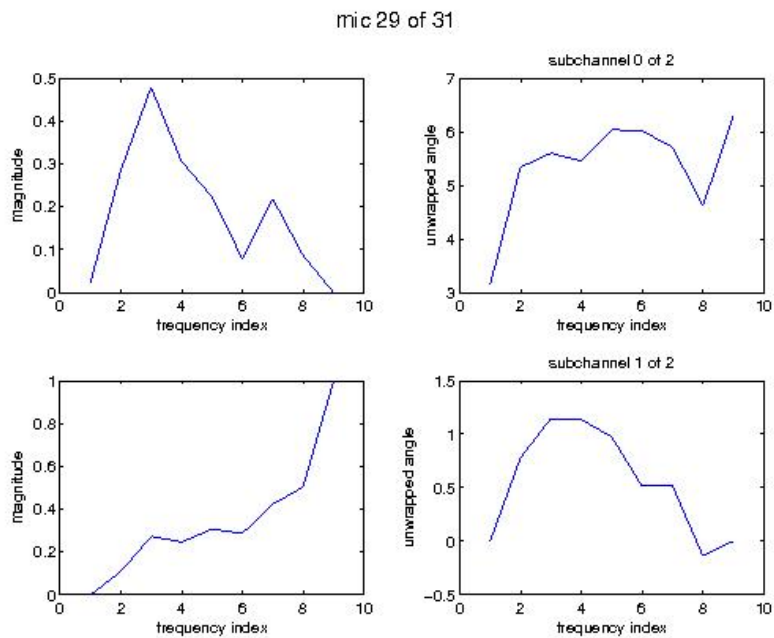


Fig. 8.9. The synthesis filters frequency responses of a filterbank whose subchannels are both active.

mic 23 of 31

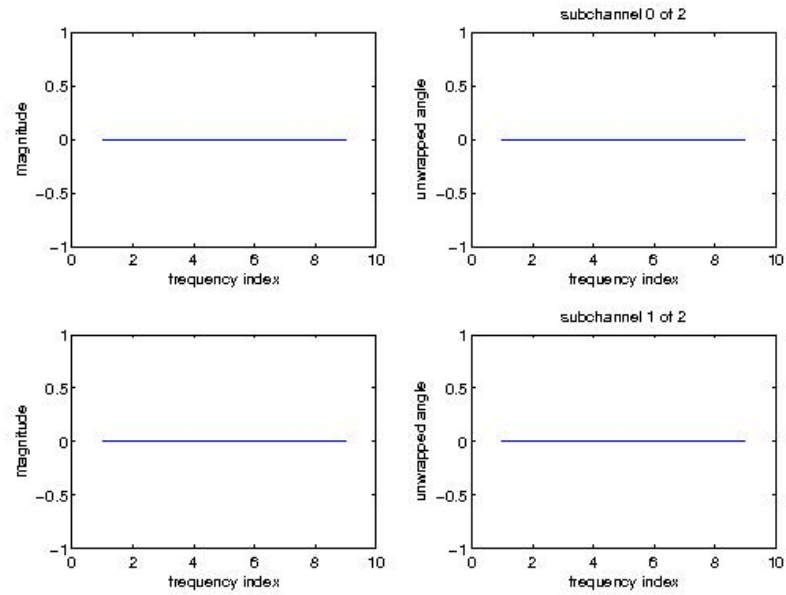


Fig. 8.10. The synthesis filters frequency responses of a filterbank whose subchannels are both inactive.

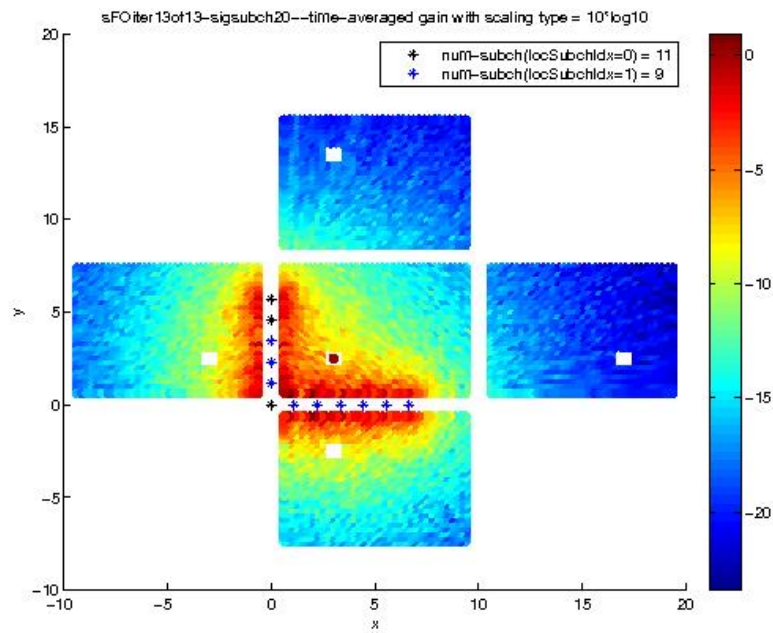


Fig. 8.11. Performance on evaluation grid consisting of 4960 interferences in db .

Chapter A: Properties of Γ_r

In the following, we assume that N_f , the number of discretized frequencies, is a multiple of D , the downsampling factor, that is $N_f = M \cdot D$ with M a positive integer. We now show that $\Gamma_r(s_1, s_2, f)$ is M -periodic in f , that is

$$\Gamma_r(s_1, s_2, f + M) = \Gamma_r(s_1, s_2, f) \quad (\text{A.1})$$

Proof. To simplify notation, we let $\phi_{s_1, s_2, r}(e^{j2\pi(\frac{f-Md_2}{N_f})}) = \dot{F}_{s_1, r}(e^{j2\pi(\frac{f-Md_2}{N_f})})\overline{F}_{s_2, r}(e^{j2\pi(\frac{f-Md_2}{N_f})})$.

Also let $d'_2 = d_2 - 1$. Then we have

$$\begin{aligned}
\Gamma_r(s_1, s_2, f + M) &= \sum_{d_2=0}^{D-1} \phi_{s_1, s_2, r} \left(e^{j2\pi \left(\frac{f-M(d_2-1)}{N_f} \right)} \right) \\
&= \sum_{d'_2=-1}^{D-2} \phi_{s_1, s_2, r} \left(e^{j2\pi \left(\frac{f-Md'_2}{N_f} \right)} \right) \\
&= \phi_{s_1, s_2, r} \left(e^{j2\pi \left(\frac{f+M}{N_f} \right)} \right) + \sum_{d'_2=0}^{D-2} \phi_{s_1, s_2, r} \left(e^{j2\pi \left(\frac{f-Md'_2}{N_f} \right)} \right) \\
&= \phi_{s_1, s_2, r} \left(e^{j2\pi \left(\frac{f+M}{N_f} \right)} e^{j2\pi \left(\frac{-MD-M+M}{N_f} \right)} \right) + \sum_{d'_2=0}^{D-2} \phi_{s_1, s_2, r} \left(e^{j2\pi \left(\frac{f-Md'_2}{N_f} \right)} \right) \\
&= \phi_{s_1, s_2, r} \left(e^{j2\pi \left(\frac{f-M(D-1)}{N_f} \right)} \right) + \sum_{d'_2=0}^{D-2} \phi_{s_1, s_2, r} \left(e^{j2\pi \left(\frac{f-Md'_2}{N_f} \right)} \right) \\
&= \sum_{d'_2=0}^{D-1} \phi_{s_1, s_2, r} \left(e^{j2\pi \left(\frac{f-Md'_2}{N_f} \right)} \right) \\
&= \Gamma_r(s_1, s_2, f)
\end{aligned}$$

where the fourth line uses the fact that the z -transform is 2π periodic in w for $z = e^{jw}$. \square

We now show that the matrix $\mathbf{\Gamma}_{r, S \times S}(f)$ is positive semi-definite, that is

$$\mathbf{x}^* \mathbf{\Gamma}_{r, S \times S}(f) \mathbf{x} \geq 0 \tag{A.2}$$

for any non-zero column vector \mathbf{x} in \mathbf{C}^S .

Proof. From (7.31), we have

$$\begin{aligned}
& \mathbf{x}^* \mathbf{\Gamma}_{r,S \times S}(f) \mathbf{x} \\
&= \sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} \bar{x}_{s_1} x_{s_2} \sum_{d_2=0}^{D-1} \dot{F}_{s_1,r}(e^{j2\pi(\frac{f-Md_2}{N_f})}) \overline{\dot{F}}_{s_2,r}(e^{j2\pi(\frac{f-Md_2}{N_f})}) \\
&= \sum_{d_2=0}^{D-1} \sum_{s_1=0}^{S-1} \bar{x}_{s_1} \dot{F}_{s_1,r}(e^{j2\pi(\frac{f-Md_2}{N_f})}) \sum_{s_2=0}^{S-1} x_{s_2} \overline{\dot{F}}_{s_2,r}(e^{j2\pi(\frac{f-Md_2}{N_f})}) \\
&= \sum_{d_2=0}^{D-1} \left| \sum_{s=0}^{S-1} \bar{x}_s \dot{F}_{s,r}(e^{j2\pi(\frac{f-Md_2}{N_f})}) \right|^2 \\
&\geq 0
\end{aligned}$$

where x_k is the k -th entry of the vector \mathbf{x} . □

Chapter B: Parseval Relationship For Energy of Filters

The following equation gives a relationship between the l_2 energy of a filter g 's tap coefficients and the l_2 energy of its transfer function.

$$\sum_l |g[l]|^2 = \frac{1}{2\pi} |G(w)|^2 dw \tag{B.1}$$

Proof. The following inverse Discrete Time Fourier Transform relates the tap coefficients of a filter g to its frequency response G

$$g[l] = \frac{1}{2\pi} \int_0^{2\pi} G(w) e^{jwl} dw \tag{B.2}$$

The following Poisson summation formula gives us an useful “time-frequency” relationship.

$$\sum_n e^{2\pi jnx} = \sum_k \delta(x - k) \quad (\text{B.3})$$

We note that the delta distribution $\delta(x)$ has the following scaling property:

$$|a|\delta(ax) = \delta(x) \quad (\text{B.4})$$

We can now relate the $l2$ energy of the filter’s tap coefficients to the $l2$ energy of the filter’s transfer function.

$$\begin{aligned} \sum_l |g[l]|^2 &= \sum_l \left| \frac{1}{2\pi} \int_0^{2\pi} G(w) e^{jlw} dw \right|^2 \\ &= \sum_l \frac{1}{4\pi^2} \int_0^{2\pi} G(w_1) e^{jlw_1} dw_1 \cdot \int_0^{2\pi} \overline{G(w_2)} e^{-jlw_2} dw_2 \\ &= \frac{1}{4\pi^2} \int_0^{2\pi} \int_0^{2\pi} G(w_1) \overline{G(w_2)} \underbrace{\sum_l e^{2\pi jl \frac{w_1 - w_2}{2\pi}}}_{\sum_n \delta(\frac{w_1 - w_2}{2\pi} - n)} dw_1 dw_2 \\ &= \frac{1}{2\pi} \sum_n \int_0^{2\pi} \int_0^{2\pi} G(w_1) \overline{G(w_2)} \delta(w_1 - w_2 - 2\pi n) dw_1 dw_2 \\ &= \frac{1}{2\pi} |G(w)|^2 dw \end{aligned}$$

Note that $w_1 - w_2 < 2\pi$ since both w_1 ’s and w_2 ’s limits are from 0 to 2π and hence the sum over n produces non-zero $\delta(w_1 - w_2 - 2\pi n)$ only when $n = 0$. (We ignore the difference of exactly 2π because it has measure 0.)

□

B.1 References

- [1] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.
- [2] E.J. Candes and Y. Plan. Near-ideal model selection by ℓ_1 minimization. *Annals of Statistics*, 37(5A):2145–2177, 2009.
- [3] S. Kirkpatrick and MP Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [4] H. Lebreit and S. Boyd. Antenna array pattern synthesis via convex optimization. *Signal Processing, IEEE Transactions on*, 45(3):526–532, 1997.
- [5] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebreit. Applications of second-order cone programming. *Linear algebra and its applications*, 284(1):193–228, 1998.
- [6] C. Ling, S. Wee, C. Wei, and Y. Zhu Liang. Linear sparse array synthesis via convex optimization. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 4233–4236, 2010.
- [7] R.A. Haubrich. Array design. *Bulletin of the Seismological Society of America*, 58(3):977–991, 1968.
- [8] S. Lang, C. Duckworth, and I. McClellan. Array design for mem and mlm array processing. In *Proceedings of the IEEE*, pages 145–148, Atlanta, GA, USA, 1981.
- [9] P. Pal and P. P. Vaidyanathan. Nested arrays: A novel approach to array processing with enhanced degrees of freedom. *Signal Processing, IEEE Transactions on*, 58(8):4167–4181, 2010.
- [10] D. B. Ward and M. S. Brandstein. Grid-based beamformer design for room-environment microphone arrays. In *Applications of Signal Processing to Audio and Acoustics, 1999 IEEE Workshop on*, pages 23–26, 1999.
- [11] M. S. Brandstein and D. B. Ward. Cell-based beamforming (ce-babe) for speech acquisition with microphone arrays. *Speech and Audio Processing, IEEE Transactions on*, 8(6):738–743, 2000.
- [12] J.B. Allen and D.A. Berkley. Image method for efficiently simulating small-room acoustics. *J. Acoust. Soc. Am*, 65(4):943–950, 1979.
- [13] R.G. Baraniuk, E. Candes, R. Nowak, and M. Vetterli. Compressive sampling. *IEEE Signal Processing Magazine*, 25(2):12–13, 2008.
- [14] E. J. Candes and M. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.
- [15] E.J. Cands and Y. Plan. Near-ideal model selection by ℓ_1 minimization. *The Annals of Statistics*, 37(5A):2145–2177, 2009.
- [16] Emmanuel J Candes. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9):589–592, 2008.

- [17] Bernard D Steinberg. The peak sidelobe of the phased array having randomly located elements. *Antennas and Propagation, IEEE Transactions on*, 20(2):129–136, 1972.
- [18] Richard M Leahy and Brian D Jeffs. On the design of maximally sparse beamforming arrays. *Antennas and Propagation, IEEE Transactions on*, 39(8):1178–1187, 1991.
- [19] Sverre Holm, Bjrnar Elgetun, and Geir Dahl. Properties of the beam pattern of weight-and layout-optimized sparse arrays. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 44(5):983–991, 1997.
- [20] Thomas Baran, Dennis Wei, and Alan V Oppenheim. Linear programming algorithms for sparse filter design. *Signal Processing, IEEE Transactions on*, 58(3):1605–1617, 2010.
- [21] David G Leeper. Isophoric arrays-massively thinned phased arrays with well-controlled sidelobes. *Antennas and Propagation, IEEE Transactions on*, 47(12):1825–1835, 1999.
- [22] Giacomo Oliveri and Andrea Massa. Bayesian compressive sampling for pattern synthesis with maximally sparse non-uniform linear arrays. *Antennas and Propagation, IEEE Transactions on*, 59(2):467–481, 2011.
- [23] P.P. Vaidyanathan. *Multirate Systems And Filter Banks*. Dorling Kindersley, 1993.
- [24] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.
- [25] Shuichi Ohno and Hideaki Sakai. Optimization of filter banks using cyclostationary spectral analysis. *Signal Processing, IEEE Transactions on*, 44(11):2718–2725, 1996.
- [26] Andrea Trucco and Vittorio Murino. Stochastic optimization of linear sparse arrays. *Oceanic Engineering, IEEE Journal of*, 24(3):291–299, 1999.
- [27] Siddharth Joshi and Stephen Boyd. Sensor selection via convex optimization. *Signal Processing, IEEE Transactions on*, 57(2):451–462, 2009.
- [28] Dragana Bajovic, Bruno Sinopoli, and Joo Xavier. Sensor selection for event detection in wireless sensor networks. *Signal Processing, IEEE Transactions on*, 59(10):4938–4953, 2011.
- [29] Ling Cen, Wee Ser, Zhu Liang Yu, Susanto Rahardja, and Wei Cen. Linear sparse array synthesis with minimum number of sensors. *Antennas and Propagation, IEEE Transactions on*, 58(3):720–726, 2010.
- [30] Venkat Roy, Sundeep Prabhakar Chepuri, and Geert Leus. Sparsity-enforcing sensor selection for doa estimation. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on*, pages 340–343. IEEE.
- [31] Jian Wang, Wei-Xing Sheng, Yu-Bing Han, and Xiao-Feng Ma. Adaptive beamforming with compressed sensing for sparse receiving array. *Aerospace and Electronic Systems, IEEE Transactions on*, 50(2):823–833, 2014.
- [32] Siew Eng Nai, Wee Ser, Zhu Liang Yu, and Huawei Chen. Beam pattern synthesis for linear and planar arrays with antenna selection by convex optimization. *Antennas and Propagation, IEEE Transactions on*, 58(12):3923–3930, 2010.

- [33] Zhi Guo Feng, Ka Fai Cedric Yiu, and Sven E Nordholm. Placement design of microphone arrays in near-field broadband beamformers. *Signal Processing, IEEE Transactions on*, 60(3):1195–1204, 2012.
- [34] J. L. Araque Quijano, M. Righero, and G. Vecchi. Sparse 2-d array placement for arbitrary pattern mask and with excitation constraints: A simple deterministic approach. *Antennas and Propagation, IEEE Transactions on*, 62(4):1652–1662, 2014. 0018-926X.
- [35] James D Johnston. A filter family designed for use in quadrature mirror filter banks. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'80.*, volume 5, pages 291–294. IEEE.
- [36] Tongwen Chen and BA Francis. Design of multirate filter banks by h optimization. *IEEE Transactions on Signal Processing*, 43:2822–2830, 1995.
- [37] Eric Viscito and Jan P Allebach. The analysis and design of multidimensional fir perfect reconstruction filter banks for arbitrary sampling lattices. *Circuits and Systems, IEEE Transactions on*, 38(1):29–41, 1991.
- [38] Truong Q Nguyen. Digital filter bank design quadratic-constrained formulation. *Signal Processing, IEEE Transactions on*, 43(9):2103–2108, 1995.
- [39] Ha Hoang Kha, Hoang Duong Tuan, and Truong Q Nguyen. Efficient design of cosine-modulated filter banks via convex optimization. *Signal Processing, IEEE Transactions on*, 57(3):966–976, 2009.
- [40] Timothy Davidson. Enriching the art of fir filter design via convex optimization. *Signal Processing Magazine, IEEE*, 27(3):89–101, 2010.
- [41] Michael Grant and Stephen Boyd. Cvx: Matlab software for disciplined convex programming, version 2.1, March 2014.