Title

Effort compression: signal-dependent gain management for the pressure matching beamforming method

Author
Patros, Elliot M

Publication Date
2019

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Effort compression: signal-dependent gain management for the pressure matching beamforming method

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Music

by

Elliot M. Patros

Committee in charge:

Professor Miller Puckette, Chair
Professor Anthony Burr
Professor Sarah Creel
Professor Tamara Smyth
Professor Shahrokh Yadegari

2019

The dissertation of Elliot M. Patros is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2019

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

CTC:	Cross-talk cancellation
DAW:	Digital audio workstation
DFT:	Discrete Fourier transform
DOA:	Direction of arrival
EM:	Effort management
FIR:	Finite impulse response
HRTF:	Head related transfer function
iDFT:	Inverse discrete Fourier transform
ILD:	Interaural level difference
ITD:	Interaural time difference
LTI:	Linear time-invariant
PMM:	Pressure matching method
RMS:	Root mean square
SFC:	Sound field control
SVD:	Singular value decomposition
ULA:	Uniform linear array

LIST OF SYMBOLS

e	≈ 2.71828	Euler's number
i	$\sqrt{-1}$	Imaginary number
ω	$2\pi f$	Radial frequency $(\mathrm{rad}\cdot\mathrm{s}^{-1})$ or $(2\pi\,\mathrm{Hz})$
c	343	Speed of sound propagation $\left(\mathrm{m}\cdot\mathrm{s}^{-1}\right)$
k	$\omega\cdot c^{-1}$	Wave number
N	$1 \leq N < \infty$	Number of loudspeakers
n	$1 \leq n \leq N$	Index of N
M	$1 \leq M < \infty$	Number of ears and input channels (stereo when M=2)
m	$1 \leq m \leq M$	Index of M
L, R	1, 2	Left and right ears channel indices when $M = 2$
$\boldsymbol{x}$	$\in \mathbb{C}^N$	Spatial filter coefficients
$\boldsymbol{y}$	$\in \mathbb{C}^M$	Control point responses
A	$\in \mathbb{C}^{M\times N}$	Propagation matrix
I	$\in \mathbb{R}^{M\times M}$	The Identity matrix
X	$\in \mathbb{C}^{N\times M}$	Spatial filter convolution matrix
$\boldsymbol{s}$	$\in \mathbb{C}^M$	Input audio signal (one bin of a DFT block)
$\boldsymbol{w}$	$\in \mathbb{C}^N$	Array output (combination of filters and input)
r	$\in \mathbb{R}$	Propagation distance (from a source to a reciever)
$G(r)$	$\in \mathbb{C}$	The Green's function of the Helmholtz equation
$E()$	$0 \leq E() \leq \infty$	Error (of a vector)
$K()$	$0 \leq K() \leq \infty$	Effort (of a vector)
β	$0 \leq \beta \leq \infty$	Tikhonov regularization parameter
$J()$	$\in \mathbb{R}$	Tikhonov regularization cost function
Q, Λ		Eigenvectors and eigenvalues $\left(A = Q\Lambda Q^{-1}\right)$.
V, Σ, U		Singular value decompisition terms $\left(A = V\Sigma U^H\right)$.

LIST OF FIGURES

ACKNOWLEDGEMENTS

There are a lot of people to thank, who contributed either directly to the making of this dissertation, or to my experience as a graduate student.

First, to my committee:

Miller Puckette, thank you for your ability to see N+1 sides of any issue. Our conversations these years have helped me develop the ability to combine creative ideas with technical ones.

Tamara Smyth, thank you for your advice on my writing, your help debugging code, for introducing me to computer music math, and for encouraging me to work harder.

Shahrokh Yadegari, thank you for your guidance in the spatialization laboratory (UCSD's SpatLab), and for pushing technological development from a musical point of view.

Katharina Rosenberger, thank you for your work finding creative applications for array tech and beamforming. It's a deceptively challenging project, and our meetings were an important part of the motivation for this work.

Sarah Creel, thank you for supporting my research, while at the same time, asking hard questions. Also, thank you for sharing your expertise on the perceptual aspects of this project, especially about the listener's experience, which has absolutely reshaped my thought process as a researcher.

Anthony Burr, thank you for your ability to quickly and intuitively learn both the background and novel aspects of this research project. Your experience with studio production techniques helped shed light on an underrepresented aspect of this research field.

Second, to my colleagues:

Tara Afghah, thank you for collaborating with me on so many projects. Sharing ideas together has helped me become a better researcher. Thank you also for weathering the challenges of navigating academia and industry with me.

Peter Otto, thank you for all your work supporting the greater research project of acoustic beamforming and binaural audio. None of this would be possible without it.

Finally, to my family.

Thank you Clay, Mom, and Dad, for your unconditional support, and encouragement.

Caroline Miller. Thank you for everything. Your hard work, intelligence, your sense of humor, and your compassion are an inspiration to me. I could not ask for a better partner.

VITA

2012	B.F.A. in Music Composition, University of Wisconsin, Milwaukee
2013-2014	ICAM Studios Research Assistant, University of California, San Diego
2016	Contract Audio Programmer, Magic Leap
2014-2018	Graduate Student Researcher, Sonic Arts at CalIT2
2018	Contract Audio Programmer, Dreamcraft Attractions
2018-Present	Research consultant at Dysonics Inc.
2019	Ph.D. in Music, University of California, San Diego

PUBLICATIONS

Tahereh Afghagh, Elliot Patros, and Miller Puckette, "A pseudoinverse technique for the pressure-matching beamforming method", *Audio Engineering Society*, 145, 2018

Elliot Patros, Tahereh Afghagh, and Miller Puckette, "Acoustic Beamforming", *Int. Pat. No. WO 2019/118521* 2019

Tahereh Afghagh, Elliot Patros, and Miller Puckette, "The Physical Evaluation of the Efficiency of an Enhanced Pressure-Matching Beamforming Method Using Eigen Decomposition Pseudoinverse Mathematical Approach", *Audio Engineering Society*, IIA, 2019

Elliot Patros, David Romblom, "Optimal Crosstalk Cancellation Filter Sets Generated by Using an Obstructed Field Model and Methods of Use", *Patent No. PCT/US19/62381 Pending*, 2019

Perry Teevens, Robert J.E. Dalton Jr., David E. Romblom, Peter G. Otto, Elliot M. Patros,"Apparatus and Method to Provide Situational Awareness Using Positional Sensors and Virtual Acoustic Modeling", *Patent No. PCT/US19/62378 Pending*, 2019

Elliot Patros, Tahereh Afghagh, Peter Otto, "Analysis and reproduction of stereo sound fields with compact uniform linear arrays", *Acoustical Society of America*, San Diego Convention, 2019

ABSTRACT OF THE DISSERTATION

Effort compression: signal-dependent gain management for the pressure matching beamforming method

by

Elliot M. Patros

Doctor of Philosophy in Music

University of California San Diego, 2019

Professor Miller Puckette, Chair

This work is focused on the trade-off between sound quality and loudspeaker gain that occurs when using the pressure matching beamforming method to generate stereo crosstalk cancellation with uniform linear arrays. An approach to gain management is proposed that leverages characteristics of subclasses of stereo audio signals (e.g. binaural) to partially overcome this tradeoff by improving both loudspeaker gain and sound quality together. The novelty of the proposed approach is based on an observation that loudspeaker gain management, which is normally accomplished by manipulating beamforming filters, can also be accomplished by manipulating input audio signals. It is shown that decoupling gain management from beamforming filters allows it to be signal-dependent. Also, that signal-dependent gain management gives beamforming filter designers more meaningful methods for controlling both the type and amount of sound quality that is sacrificed to manage gain. A general approach to building a signal-dependent gain management algorithm is suggested. Some specific implementations are given as examples. Finally, a discussion is presented about how signal-dependent gain management potentially establishes a future research project where perceived sound quality, as opposed to numerical sound quality, is maximized.

Prependix

Notation

Throughout this work, the following notational conventions are used. Scalars are lower case, non-bold: x. Vectors are lower case, bold: $\boldsymbol{x}$. Matrices are upper case, non-bold: X. Continuous functions are also upper case, non-bold. Continuous functions are distinguishable from matrices because they are always notated with respect to their variables, which are surrounded by parentheses: $X(n)$. Discrete indices of vectors and matrices are subscripts: $\boldsymbol{x}_n$. The lengths of indexed structures are upper-case, and use the same letter as the associated index: $\boldsymbol{x}_n$ has N elements.

Other notational conventions about indexing vectors and matrices in this work are described as follows: indexing a vector results in a number (e.g. $\boldsymbol{x}_n$ is a the n^{th} number in $\boldsymbol{x}$); indexing a matrix once results in a vector (i.e. X_m is the m^{th} column vector, or, X_n is the n^{th} row vector in $X \in \mathbb{C}^{M \times N}$); and, indexing a matrix twice results in a number (e.g. X_{mn} is the number at row m and column n). The hat overtext $\hat{}$ indicates that some variable should be equal to the variable without the overtext. For example, ideally $\hat{\boldsymbol{x}} = \boldsymbol{x}$, and it's bad when $\hat{\boldsymbol{x}} \neq \boldsymbol{x}$.

Finally, the following superscripts are used to indicate common matrix operations.

X^{-1}	Matrix inverse
X^{+}	Pseudoinverse
X^{*}	Complex conjugate
X^{T}	Transpose
X^{H}	Hermitian (complex conjugate) transpose

Assumptions about the configuration

This work is about using acoustic beamforming and loudspeaker arrays to play stereo audio with crosstalk cancellation. It is a difficult problem because crosstalk cancellation needs too much loudspeaker gain to perform well, especially at low frequencies. A novel approach to managing loudspeaker gain is proposed. However, the results shown are based on several assumptions about physical and digital configurations. See Figure 0.1 for an illustration.

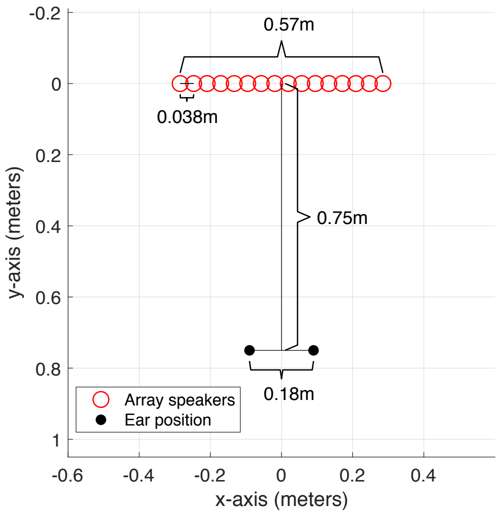
First, that the target listener is one person who is

- located on axis. That is, who's line of sight is perpendicular to the line intersecting each loudspeaker;
- located between 0.5 and 1.5 meters from the array;
- listening to stereo or binaural (i.e. 2-channel) audio with localization cues.

Second, loudspeakers are arranged in a ULA (Uniform Linear Array) with

- many loudspeakers ($\gg 2$);
- discrete loudspeaker spacing (i.e. $\Delta\boldsymbol{x}_{\text{pos}} > 0$);
- close loudspeaker spacing $\Delta\boldsymbol{x}_{\text{pos}} < \approx 50\,\text{mm}$.

Finally, several filter design parameters are constant. They are listed here to improve the reproducibility of results shown throughout this work.



Speed of sound	c	$= 343\,\mathrm{m{\cdot}s^{-1}}$
Sampling rate	f_s	$= 48\,\mathrm{kHz}$
DFT blocks have	bins	$= 1024$
Speaker count	M	$= 16$
Inter-ear distance	$\Delta\boldsymbol{y}_{\text{pos}}$	$= 0.18\,\mathrm{m}$
Inter-speaker distance	$\Delta\boldsymbol{x}_{\text{pos}}$	$= 0.038\,\mathrm{m}$
Listener distance		$= 0.75\,\mathrm{m}$

Figure 0.1: Description and illustration of the constant beamforming configuration.

Chapter 1

Introduction

By assigning spatial filters to discrete channels of a loudspeaker array, binaural audio can be produced at the ears of a nearby listener to create the sensation of 3D surround sound. However, acoustic limitations impose a difficult trade-off from the perspective of a spatial filter designer. On one hand, filters that maximize sound quality for a listener put too much strain, or effort, on the array. High array effort corresponds in practice to undesirable, nonlinear loudspeaker behaviors like distortion. On the other hand, filters that minimize effort do so by sacrificing sound quality. The branch of research that attempts to navigate this trade-off is called effort management (EM).

The primary contribution of this work is the introduction of a new approach to EM that leverages signal properties present in binaural audio to simultaneously improve both the effort and sound quality of its spatial filters. The novelty of the approach is based on an observation that EM, which is normally cast as a spatial filter transformation, can instead be cast as a transformation on input signals directly. That is, decoupling EM from filters allows it to be signal-dependent. Signal-dependent EM, called ‘effort compression’, is rationalized by showing that array effort is also signal-dependent, and, that binaural audio costs significantly less effort to spatially filter than other types of stereo audio. So, for example, a tangible benefit of effort compression is that the strength of EM can be automatically adjusted to an ideal value for a particular input signal and in real-time. This potentially allows a single filter set to work well with different types of audio without compromising performance.

1.1 Outline and contributions

The contributions of this work are organized as follows.

Chapter 1 discusses background concepts; first, about general methods of spatial filtering, called sound field control (SFC), and then, about a specific SFC method used throughout this work, called the pressure matching method (PMM). The intention is for this chapter to build intuition about this work's contributions, presented later.

Chapter 2 discusses the theory needed both to reinforce concepts mentioned in chapter 1, and which are also required to make observations and contributions about PMM in later chapters. The discussions in chapters 1 and 2 are most relevant to PMM applications that use single-channel input signals; and need to be reworked for use with multi-channel input signals.

Chapter 3 [†] establishes a theoretical framework for effort compression which, takes the form of a multi-channel signal flow graph for PMM, and, establishes a mathematical model of multi-channel PMM.

Chapter 4 [†] presents the concept and theory of effort compression along with several example implementations. The primary goal of this chapter, more so than to advocate for a particular EM implementation, is to propose a way of thinking about SFC in general that considers the effect of input signals on SFC metrics. The idea is to communicate that many (if not all) conventional EM techniques can be transformed into signal-dependent ones. And, that many more effort compression techniques are yet to be discovered.

Finally, chapter 5 [†] directly addresses an idea that's hinted at throughout the previous

[†] Represents a novel contribution introduced by this work.

discussions of effort compression. It is that the notion of effort compression, which includes the possibility of nonlinear EM, creates a mechanism by which filter designers can more meaningfully control the *kind* of error that's introduced when constraints on effort make error unavoidable. Ostensibly this perspective could contribute to a future discovery of EM strategies that reduce *perceived* error. At the time of writing, perceived error has not been explored as thoroughly by SFC literature as *numerical* error.

1.2 Sound field control

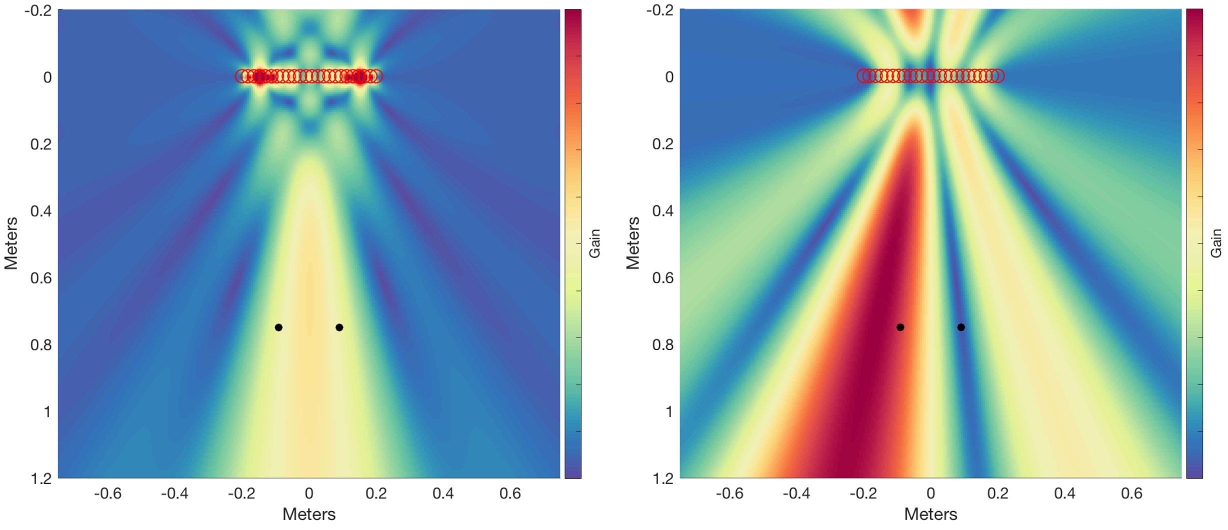
The term sound field control (SFC) describes any method[1] where a loudspeaker array is coaxed into producing user-specified acoustic responses in the local sound field. Though the SFC concept is completely general — any response can ostensibly be produced anywhere — the bulk of SFC research is specifically focused on producing broadband acoustic cancellation. This work is no exception. It is focused on a method for producing a particular type of cancellation, called cross-talk cancellation (CTC), in which sound transmitted to a listener is intended to mimic the experience of wearing headphones. CTC is able to mimic headphones by producing cancellation at one ear and not at the other. In contrast, another common type of SFC cancellation is called private audio, which produces cancellation everywhere except toward a target listener. See Figure 1.1 for an artist's representation of these two SFC implementations.

SFC problem statement

The following problem statement describes SFC methods generally. It is described here in order to introduce terminology used in this work, and, consists of three parts: an objective, a solution, and a model.

The SFC objective is defined in terms of a data structure called a 'control point', which is

[1] Notable SFC methods include PMM: [2], PMM extensions: [3–10], ACM: [11], ACM extensions: [12–16], planarity control: [17, 18], spectral division [19–21], extensions of crosstalk cancellation: [22–26]



(a) Private audio

(b) CTC

Figure 1.1: Examples of sound pressure for private and crosstalk cancellation sound field control at 2 kHz, as shown from a top-down view. Red circles are loudspeaker positions; black dots are control points placed at a listener's ears.

the combination of a discrete physical location and a user-specified acoustic response. Together, the acoustic responses and their associated physical locations are often thought of as monopole point-receivers, or virtual microphones [27]. An SFC user defines the objective by explaining where the control points should be, and what they should ideally record. Choi and Kim [11] propose the terms 'dark' point and 'bright' point to mean control points where cancellation is desired, or not, respectively. Both terms are used frequently throughout SFC literature, as well as in this work.

The second part of the SFC problem statement, the solution, takes the form of another data structure called 'spatial filters'. Spatial filters are conventional finite impulse response (FIR) filter coefficients assigned to discrete channels of a loudspeaker array. Together, the filter coefficients and their associated loudspeakers are often thought of as monopole point-sources. The SFC user, now more accurately called the filter designer, uses an SFC method to find what each point-source should play in order to most closely reproduce their specified control point responses. That is, when filters are rung simultaneously by an impulse, their output signals will propagate from the array, combine in the sound field, and create interference. So, filters should be designed in such a way that the resultant interference patterns cause the specified control point responses to occur at the specified locations. If one could see what sound pressure looks like near the array during SFC, beams would appear to originate from the center of the array, aimed at control points. For this reason, SFC is sometimes also called ‘beamforming’.

The third and final part of the general SFC problem statement is a ‘model’ of how the sound field is expected to behave. More specifically, the model describes acoustic transformations caused by propagation from each source to each receiver. Since source and receiver locations are all discrete, the sound field can be modeled by a matrix of transfer functions, called the propagation matrix. Each element of the matrix contains a transfer function between a receiver represented by the index of its row, and a source represented by the index of its column.

1.3 The pressure matching method

The pressure matching method (PMM) is an instance of SFC. It can be used to produce any type of cancellation, including both CTC and private audio. PMM distinguishes itself from other SFC methods in several ways. Those relevant to this work are its signal representation (how audio signals are notated), and by extension, its formulation (how a filter designer uses it).

PMM represents all types of signals, including filters, control point responses, transfer functions, and even audio streams in the complex frequency domain. The representation has several useful properties. For example, complex signals are only one transformation away from several other, more traditional audio representations; including a phase/magnitude tuple in the frequency domain via polar transform; or into a signal of time domain pressure via inverse Fourier transform.

Another useful property of the complex frequency domain representation is that multiple parallel signal transformations can be combined into a single transfer function by summation.

Likewise, multiple signal transformations in series can be combined into a single transfer function by multiplication. These are well-known properties of linear time-invariant (LTI) transformations [28]. They are mentioned here explicitly because they allow a system of transfer functions, like the sound field model, to be described by a linear system of equations. PMM leverages this representation to cast SFC as a linear optimization problem of the sort

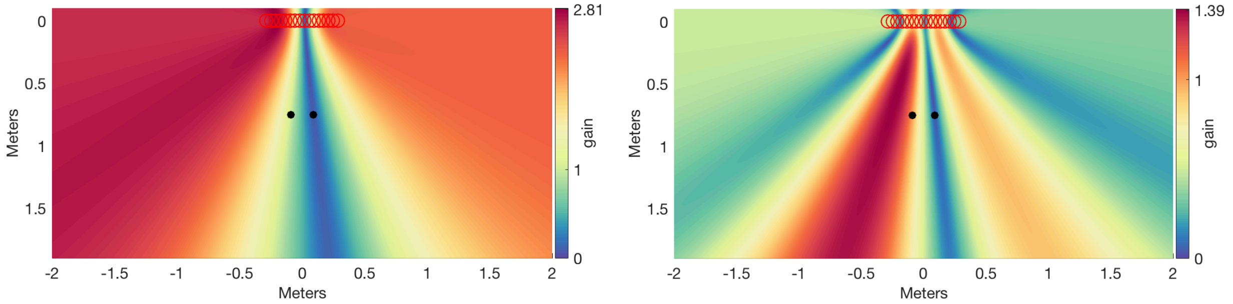
$$\boldsymbol{y} = A\boldsymbol{x}\,, \tag{1.1}$$

which is recognizable as a least squares problem [29]. Connecting language from the previous section about SFC to the variables in Eq. 1.1 helps give them meaning: elements of the vector $\boldsymbol{y}$ are control point responses, and elements of the vector $\boldsymbol{x}$ are spatial filter coefficients; unknown. The matrix *A* is the sound field model or propagation matrix.

The shape of the propagation matrix depends on the SFC objective, and strongly affects how filters are solved. For example, matrix *A* is ‘tall’ when there are more control points than loudspeakers. It’s typical for private audio problems to yield tall systems, since every point in the room could be associated with a desired response. Conversely, matrix *A* is ‘wide’ when there are more loudspeakers than control points. Wide systems arise in CTC problems that use arrays, because control point locations are defined to be only at the listener’s two ears. When several assumptions about the configuration of the array and listener are met[2], a wide propagation matrix can be guaranteed to have full row rank at non-zero frequencies. This type of system is underdetermined [29].

Underdetermined systems are not given much attention in literature, since they are trivial to solve with unconstrained least squares. And in fact, CTC filters *can* be generated with unconstrained least squares, at least in a numerical sense. Acoustically however, filters generated with unconstrained PMM have both perceptually-obvious and well-documented flaws [2–10].

[2] See Prependix



(a) 400 Hz.

(b) 1000 Hz.

Figure 1.2: Problems related to gain, wavelength, and the pressure gradient are easily seen from this perspective. The values 400 and 1000 Hz were chosen because they bookend the range of frequencies where crosstalk cancellation is both possible yet difficult to achieve using PMM. Red circles are loudspeaker positions; black dots are control points placed at a listener's ears.

1.4 Motivation

This work focuses on a specific problem faced by PMM when its objective is to generate CTC filters. In short, it is difficult to generate good-sounding CTC filters at low frequencies. The next few paragraphs will discuss why this is the case, what is so bad about naïvely-generated low-frequency filters, what has been done about it until now, and why it is still a difficult problem.

Why is it difficult to generate CTC filters at low frequencies?

Figure 1.2 shows (like Figure 1.1) monochromatic sound pressure over space. This time however, the figure shows sound field responses of filters which are exact solutions to the CTC problem as given by unconstrained PMM, and, includes low-frequency responses. Pay attention to size of each beam, or lobe, in terms of its radius as it emanates from the array. It can be seen that beam radius is related to what's called the pressure gradient, or, the rate of change of sound pressure over space. As frequency decreases, the pressure gradient decreases as well. Put another way, as wavelength increases, so does beam radius. However it's put, a physical constraint of the CTC problem is that pressure takes more space to change by some amount for low-frequencies than it does for high-frequencies.

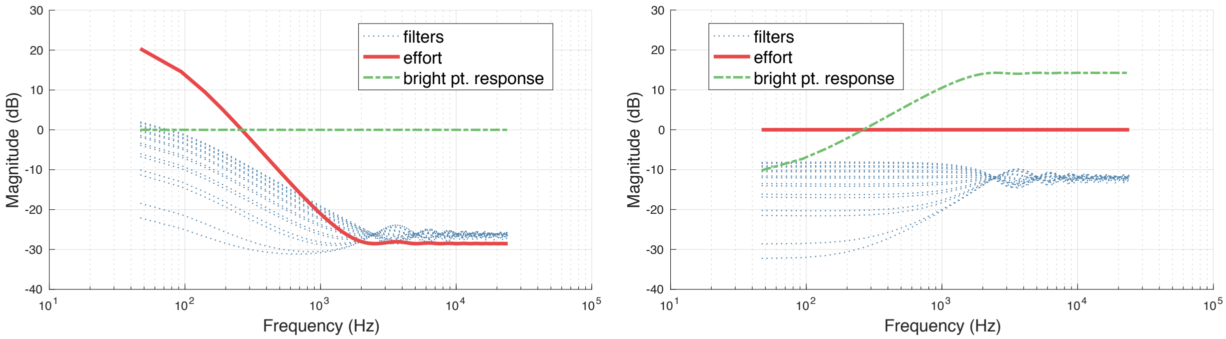
Meanwhile, the distance between a listener's ears, or control points, does not change with frequency. This is why it's difficult to generate CTC filters at low frequencies. In order to simultaneously produce unit gain at one ear and zero gain at the other (relatively close-by) ear, the peak sound pressure has to increase with wavelength. The legends on Figure 1.2 are marked to show which colors encode zero gain (at the right ear), unit gain (at the left ear), and peak gain (occurs in a frequency dependent location, farther from the listener as frequency decreases). It can be seen that range of pressure in Figure 1.2a is "squashed" compared to Figure 1.2b, which shows higher frequencies.

Why is it bad for low frequencies to require more filter gain than high frequencies?

The difficulty in producing low-frequency CTC is that exact solutions — filters that produce exactly the specified response — require extra gain at low frequencies. This causes lots of problems. First for example, full-bandwidth filters end up with huge gain distributions across the spectrum; ranging anywhere from roughly 20 to 50 dB depending on the configuration. See Figure 1.3 for an illustration of one example. Second, most of the energy it takes to produce specified ear signals is aimed away from the listener at low-frequencies. This is not true for high-frequencies however, meaning off-axis frequency responses sound low-passed. Third, it means that the direct-to-reflected signal ratio is worse for most rooms at lower frequencies. Finally, high gain distributions increase the chance of clipping loudspeakers. For all these reasons, listeners are sensitive to highly varied gain distributions across frequency [23].

What has been done about gain distributions in CTC filters?

It is well-documented that either generating or post-processing CTC filters to achieve flat spectra is only possible at the cost of reduced audio quality [3, 5, 6, 23, 27]. Two terms are given to the competing metrics: 'effort' and 'error'. Effort is a measure of gain distributed across an array's filters. Error is a measure of difference between the control point responses that were specified, and those that were produced by a set of filters. The branch of research that attempts to



(a) Normalized bright point responses require high effort.

(b) Normalized effort creates low-passed bright point responses.

Figure 1.3: This figure shows how the crosstalk cancellation system acts like a low-pass filter between filter gains (blue dots), their corresponding effort (red line), and ear responses (green dashes).

navigate this trade-off is called effort management (EM).

Several unique EM strategies are found in PMM literature. Each proposes to lower effort at the cost of lowering reproduction accuracy as well; in other words, by introducing error. Kirkeby et al. [2], in the seminal PMM paper, propose 0th-order, or Tikhonov, regularization as an EM strategy. Several extensions to PMM have also been proposed. In chronological order:

- Weighted least squares is proposed by Chang and Jacobsen [3].
- Constrained optimization with a convex solver is proposed by Bai et al. [5].
- Frequency-dependent regularization is proposed by Olivieri et al. [6].
- Regularized weighted least squares is proposed by Olivieri et al. [8].
- Regularization by eigenvalue clipping is proposed by Afghah et al. [10].

Why is EM still a difficult problem?

The relatively high number of unique EM strategies share a common approach: that EM is baked into spatial filters. This means that spatial filters are responsible for EM *in addition* to SFC. While there are benefits (e.g. computational efficiency) to combining SFC and EM transformations into a single filter set, the approach has limitations. Time-invariant EM, by definition, transforms every input signal the same way. Meanwhile, not all audio signals invoke the same amount of effort to filter. This is a problem because spatial filters are normally generated in order to keep worst-case effort below some threshold. As a result, all input signals — even those that are feasible to reproduce without EM — are processed as though they invoke worst-case effort. In other words, LTI EM introduces more error than necessary into audio signals that cost less than the maximum amount of effort to filter.

Why is LTI EM a problem for binaural audio?

CTC is often used with binaural audio, which is one way to create what's known as a virtual auditory environment [30–32]. If reproduction quality is good [33–35], binaural audio delivered via CTC can recreate the impression of full surround sound or natural directional hearing. The combination of binaural audio and CTC is interesting because it makes surround sound possible with smaller form factor systems than traditional multi-channel surround systems that may otherwise be used to create the same effect. The uniform linear array (ULA) is one example of a smaller form factor system[3]. In this work, binaural audio is defined as the linear combination of one or more unit-bounded, monophonic signals and their associated head related transfer function (HRTF).

At the same time, CTC is also often used with arbitrary stereo content; that is, any two-channel audio signal. This type of content, called 'general stereo' from now on, does not necessarily adhere to the same rigorous constraints as binaural audio, or even to those of classic stereo panning laws like amplitude panning [36, 37]. Chapter 3 shows that constraints on the type of input signal directly effect the worst-case filter effort. General stereo, for example, invokes exponentially higher worst-case effort than binaural audio in the context of CTC beamforming. However in practice, it is not unusual to use a single filter set to reproduce both binaural and general stereo audio. This causes binaural audio to be disproportionately penalized by EM.

[3] See Prependix for more details about the configurations used in this work.

Chapter 2

Theory

This chapter reviews the theoretical background needed to generate and analyze spatial filters with PMM. Free-field acoustics are reviewed in section 2.1, which are conventionally used by PMM to build the propagation matrix, generate filters, and analyze them. Least squares theory is reviewed in section 2.2. Conventional PMM filter generation and analysis techniques are reviewed in section 2.3. Finally, eigendecomposition and singular value decomposition (SVD) are reviewed in section 2.4. These tools (especially SVD) are reviewed because they can provide alternative perspectives on regularization, regularization's effect on invertible matrices like the propagation matrix, and by extension, its influence on the SFC metrics effort and error.

2.1 Acoustics

SFC methods are often only concerned with acoustics insofar as it provides a means to build a sound field model; or in other words, a model of interference patterns caused by multiple propagating waves. The simplest sound field model is the free-field, which makes two assumptions. First, that sources and receivers are ideal monopoles. Second, that the sound field is free of reflections, obstructions, or changes to the acoustic properties of the propagation medium.

Some branches of SFC, like acoustic contrast maximization [11], propose to exert control over spatial filters by manipulating the sound field model directly. Other branches of SFC, like PMM, offer additional types of control over filters through techniques like regularization. This work is focused exclusively on the latter type of control.

2.1.1 Propagation

The free field transfer function between a monopole source and receiver is described by the Green's function of the Helmholtz equation [38]; just called *the* Green's function for short from here. It is defined

$$G(r,\omega) = \frac{\exp(-ikr)}{4\pi r} \tag{2.1}$$

where r is absolute propagation distance between a source and receiver[1]. The Green's function is conventionally used to populate the propagation matrix $A \in \mathbb{C}^{M\times N}$ in pressure matching [2]. For clarity, M is the number of rows, receivers, or ears in the sound field; N is the number of columns, sources, or speakers. Matrix A is built

$$A = \begin{bmatrix} G(r_{11}) & \cdots & G(r_{1N}) \\ \vdots & \ddots & \vdots \\ G(r_{M1}) & \cdots & G(r_{MN}) \end{bmatrix} \tag{2.2}$$

where the distance r_{mn} is defined as that between receiver m and source n.

[1] The Green's function depends on frequency f; notated in Eq. 2.1 by the variables $\omega = 2\pi f$ for radial frequency, and $k = \omega/c$ for the wavenumber. For notational clarity, arguments related to frequency are excluded for the remainder of this work.

2.1.2 Interference

Sound pressure propagating from multiple distinct sources combines in the sound field to create interference patterns. Interference causes pressure at a point in space to vary according to a function of source locations and their associated filters $\boldsymbol{x}_n$. That function is a linear combination of the Green's functions for each source and receiver pair. For example at point m, the pressure $\boldsymbol{y}_m$ is calculated

$$\boldsymbol{y}_m = \sum_{n=1}^{N} G(r_{mn})\boldsymbol{x}_n \,. \tag{2.3}$$

Notice that the combined response of several source signals at a single point takes the form of a dot product between two complex vectors: the signals vector $\boldsymbol{x}$ and the vector $G(r_n)$ of transfer functions between all loudspeakers and the m^{th} receiver. Several dot products are calculated simultaneously in matrix multiplication. So, the responses at several points can be calculated by a single expression

$$\underset{(M\times 1)}{\boldsymbol{y}} = \underset{(M\times N)}{A} \underset{(N\times 1)}{\boldsymbol{x}} \,. \tag{2.4}$$

2.2 Least squares

PMM uses least squares to find filters $\boldsymbol{x}$ that cause naturally occuring interference patterns to best fit a specified set of control point responses $\boldsymbol{y}$. This is possible because the relationship between source signals, receiver signals, and their associated transfer functions are described by a linear system (Eq. 2.4).

Specifically, the filter coefficients that most closely cause a specified pressure to occur at control points can be found by solving either the inverse or the pseudoinverse of the propagation matrix. Assuming it's invertible, filter coefficients can be found by calculating the inverse

$$\boldsymbol{x} = A^{-1}\boldsymbol{y} \,. \tag{2.5}$$

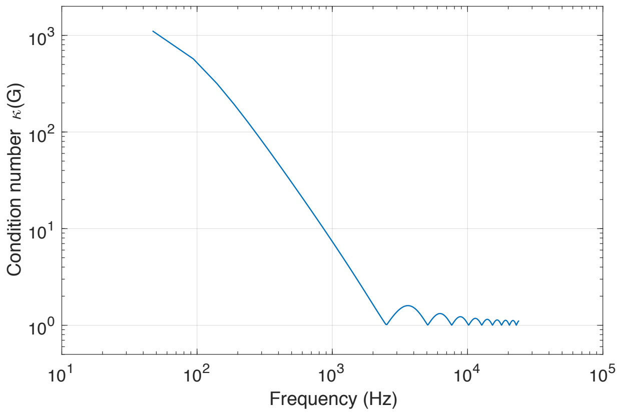


Figure 2.1: The condition number of the propagation matrix increases exponentially as frequency decreases. Higher condition numbers increase the tendency for high effort, as well as the ability of small errors to influence error.

For a few reasons however, practical PMM problems are unlikely to yield invertible propagation matrices. First, a propagation matrix is only square when there are an equal number of sources and receivers. Only square matrices are invertible. Second, practical PMM systems are often poorly conditioned even if they are square (see Figure 2.1). A physical symptom of poorly conditioned matrices is that small deviations to the input cause disproportionately large deviations to the output. This manifests as high error when a listener moves slightly, or if model parameters are even marginally misrepresented. Another physical symptom of poorly conditioned matrices is that loudspeakers with relatively little influence over the system are disproportionately amplified by a matrix inversion. This manifests as high effort in filters.

So depending on the reason, PMM either cannot or should not be solved by matrix inverse like Eq. 2.5 . Instead, Kirkeby et al. [2] propose to use Tikhonov regularization with least squares as a remedy to both symptoms of poorly conditioned matrices. Regularized least squares, at a high level, works by substituting a poorly conditioned matrix for a similar matrix that is well conditioned. Then, the unknowns vector $\boldsymbol{x}$ is solved by inverting the well conditioned matrix. Provided the well conditioned matrix is close enough to the original, the approximation $\boldsymbol{x}$ should be nearly correct.

In this work, the notation for pseudoinverse

$$\boldsymbol{x} = A^{+}\boldsymbol{y} \tag{2.6}$$

is extended to also include approximate matrix inversions like regularized least squares. The notation is used whenever a particular answer to an inverse problem is not important (usually to emphasize another idea). In order to describe Tikhonov regularization in more depth, however, it's necessary to first explain how least squares approximations are found for non-square systems.

2.2.1 Overdetermined least squares

Consider a tall system where matrix A has more rows than columns; i.e. there are more receivers than sources. Tall systems can be interpreted geometrically by picturing a vector $\boldsymbol{y}$ in a higher-dimensional space than the span of $A\boldsymbol{x}$. From the perspective of $\boldsymbol{y}$ in M-space, $A\boldsymbol{x}$ is just a subplane. And because ideally $A\boldsymbol{x} = \boldsymbol{y}$, tall systems are only solvable when $\boldsymbol{y}$ happens to lie on the subplane $A\boldsymbol{x}$, which would be lucky (and also means that A is rank deficient). Otherwise, the system has no solution. When tall systems have no solution, they are called 'overdetermined'.

Since the solution to an overdetermined system probably does not exist, the only alternative is to search instead for an approximation. Now, the suggestion of an approximation implies a measure of difference between $A\boldsymbol{x}$ and $\boldsymbol{y}$. In overdetermined least squares problems, that measure is the distance squared between the two, called 'error'. Error is expressed as a cost function

$$E(\boldsymbol{x}) = \|A\boldsymbol{x} - \boldsymbol{y}\|_2^2 \tag{2.7}$$

that depends on $\boldsymbol{x}$. Throughout this work, the function $E()$ will be used to represent the error caused by an approximation to a linear system.

To find a "best" approximation, it's necessary first to define some measure by which different approximations can be compared. If two different approximations were compared, it makes sense that the "better" one would have lower error as it's defined by the cost function for error (Eq. 2.7). Then, by extension, the best approximation is the one that minimizes that function. This objective is notated

$$\min_{\boldsymbol{x}} \|A\boldsymbol{x} - \boldsymbol{y}\|_2^2 \,. \tag{2.8}$$

The particular vector $\boldsymbol{x}$ that causes $E(\boldsymbol{x})$ to reach its minimum can be found analytically[2].

$$\boldsymbol{x} = \left(A^H A\right)^{-1} A^H \boldsymbol{y} \tag{2.9}$$

So, the least squares approximation to an overdetermined system is expressed

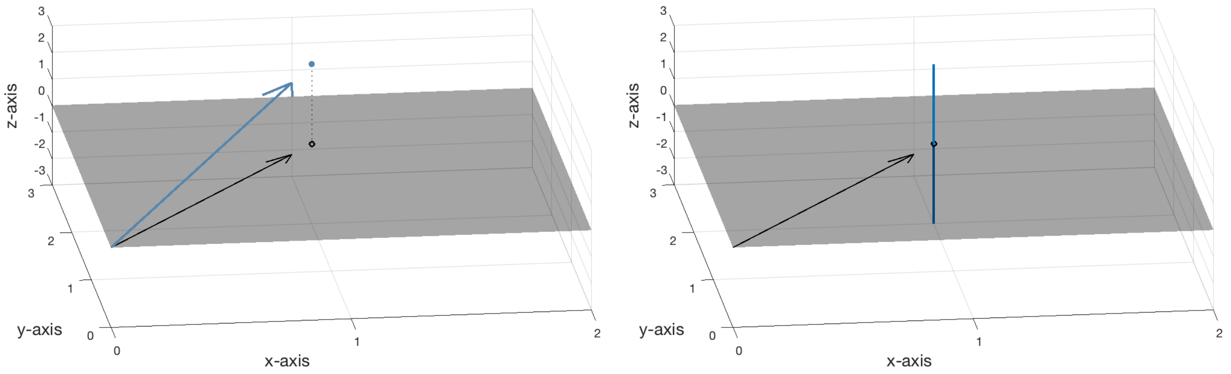
$$\boxed{\operatorname*{argmin}_{\boldsymbol{x}} \|A\boldsymbol{x} - \boldsymbol{y}\|_2^2 \quad = \quad \left(A^H A\right)^{-1} A^H \boldsymbol{y}} \tag{2.10}$$

Overdetermined toy example

The approximation of an overdetermined system has the following geometric rationale. The nearest point to $\boldsymbol{y}$ that can be reached by the subplane $A\boldsymbol{x}$ is the projection of $\boldsymbol{y}$ onto that subplane. The composition $\left(A^H A\right)^{-1} A^H$ is a projection operator which does just that. As an example, consider a tall system in $\mathbb{R}3 \times 2$, with the values

$$\boldsymbol{y} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad \boldsymbol{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

[2] Appendix A.1 shows the complete derivation.



(a) Overdetermined

(b) Underdetermined

Figure 2.2: Geometric representation of the toy examples for over- and underdetermined least squares.

The projection of $\boldsymbol{y}$ onto the subplane comprised of all possible values for $A\boldsymbol{x}$ creates the unique solution $\boldsymbol{x} = [1, 2]^T$. This particular vector is, by the metric of distance squared, the closest point to $\boldsymbol{y}$ and therefore the best approximate solution to this problem. Figure 2.2a illustrates the geometry of overdetermined least squares approximations using the example just given. The desired solution (blue), the least squares solution (black vector), and the span $A\boldsymbol{x}$ (black plane) in the overdetermined 3×2 example problem are shown.

2.2.2 Underdetermined least squares

Consider a wide system where matrix A has more columns than rows; i.e. there are more sources than receivers. Wide systems can be interpreted geometrically by picturing the continuous set of points that all project from higher-dimensions down onto the same lower-dimensional point. When there are infinite solutions for a wide system, it is called ‘underdetermined’.

Underdetermined systems are ill-posed because they have infinite solutions instead of just one. To find a best solution, it’s (just like the strategy for overdetermined systems) necessary to define some measure by which different solutions can be compared. In underdetermined least squares problems, that measure is distance squared from the origin to a solution, called ‘effort’.

Effort is expressed as a cost function

$$K(\boldsymbol{x}) = \|\boldsymbol{x}\|_2^2 \tag{2.11}$$

that depends on $\boldsymbol{x}$. Throughout this work, the function $K()$ will be used to represent effort.

If two different solutions were compared, it makes sense that the better one should have lower effort as it's defined by the cost function for effort (Eq. 2.11). Then, by extension, the best solution is the one that minimizes that function. This objective is notated

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_2^2 \,. \tag{2.12}$$

The particular vector $\boldsymbol{x}$ that causes $K(\boldsymbol{x})$ to reach its minimum can be found analytically[3].

$$\boldsymbol{x} = A^H \left(AA^H\right)^{-1} \boldsymbol{y} \tag{2.13}$$

So, the least squares solution to an underdetermined system is expressed

$$\boxed{\operatorname*{argmin}_{\boldsymbol{x}} \|\boldsymbol{x}\|_2^2 \quad = \quad A^H \left(AA^H\right)^{-1} \boldsymbol{y}} \tag{2.14}$$

Underdetermined toy example

Similarly to the overdetermined approximation, $\boldsymbol{x}$ is found via a projection operator. This time however, the projection operator is the composition $A^H \left(AA^H\right)^{-1}$. When A has full row rank, an entire subplane of points all project onto the same, lower-dimensional point, which makes them all valid solutions. The subplane of solutions is defined $\text{null}(A)\, b + \boldsymbol{x}$. That is, all linear combinations of the nullspace of A, translated by any (though usually the least squares) solution.

[3] Appendix A.2 shows the complete derivation.

As an example, consider a wide system in $\mathbb{R}2 \times 3$ with values

$$\boldsymbol{y} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad \boldsymbol{x} = \begin{bmatrix} 1 \\ 2 \\ z \end{bmatrix},$$

where the element z in vector $\boldsymbol{x}$ is a placeholder variable that can be any real number and still satisfy $A\boldsymbol{x} = \boldsymbol{y}$. This examples shows that the set of 3-dimensional vectors $\boldsymbol{x}$ are all projected by A onto the 2-dimensional point $\boldsymbol{y}$, regardless of the value of z. In this example, the set $\boldsymbol{x} = [1, 2, z]^T$ are solutions that form a subplane with $3-2=1$ degree(s) of freedom; aka a line. Figure 2.2b illustrates the underdetermined least squares example just given. The line of solutions (blue) and the least squares solution (black vector) to the underdetermined 2×3 example problem are shown.

2.2.3 Checkpoint: connecting least squares concepts to filter design

It's time to regroup and connect ideas from the previous chapter about PMM to the current discussion about linear systems.

Least squares in terms of filter design

The *indices* of the vector $\boldsymbol{x}$ correspond to the array's loudspeaker channel numbers. Its *values*, on the other hand, are the spatial filter coefficients for each corresponding loudspeaker. So when measuring the effort of a vector of spatial filters, the result $K(\boldsymbol{x}) = \|\boldsymbol{x}\|_2^2$ should be interpreted as a measure of gain required to reproduce control point responses for that particular configuration.

The *indices* of vector $\boldsymbol{y}$ correspond to the indices of control points. In CTC, control point indices are normally defined to correspond to input channel numbers (i.e. the first index of $\boldsymbol{y}$ corresponds to input channel one, and so on). The *values* of $\boldsymbol{y}$ are the specified control point responses. So when measuring the error between specified control point responses $\boldsymbol{y}$ and the actual control point responses of particular solution $A\boldsymbol{x}$, the result $E(\boldsymbol{x}) = \|A\boldsymbol{x} - \boldsymbol{y}\|_2^2$ should be interpreted as a measure of sound quality for the listener. Take this for now with a grain of salt: sound quality is represented by error only insofar as diminishing sound quality can be represented by increasing distance between two vectors. Small variations in perceived sound quality that may be important to careful listeners tend to be poorly-represented by this metric.

Underdetermined filter design

Underdetermined systems arise in CTC applications because there are more loudspeakers (columns) than control points (rows) in the propagation matrix. When a system is underdetermined, the least squares solution minimizes effort. Error, on the other hand, is always zero when a least squares problem is underdetermined, and cannot be directly influenced.

Overdetermined filter design

Overdetermined systems arise in private audio applications because there are fewer loudspeakers than control points in the propagation matrix. When a system is overdetermined, the least squares approximation minimizes error. Meanwhile effort is just along for the ride; it cannot be directly influenced by the strategy for approximating overdetermined problems.

Designing filters with competing objectives

Remember from chapter 1, that CTC filters solved by least squares often have too much gain in practice, despite being represented by underdetermined systems where effort is minimized. And, that to control effort, Kirkeby et al. [2] proposed the use of a technique called Tikhonov regularization. Then earlier in this section, it's shown that when a system is overdetermined, error is minimized; and, when a system is underdetermined, effort is minimized. So the shape of A is connected to the object of minimization. This observation motivated the intial decision to use Tikhonov regularization as an effort management technique. Tikhonov regularization combines the over- and underdetermined objectives into a single expression, which lets a filter designer specify the relative importance of minimizing either effort or error. It is not a cure all, however, because lowering effort beyond that given by an underdetermined solution requires introducing error. In fact, the two metrics are each improved at the cost of the other. So for filter designers, Tikhonov regularization is a method by which the relative importance of minimizing either effort or error is parameterized.

2.2.4 Regularized least squares

Regularized least squares can be expressed as a linear combination of the over- and underdetermined cost functions. This implies the cost function

$$J(\boldsymbol{x}) = a_1 \left\|A\boldsymbol{x} - \boldsymbol{y}\right\|_2^2 + a_2 \left\|\boldsymbol{x}\right\|_2^2 . \tag{2.15}$$

where a_1 and a_2 are real scalar weights. However, the solution to this cost function depends uniquely on the ratio $\beta = a_2/a_1$, instead of on either a_1 or a_2 individually. To derive a unique analytical solution, Eq. 2.15 is refactored with respect to the 'regularization parameter' β.

$$\min_{\boldsymbol{x}} \left\|A\boldsymbol{x} - \boldsymbol{y}\right\|_2^2 + \beta \left\|\boldsymbol{x}\right\|_2^2 \tag{2.16}$$

Controlling Tikhonov regularization works as follows. By gradually adjusting β from 0 to infinity, priority gradually shifts from minimizing only effort to only error. That is, when β is either 0 or infinity, the regularized least squares cost function becomes equivalent to that of either overdetermined or underdetermined systems respectively.

The regularized cost function (Eq. 2.15) is minimized when[4]

$$\boldsymbol{x} = \left(A^H A + \beta I\right)^{-1} A^H \boldsymbol{y} . \tag{2.17}$$

[4] Appendix A.3 shows the complete derivation.

So, the regularized least squares approximation is expressed

$$\boxed{\underset{\boldsymbol{x}}{\operatorname{argmin}} \|A\boldsymbol{x}-\boldsymbol{y}\|_2^2+\beta\|\boldsymbol{x}\|_2^2 \quad = \quad \left(A^H A+\beta I\right)^{-1} A^H \boldsymbol{y}} \tag{2.18}$$

The effect of regularization on matrix condition

Regularized least squares has a property that while $\beta > 0$, the square term from Eq. 2.18

$$A^H A + \beta I \tag{2.19}$$

is invertible regardless of the shape of A. Because of this, it is common for some theoretically-inclined texts to call Eq. 2.18 the complete solution to regularized least squares. However, the extent to which β improves the condition of a matrix actually depends on the shape of A.

Figure 2.3 shows how the conditions of both the overdetermined square term Eq. 2.19 , and, the underdetermined square term

$$AA^H + \beta I \tag{2.20}$$

change as β varies. In the scenario it illustrates, A is a wide 2×16 matrix created for a common

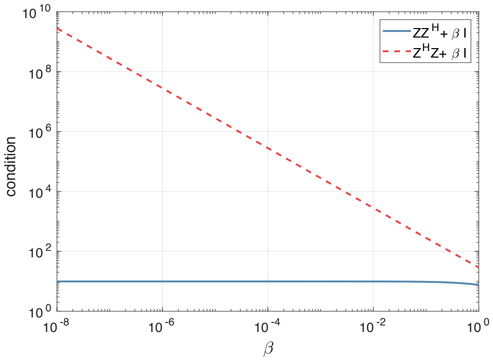


Figure 2.3: The condition of over- and underdetermined systems (at 400 Hz) when using the regularized least squares formula from Eq. 2.18 as β approaches zero. Clearly it is better to solve underdetermined systems with underdetermined formulations, even when using regularized least squares.

(underdetermined) CTC application[5]. It can be seen that while the condition of the underdetermined square term from Eq. 2.20 remains relatively low, the condition of the overdetermined square term from Eq. 2.19 increases exponentially as $\beta \rightarrow 0$. Remember from the beginning of section 2.2 that a high condition number is responsible for higher effort and a more fragile setup. Clearly then, a smaller condition number is necessary in order to reduce effort as efficiently as possible. Therefore a more robust solution — one which respects the condition of the square term — also depends on the shape of A.

The complete approach to regularized least squares, and by extension the to PMM, is built by substituting the appropriate square terms from Eqs. 2.10 and 2.14 into Eq. 2.18 , giving

$$\boxed{\boldsymbol{x} = \begin{cases} A^H\left(AA^H + \beta I\right)^{-1}\boldsymbol{y} & \text{if } M < N \\ \left(A + \beta I\right)^{-1}\boldsymbol{y} & \text{if } M = N \\ \left(A^H A + \beta I\right)^{-1} A^H\boldsymbol{y} & \text{if } M > N \end{cases}} \tag{2.21}$$

2.3 Generating and analyzing classical PMM filters

This section outlines how CTC filters are generated (subsection 2.3.1) and analyzed (subsection 2.3.3) with conventional, regularized PMM. It's important to realize that in this section, and through the rest of chapter 2, filters are generated for one input channel at a time. A workaround to generate filters for multichannel input is described at the end of this chapter. Then later, in chapter 3, it is shown how filters for multiple input channels can be generated by a single expression.

[5] See Prependix.

2.3.1 Generating CTC filters with PMM

This section describes how a filter designer uses the PMM algorithm to generate spatial filters for a CTC application. Normally, spatial filters are designed to map an impulse to one or more control points. For multi-channel input, the steps described here will need to be repeated once per channel.

First, generate a propagation matrix. Often the Green's function from Eq. 2.1 is used to generate each element, where its transfer functions depend on the distance between each speaker (source) and each ear (receiver).

Next, specify control points responses for the left (L) and right (R) input channels. In CTC, this is normally expressed

$$\boldsymbol{y}_{\mathrm{L}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{y}_{\mathrm{R}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{2.22}$$

where the ones and zeros represent bright and dark points. The vectors $\boldsymbol{y}_m$ are deceptively simple. The *indices* of the elements in each vector correspond to control point indices. Then in turn, control point indices are equal to their row in the propagation matrix. In CTC, control points often represent the left ear (first index) and right ear (second index). The *elements* in each vector represent how much of each input channel a filter designer wants to map to that control point. Take vector $\boldsymbol{y}_{\mathrm{L}}$, as an example, which has two elements; one for each ear. The particular values shown in Eq. 2.22 can be interpreted as a statement: "the left channel should have a gain of one at the left ear, and zero at the right ear; vice versa for the right channel". This statement reflects the CTC objective, which is to maximize the crosstalk ratio between bright and dark points.

In the third and final step, a value is chosen for β based on the desired balance of minimizing effort and error. See section 2.4.3 for a review of strategies on picking a good value for β. Finally, the regularized least squares approach from Eq. 2.21 is used to solve the system.

The explicit solutions to the left- and right-channel CTC filter sets are

$$\boxed{\begin{aligned} \boldsymbol{x}_\mathrm{L} &= A^H\left(AA^H+\beta I\right)^{-1}\boldsymbol{y}_\mathrm{L} \\ \boldsymbol{x}_\mathrm{R} &= A^H\left(AA^H+\beta I\right)^{-1}\boldsymbol{y}_\mathrm{R} \end{aligned}} \tag{2.23}$$

Choosing desired control point responses

An aside about Eq. 2.22: if frequency-domain wrap around must be avoided, or if PMM filters must be causal, the ones at bright points should be replaced by another unit-amplitude term that respects propagation delay. A normalized sum of Green's functions between each bright point and all N speakers would, for example, keep PMM filters causal if they replaced ones.

$$1 \quad \Longrightarrow \quad \frac{\sum_{N}^{n=1} G(r_{1n})}{\left|\sum_{N}^{n=1} G(r_{1n})\right|} \tag{2.24}$$

In this work, the value of bright spots will be written as ones instead of as in Eq. 2.24 to keep things clear. Both quantities have a magnitude of one, and the normalized sum of Green's functions will produce a linear delay term, or frequency-dependent rotations around the complex plane. A discussion about the nonzero values in $\boldsymbol{y}_m$ is relevant to another section of this work too; though it is only important for readers who are unsatisfied with the way Eq. 2.22 is written. For a more detailed discussion about this issue, see Appendix A.4.

2.3.2 Implementing CTC filters

This section briefly describes how input signals should interact with spatial filters in a beamforming implementation. The equations and perspectives needed to generate CTC filters with PMM are covered by the previous section. The conversation about β is delayed because the ability to meaningfully pick values depends on a more thorough understanding of how regularization affects filter performance (found in section 2.4.3). As a place to start in the mean time, it's enough for now to pick an arbitrary, frequency-independent value between $0 < \beta \leq 0.1$.

Once the filters for each input channel are generated ($\boldsymbol{x}_{\mathrm{L}}$ and $\boldsymbol{x}_{\mathrm{R}}$ as given by Eq. 2.23), they combine into a so-called 'convolution matrix' or 'filter matrix' $X \in \mathbb{C}^{N \times M}$ that transforms the input signal. The convolution matrix is built

$$X = [\boldsymbol{x}_{\mathrm{L}}, \boldsymbol{x}_{\mathrm{R}}]. \tag{2.25}$$

The input signal, notated $\boldsymbol{s} \in \mathbb{C}^{M}$, and the convolution matrix X combine linearly to generate the array signals vector, notated $\boldsymbol{w} \in \mathbb{C}^{N}$.

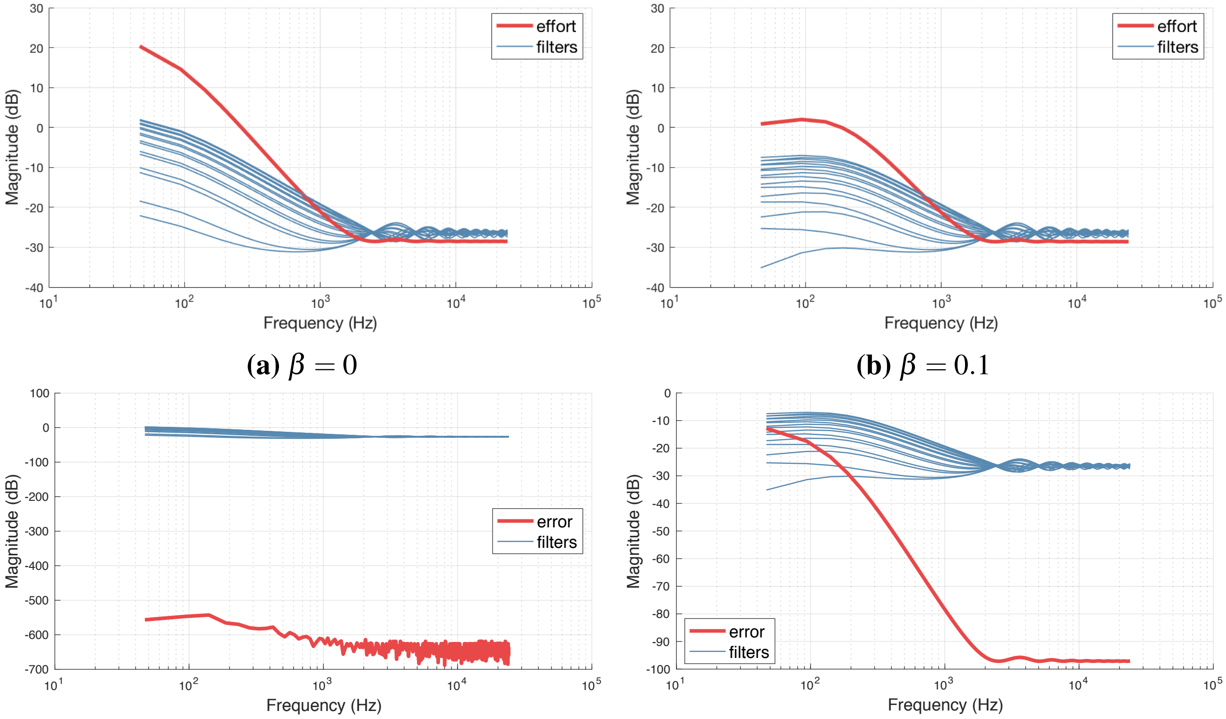
$$\boldsymbol{w} = X\boldsymbol{s} \tag{2.26}$$

The array signals are the signals actually played by the array's loudspeakers.

Variables that represent the input signal and convolution matrix are not normally mentioned in PMM literature, whether to discuss how they're generated or analyzed. Instead, the filters for just one channel are focused on. There is, in fact, a lot of information to analyze in the single-channel filters by themselves. So the remainder of this chapter reviews how PMM filters for single-channel input are conventionally analyzed. Then the next chapter contains a thorough discussion about PMM and multi-channel input signals.

2.3.3 Analyzing CTC filters with PMM

Common methods for analyzing PMM filters usually center around the two competing metrics: effort and error. Remember that because the CTC application is underdetermined, its possible to generate filters that exactly reproduce specified control point responses. At the same time, the filter effort required by these filters is usually too high at low frequencies. In this case, error is introduced into filters at problematic frequencies through regularization to reduce effort. So, analyzing CTC filters generated with PMM begins by focusing on the effect of regularization on effort and error.



(a) $\beta = 0$

(b) $\beta = 0.1$

(c) $\beta = 0.$

(d) $\beta = 0.1.$

Figure 2.4: Effort and error for left channel filters (red) and corresponding filter gains (blue) over frequency for various values of β.

The effect of regularization on effort and error

It has already been shown that increasing β has the effect of lowering effort and increasing error. The control point response error $E(\boldsymbol{x})$ for a set of filters $\boldsymbol{x}$ is expressed

$$E(\boldsymbol{x}) = \|A\boldsymbol{x} - \boldsymbol{y}\|_2^2 \,. \tag{2.27}$$

Likewise, The effort $K(\boldsymbol{x})$ of a set of filters $\boldsymbol{x}$ is expressed

$$K(\boldsymbol{x}) = \|\boldsymbol{x}\|_2^2 \,. \tag{2.28}$$

Hopefully no surprises here[6].

[6] These equations are identical to Eq. 2.7 and Eq. 2.11 , except now the input is assumed to be filters from Eq. 2.23 specifically.

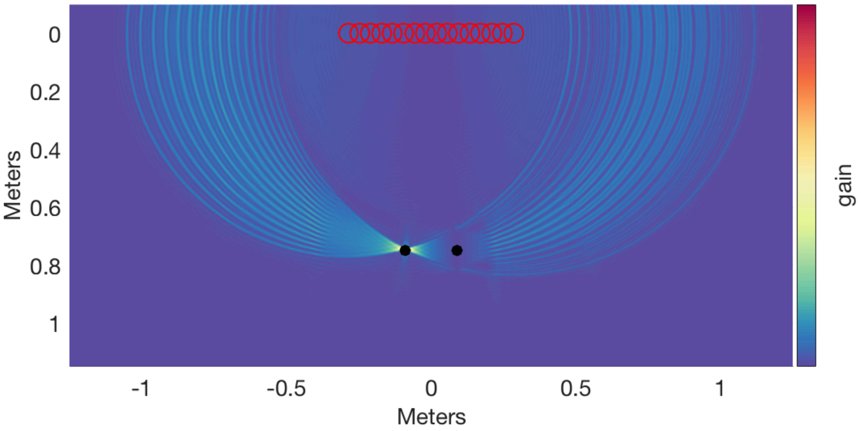


Figure 2.5: An example of time domain responses of PMM crosstalk cancellation filters over a grid of control points around the array (red circles) and listener's ears (back dots).

In this section, Eqs. 2.27 and 2.28 are used to create Figure 2.4, which compare the effort and error of CTC filters for $\beta = 0$ and $\beta = 0.1$. Subfigures **(a)** and **(b)** show the effort of filter gains for regularized and non-regularized CTC, while subfigures **(c)** and **(d)** show their error. Notice how much error must be introduced in order to reduce the filter effort of the lowest frequencies.

Control point responses in the sound field

To visualize the interference patterns, dense groups (e.g. a grid) of control points around the array and listener can be plugged into A. Then, their responses to a particular filter set are found by solving $A\boldsymbol{x}$. This is possible because plugging new control points into A changes its width, not its height. For example, Figures 1.1 and 1.2 from chapter 1 were generated this way. In those figures, sound pressure magnitudes at a specific frequency are encoded by color. These values are found simply by solving the absolute value of complex responses

$$\text{Absolute magnitude responses} := |A\boldsymbol{x}| \,. \tag{2.29}$$

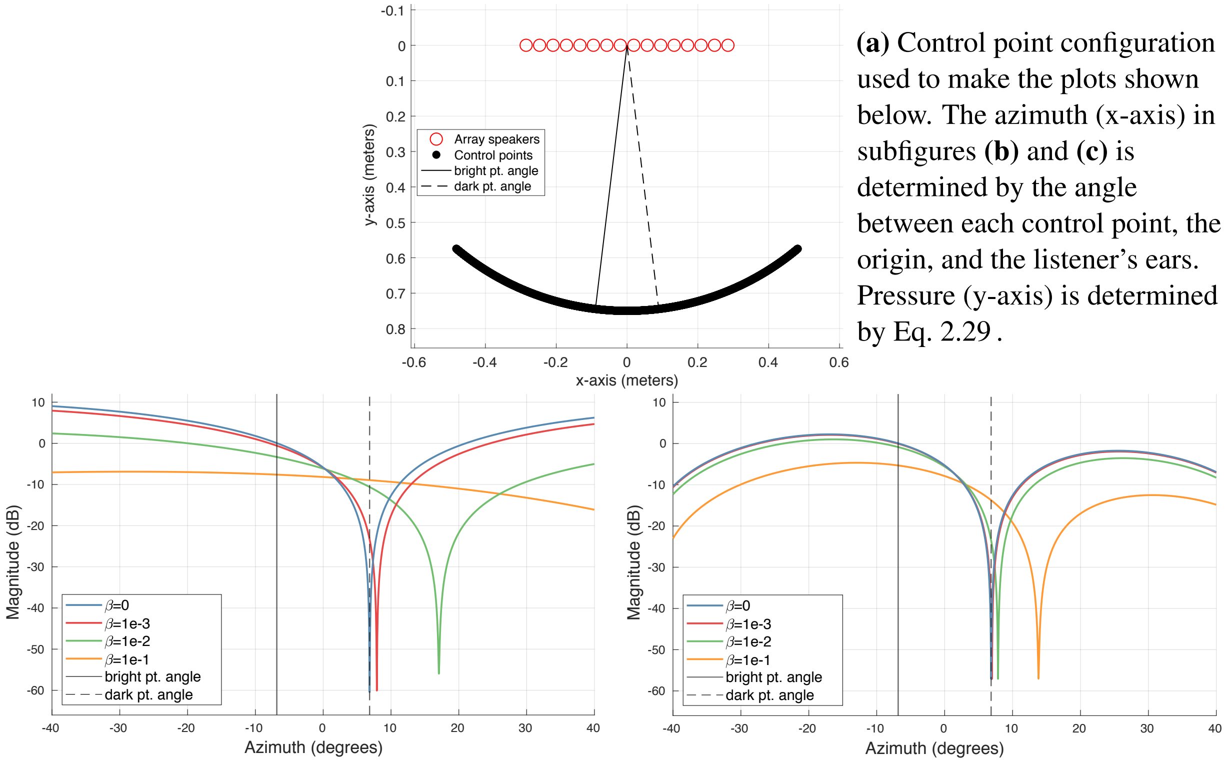
It's also possible, by extension, to measure control point responses in the time domain by taking the inverse discrete Fourier transform (iDFT) $\mathscr{F}^{-1}[\,]$ of a set of complex responses over uniformly distributed frequencies $\boldsymbol{\omega}$.

$$\text{Time domain responses} := \mathscr{F}^{-1}\left[A(\boldsymbol{\omega})\,\boldsymbol{x}(\boldsymbol{\omega})\right]. \tag{2.30}$$

An example of a filter sets' time domain responses is shown in Figure 2.5.

The effect of regularization on the sound field

It has been shown that increasing the regularization parameter β increases error; but what kind of error is introduced? Regularization has several effects on sound field pressure responses, which can be seen by using Eq. 2.29 to graph pressure at control points near a listener, which is shown in Figure 2.6.



(a) Control point configuration used to make the plots shown below. The azimuth (x-axis) in subfigures **(b)** and **(c)** is determined by the angle between each control point, the origin, and the listener's ears. Pressure (y-axis) is determined by Eq. 2.29 .

(b) Sound field pressure over azimuth at 400 Hz.

(c) Sound field pressure over azimuth at 1000 Hz.

Figure 2.6: Absolute pressure over azimuth in the sound field at various frequencies and values of β. The perspective shown in subfigures (b) and (c) is described by subfigure (a).

Interpret Figure 2.6 in the following way: the curves in subfigures **(b)** and **(c)** represent sound pressure magnitude over an arc of control points around the array, and that intersect the listener's ears. The control point configuration is shown in subfigure **(a)**. The vertical lines mark the angles of the left and right ears. Finally, good CTC filters would appear on this graph to have a spatial peak, or high gain, at the left ear (the solid black vertical line), and a spatial notch, or low gain, at the right ear (the dotted black vertical line).

Interpreting Figure 2.6 shows the effects of regularization on sound field pressure are, first, that increasing β smooths the sound field pressure gradient. As β increases, the heights of local sound pressure maxima are reduced. This is especially apparent at lower frequencies (subfigure **(b)**). Second, regularization causes the magnitude difference between bright and dark spots to decrease. As β increases, bright spot magnitudes decreases and dark spot magnitudes increases. It is possible to see that this effect is accomplished in part by shifting or rotating the spatial notch away from both control points.

So apparently, Tikhonov regularization both smooths sound field pressure and shifts the spatial notch away from the listener. These graphs also show that the effects of increasing β are stronger at lower frequencies. This is connected to the fact lower-frequency matrices have higher condition numbers (see Figure 2.1).

2.4 A deeper look at conventional PMM

This section extends concepts discussed in the previous section about analyzing PMM filters. The types of analyses introduced here will contribute to the development of more effective EM strategies in the following chapters. In subsection 2.4.1, the matrix factorization technique called eigendecomposition is discussed. It is used to help connect two concepts that until now have been treated informally: the linear algebra metric matrix condition, and the PMM metrics effort and error. In subsection 2.4.2, the concept of matrix decomposition is extended to include a discussion of SVD, which depends on eigendecomposition, and, is more powerful for this use-case. Finally, in subsection 2.4.3, an analytical solution is given for the regularization parameter β that lets a filter designer set matrix condition to a desired value.

2.4.1 Eigendecomposition and Tikhonov regularization

The matrix analysis tool called eigendecomposition provides a deeper look into how regularization effects matrix condition. It is used here to analyze the propagation matrix.

The eigendecomposition theorem states that any square matrix can be decomposed into eigenvectors, columns of matrix Q, and associated eigenvalues, the diagonal elements of matrix Λ [29]. However, the propagation matrices used in the particular PMM configuration examined by this work[7], are not square. So instead of using eigendecomposition to factor the propagation matrix A directly, the square terms from Eq. 2.21 must be used. They are

$$\begin{cases} AA^H & \text{if } M < N, \quad \text{(the CTC square term)} \\ A & \text{if } M = N, \\ A^H A & \text{if } M > N. \quad \text{(the private audio square term)} \end{cases} \tag{2.31}$$

Computing the eigendecomposition of a CTC square term results in the factorization

$$AA^H = Q\Lambda Q^{-1}, \tag{2.32}$$

where

$$Q = \begin{bmatrix} \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_M \end{bmatrix}, \qquad \Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_M \end{bmatrix}.$$

Using the CTC square term guarantees that the input to eigendecomposition will be square,

[7] According to the Prependix.

symmetric, and full-rank at non-zero frequencies. But why are those properties important? First, they ensure that the condition $\kappa(\,)$ of the CTC square term is defined by the ratio between the maximum and minimum eigenvalues

$$\kappa\left(AA^H\right) := \frac{\lambda_{\max}}{\lambda_{\min}}\,. \tag{2.33}$$

Using the CTC square term also ensures that the minimum eigenvalue will be greater than zero, and that the set of eigenvectors will be linearly-independent. Both of these properties are required for the factorization given by eigendecomposition to be invertible.

The inverse of a square matrix is calculated with its eigendecomposition factors

$$\begin{aligned}\left(AA^H\right)^{-1} &= \left(Q\Lambda Q^{-1}\right)^{-1} \\ &= Q\Lambda^{-1}Q^{-1}\,.\end{aligned} \tag{2.34}$$

It is important to realize that since matrix Λ is diagonal, its inverse is calculated by taking the reciprocal of its diagonal elements.

$$\Lambda^{-1} := \begin{bmatrix} \frac{1}{\lambda_1} & & \\ & \ddots & \\ & & \frac{1}{\lambda_M} \end{bmatrix} \tag{2.35}$$

These definitions help to more clearly show how the condition of the CTC square term is changed by applying Tikhonov regularization. Adding regularization into Eq. 2.32 gives

$$AA^H + \beta I = Q(\Lambda + \beta I)\,Q^{-1}\,. \tag{2.36}$$

So eigendecomposition exposes how Tikhonov regularization improves the condition of a matrix: by adding β to each of its eigenvalues. The same perspective can be used to show how the condition of the regularized inverse of the CTC square term is changed as well.

$$\left(AA^H + \beta I\right)^{-1} = Q(\Lambda + \beta I)^{-1} Q^{-1} . \tag{2.37}$$

That is, small eigenvalues are increased by β before taking their reciprocal. This prevents them from blowing up into extremely large values, which creates instability and fragility as discussed by section 2.2. In summary, Tikhonov regularization improves the condition of a square symmetric matrix by adding β to its eigenvalues, which decreases the ratio between them.

$$\left(\kappa\left(AA^H\right) = \frac{\lambda_{\max}}{\lambda_{\min}}\right) \quad > \quad \left(\kappa\left(AA^H + \beta I\right) = \frac{\lambda_{\max} + \beta}{\lambda_{\min} + \beta}\right) \tag{2.38}$$

This observation will be useful in a later chapter when examples of effort compression are proposed.

2.4.2 Singular value decomposition and Tikhonov regularization

In this work, singular value decomposition (SVD) is used in lieu of eigendecomposition. Both factorizations are useful for making the types of observations about matrix condition and regularization that were demonstrated in the previous subsection. However the SVD is more flexible, since it can be applied to any matrix; not just square matrices. Its terms also have more useful properties, and are used in later chapters about effort compression. This subsection contains a brief overview of its properties that will be relied on for the rest of the work.

The SVD of matrix A is defined

$$A = U \Sigma V^H . \tag{2.39}$$

The matrix Σ is a diagonal matrix whose diagonal elements are called singular values. Singular values are defined as the square root of the eigenvalues of AA^H and A^HA. They're also normally sorted so that $\boldsymbol{\sigma}_1 > \boldsymbol{\sigma}_2 > \ldots \boldsymbol{\sigma}_M$. The condition of a matrix is always defined by singular values, like Eq. 2.33 , but, includes non-square matrices too.

$$\kappa(A) := \frac{\sigma_{\max}}{\sigma_{\min}} \tag{2.40}$$

Because matrix Σ is diagonal, its inverse is found by taking the reciprocal of its diagonal elements. So, a matrix is only invertible if its smallest singular value is greater than 0: $\boldsymbol{\sigma}_{\min} > 0$.

Matrices U and V, on the other hand, are defined as the eigenvectors of AA^H and A^HA respectively. They are always both linearly-independent and normal, and also, are orthogonal when A has full rank. These properties come in handy because the Hermitian transpose of an orthonormal matrix is identical to its inverse

$$U^H := U^{-1}, \quad \text{and} \quad V^H := V^{-1}. \tag{2.41}$$

This identity will be used in later chapters. Finally, the inverse of A in terms of SVD is given

$$\begin{aligned} A^{-1} &= (U\Sigma V^H)^{-1} \\ &= \left(V^H\right)^{-1}\Sigma^{-1}U^{-1} \\ &= V\Sigma^{-1}U^H \end{aligned} \tag{2.42}$$

This also means Tikhonov regularization can be expressed in terms of the SVD of the CTC square term, as follows.

$$AA^H = U\Sigma V^H \tag{2.43}$$

$$\begin{aligned} \left(AA^H\right)^+ &= \left(U\Sigma V^H\right)^+ \\ &= V\Sigma^+U^H \end{aligned} \tag{2.44}$$

$$\left(AA^H + \beta I\right)^+ = V(\Sigma + \beta I)^{-1}U^H \tag{2.45}$$

Or, in terms of the SVD of the non-square propagation matrix

$$A^{+} = V \frac{\Sigma}{\Sigma^2 + \beta I} U^H . \tag{2.46}$$

2.4.3 Finding the ideal regularization parameter

Using SVD, it is possible to analytically find the value of β that achieves a desired condition number $\hat{\kappa}(A)$ with Tikhonov regularization. The only constraints on this problem are that actual condition is greater than desired condition, and that desired condition does not equal one[8]. Both constraints can be notated together as

$$\kappa(A) > \hat{\kappa}(A) > 1 . \tag{2.47}$$

Let σ_n denote singular values of A such that $\sigma_1 > \cdots > \sigma_N > 0$, and let $\hat{\kappa}$ represent a shorthand for desired condition. Then, the value of β that achieves a desired condition can be found in terms of singular values

$$\begin{aligned}
\frac{\sigma_1 + \beta}{\sigma_2 + \beta} &= \hat{\kappa} \\
\sigma_1 + \beta &= \hat{\kappa}\sigma_2 + \hat{\kappa}\beta \\
\beta - \hat{\kappa}\beta &= \hat{\kappa}\sigma_2 - \sigma_1 \\
\beta(1 - \hat{\kappa}) &= \hat{\kappa}\sigma_2 - \sigma_1 \\
\beta &= \frac{\hat{\kappa}\sigma_2 - \sigma_1}{1 - \hat{\kappa}} \\
&= \frac{\sigma_1 - \hat{\kappa}\sigma_2}{\hat{\kappa} - 1}
\end{aligned} \tag{2.48}$$

[8] When the desired condition is one, the Tikhonov regularization solution for $\beta = \infty$. Also, if the desired condition is higher than the actual condition, then there's no need to regularize.

So the value of β that achieves desired condition with Tikhonov regularization is

$$\boxed{\beta = \frac{\sigma_1 - \hat{\kappa}\sigma_2}{\hat{\kappa} - 1}} \tag{2.49}$$

To find a regularization parameter that achieves desired filter effort, Olivieri et al. [6] suggest two numeric algorithms, the Iterative Tikhonov and Normalized Tikhonov methods. There is no analytical solution for the regularization parameter that achieves a desired filter effort.

2.5 Conclusion

This chapter has shown how to generate and analyze PMM filters with conventional techniques. It has also shown how filters are conventionally thought of on an individual, per-channel basis, even when used with multi-channel input signals like stereo. The next chapter marks the beginning of a departure from traditional perspectives on SFC. In particular, the variables for multi-channel input signals $\boldsymbol{s}$ and array signals $\boldsymbol{w}$ are invoked to discuss how multi-channel input signals interact with the PMM algorithm. They also serve to establish a framework for effort compression. Later, the effect of multi-channel input on effort is explored.

Chapter 3

Multi-channel effort management

The purpose of this chapter is to show how the PMM metrics effort and error are affected by the combination of EM and dynamic, multichannel input. Observations made in this chapter should serve, first, to increase practical intuition about how PMM works with multi-channel input, then, to draw attention to related issues that are not directly addressed by conventional PMM literature, and ultimately, to begin the development of a framework for effort compression.

Specifically, this chapter focuses on what an array's loudspeakers actually play when multiple input channels combine with their associated spatial filters. Before the end of this chapter, it will become clear that, while each channel of input may combine with their filters in obvious ways, multiple channels of spatially filtered audio combine into array signals in non-obvious ways; especially in terms of a new SFC metric 'array effort', which will replace filter effort from previous chapters. Finally, anything that relates to the array's output signals in multi-channel contexts will from here on be called 'multi-channel PMM'.

3.1 Establishing multi-channel PMM

In the previous chapter, the single-channel PMM method for CTC filters (Eq. 2.23) was shown. Though it's supposed to be used to generate filters for one input channel at a time, it was shown later, in section 2.3.2, how to use single-channel PMM with multi-channel input. It was also briefly shown how a multi-channel input signal and filters combine together at the array: filters for each input channel form a convolution matrix (Eq. 2.25), which produces array signals through matrix multiplication with the input signal. This section combines these two ideas (conventional filter generation and the convolution matrix) by showing how the entire convolution matrix can be generated in a single expression.

This section's main argument is that, despite similar notation, generating an entire filter matrix at once is significantly different than the conventional, single-channel approach. In this work, establishing a method to convert single-channel formulations into multi-channel ones is a necessary step toward being able to analyze the relationship between input signals and PMM metrics.

3.1.1 One-lining the convolution matrix

To start, recall from Eq. 2.25 that the convolution matrix X is constructed by concatenating the left- and right-channels' filters.

$$X = [\boldsymbol{x}_{\mathrm{L}}, \boldsymbol{x}_{\mathrm{R}}]$$

Also, recall from Eqs. 2.22 and 2.23 how filters for each input channel are solved[1].

$$\boldsymbol{x}_{\mathrm{L}} = A^{+}\boldsymbol{y}_{\mathrm{L}}\,, \qquad \text{and} \qquad \boldsymbol{x}_{\mathrm{R}} = A^{+}\boldsymbol{y}_{\mathrm{R}}$$

[1] As a reminder, the notation A^{+}, which means the pseudoinverse of A, is (ab)used in this work for two reasons. First, since regularized least squares is a pseudoinverse method, A^{+} can be used as a shorthand for that particular spatial filter generation strategy. Second, as new strategies for solving PMM are introduced, the notation A^{+} can also be used to represent the spatial filter generation strategy for them too, when the distinction is not important.

The solutions for filters can be rewritten by substituting the particular values of $\boldsymbol{y}_m$ that are used in Eq. 2.22 for CTC.

$$\begin{aligned} \boldsymbol{x}_{\mathrm{L}} = A^{+}\boldsymbol{y}_{\mathrm{L}} \quad &\Longrightarrow \quad \boldsymbol{x}_{\mathrm{L}} = A^{+}\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \\ \boldsymbol{x}_{\mathrm{R}} = A^{+}\boldsymbol{y}_{\mathrm{R}} \quad &\Longrightarrow \quad \boldsymbol{x}_{\mathrm{R}} = A^{+}\begin{bmatrix} 0 \\ 1 \end{bmatrix}. \end{aligned} \tag{3.1}$$

Then, substituting the result back into the convolution matrix form lets Eq. 2.25 be rewritten as

$$X = \begin{bmatrix} A^{+}\begin{bmatrix} 1 \\ 0 \end{bmatrix}, A^{+}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix}. \tag{3.2}$$

This shows that the convolution matrix X is just columns of the (pseudo)inverted propagation matrix A^{+}, indexed by the non-zero elements of $\boldsymbol{y}$. This is important. It means columns of A^{+} are themselves spatial filters that attempt to uniquely reproduce each input channel at its corresponding control point. It follows then that when the matrix A^{+} is factored out of the above expression, and the goal of PMM is to deliver M input channels uniquely to M control points (as it is for CTC), the vectors y_m are turned into the identity

$$X = A^{+}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \Longrightarrow \qquad X = A^{+}I \tag{3.3}$$

and reveals that

$$X = A^{+}. \tag{3.4}$$

That is, the convolution matrix in CTC is equal to the pseudoinverse of the propagation matrix.

The observation shown by Eq. 3.4 has several benefits for CTC filter designers. It demonstrates that filters for all input channels can be either generated or analyzed together via the pseudoinverse of the propagation matrix alone. It helps express that PMM filters are generated by attempting to preemptively "undo" the transformation caused by acoustic propagation. Perhaps most significantly, it allows the single-channel PMM formulation to be converted into the multi-channel formulation[2].

So, for example, regularized single-channel PMM can be converted to its multi-channel form using Eq. 3.4 whenever the vectors $\boldsymbol{y}_m$ can be factored out; as is the case with CTC.

$$\boxed{\left.\begin{aligned} \boldsymbol{x}_\mathrm{L} &= A^H\left(AA^H+\beta I\right)^{-1}\boldsymbol{y}_\mathrm{L} \\ \boldsymbol{x}_\mathrm{R} &= A^H\left(AA^H+\beta I\right)^{-1}\boldsymbol{y}_\mathrm{R} \\ X &= [\boldsymbol{x}_\mathrm{L},\boldsymbol{x}_\mathrm{R}] \end{aligned}\right\} \quad \Longrightarrow \quad X = A^H\left(AA^H+\beta I\right)^{-1}} \tag{3.5}$$

Otherwise more generally, $\boldsymbol{y}_m$ can be treated like $\boldsymbol{x}_m$ above, where its columns are concatenated into the matrix Y

$$Y = [\boldsymbol{y}_\mathrm{L},\boldsymbol{y}_\mathrm{R}], \tag{3.6}$$

which gives

$$\left.\begin{aligned} X &= [\boldsymbol{x}_\mathrm{L},\boldsymbol{x}_\mathrm{R}] \\ Y &= [\boldsymbol{y}_\mathrm{L},\boldsymbol{y}_\mathrm{R}] \end{aligned}\right\} \quad \Longrightarrow \quad X = A^+Y \tag{3.7}$$

The latter multi-channel PMM formulation (Eq. 3.7) allows the vectors $\boldsymbol{y}_m$ to have arbitrary values instead of columns of the identity like in Eq. 2.22 . Furthermore, the effort compression methods presented in later chapters work just as well with this more general notation. However, this work prefers the notation in Eq. 3.5 over Eq. 3.7, since in this particular application, Y will always equal the identity and it keeps the notation simpler to exclude it.

[2] A longer discussion about this conversion, and alternative channel- to multi-channel conversions are presented in Appendix A.4. In short, this exact notation causes non-causal filters. It is trivial to fix (see Eq. 3.7), but adds visual noise that detracts from the intended point.

3.2 Establishing signal-dependent PMM

The previous section shows that columns of A^+ are themselves spatial filters. This section builds on that observation in order to explicitly show how multi-channel SFC components — the input, output, and array signals — affect effort and error.

So this section begins by formally introducing variables for these new components[3]. The input signal vector $\boldsymbol{s} \in \mathbb{C}^M$ represents one bin of a discrete Fourier transform (DFT) on M-channels of audio. Since this work deals exclusively with stereo input signals, the size of $\boldsymbol{s}$ is assumed to be $M = 2$. To be clear how these new variables fit into the PMM signal flow, one channel of the input $\boldsymbol{s}_m$ is multiplied by one column of the filter matrix to give the signal played by the array *for that channel*. This is expressed[4]

$$\underset{(N\times 1)}{\boldsymbol{w}_m} = \underset{(N\times 1)}{X_m} \underset{(1\times 1)}{\boldsymbol{s}_m} . \tag{3.8}$$

Likewise, the composite signals played by the array $\boldsymbol{w}$, called the array signals, are the result of applying the entire filter matrix to the entire input signal.

$$\underset{(N\times 1)}{\boldsymbol{w}} = \underset{(N\times M)}{X} \underset{(M\times 1)}{\boldsymbol{s}} \tag{3.9}$$

At last the output signal $\hat{\boldsymbol{s}}$ is introduced. The output signal is the signal measured at control points after the input signal is filtered and propagates. Remember that the meaning of the hat symbol over $\hat{\boldsymbol{s}}$ indicates that its desired value is equal to the input signal; ideally $\hat{\boldsymbol{s}} = \boldsymbol{s}$. Formally, the output signal is the combination of the array signals and the propagation matrix.

$$\underset{(M\times 1)}{\hat{\boldsymbol{s}}} = \underset{(M\times N)}{A} \underset{(N\times 1)}{\boldsymbol{w}} \tag{3.10}$$

[3] These variables were introduced in chapter 2, but thoroughly glossed over.

[4] Indexing vector $\boldsymbol{w}$ by m is an abuse of notation meant to express the combination of filters and input for channel m. This makes $\boldsymbol{w}_m$ a different vector than the combination of multiple channels with their filters.

3.2.1 Checkpoint: the complete PMM signal flow

At this point, notation has been introduced that represents the entire PMM signal flow invoked by multichannel beamforming. It's a good time to summarize. Eq. 3.9 shows how the input signal $\boldsymbol{s}$ is filtered by the convolution matrix X to create array signals $\boldsymbol{w}$. Eq. 3.10 shows how the array signals propagate through the sound field matrix A to create output signals $\hat{\boldsymbol{s}}$. The process that connects Eqs. 3.10 and 3.9 can be described by the following "signal flow graph".

$$\overbrace{\boldsymbol{s} \rightarrow X \rightarrow \boldsymbol{w}}^{\boldsymbol{w} = X\boldsymbol{s}} \underbrace{\rightarrow A \rightarrow \hat{\boldsymbol{s}}}_{\hat{\boldsymbol{s}} = A\boldsymbol{w}} \tag{3.11}$$

The first half of this expression, $\boldsymbol{w} = X\boldsymbol{s}$, will be discussed in the next section (section 3.3), which shows the effect of input signals on effort *at the array*. Then, adding the second half of the expression, $\hat{\boldsymbol{s}} = A\boldsymbol{w}$, to the discussion in section 3.4 will help demonstrate the relationship between EM, error, and the input signal.

3.3 The effect of input signal on array effort

Starting in this section and for through the remainder of this work, the array signals $\boldsymbol{w}$ from Eq. 3.9 will serve as the input to the cost function for effort (Eq. 2.11). That is, array signals $\boldsymbol{w}$ becomes the input to the effort cost function

$$K(\boldsymbol{w}) = \|\boldsymbol{w}\|_2^2 \tag{3.12}$$

instead of, for example, filters $\boldsymbol{x}$. To see that input signals do actually effect array effort, Eq. 3.9 is used to substitute $X\boldsymbol{s}$ for $\boldsymbol{w}$, giving

$$K(\boldsymbol{w}) = \|X\boldsymbol{s}\|_2^2 \,. \tag{3.13}$$

Before digging into this equation however, it's first used to build intuition about the range of values for array effort that are produced by a given stereo input signal, as well as about the likelihood that they are produced.

3.3.1 Stereo effort in unit circle

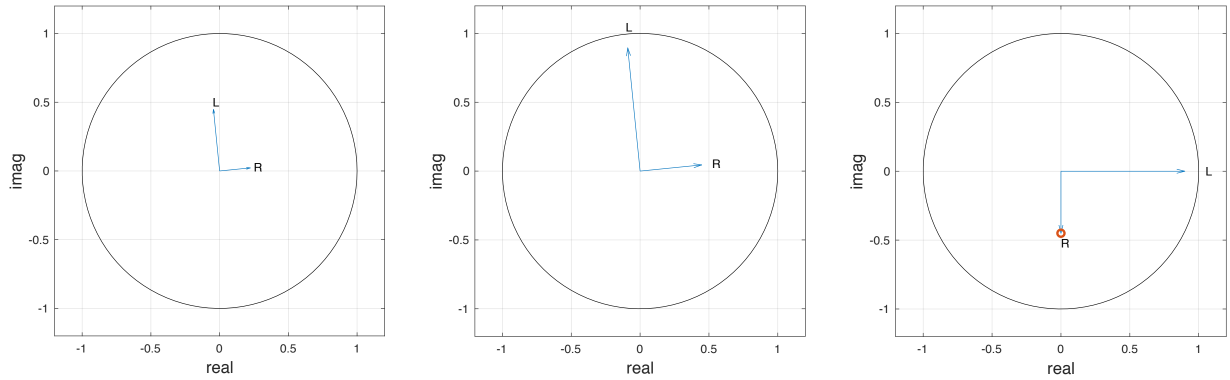
Consider that the set of stereo input signal s are finite, since s is band-, channel-, and gain-limited. This fact is used, in conjunction with Eq. 3.13 from the previous section, to graph a representation of how a given stereo signal effects array effort.

This section introduces a method for graphically representing PMM metrics based on the set of normalized stereo input signals. It will be used specifically to show how interchannel phase angle and magnitude difference influence array effort. In these graphs, one DFT bin of a stereo signal is mapped onto one point of the complex plane. Then the array effort that corresponds to that signal is encoded by color. To skip ahead to that graph, see Figure 3.2. First however, details of the stereo signal to unit circle mapping are discussed.

Description of stereo map

This section explains how a stereo signal is mapped to a single point on the complex plane to form a unit circle. See Figure 3.1 for an example of the proposed mapping.

First, both channels are normalized so that the louder channel has a magnitude of 1 (subfigure **(a)**). Second, both channels are rotated until the louder channel lands on the point $1+0i$ (subfigure **(b)**). Afterwards, the effort caused by that signal is plotted at the complex coordinates of the quieter channel (at the red circle in subfigure **(c)**). This point is necessarily bounded by the unit circle. So, the plotted point represents a unique phase angle and gain ratio between two channels. And since PMM metrics depend uniquely on the interchannel qualities of a stereo signal, this mapping represents all possible values for the array effort of a stereo CTC system.

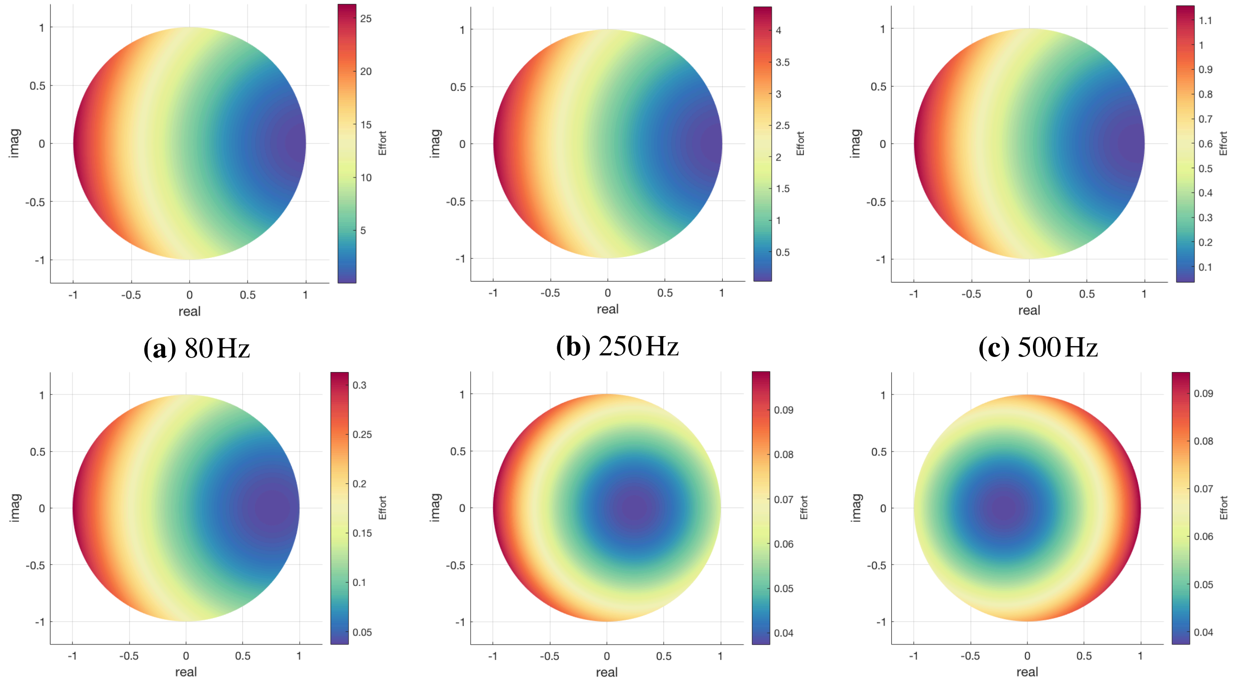


(a) One bin of a complex stereo signal.

(b) Step 1: normalize both channels.

(c) Step 2: rotate. The quieter channel (red circle) is plotted.

Figure 3.1: An example of one bin of a stereo signal, mapped onto the unit circle.



(a) 80 Hz

(b) 250 Hz

(c) 500 Hz

(d) 1 kHz

(e) 2 kHz

(f) 4 kHz

Figure 3.2: Effort to reproduce stereo signals at various frequencies; plotted at the normalized, interaural phase angle.

Observations about array effort of stereo signals from the unit circle map

Examining Figure 3.2 leads to several observations.

First, the maximimum (or worst-case) array effort that can be invoked by one bin of a normalized stereo input signal decreases exponentially with frequency. For example, subfigure **(a)** shows that the worst-case effort at 80 Hz is roughly 5 times higher than at 250 Hz. Meanwhile at 250 Hz, worst-case effort is roughly 4 times higher than at 500 Hz. So on and so forth[5]. From this, it is possible to confirm that worst-case array effort is a function of the ratio between wavelength and the distance between a listener's ears.

The relationship between worst-case effort and frequency in these plots indicates that EM is not urgent for generalized stereo input over about 1 kHz, which is consistent with observations made in practical beamforming implementations [6]. If regularization is necessary at higher frequencies, it would have to be for reasons other than effort management. For example, it can be used to influence (e.g. smooth out) the sound field pressure gradient, as was shown in Figure 2.6.

Second, it can be observed that the location of minimum and maximum effort values are similar for frequencies with wavelengths larger than half the distance between ears, called ITD-centric frequencies in reference to duplex theory [39] [6]. The extrema of effort for these frequencies occurs either at or near the points $1+0i$ (which represents when both channels have unit gain and *equal* phase) and $-1+0i$ (which represents when both channels have unit gain and *opposite* phase). Conversely, high frequency extrema seem to occur at other points, but still always along the real axis; i.e. never with imaginary components. This implies that interchannel gain differences have a stronger effect on array effort than interchannel phase.

Finally, these graphs show that at frequencies higher than roughly 2 kHz, that CTC starts to occur naturally, or at least that PMM needs less effort to deliver stereo signals than monophonic

[5] The complete picture of worst-case array effort for stereo signals over frequency is shown by Figure 3.4.

[6] Duplex theory identifies two localization cues used to binaurally estimate azimuthal direction of arrival (DOA): interaural time difference (ITD) and interaural level difference (ILD). ITD is used primarily to identify low-frequencies and ILD is used primarily to identify high-frequency components.

ones. One can see, for example, that it requires more effort at 4 kHz to deliver the signals represented by the point $1+0i$ than by those represented by the point $0+0i$.

3.3.2 Worst-case binaural effort

The last section shows that array effort is a function of stereo signal properties, including frequency, interchannel phase angle, and magnitude difference. Figure 3.2 illustrates this by mapping array effort to the set of all stereo signals. However the set of *binaural* signals (when defined as any monophonic signal convolved with any HRTF) is a subset of stereo signals. For example, it is possible to design a low-frequency (e.g. 80 Hz) stereo signal where the left and right channels have opposite phase. This hypothetical signal would be represented by the point $-1+0i$ in Figure 3.2. That same signal could not occur in binaural audio, since the largest interaural phase angle occurs at a DOA of ±90°, and it is a ratio of ear distance to wavelength.

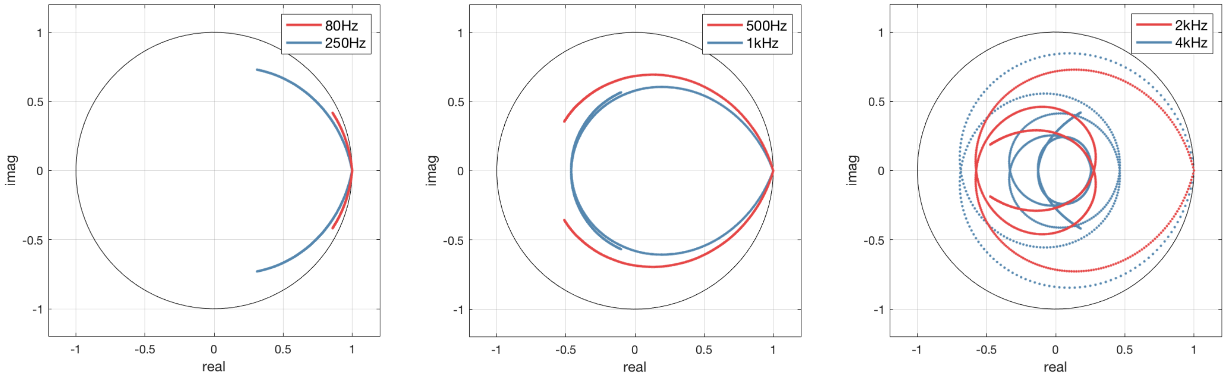
The maximum interaural phase angle for a frequency received by two points in the free field can be calculated. If the speed of sound is $c = 343\,\mathrm{m{\cdot}s^{-1}}$, then the wavelength of a $f = 80\,\mathrm{Hz}$ signal is

$$\lambda = \frac{c}{f} \approx 4.29\,\mathrm{m}\,. \tag{3.14}$$

Then if the average interaural distance is $\Delta y_{\mathrm{pos}} = 0.18\,\mathrm{m}$, the largest interaural phase angle between two ears in the free field is only

$$\begin{aligned} \frac{2\pi\,\Delta y_{\mathrm{pos}}}{\lambda} &= \mathrm{rad} \\ \frac{2\pi \cdot 0.18}{4.29} &\approx 0.26\,\mathrm{rad} \end{aligned} \tag{3.15}$$

This is only useful to get a rough idea of the maximum interaural phase angle per frequency in a binaural context. The space between the ears is not a free field, and, interaural delay is nonlinear [1]. So, Figure 3.3 offers a more precise description of which signals can occur in binaural audio by mapping an HRTF set onto the unit circle (as before; see Figure 3.1).

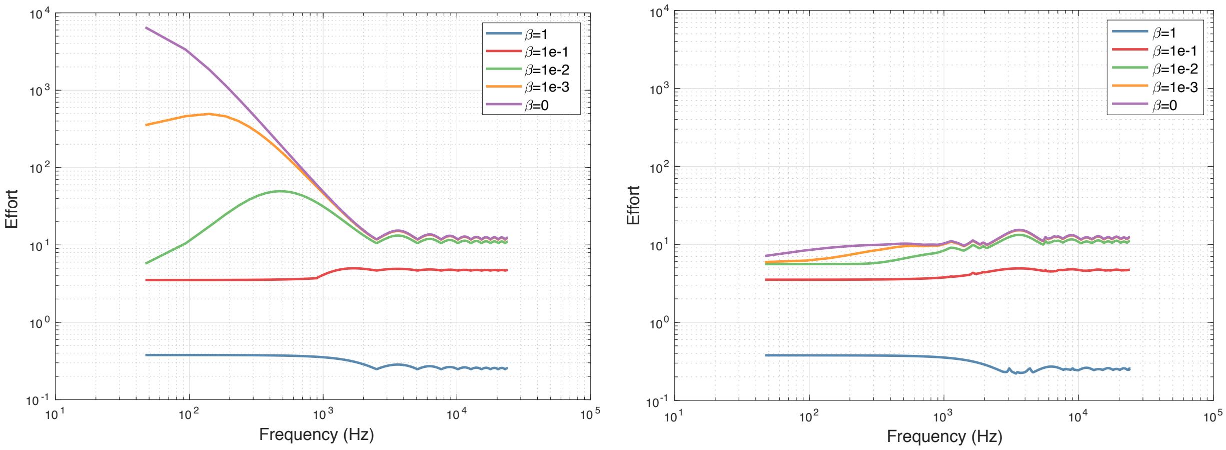


(a) Low frequency HRTF.   **(b)** Low-mid HRTF.   **(c)** High-mid HRTF.

Figure 3.3: Every azimuthal spherical head model HRTF [1] mapped to unit circle at various frequencies. Notice how only a small subset of interaural phase angles and magnitude ratios are achievable with binaural audio.

To notice constraints on the types of signals that can occur in binaural audio leads to a more specific question about EM and CTC: what is the worst-case effort for binaural signals? The answer is shown in Figure 3.4, which graphs worst-case effort over frequency for both general stereo and binaural signals. Subfigure **(a)** shows the worst-case effort for general stereo; represented by all points in the unit circle. Subfigure **(b)** on the other hand, shows the worst-case effort for all possible binaural signals (those represented by Figure 3.3). Furthermore, these graphs show how worst-case effort responds to different amounts of regularization.

Examining Figure 3.4 leads to several observations about the worst-case array effort of binaural signals. First, worst-case effort at ITD-centric frequencies is several orders of magnitude lower for binaural audio than stereo audio. Second, worst-case binaural effort is considerably flatter across the spectrum than general stereo. Together, these observations mean that when the input is binaural, low-frequencies need less EM than when the input is general stereo. This creates a problem for spatial filters that may be used with both types of audio. Specifically, the amount of EM that's necessary for general stereo will introduce more error into binaural signals than is necessary to constrain effort. To learn more about this, the next section introduces a method for measuring both how much and what kind of error is introduced by EM.



(a) Worst-case stereo effort.

(b) Worst-case binaural effort.

Figure 3.4: Worst-case effort for stereo and binaural crosstalk cancellation using various amounts of Tikhonov regularization.

3.4 The effect of effort management on signal error

In the previous section, stereo input signals were connected to array effort in order to build intuition about their relationship. The connection between input and effort is represented by the first half of the PMM signal flow graph (Eq. 3.11).

$$\overbrace{\boldsymbol{s} \to X \to \boldsymbol{w}}^{\boxed{\boldsymbol{w} = X\boldsymbol{s}}} \underbrace{\to A \to \hat{\boldsymbol{s}}}_{\hat{\boldsymbol{s}} = A\boldsymbol{w}}$$

It was shown that the interchannel phase angle and magnitude difference uniquely determine array effort for the set of normalized input signals.

In this section, stereo input signals are connected to the error introduced by EM. The connection between input and error is represented by the second half of the PMM signal flow.

$$\overbrace{\boldsymbol{s} \to X \to \boldsymbol{w}}^{\boldsymbol{w} = X\boldsymbol{s}} \underbrace{\to A \to \hat{\boldsymbol{s}}}_{\boxed{\hat{\boldsymbol{s}} = A\boldsymbol{w}}}$$

The goal of this section is to show that the entire PMM signal flow can be reduced to an input signal transformation, where $\boldsymbol{s} \neq \hat{\boldsymbol{s}}$ can only occur because of EM.

3.4.1 PMM as a signal transformation

The entire PMM system can be reduced to a map from input to output. This can be seen by combining both halves of the PMM signal flow ($\hat{\boldsymbol{s}} = A\boldsymbol{w}$ and $\boldsymbol{w} = X\boldsymbol{s}$) through substitution of their common factor $\boldsymbol{w}$.

$$\hat{\boldsymbol{s}} = AX\boldsymbol{s} \tag{3.16}$$

This expression states that the input signal $\boldsymbol{s}$ is transformed, first by the PMM convolution matrix, then second by the propagation matrix, yielding the output signal $\hat{\boldsymbol{s}}$. And because earlier $X = A^{+}$ was defined, filtering and propagation can also be described by the expression

$$\begin{aligned} \hat{\boldsymbol{s}} &= AX\boldsymbol{s} \\ &= AA^{+}\boldsymbol{s} \\ &= \hat{I}\boldsymbol{s} \end{aligned} \tag{3.17}$$

So, because $\hat{I} = AX^{+}$, the composition $\hat{I}$ represents transformations undergone by the input signal on its way to becoming an output signal.

$$\hat{\boldsymbol{s}} = \hat{I}\boldsymbol{s} \tag{3.18}$$

This expression represents a core objective of PMM: to generate filters that map input signals to control point responses through a sound field with as little change as possible[7]. This section leverages Eq. 3.18 , and in particular the idea of $\hat{I}$, to analyze effort management strategies by converting the error they introduce into signal transformations.

[7] As an aside, Eq. 3.18 serves to support Eq. 3.4 , since it would fail if $X \neq A^{+}$.

Tikhonov regularization as a signal transformation

Tikhonov regularization is substituted for A^+ in Eq. 3.18 as an example of how to use it to analyze the PMM signal transformation. First,

$$X = A^H \left(AA^H + \beta I\right)^{-1} . \tag{3.19}$$

Plugging the result in for X gives

$$\hat{\boldsymbol{s}} = AA^H \left(AA^H + \beta I\right)^{-1} \boldsymbol{s} . \tag{3.20}$$

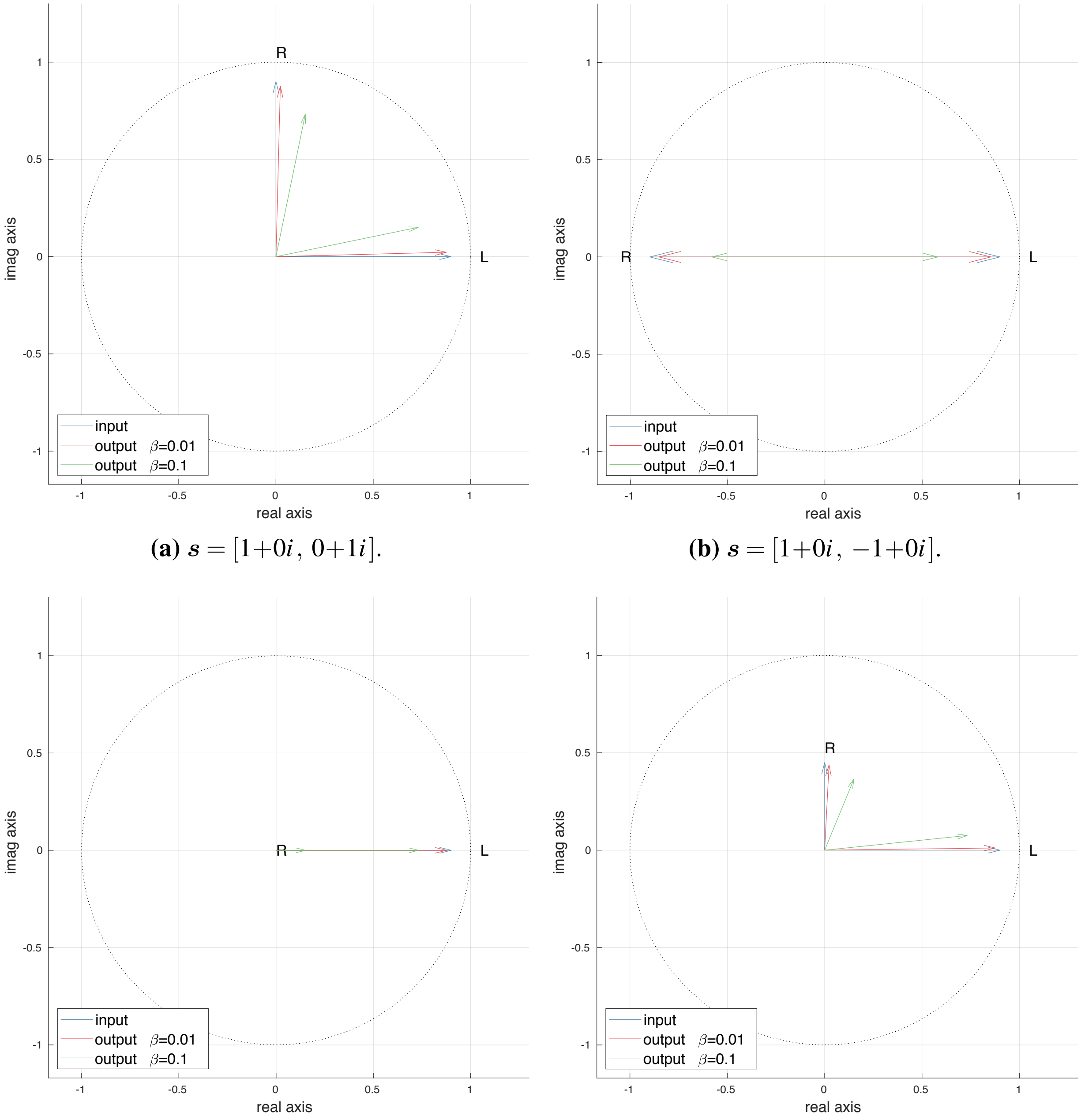
It's possible to see that when $\beta = 0$, the system reduces to $\hat{\boldsymbol{s}} = \boldsymbol{s}$.

$$\begin{aligned} \hat{\boldsymbol{s}} &= AA^H \left(AA^H + 0I\right)^{-1} \boldsymbol{s} \\ \hat{\boldsymbol{s}} &= AA^H \left(AA^H\right)^{-1} \boldsymbol{s} \\ \hat{\boldsymbol{s}} &= I\boldsymbol{s} \\ \hat{\boldsymbol{s}} &= \boldsymbol{s} \end{aligned} \tag{3.21}$$

Conversely, as β increases, error is increased.

$$\begin{aligned} \hat{\boldsymbol{s}} &= AA^H \left(AA^H + \beta I\right)^{-1} \boldsymbol{s} \\ \hat{\boldsymbol{s}} &= \hat{I}\boldsymbol{s} \\ \hat{\boldsymbol{s}} &\neq \boldsymbol{s} \end{aligned} \tag{3.22}$$

The transformation that regularization imparts on input signals is expressed by the composition $\hat{I} \in \mathbb{C}^{2\times 2}$. So geometrically, regularization scales and (possibly) rotates $\boldsymbol{s}$ to produce $\hat{\boldsymbol{s}}$. Some examples of these transformations are shown in Figure 3.5. They are of stereo signals at 500 Hz with various interchannel relationships, for near field listeners, both on- and off-axis.



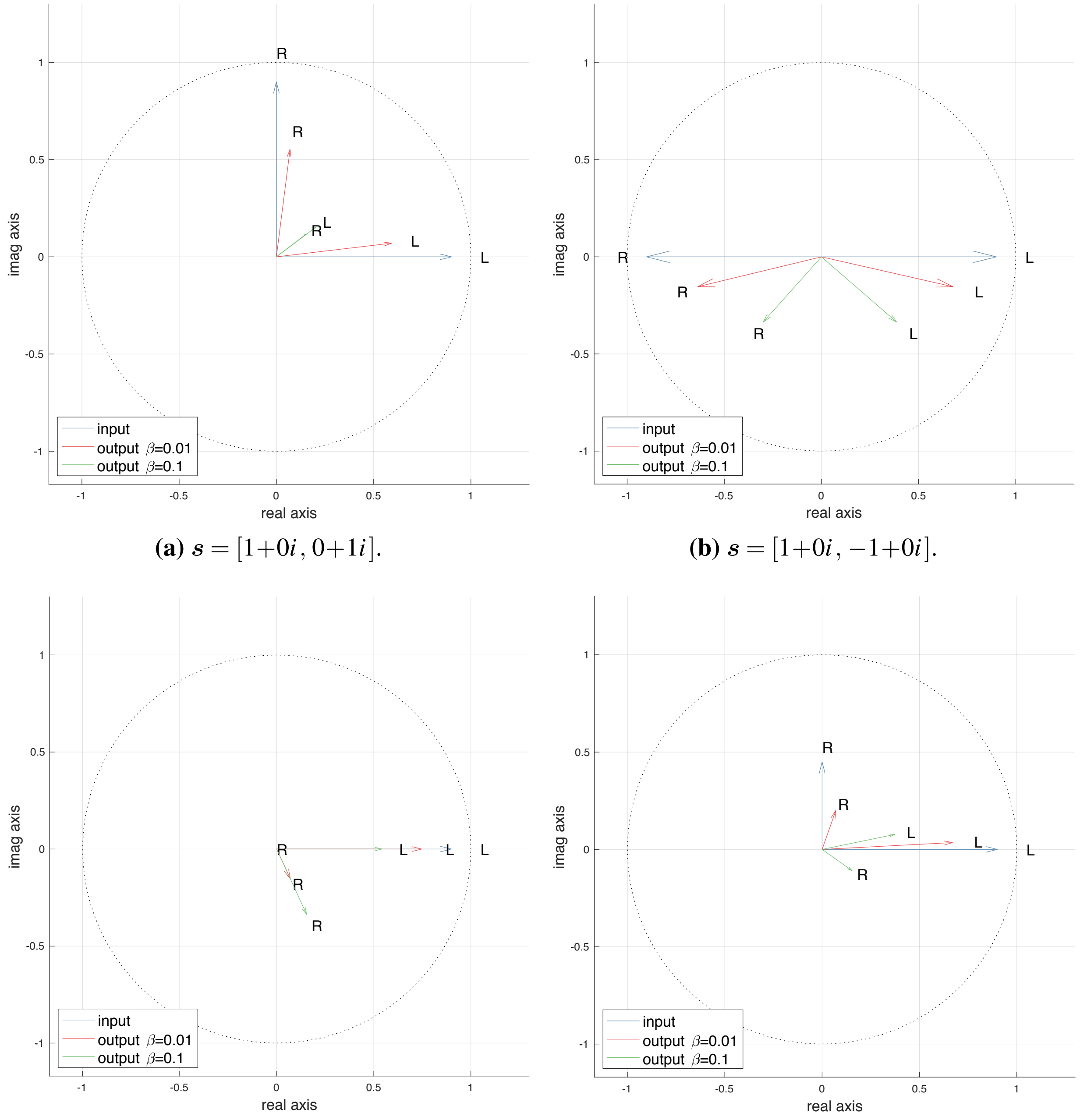
(a) $\boldsymbol{s} = [1{+}0i,\ 0{+}1i]$.

(b) $\boldsymbol{s} = [1{+}0i,\ -1{+}0i]$.

(c) $\boldsymbol{s} = [1{+}0i,\ 0{+}0i]$.

(d) $\boldsymbol{s} = [1{+}0i,\ 0.5{+}1i]$.

Figure 3.5: Error introduced by Tikhonov regularization for *on-axis* listeners at various frequencies. Left and right channels are labeled “L” and “R”. The output signal for $\beta = 0$ is not shown because it is identical to the input signal.



(a) $s = [1+0i\,,\, 0+1i]$.

(b) $s = [1+0i\,,\, -1+0i]$.

(c) $s = [1+0i\,,\, 0+0i]$.

(d) $s = [1+0i\,,\, 0.5+1i]$.

Figure 3.6: Error introduced by Tikhonov regularization for off-axis listeners at various frequencies. Left and right channels are labeled "L" and "R". The output signal for $\beta = 0$ is not shown because it is identical to the input signal.

3.4.2 Observations about the effect of regularization on error

Several observations can be made about the transformations $\hat{I}$ (shown in Figures 3.5 and 3.6). First, it can be seen that as regularization increases, both interchannel phase angle and magnitude difference decrease together. These artifacts could also be produced by classical mid-side (M/S) decomposition; specifically, by attenuating the side component of a M/S-decomposed stereo signal. Together, both of these artifacts produce a major impact on binaural audio perception, called stage compression by Fazi and Hamdan [9].

Second, EM transformations are only symmetric when the system is symmetric; that is, only for on-axis listeners. In addition to creating asymmetric transformations, the degree of error introduced by regularizing filters for off-axis listeners is higher. As the listening position moves increasingly off-axis, the interchannel phase angle is eventually inverted which, in turn, reverses the spatial image of that signal. At least in the case of stereo signals that carry localization cues, asymmetric artifacts are significantly detrimental to perceived sound quality.

Finally, since PMM filters are LTI, stage compression is time-invariant by definition. This means that even binaural signals which can be reproduced within a given effort constraint are compressed by conventional EM in order to compensate for worst-case scenarios.

3.5 Conclusion

The significant differences between worst-case effort caused by stereo versus binaural signals, combined with the fact that error introduced by EM is time-invariant, serve together as the primary motivation for ‘effort compression’; introduced in the next chapter.

In section 3.1, it was shown that PMM can be reformulated to generate the entire convolution matrix in a single expression, rather than in several disjointed expressions. Then, an expression of the multi-channel PMM signal flow — one which includes the input, output, and array signals — was presented in section 3.2. In section 3.3, the PMM signal flow graph was used to establish new SFC metrics ‘array effort’ and ‘input signal error’. Finally, section 3.4 connects ideas from the previous sections to show that output signals, which are normally created by baking EM directly into spatial filters, can also be created by applying EM transformations to the input signals directly.

Some EM techniques may be simpler to express as signal transformations rather than as filter transformations. In section 3.4.1 for example, the $\hat{I}$ of Tikhonov regularization causes a reduced interchannel phase angle and magnitude difference, which is in turn causes stage compression. It was suggested that the same $\hat{I}$ could also be produced by a signal transformation; specifically, the interchannel phase angle and magnitude difference can be reduced together by attenuating the side component of a M/S-decomposed stereo signal. From this, one can gather that different types of EM transformations will produce different types of artifacts on stereo signals, which may be more or less detrimental to *perceived* audio quality. So the choice of which EM technique to use, assuming they are more or less equally able to manage effort, should be made in order to minimize perceived error rather than numerical error.

Chapter 4

Signal-dependent effort management

This chapter contains a discussion about, as well as several examples of, signal-dependent EM or ‘effort compression’. In previous chapters, the limitations of and artifacts caused by LTI EM were discussed. Now, the multi-channel and signal-dependent frameworks previously developed will serve as the basis for the derivation of effort compression.

This chapter assumes that the input signal is streamed in blocks of samples as it is for real-time systems like digital audio workstation (DAW) or patcher plugins. Furthermore, it assumes that each block has been transformed into the frequency domain via DFT. Unless otherwise noted, references to implementation details assume this type of frequency-domain, block-wise signal processing architecture.

In section 4.1, the expression (Eq. 3.13) that relates stereo input signals to array effort, and, that effort compression relies on, is simplified. In section 4.2, several examples of effort compression methods are proposed. In section 4.3, the proposed effort compression methods are analyzed and compared.

4.1 Expressing multi-channel array effort

This section aims to formalize the relationship between multi-channel, signal-dependent effort $K(\boldsymbol{w})$ and components of the PMM system that can be changed, for example the input signal (through signal processing) or the singular values of a filter matrix (through regularization). The expressions derived here are the basis for signal-dependent EM, presented in the next section. Ultimately, the goal of this section is to find variables in the PMM system that offer filter designers direct and meaningful control over array effort and signal error.

Recall PMM's multi-channel formulation for generating array signals.

$$\boldsymbol{w} = X\boldsymbol{s}$$

The convolution matrix X, which is also the pseudoinverse of the propagation matrix A^+, can be expressed in terms of the SVD of A.

$$\begin{aligned} \boldsymbol{w} &= X\boldsymbol{s} \\ \boldsymbol{w} &= A^+\boldsymbol{s} \\ \boldsymbol{w} &= V\Sigma^+U^H\boldsymbol{s} \end{aligned} \tag{4.1}$$

SVD is useful because it gives direct access to the singular values of X, which in turn makes it easy to anticipate the PMM metrics: condition, effort, and error. In this section, it's used to reveal the dependence of array effort on input signal and singular values.

Starting with the expression for array effort (from Eq. 3.13),

$$\begin{aligned} K(\boldsymbol{w}) &= \|\boldsymbol{w}\|_2^2 \\ K(\boldsymbol{w}) &= \|X\boldsymbol{s}\|_2^2 \end{aligned}$$

the filter matrix X is substituted for its SVD terms. Simplifying gives

$$\begin{aligned} K(\boldsymbol{w}) &= \left\| V\Sigma^{+}U^{H}\boldsymbol{s}\right\|_2^2 \\ &= \left(V\Sigma^{+}U^{H}\boldsymbol{s}\right)^{H}\left(V\Sigma^{+}U^{H}\boldsymbol{s}\right) \\ &= \boldsymbol{s}^{H}U\left(\Sigma^{+}\right)^{H}V^{H}V\Sigma^{+}U^{H}\boldsymbol{s} \\ &= \boldsymbol{s}^{H}U\left(\Sigma^{+}\right)^{H}\Sigma^{+}U^{H}\boldsymbol{s}\,. \end{aligned} \tag{4.2}$$

At this point it becomes difficult to simplify further using matrix notation. However, the symmetry of the bottom equation is recognizable as the quadratic form. Assigning temporary variables to these terms will help expose the quadratic form more clearly.

Combining the left and right hand sides of the equation creates two 2×1 vectors that are related by conjugate transpose. Call them $\boldsymbol{a}$ and $\boldsymbol{a}^H$. The rest will combine to make a 2×2 matrix. Call it B. So the result of Eq. 4.2 can be rewritten

$$\begin{aligned} K(\boldsymbol{w}) &= \underbrace{\boldsymbol{s}^{H}U}_{\boldsymbol{a}^{H}}\underbrace{\left(\Sigma\Sigma^{H}\right)^{+}}_{B}\underbrace{U^{H}\boldsymbol{s}}_{\boldsymbol{a}} \\ &= \boldsymbol{a}^{H}\boldsymbol{B}\boldsymbol{a}\,. \end{aligned} \tag{4.3}$$

As a sanity check, it makes sense that effort can be rewritten in the quadratic form because it is well-known that least squares objective functions are quadratic [29].

What potentially exploitable properties do $\boldsymbol{a}$ and B have? The vector $\boldsymbol{a}$ is the combination of right eigenvalues and the input signal. Is that okay? Well, columns of U are linearly-independent, orthonormal vectors. This means that the transformation U is just a change of basis; any transformations to $\boldsymbol{a}$ are congruent to transformations of $\boldsymbol{s}$ directly. And in any case, the transformation

$$\boldsymbol{a} = U^{H}\boldsymbol{s} \tag{4.4}$$

can be easily undone with the inverse

$$\boldsymbol{s} = U\boldsymbol{a} \tag{4.5}$$

because

$$U^H = U^+ . \tag{4.6}$$

So it's okay to leave $\boldsymbol{a}$ alone for now since the input signal can always be extracted from it later.

Continuing on, if A is well-conditioned[1], then B is guaranteed to be a diagonal matrix with real, positive values. With that in mind, it's time to abandon matrix notation and directly reveal the elements of $\boldsymbol{a}$ and B. The diagonal values of matrix B are

$$\begin{aligned} B &= \left(\Sigma\Sigma^H\right)^+ \\ &= \left(\begin{bmatrix} \sigma_1 & \\ & \sigma_2 \end{bmatrix} \begin{bmatrix} \sigma_1 & \\ & \sigma_2 \end{bmatrix}^H\right)^+ \\ &= \begin{bmatrix} \sigma_1^2 & \\ & \sigma_2^2 \end{bmatrix}^+ \\ &= \begin{bmatrix} \frac{1}{\sigma_1^2} & \\ & \frac{1}{\sigma_2^2} \end{bmatrix} . \end{aligned} \tag{4.7}$$

Furthermore, the values of $\boldsymbol{a}$ are

$$\begin{aligned} \boldsymbol{a} &= U^H\boldsymbol{s} \\ &= \begin{bmatrix} u_{11}^* & u_{21}^* \\ u_{12}^* & u_{22}^* \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \\ &= \begin{bmatrix} u_{11}^* s_1 + u_{21}^* s_2 \\ u_{12}^* s_1 + u_{22}^* s_2 \end{bmatrix} \end{aligned} \tag{4.8}$$

[1] See Prependix

which for notational clarity, call

$$\begin{aligned}\boldsymbol{a} &= \begin{bmatrix} u_{11}^* s_1 + u_{21}^* s_2 \\ u_{12}^* s_1 + u_{22}^* s_2 \end{bmatrix} \\ &= \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}\end{aligned} \tag{4.9}$$

So the expression for array effort in terms of the SVD of X and a stereo input signal can be written

$$\begin{aligned}K(\boldsymbol{w}) &= \boldsymbol{a}^H B \boldsymbol{a} \\ &= \begin{bmatrix} a_1^* & a_2^* \end{bmatrix} \begin{bmatrix} \frac{1}{\sigma_1^2} & \\ & \frac{1}{\sigma_2^2} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \\ &= \frac{a_1^* a_1}{\sigma_1^2} + \frac{a_2^* a_2}{\sigma_2^2} \\ &= \frac{|a_1|^2}{\sigma_1^2} + \frac{|a_2|^2}{\sigma_2^2}.\end{aligned} \tag{4.10}$$

Which means that when $M = 2$,

$$\boxed{K(\boldsymbol{w}) = \frac{|a_1|^2}{\sigma_1^2} + \frac{|a_2|^2}{\sigma_2^2}} \tag{4.11}$$

The expression Eq. 4.11 is recognizable as the function of an ellipse. In other words, as either the input signals or the SVD of A change (especially the singular values of X, but also the matrix U), array effort responds deterministically. This is useful because it means that several different strategies for transforming either input signals or singular values will have deterministic, and potentially analytical relationships to array effort.

4.2 Methods for effort compression

Eq. 4.11 shows that array effort depends on two potentially time-variant SFC components: singular values of the propagation matrix and input signals. Furthermore, it shows that the geometry of the relationship between effort and these components is an ellipse. With that in mind, several signal or filter transformations could be used to constrain array effort. This section presents some examples of effort compression. The list of examples presented here is far from exhaustive; in fact, the suggestion of signal-dependent EM itself presents a considerable amount of future work.

The choice of *which* method for effort compression to use is application-dependent at best, or at worst, completely subjective. In the following sections, the presentation of each effort compression method will include some information about potentially important implementation details. The implementation details considered in this work include: first whether a method is mathematically tractable to implement in the time domain, second, whether it is computationally tractable to implement in real-time systems, and third, a description of the type of error it adds to a signal.

Regarding tractability: in terms of mathematics, some effort compression methods may or may not be trivial to implement in the time domain. In terms of computation, other methods may not yet be tractable at large scales — even offline. Therefore, what this section presents is the notion that signal-dependent EM is possible, and, that deriving signal transformations from filter transformations relies on Eq. 4.11.

Regarding error: ideally, the character of the error introduced by regularization should be non-obvious to a listener. Less generally, in the case of binaural audio, a good place to introduce error is on signal components that do not contribute highly to localization. This point relates to future work, discussed in chapter 5.

Implementing frequency-domain effort compression

The solution to an PMM problem is an FIR filter set if control point responses are specified as impulses. Furthermore, convolution in the time domain is equivalent to multiplication in the frequency domain; the latter being exponentially cheaper to compute. So conventional PMM filters are both solved and implemented in the frequency domain. A typical real-time implementation will operate on signal blocks, transform them into the frequency domain, multiply them by the convolution matrix, transform the combination back into the time domain, and finally, send the result to the output.

Signal-dependent effort compression has two possible implementation strategies in the frequency domain. First, EM is baked into filters. This is the case for regularized EM. A real-time, signal-dependent implementation will have to generate a new convolution matrix every block based on the current input. Second EM is decoupled from filters. This is the case for signal transformations. A real-time, signal-dependent implementation can use a single convolution matrix (or its SVD factors) which should be generated beforehand and loaded from memory as an initialization step. In either case, the expression for real-time, frequency domain EM is written

$$\underset{(M\times 1)}{\hat{\boldsymbol{s}}} = \underset{(N\times M)}{X}\underset{(M\times M)}{C}\underset{(M\times 1)}{\boldsymbol{s}}, \tag{4.12}$$

where C is a $M \times M$ matrix that optionally transforms the input signal. Matrix C should have EM transfer functions for each input channel along its diagonal.

Whenever effort compression is either nonlinear or time-variant, and is implemented in the frequency domain, it is necessary to correct artifacts that are created by these types of transformations in the frequency domain [40]. To correct them, the input window (which is normally applied to a signal *before* the DFT) should be the square root of whatever it normally is. An output window (that is additionally applied to a signal *after* the iDFT) should also be used. The input and output windows should be identical.

Implementing time-domain effort compression

When EM is decoupled from filters, it may be possible to implement as a time domain transformation rather than a frequency domain one. Time domain signal transformations benefit from being able to smooth out nonlinear artifacts over time at sample rate, which is not possible in the frequency domain. However, it is difficult to reproduce many types of frequency domain transformations in the time domain; including several transformations used by PMM. In some cases, a solution that's analogous to the standard audio engineering plugin 'multi-band compression' could be used: subdivide a time domain signal into several (e.g. 3 or 4) frequency bands every block. The maximum unconstrained effort of each band is then found by frequency domain analysis. Then, the EM transformation that constrains the worst-case frequency in each band is applied in the time domain. Other multi-band compression parameters, for example ratio, knee, attack, and decay, could be applied to some types of EM transformations.

4.2.1 Effort compression: SV-clipping

As was shown in chapter 2, regularization is a method for conditioning, or, decreasing the condition of matrix A. Conditioning a matrix serves in turn to decrease filter effort. Tikhonov regularization, as one specific type of regularization, conditions a matrix by increasing each of its singular values by β. More generally, conditioning a matrix is accomplished by any strategy that decreases the ratio between $\sigma_{\max}$ and $\sigma_{\min}$ [2].

In reaction to this perspective, Afghah et al. [10] propose the eigenvalue-clipping pseudoinverse method as an extension to classical (Tikhonov regularization-based) PMM. Its authors find that a desired condition can be reached faster when only the smallest eigenvalues are increased through regularization, rather than all eigenvalues. This section introduces a similar regularization method, called SV-clipping, where SVD is used in lieu of eigendecomposition. Also in this section, SV-clipping is converted into an effort compression method.

[2] When $M = 2$, $\sigma_{\max} := \sigma_1$, and $\sigma_{\min} := \sigma_2$.

To start, recall how the condition of a matrix $\kappa(A)$ is defined by the ratio of its minimum and maximum singular values (shown in Eq. 2.40).

$$\kappa(A) := \frac{\sigma_{\text{max}}}{\sigma_{\text{min}}}$$

SV-clipping proposes to condition a matrix by adding β to σ_{min}. Using $\hat{\kappa}$ as a shorthand for desired condition of matrix A, SV-clipping is expressed

$$\hat{\kappa} = \frac{\sigma_1}{\sigma_2 + \beta} \,. \tag{4.13}$$

Then, the regularization parameter that achieves a desired condition with SV-clipping can be found analytically

$$\begin{aligned} \frac{\sigma_1}{\sigma_2 + \beta} &= \hat{\kappa} \\ \frac{\sigma_2 + \beta}{\sigma_1} &= \frac{1}{\hat{\kappa}} \\ \sigma_2 + \beta &= \frac{\sigma_1}{\hat{\kappa}} \\ \beta &= \frac{\sigma_1}{\hat{\kappa}} - \sigma_2 \end{aligned} \tag{4.14}$$

By comparing this result to the solution for Tikhonov regularization (Eq. 2.49),

$$\beta_{\text{SV-clipping}} = \frac{\sigma_1}{\hat{\kappa}} - \sigma_2 \,, \qquad \text{and} \qquad \beta_{\text{Tikhonov}} = \frac{\sigma_1 - \hat{\kappa}\sigma_2}{\hat{\kappa} - 1} \,,$$

it can be shown that SV-clipping always produces a smaller regularization parameter for a given desired condition than Tikhonov regularization[3]. This also means that SV-clipping achieves a desired condition for less error than Tikhonov regularization. Starting from the assertion that the

[3] Remember that $\sigma_1 \geq \sigma_2 \geq 0$ and $\hat{\kappa} \geq 1$.

two methods are not equal

$$\begin{aligned}
\frac{\sigma_1}{\hat{\kappa}} - \sigma_2 &\neq \frac{\sigma_1 - \hat{\kappa}\sigma_2}{\hat{\kappa}-1} \\
\frac{\sigma_1}{\hat{\kappa}} - \frac{\hat{\kappa}\sigma_2}{\hat{\kappa}} &\neq \frac{\sigma_1 - \hat{\kappa}\sigma_2}{\hat{\kappa}-1} \\
\frac{\sigma_1 - \hat{\kappa}\sigma_2}{\hat{\kappa}} &\neq \frac{\sigma_1 - \hat{\kappa}\sigma_2}{\hat{\kappa}-1} \\
\underbrace{(\sigma_1 - \hat{\kappa}\sigma_2)}_{d}\frac{1}{\hat{\kappa}} &\neq \underbrace{(\sigma_1 - \hat{\kappa}\sigma_2)}_{d}\frac{1}{\hat{\kappa}-1} \\
\frac{d}{\hat{\kappa}} &\neq \frac{d}{\hat{\kappa}-1}
\end{aligned} \tag{4.15}$$

the conditional inequality is found

$$\begin{cases} \frac{d}{\hat{\kappa}} < \frac{d}{\hat{\kappa}-1}, & \text{if } d > 0 \ ^{\ddagger} \\ \frac{d}{\hat{\kappa}} = \frac{d}{\hat{\kappa}-1}, & \text{if } d = 0 \\ \frac{d}{\hat{\kappa}} > \frac{d}{\hat{\kappa}-1}, & \text{if } d < 0 \end{cases} \tag{4.16}$$

which states that when the placeholder variable $d > 0$, the regularization parameter produced by the SV-clipping method is guaranteed to be smaller than the one produced by Tikhonov regularization. Regularization is only needed when $d > 0$ (the case for which is marked by the symbol ‡) since the alternative cases only occur when the desired condition is equal to or greater than the actual condition.

The amount of SV-clipping needed to achieve a desired array effort can also be found analytically. For notational clarity, the value $\hat{\sigma}_2$ is introduced, where

$$\hat{\sigma}_2 = \sigma_2 + \beta \,. \tag{4.17}$$

Starting from the signal-dependent equation for array effort (Eq. 4.11), the smallest singular value

$\hat{\sigma}_2$ that achieves desired array effort $\hat{K}(\boldsymbol{w})$ is

$$\begin{aligned}
\frac{|a_1|^2}{\sigma_1^2} + \frac{|a_2|^2}{\hat{\sigma}_2^2} &= \hat{K}(\boldsymbol{w}) \\
\frac{|a_2|^2}{\hat{\sigma}_2^2} &= \hat{K}(\boldsymbol{w}) - \frac{|a_1|^2}{\sigma_1^2} \\
\frac{1}{\hat{\sigma}_2^2} &= \frac{\hat{K}(\boldsymbol{w}) - \frac{|a_1|^2}{\sigma_1^2}}{|a_2|^2} \\
\hat{\sigma}_2^2 &= \frac{|a_2|^2}{\hat{K}(\boldsymbol{w}) - \frac{|a_1|^2}{\sigma_1^2}} \\
\hat{\sigma}_2 &= \sqrt{\frac{|a_2|^2}{\hat{K}(\boldsymbol{w}) - \frac{|a_1|^2}{\sigma_1^2}}} .
\end{aligned} \tag{4.18}$$

Signal dependent SV-clipping regularization, then, is implemented by setting the smallest singular value of A such that

$$\boxed{\hat{\sigma}_2 = \begin{cases} \sigma_2\,, & \text{if } K(\boldsymbol{w}) <= \hat{K}(\boldsymbol{w}) \\ \sqrt{\frac{|a_2|^2}{\hat{K}(\boldsymbol{w}) - \frac{|a_1|^2}{\sigma_1^2}}} & \text{else} \end{cases}} \tag{4.19}$$

To implement SV-clipping in terms of the signal-dependent expression Eq. 4.12, the modified propagation matrix X should be generated each block

$$X = V\hat{\Sigma}^{+}U^{H}\,, \tag{4.20}$$

where

$$\hat{\Sigma} = \begin{bmatrix} \sigma_1 & \\ & \hat{\sigma}_2 \end{bmatrix}, \quad \text{and} \quad \hat{\Sigma}^{+} = \begin{bmatrix} \frac{1}{\sigma_1} & \\ & \frac{1}{\hat{\sigma}_2} \end{bmatrix}. \tag{4.21}$$

By performing this substitution every DFT block, array effort is clipped to a maximum value of $\hat{K}(\boldsymbol{w})$. In SV-clipping, the input transformation matrix $C = I$. Also, notice that the only value changing each block is $\hat{\sigma}_2$, which means the SVD of A can be computed or loaded as an initialization step before the real-time sequence. Finally, SV-clipping is potentially tractable to implement in real-time, but only in the frequency domain.

4.2.2 Effort compression: input signal attenuation

Input signal attenuation allows a filter designer to manage effort without incurring asymmetric or phase-related artifacts. It is also tractable to implement both in the time domain and in real-time. Since the numerator of the ellipsoidal effort function Eq. 4.11 is the input signal, scaling effort is proportional to scaling the signal.

The first step to finding the input audio gain scalar that achieves desired effort is to define how effort will be changed by changing the gain of the input. Any desired effort $\hat{K}(\boldsymbol{w})$ can be achieved by changing the input signal gain, notated g.

$$\hat{K}(\boldsymbol{w}) = \|X(g\boldsymbol{s})\|_2^2 \tag{4.22}$$

Next, to solve for g, define a scalar q that maps actual effort to desired effort.

$$\hat{K}(\boldsymbol{w}) = qK(\boldsymbol{w}) \tag{4.23}$$

Solving for q gives

$$q = \frac{\hat{K}(\boldsymbol{w})}{K(\boldsymbol{w})}. \tag{4.24}$$

Then, relating q to the gain term g sets up an analytical solution to

$$\|X(g\boldsymbol{s})\|_2^2 = q\,\|X\boldsymbol{s}\|_2^2, \tag{4.25}$$

where solving for g gives

$$\begin{aligned}
\|X(g\boldsymbol{s})\|_2^2 &= q\,\|X\boldsymbol{s}\|_2^2 \\
(X(g\boldsymbol{s}))^H(X(g\boldsymbol{s})) &= q(X\boldsymbol{s})^H(X\boldsymbol{s}) \\
(g\boldsymbol{s})^H X^H X(g\boldsymbol{s}) &= q\boldsymbol{s}^H X^H X\boldsymbol{s} \\
g^2\big(\boldsymbol{s}^H X^H X\boldsymbol{s}\big) &= q\big(\boldsymbol{s}^H X^H X\boldsymbol{s}\big) \\
g^2 &= q \\
g^2 &= \frac{\hat{K}(\boldsymbol{w})}{K(\boldsymbol{w})} \\
g &= \sqrt{\frac{\hat{K}(\boldsymbol{w})}{K(\boldsymbol{w})}}\,.
\end{aligned} \tag{4.26}$$

So the gain g that achieves desired effort when applied to both channels is

$$\boxed{g = \sqrt{\frac{\hat{K}(\boldsymbol{w})}{K(\boldsymbol{w})}}} \tag{4.27}$$

The input signal gain g should be non-negative to avoid inverting phase (which is important if this method is used with other loudspeakers, like a subwoofer). So disregard the negative solution implied by taking the square root. Then to implement this method in terms of the signal-dependent expression Eq. 4.12, the input signal transformation matrix C should be generated each block

$$C = \begin{bmatrix} g & \\ & g \end{bmatrix} \tag{4.28}$$

Alternatively, this method could be implemented in the time domain, where gain is applied smoothly to bands, like multi-band compressors. The amount of attenuation needed to constrain the effort of each band, however, has to be found with frequency-domain analysis. That value — the minimum value of g for each band — will constrain the entire band in the time domain.

4.2.3 Effort compression: smallest signal transformation

For some applications, it may be desirable to minimize numerical error. In that case, a search for the closest signal to the input signal that satisfies effort constraints can be interpreted geometrically as a search for the shortest distance from the point described by actual effort to any point within the bounds of an ellipse that describes desired effort. This can be cast as a convex optimization problem, and therefore solved with a convex solver (e.g. CVX [41]). In standard form, this problem is notated

$$\begin{aligned} \underset{\hat{\boldsymbol{s}}}{\text{minimize}} \quad & \|\boldsymbol{s} - \hat{\boldsymbol{s}}\|_2 \\ \text{subject to} \quad & \|X\hat{\boldsymbol{s}}\|_2 \leq \sqrt{\hat{K}(\boldsymbol{w})}\,. \end{aligned} \tag{4.29}$$

Currently, convex solvers have far from fast enough implementations to solve this type of problem in real-time. Furthermore, there is no trivial transformation from the frequency to the time domain. Finally, for asymmetric problems like off-axis listening, the transformation to the signal may not be symmetric. That is, the transformations applied to the left and right channels may be different from each other.

Asymmetric transformations may be mitigated somewhat by minimizing the inf-norm rather than the 2-norm. In standard form, this problem is notated

$$\begin{aligned} \underset{\hat{\boldsymbol{s}}}{\text{minimize}} \quad & \|\boldsymbol{s} - \hat{\boldsymbol{s}}\|_\infty \\ \text{subject to} \quad & \|X\hat{\boldsymbol{s}}\|_2 \leq \sqrt{\hat{K}(\boldsymbol{w})}\,. \end{aligned} \tag{4.30}$$

The smallest signal transformation is found by casting it as a search for the closest signal that satisfies a set of constraints. As a result, these techniques generate the output signal directly, which means that in terms of the signal-dependent expression Eq. 4.12, the input signal $\boldsymbol{s}$ can simply be replaced by $\hat{\boldsymbol{s}}$ as given by either Eq. 4.29 or Eq. 4.30 every block.

4.2.4 Effort compression: side component attenuation

In chapter 3, plots showing how Tikhonov regularization transformed input signals revealed that the transformation was identical to side-component attenuation via M/S decomposition. The amount of side-component attenuation required to constrain effort can be found numerically with a convex solver.

The M/S decomposition of $\boldsymbol{s}$ is defined

$$\begin{aligned} \boldsymbol{s}_{\mathrm{M}} &= \frac{\boldsymbol{s}_{\mathrm{L}} + \boldsymbol{s}_{\mathrm{R}}}{2} &\qquad \Longleftrightarrow \qquad & \boldsymbol{s}_{\mathrm{L}} = \boldsymbol{s}_{\mathrm{M}} + \boldsymbol{s}_{\mathrm{S}} \\ \boldsymbol{s}_{\mathrm{S}} &= \frac{\boldsymbol{s}_{\mathrm{R}} - \boldsymbol{s}_{\mathrm{R}}}{2} &\qquad \Longleftrightarrow \qquad & \boldsymbol{s}_{\mathrm{R}} = \boldsymbol{s}_{\mathrm{M}} - \boldsymbol{s}_{\mathrm{S}} \end{aligned} \tag{4.31}$$

The search for a new side-component $\hat{s}_{\mathrm{S}}$ that's as close as possible to the original can be found with the standard form notation

$$\begin{aligned} \underset{\hat{\boldsymbol{s}}_{\mathrm{S}}}{\text{minimize}} \quad & \|\boldsymbol{s}_{\mathrm{S}} - \hat{\boldsymbol{s}}_{\mathrm{S}}\|_2 \\ \text{subject to} \quad & \left\| X \begin{bmatrix} \boldsymbol{s}_{\mathrm{M}} + \hat{\boldsymbol{s}}_{\mathrm{S}} \\ \boldsymbol{s}_{\mathrm{M}} - \hat{\boldsymbol{s}}_{\mathrm{S}} \end{bmatrix} \right\|_2 \leq \sqrt{\hat{K}(\boldsymbol{w})}. \end{aligned} \tag{4.32}$$

Like Eqs. 4.29 and 4.30, this technique generates the output signal directly, which means in terms of the signal-dependent expression Eq. 4.12, the input signal $\boldsymbol{s}$ can simply be replaced by $\hat{\boldsymbol{s}}$ every block, where

$$\hat{\boldsymbol{s}} = \begin{bmatrix} \boldsymbol{s}_{\mathrm{M}} + \hat{\boldsymbol{s}}_{\mathrm{S}} \\ \boldsymbol{s}_{\mathrm{M}} - \hat{\boldsymbol{s}}_{\mathrm{S}} \end{bmatrix}. \tag{4.33}$$

Unlike Eqs. 4.29 and 4.30 , it is possible to encounter infeasible solutions with this method.

4.3 Comparing methods for effort compression

Four effort compression methods are proposed in the previous section:

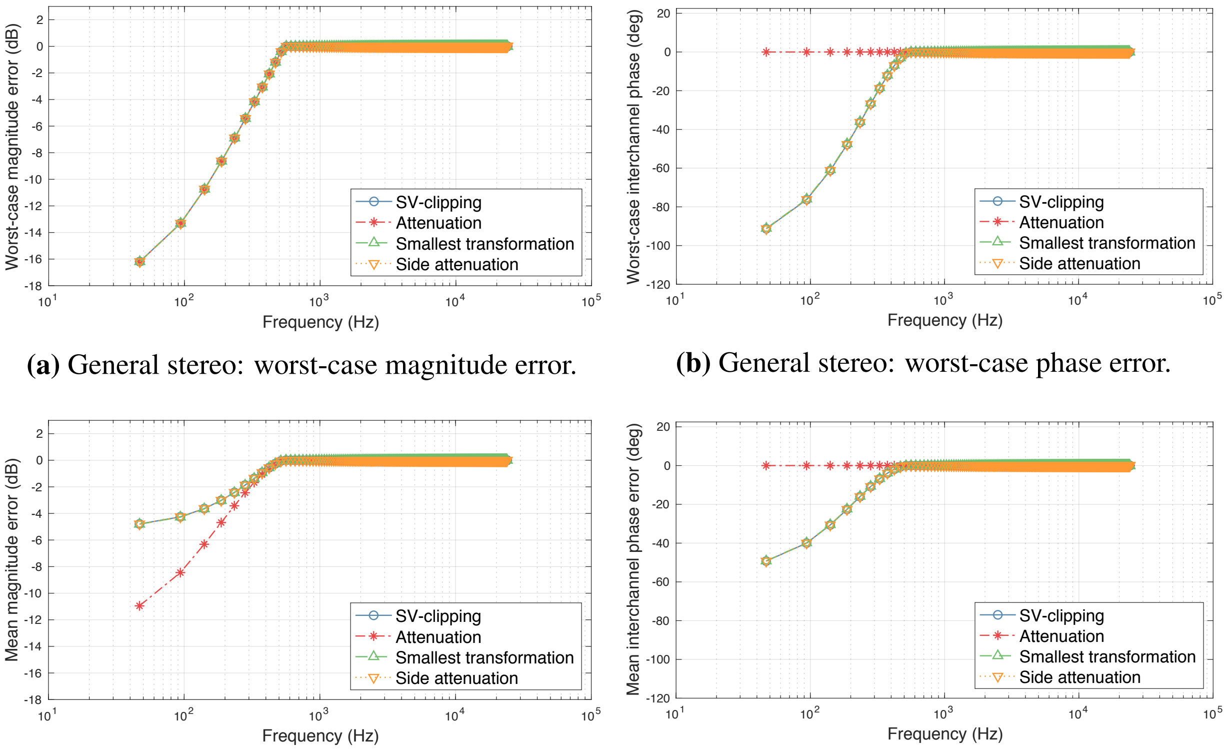
- Singular value clipping
- Input signal attenuation
- Smallest signal transformation
- Smallest side-component transformation

Of these, SV-clipping counts as signal-dependent regularization. That is, it modifies the propagation matrix, rather than the signal every block. This prevents the transformation from being trivially implemented in the time domain.

Input signal attenuation is a signal transformation, where all input channels are attenuated equally. Unlike the other effort compression methods, this preserves the interchannel phase angle, and by extension the sound stage [9] at the cost of introducing a high-passed characteristic. The input signal attenuation method is also unique in that it is trivial to transform into the time domain. Like multi-band compression, the input signal could be subdivided into frequency bands, each of which is attenuated with a gradually changing gain value in the time domain. The amount of attenuation for each band is equal to the amount of attenuation required to constrain the peak gain in that band.

The remaining two methods: smallest signal transformation and smallest side-component transformations both rely on convex solvers like CVX [41]. The results of the convex solver can be thought of as substitutions for the input signal. However, these methods are not trivial to convert to time domain transformations, nor (at the time of writing) are they tractable real-time methods.

The rest of this section analyzes and compares the proposed effort compression methods in terms of both the worst-case and mean error the introduce.



(a) General stereo: worst-case magnitude error.

(b) General stereo: worst-case phase error.

(c) General stereo: mean magnitude error.

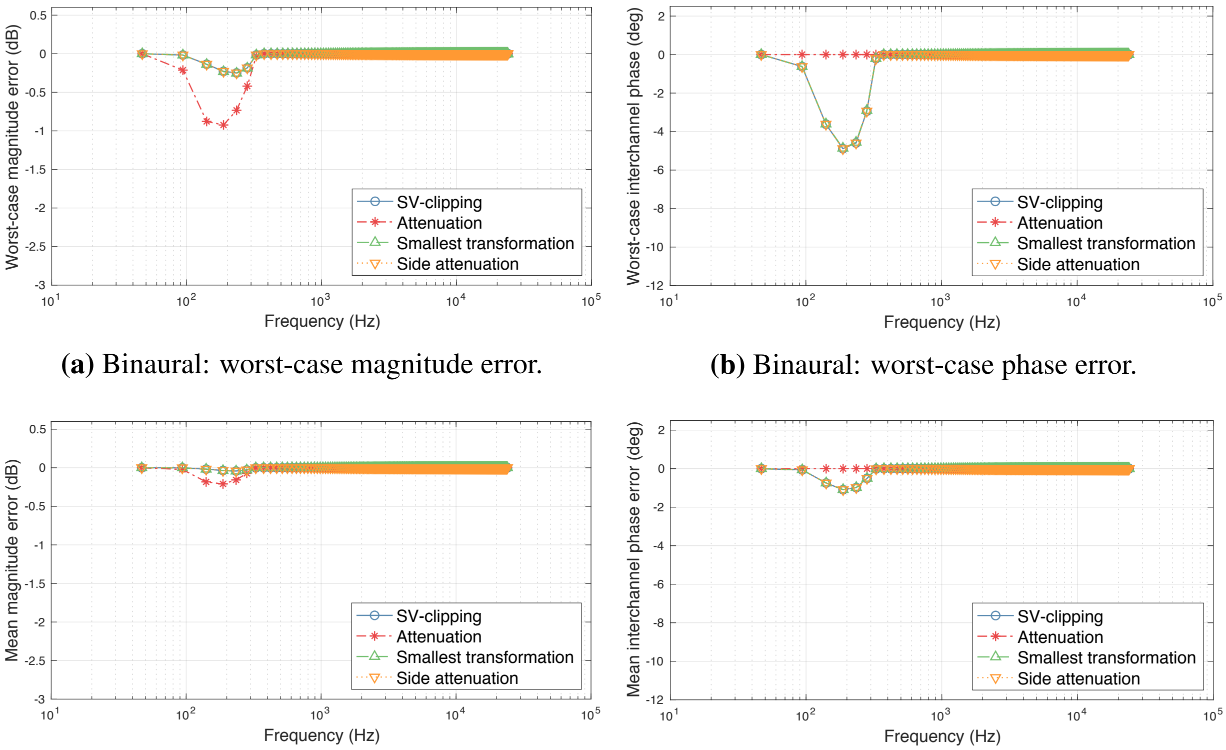
(d) General stereo: mean phase error.

Figure 4.1: When input signal is general stereo, these subfigures show the worst-case and mean magnitude and phase error introduced by the effort compression strategies presented in chapter 4.

Analyzing effort compression for general stereo input

Figure 4.1 shows the worst-case signal transformations for each of the proposed effort compression methods as applied to general stereo. Signal transformations are measured in terms of the difference between the input and output signal required to constrain array effort to a given value; in this case $\hat{K}(\boldsymbol{w}) = 1$. The worst-case magnitude difference in subfigure **(a)** shows the maximum amount of attenuation caused by effort compression. In this configuration, worst-case magnitude always occurred for the input signal $[1, -1{+}0i]^T$. The worst-case interaural phase difference in subfigure **(b)** shows the maximum amount of stage-compression in degrees caused by effort compression.

Then, subfigures **(c)** and **(d)** show the average amount of magnitude or phase error



(a) Binaural: worst-case magnitude error.

(b) Binaural: worst-case phase error.

(c) Binaural: mean magnitude error.

(d) Binaural: mean phase error.

Figure 4.2: When input signal is binaural, these subfigures show the worst-case and mean magnitude and phase error introduced by the effort compression strategies presented in chapter 4.

distributed across the set of general stereo signals. It can be seen that all methods, except the input signal attenuation method, produce the same results. This does not hold true for off-axis listener configurations. For off-axis configurations on the other hand, the results of each method differ. Finally, for all methods and scenarios, EM is not needed for general stereo above approximately 800 Hz.

Analyzing effort compression for binaural input

Figure 4.2 shows the worst-case signal transformations for each of the proposed effort compression methods as applied to binaural audio. The most important thing to notice about Figure 4.2 is that the scale of error is reduced by roughly an order of magnitude across the board. Again, the input signals are normalized and the desired effort constraint is $\hat{K}(\boldsymbol{w}) = 1$. Another improvement over general stereo, EM is not needed for binaural above approximately 400 Hz, rather than approximately 800 Hz.

4.4 Conclusion

This chapter introduced the concept of, as well as several implementations of effort compression for PMM. An expression (Eq. 4.11) that relates the input signal, as well as the singular values of the propagation matrix to array effort is shown in section 4.1. This expression is the means by which stereo input signal transformations can be framed in terms of their effect on array effort. In section 4.2, four examples of effort compression methods are presented: SV-clipping, input signal attenuation, smallest signal transformation, and smallest side-component transformation. Finally, in section 4.3, these methods are analyzed in terms of the worst-case and mean magnitude and phase error they introduce. It was also shown that the SV-clipping method actually produces the same signal transformation as the CVX-based smallest signal transformation method, which makes it an interesting alternative to signal-dependent Tikhonov regularization.

Chapter 5

Summary and future work

5.1 Summary

This work has focused on cross-talk cancellation (CTC) beamforming for stereo (especially binaural) audio. It was shown that, for a particular sort of configuration, forming independent low-frequency beams at a listener's ears presents several problems for which no perfect solution exists. This manifests as an unavoidable trade-off between sound quality and reproducibility. The vein of research that attempts to navigate this trade-off is called effort management (EM). While previous EM research has tended to focus on spatial filter transformations, this work proposes instead to transform input signals directly. This is called effort compression. The most difficult problem to overcome in the development of effort compression is that it's necessary to know ahead of time how much a given transformation will reduce array effort. In the case of stereo audio, the relationship between array effort, input signals, and the propagation matrix form an ellipse. As a consequence, several effort compression strategies are immediately obvious, and have analytic or numeric solutions.

5.2 Future work: minimizing perceived error from effort compression

The effort compression strategies presented in chapter 4 are examples; more effort compression strategies could be discovered in the future. Since effort compression is cast as an input signal transformation, it's conceptually feasible that the types of artifacts caused by effort compression could be directly controlled. Therefore, it may be possible to design and select effort compression strategies in terms of their ability to preserve qualities of audio that may be considered especially important to a filter designer. Also, other types of signal transformations could be used together with effort compression to counteract artifacts caused by it.

Preserving localization cues

Effort management strategies often reduce both interchannel phase angle and magnitude difference together, as was shown in chapter 3. When input audio contains localization cues, as binaural audio does for example, the reduction of phase and magnitude differences are perceived by listeners as stage compression [9]. In these cases, it would be desirable to use an effort compression strategy that preserves localization cues. This could be accomplished by constraining effort with respect to duplex theory [39]. As a brief summary, duplex theory states that azimuthal localization cues, which most strongly impact the perception of a sound stage, are frequency dependent. More specifically, low frequency azimuthal localization depends primarily on interaural time difference (ITD), while high frequency azimuthal localization, on the other hand, depends primarily on interaural level difference (ILD). Duplex theory could be leveraged in an effort compression strategy by attenuating the complimentary cue for a given frequency range: that is, attenuate ILD for low frequencies, and ITD for high frequencies. However, to do so would require the ability to selectively reduce either interchannel phase angle, or interchannel magnitude difference.

Increasing overall loudness

The maximum overall loudness of a CTC instance is limited by the peak gain of low-frequency beams that are aimed away from the listener (as was shown in Figure 1.2). There are several studio production techniques that are used to increase perceived loudness, which is distinct from numerical measures of loudness like peak gain or root mean square (RMS). For example, side-chained multi-band compression, spectral saturation, and soft-clipping are all used to increase perceived loudness in some types of pop music, specifically without increasing numerical loudness. These techniques could also be used to overcome a loudness limit caused by effort compression and CTC.

Low frequency extension

As was shown in chapter 4, low frequencies are the most strongly attenuated by effort compression. General stereo CTC, for example, has a lower limit in practice of roughly 500 Hz at least. Because of this, CTC imparts a high-passed characteristic onto audio, even if loudspeakers used in an array could reproduce frequencies lower than that limit. There are several studio production techniques that are used to create the perception of boosted low-frequencies, which could be used in this context. For example, some listeners report to hear a missing fundamental frequency when they are presented only its harmonics [42]. Synthetically reinforcing the harmonics of a low-frequency component in an audio signal could for this reason improve the lows' response.

Appendix A

Appendices

A.1 Derivation of the overdetermined least squares approximation

The least squares approximation for overdetermined systems can be derived starting from its cost function, defined

$$E(\boldsymbol{x}) = \|\boldsymbol{y} - A\boldsymbol{x}\|_2^2 \,. \tag{A.1}$$

Expanding $E(\boldsymbol{x})$ gives

$$\begin{aligned} E(\boldsymbol{x}) &= (\boldsymbol{y} - A\boldsymbol{x})^H (\boldsymbol{y} - A\boldsymbol{x}) \\ &= \boldsymbol{y}^H \boldsymbol{y} - \boldsymbol{y}^H A\boldsymbol{x} - \boldsymbol{x}^H A^H \boldsymbol{y} + \boldsymbol{x}^H A^H \boldsymbol{x} \\ &= \boldsymbol{y}^H \boldsymbol{y} - 2\boldsymbol{y}^H A\boldsymbol{x} + \boldsymbol{x}^H A^H A\boldsymbol{x} \end{aligned} \tag{A.2}$$

Taking the derivative of the last line gives

$$\frac{\partial}{\partial \boldsymbol{x}} E(\boldsymbol{x}) = -2A^H \boldsymbol{y} + 2A^H A\boldsymbol{x} \,. \tag{A.3}$$

Since for least squares, the local minimum is the only point at which the derivative is zero, set it to zero

$$\frac{\partial}{\partial \boldsymbol{x}} E(\boldsymbol{x}) = 0 \tag{A.4}$$

and, by plugging this into the above equation for $\boldsymbol{x}$, get

$$A^H A \boldsymbol{x} = A^H \boldsymbol{y}\,. \tag{A.5}$$

Then, assume that $A^H A$ is invertible. The solution is given by

$$\boldsymbol{x} = \left(A^H A\right)^{-1} A^H \boldsymbol{y}\,. \tag{A.6}$$

This is the least squares approximation of an overdetermined system.

$$\boxed{\underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{y} - A\boldsymbol{x}\|_2^2 \quad = \quad \left(A^H A\right)^{-1} A^H \boldsymbol{y}} \tag{A.7}$$

A.2 Derivation of the underdetermined least squares solution

The least squares approximation for underdetermined systems can be derived by casting it as the optimization problem

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \|\boldsymbol{x}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{y} = A\boldsymbol{x}\,. \end{aligned} \tag{A.8}$$

Minimization with constraints can be done with Lagrange multipliers. So, define the Lagrangian

$$\mathscr{L}(\boldsymbol{x}, \boldsymbol{\mu}) = \|\boldsymbol{x}\|_2^2 \,. \tag{A.9}$$

Take the derivatives of the Lagrangian

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{x}} \mathscr{L}(\boldsymbol{x}) &= 2\boldsymbol{x} - A^H \boldsymbol{\mu} \\ \frac{\partial}{\partial \boldsymbol{\mu}} \mathscr{L}(\boldsymbol{x}) &= 2\boldsymbol{x} - \boldsymbol{y} = A\boldsymbol{x} \end{aligned} \tag{A.10}$$

Again, because the only point with a derivative of zero is the global minimum, set the derivatives to zero to get

$$\boldsymbol{x} = \frac{1}{2} A^H \boldsymbol{\mu} \tag{A.11}$$

$$\boldsymbol{y} = A\boldsymbol{x} \tag{A.12}$$

Plugging the solutions for $\boldsymbol{x}$ into $\boldsymbol{y}$ gives

$$\boldsymbol{y} = \frac{1}{2} A A^H \boldsymbol{\mu}. \tag{A.13}$$

Then, assuming AA^H is invertible

$$\boldsymbol{\mu} = 2\left(AA^H\right)^{-1} \boldsymbol{y}. \tag{A.14}$$

Plugging the solution for $\boldsymbol{\mu}$ into the solution for $\boldsymbol{x}$ gives the least squares solution

$$\boldsymbol{x} = A^H \left(AA^H\right)^{-1} \boldsymbol{y}. \tag{A.15}$$

So

$$\boxed{\begin{aligned} &\operatorname*{argmin}_{\boldsymbol{x}} \|\boldsymbol{x}\|_2^2 \quad = \quad A^H \left(AA^H\right)^{-1} \boldsymbol{y} \\ &\text{s.t. } \boldsymbol{y} = A\boldsymbol{x} \end{aligned}} \tag{A.16}$$

A.3 Derivation of the regularized least squares approximation

The regularized least squares approximation for over- and underdetermined systems can be derived by casting it as the optimization problem

$$J(\boldsymbol{x}) = \|A\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \beta \|\boldsymbol{x}\|_2 \,, \tag{A.17}$$

where $\beta > 0$. The derivative of the regularized least squares cost function is

$$\frac{\delta}{\delta \boldsymbol{x}} J(\boldsymbol{x}) = 2A^H (A\boldsymbol{x} - \boldsymbol{y}) + 2\beta \boldsymbol{x} \,. \tag{A.18}$$

Again, because the only point with a derivative of zero is the global minimum, set the derivatives to zero

$$\frac{\delta}{\delta \boldsymbol{x}} J(bx) = \mathbf{0} \tag{A.19}$$

and plug them into the above equation to get

$$\begin{aligned} A^H A\boldsymbol{x} = \beta \boldsymbol{x} = A^H \boldsymbol{y} \\ \left(A^H A + \beta I\right) \boldsymbol{x} = A^H \boldsymbol{y} \end{aligned} \tag{A.20}$$

Then, the solution to $\boldsymbol{x}$ is given by

$$\boldsymbol{x} = \left(A^H A + \beta I\right)^{-1} A^H \boldsymbol{y} \tag{A.21}$$

So

$$\boxed{\underset{\boldsymbol{x}}{\operatorname{argmin}} \, \|A\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \beta \, \|\boldsymbol{x}\|_2^2 \quad = \quad \left(A^H A + \beta I\right)^{-1} A^H \boldsymbol{y}} \tag{A.22}$$

A.4 Does multi-channel PMM affect causality?

When $\boldsymbol{y}_m$ is defined as ones and zeros, PMM filters are not causal. This is because the linear delay term in the propagation matrix is negated by matrix inversion, and when combined with ones and zeros, creates filters with negative gain. To make causal filters, elements in $\boldsymbol{y}_m$ should have at least enough delay to offset this. So, for example, instead of

$$\boldsymbol{y}_L = \begin{bmatrix} 1 & \\ & 1 \end{bmatrix} \tag{A.23}$$

causal filters could be created by

$$\boldsymbol{y}_1 = \begin{bmatrix} 1\,G(r) \\ 0\,G(r) \end{bmatrix} ? \tag{A.24}$$

Doesn’t this observation invalidate the premise of multi-channel PMM: that the vectors $\boldsymbol{y}_m$ can be factored out? For two reasons, no, not necessarily. First, defining $\boldsymbol{y}$ as ones and zeros is a notational convenience. One could, for example, reintroduce the delays required for causal filters in multi-channel notation by appending the matrix

$$Y = G(r)I = \begin{bmatrix} G(r) & \\ & G(r) \end{bmatrix} \tag{A.25}$$

to any multi-channel formulation. Here, r is whatever distance you like for causality's-sake, e.g. that from the origin to the listener. So, a more complete notation would show, for example,

$$X_{\text{Tik}} = A^H \left(AA^H + \beta I\right)^{-1} Y\,. \tag{A.26}$$

Second, the propagation delay required to build causal filters can be added other places too, since the system linear. For example, the Green's functions can be modified to have non-causal delays that are "undone" by inversion, yielding causal filters. In this case,

$$G(\hat{r}) = \frac{\exp(-ik\hat{r})}{4\pi r}\,, \tag{A.27}$$

where $\hat{r} = R - \max(R)$.

The decision to notate $\boldsymbol{y}_m$ with ones and zeros, and resultingly to factor it out, was made for a few reasons. First, because details about filter causality primarily serve to detract from the main point of multichannel PMM; they add more visual noise to equations than information. Second, because non-causal filters generated in the frequency domain will wrap around, and be perceived by a listener only as an initial delay on the order of milliseconds.

Bibliography

[1] R S Woodworth and G Schlosberg. *Experimental psychology*. Holt, Rinehard and Winston, New York, 1962.

[2] Ole Kirkeby, Philip A. Nelson, Felipe Orduna-Bustamante, and Hareo Hamada. Local sound field reproduction using digital signal processing. *The Journal of the Acoustical Society of America*, 100(3):1584 – 1593, 1996.

[3] Ji-Ho Chang and Finn Jacobsen. Sound field control with a circular double-layer array of loudspeakers. *The Journal of the Acoustical Society of America*, 131:4518 – 4525, Apr 2012.

[4] Mincheol Shin, Filippo M. Fazi, Philip A. Nelson, and Fabio C. Hirono. Controlled sound field with a dual layer loudspeaker array. *Journal of Sound and Vibration*, pages 1 – 62, Jan 2014.

[5] Mingsian R Bai, Jheng-Ciang Wen, Hoshen Hsu, Yi-Hsin Hua, and Yu-Hao Hsieh. Investigation on the reproduction performance versus acoustic contrast control in sound field synthesis. *The Journal of the Acoustical Society of America*, 136(4):1591 – 1600, 2014.

[6] Ferdinando Olivieri, Filippo Fazi, Mincheol Shin, and Philip Nelson. Pressure-matching beamforming method for loudspeaker arrays with frequency dependent selection of control points. In 138*th* *Audio Engineering Society Convention*, volume 2, May 2015.

[7] Ferdinando Olivieri, Filippo M. Fazi, Philip A. Nelson, and Simone Fontana. Comparison of strategies for accurate reproduction of a target signal with compact arrays of loudspeakers for the generation of zones of private sound and silence. *The Journal of the Audio Engineering Society*, 64(11):905 – 917, Dec 2016.

[8] Ferdinando Olivieri, Filippo M. Fazi, Simone Fontana, Dylan Menzies, and Philip Nelson. Generation of private sound with a circular loudspeaker array and the weighted pressure matching method. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pages 1 – 21, May 2017.

[9] Filippo M. Fazi and Eric Hamdan. Stage compression in transaural audio. *144th AES Convention*, pages 4 – 9, May 2019.

[10] Tahereh Afghah, Elliot Patros, and Miller Puckette. A pseudoinverse technique for the pressure-matching beamforming method. *AES 145th Convention*, pages 1 – 6, Oct 2018.

[11] Joung-Woo Choi and Yang-Hann Kim. Generation of an acoustically bright zone with an illuminated region using multiple sources. *The Journal of the Acoustical Society of America*, 111(4):1695 – 1700, 2002.

[12] Mincheol Shin, Sung Q Lee, Filippo M Fazi, Philip A Nelson, Daesung Kim, Semyung Wang, Kang Ho Park, and Jeongil Seo. Maximization of acoustic energy difference between two spaces. *The Journal of the Acoustical Society of America*, 128(1):121 – 131, 2010.

[13] Matthew Jones and Stephen J Elliott. Personal audio with multiple dark zones. *The Journal of the Acoustical Society of America*, 124(6):3497 – 3506, 2008.

[14] Ji-Ho Chang, Chan-Hui Lee, Jin-Young Park, and Yang-Hann Kim. A realization of sound focused personal audio system using acoustic contrast control. *The Journal of the Acoustical Society of America*, 125(4):2091 – 2097, 2009.

[15] Stephen J Elliott, Jordan Cheer, Jung-Woo Choi, and Youngtae Kim. Robustness and regularization of personal audio systems. In *IEEE Trans on Audio, Speech, and Language Proc*, volume 20, pages 2123 – 2133, Sep 2012.

[16] Yefeng Cai, Ming Wu, and Jun Yang. Sound reproduction in personal audio systems using the least-squares approach with acoustic contrast control constraint. *The Journal of the Acoustical Society of America*, 135(2):734 – 741, 2014.

[17] Philip Coleman, Philip J.B. Jackson, Marek Olik, Martin Moller, Martin Olsen, and Jan Abildgaard Pedersen. Acoustic contrast, planarity and robustness of sound zone methods using a circular loudspeaker array. *The Journal of the Acoustical Society of America*, 135 (4):1929 – 1940, 2014.

[18] Philip Coleman, Philip J.B. Jackson, and Marek Olik. Personal audio with a planar bright zone. *The Journal of the Acoustical Society of America*, 136(4):1725 – 1735, Oct 2014.

[19] Jens Ahrens and Sascha Spors. Sound field reproduction using planar and linear arrays of loudspeakers. In *IEEE Trans on Audio, Speech, and Language Processing*, volume 18, pages 2038 – 2050, Nov 2010.

[20] Takuma Okamoto, Seigo Enomoto, and Ryouichi Nishimura. Least squares approach in wavenumber domain for sound field recording and reproduction using multiple parallel linear arrays. *Applied Acoustics*, 86(Supplement C):95 – 103, 2014.

[21] Xiangning Liao, Jordan Cheer, Stephen Elliott, and Sifa Zheng. Design of a loudspeaker array for personal audio in a car cabin. *Journal of the Audio Engineering Society*, 65: 226–238, 03 2017.

[22] Jerry Bauck. A simple loudspeaker array and associated crosstalk canceler for improved 3d audio. *Journal of the Audio Engineering Society*, 49:3 – 13, Jan 2001.

[23] Edgar Y. Choueiri. Optimal crosstalk cancellation for binaural audio with two loudspeakers. pages 1 – 24, 2011.

[24] Hiroaki Kurabayashi, Makoto Otani, Masami Hashimoto, and MizueKayama. Development of dynamic crosstalk cancellation system for multiple-listener binaural reproduction. *Acoustical Science and Technology*, 36:537 – 539, 11 2015.

[25] Marcos Simón Gálvez and Filippo Fazi. Loudspeaker arrays for transaural reproduction. In 22^{nd} *Int Congress on Sound and Vibration*, Jul 2015.

[26] Simon Galvez and Filippo Fazi. A loudspeaker array for 2 people transaural reproduction. In 24^{th} *Int. Congress on Sound and Vibration*, Jul 2017.

[27] Ole Kirkeby and Philip A. Nelson. Reproduction of plane wave sound fields. *The Journal of the Acoustical Society of America*, 94(5):2992 – 3000, 1993.

[28] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, Inc., 2^{nd} edition, 1999.

[29] Gilbert Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 3^{rd} edition, 2003.

[30] H. Lehnert and J. Blauert. Virtual auditory environment. In *Fifth International Conference on Advanced Robotics: Robots in Unstructured Environments*, volume 1, pages 211 – 216, Jun 1991.

[31] Jens Blauert. *Spatial Hearing: The Psychophysics of Human Sound Localization*. The MIT Press, Jan 2001.

[32] Jens Blauert, editor. *Communication Acoustics*. Springer, 2005.

[33] Elizabeth M Wenzel, Marianne Arruda, Doris J Kistler, and Frederic L Wightman. Localization using nonindividualized head-related transfer functions. *The Journal of the Acoustical Society of America*, pages 111 – 123, Jul 1993.

[34] Jesper Sandvad and Dorte Hammershoi. Binaural auralization, comparison of FIR and IIR filter representation of HIRs. In *Audio Engineering Society Convention 96*, Feb 1994.

[35] Abhijit Kulkarni, S. K. Isabelle, and H. S. Colburn. Sensitivity of human subjects to head-related transfer-function phase spectra. *The Journal of the Acoustical Society of America*, 105(5):2821–2840, 1999.

[36] Alan Dower Blumlein. Improvements in and relating to sound-transmission, sound-recording and sound-reproducing systems. *Audio Engineering Society*, 1986.

[37] J.C. Bennett, K. Barker, and F.O. Edeko. A new approach to the assessment of stereophonic sound system performance. *Journal of the Audio Engineering Society*, 33(5):314, May 1985.

[38] Earl G. Williams. *Fourier Acoustics*. Academic Press, 1999.

[39] Lord Rayleigh. On the perception of the direction of sound. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 13(74):214–232, 1907.

[40] Julius O. Smith. *Introduction to Digital Filters with Audio Applications*. `https://ccrma.stanford.edu/~jos/sasp/Choice_WOLA_Window.html`, accessed Sep 2019. online book.

[41] Stephen Boyd and Liven Vandenberghe. *Convex Optimization*. University Press, Cambridge, 7th edition, 2009.

[42] John Clark, Colin Yallop, and Janet Fletcher. *An Introduction to Phonetics and Phonology*. Blackwell Publishing, 2007.