



## 可聴音を利用した近距離情報伝送技術に関する研究

著者	松岡 保静
発行年	2016
学位授与大学	筑波大学 (University of Tsukuba)
学位授与年度	2015
報告番号	12102甲第7707号
URL	<a href="http://hdl.handle.net/2241/00143825">http://hdl.handle.net/2241/00143825</a>

# 可聴音を利用した近距離情報伝送技術 に関する研究

2016年 3月

松岡 保静

# 可聴音を利用した近距離情報伝送技術 に関する研究

松岡 保静

システム情報工学研究科

筑波大学

2016年 3月

## 博士論文概要 2015 年度（平成 27 年度）

# 可聴音を利用した近距離情報伝送技術に関する 研究

本研究では、携帯端末における近距離情報取得手段のさらなる多様化を目的として、音インタフェースに着目し、音波情報伝送による情報取得手段の確立を目指す。音を利用する利点としては、携帯端末にはマイクおよびスピーカが備わっており、特殊なハードウェアを搭載することなく利用できる。また、テレビや郊外のスピーカ等も情報発信源として利用でき、ユビキタスな通信も可能になる。無線とは異なり、壁等の遮蔽物で音波を遮蔽することもでき、情報取得エリアを制限しやすい。欠点としては、伝送速度が低いことや、伝送信号が雑音として聴こえるということである。本研究では、音を利用した情報通信技術を実用的にするための課題に取り組み、OFDM 変調を応用した音波情報伝送技術「音響 OFDM」を提案する。音響 OFDM は、音声や音楽等の音源信号の一部の周波数を利用し、聴覚に不快にならないように OFDM 信号を変形して重畳する。これにより、聴覚に不快にならない音で、短いテキスト程度の情報を瞬時に伝送できるようになる。近距離情報通信技術のひとつの手段として、音を利用した情報伝送手段を確立した。

キーワード：近距離通信、音響信号処理、データハイディング

# Doctor Thesis Overview Year 2015

## Near Field Communication Technology with Audible Sound

This paper proposes a method of near field communications on mobile devices by the use of audible sound. The advantage of audible sound is the use of the loud-speaker and microphones for the communications. It does not need the additional hardware devices because the mobile phones already have the loud-speaker and microphone for voice communications. It can use the loud-speakers of TV or public space for transmitters. In difference with the use of radio waves, the sound communication is easy to limit the scope of the communications with walls or screens. The disadvantage is the low speed of the transmission rate and noisy to human ears. This research addresses the problems of audible sound communications and proposes the applicable method Acoustic OFDM by using OFDM modulations. Acoustic OFDM uses the part of the frequency in audio signal such as voice or music. It transforms the OFDM signal in harmony with the audio signal, and adds it to the audio signal. It provides the method of transmitting a short text message over sound not to be noisy to human-ears. It diversifies the means of near field communications.

**Keywords:** Near Field Communication, Acoustic Signal Processing, Data Hiding

# 目次

<b>第1章</b>	<b>序論</b>	<b>1</b>
1.1	背景	1
1.2	研究の目的	1
1.3	本論文の構成	2
<b>第2章</b>	<b>関連研究</b>	<b>4</b>
2.1	近距離情報通信技術	4
2.2	音波情報伝送技術	5
2.3	音響データハイディング技術	6
2.3.1	情報伝送の形態	6
2.3.2	データハイディング技術	8
2.3.3	データハイディング技術の応用	11
<b>第3章</b>	<b>音響 OFDM 技術の提案</b>	<b>12</b>
3.1	OFDM の特徴	12
3.1.1	マルチキャリア伝送方式	12
3.1.2	ガードインターバル	14
3.2	音響 OFDM の原理	15
3.2.1	音波の指向性	15
3.2.2	エンコード方式	16
3.2.3	デコード方式	17
3.2.4	シンボル同期	18
3.2.5	ドップラーシフト補正	19
3.2.6	ステレオダイバーシティ方式	23
<b>第4章</b>	<b>音響 OFDM システムの設計と実装</b>	<b>28</b>
4.1	情報源符号化・復号化	28
4.2	データフレーム分割・再構築	28
4.3	通信路符号化・復号化	32
4.4	インタリーブ	33
4.5	データフレーム同期信号・同期処理	34
4.6	スペクトル法絡計算	36
4.7	OFDM 変調	36

4.8	ガード時間生成 . . . . .	37
4.9	シンボル同期信号生成・シンボル同期 . . . . .	38
4.10	OFDM 復調 . . . . .	39
4.11	音響 OFDM システムの実装 . . . . .	39
4.12	実用化への課題 . . . . .	41
<b>第 5 章</b>	<b>音響 OFDM の性能評価</b>	<b>43</b>
5.1	音質評価試験 . . . . .	43
5.2	伝送性能試験 . . . . .	44
5.2.1	伝送距離 . . . . .	45
5.2.2	指向性 . . . . .	46
5.2.3	符号化耐性 . . . . .	46
5.3	性能評価による考察 . . . . .	47
5.4	ステレオ再生の評価 . . . . .	48
5.5	ドップラーシフト補正の評価 . . . . .	49
<b>第 6 章</b>	<b>シンボル非同期伝送方式</b>	<b>51</b>
6.1	シンボル同期信号の省略 . . . . .	51
6.2	伝送システムの設計と実装 . . . . .	53
6.3	性能評価 . . . . .	53
6.4	応用例 . . . . .	54
<b>第 7 章</b>	<b>結論</b>	<b>58</b>
7.1	新規性 . . . . .	58
7.2	有用性 . . . . .	58
7.3	実用性 . . . . .	59
7.4	今後の展望 . . . . .	59
	<b>謝辞</b>	<b>60</b>
	<b>参考文献</b>	<b>61</b>
	<b>付録 A 付録</b>	<b>69</b>
A.1	Reed-Solomon 符号化・復号化 . . . . .	69
A.2	CRC エンコーダ・デコーダ . . . . .	74
A.3	畳み込み符号化・復号化 . . . . .	76

A.4	インタリーブ・デインタリーブ . . . . .	80
A.5	シンボル同期信号生成・処理プログラム . . . . .	83
A.6	データフレーム同期処理プログラム . . . . .	86



## 目 次

1.1	本論文の構成	3
2.1	音声品質と情報伝送速度の関係	6
2.2	情報伝送の形態	7
2.3	空中伝搬における関連研究と本研究の位置付け	10
3.1	OFDMの変調方式	13
3.2	ガードインターバル	14
3.3	音波の指向性	16
3.4	音響OFDMのエンコード方式	17
3.5	音響OFDMのデコード方式	18
3.6	シンボル同期信号	19
3.7	ドップラーシフト	20
3.8	ドップラーシフト補正用パイロット信号	21
3.9	モノラル再生	23
3.10	ステレオ再生	24
3.11	送信ダイバーシティ方式	25
3.12	送信ダイバーシティの効果	27
4.1	エンコード・デコード手順	29
4.2	マルチフレームの消失訂正	30
4.3	マルチフレームの構成	31
4.4	フレーム構成	35
4.5	音響OFDMシンボル構成	38
5.1	主観評価試験	44
5.2	伝送距離	45
5.3	指向特性	46
5.4	符号化耐性	48
5.5	ステレオ再生の検証	49
5.6	ドップラーシフトによる影響	50
6.1	シンボル非同期方式のフレーム構成	52
6.2	シンボル非同期方式の復調方法	53

6.3	OFDM 信号の S/N 比による受信性能 . . . . .	55
6.4	シンボル非同期方式の性能 . . . . .	55
6.5	マスキング音の利用 . . . . .	56

## 表 目 次

4.1	生成多項式 . . . . .	32
4.2	生成多項式 (畳み込み符号) . . . . .	33
4.3	インタリーブ前 . . . . .	34
4.4	インタリーブ後 . . . . .	34
4.5	インタリーブ前 . . . . .	34
4.6	インタリーブ後 . . . . .	34
4.7	音響 OFDM システムパラメータ . . . . .	40
6.1	音響 OFDM システムパラメータ . . . . .	54
6.2	周波数と音名の一覧 . . . . .	57

# 第1章 序論

可聴音を利用した近距離情報伝送技術の研究において、本省では研究の背景と目的について述べる。

## 1.1 背景

近年、機器同士の通信を利用した M2M(Machine to Machine) や、オフラインとオンラインを結ぶ O2O(Offline to Online) 等の新事業領域が開拓されつつあり、その中で様々なデバイスと携帯端末がデータを伝送する近距離情報通信技術が着目されている。この近距離情報通信技術には、現状では電波を利用した近距離無線通信が利用されているが、用途によっては、音波を利用した方が有効な場合もある。音を利用する場合、壁等によって音波を遮蔽できるため、限られた空間のみ情報伝送を可能にする等の対応が容易にできる。また、携帯端末にはマイクおよびスピーカが備わっており、特殊なハードウェアを搭載することなく利用できる。テレビや郊外のスピーカ等も情報発信源として利用でき、ユビキタスな通信も可能になる。例えば、テレビやラジオの音声に番組の Web サイトの URL 情報を重畳して伝送し、携帯端末で録音してその情報を抽出できれば、番組の Web サイトに自動的にアクセスできるようになる。

このような背景から、本研究では、音波を利用した近距離情報伝送技術に着目し、携帯端末における情報取得手段のさらなる多様化を目的として音波情報伝送による情報取得手段の確立を目指す。

## 1.2 研究の目的

本研究では、携帯端末に搭載されているマイク・スピーカで情報伝送が可能な音波情報伝送技術を実現する。上記のマイクやスピーカは、人の可聴領域の周波数のみ録音や再生ができる周波数特性であることが多い。そのため、この周波数領域で通信するとなると人間に聞こえる音で通信することになる。情報伝送に用いる変調

信号の音は、通常はノイズとして聞こえるため、人間の聴覚には不快な音となってしまふ。

本研究では、可聴領域の周波数で人間の聴覚に不快にならないように情報を伝送でき、さらに伝送速度も向上できる技術を実現させる。まず、可聴領域の音で聴覚に知覚できないように情報を埋め込む技術としては、すでに電子透かし技術等で利用されている技術がある。これらの既存技術の利用も可能であるが、いずれも伝送速度が10bps程度と低く、簡単なテキスト情報(URLやメールアドレス等)を伝送するにも数十秒もかかってしまい、ユーザが許容するレスポンスタイムをはるかに越えてしまふ。

また、超音波を用いる手法では、伝送速度は高くできるが、市販のオーディオデバイスや携帯端末で超音波を再生・録音できるものは少なく、またテレビやラジオ放送では超音波帯域の信号を送ることはできないため、利用できる場面がほとんど無くなる。以上のような背景から、下記の条件を満たす音波情報伝送技術を実現させることを本研究の目的とする。

1. URLや簡単なテキスト情報を1~2秒で伝送できる。
2. 可聴帯域の音波で情報を伝送し、市販のオーディオデバイスや携帯端末で情報伝送が可能である。
3. 伝送信号を聴覚に不快にならないように音声・音楽に重畳する。

これらの条件を満たした音波情報伝送技術を「音響 OFDM」[1][2][3]として提案し、伝送システムの設計・実装、性能評価を行う。

### 1.3 本論文の構成

第1章では本研究の背景および目的を書いた。第2章では、音波を利用した通信技術における関連研究について述べる。第3章では、本研究で提案する音響 OFDM 技術について説明する。これは著者の論文 [3][4] の内容に相当する。第4章では、音響 OFDM 技術を用いた音波情報伝送システムの設計と実装について説明する。ここでは、著者の論文 [2] の内容に相当する。第5章では、音響 OFDM 技術の性能評価試験について説明する。これは著者の論文 [1] の内容に相当する。

第6章では、伝送速度よりも通信距離を優先する場合の伝送方法として、シンボル非同期伝送方式についても述べる。これは著者の論文 [5] に相当する。最後に第7章で本研究の結論についてまとめる。図 3.1 に本論文の構成と研究論文の関係を示す。

付録として、音響 OFDM システムを構成するソフトウェアのソースコードおよびその説明を添付する。



図 1.1: 本論文の構成

## 第2章 関連研究

本章では、まず近距離情報通信技術の概要について述べ、近距離情報通信に音波を利用する有効性について説明する。次に、音波を利用した情報通信技術について説明し、本研究と密接に関わる音響透かし技術について解説する。

### 2.1 近距離情報通信技術

携帯電話等の中長距離無線アクセス方式に加え、近年、数センチ～数メートル程度のデータ伝送を目的とした様々な近距離情報通技術が開発・実用化されている。これらはPAN (Personal Area Network) とも称されており、無線を用いたPANの代表的なものとしては、Zigbee[6], Bluetooth[7], NFC(Near Field Communication)[8]等がある。

近距離通信技術を利用した実用例として、M2M (Machine to Machine) [9]のサービスや、O2O (Online to Offline) [10]等のサービスがある。M2Mとは、人間の介在無しに、機器同士がコミュニケーションをして動作するシステムである。O2Oとはオンラインとオフラインの購買活動が連携しあうシステムであり、連携の形態によっては、Offline to Onlineとも称されることもある。

これらのサービスやシステムは、無線による近距離通信技術が用いられている。しかしながら利用形態やシステムによっては、無線ではなく音波を利用することが有効な場合もある。音波を利用する有効性としては、以下のものが考えられる。

1. 特殊なハードウェアを必要とせず、スピーカとマイクで通信が可能になる。
2. 壁や障害物によって、通信可能エリアを遮蔽しやすい。
3. 電波を利用できない場所での代替手段となりうる。

特に、1に関しては、携帯端末には通話用のスピーカやマイクが搭載されており、またテレビやラジオ等のスピーカを利用すれば、これらが情報発信源ともなりうるため、既存のデバイスをそのまま利用できる。音波を利用する欠点としては、情報伝送速度が電波に比べて著しく低いことである。これは、通常のスピーカやマイク

で利用できる周波数の幅が、電波に比べて狭いためである。そのため、限られた周波数の中で効率的に情報を伝送する必要がある。

本研究では、音波を利用した近距離通信技術の確立を目的とし、情報伝送速度を実用的なレベルまで高くできる技術を提案し、システムの設計・評価を行う。

## 2.2 音波情報伝送技術

音波を利用した情報伝送技術としては、大きく3つの種類に分けられる。ひとつは、伝送信号の音をそのまま再生して伝送する手法 [11][12][13] である。これらは可聴音帯域で振幅変調や位相変調した信号を何の加工もせずにそのまま伝送する方法であり、100bps~1kbps 程度の伝送速度がでるが、伝送音がノイズとして聞こえてしまうため、聴覚に不快感を与えるというデメリットがある。

ふたつめは、音響透かし技術に基づく手法 [18][19] であり、音声や音響信号に、人間の聴覚にはわからないように情報を埋め込んで伝送する技術である。これらは既存の音声信号や音響信号に、聴覚心理マスキングモデルに基づいて聴覚にはほとんど認知できないレベルの音を利用し、情報を伝送する。この手法を利用すれば、聴覚に不快な音をださずに情報を伝送できるが、情報伝送速度が10bps 程度しかだせないため、1秒に1文字程度の情報しか伝送できず、実用的な情報伝送は困難である。

最後に、超音波を利用した情報伝送技術もある [14]。これは人間の聴覚で知覚できる20kHz までの周波数よりも高い周波数の音で伝送する手法である。超音波を利用すれば人間の聴覚には聞こえない音で伝送でき、周波数も可聴帯域よりも広く利用できるため、伝送速度も向上する。しかし、通常のスピーカやマイクでは超音波を再生・拾音できないため、特殊なハードウェアが必要ないというメリットがなくなってしまう。これは音波を利用するメリットの中で最も重要なことでもあるため、超音波を利用した伝送技術については、検討の対象外とする。

水中の通信については電波が伝送できないため、音波を利用した水中通信技術 [15][16][17] が多々あるが、これらも特殊なデバイスを利用して広帯域の周波数を用いて伝送するものが特徴であり、通常のスピーカ、マイクで空中伝搬させる技術とは異なる分野であるため、検討の対象外とする。

上記で述べた既存の音波情報通信技術を検討すると、図2.1で示すように音声品質（聴覚的不快感の無さ）と情報伝送速度のトレードオフの関係にあると言える。本研究では、このトレードオフの関係を越える音波情報伝送技術の確立を目的とし、「音響 OFDM」という技術を提案する。具体的には、既存の音声・音響信号に、聴覚的



な不快感を与えずに情報を埋め込むことができ、情報伝送速度も数百 bps 程度の実用できるレベルを提供できる技術を提案する。

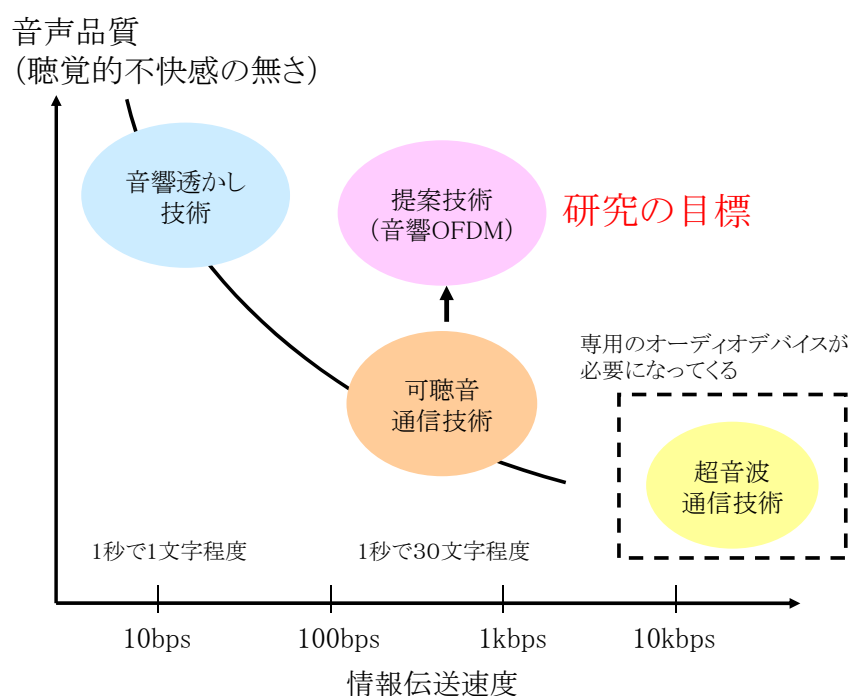


図 2.1: 音声品質と情報伝送速度の関係

## 2.3 音響データハイディング技術

音響データハイディング技術の特徴について説明し、既存のさまざまな技術やその応用方法について記述する。

### 2.3.1 情報伝送の形態

まず、音響データハイディング技術で重要となるのが埋め込みデータの頑健性であると考えられる。頑健性とはデータを埋め込むエンコーダからデータを抽出するデコーダの間でなんらかの処理が行われてもデータを抽出できる耐性である。[20]では、このエンコーダからデコーダへの伝達環境を図1に示す4種類に分類している。(A)はデジタルコピーであり、エンコードされたデータをそのままデコードする場合で、頑健性への制約は最も少ない。(B)は、エンコードされたデータにサンプ

リングレート変換（アップサンプリングやダウンサンプリング処理）をしたデータをデコードする場合であり，振幅や位相は変わらないが，時間的特長が失われやすい．(C)は，エンコードしたデータをアナログ信号に変換し，アナログケーブルで伝送されてリサンプリングされたデータをデコードする場合であり，振幅やサンプリング点，各サンプルの量子化等さまざまな値が変わるが，位相はほとんど変わらずに伝達できる．(D)は，エンコードしたデータをアナログ信号に変換し，電気音響変換をしてスピーカからマイクロフォンに空中伝播したものをリサンプリングしてデコードする場合であり，(C)の場合に比べて非線形効果による位相の変化やエコー等も生じる．コンテンツの不正コピー防止，著作権保護に応用する音響電子透かしの場合，一般的には(C)までの頑健性が十分であるが，情報を伝送することを主目的とする場合は(D)までの頑健性が必要である．

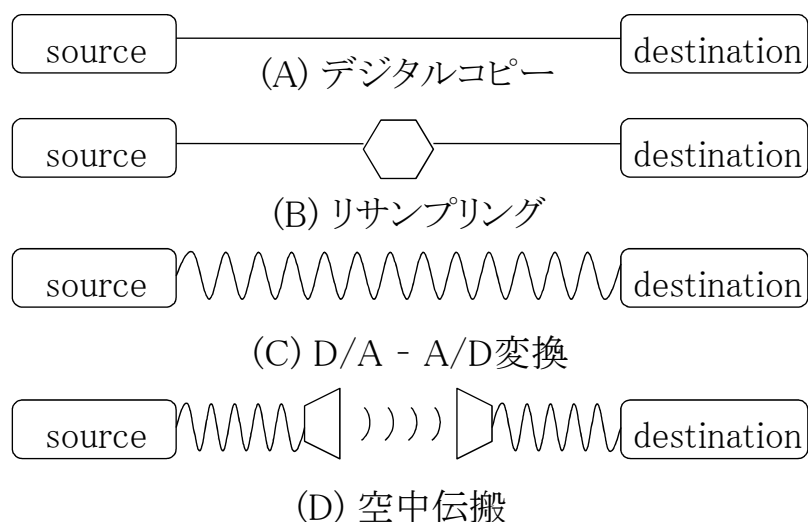


図 2.2: 情報伝送の形態

本研究では，近距離情報伝送技術として音波を利用するため，(D)までの頑健性が必要であり，実用的な5～10m程度の空中伝播が可能な音響データハイディング技術確立することを目指す。

また，データハイディング技術の要求条件としては，[21]によれば，3種類に分類できる．ひとつは，ノン・ブラインドな利用形態であり，これはオリジナル音源とデータを埋め込んだ音源の両方を比べて情報を抽出することができ，一番情報を抽出しやすい。

2つ目は，ブラインドな利用形態であるがオリジナル音源との同期がとれている

ことが前提である場合である。つまり、オリジナル音源は無く、データを埋め込んだ音源しか無いが、音源の先頭はオリジナル音源と同じである場合である。この場合、オリジナル音源と比較することはできないが、情報が埋め込まれている箇所がわかるため、まだ情報を抽出しやすい。

3つ目は、ブラインドな利用形態であり、さらにオリジナル音源との同期もとれない場合である。オリジナル音源は無く、データを埋め込んだ音源も、オリジナル音源のどの部分かはわからないため、一番情報を抽出しにくくなる。

本研究では、近距離情報伝送技術としてデータハイディング技術を利用するため、データを埋め込んだ音源を空中伝搬させることになり、受信側にオリジナル音源もなく、同期もとれないため、上記の3番目の形態でのデータハイディング技術が必要となる。

### 2.3.2 データハイディング技術

音響信号へのデータハイディング技術の重要な点として、まずデータを埋め込んだ信号ともとの信号が聴覚的に区別つかないことが重要となる。これにはHAS(Human Auditory System)の特性に基づいたさまざまな手法が提案されている。代表的な技術としては、エコー操作方式、位相操作方式、スペクトル拡散方式等があるが、これらの手法を用いて情報を空中伝搬させようとすると、以下のような問題がある。

#### エコー操作方式

聴覚が短いエコーを知覚できないという特性を利用し、エコーの遅延や振幅を変化させることによって伝送ビットを識別できるようにする方法である[22][23]。これは主に、複数種類のエコーパターンを用意し、伝送ビットによってエコーパターンを変化させて伝送する方式である。エコーパターンには単純に2つのインパルスからなるものや、PN系列等で生成したもの等があり、複雑なエコーパターンになるほど、データハイディングとしての頑健性や秘匿性が向上する。しかしながら、空中伝搬による音波情報伝送では、反射波やスピーカの減衰振動等によって、さまざまなエコーが生じるため、伝送情報を識別することが困難になる。

#### 位相操作方式

エコー操作方式と同様に、聴覚が位相の変化に鈍感であることを利用して、位相をある規則に従って変化させることで伝送ビットを識別できるようにする方法であ

る [24][25]. 源信号がある場合には、源信号と比較して位相の変化によって伝送情報を識別し、源信号が無い場合は、基準となる信号の位相と比較して位相の変化を識別し伝送情報を抽出する。これもスピーカでの再生や空中伝搬を行うと、位相の非線形な歪みが生じるため、伝送情報の識別が難しくなる。

### スペクトル拡散方式

聴覚心理モデルを用いて周波数マスキング閾値を計算し、伝送信号に PN (Pseudo Noise) 系列を乗算して全周波数帯域に拡散した信号をマスキング閾値以下になるようにして埋め込む方法である [26][27]. 空中伝搬や環境雑音にも強い耐性があるが、PN 系列の拡散率を高くしないと信号を抽出できないため、必然的に伝送速度が低くなる (10bps 程度). したがって、簡単なテキスト情報を伝送するにも数十秒程度かかってしまい、近距離情報通信という観点から見ると、実用性に困難がある。

### その他の方式

上記の 3 種類の方式が代表的なデータハイディング方式であるが、それ以外にも多種多様な方法が提案されている。

まず、フィルタバンクによっていくつかのサブバンドに分け、2つのサブバンドを 1 組の伝送ビットとし、サブバンド間の位相差や振幅差によって伝送ビットを識別する方法がある [28][29]. これもサブバンド間の位相や振幅を比べることで伝送ビットを識別しているため、空中伝搬やスピーカ再生によって伝送ができなくなる場合があるが、複数のサブバンドに分割して伝送ビットを並列に伝送できるため、伝送速度を高くできる特徴がある。

また、スペクトル拡散方式の性能をよくするために、元の音源を PN 系列と無相関に近いように位相を調節する方法も提案されている [30]. これは、元の音源をサブバンドごとに分割し、サブバンドごとに PN 系列との相関が低くなるように位相をずらして合成し、そこに PN 系列を乗算した伝送データを重畳する方法である。これにより、伝送性能を改善することが可能になる。

ステレオ信号の右チャンネルと左チャンネルを比較して情報を伝送する方法もある [31][32]. これはステレオ信号に対して、低周波領域の定位を、埋め込む伝送ビットによって偏移させることで情報を伝送する方法である。

図 2.3 に、上記で記述した既存研究を、音波情報伝送の空中伝搬に応用した場合における音質と情報伝送速度の関係を記載した。最も音質がよいのはスペクトル拡散方式 [26][27] であり、これは空中伝搬にもある程度の頑健性がある。然しながら、元々

の方式上、拡散率を高くしないと情報を抽出できないため、情報伝送速度は 10bps 以下となる。エコー操作方式 [22][23] や位相操作方式 [24][25] は、空中伝搬には弱く、空中伝搬にまで耐えられるように設計したとすると、スペクトル拡散方式よりも音質、情報伝送速度ともに劣ると考えられる。フィルタバンクを使って情報を伝送する方式 [28][29] は、設計次第では 10bps 以上の伝送速度も可能であるが、それに伴い音質は劣化する。ステレオ比較方式 [31][32] も 80bps 程度の伝送速度が可能であるが、低周波領域の定位が変わる等の影響があるため、音質はやや劣化すると考えられる。

一方、可聴音周波数帯の搬送波を使って、振幅変調や位相変調をして伝送する方法 [11][12][13] では、使う周波数幅によっては、数百 bps～1kbps 程度の情報伝送が可能になる。しかし、伝送信号はホワイトノイズに近い音でこれをそのまま再生するため、音質（聴覚的不快感の無さ）を考慮すると、音質は悪いと考えられる。

本研究（音響 OFDM）の目標としては、この音質と情報伝送速度のトレードオフ関係を越えた技術を目指し、情報伝送速度としては数百 bps 程度で伝送でき、音源信号に重畳して伝送しながら元の音源信号の音質を極力劣化させない方式を提案する。

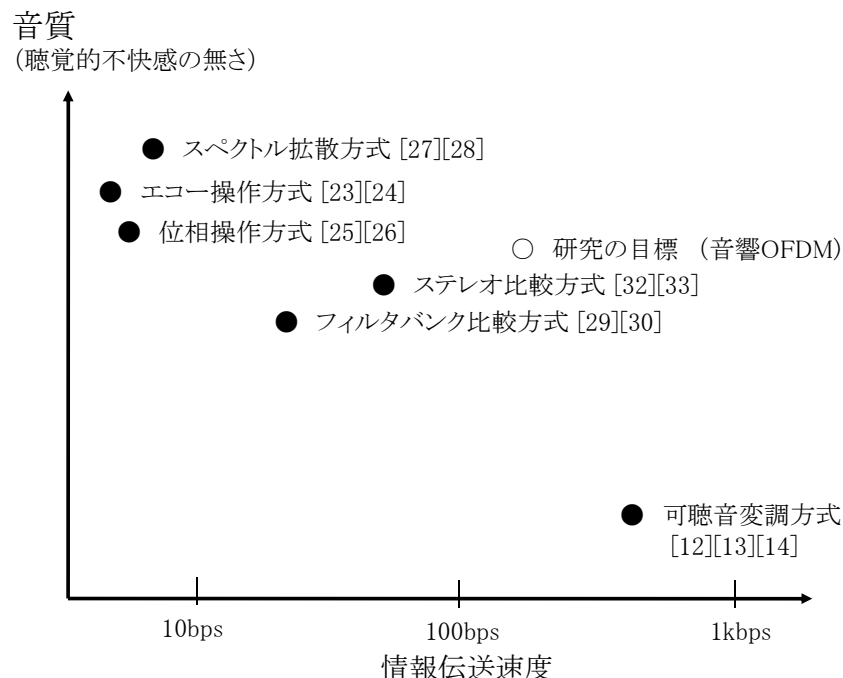


図 2.3: 空中伝搬における関連研究と本研究の位置付け

### 2.3.3 データハイディング技術の応用

本研究では、音響データハイディング技術を近距離音波情報伝送に用いるが、それ以外にどのような用途にデータハイディング技術が利用されているかを説明する。

まず、最も主要な利用方法としては、音源データのデジタルコピーを防止するためのコピープロテクション情報を埋め込む方法 [33][34] である。これは、音源データが不正にコピーされることを防ぐことを目的としており、主にデジタルコピーを防ぐようにできている。スピーカから再生したものを録音したコピーに関しては音源品質が劣化しているということで、空中伝搬によるコピーまでは対応していないものが多い。

次に主要な利用方法としては、音源データのメタデータ情報を埋め込む方法 [35] である。これは音源データのタイトル等を音源データそのものに埋め込んでおくことで、アナログ形態を介してコピーされてもメタデータを復元できるようにしている。

さらに音響符号化の面でもデータハイディング技術が用いられる場合もある。まず、狭帯域符号化の場合に高周波成分の情報を低周波帯に埋め込んでおく方法がある [36][37]。狭帯域符号化では高周波成分が削られてしまうため、音声品質が劣化するが、高周波成分の情報が何かしら得られれば、復号側で高周波成分を疑似的に作ることが可能になる。そのため、高周波成分の情報を低周波帯域に埋め込んでおくことで、狭帯域符号化を通して音声品質が著しく劣化されないようにできる。

また狭帯域符号化でなく、MP 3等の音響符号化においても、圧縮率を高めると高周波成分の MDCT 係数等が削られていく。したがって、高周波成分の MDCT 係数を低周波帯域に埋め込んでおく方法も提案されている [38]。

聴覚障害者向けに応用する方法も提案されている [39]。これは音声データの字幕情報を音声データに埋め込んでおき、聴覚障害者にも文字で音声の内容がわかるような応用方法である。

上記で説明したように、音響データハイディング技術は様々な用途に応用される方法が提案されている。本論文では、以降、空中伝搬をしても情報を伝送できる近距離情報伝送に向けた音響データハイディング技術について検討し、次章にて「音響 OFDM」技術を提案する。

## 第3章 音響 OFDM 技術の提案

この章では、本研究で提案する「音響 OFDM」技術について説明する。本章は、筆者論文 [3] の内容を詳解に説明する。音響 OFDM は、次世代の移動通信にも有望とされている OFDM 方式を応用する。OFDM 方式は、複数の狭帯域信号を周波数多重して並列伝送を行うため周波数利用効率に優れている。また、遅延波干渉の影響に対処しやすい点も反射波の影響が強い音波通信に有効である。本章では、まず OFDM の基本的な特徴について述べ、次に音響 OFDM の詳細について述べる。

### 3.1 OFDM の特徴

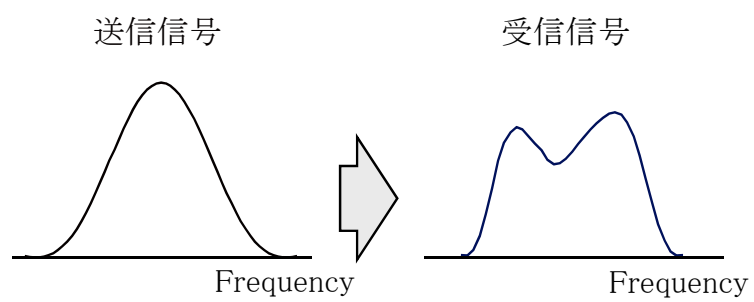
OFDM の特徴としては、直交周波数の搬送波による複数の狭帯域信号を並列に並べることであり、これによりフェージングや反射波による干渉に強くなる等の特徴がある。以下に、これらについて示す。

#### 3.1.1 マルチキャリア伝送方式

OFDM の特徴のひとつとしては、狭帯域の搬送波を密に並べるマルチキャリア伝送方式である。図 3.1 は従来のシングルキャリア伝送方式と OFDM によるマルチキャリア伝送方式での周波数選択性フェージングを表す。シングルキャリア伝送方式では搬送波のスペクトルが歪むため、これを補正するための複雑な等化処理が必要となる。一方、マルチキャリア伝送方式では各搬送波が受けるフェージングはフラットに見えるため、等化処理が簡単になり伝送性能も向上する [40]。また、複数のキャリアで伝送する場合において、通常はお互いのキャリアが干渉しあわないようにガードバンドを設けてキャリアを並列に並べるが、OFDM の場合は、直交周波数という条件を満たしたキャリアを並べるため、ガードバンドを設けなくても互いのキャリアが干渉しないようになっている。従って、複数のキャリアを密に並べることができ、伝送効率を高めることができる。

ここで、直交周波数の条件を記載する。直交周波数とは、隣のキャリアと重なる点がちょうどゼロになるためのキャリア幅になる。あるキャリアの周波数の角周波

シングルキャリア伝送



マルチキャリア伝送

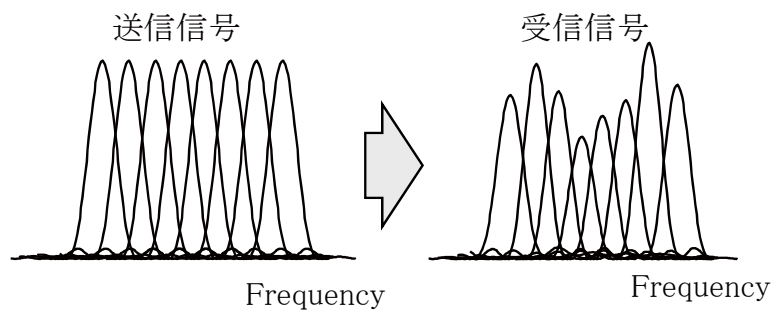


図 3.1: OFDM の変調方式



数を  $\omega$  とし，隣接するキャリアの周波数を  $\omega + \Delta\omega$  とすると，直交周波数は  $\Delta\omega$  になる．シンボル長を  $T$  とすると，直交周波数の条件は下記の式になる．

$$\int_0^T \cos \omega t \cdot e^{-j(\omega+\Delta\omega)t} dt = 0$$

この式により，直交周波数の角周波数  $\Delta\omega$  は，下記の式を満たしたときに直交周波数となる．

$$\Delta\omega = \frac{2n\pi}{T}$$

したがって，直交周波数は， $1/THz$  となる．伝送周波数帯の帯域幅を  $FHz$  とすると，FT本のサブキャリアを並列に伝送できることになる．1つのサブキャリアで1ビットの情報を伝送するとすると，伝送速度は， $FT/T = Fbps$  となり，伝送帯域幅  $1Hz$  で  $1bps$  の伝送が可能になる．実際には，同期信号や誤り訂正符号等のデータも伝送する必要があるため，伝送速度はこれよりも低くなる．

### 3.1.2 ガードインターバル

OFDMにはガード時間を挿入することで反射波等によるマルチパス干渉に対処できるという特徴もある．図3.2に示すように，各OFDMフレームはデータシンボル区間とガード時間の区間で構成され，ガード時間の区間はデータシンボルの後方をコピーして生成される [41]．これにより，ガード時間とOFDMフレームは位相が連続になり，ある程度の遅延波が重なっても直交性を保つことができる．このガード時間によりスピーカの減衰振動や反射波による遅延波にも対応でき，ロバストなデータ伝送が可能になる．

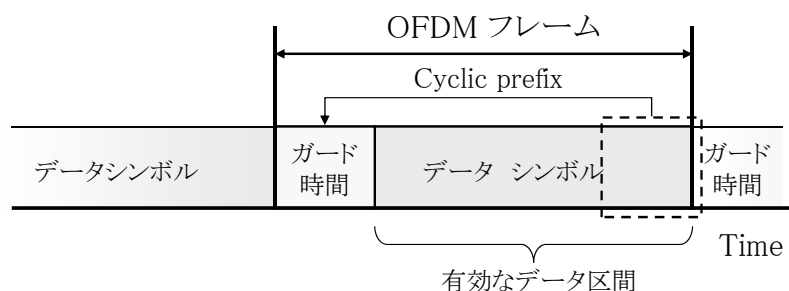


図 3.2: ガードインターバル

ガードインターバルを越える遅延波を受信すると、OFDM シンボル内に位相の不連続な信号が混ざることになり、OFDM の特徴である直交周波数によるマルチキャリア伝送の直交条件がくずれてくる。そのため、ガード時間を越える遅延波が含まれると、著しく受信精度が悪くなる。したがって、実用面では、運用シーンを考慮してどれぐらいの反射波による遅延波が届く可能性があるかを算出し、ガードインターバルの長さを設計する必要がある。

## 3.2 音響 OFDM の原理

本節では、上述した OFDM 変復調を応用した音波情報伝送技術「音響 OFDM」を提案し、この音響 OFDM の各要素技術について詳解する。

### 3.2.1 音波の指向性

音響 OFDM では、OFDM 変調信号を伝送する周波数は、可聴音域の中でも高域の周波数帯を用いる。これは、人間の聴覚にとって、低域の音よりも高域の音の方が変化に鈍感であるという特徴のためである。しかしながら、周波数が高くなるほど音波の指向性も鋭くなっていく [42]。ここでは、音波の指向性の特徴について示し、実用に妥当な伝送周波数帯を考察する。

音波の指向性は、周波数とスピーカの開口面積で影響する。音波の指向性の計算方法としては、点音源をスピーカの開口面に敷き詰めた場合にそれぞれの点音源の相互干渉を考慮することによって計算できる [43]。点音源には指向性は無いため、どの周波数でも無指向で広がっていくが、これをひとつの面に並べたときにそれぞれの点音源が干渉しあい、高周波数になるにつれて指向性が鋭くなっていく。

スピーカの開口面の口径を  $a$ 、周波数を  $f$ 、音速を  $V$ 、そして指向角度を  $\theta$  とすると、それぞれの角度における振幅特性は  $k = 2\pi f/V$  となる  $k$  と、ベッセル関数  $J_1$  [44] を用いて、下記の計算式による  $D(\theta)$  として計算できる。

$$D(\theta) = \frac{2J_1(ka \sin \theta)}{ka \sin \theta}$$

人間の聴覚に聞こえる周波数としては主に 20Hz～20kHz と言われている。しかし、全てのスピーカが 20kHz まで平坦な周波数特性を持っているとは限らない。スピーカによっては、10kHz 程度までしかフラットな特性をもっていないものもある。また、上記の式により周波数が高くなるほど指向特性も鋭くなり、スピーカの正面でしか音を拾えなくなってくる。例えば、図 3.3 に示す通り、65mm の口径のスピーカ

で8kHzの周波数を再生すると、20度程度までが音を拾える範囲になる。開口面が大きくなれば指向性も鋭くなり、小さくなれば指向性は広がっていく。

このような実用的な側面から、音響OFDMでは、OFDM変調してデータを伝送する周波数帯を6.4~8.0kHz付近を利用する。これにより、ほぼすべてのスピーカで再生が可能であり、指向特性も極端に狭くなることなく情報を伝送できる。

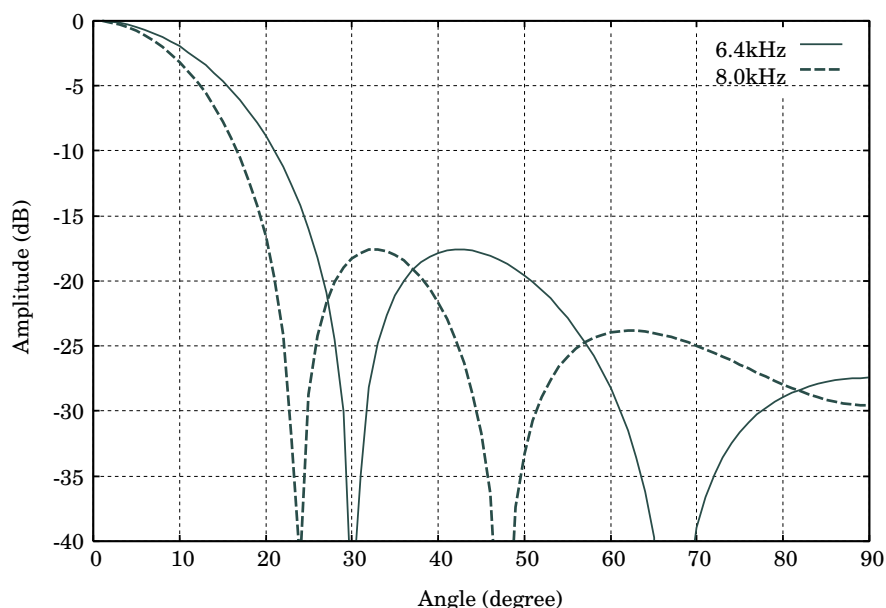


図 3.3: 音波の指向性

### 3.2.2 エンコード方式

音響OFDMの特徴は、OFDM変調信号を聴覚に不快にならないように変形して音声・音楽に重畳する。一般的にフラットなパワースペクトルの音はノイズのように聞こえ聴覚には不快に感じることが多い[45].. 一方、特定の周波数にパワーが偏っている音は、ノイズよりもトーンに近い音になる。通常のOFDM変調信号は、全てのサブキャリアのパワーが一樣であるため、ノイズのような音に聞こえるが、各サブキャリアのパワーを、重畳する音声・音楽のパワースペクトルに合わせることで、不快なノイズ音のある程度音声・音楽に調和する音に変換できる。

図3.4に音響OFDMの基本となる伝送方式を示す。(1)が伝送データを重畳する前のオリジナル音源である。まず、オリジナル音源をフーリエ変換し、周波数スペクトルを求める。その後ローパスフィルタで高域成分を除去する。これが(2)の低域音響

信号となる。次に、高域の周波数のサブキャリアを伝送信号で変調した OFDM 変調信号を生成する。サブキャリアの変調には位相変調を用いる。つまり伝送ビット列から各サブキャリアの位相を決定し、IFFT（逆フーリエ変換）することで、OFDM 変調信号が生成される。これが (3) の OFDM 変調信号となる。この OFDM 変調信号のままでは聴覚にはただの不快感なホワイトノイズの雑音として聞こえてしまう。したがって、この OFDM 変調信号をオリジナル音源のスペクトル包絡に合わせてサブキャリアのパワーを調節し、高域音響信号を生成する。これが (4) の高域音響信号となる。最後に、低域音響信号と高域音響信号を合成して合成音響信号を生成する。これが (5) の合成音響信号である。この合成音響信号をスピーカから再生する。

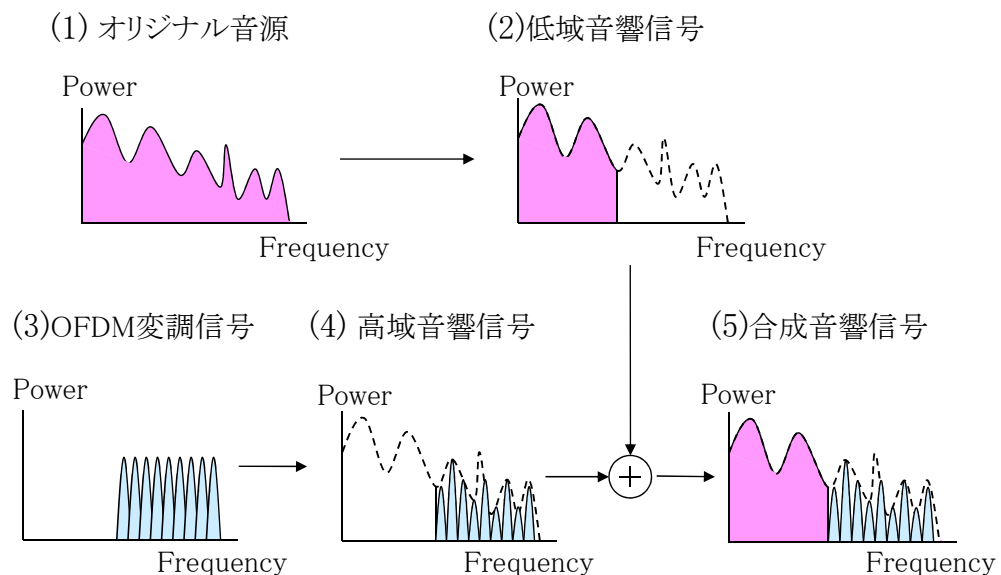


図 3.4: 音響 OFDM のエンコード方式

このようにエンコードした音響信号は、伝送信号としての OFDM 変調信号を含みながらも、元のオリジナル音源の周波数スペクトルに合った音響信号として再生できるため、元のオリジナル音源とほぼ類似した音として再生できる。

### 3.2.3 デコード方式

受信側で音響 OFDM で伝送された信号をどのように抽出するかであるが、まず OFDM の変調信号を抽出し、OFDM のシンボルの同期がとれる必要がある。

図 3.5 に、音響 OFDM の基本となる受信処理を示す。(1) が受信機で録音した録音音響信号である。この録音音響信号を直線位相フィルタを利用して、(2) 低域受信信号と (3) 高域受信信号に分離する。OFDM の各サブキャリアの変調には位相変調を用いるため、位相を変形しないように直線位相フィルタである必要がある。次に、低域受信信号から OFDM のシンボルの同期をとる。これは次節にて詳しく解説するが、OFDM シンボルの区切りを識別する処理である。この OFDM シンボルの区切りを識別した後、高域受信信号に対して OFDM の復調を行い、伝送ビット列を抽出する。OFDM の復調には FFT を用い、各サブキャリアの位相情報を抽出して伝送ビットを識別する。

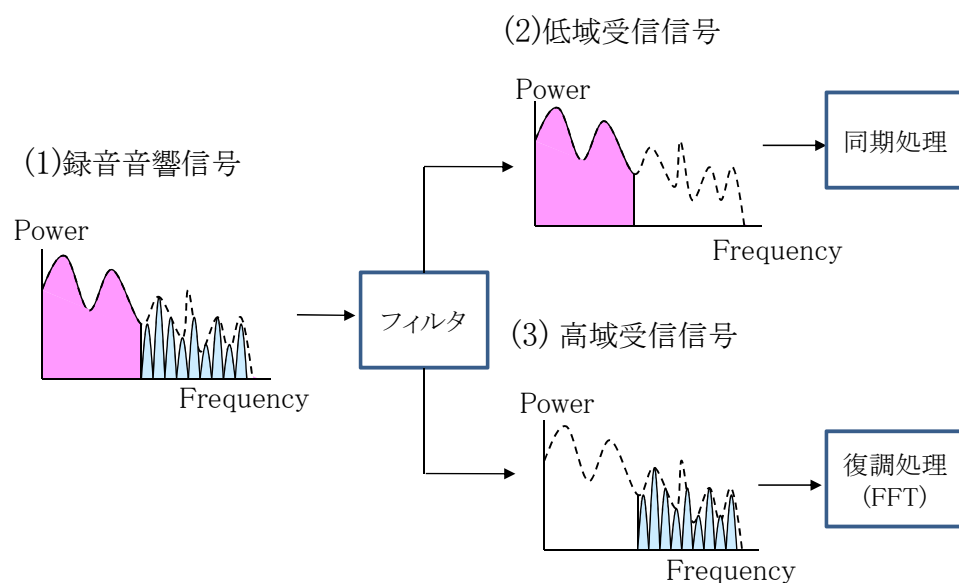


図 3.5: 音響 OFDM のデコード方式

### 3.2.4 シンボル同期

受信側で OFDM の復調をするためには、OFDM 送信シンボルの境界を検出する必要がある [46][47]。ガード時間と OFDM 変調信号との相関で検出する方法等 [48] もあるが、反射などで遅延波があると精度が落ちるため、シンボル同期用の信号を追加する。このシンボル同期用の信号には M 系列符号 [49] を用いる。この M 系列符号は、同期がとれたときの自己相関値が高く、同期ずれているときの相関は低いという特徴があり、自己同期をとるために有効な符号系列である。この符号系列は

スペクトル拡散 [50][51] 等にも用いられている。

送信側では，図 3.6 に示すように，シンボル同期用の M 系列符号を PSK で位相変調してシンボル同期信号を生成し，低域のオーディオ信号に重畳する．シンボル同期用の符号系列は，表 1 に示すシンボル同期信号生成多項式によって生成される M 系列符号である．受信側では，受信信号の低域信号とシンボル同期信号の相関計算をし，相関値の最も高くなるポイントを OFDM 送信シンボルの先頭とし，OFDM 復調を行う．

このシンボル同期信号は，拡散信号のため，一種の白色ノイズのように聞こえることがある．この雑音感を低減するため，聴覚心理モデル [52][53][54] を利用して，音声・音楽の低域信号に周波数マスキングの閾値以下になるようにして重畳することが望まれる．まず，音声・音楽がある低周波数帯域の周波数マスキングの閾値を計算する．次に M 系列で低周波数域帯に拡散した OFDM フレーム同期信号を周波数マスキングの閾値以下にレベルを調節して重畳する．これにより OFDM フレーム同期信号の音は聴覚には知覚できなくなる．

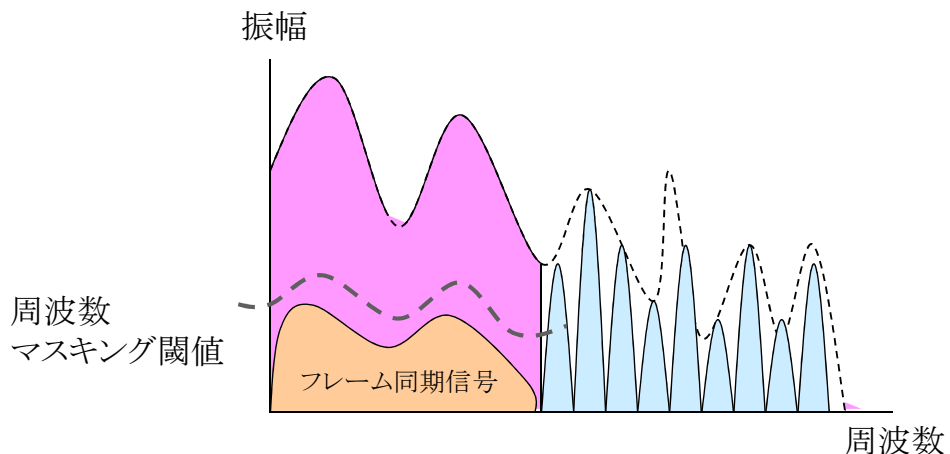


図 3.6: シンボル同期信号

### 3.2.5 ドップラーシフト補正

OFDM の特徴として，ドップラーシフトに弱いという特徴がある．電波による無線通信の場合，電波の速度は光速 30 万 km/s であるため，歩行程度の受信機の移動であれば，ドップラーシフトは無視できるが，音波の場合，音速が 340m/s と小さい

ため、歩行程度でも受信機が動くところによるドップラーシフトが大きい。無線通信の場合でも、高速移動するとドップラーシフトを無視できなくなってくるため、周波数同期の仕組みは使われている [56][57]。しかしながら、無線通信の場合と音波通信の場合では、ドップラーシフトの補正方法が大きく異なってくる。図 3.7 に無線通信の場合と音波通信の場合のドップラーシフト補正の違いを示す。

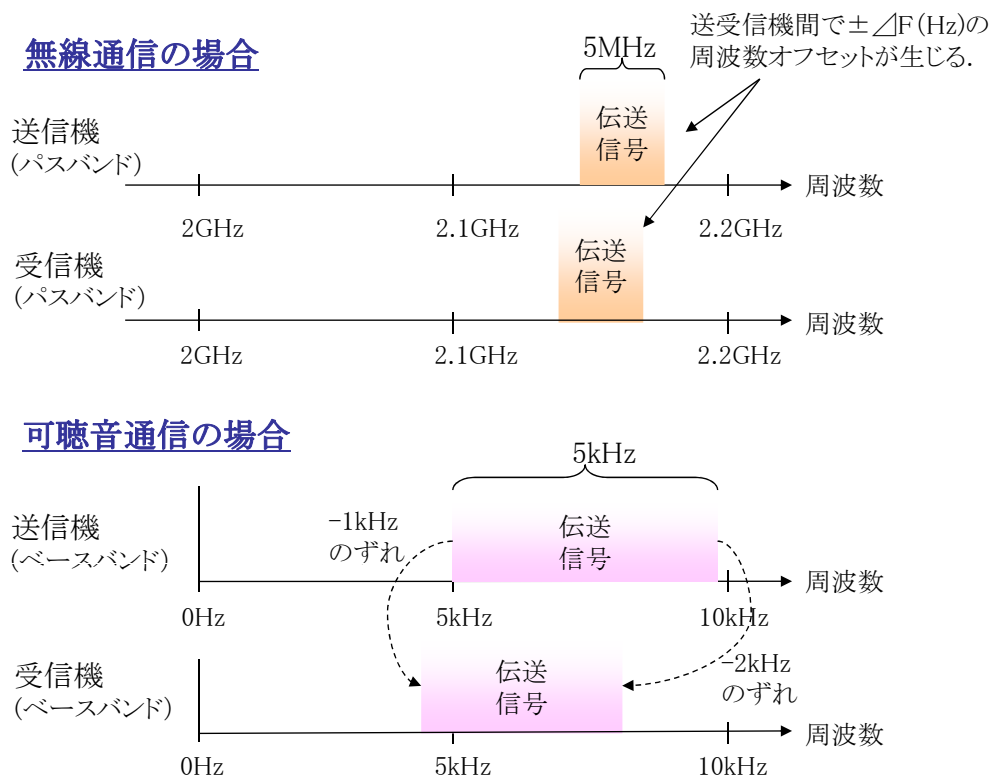


図 3.7: ドップラーシフト

無線通信の場合、多くの場合は高い周波数帯の一部を利用して OFDM 変調信号を送信する。例えば、図 3.7 の例だと、2150MHz~2155MHz の 5MHz だけを利用して送信する。このように無線通信の場合は、送信する周波数帯の幅が、その周波数に比べて小さい。また電波の速度も高速で速いため、ドップラーシフトの影響も小さい。受信機が 300km/s のスピードで動いたとしても、周波数のずれは 0.1 パーセント (100ppm) となる。上記の例で、0.1 パーセントのドップラーシフトが生じた場合に、送信周波数帯がどのように変化するかを計算すると、下側周波数は、2150MHz から 2147.8MHz にずれ、上側周波数は 2155MHz~2153.8MHz にずれる。したがって、どちらも約 2.2MHz ずれることになり、2.2MHz だけ平行移動したとほぼ同等になる。

したがって、無線通信の場合は、このドップラーシフトを補正するためには、2.2MHzの正弦波を重畳して、2.2MHz平行移動させて元に戻してから復調することができる。

しかし、音波の場合のドップラーシフトは、このような平行移動では対処できない。可聴音で伝送する場合、伝送する周波数帯の幅がその周波数に比べても小さくないため、ずれる幅が一定にはみなせない。また、音速も340m/sと遅いため、ドップラーシフトの影響も大きい。図3.7の例では、音波のドップラーシフトの例を示している。例えば、5kHz~10kHzの5kHz幅で変調信号を伝送した場合を考えると、受信機が70m/sで移動した場合、約20パーセント(20000ppm)の周波数のずれが生じる。この場合、伝送周波数帯の下側周波数は5kHzから4kHzにずれ、上側周波数は10kHzから8kHzになる。したがって伝送周波数帯も5kHz幅から4kHz幅と縮まり、平行移動では対処できなくなる。

そのため、音波情報伝送におけるドップラーシフトを補正するためには、リサンプリングによるピッチシフト[55]を利用する。図3.8にしめすように、ドップラーシフトが発生するようなケースでは、ドップラーシフト補正用のパイロット信号を伝送する。このドップラーシフト用のパイロット信号は、OFDM変調信号よりも高い周波数で伝送し、聴覚にあまり影響ないレベルで重畳する。このドップラーシフト補正用のパイロット信号が受信側でどれぐらいずれているかに基づいて、ピッチシフトを行い伝送信号の周波数シフトを補正する。

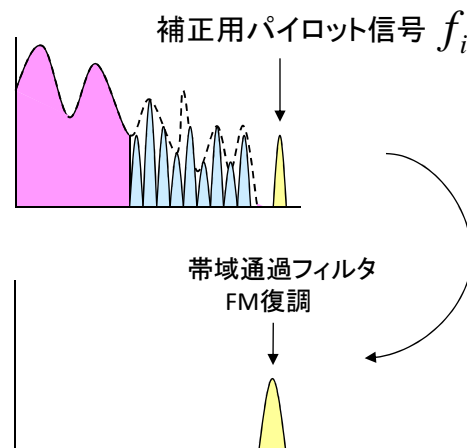


図 3.8: ドップラーシフト補正用パイロット信号

このドップラーシフトの補正は、周波数同期ともいわれる。周波数同期に関しては、ドップラーシフトだけでなく、送信機と受信機の振動子の周波数ずれに対する補



正も含まれる。音波伝送の場合、スピーカ側の D/A 変換とマイク側の A/D 変換の周波数がずれていると、OFDM の直交周波数条件がくずれ、正確に受信できなくなるため、たとえ受信機が動かない場合でも、送信機と受信機間で周波数のずれがある可能性がある場合は、周波数同期が必要となる。TV やラジオ等の放送機や携帯端末等の場合は、通信をするため正確な振動子が入っており、周波数のずれは 10ppm 程度以内である。このような機器同士の通信の場合は周波数同期は無くても十分に通信が可能である。

音波伝送の場合の周波数同期は、前述したとおり、リサンプリングによるピッチシフトで行うが、この方法について詳解する。

例えば、44100Hz サンプリングの信号を受信したときに、1000ppm の周波数ずれがあった場合を考える。この時、受信した信号は、44144Hz サンプリングした信号と同等になる。したがって、44144Hz から 44100Hz にリサンプリングすることによって周波数のずれを補正できる。このリサンプリングには sinc 補間というインターポレーションフィルタを用いて処理できる [55]。このフィルタによって任意のサンプリング周波数変換が可能である。

また、受信機の移動に伴い、ドップラーシフトが生じた場合も同様の方法で補正できる。まず、前述したとおり、ドップラーシフト用のパイロット信号を伝送信号に重畳し、この周波数を  $f_i$  とする。この周波数は受信機側にも既知である。受信側は帯域通過フィルタ (BPF) でパイロット信号のみを抽出し、周波数  $f_i$  で FM 復調をすることによって、周波数のずれを検出できる。受信機が送信機 (スピーカ) から遠ざかって移動しているときの移動速度を  $v(t)$  とし、受信したパイロット信号の周波数 ( $f_i$  からの周波数のずれを含む) を  $f_o$ 、音速を  $V$  とすると、

$$f_o = \frac{V - v(t)}{V} f_i = f_i - \frac{v(t)}{V} f_i$$

となる。受信した周波数  $f_o$  のパイロット信号を周波数  $f_i$  で FM 復調すると時刻  $t$  における周波数のずれ  $z(t)$  が下記の式によって算出できる。

$$(FM\text{demodulation}) \Rightarrow z(t) = -2\pi \frac{v(t)}{V} f_i [\text{radian}]$$

周波数オフセット  $z(t)$  に基づいてリサンプリングを行い、ピッチを変換することによって受信側は送信側で送信したキャリア周波数を正確に補正できる。周波数オフセット  $z(t)$ 、パイロット信号周波数  $f_i$ 、サンプリング周波数  $f_s$  をもちいて、時刻  $t$  におけるサンプリング周波数のずれを算出すると下記の式になる。

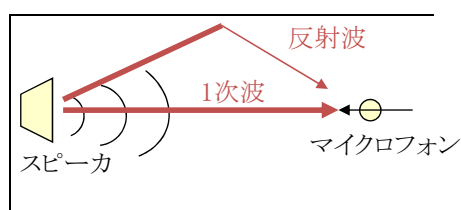
$$(Sampling\text{Deviation}) = \frac{z(t)f_s}{2\pi f_i} [\text{sample}]$$

この方法により、送受信機間の振動子のずれや、ドップラーシフトによる周波数のずれを補正できる。送受信機の振動子のずれは  $z(t)$  における DC(Direct Current) 成分となる。

### 3.2.6 ステレオダイバーシティ方式

一般的に音声や音楽を再生する機器には、ステレオ再生用に左右2つのスピーカが備わっている場合が多い。モノラル信号を再生する場合は、左右のスピーカから同じ信号を再生し、ステレオ信号を再生する場合は左用と右用の信号をそれぞれのスピーカから再生する。ステレオ再生した場合の問題として、音波の干渉の問題がある。

図3.9は、ひとつのスピーカから再生した場合の干渉の具合を示している。室内等で再生した場合には、壁等による反射波の影響で少なからず干渉が生じる。この干渉によって、信号を強めあう周波数もあれば、弱めあってしまう周波数もある。これはスピーカからの距離や反射波の遅延時間によって変わってくる。しかし、ひとつのスピーカから再生するモノラル再生の場合は、スピーカから直接マイクに届く1次波に比べて、反射による遅延波は振幅が弱まっているため、干渉の度合いは少なく、信号が全く受信できないほどの問題にはならない。



反射波が干渉して周波数選択フェーリングが生じるが、1次波と反射波のパワーの差が大きいため、フェーリングの度合いは少ない。

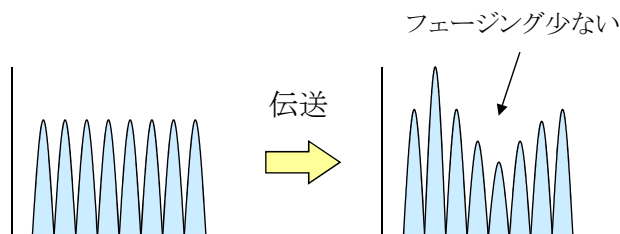
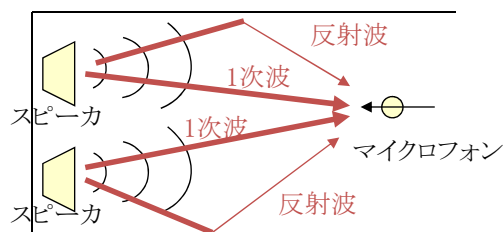


図 3.9: モノラル再生

一方、2つのスピーカから再生するステレオ再生の場合は、図3.10に示すように、反射波だけでなく、2つのスピーカから再生された1次波同士が干渉することになる。これらはほとんど同じ振幅で届くため、強めあう周波数は2倍の信号強度になるが、弱めあう周波数に関してはゼロに近い信号強度になる可能性もあり、受信できなくなるキャリアがでてくる。そのため、音響 OFDM の信号をステレオ再生する場合には、ステレオ再生による干渉を回避できるように送信ダイバーシティという方式を利用する。マイクロフォンが2つあれば、MIMO(Multiple Input Multiple Output)[58][59]等の手法で伝送速度を向上させることもできるが、携帯端末に搭載されているマイクロフォンはたいてい1つであるため、ここでは、2つのスピーカ、1つのマイクロフォンによる送信ダイバーシティ方式について詳解する。



1次波同士が干渉してしまうので、激しい周波数選択フェージングが生じる。

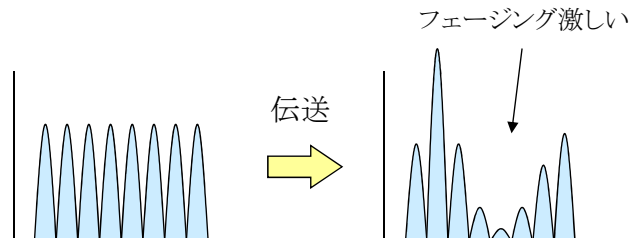


図 3.10: ステレオ再生

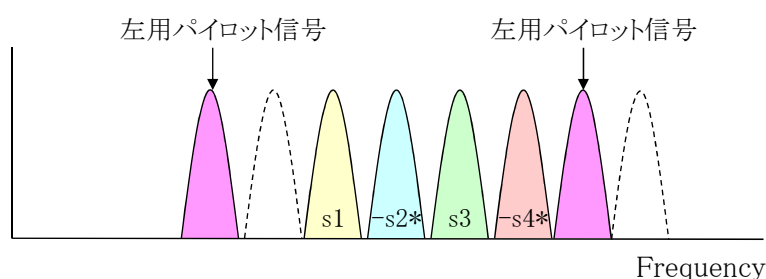
送信ダイバーシティ方式には、空間と時間を利用した時空間ブロック符号化 (STBC: Space Time Block Code)[61] と、空間と周波数を利用した周波数空間ブロック符号化 (SFBC: Space Frequency Block Code)[60] の2種類の方式がある。音響 OFDM では、音波による情報伝送のため、OFDM のシンボル長が長く、シンボル間で時間変動が大きくなる可能性があるため、空間と周波数を利用した周波数空間ブロック符号化を利用する。

図 3.11 に送信ダイバーシティ方式を利用する場合の音響 OFDM のサブキャリア構成の例を示す。まず、送信ダイバーシティ復号をするためには、右チャンネル、左

チャンネルからのそれぞれの伝達関数が必要となるため、右チャンネル、左チャンネル、それぞれに伝達関数を算出するためのパイロット信号を送信する。このパイロット信号は、各チャンネルごとに別々に算出する必要があるため、右チャンネルでパイロット信号を送信する周波数に関しては、左チャンネルでは何も送信しないようにする。逆も同様である。これにより、受信側で既知であるパイロット信号を復号することで、左右それぞれのチャンネルの伝達関数（位相特性や周波数特性）を計算し、パイロット信号のキャリアの間にある信号伝送用の各サブキャリアに対しても、伝達関数を推定できる。

次に信号の配置であるが、周波数空間ブロック符号を利用する場合、2つのキャリアで1つのセットとなる。2つのキャリアと左右のチャンネルの組み合わせによる4つの伝送経路を用いて、2種類の情報（ビット）を送信する送信ダイバーシチには式(3.1)の生成行列  $G$  で示される符号化率1の周波数空間ブロック符号化を行う。

### 左チャンネル



### 右チャンネル

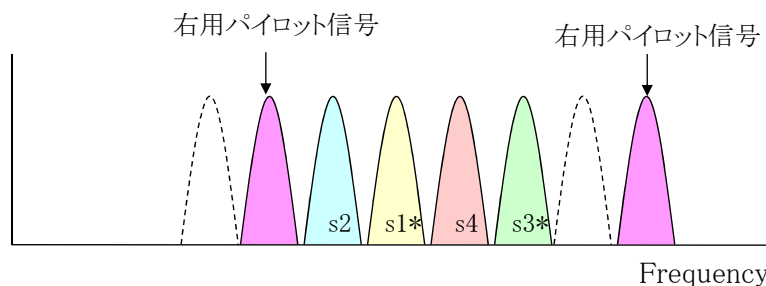


図 3.11: 送信ダイバーシチ方式

$$G = \begin{pmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{pmatrix} \quad (3.1)$$

ここで  $s^*$  は  $s$  の複素共役で、 $s_1, s_2$  はそれぞれ OFDM のサブキャリア  $n, (n+1)$

の送信シンボルを表し、スピーカLからそれぞれ  $s_1, -s_2^*$  を、スピーカRからそれぞれ  $s_2, s_1^*$  を送信する。受信側での信号検出は、サブキャリア  $n, (n+1)$  における受信シンボル  $r_1, r_2$  と、LRそれぞれのスピーカからの伝達関数  $h_L, h_R$  を用いて、式(3.2)~(3.5)により検出できる。

(受信信号)

$$r_1 = h_L s_1 + h_R s_2 \quad (3.2)$$

$$r_2 = h_R s_1^* - h_L s_2^* \quad (3.3)$$

(送信ダイバーシチ復号)

$$\begin{aligned} \hat{s}_1 &= h_L^* r_1 + h_R r_2^* \\ &= h_L^* (h_L s_1 + h_R s_2) + h_R (h_R^* s_1 - h_L^* s_2) \\ &= (|h_L|^2 + |h_R|^2) s_1 \end{aligned} \quad (3.4)$$

$$\begin{aligned} \hat{s}_2 &= h_R^* r_1 - h_L r_2^* \\ &= h_R^* (h_L s_1 + h_R s_2) - h_L (h_R^* s_1 - h_L^* s_2) \\ &= (|h_L|^2 + |h_R|^2) s_2 \end{aligned} \quad (3.5)$$

$\hat{s}$  は、送信ダイバーシチ復号で分離したそれぞれのシンボルであり、これにより  $s_1, s_2$  を抽出でき、送信ダイバーシチ効果を得ることができる。2つのスピーカから単純に同じ信号を送信した場合、干渉によって受信信号が抽出できなくなる地点が多数存在するが、送信ダイバーシチを用いる場合、どの地点においても受信信号電力を高めることができる。高周波数の音波は指向性が鋭いため、ステレオの送信ダイバーシチで再生することにより、伝送信号の届く範囲を広げることができる。

図3.12は、送信ダイバーシチの効果を示すシミュレーションの図である。左側は、送信ダイバーシチを用いずに単純に同じ信号を左右のスピーカから伝送した場合の例である。グラフは音圧レベルを表しているが、左右のスピーカからの信号が干渉することによって、受信信号を弱めあってしまう場所がでてくる。これに対して、右側の例は、送信ダイバーシチを利用した場合で、左右からそれぞれ送信ダイバーシチ符号化した信号を伝送する。この場合、左右から伝送される信号は同じものではないため、干渉によって信号が消えてしまうことはなくなる。また送

送信ダイバーシティ復号によって、左右それぞれのスピーカからの信号を受信信号を強度するために復号できるため、受信信号レベルを高めることができる。これにより、送信ダイバーシティを用いることで、どの場所においても安定して信号を受信できるようになる。

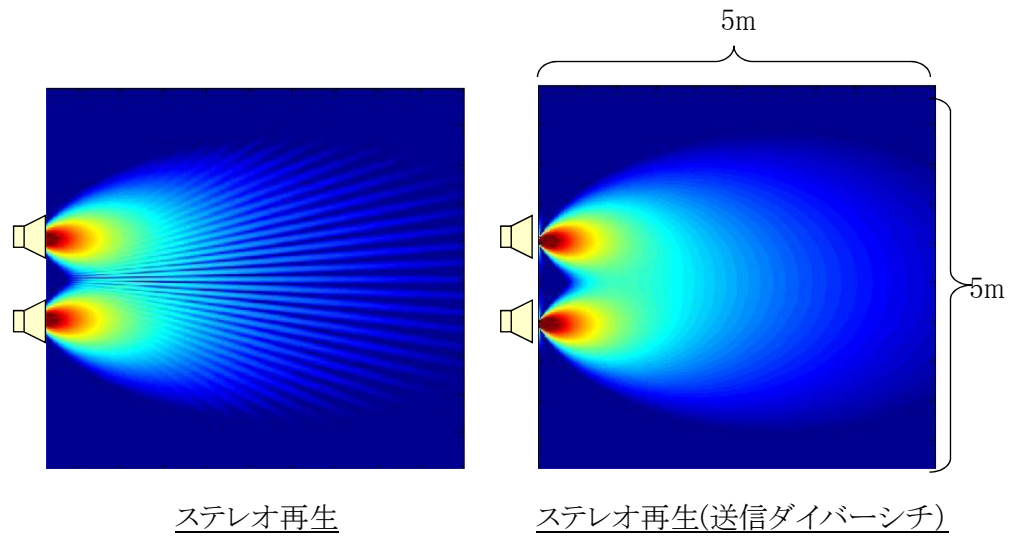


図 3.12: 送信ダイバーシティの効果

## 第4章 音響 OFDM システムの設計と実装

本章では、前章で提案した音響 OFDM 技術を利用して、実際に近距離情報通信システムを設計し、実装できるレベルまで説明する。本章は、筆者論文 [2] の内容を詳解に説明する。通信システムとして実装するためには、OFDM の変復調だけでなく、誤り訂正符号、インタリーブ、データフレーム構成等を設計する必要がある。本章では、これらの設計に必要な要素技術について説明し、システムの実装例についても説明する。

### 4.1 情報源符号化・復号化

情報源符号化では、入力される文字や ID 等のデータをビット列に変換する。URL やメールアドレス等に関しては ASCII コードに基づいて、ビット列に変換する。音響 OFDM では固定長のデータフレームを生成して伝送するため、ビット列がデータフレーム長に満たない場合は、埋め草ビットを付加（パディング）し、データフレーム長のビット列を生成する。情報源復号化では、データフレームのビット列から、埋め草ビットを除去し、ASCII コード等に基づいて文字や ID 等のデータに変換する。

### 4.2 データフレーム分割・再構築

音響 OFDM では、データフレームという単位でデータを伝送する。これは OFDM のあるシンボル数をひとつの区切りとしてデータフレームとし、このデータフレーム内で誤り訂正符号等を施して伝送を行う。伝送データがひとつのデータフレームのサイズで全て伝送できれば、そのデータフレームを繰り返し伝送すれば受信側はどこから受信しても 1 フレーム分受信すればデコードが可能である。しかし、伝送データがある程度大きい場合、複数のデータフレームに分割して伝送する必要がある。この複数のデータフレームを順番に繰り返し伝送し、受信側も全て正常に受信

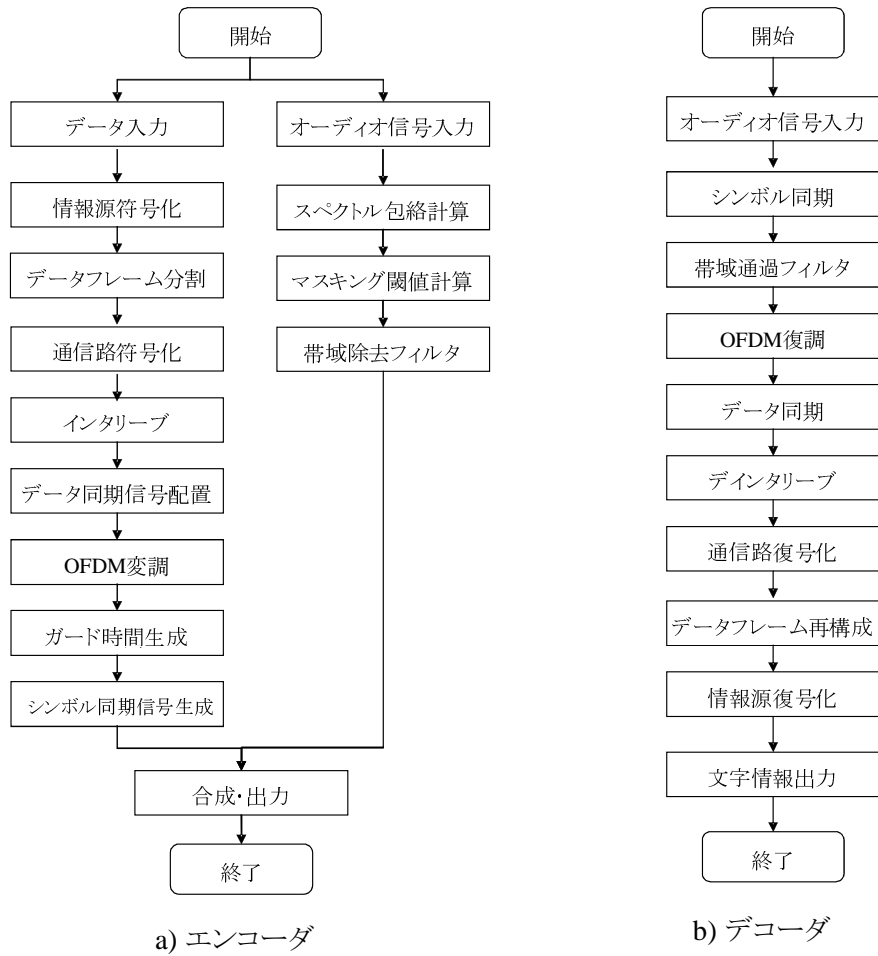


図 4.1: エンコード・デコード手順



できれば、分割したデータフレーム分のデータフレームを受信すれば復号ができるようになる。しかし、途中で受信状況が悪くなり、ひとつでもデータフレームの受信がエラーになると、そのデータフレームが再び伝送されるまでデコードができなくなる。これは、受信効率が悪くなるため、音響 OFDM ではデータフレーム単位でも消失訂正符号を用いる。

本節では、伝送データを固定長サイズのデータフレームに分割して伝送するための送信側における分割処理、受信側における再構成処理を示す。分割データの消失を訂正するために、Reed-Solomon 符号 [62][63][64] による冗長フレームを生成する。消失訂正符号としての Reed-Solomon 符号では、 $k$  個のフレームから  $(n - k)$  個の冗長フレームを生成し、合計  $n$  個のフレームのうち、任意の  $k$  個のフレームを受信できれば、元のフレームを復元できるという特徴がある。図 4.2 に分割処理および冗長フレーム生成の概念を示す。

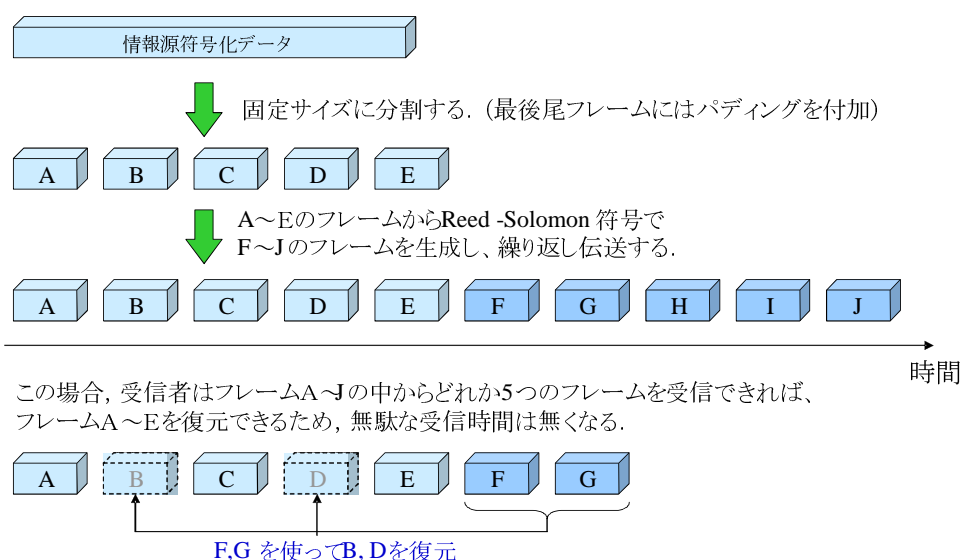


図 4.2: マルチフレームの消失訂正

まず、データを固定長サイズに分割する。余ったデータもパディング情報を付加して固定サイズに揃えてフレームA~Eを生成する。次にデータフレームA~Eから Reed-Solomon 符号で冗長フレームF~Jを生成する。Reed-Solomon 符号のアルゴリズムについては、付録に示す。A~Jのフレームの各々に1バイトのデータフレームヘッダを付加し、再構成するための情報を格納する。データフレームヘッダについては図3にて後述する。受信側では、A~Jのフレームの中から分割フレーム

数に相当する数（この例では5個）の任意のフレームを受信すればA～Eのデータフレームを復元できる。この例では、A、C、E、F、GのフレームからB、Dのフレームを復元してA～Eのフレームを再構成する。

受信したデータフレームが Reed-Solomon 符号における何番目のフレームかを識別できる必要がある。図 4.3 にデータフレームヘッダの構成を示す。データフレームヘッダは1バイトで構成されフレームの先頭に付加される。1バイトの領域は、予約ビット (R:1ビット)、FEC フラグ (F:1ビット)、フレーム数 (3ビット)、シーケンス番号 (3ビット) で構成される。予約ビットは将来の拡張のために確保し、通常は常に0を格納する。FEC フラグは、データフレームが分割フレームか冗長フレームかを判別するためのビットであり、分割フレームの場合は0、冗長フレームの場合は1を格納する。フレーム数 (3ビット) には、分割フレーム数を挿入する。最大7分割まで可能になる。シーケンス番号 (3ビット) には、分割フレームの場合は先頭から何番目のフレームかを示し、冗長フレームの場合には Reed-Solomon 符号の符号化に利用する生成行列 (付録に記載) の何行目の計算で生成されたかを示す。したがって最大で、データフレームが7個、冗長フレームが7個の合計14個のフレームを伝送することが可能である。

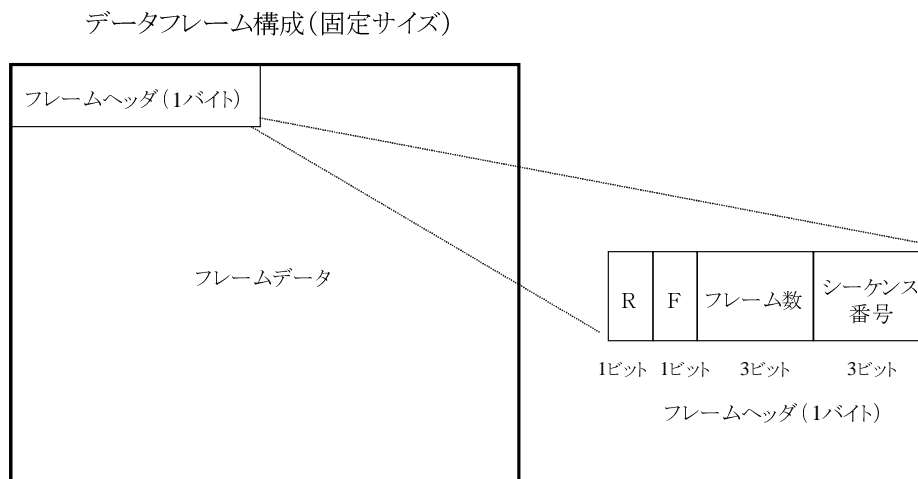


図 4.3: マルチフレームの構成

受信側における再構成では、受信したフレームを保存しておき、分割フレームおよび冗長フレームの合計数が、フレームヘッダのフレーム数に格納されている数に達した時点で、Reed-Solomon 復号 (付録に記載) をし、分割フレームを復元する。

復元した分割フレームを再構成し、情報元符号化データを生成する。

### 4.3 通信路符号化・復号化

データフレームの伝送において、ビット誤りを検出および訂正する処理が必要になる。音響 OFDM システムでは、誤り検出に CRC[65][66]、誤り訂正符号に畳み込み符号 [67][68]、誤り訂正復号に軟判定ビタビ復号 [69][70] を利用する。

CRC ビットは付加する CRC ビット長に応じて表 4.1 で示す生成多項式を用いて生成する。符号化では、情報源符号化された情報ビット列に対して生成多項式のビット列で除算した剰余ビット列を後方に付加する。誤り検出処理では、受信した情報ビット列を生成多項式のビット列で除算し、余りが存在したら情報ビット列に誤りがあると判定する。詳細なアルゴリズムは、付録に示す。

表 4.1: 生成多項式

CRC 長	生成多項式
6 ビット	$x^6 + x + 1$
7 ビット	$x^7 + x + 1$
8 ビット	$x^8 + x^4 + 1$
9 ビット	$x^9 + x^4 + 1$
10 ビット	$x^{10} + x^3 + 1$

誤り訂正の符号化手順としては、まず CRC ビットを付加した情報ビット系列に対して、畳み込み符号の終端コードを付加する。畳み込み符号の拘束長を  $K$  とすると、 $K-1$  ビットの 0 を終端に付加することになる。この終端コードを付加したビット系列を畳み込み符号器に入力し、出力されたビットを整理して符号化ビット系列を生成する。畳み込み符号器の生成多項式には、拘束長、符号化率に応じて、表 4.2 に示す生成多項式を用いる。詳細なアルゴリズムは、付録に示す。

誤り訂正の復号化手順としては、受信した信号系列に対して、符号化で行われたインタリーブを元に戻すデインタリーブを行う。インタリーブ・デインタリーブについては、次節で記述する。このデインタリーブした信号系列に対して軟判定ビタビ復号を行う。まず、OFDM の各サブキャリアの D-BPSK 変調を復調する際に、軟判定ビタビ復号に用いるブランチメトリックを計算する。ブランチメトリックの計算には次式を用いる。

$$\lambda(1) = \text{Re}[z_n \cdot z_{n-1}^*]$$

表 4.2: 生成多項式 (畳み込み符号)

拘束長	符号化率	生成多項式
4	1/2	$1 + D + D^3, 1 + D + D^2 + D^3$
4	1/3	$1 + D^2 + D^3, 1 + D + D^3, 1 + D + D^2 + D^3$
5	1/2	$1 + D^3 + D^4, 1 + D + D^2 + D^4$
5	1/3	$1 + D^2 + D^4, 1 + D + D^3 + D^4, 1 + D + D^2 + D^3 + D^4$
6	1/2	$1 + D + D^3 + D^5, 1 + D^2 + D^3 + D^4 + D^5$
6	1/3	$1 + D^3 + D^4 + D^5, 1 + D^2 + D^4 + D^5, 1 + D + D^2 + D^3 + D^5$

$$\lambda(-1) = -Re[z_n \cdot z_{n-1}]$$

ここで、 $z_n$  は、時刻  $n$  におけるキャリアの振幅を  $(x^2 + y^2)^{0.5}$  で、絶対位相を  $\tan^{-1}(x/y)$  で表せる複素数  $x + jy$  である。パスメトリックはブランチメトリックの合計となり、パスメトリックの最も大きいパスが軟判定ビタビ復号の出力ビット系列となる。詳細なアルゴリズムは、付録に示す。この出力ビット系列から畳込み符号の終端コードを削除したビット系列に対して、CRC の誤り検出を行い、誤りが無ければ正常な情報ビット系列として出力する。

## 4.4 インタリーブ

OFDM 変調では、符号化系列を時間方向と周波数方向に配置するが、時間方向もしくは周波数方向に順番に配置すると、ある時間もしくはある周波数においてフェージングが生じた際に、その部分にバースト的なエラーが生じてしまうため、インタリーブ [73][74] でビットの配置を変えることによって、誤り耐性を向上させることができる。音響 OFDM システムでは、以下に示す 2 つのインタリーブ (配置変換) を組み合わせてランダム性を高める。下記の例では縦軸が周波数軸で横軸が時間軸である。インタリーブ前のビット配置は、符号化ビット列を周波数軸方向に低周波数から高周波数方向に順番に並べ、最高周波数までビットを配置した後、次の時間スロットの低周波数から高周波数方向に順番に並べていく。

### インタリーブ 1

このインタリーブでは、奇数番目のシンボルを先頭から並べ、偶数番目のシンボルを後方から並べる配置変換を行う。

表 4.3: インタリーブ前    表 4.4: インタリーブ後

1	6	11	16
2	7	12	17
3	8	13	18
4	9	14	19
5	10	15	20

1	11	20	10
3	13	18	8
5	15	16	6
7	17	14	4
9	19	12	2

## インタリーブ 2

このインタリーブでは、符号化系列を4シンボルおきに配置する。最後尾まで配置したら、先頭の2番目のシンボルに戻り、再び4シンボルおきに配置する。同様に、先頭の3番目のシンボル、4番目のシンボルに戻りながら、4シンボルおきに配置する。

表 4.5: インタリーブ前    表 4.6: インタリーブ後

1	6	11	16
2	7	12	17
3	8	13	18
4	9	14	19
5	10	15	20

1	7	13	19
6	12	18	5
11	17	4	10
16	3	9	15
2	8	14	20

初期のビット配置は上記の例の通り、符号化ビット列を周波数軸方向に低周波数から高周波数方向に順番に並べ、最高周波数までビットを配置した後、次の時間スロットの低周波数から高周波数方向に順番に並べていく。このビット配列に上記のインタリーブを、インタリーブ1、インタリーブ2の順番で適用する。インタリーブ2では、ビット列を31シンボルおきに配置変換する。復号には、上記のインタリーブの逆変換であるデインタリーブをインタリーブ2、インタリーブ1の順番で行う。詳細なアルゴリズムは付録に示す。

## 4.5 データフレーム同期信号・同期処理

音響 OFDM システムにおけるデータフレームは、図 4.4 に示す構造になる。時間軸方向に 31 の OFDM フレームと、周波数軸方向に 37 のサブキャリアから構成される。最低周波数の 1 サブキャリア（図では、サブキャリア 0）にデータ同期用の符号系列を配置する。その上の周波数の 36 のサブキャリア（図 4.4 では、サブキャ

リア1~36) にデータのビット系列を配置する. 1つのサブキャリア, 1つの OFDM フレームで1ビットの情報を配置するため, 1つのデータフレームで 36x31 の 1116 ビットを配置する.

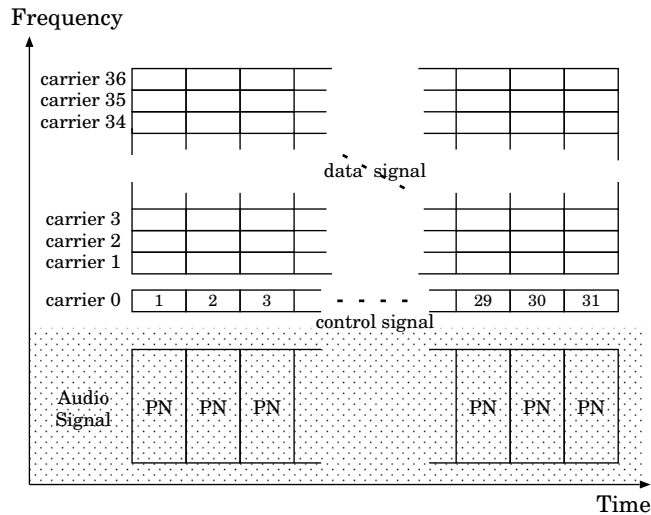


図 4.4: フレーム構成

データフレームの先頭を識別するために, データ同期処理が必要となる. また, 符号系列の種類によってシングルフレームの繰り返し伝送か, マルチフレームの伝送かを識別できるようにする. 本規格では以下に示す2種類の符号系列でシングルフレーム伝送かマルチフレーム伝送かを識別する.

【シングルフレーム伝送用】

0,0,0,0,0,1,1,1,0,0,1,0,0,0,1,0,1,0,1,1,1,1,0,1,1,0,1,0,0,1,1

【マルチフレーム伝送用】

0,0,0,0,1,0,0,1,0,1,1,0,0,1,1,1,1,1,0,0,0,1,1,0,1,1,1,0,1,0,1

どちらも ( $n = 31, k = 6, d = 15$ ) の BCH 符号 [71][72] で符号化された符号長 31 の符号系列である. BCH 符号は巡回符号であるため, 系列が巡回していても符号語になっており, 誤り訂正の復号が可能である. このデータ同期用の符号系列は繰り返し伝送されるため, 受信側でどのビットから受信を開始しても 31 ビット分の符号系列を受信すれば誤り訂正の復号が可能である. 誤り訂正の復号には以下の生成符

号系列を用いる。ここで、符号系列の先頭ビットの定義を決めることで、受信側でデータ同期用の符号系列が何ビット巡回しているかが判別できるようになる。先頭ビットの定義は符号系列の中で0が最も長く続く箇所の最初の0を符号系列の先頭ビットとする。上記に記載の符号系列は先頭ビットから記載されている。送信側で、この先頭ビットとデータフレームの先頭を合わせて配置しておくことで、受信側ではデータ同期用符号系列が何ビット巡回しているかを検出し、データフレームの先頭を識別できるようになる。BCH符号の誤り訂正復号には以下の生成符号系列を用いる。

【BCH 符号生成符号系列】

1,1,1,0,0,1,0,0,0,1,0,1,0,1,1,1,0,1,1,0,1,0,0,1,1,0,0,0,0,0

シングルフレーム伝送の場合、データは分割されず、常に同一のデータフレームが伝送されるため、シーケンス番号や冗長フレームが不要である。したがって、シングルフレーム伝送の場合は、データフレーム内にフレームヘッダは付加されずにデータだけが格納される。

## 4.6 スペクトル法絡計算

入力されたオーディオ信号をフーリエ変換し、オーディオ信号の周波数スペクトルを計算する。フーリエ変換は、OFDMのシンボル長と同じサンプル数の時間信号に対して適用する。本規格では、OFDMのシンボル長が44.1kHz サンプリングで1024サンプルなので、フーリエ変換も1024サンプルの時間信号に対して行う。フーリエ変換すると0~44.1kHzまでの1024種類の周波数情報（振幅、位相）が得られるが、ナイキスト周波数で折り返して同じ情報になるため、情報量としては、0~22.05kHzまでの512種類の周波数情報となる。ここで算出した周波数情報の振幅情報は、OFDM変調やマスクング閾値計算に用いる。

## 4.7 OFDM変調

インタリーブしたビット配列に基づき、サブキャリアをD-BPSKで変調する。D-BPSK変調では、ひとつ前のOFDMシンボルの同サブキャリアにおける位相との相対位相差で情報ビットを識別できるようにする。伝送するビットが0の場合には

前シンボルの位相を $\pi$ シフトさせる。伝送するビットが1の場合には前シンボルと同位相にする。エンコードの最初のOFDMシンボルに関してはランダムなビット列(任意)を配置してよい。OFDMのデータフレームは繰り返し伝送されるので、2周目以降のデータフレームの先頭フレームは1つ前のデータフレームの最終フレーム(31フレーム目)を参照して位相を決定する。次にOFDM変調信号がオーディオ信号に調和するように各サブキャリアの振幅を調節する。振幅の調節にはオーディオ信号から計算したスペクトル包絡情報を用いる。まず、オリジナル音源のオーディオ信号を解析しスペクトル包絡情報を求める。その後、帯域阻止フィルタでOFDM伝送周波数帯域の成分を除去する。次に、入力されたデータに基づきOFDM伝送周波数帯域のサブキャリアを変調したOFDM変調信号を生成する。このOFDM変調信号のサブキャリアのパワーを、オリジナル音源のスペクトル包絡に合わせて調節し、振幅調整した信号を生成する。最後に、帯域除去オーディオ信号とOFDM伝送周波数帯域信号を合成して、合成オーディオ信号を生成する。これにより、OFDM変調信号を聴覚に不快にならないように変形して音声・音楽に重畳できる。ノイズとの振幅比を維持するために、スペクトル包絡でパワーが少ないサブキャリアについては、サブキャリアの振幅をスペクトル包絡以上に設定しておくことで受信耐性を高くできる。また、音質の劣化が許容できるのであれば、スペクトル包絡にパワーを調節するステップを省略して、全て同じ振幅のサブキャリアで伝送しても良い。

## 4.8 ガード時間生成

OFDMにはガード時間を挿入することで反射波等によるマルチパス干渉に対処できるという特徴もある。図4.5の(a)に示すように、各OFDMフレームはOFDMシンボル区間とガード時間の区間で構成される。ガード時間の区間はデータシンボルの後方を複製(Cyclic prefix)して生成される。これにより、ガード時間とOFDMフレームは位相が連続になり、この区間長以内で遅延波が重なっても直交性を保つことができる。このガード時間により、スピーカの減衰振動や反射波による遅延波にも対応でき、ロバストなデータ伝送が可能になる。図4.5の(a)で伝送性能としては十分であるが、ガード時間と前OFDMフレームの境界は位相が不連続になり、音響OFDMでは、これらの伝送信号が人間の耳に聞こえるため、位相不連続な部分が不快な音に聞こえてしまう場合がある。そのため、音響OFDMでは、図4.5の(b)に示すように後方にもガード時間を挿入する。後方のガード時間はOFDMシンボルの前方をコピーして生成する。この後方ガード時間と、ひとつ後ろのOFDMフレームの前方ガード時間をクロスフェードさせてオーバーラップすることで、フレーム



間の位相の不連続を軽減する。本規格では、OFDM シンボルが 44.1kHz で 1024 サンプル、ガード時間が 1008 サンプルで OFDM フレーム長は 2032 サンプルとなる。後方ガード時間のサンプル数は任意であるが 200 サンプル程度を推奨する。また、後方ガード時間の代わりにマスキング音を挿入して位相不連続点で生ずるノイズを時間マスキングすることもできる。このマスキング音として、単一または複数の周波数の正弦波を挿入し、この周波数をフレーム境界毎に変化させることによって、メロディを構成することもできる。このメロディを合図音として、伝送信号が入っていることをユーザに気づかせるために使うこともできる。

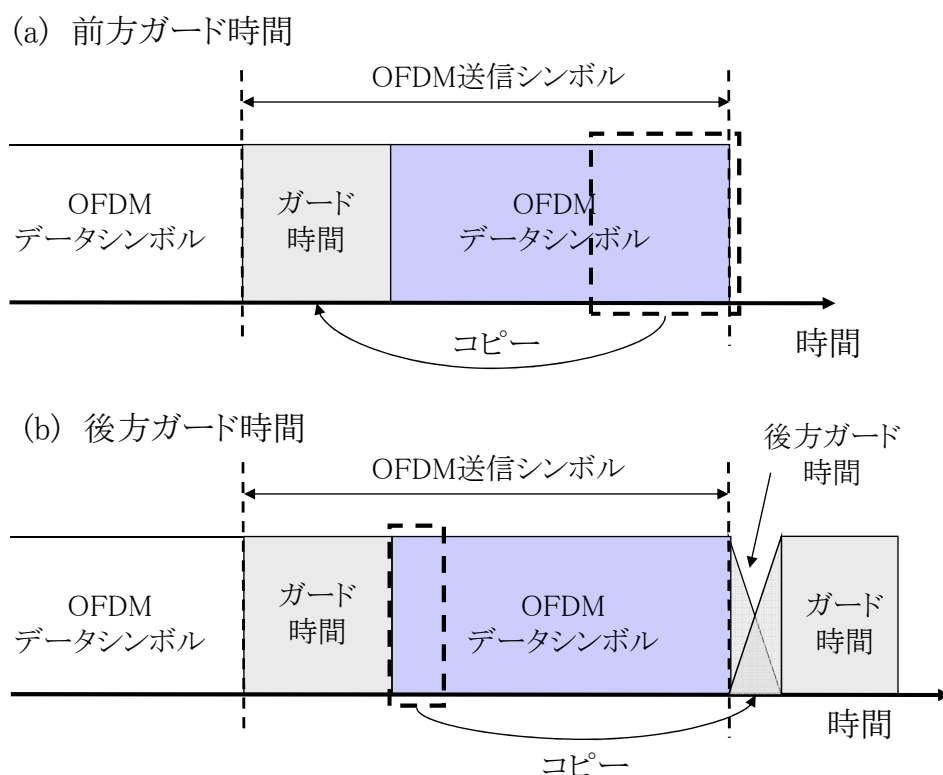


図 4.5: 音響 OFDM シンボル構成

## 4.9 シンボル同期信号生成・シンボル同期

シンボル同期信号の生成には、M 系列符号を PSK 変調して生成する。44.1kHz サンプルングの場合に 16 サンプルで 1 つの符号を変調すると、チップレートが 2756Hz の同期信号となる。したがって、中心周波数が 2756Hz で、0~5512Hz に広がる信

号となる。同期信号の拡散率を 127 とすると、 $16 \times 127$  の 2032 サンプルで 1 つの OFDM シンボルを形成することになる。この同期信号を前述したマスキング閾値以下の振幅レベルに調節して、オーディオ信号に重畳する。

伝送システムによって OFDM 信号の周波数帯が異なる場合もある。例えば AM 放送での適用を考えた場合、OFDM 信号の周波数は 7kHz 以下程度に設定する必要がある。受信側で複数の伝送システムに対応するために、OFDM 信号の周波数帯をフレーム同期信号から検出する。フレーム同期信号の拡散に用いる PN 系列には表 1 で示す生成多項式で生成される M 系列を用いるが、別の M 系列を使うことで別の OFDM 周波数帯を使うこともできる。受信側で複数の M 系列で相関計算し、最大相関値が最も高くなる M 系列を選択して、その M 系列に対応した周波数帯で OFDM 信号を復調する。

## 4.10 OFDM 復調

OFDM 復調は、帯域通過フィルタで抽出したオーディオ信号に対して、フレーム同期で識別した OFDM フレームの先頭に基づいて OFDM シンボル区間を取り出し、OFDM シンボル区間の時間信号をフーリエ変換して周波数情報を取得する。得られた周波数情報のうち、OFDM のサブキャリア周波数にあたる周波数情報に注目する。誤り訂正の復号に硬判定したビット列を用いるのであれば、ひとつ前の OFDM シンボルの同サブキャリアの位相と比較して、位相差が  $-\pi/2$  から  $+\pi/2$  であれば 1、それ以外であれば 0 とする。誤り訂正の復号に、前述した軟判定のメトリックを利用する場合は、ひとつ前の OFDM シンボルの同サブキャリアの振幅・位相と現シンボルの振幅・位相を用いて受信ビット系列の軟値を算出する。軟値の算出式は、通信路符号化・復号化で記載した通りである。

## 4.11 音響 OFDM システムの実装

本節で説明した音響 OFDM システムにおいて、検証・実験に用いるシステムパラメータを表 4.7 に示す。このシステムパラメータによる音響 OFDM のエンコーダ・デコーダをソフトウェアとして実装した。このシステムによる性能評価および音質評価については、次章にて記載する。

まず、信号を生成するための音声信号処理をするサンプリング周波数は 44100Hz である。OFDM 信号は、6400Hz~8000Hz に挿入され、1 キャリアの周波数幅は 43Hz である。これは OFDM シンボル長 1024 サンプルを逆フーリエ変換して生成するた

表 4.7: 音響 OFDM システムパラメータ

パラメータ項目	値
サンプリング周波数	44.1 kHz
OFDM フレーム長	2032 サンプル
シンボル長	1024 サンプル
ガード時間	1008 サンプル
サブキャリア数	36
同期信号周波数帯	0 - 5512.5Hz
OFDM 信号周波数帯	6400 - 8000 Hz
同期信号拡散率	127
同期信号チップレート	2756Hz
同期信号生成多項式	$x^6 + x + 1, (x^6 + x^5 + x^2 + x + 1)$
情報系列長	360
誤り訂正符号化率	1/3
誤り訂正符号 拘束長	6
誤り検出用 CRC 長	7
誤り検出用 CRC 多項式	$x^7 + x + 1$

め、ひとつの直交周波数が 1024/44100 で約 43Hz となるためである。この 43Hz のキャリアを 36 個並列にならべて伝送する。実際には同期信号用のキャリアも含めて 37 キャリアを伝送する。ガード時間は、TV 視聴を想定して、最も長い有効な反射波の遅延時間として約 20ms とし、ガード時間のサンプル数を 1008 サンプルとした。したがって、ガード時間も含めた OFDM フレーム長は 2032 サンプルである。この 2032 サンプルに拡散率 127 のシンボル同期信号を低域音響信号に重畳する。1 チップのサンプル数は 16 サンプルとし、127 チップの M 系列符号を PSK 変調して 2032 サンプルのシンボル同期信号を生成する。

OFDM で伝送するデータフレームは 36 キャリア× 31 シンボルで 1116 ビットを 1 データフレームとして伝送する。このデータフレームの先頭を識別するために、37 番目のキャリアを利用して前述したシングルフレーム用の BCH 符号語およびマルチフレーム用の BCH 符号語のビット列を伝送し、データフレームの先頭を識別する。1116 ビットのビット列は、符号化率 1/3 の畳み込み符号で符号化したビット列であり、情報系列のビット長は 1116 ビットの 1/3 の 372 ビットとなる。このうち、5 ビットは畳み込み符号の終端符号として利用され、さらに 7 ビットは誤り検出用の CRC ビットに利用されるため、実際に情報として伝送できるビット長は 360 ビットとなる。この 360 ビットを 31 フレーム（約 1.4 秒）で伝送するため、伝送速度としては、約 250bps となる。

このパラメータによる音響 OFDM システムをソフトウェアとして実装し、任意の文字列を音波によって伝送できることを確認した。詳細なパフォーマンスについては、次章にて解説する。

## 4.12 実用化への課題

音響 OFDM の応用として考えられるのは、スピーカを情報発信源とし、携帯端末で音声を録音して情報を取得するという形態のアプリケーションである。例えば、テレビやラジオの音声に番組の Web サイトの URL 情報を重畳して伝送し、携帯端末で録音してその情報を抽出できれば、番組の Web サイトに自動的にアクセスできるようになる。ここで受信側としては、近年、携帯端末としてスマートフォンが普及しつつあるため、スマートフォンのアプリケーションとして実装できる必要がある。音響 OFDM のメリットの一つとして、特殊なハードウェアを必要とせずに実現できるという点を挙げたため、受信信号処理の全てはソフトウェアで実装する必要がある。しかし、フィルタや FFT 等の演算は計算量が大きいため、スマートフォンではリアルタイム処理ができない可能性がある。そのため、計算量を小さく実装するために、以下の処理を施した。

### 固定小数点演算の利用

フィルタ演算や FFT の演算は小数点演算がでてくるため、通常は浮動小数点演算を用いて処理をする。しかし、浮動小数点演算のかけ算やわり算は処理時間が固定小数点演算と比べて大きくなるため、スマートフォンへの実装については、固定小数点演算を利用した。固定小数点演算の方法は、64 ビットの整数型変数 (INT64) を用い、32 ビットの整数同士の掛け算を実行できるようにした。またフィルタの係数等の小数値に関しては、16 ビットシフトさせて小数部分を整数部分にシフトさせてから演算を行うことで、整数演算で実行できるようにした。これらの処理により、浮動小数点演算で実行する時よりも固定小数点演算で実行した場合の方が、全体の処理時間が約 1/3 に短縮できた。演算処理能力の低いスマートフォンや携帯端末等では、固定小数演算を利用しないとリアルタイムでの動作が難しいものもある。

### サンプリング周波数

音響 OFDM の信号生成は、前章で述べた通り、44.1kHz サンプリングで行う。受信側でも 44.1kHz サンプリングで録音して処理をできれば精度がよくなるが、演

算量が大きくなるため、半分の 22.05kHz サンプリングで録音、処理を行うように実装した。サンプリング周波数を落とすことで、ナイキスト周波数も 11025Hz に下がるが、音響 OFDM の伝送信号は 6.4kHz~8kHz 付近に重畳するため、伝送信号が劣化することはない。サンプリング周波数を半分にすることで、処理時間を約 1/4 に短縮できた。

## ハーフバンドフィルタの利用

受信側で、録音信号から伝送信号を抽出する際に、低域音響信号と高域 OFDM 信号を分離する必要がある。この分離には、直線位相型のフィルタを用いる。ここで、このフィルタにハーフバンドフィルタを用いる。ハーフバンドフィルタとは、ナイキスト周波数の半分の周波数で分離するフィルタであり、ハーフバンドフィルタの特徴としては、フィルタ係数の半分がちょうどゼロになることである。つまりフィルタの演算量が半分になることになる。これを利用して演算量の削減を行った。サンプリング周波数は 22050Hz で、ナイキスト周波数は 11025Hz であるため、このハーフバンドフィルタは、5512.5Hz で信号を分離することになる。音響 OFDM の伝送信号は、6.4kHz~8kHz 付近で配置されているため、ハーフバンドフィルタで分離できる。

上記の処理を施すことによって、受信側の信号処理の演算量を大幅に削減できる。実際に Android 4.0 のスマートフォン端末用に音響 OFDM のデコーダソフトを実装し、録音しながらのデコード処理を実現できた。携帯端末や小型の機器に実装する場合には、このような処理方法で実装する必要がでてくる。

次に、再生側の課題としては、TV やラジオの放送波を利用する場合、放送法による制限が課題となってくる。音声の中に重畳した OFDM 信号がデータ放送として識別されると、データ放送としての放送法が適用されることになり、通常の音声・映像の放送波で伝送できない可能性がある。また、郊外で音響 OFDM の信号を再生する場合には、音圧レベルを適正にする必要もでてくる。高周波の音は、聴覚には知覚できなくなってくるが、音圧としては 70dB(A) 以下にする必要があるため、それらを考慮してエンコード、再生器のボリュームを調整する必要がある。

## 第5章 音響 OFDM の性能評価

本章では、提案方式である音響 OFDM の原理による音質の評価や、前章で記載した音響 OFDM システムによる伝送性能の評価について記載する。本章は、筆者論文 [1] の性能評価部分を詳解に説明する。音響 OFDM の新規性であり最も特徴的な点として、OFDM 信号をスペクトル包絡に合わせてサブキャリアを形成するという点がある。この効果の評価として、OFDM 信号を加工しないフラットなスペクトルの OFDM 信号を重畳した場合と比較して、音質と伝送性能の両方を検証する。音質としては、音響 OFDM の OFDM 信号の加工によって、音質の劣化がどれくらい抑えられるのかを検証する。また、伝送性能としては、フラットな OFDM 信号に比べて、伝送性能が劣化するのかわからないのかを検証する。フラットな OFDM 信号を音源に重畳したものを Flat OFDM と呼ぶことにする。

### 5.1 音質評価試験

音響 OFDM の音質劣化を評価するために、主観評価試験を行った。主観評価試験には、MUSHURA[75] という手法を用いた。この評価試験では、オリジナル音源に加えて、3.5kHz および 7kHz で帯域制限した音源と 128kbps で符号化した MP3 音源を比較音源として使用した。音響 OFDM (Acoustic OFDM と記載) のスペクトル包絡に合わせて OFDM 信号の音質特性を評価するために、スペクトル包絡に合わせて全てのキャリアの振幅をフラットにして重畳する OFDM 信号 (Flat OFDM と記載) も比較対象として評価試験を行った。Flat OFDM の OFDM 信号の平均振幅は、音響 OFDM と同様に設定した。評価試験に使用した音源は、ロック調、ポップ調、ジャズ調、スピーチの 4 種類を用い、それぞれに対して、音響 OFDM, Flat OFDM, MP3, LPF(3.5kHz 制限, 7kHz 制限) の音源を作成して評価を行った。評価者は 20 名で行い、それぞれの音源に対して 0~100 点のスコアをつけた。図 5.1 に、この主観評価試験の結果を示す。

オリジナル音源は、どの種類の音源でも最も高いスコアで、ほぼ 100 点に近いスコアがつけられた。これに対して、音響 OFDM は、音源の種類によってスコアが異なり、ロック調の音源については MP3 とほぼ同じレベルのスコアがつけられた。続

いてポップ調，ジャズ調の音源とスコアが低くなるが，Flat OFDM や LPF(3.5kHz 制限，7kHz 制限) の音よりは高いスコアがつけられた．最後にスピーチに関しては，Flat OFDM や LPF(3.5kHz 制限，7kHz 制限) の音とほぼ同じスコアになった．これはスピーチの音源は，元々高域に音がほとんど無いため，音響 OFDM のスペクトル包絡による音質劣化抑制効果がほとんど無いためと考えられる．逆にロック調の音源は，高域にもたくさんの音が存在するため，音響 OFDM のスペクトル包絡による音質劣化抑制効果が十分に発揮されていると考えられる．このように，音源の種類によるが，比較的音が多く鳴っている（多くの周波数スペクトルが存在する）音源に関しては，フラットな OFDM 信号に比べて音質の劣化を抑制できることがわかった．

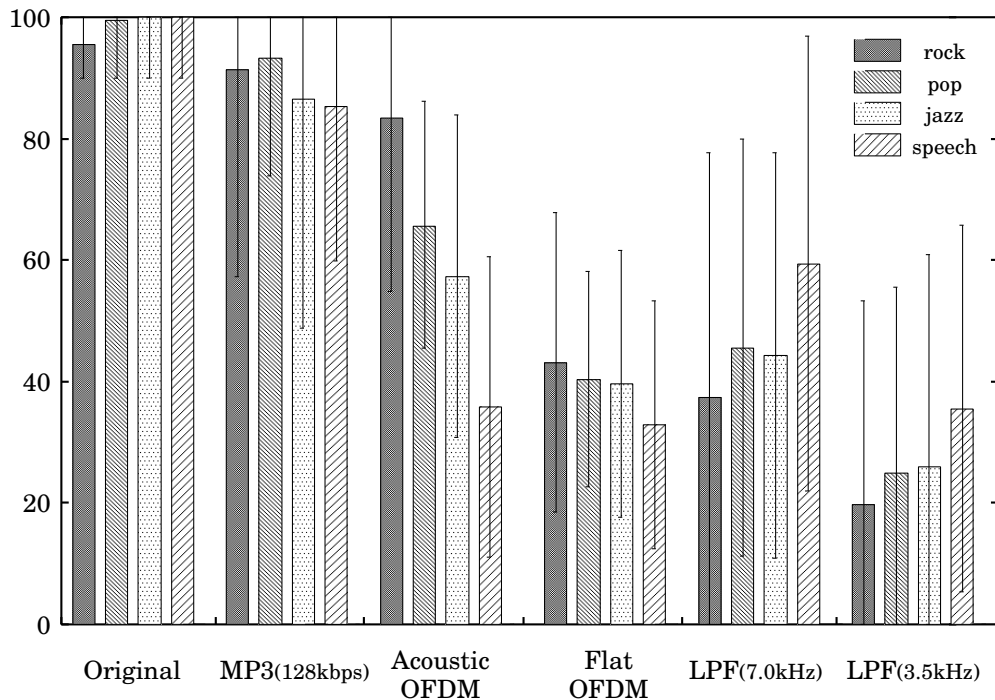


図 5.1: 主観評価試験

## 5.2 伝送性能試験

音響 OFDM と Flat OFDM の伝送性能を測定し、伝送距離、指向特性、符号化耐性について比較した．スピーカは，TimeDomain TD307MK を用い，正面 1m の地点で 65dB SPL となる音量で再生した．暗騒音は約 35dB SPL の環境で，リビングルー

ムに模擬した部屋で実験を行った。録音は携帯端末の録音機能を使って行った。音源には上記の主観評価試験で用いたロック調の音源をエンコードしたものを利用して行った。以下にそれぞれについて示す。

### 5.2.1 伝送距離

音響 OFDM と Flat OFDM の伝送可能距離を測定した。360 ビットのデータを1つのデータ単位として、データを正常に取得できた取得率を、スピーカからの各伝送距離で測定した。スピーカからの出力は、TV 視聴時の平均ボリュームとして、65dB(A) と設定した。音響 OFDM と Flat OFDM の OFDM 信号の平均レベルは同じになるように設定し、音響 OFDM のスペクトル包絡に合わせることで伝送性能が変わるかどうかを測定した。図 5.2 に伝送距離試験の結果を示す。スピーカから 2m 付近までは、どちらもエラー無くデータを取得できることを確認した。伝送距離が 2m~4m 付近では、どちらもエラーが発生してくるが Flat OFDM の方が若干、データ取得率が高かった。伝送距離が 4m~5m では、逆に音響 OFDM の方が若干、データ取得率が高かった。これにより、音響 OFDM のスペクトル包絡に合わせることによる伝送性能の影響はあまり無いと考えられる。

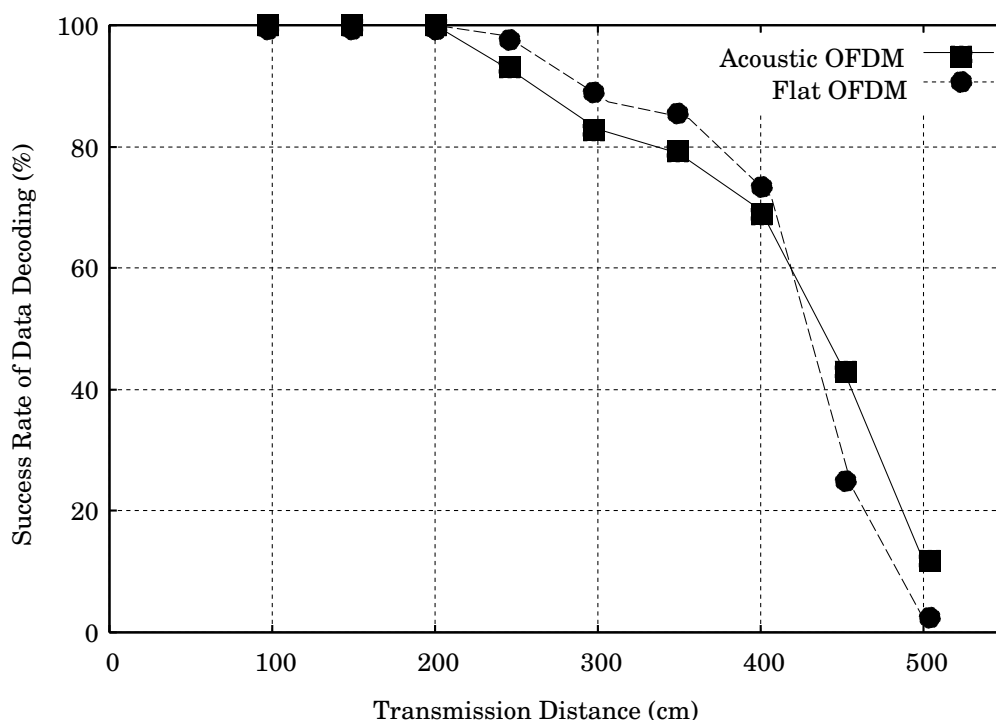


図 5.2: 伝送距離



## 5.2.2 指向性

次に指向性に対するデータ取得率を測定した。これは、音響 OFDM の OFDM 信号が高周波であるため、データを取得できる範囲が角度的に狭くなると考えられるためである。指向性の伝送試験は、スピーカからの伝送距離 2m の同心円上で、スピーカから正面の地点を 0 度とし、60 度付近まで移動してデータ取得率を測定した。図 5.3 に指向性試験の結果を示す。指向角度 20 度までは、エラーなく受信できた。20 度から 40 度付近では、音響 OFDM, Flat OFDM のどちらもエラーが発生してくるが、Flat OFDM の方が若干、取得率が高かった。指向角度が 40 度以上になると逆に音響 OFDM の方が取得率が高かった。これにより、指向性試験においても、音響 OFDM のスペクトル包絡による影響はほとんど無いと考えられる。

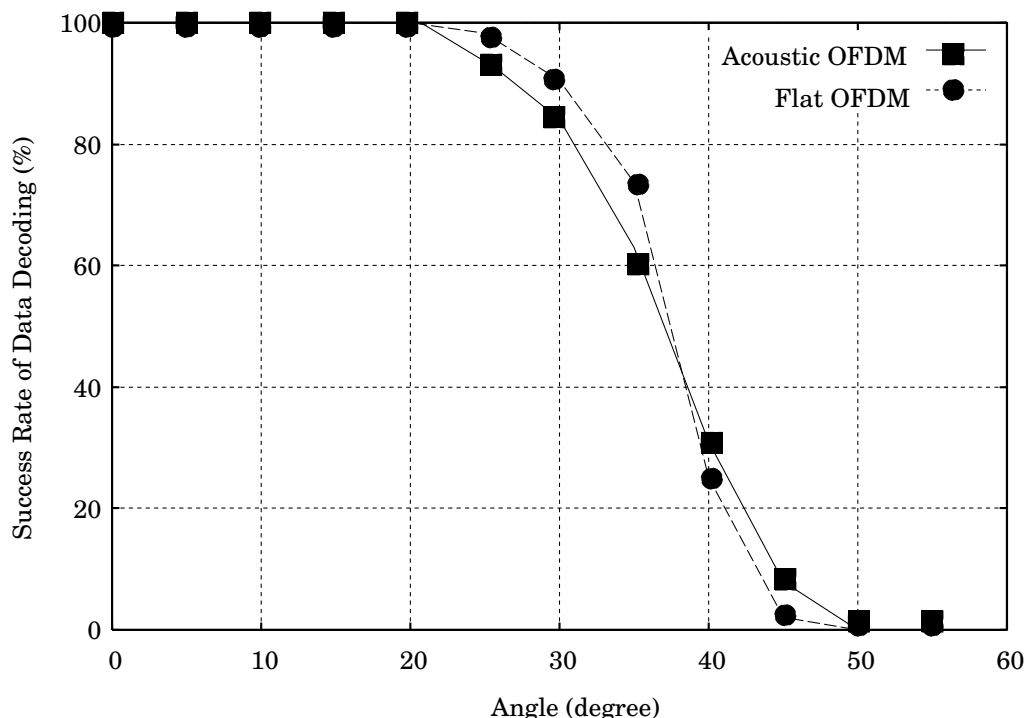


図 5.3: 指向特性

## 5.2.3 符号化耐性

音響信号に対して、データサイズを圧縮する符号化技術は、様々な場面で利用されている。例えば、TV 放送の場合では、音声品質を保ちながら情報伝送量を極力小さくするために、AAC(Advanced Audio Coding)[76][77] という符号化が用いられている。

AACの符号化は、聴覚にあまり知覚できない高音域の音や、大きな音でマスクングされる小さな音等を削除することで、圧縮率を高くしている。したがって、圧縮率を高くすると音響 OFDM の伝送信号が消えてしまう可能性がある。

この AAC 符号化は、TV 放送だけでなく DVD やインターネットのストリーミング等にも利用されているため、この AAC 符号化に対して、音響 OFDM の信号がどれぐらい耐性があるかを検証した。

検証には、音響 OFDM のエンコーダとデコーダの間に AAC の符号化と復号化の処理を行い、音響 OFDM のデータフレームの取得率を測定した。空中伝搬は行わずに、単純に AAC 符号化による影響のみを測定した。AAC 符号化の符号化レートを変えながら、それぞれの符号化レートにおける耐性を検証した。

図 5.4 は、AAC の符号化レートと音響 OFDM のデータフレーム受信の成功率を示す。この検証も、音響 OFDM の場合と、Flat OFDM の場合で相違があるかを確認するため、両方の場合で検証を行った。どちらも場合も、符号化レートが高くなれば、受信成功率が上がり、80kbps 以上になれば 100% の受信成功率となった。また、符号化レートが 32kbps 以下になると受信成功率は 0 になった。音響 OFDM と Flat OFDM の相違はほとんど無かった。

この検証は空中伝搬をしていないため、空中伝搬での歪み等も考慮して AAC の符号化レートは、最低でも 96kbps はあった方がよいことを確認した。

### 5.3 性能評価による考察

音響 OFDM の性能評価として、音質の影響を検証する主観評価試験と、伝送性能を検証する実験を行った。音響 OFDM の特徴としては、聴覚に不快にならないように伝送信号を加工し、情報伝送を可能にすることである。この点で、単純な OFDM 信号 (Flat OFDM) に比べて、音響 OFDM で生成した信号は、音源による差異もあるが、音質の劣化を抑制できていることがわかった。

また、伝送性能においては、単純な OFDM 信号 (Flat OFDM) と差分がほとんど無いことが伝送性能実験からもわかったため、伝送性能を落とすことなく音質の劣化を抑制できる技術であることを立証できた。

本実験に用いた音響 OFDM システムの伝送速度は 360bps であり、1 秒で 45 バイト程度のデータを伝送できる。これは、URL やメールアドレス等の短いテキスト情報であれば、1~2 秒で伝送でき、利用者が許容できる範囲であると考えられる。また、伝送距離においても TV 視聴を考慮すると、一般的な再生レベルで TV 正面の

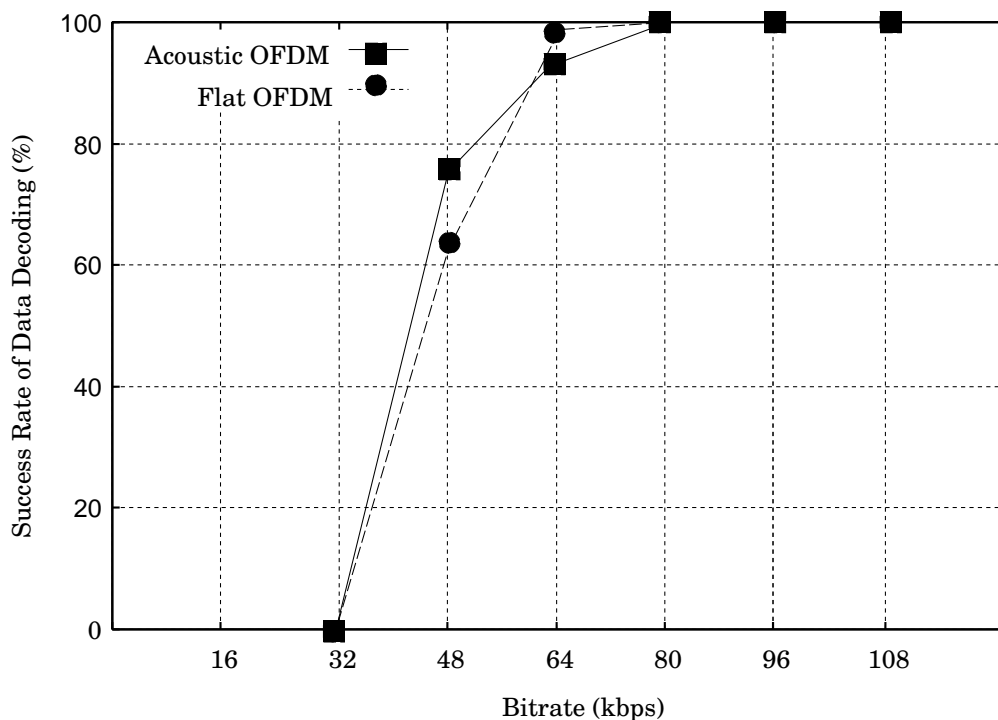


図 5.4: 符号化耐性

3~4 m, 指向角度が 20 度の範囲で, データが取得できることも確認できた. 家庭等で TV を視聴する場合においては十分な伝送性能であると言える.

## 5.4 ステレオ再生の評価

第 4 章で記載した音響 OFDM システムに第 3 章で提案したステレオ送信ダイバーシチの手法を追加実装して, 実際にステレオ再生を行い, 送信ダイバーシチの無い場合と比較を行った. スピーカは 30cm 離れた 2 点に同じスピーカを置いて検証した. スピーカから 1m 離れた地点で, 指向角度を正面からずらしながら計測した. 図 5.5 に, スピーカの正面からマイク位置の指向角度と, フレーム受信成功率の関係を記載する. 送信ダイバーシチを行った場合は, 指向角度- 20 度~20 度の範囲であれば, ほぼ安定して受信できることが確認できた. 送信ダイバーシチを行わない場合は, スピーカ正面でもフレーム受信成功率が低くなり, どの地点でも受信成功率は低くなった. このことから, ステレオ再生をする場合は, 送信ダイバーシチの手法が必要となることがわかった.

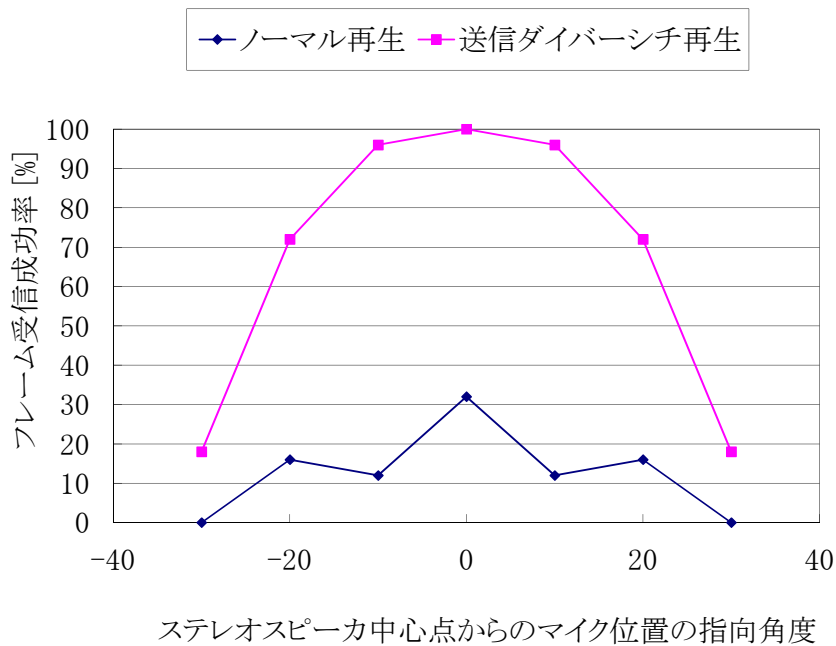


図 5.5: ステレオ再生の検証

## 5.5 ドップラーシフト補正の評価

第4章で記載した音響 OFDM システムに第3章で提案したドップラーシフト補正の手法を追加実装して、実際に送信側と受信側の周波数を少しずつずらしながら、ドップラーシフトの影響と補正効果を検証した。周波数のずれは ppm 単位で表記し、100ppm ずれたということは、0.01 周波数のずれ (ppm) とスピーカ正面 1m の地点での受信成功率の関係を示す。まず、ドップラーシフトの影響としては、400ppm 程度のずれまでは、ドップラーシフト補正無しでも問題なく受信が可能である。しかし、500ppm を越えるとほぼ受信が不可能になってくる。この音響 OFDM システムでは、1つのサブキャリアの周波数の幅が 43Hz であり、6000~8000Hz の周波数帯で伝送しているため、500ppm の周波数のずれは約 3~4 Hz の周波数のずれとなる。これはサブキャリア 43Hz に対してやく 10% 程度のずれが生じると受信ができなくなるということがわかった。

ドップラーシフト補正を行うと、周波数のずれが 500ppm を越えても、リサンプリング処理によってほぼ元の周波数に戻せるため、受信成功率の低下を抑えることができた。スマートフォン等、近年の携帯端末は、通信のための精度の良い振動子が搭載されており、サンプリング周波数のずれは 20ppm 以下になっている。したがっ

て、これらの携帯端末で利用する上ではドップラーシフト補正はなくても十分であるが、他の安価な組み込み機器での利用を検討する場合や、移動しながらの受信を検討する場合には、このドップラーシフト補正を利用した方が良いことがわかった。

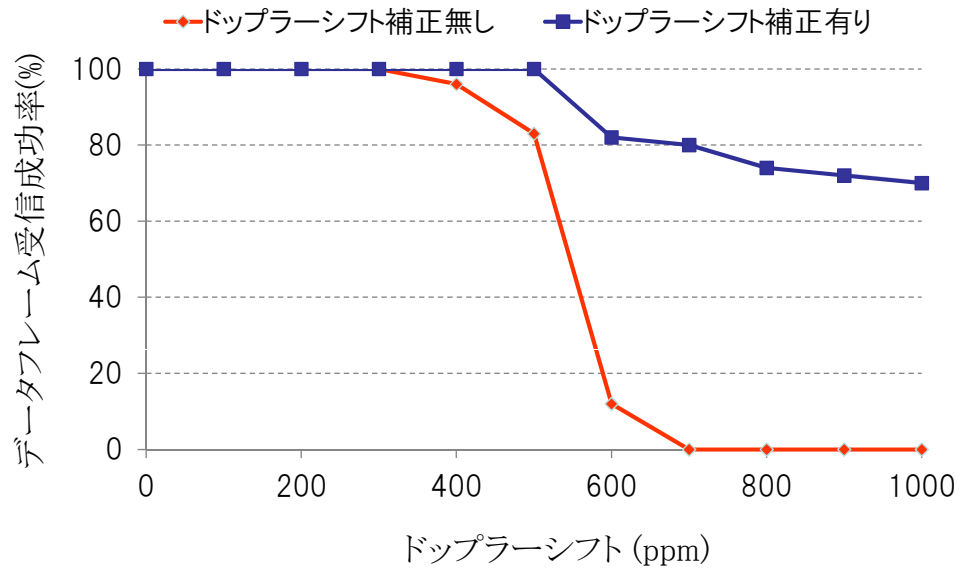


図 5.6: ドップラーシフトによる影響

## 第6章 シンボル非同期伝送方式

本章では、音響 OFDM による近距離情報伝送において、ID 等の短いテキスト情報をより頑健性高く伝送する場合等、伝送速度よりも通信距離を優先したい場合に有効になる伝送方式について述べる。基本的には前述した音響 OFDM 伝送方式と同じではなるが、シンボル同期の方法が異なるため、そこに焦点をあてて解説する。

### 6.1 シンボル同期信号の省略

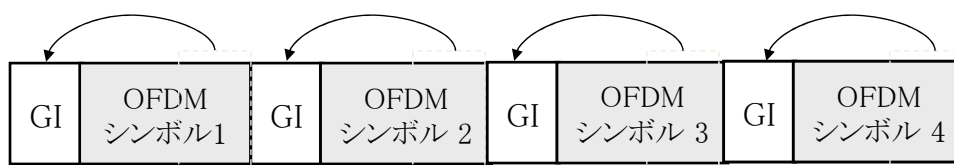
音響 OFDM の伝送でエラーが発生する要因のひとつとして、シンボル同期がとれないことがある。シンボル同期は低域音源に拡散して重畳しているため、低域音源との拡散符号の相関が強いと、逆拡散時にピークを誤検出してしまう場合がある。このため、OFDM 信号の S/N 比が高く受信できていたとしても、シンボル同期のエラーで受信が失敗することがある。そこで、伝送速度を下げても通信の頑健性を優先する場合には、シンボル同期が不要になる伝送方法を提案する。

シンボル同期とは OFDM のシンボルの先頭を識別するためのものであり、このシンボルの先頭を識別せずに適当な位置から復号をはじめるとシンボルの境界をまたがって復号をする可能性があり、サブキャリアの直交性が保たれなくなる。また受信信号も次のシンボルと混ざるため、正しい伝送データを受信できなくなる。したがって、シンボル同期を不要にするためには、ある程度、シンボル区間を伸長して、復号区間がシンボル境界をまたがらないようにする必要がある。

図 6.1 に、音響 OFDM のシンボル同期不要のシンボル構成を示す。通常の OFDM フレーム (A) とは異なり、シンボル非同期伝送方式の音響 OFDM 信号 (B) では、データシンボルが後方にも、「ガード時間+シンボル長」の分だけ伸張される。したがって、1 フレームの長さは 2 シンボルと 2 ガード時間を足した長さになる。これは通常の OFDM フレームのちょうど 2 倍の長さとなるため、データ伝送速度は半分になることになる。

受信側では、図 6.2 にしめすように、「ガード時間+シンボル長」離れた 2 点でデコードを試し、正確に受信できた方の結果をデコード結果とする。「ガード時間+シンボル長」離れた 2 点のどちらかは、シンボル境界をまたがらずに必ずデータシン

(A) 通常のOFDM変調信号



(B) シンボル非同期伝送方式のOFDM変調信号

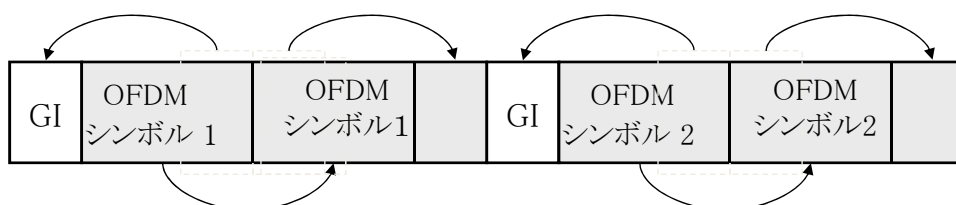


図 6.1: シンボル非同期方式のフレーム構成

ボル区間をデコードできるため、シンボル同期をとる必要がなくなり、シンボル同期エラーによる受信率の低下を避けることができる。正確に受信できたかどうかは、CRC チェックの判定で識別できるため、CRC チェックでデータの正当性が確認された方のデコード結果を出力する。場合によっては、どちらも正確に受信できる場合もある。その場合は、どちらのデコード結果も CRC チェックで正当性が確認できるが、その場合はどちらも同じデコード結果になるため、どちらを出力してもよい。

この伝送方式の場合、データ伝送速度は半分になるため、短い URL や ID のようなデータを確実に伝送したい場合に有効な方式と考えられる。また、シンボル同期信号を埋め込む必要がないため、原音信号の低域にあまりパワーが無いような音源でも利用しやすいという特徴もある。OFDM の変調信号の周波数を可聴域の中でも高い周波数 (15kHz 以上) にすれば、OFDM 信号自体の音もほとんど聞こえなくなってくるため、原音が無いような場合でも適用可能である。ただしこの場合は、音波の指向性が鋭くなったり、またマイクによっては拾音できないなどの問題もでてくる可能性がある。演算量としても、シンボル同期の畳み込み演算が不要になるため、簡易な実装が可能になる。

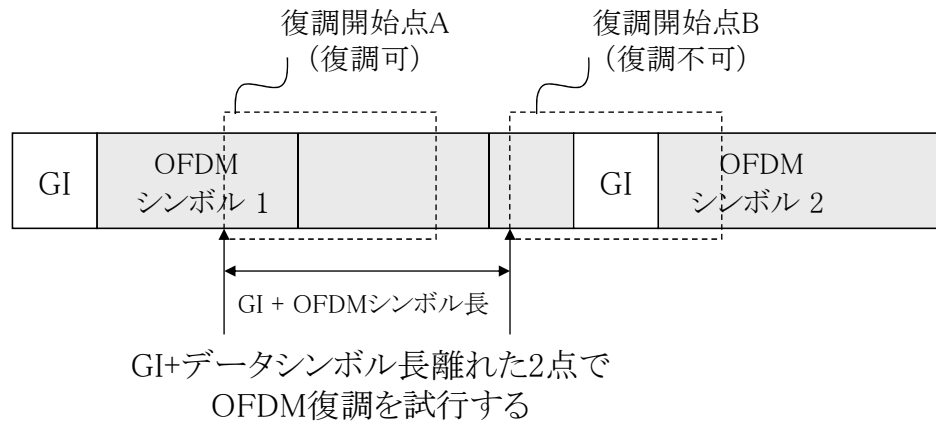


図 6.2: シンボル非同期方式の復調方法

## 6.2 伝送システムの設計と実装

シンボル非同期方式の音響 OFDM のシステム実装例として、実装パラメータを表 6.1 に示す。シンボル長は通常の音響 OFDM システムと同様に 1024 サンプルとし、FFT 可能なサンプル数とする。ガード時間は 956 サンプルとし、シンボル長とガード時間のサンプル数の 2 倍の 3960 サンプルが OFDM フレーム長となる。サブキャリア数は 40 とし 15 フレームでひとつのデータ単位とする。したがって、600 ビットの伝送ビット列がひとつのデータ単位となる。この 600 ビットの中で、CRC や誤り訂正符号を行うため、まず符号化率  $1/4$ 、拘束長 6 の畳み込み符号を行うため、情報ビット系列は  $600/4 - 5$  となり、145 ビットとなる。このうち 9 ビットが CRC ビットとなるため、実際に伝送できる情報量は、136 ビットとなる。つまり 17 バイトのデータを伝送できることになる。伝送速度は、この 17 バイトを  $3960 \text{ サンプル} \times 15$  フレームで伝送するため、約 1.5 秒で 17 バイトのデータを伝送できることになる。ID 等の短いテキストデータを伝送する際には実用的な範囲であると考えられる。

## 6.3 性能評価

上記の伝送システムで、データフレーム (17 バイト) の伝送性能がどれぐらいかを評価するために、OFDM 周波数帯において、ホワイトノイズを生成し、この OFDM 信号とホワイトノイズの S/N 比に対して受信成功率がどれぐらいになるかを測定した。図 6.3 は、OFDM 信号の周波数帯における SN 比とそれに対するデータフレー



表 6.1: 音響 OFDM システムパラメータ

パラメータ項目	値
サンプリング周波数	44.1 kHz
OFDM フレーム長	3960 サンプル
シンボル長	1024 サンプル
ガード時間	956 サンプル
サブキャリア数	40
OFDM 信号周波数帯	6400 - 8000 Hz
情報系列長	136
誤り訂正符号化率	1/4
誤り訂正符号 拘束長	6
誤り検出用 CRC 長	9
誤り検出用 CRC 多項式	$x^9 + x^4 + 1$

ム (17 バイト) の受信成功率を示している。SN 比が-3dB 以下になると受信はできなくなる。SN 比が-3dB 以上になれば受信できる可能性が上がってきて、SN 比が 0dB まであがればほぼ受信は成功するという結果になった。これは信号とノイズのパワーバランスがほぼ同等程度であれば受信が可能ということである。OFDM の周波数帯は 6400Hz~8000Hz を利用しているため、この周波数帯にノイズがあまりなければ受信が十分に可能であるといえる。人の話声や雑踏等は、低周波数のパワーが大きいため、あまり問題にはならないが、金属音や何かを叩いた音などは高周波にもノイズが大きいため、受信成功率が下がってしまうという特徴がある。

上記の伝送システムと、第 4 章で記載した音響 OFDM システム (シンボル同期方式) で、実際に空中伝搬試験を行い、伝送耐性の比較を行った。図 6.4 に空中伝搬における伝送距離と受信成功率の関係を記載する。試験環境は第 5 章で記載した環境と同様である。シンボル同期を行う方式では、3m 程度で受信成功率が落ち始める。一方、シンボル非同期方式では、伝送速度は半分以下になるが、5m 程度でも 80% 以上の受信成功率であり、伝送速度よりも伝送耐性を優先したい場合は、シンボル非同期方式が有用な場合もあると考えられる。

## 6.4 応用例

これまで、音響 OFDM では原音信号に聴覚に不快にならないように OFDM 信号を重畳するという技術を提案し、解説してきた。しかし、近距離情報伝送の利用用途としては、原音信号がなく単純に情報を伝送したいだけの場合もある。その場合

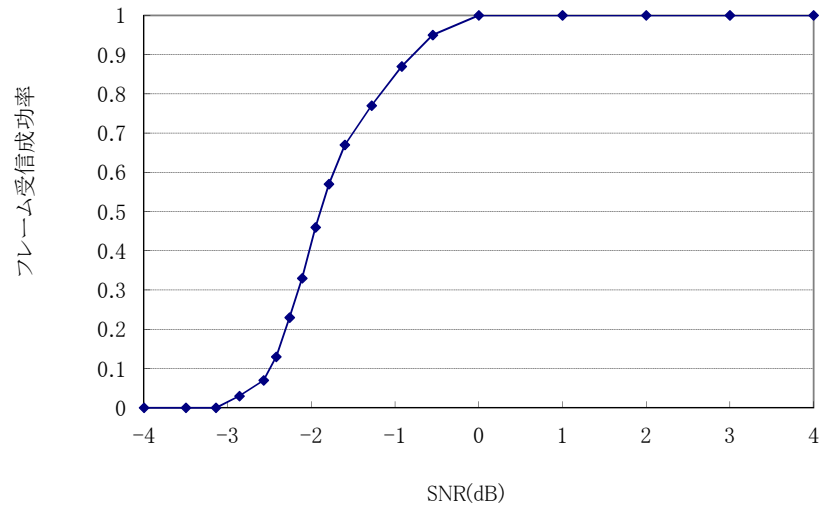


図 6.3: OFDM 信号の S/N 比による受信性能

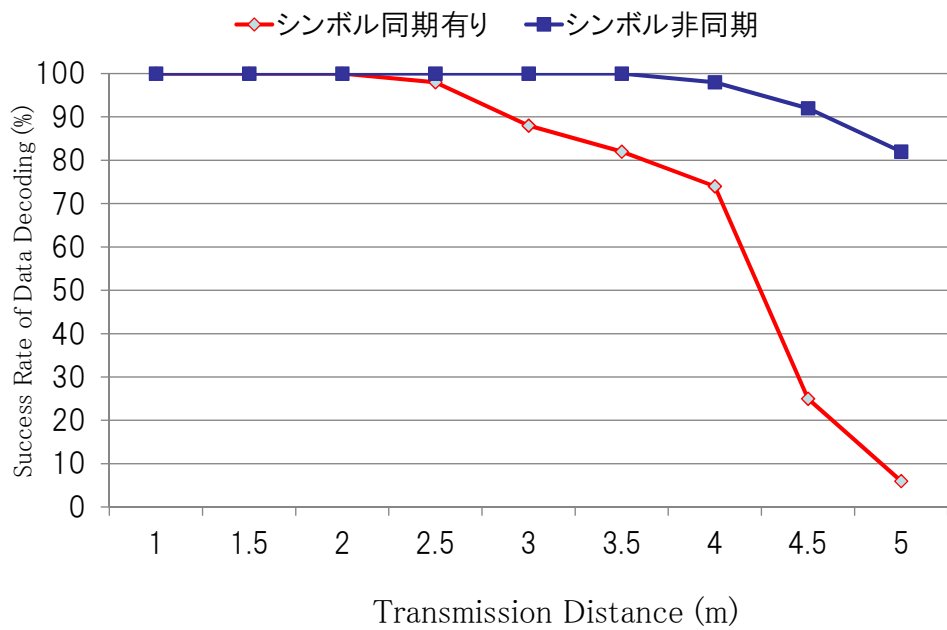


図 6.4: シンボル非同期方式の性能

は、上述したシンボル非同期方式の OFDM 伝送方法で、OFDM 信号だけを再生すれば良い。シンボル同期信号も不要なため、低域に音が無くても問題がない。ただ OFDM 信号がホワイトノイズのように聞こえてしまうため、この音が不快に感じる場合もある。そこで、この OFDM 信号の音を目立たなくさせる方法を提案する。

図 6.5 は、OFDM フレームの境界にマスキング音を入れる方法である。マスキング音とは、時間マスキング [78][79] という効果を用いて、後に続く音を聴覚的に目立たないようにする音である。時間マスキングとは、大きい音の前後の音は聴覚的に聞こえなくなるという効果で、この効果をもちいて、OFDM 信号のノイズ的な音を目立たなくさせる。特に OFDM フレームの境界は位相が不連続になるため、ここにマスキング音を入れることで位相不連続によるノイズを聞こえなくさせることができる。

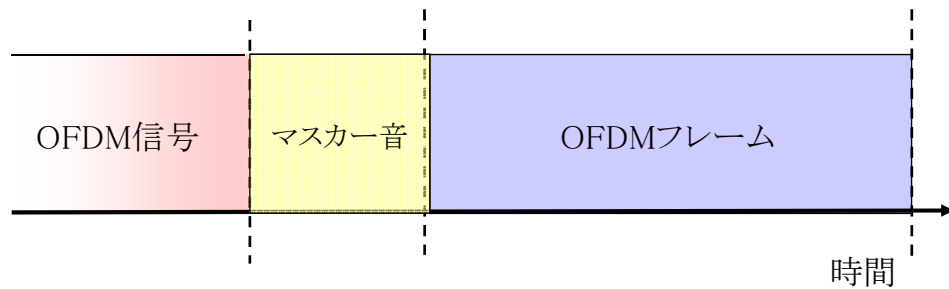


図 6.5: マスキング音の利用

マスキング音には、ある特定の周波数の正弦波等、トーン的な音を入れることが望ましい。このマスキング音を OFDM フレームの境界ごとに違う周波数の音にしてメロディを構成することもできる。周波数と音名の関係は表 6.2 にしめす。ここに示す音を利用して、OFDM フレーム境界ごとに異なる音を選択してメロディを構成すれば、単なるノイズ音ではなく音楽性のある音で伝送信号を送ることも可能になる。表 6.2 には、12 音階の音の周波数を記載しているが、周波数が 2 倍になれば 1 オクターブ高い音になり、周波数が半分になれば 1 オクターブ低い音になるため、これらの音から周波数を選択し、簡単な音楽を構成することが可能である。

このようにマスキング音を利用することによって、原音信号が無い場合でも音響 OFDM の伝送方式を利用して情報を音で伝送することが可能であり、不快な音をなるべく聞こえないようにもできる。

表 6.2: 周波数と音名の一覧

音名	周波数
A	440.00 Hz
A#	466.16 Hz
B	493.88 Hz
C	523.25 Hz
C#	554.37 Hz
D	587.33 Hz
D#	622.25 Hz
E	659.26 Hz
F	698.46 Hz
F#	739.99 Hz
G	783.99 Hz
G#	830.61 Hz

## 第7章 結論

本章では，提案方式である音響 OFDM の新規性，有用性，実用性について考察し，今後の展望についても述べる．

### 7.1 新規性

本論文では，音波情報伝送技術として，音響 OFDM を提案し，システムとして実装して性能評価まで行った．本技術は，音波情報伝送技術の中でも音響透かし技術に基づいた手法であり，既存の音源に聴覚には知覚できないように情報を重畳して伝送する方法である．この音響透かし技術の中で，100bps 以上の伝送速度を提供できる技術はこれまでになかった．したがって，伝送速度の飛躍的向上という意味で新規性があると言える．また，OFDM の変復調を可聴音の音波に適用した例もなかった．本技術では，OFDM の変調信号をスペクトル包絡を合わせて変形し，ノイズ音を音源に調和できる音に変換する点に新規性がある．この可聴音という特性を考慮して OFDM 変復調を応用した技術が音響 OFDM であり，新規性の高い技術であると考えられる．

### 7.2 有用性

音響 OFDM の伝送速度は数百 bps 程度であり，本論文で記載したシステムでも約 300bps の伝送速度がある．これは，URL やメールアドレスの短いテキストデータであれば，1～2 秒程度で伝送できることである．ユーザはスマートフォン等で音響 OFDM のデコーダを起動させれば，1～2 秒でデータを取得できる．これはユーザが許容できるレスポンスタイムの範囲内であると考えられる．また，伝送距離においても通常の TV 視聴レベルで 3～4m の伝送が可能である．したがって，一般的な家庭の TV 視聴範囲において受信が可能であると考えられる．これにより，TV やラジオの音声に番組の Web サイトの URL を重畳しておいて，それをスマートフォンで取得するというアプリケーションが可能になる．このような観点から，既存の

音声に短いテキスト情報を入れて伝送するアプリケーションに音響 OFDM は有用であると考えられる。

### 7.3 実用性

音響 OFDM のシステムの実装において、スマートフォンや小型の機器へ実装する際の演算量の削減について述べた。実際にスマートフォンや小型デバイスにも実装を行い、音響 OFDM が実現可能であることも確認できた。音波情報伝送技術の応用としては、現在、ショッぷらっと [80] や Air Stamp[81] 等のサービスがある。これらは、音波が壁等の遮蔽物で遮断できることを利用し、限られた空間内だけ通信可能にするために音波を用いている。このようなサービスにも音響 OFDM 技術は適用可能であり、様々なサービス、アプリケーションに応用可能である点で実用性が高いと考えられる。

### 7.4 今後の展望

近年、M2M(Machine to Machine) や、O2O(Offline to Online) 等の新事業領域が開拓されつつあり、近距離情報通信の多様化が進んでいる。音響 OFDM も、近距離情報通信技術のひとつの手段として、利用できると考えている。音波情報伝送技術の利点としては、特殊なハードウェアを搭載することなくスピーカとマイクで通信できる点で、携帯端末が普及する中で、携帯端末のスピーカやマイクを利用して通信ができることは、今後の多様なアプリケーションの創出に有用であると考えている。 今後は、技術の性能向上だけでなく、サービス、アプリケーションの開発に向けてより実用的な課題にも取り組んでいき、近距離情報通信の利便性、有用性の向上に貢献していきたいと考えている。

# 謝辞

本研究を遂行し学位論文をまとめるにあたって、多くのご支援とご指導を賜りました牧野昭二教授に深く感謝しております。研究内容だけでなく、進学相談等から発表練習等ご指導いただいたこと多大なる感謝をしております。そして、博士論文の研究内容だけでなく、マイクロフォンアレイや音声認識等についても様々なアドバイスを頂いたことにも感謝しております。

また、博士課程への進学および研究全般にわたる多大なご支援、ご指導を賜りました山田武志准教授に深く感謝しております。

本論文をまとめるにあたり、多くの助言を頂きました。本学システム情報系の福井和弘教授、亀山啓輔教授、国立情報学研究所の小野順貴准教授に心よりお礼申し上げます。

研究遂行にあたり、様々な助言やご協力をいただいた、NTTドコモ先進技術研究所およびサービスイノベーション部の皆様にも深く感謝いたします。

最後に、これまで自分の思う道を進むことに対し、温かく見守り支援して下さった家族に対しては深い感謝の意を表して謝辞と致します。

## 参考文献

- [1] H.Matsuoka, Y.Nakashima, T.Yoshimura: Acoustic OFDM system and performance analysis, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol.91, No.7, pp.1652-1658, 2008.
- [2] H.Matsuoka, Y.Nakashima, T.Yoshimura: Acoustic OFDM system and its extension, *The Visual Computer*, Vol.25, No.1, pp.3-12, 2009.
- [3] H.Matsuoka, Y.Nakashima, T.Yoshimura: Acoustic OFDM: Embedding high bit-rate data in audio, *Advances in Multimedia Modeling. Springer Berlin Heidelberg*, pp.498-507, 2008.
- [4] H.Matsuoka, Y.Nakashima, T.Yoshimura: Acoustic communication with OFDM signal embedded in Audio, *In Audio Engineering Society Conference 29th*, 2006.
- [5] 松岡 保静: 音響データ通信技術: 音響 OFDM, 日本音響学会誌 (小特集) 携帯情報機器における音響技術, Vol.68, No.3, pp.143-147, 2012.
- [6] Zigbee Alliance: <http://www.zigbee.org/>
- [7] Bluetooth SIG (Special Interest Group): <http://www.bluetooth.org/>
- [8] NFC Forum: <http://nfc-forum.org/>
- [9] G.Wu, S.Talwar, K.Johnsson, N.Himayat, K.D.Johnson: M2M: From mobile to embedded internet, *IEEE Communications Magazine*, Vol.49, No.4, pp.36-43, 2011.
- [10] X.Zuo, A.Chin, A.Fan, B.Xu, D.Hong, Y.Wang, X.Wang: Connecting people at a conference: a study of influence between offline and online using a mobile social application, *In Green Computing and Communications (GreenCom 2012)*, IEEE International Conference, 2012.



- [11] C.V.Lopes, P.M.Q.Aguiari: Aerial Acoustic Communications., *In IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001.
- [12] V.Gerasimov, W.Bender: Things that talk: using sound for device-to-device and device-to-human communication, *In IBM Systems Journal*, 2000.
- [13] R.Nandakumar, K.K.Chintalapudi, V.R.Dhwani: Secure peer-to-peer acoustic nfc, *In Proceedings of ACM SIGCOMM 2013*, 2013.
- [14] M.Hanspach, M.Goetz: On Convert Acoustical Mesh Networks in Air, *Journal of Communications*, Vol.8, No.11, pp.758-767, 2013.
- [15] I.F.Akyildiz, D.Pmppili, T.Melodia: Underwater acoustic sensor networks, *Ad Hoc Networks*, Vol.3, No.3, pp.257-279, 2005.
- [16] E.M.Sozer, M.Stojanovic: Reconfigurable acoustic modem for underwater sensor networks, *In Proceedings of the 1st ACM International Workshop on Underwater Networks*, pp.101-104, 2006.
- [17] J.Y.Wills, W.J.Heidemann: Low-power acoustic modem for dense underwater sensor networks, *In Proceedings of the 1st ACM International Workshop on Underwater Networks*, pp.79-85, 2006.
- [18] F.Hartung, M.Kutter: Multimedia watermarking techniques, *Proceedings of the IEEE*, Vol.87, No.7, pp.1079-1107, 1999.
- [19] M.Swanson, B.Zhu, A.Tewfik: Current state-of-the-art, challenges and future directions for audio watermarking, *In Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pp.19-24, 1999.
- [20] W.Bender, D.Gruhl, N.Morimoto, A.Lu: Techniques for data hiding, *IBM Systems Journal*, Vol.35, No.3-4, pp.313-335, 1996.
- [21] S.W.Foo: Three techniques of digital audio watermarking, *Digital Audio Watermarking Techniques and Technologies*, Hershey PA, IGI Global, pp.104-122, 2008.
- [22] D.Gruhl, A.Lu, W.Bender: Echo Hiding, *In Information Hiding 1996*, pp.295-315, 1996.

- [23] B.S.Ko, R.Nishimura, Y.Suzuki: Time-spread echo method for digital audio watermarking, *IEEE Transaction on Multimedia*, Vol.7(2), pp.212-221, 2005.
- [24] Y.Yardimeci, A.E.Cetin, R.Ansari: Data hiding in speech using phase coding, *In Eurospeech 97*, pp.1679-1682, September 1997.
- [25] Barkouti, Wahid, Sihem Nasri, Adnane Cherif: Audio Watermarking Based On The PSK Modulation, *International Journal on Computer Science and Engineering (IJCSE)*, Vol.3.9, 2011.
- [26] L.Boney, A.H.Tewfik, K.N.Hamdy: Digital Watermarks for Audio Signals, *In IEEE Intl. Conf. on Multimedia Computing and Systems*, pp.473-480, March 1996.
- [27] D.Kirovski, S.M.Henrique: Spread-spectrum watermarking of audio signals, *Signal Processing, IEEE Transactions on* 51.4, pp.1020-1033, 2003.
- [28] A.Nishimura: Audio watermarking based on sinusoidal amplitude modulation, *In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2006*, Vol.4, pp.797-800, 2006.
- [29] A.Nishimura: Data hiding for speech sounds using subband amplitude modulation robust against reverberations and background noise, *in Proceedings of IHH-MSP2006*, pp.7-10, 2006.
- [30] H.Matsuoka: Spread spectrum audio steganography using sub-band phase shifting, *in Proceedings of IHH-MSP2006*, pp.3-6, 2006.
- [31] T.Modegi: Development of audio watermark technology to be extracted contactlessly by cell phone, *IEEJ Transactions on Electronics. Information Systems*, Vol.126, No.7, pp.825-831, 2006.
- [32] T.Modegi: Construction of ubiquitous acoustic spaces using audio watermark technology and mobile terminals, *IEEJ Transactions on Electrical and Electronic Engineering*, Vol.2, No.6, pp.608-619, 2007.
- [33] T.Ono: Watermark embedded in sound, *In Digital Watermark and Content Protection* (In Japanese), Ohm-sha, pp.122-138, 2001.

- [34] P.Bassia, I.P.Pitas: Robust audio watermarking in the time domain, *In Proceedings of European Signal Processing Conference*, pp.25-28, 1998.
- [35] A.Kunisa: Host-cooperative metadata embedding framework, *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2007.
- [36] N.Aoki: A band extension technique for G.711 speech using steganography, *IEICE Transaction on Communications*, Vol.89, pp.1896-1898, 2006.
- [37] N.Aoki: A band extension technique for narrow-band telephony speech based on full wave rectification, *IEICE Transaction on Communications*, Vol.93, No.3, pp.729-731, 2010.
- [38] A.Ito, S.Makino: Data hiding is a better way for transmitting side information for MP3 bitstream, *In Proceedings of the 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp.495-498, 2009.
- [39] T.Munekata, T.Yamaguchi, H.Handa, R.Nishimura, Y.Suzuki: Portable acoustic caption decoder using IH techniques for enhancing lives of the people who are deaf or hard-of-hearing, *International Journal of Innovative Computing, Information and Control*, Vol.5(7), pp.1829-1836, 2009.
- [40] S.B.Weinstein, P.M.Ebert: Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform, *In IEEE Transactions of Communications*, volume COM-19, pages 628-634, October 1971.
- [41] M.Speth, et al: Optimum Receiver Design for Wireless Broadband Systems Using OFDM-Part I, *In IEEE Transactions on Communications*, Vol.47, pp.1668-1677, November 1988.
- [42] S.T.Christensen, O.Niels: Shape optimization of a loudspeaker diaphragm with respect to sound directivity properties, *Control and Cybernetics 27*, Page 177-198, 1998.
- [43] K.R.Holland: Loudspeaker and Headphone Handbook, *Ed. J.Borwick*, Page 11-13, Focal Press, 2001.

- [44] G.N.Watson: A treatise on the theory of Bessel functions, *Cambridge university press*, 1995.
- [45] S.Boll: Suppression of acoustic noise in speech using spectral subtraction, *Acoustics, Speech and Signal Processing, IEEE Transactions on 27.2*, pp. 113-120, 1979.
- [46] T.M.Schmidl, C.C.Donald: Robust frequency and timing synchronization for OFDM., *Communications, IEEE Transactions on 45.12*, pp.1613-1621, 1997.
- [47] A.J.Coulson: Maximum likelihood synchronization for OFDM using a pilot symbol: algorithms, *Selected Areas in Communications, IEEE Journal on 19.12*, pp.2486-2494, 2001.
- [48] M.Hsieh, and C.Wei: A low-complexity frame synchronization and frequency offset compensation scheme for OFDM systems over fading channels, *Vehicular Technology, IEEE Transactions on 48.5*, pp. 1596-1609, 1999.
- [49] C.Martin, A.Lempel: On fast M-sequence transforms, *Information Theory, IEEE Transactions on 23.1*, pp. 135-137, 1977.
- [50] R.L.Peterson, E.Z.Rodger, E.B.David: Introduction to spread-spectrum communications, *New Jersey: Prentice Hall Vol.995*, 1995.
- [51] R.L.Pickholtz, L.S.Donald, B.M.Laurence: Theory of spread-spectrum communications—a tutorial, *Communications, IEEE Transactions on 30.5*, pp. 855-884, 1982.
- [52] J.Johnston: Transform Coding of Audio Signals Using Perceptual Noise Criteria, *IEEE Journal on Selected Areas in Communication*, Vol.6, pp.314-323, 1988.
- [53] S.Van De Par, A.Kohlrausch, G.Charestan, R.Heusdens: A new psychoacoustical masking model for audio coding applications, *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference*, Vol. 2, 2002.
- [54] M.D.Swanson, B.Zhu, A.H.Tewfik, L.Boney: Robust audio watermarking using perceptual masking, *Signal processing, 66(3)*, pp.337-335, 1998.

- [55] J.O.Smith, P.Gossett: A flexible sampling-rate conversion method, *In Proceedings of the International Conference on Acoustic, Speech, and Signal Processing*, Vol.2, pp.19.4.1-19.4.2, March 1984.
- [56] G.Santella: A frequency and symbol synchronization system for OFDM signals: architecture and simulation results, *Vehicular Technology, IEEE Transactions on 49.1*, pp.254-275, 2000.
- [57] A.J.Coulson: Maximum likelihood synchronization for OFDM using a pilot symbol: algorithms, *Selected Areas in Communications, IEEE Journal on 19.12*, pp.2486-2494, 2001.
- [58] A.J.Paulraj, D.A.Gore, R.U.Nabar, H.Bolcskei: An overview of MIMO communications-a key to gigabit wireless, *Proceedings of the IEEE, 92(2)*, pp. 198-218, 2004.
- [59] G.L.Stuber, J.R.Barry, S.W.Mclaughlin, Y.I.Li, T.G.Pratt: Broadband MIMO-OFDM wireless communications, *Proceedings of the IEEE, 92(2)*, pp. 271-294, 2004.
- [60] K.F.Lee, B.W.Douglas: A space-frequency transmitter diversity technique for OFDM systems, *Global Telecommunications Conference IEEE*, Vol.3, 2000.
- [61] D.Agrawal, V.Tarokh, A.Naguib, N.Seshadri: Space-time coded OFDM for high data-rate wireless communication over wideband channels, *In Vehicular Technology Conference VTC98 IEEE*, Vol.3, pp. 2232-2236, 1998.
- [62] S.B.Wicker, K.B.Vijay: Reed-Solomon codes and their applications, *John Wiley Sons*, 1999.
- [63] J.S.Plank: A tutorial on Reed-Solomon coding for fault-tolerance in RAID-like systems, *Software Practice and Experience, 27.9*, pp. 995-1012, 1997.
- [64] J.S.Plank, X.Lihao: Optimizing Cauchy Reed-Solomon codes for fault-tolerant network storage applications, *Network Computing and Applications, NCA 2006*, Vol.3, pp. 173-180, 2006.

- [65] P.Koopman, C.Trilib: Cyclic redundancy code (CRC) polynomial selection for embedded networks, *2004 International Conference on Dependable Systems and Networks, IEEE*, 2004.
- [66] D.V.Sarwate: Computation of cyclic redundancy checks via table look-up, *Communications of the ACM 31.8*, pp. 1008-1013, 1988.
- [67] R.Johannesson, S.Z.Kamil: Fundamentals of convolutional coding, *Wiley-IEEE press*, 1999.
- [68] E.Malkamaki, L.Harry: Evaluating the performance of convolutional codes over block fading channels, *IEEE Transactions on Information Theory*, 45.5, pp. 1643-1646, 1999.
- [69] J.Hagenauer, P.Hoeher: A Viterbi algorithm with soft-decision outputs and its applications, *In Global Telecommunications Conference and Exhibition'Communications Technology for the 1990s and Beyond'(GLOBECOM)*, pp. 1680-1686, 1989.
- [70] C.Berrou, P.Adde, E.Angui, S.Faudeil: A low complexity soft-output Viterbi decoder architecture, *IEEE International Conference ICC'93*, Vol.2, pp. 737-740, 1993.
- [71] A.M.Michelson, H.L.Allen: Error-control techniques for digital communication, *New York, Wiley-Interscience*, 1985.
- [72] G.Forney: On decoding BCH codes, *IEEE Transactions on Information Theory 11.4*, pp. 549-557, 1965.
- [73] S.Lei, K.L.Vincent: Performance analysis of adaptive interleaving for OFDM systems, *Vehicular Technology, IEEE Transactions on 51.3*, pp. 435-444, 2002.
- [74] X.F.Wang, R.S.Yousef, Z.Mao: On the code and interleaver design of broadband OFDM Systems, *Communications Letters, IEEE 8.11*, pp. 653-655, 2004.
- [75] ITU-R Recommendation BS.1534: Method for the subjective assessment of intermediate quality level of coding systems, In 2001.

- [76] D.Domazet, K.Mario: Advanced software implementation of MPEG-4 AAC audio encoder, *Video/Image Processing and Multimedia Communications, 2003. 4th EURASIP Conference*, Vol. 2, 2003.
- [77] E.Alexandre, P.Antonio, S.Manuel: Low-complexity bit-allocation algorithm for MPEG AAC audio coders, *Signal Processing Letters, IEEE 12.12*, pp. 824-826, 2005.
- [78] W.Lie, C.Li-Chun: Robust and high-quality time-domain audio watermarking subject to psychoacoustic masking, *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium*, Vol.2, 2001.
- [79] P.Bassia, P.Ioannis, N.Nikos: Robust audio watermarking in the time domain, *Multimedia, IEEE Transactions on 3.2*, pp.232-241, 2001.
- [80] ショップらっと: <http://shoplat.net/>
- [81] Air Stamp: [https://www.nttdocomo.co.jp/info/news\\_release/2014/03/04\\_00.html](https://www.nttdocomo.co.jp/info/news_release/2014/03/04_00.html)

# 付録 A 付録

本付録では、音響 OFDM システムの実装に必要な各要素技術のソースコードを添付し、解説する。

## A.1 Reed-Solomon 符号化・復号化

データ分割時における冗長フレーム作成のために、Reed-Solomon 符号を用いる。Reed-Solomon 符号化には下記に示す Vandermonde 行列を用いる。この行列の  $n$  行目は、 $(1^n, 2^n, 3^n, 4^n, \dots)$  となっている。各分割フレームを先頭から 1 バイトずつ取り出し、冗長フレームも 1 バイトずつ生成していく。下記に 4 つの分割フレームから 4 つの冗長フレームを生成する例を示す。4 つの分割フレームから 1 バイトずつのデータを取り出し、それぞれを  $( )$  とする。このベクトルを Vandermonde 行列に掛けて算出されたベクトルは  $( )$  となり、それぞれの項が各冗長フレームの 1 バイトのデータとなる。先頭のバイトデータから最終のバイトデータまで 1 バイトずつ同様の処理で符号化をする。

$$\begin{pmatrix} 1^1 & 2^1 & 3^1 & 4^1 \\ 1^2 & 2^2 & 3^2 & 4^2 \\ 1^3 & 2^3 & 3^3 & 4^3 \\ 1^4 & 2^4 & 3^4 & 4^4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix}$$

このように符号化した  $x_1, x_2, x_3, x_4, f_1, f_2, f_3, f_4$  の中から、いずれの 4 シンボルからでも  $x_1, x_2, x_3, x_4$  を復元できる。例えば、 $f_1, f_2, x_2, f_4$  から復元する場合は下記に示す式で復元できる。

$$\begin{pmatrix} 1^1 & 2^1 & 3^1 & 4^1 \\ 1^2 & 2^2 & 3^2 & 4^2 \\ 0 & 1 & 0 & 0 \\ 1^4 & 2^4 & 3^4 & 4^4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ x_2 \\ f_4 \end{pmatrix}$$

データフレームのヘッダには各々シーケンス番号が付加される。分割フレームの場合は、先頭から何番目のフレームかを示し、冗長フレームの場合は Vandermonde 行列の何行目で演算されて生成されたデータかを示す。上記の例の場合、 $f_1$  のシー



ケンス番号は1に、 $f_2$ のシーケンス番号は2になる。受信側における復号化処理では、上記の例に示すとおり、受信したフレームを並べて、それに相当する生成行列を作り、その生成行列の逆行列を計算することで $(x_1, x_2, x_3, x_4)$ を復元できる。計算は全てGF(256)のガロア体で行う。

```
Code1
#define RS_GFSIZE (256)
#define RS_GFPRIM (285)

static unsigned char rs_gflog[RS_GFSIZE];
static unsigned char rs_gfinverse[RS_GFSIZE];

void rs_gfinit(){
    unsigned int n, log;

    n = 1;
    for(log = 0; log < (RS_GFSIZE - 1); log++){
        rs_gflog[n] = log;
        rs_gfinverse[log] = n;
        n = n << 1;
        if(n >= RS_GFSIZE){
            n = n ^ RS_GFPRIM;
        }
    }
}
```

Code1は、リードソロモン符号の演算に用いるガロア体の初期化を行う。ガロア体はGF(256)であり、乗算、除算のためのテーブルを作成する。GF(256)はGF(2)の拡大体であり、生成多項式 $x^8 + x^4 + x^3 + x^2 + 1$ を用いて生成される。後述するBCH符号に用いるガロア体とは異なる。

Code2

```
unsigned char rs_gfmult(a, b)
    unsigned char a, b;
{
    unsigned int sum;

    if(a == 0 || b == 0){
        return 0;
    }
    sum = rs_gflog[a] + rs_gflog[b];
    if(sum >= RS_GFSIZE-1){
        sum -= (RS_GFSIZE - 1);
    }
    return rs_gfinverse[sum];
}
```

Code2はリードソロモン符号の演算に用いるガロア体の乗算 rs\_gfmult である。a\*b の演算結果を返す。

Code3

```
unsigned char rs_gfdiv(a, b)
    unsigned char a, b;
{
    int sum;

    if(a == 0) return 0;
    if(b == 0) return -1;
    sum = rs_gflog[a] - rs_gflog[b];
    if(sum < 0) sum += (RS_GFSIZE - 1);
    return rs_gfinverse[sum];
}
```

Code3はリードソロモン符号の演算に用いるガロア体の除算 rs\_gfdiv である。a/b の演算結果を返す。

Code4

```
unsigned char rs_gfpower(a, b)
    unsigned char a, b;
{
    int i;
    unsigned char sum;

    sum = 1;
    for(i=0; i < b; i++){
        sum = rs_gfmult(sum, a);
    }
    return sum;
}
```

Code4はリードソロモン符号の演算に用いるガロア体のべき乗rs\_gfpowerである。 $a^b$ の演算結果を返す。

Code5

```
void rs_encode(data, fec, ndata, nfec)
    unsigned char *data, *fec;
    unsigned int ndata, nfec;
{
    int i, j, k;

    bzero(fec, (nfec * sizeof(unsigned char)));
    for(i = 0; i < ndata; i++){
        for(j = 0; j < nfec; j++){
            fec[j] ^= rs_gfmult(data[i], rs_gfpower((i+1), (j+1)));
        }
    }
}
```

Code5はリードソロモン符号の符号化処理rs\_encodeである。データの配列data, 生成する冗長データの配列fec, データ数ndata, 冗長データ数nfecを引数とする。

Code6

```
void rs_decode(d_data, d_seq, f_data, f_seq,
              r_data, r_seq, ndata, nfec)
    unsigned char *d_data, *f_data, *r_data;
    unsigned char *d_seq, *f_seq, *r_seq;
    int ndata, nfec;
{
    int i, j, k;
    unsigned char **matrix;
    unsigned char multiple;
    int total = ndata + nfec;

    matrix =
        (unsigned char **)malloc(nfec * sizeof(unsigned char *));
    for(i = 0; i < nfec; i++){
        matrix[i] =
            (unsigned char *)malloc(nfec * sizeof(unsigned char));
    }

    for(i = 0; i < ndata; i++){
        for(j = 0; j < nfec; j++){
            f_data[j] ^=
rs_gfmult(d_data[i], rs_gfpower(d_seq[i], f_seq[j]));
        }
    }

    for(i = 0; i < nfec; i++){
        for(j = 0; j < nfec; j++){
            matrix[j][i] = rs_gfpower(r_seq[i], f_seq[j]);
        }
    }

    for(i = 0; i < nfec; i++){
        multiple = matrix[i][i];
        f_data[i] = rs_gfdiv(f_data[i], multiple);
        for(j = 0; j < nfec; j++){
            matrix[i][j] = rs_gfdiv(matrix[i][j], multiple);
        }
    }
}
```

```

    for(j = 0; j < nfec; j++){
        if(j != i){
            multiple = matrix[j][i];
            f_data[j] ^= rs_gfmult(f_data[i], multiple);
            for(k = 0; k < nfec; k++){
                matrix[j][k] ^= rs_gfmult(matrix[i][k], multiple);
            }
        }
    }
}

for(i = 0; i < nfec; i++){
    r_data[i] = f_data[i];
}
}

```

Code6 は、リードソロモン符号の復号化処理 rs\_decode である。復号に用いるデータの配列 d.data, 対応するデータのシーケンス番号の配列 d.seq, 復号に用いる冗長データの配列 f.data, 対応する冗長データのシーケンス番号の配列 f.seq, 消失したデータの復元用データ配列 r.data, 対応する消失データのシーケンス番号の配列 r\_data, 復号に用いるデータの数 ndata, 復号に用いる冗長データの数 nfec を引数とする。

## A.2 CRC エンコーダ・デコーダ

音響 OFDM では通信路符号化に、CRC と畳込み符号を用いる。ここでは、CRC のエンコーダ・デコーダおよび畳込み符号、軟判定ビタビ復号のプログラム例を記載する。

Code7

```
#define CRCbit          (7) // Number of CRC Bit
#define INFO_LENGTH    (360) // Number of Data Bit
int crcg[CRCbit + 1] = {1,0,0,0,0,0,1,1}; // CRC Generator

void crc_encode(char *info, char *data)
{
    int i, j;
    char crc[INFO_LENGTH + CRCbit];
    bzero(crc, INFO_LENGTH + CRCbit);

    for(i = 0; i < INFO_LENGTH; i++){
        data[i] = info[i];
        crc[i] = info[i];
    }

    for(i = 0; i < INFO_LENGTH; i++)
        if(crc[i])
            for(j = 0; j < CRCbit + 1; j++)
                crc[i+j] ^= crcg[j];

    for(i = INFO_LENGTH; i < INFO_LENGTH + CRCbit; i++)
        data[i] = crc[i];
}
```

Code7は、CRC エンコーダの関数 `crc_encode` であり、入力の引数 `info` のビット列から CRC チェックサムを計算し、CRC のチェックサムを `info` のビット列の後方に付け足したものを引数 `data` に格納する。

Code8

```
#define CRCbit          (7) // Number of CRC Bit
#define INFO_LENGTH    (360) // Number of Data Bit
int crcg[CRCbit + 1] = {1,0,0,0,0,0,1,1}; // CRC Generator

int crc_decode(char *data){
    int i, j;
    char info[INFO_LENGTH], crc[INFO_LENGTH + CRCbit];

    for(i = 0; i < INFO_LENGTH; i++){
        info[i] = data[i];  crc[i] = data[i];
    }

    for(i = 0; i < INFO_LENGTH; i++)
        if(crc[i])
            for(j = 0; j < CRCbit + 1; j++)
                crc[i+j] ^= crcg[j];

    for(i = INFO_LENGTH; i < INFO_LENGTH + CRCbit; i++)
        if(crc[i] ^ data[i])
            return(0);

    return(1);
}
```

Code8は、CRC デコーダの関数 `crc_decode` であり、入力の引数 `data` のビット列に対してCRCチェックを行い、CRCが正しければ1を返し、CRCが間違っていれば0を返す。

### A.3 畳み込み符号化・復号化

音響OFDMでは、OFDM伝送におけるビット誤りの訂正を畳み込み符号で行う。ここでは畳み込み符号の符号化と、復号に用いる軟判定ビタビ復号のプログラムについて記載する。

Code9

```
#define Nregister      (5)
#define Nrate         (3)
#define DATA_LENGTH (372)

void conv_encode(char *data, char *code)
{
    int i, j, k;
    int registers[Nregister + 1];
    int exclusive[Nrate];

    for(i = 0; i < (Nregister + 1); i++)
        registers[i] = 0;

    for(i = 0; i < DATA_LENGTH; i++){
        for(j = Nregister; j > 0; j--){
            registers[j] = registers[j-1];
            registers[0] = data[i];
        }

        for(j = 0; j < Nrate; j++){
            exclusive[j] = 0;
        }

        for(j = 0; j < (Nregister + 1); j++){
            for(k = 0; k < Nrate; k++){
                exclusive[k] ^= registers[j] & g[k][j];
            }
        }

        for(j = 0; j < Nrate; j++){
            code[(i * Nrate) + j] = exclusive[j];
        }
    }
}
```

Code9は畳込み符号化の関数 conv\_encode であり、引数の data によるビット列を畳み込み符号化して符号化出力のビット列を引数の code に格納する。



Code10

```
#define Nregister      (5)
#define Nrate         (3)
#define Nstate        (32)
#define DATA_LENGTH (372)
int g[Nrate][Nregister + 1] =
    { {1,0,0,1,1,1}, {1,0,1,0,1,1}, {1,1,1,1,0,1} };

double branch_metric_soft(int state, int branch, double *data)
{
    int i, j;
    double metric;
    int exclusive[Nrate];

    for(i = 0; i < Nrate; i++)
        exclusive[i] = 0;

    for(i = 0; i < Nregister; i++)
        for(j = 0; j < Nrate; j++)
            exclusive[j] ^= ((state >> i) & 0x01) & g[j][(Nregister - i)];

    for(i = 0; i < Nrate; i++)
        exclusiver[i] ^= branch & g[i][0];
    metric = 0.0;
    for(i = 0; i < Nrate; i++){
        if(exclusive[i]){
            metric += data[i];
        }else{
            metric -= data[i];
        }
    }
    return metric;
}

void viterbi_decode_soft(double *recv, char *recover)
{
    int i, state, pstate, branch, pbranch;
    double metric[Nstate][2], path_metric[Nstate],
           next_path_metric[Nstate];
    int path_memory[DATA_LENGTH][Nstate];
    double binary[2];
    int nerror;
```

```

path_metric[0] = 0;
for(i = 1; i < Nstate; i++)
    path_metric[i] = (-100) * DATA_LENGTH;;

for(i = 0; i < DATA_LENGTH; i++){
    for(state = 0; state < Nstate; state++)
        for(branch = 0; branch < 2; branch++)
            metric[state][branch]
= branch_metric_soft(state, branch, &recv[(i*Nrate)]);

for(state = 0; state < Nstate; state++){
    pbranch = (state >> (Nregister - 1)) & 0x01;
    pstate = (state << 1) & 0x1f;
    binary[0] = path_metric[pstate] + metric[pstate][pbranch];
    pstate = pstate + 1;
    binary[1] = path_metric[pstate] + metric[pstate][pbranch];

    if(binary[0] > binary[1]){
        path_memory[i][state] = 0;
        next_path_metric[state] = binary[0];
    }else{
        path_memory[i][state] = 1;
        next_path_metric[state] = binary[1];
    }
}
for(state = 0; state < Nstate; state++)
    path_metric[state] = next_path_metric[state];
}
/* Trace Back */
state = 0;
for(i = (DATA_LENGTH - 1); i >= 0; i--){
    recover[i] = (state >> (Nregister - 1)) & 0x01;
    if(path_memory[i][state] == 0){
        state = (state << 1) & 0x1f;
    }else{
        state = ((state << 1) & 0x1f) + 1;
    }
}
}
}

```

Code10は軟判定ビタビ復号の関数 `viterbi_decode_soft` とそれに用いるブランチメトリックを計算する関数 `branch_metric_soft` である。 `viterbi_decode_soft` は、軟値のビット列である `recv` を第1引数とし、軟判定ビタビ復号をして誤りビットを訂正したビット列を `recover` に格納する。

## A.4 インタリーブ・デインタリーブ

畳み込み符号はバーストエラーの訂正に弱いため、バーストエラーをランダムエラーに近くなるように、ビット系列を並べ替えるインタリーブとそれを元にもどすデインタリーブを行う。ここでは、本論文で説明した音響OFDMシステムにおけるインタリーブとデインタリーブのプログラムについて記載する。

Code11

```
#define CODE_LENGTH (1116)

void code_interleave(char *code)
{
    int i, m, n;
    char tmp1[CODE_LENGTH];
    char tmp2[CODE_LENGTH];

    for(i = 0; i < CODE_LENGTH; i++){
        if(i % 2){
            tmp1[CODE_LENGTH - 1 - (i / 2)] = code[i];
        }else{
            tmp1[i / 2] = code[i];
        }
    }

    m = 0; n = 0;
    for(i = 0; i < CODE_LENGTH; i++){
        tmp2[m+n] = tmp1[i];
        m += Ninterleave;
        if(m >= CODE_LENGTH){
            m = 0; n++;
        }
    }

    for(i = 0; i < CODE_LENGTH; i++)
        code[i] = tmp2[i];
}
```

Code11は符号化ビット列をインタリーブする関数code\_interleaveであり、引数のビット列codeに対して、本稿で記載したインタリーブ1とインタリーブ2を順番に行い、インタリーブ後のビット列をcodeに格納する。

Code12

```
#define CODE_LENGTH (1116)

void code_deinterleave(double *code)
{
    int i, m, n;
    double tmp1[CODE_LENGTH];
    double tmp2[CODE_LENGTH];

    m = 0; n = 0;
    for(i = 0; i < CODE_LENGTH; i++){
        tmp1[i] = code[m+n];
        m += Ninterleave;
        if(m >= CODE_LENGTH){
            m = 0; n++;
        }
    }

    for(i = 0; i < CODE_LENGTH; i++){
        if(i % 2){
            tmp2[i] = tmp1[CODE_LENGTH - 1 - (i / 2)];
        }else{
            tmp2[i] = tmp1[i / 2];
        }
    }

    for(i = 0; i < CODE_LENGTH; i++)
        code[i] = tmp2[i];
}
```

Code12は、受信した軟値系列をデインタリーブする関数 `code_deinterleave` であり、引数である軟値の系列 `code` に対して、本稿で記載したインタリーブ方法の逆変換をインタリーブ2、インタリーブ1の順番で行い、変換後の系列を `code` に格納する。

## A.5 シンボル同期信号生成・処理プログラム

音響 OFDM では OFDM シンボルの同期に、PN 系列による同期信号を生成する。ここでは、シンボル同期信号の生成および同期処理のプログラム例を記載する。

```
Code13
#define Fsmp          (2032)
#define CYCLE         (63)
#define CHIPWIDTH     (32)
#define GENERATOR     (11)
#define MAXBIT        (0x0040)
#define XORBIT        (0x0043)

char * pngen()
{
    int i;
    unsigned int shift = GENERATOR;
    char *pn;
    pn = (char *)calloc(CYCLE, sizeof(char));

    for(i = 0; i < CYCLE; i++){
        if(shift & 0x01){
            pn[i] = 1;
        }else{
            pn[i] = 0;
        }
        shift = shift << 1;
        if(shift & MAXBIT){
            shift = shift ^ XORBIT;
        }
    }
    return(pn);
}
```

Code13 は OFDM フレーム同期信号生成用のビット系列生成関数 pngen である。pngen は信号生成に用いるビット系列を生成して戻り値として返す。ビット系列の生成には、生成多項式  $x^6 + x + 1$  の M 系列符号を用いる。

Code14

```
void syncgen(char *xn, char *pn)
{
    int i, j;
    double pskw = (2 * M_PI) / CHIPWIDTH;
    bzero(xn, (sizeof(double) * Fsmp));

    for(i = 0; i < CYCLE; i++){
        if(pn[i]){
            for(j = 0; j < CHIPWIDTH; j++)
                xn[(i * CHIPWIDTH) + j] = sin(pskw * j);
        }else{
            for(j = 0; j < CHIPWIDTH; j++)
                xn[(i * CHIPWIDTH) + j] = (-1) * sin(pskw * j);
        }
    }
}
```

Code14は、信号生成関数のsyncgenである。syncgenは、pngenで生成したビット系列を引数pnとして、pnから1サンプルあたりの角度変移pskwの正弦波を用いてOFDMフレーム同期信号を生成する。生成した信号を引数xnに格納する。

Code15

```
#define Fsmp                (2032)
#define CYCLE                (63)
#define CHIPWIDTH           (32)

int symbol_sync(short *x, char *pn)
{
    int i, j, k, n, tw;
    int sum[Fsmp], val, max;

    for(i = 0; i < Fsmp; i++)
sum[i] = 0;
    for(i = 0; i < (Nframe + 1); i++){
for(j = 0; j < Fsmp; j++){
    n = (i * Fsmp) + j;
    for(k = 0; k < CYCLE; k++){
if(pn[k]){
    sum[j] += x[n];
}else{
    sum[j] -= x[n];
}
n += CHIPWIDTH;
    }
    }
    }
    n = 0;  max = 0;
    tw = CHIPWIDTH / 2;
    for(i = 0; i < Fsmp - tw; i++){
val = sum[i] - sum[(i + tw)];
if(val > max){
    max = val;  n = i;
}
    }
    for(i = Fsmp - tw; i < Fsmp; i++){
val = sum[i] - sum[(i + tw - Fsmp)];
if(val > max){
    max = val;  n = i;
}
    }
    n = (n + Fsmp - (tw / 2)) % Fsmp;
    return n;
}
```



Code15 は、OFDM シンボル同期をとる関数 `symbol_sync` であり、入力信号系列 `x` と PN 系列 `pn` を引数として、`x` と `pn` の相関をとり、相関値が最も高くなるサンプル点を先頭からのオフセットとして戻り値で返す。

## A.6 データフレーム同期処理プログラム

音響 OFDM ではデータフレームの同期に、BCH 符号による巡回符号系列を用いる。巡回符号系列はシングルフレーム伝送用とマルチフレーム伝送用に以下の 2 種類の符号系列を用いる。

【シングルフレーム伝送用】

0,0,0,0,0,1,1,1,0,0,1,0,0,0,1,0,1,0,1,1,1,1,0,1,1,0,1,0,1,0,0,1,1

【マルチフレーム伝送用】

0,0,0,0,1,0,0,1,0,1,1,0,0,1,1,1,1,1,0,0,0,1,1,0,1,1,1,0,1,0,1,0,1

受信側では符号系列の種類でシングルフレーム伝送かマルチフレーム伝送かを判別し、巡回ビット数でデータフレームの同期を行う。ここでは、伝送された巡回符号系列の誤り訂正を行う BCH 復号化処理のプログラム例を記載する。

Code16

```
#define BCH_CODELEN      (31)    /* code length */
#define BCH_INFOLEN     (6)     /* information length */
#define DEGREE          (25)    /* degree of polynomial */
#define DIST            (15)    /* code distance */
#define Nerror          (7)     /* number of lost bits */
#define GFSIZE          (32)
#define GFPRIM          (37)

unsigned char gflog[GFSIZE];
unsigned char gfexp[GFSIZE];

/*
 * Generator Polynomial
 * n = 31, k = 6, d = 15 (t = 7)
 */
unsigned char gen_poly[BCH_CODELEN] =
{1,1,1,0,0,1,0,0,0,1,0,1,0,1,1,1,1,0,1,1,0,1,0,0,1,1,0,0,0,0,0};

void gf_init()
{
    unsigned int n, log;

    n = 1;
    for(log = 0; log < (GFSIZE - 1); log++){
        gflog[n] = log;
        gfexp[log] = n;
        n = n << 1;
        if(n >= GFSIZE){
            n = n ^ GFPRIM;
        }
    }
}
```

Code16は、BCH符号の演算に用いるガロア体の初期化を行う。ガロア体はGF(32)であり、乗算、除算のためのテーブルを作成する。GF(32)はGF(2)の拡大体であり、生成多項式  $x^5 + x^2 + 1$  を用いて生成される。

Code17

```
unsigned char gfmult(a, b)
    unsigned char a, b;
{
    unsigned int sum;

    if(a == 0 || b == 0){
        return 0;
    }
    sum = gflog[a] + gflog[b];
    if(sum >= (GFSIZE - 1)){
        sum -= (GFSIZE - 1);
    }
    return gfexp[sum];
}
```

Code17はBCH符号の演算に用いるガロア体の乗算 gfmult であり,  $a*b$  の演算結果を返す.

Code18

```
unsigned char gfddiv(a, b)
    unsigned char a, b;
{
    int sum;

    if(a == 0) return 0;
    if(b == 0) return -1;
    sum = gflog[a] - gflog[b];
    if(sum < 0) sum += (GFSIZE - 1);
    return gfexp[sum];
}
```

Code18はBCH符号の演算に用いるガロア体の除算 gfddiv であり,  $a/b$  の演算結果を返す.

Code19

```
unsigned char gfdiv(a, b)
    unsigned char a, b;
{
    int sum;

    if(a == 0) return 0;
    if(b == 0) return -1;
    sum = gflog[a] - gflog[b];
    if(sum < 0) sum += (GFSIZE - 1);
    return gfexp[sum];
}
```

Code19はBCH符号の演算に用いるガロア体のべき乗gfpowerであり、 $a^b$ の演算結果を返す。

Code20

```
int bch_decode(a)
    unsigned char *a;
{
    int i, j, k, n;
    int nerror = Nerror;
    int complete;
    unsigned char s[DIST];           /* syndrome vector */
    unsigned char elp[Nerror];      /* error locator */
    unsigned char errorpos[Nerror]; /* error position */
    unsigned char matrix[Nerror][Nerror]; /* decoding matrix */
    unsigned char inverse[Nerror][Nerror]; /* inverse matrix */
    unsigned char val, check;

    /* calculating syndromes */
    complete = 1;
    for(i = 0; i < DIST; i++){
        s[i] = 0;
        for(j = 0; j < BCH_CODELEN; j++){
            if(a[j]){
s[i] ^= gfexp[(i * j) % (GFSIZE - 1)];
            }
        }
        if(s[i] != 0){
            complete = 0;
        }
    }

    if(complete){
        return(0);
    }
}
```

```

while(complete == 0){
    complete = 1;
    /* calculating decoding matrix */
    for(i = 0; i < nerror; i++){
        for(j = 0; j < nerror; j++){
            matrix[i][j] = s[(nerror + i) - j];
        }
    }
    /* inverting matrix (gaussian elimination) */
    for(i = 0; i < nerror; i++){
        for(j = 0; j < nerror; j++){
            if(i == j){
                inverse[i][j] = 1;
            }else{
                inverse[i][j] = 0;
            }
        }
    }
    for(i = 0; i < nerror; i++){
        n = matrix[i][i];
        if(n == 0){
            complete = 0;
            nerror--;
            break;
        }
        for(j = 0; j < nerror; j++){
            matrix[i][j] = gfdiv(matrix[i][j], n);
            inverse[i][j] = gfdiv(inverse[i][j], n);
        }
    }
}

```

```

for(k = 0; k < nerror; k++){
    if(k == i){
        continue;
    }
    n = matrix[k][i];
    for(j = 0; j < nerror; j++){
        matrix[k][j] ^= gfmult(matrix[i][j], n);
        inverse[k][j] ^= gfmult(inverse[i][j], n);
    }
}
}

/* matrix multiplication (inverse * syndromes) */
for(i = 0; i < nerror; i++){
    elp[i] = 0;
    for(j = 0; j < nerror; j++){
        elp[i] ^= gfmult(inverse[i][j], s[(nerror + 1) + j]);
    }
}

/* determine error position */
n = 0;
for(i = 0; i < BCH_CODELEN; i++){
    val = 1;
    for(j = 0; j < nerror; j++){
        val ^= gfmult(elp[j], gfexp[(i * (j+1)) % (GFSIZE - 1)]);
    }
    if(val == 0){
        if(i == 0){
            errorpos[n] = 0;
        }else{
            errorpos[n] = BCH_CODELEN - i;
        }
        n++;
    }
}
}

```

```

/* error check */
check = BCH_CODELEN;
for(i = 0; i < nerror; i++){
    if(errorpos[i] >= check){
        return -1;
    }
    if((i!=0) || (errorpos[i] != 0)){
        check = errorpos[i];
    }
}

/* recover error */
for(i = 0; i < nerror; i++){
    if(errorpos[i] < BCH_CODELEN){
        a[errorpos[i]] ^= 0x01;
    }
}

return(nerror);
}

```

Code20 は、BCH 符号の符号化処理であり、ビット列 a を BCH 復号し、誤りのあるビットを訂正する。訂正したビット数を戻り値として返す。



## 本論文に関する公表論文リスト

### 査読付き学術論文

1. Matsuoka, H., Nakashima, Y., & Yoshimura, T. (2009). Acoustic OFDM system and its extension. *The Visual Computer*, 25(1), 3-12.
2. Matsuoka, H., Nakashima, Y., & Yoshimura, T. (2008). Acoustic OFDM system and performance analysis. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 91(7), 1652-1658.

### 査読付き国際会議

1. Matsuoka, H., Nakashima, Y., Yoshimura, T. & Kawahara, T. (2008 January), Acoustic OFDM: Embedding High Bit Rate Data in Audio. *Proc. In 14<sup>th</sup> International Multimedia Modeling Conference 2008. MMM2008.*
2. Matsuoka, H. (2006, December). Spread spectrum audio steganography using sub-band phase shifting. *Proc. In Intelligent Information Hiding and Multimedia Signal Processing, 2006. IHH-MSP'06. International Conference on (pp. 3-6). IEEE.*
3. Matsuoka, H., Nakashima, Y., & Yoshimura, T. (2006, September). Acoustic communication with OFDM signal embedded in Audio. *Proc. In Audio Engineering Society Conference: 29th International Conference: Audio for Mobile and Handheld Devices. Audio Engineering Society.*

### 解説記事

1. 松岡保静. (2012). 音響データ通信技術: 音響 OFDM (< 小特集 > 携帯情報機器における音響技術). *日本音響学会誌*, 68(3), 143-147.
2. Matsuoka, H. (2012). Acoustic OFDM Technology and System. *Multimedia Information Hiding Technologies and Methodologies for Controlling Data (Chapter 5, pp.90-103).*