

Mário Clemente Lacerda da Cruz Júnior

Uma Técnica para Ocultação de Partituras Musicais em Sinais de Áudio

Brasil

23 de janeiro de 2018

Mário Clemente Lacerda da Cruz Júnior

Uma Técnica para Ocultação de Partituras Musicais em Sinais de Áudio

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como parte dos requisitos para obtenção do grau de Mestre em Engenharia Elétrica. Área de Concentração: Comunicações.

Universidade Federal de Pernambuco – UFPE

Centro de Tecnologia de Geociências

Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Juliano Bandeira Lima

Brasil

23 de janeiro de 2018

*A minha amada noiva,
Joyce Layane de Lima Silva,
por toda paciência e apoio incondicional
desde o início até o fim desta caminhada,
ao meu sobrinho, **Pedro Henrique Lacerda da Silva,**
que tanto alegrou os meus dias mais difíceis,
e ao amigo **Elias Albuquerque** pela amizade sincera.*

Resumo

Nas últimas décadas, técnicas voltadas à ocultação de dados têm desempenhado um importante papel no contexto de segurança de informação multimídia. Neste cenário, em que se destacam esquemas de esteganografia e de marca d'água para áudio, vídeo e imagem, têm sido empregados métodos interessantes e que podem ser estendidos a aplicações correlatas às mencionadas. Nesta dissertação, propõe-se o uso de um desses métodos, o qual se baseia na transformada discreta de Fourier e na operação de modulação de fase, para ocultar partituras musicais em sinais de áudio. Essencialmente, o que se faz é, dada uma música, considerar sua partitura e escrevê-la na notação ABC, a qual emprega apenas caracteres ASCII; esses caracteres são representados por meio de uma sequência binária, a qual é ocultada no sinal de áudio da música em questão. São apresentados resultados de experimentos computacionais que indicam que a degradação causada pela inserção dos bits no áudio original é tolerável do ponto de vista de percepção auditiva e que a sequência binária pode ser extraída com taxa de erro aceitável, ainda que o áudio seja submetido a determinadas técnicas de compressão. Da referida sequência, pode-se, então, recuperar a partitura na notação ABC e, conseqüentemente, em sua forma original, isto é, com todos os símbolos gráficos usualmente empregados na escrita musical. A proposta introduzida neste trabalho permite, por exemplo, que alguém que se interesse por tocar uma música obtida num dos formatos de arquivo de áudio padrão, obtenha simultaneamente a sua partitura, sem a necessidade de um arquivo anexo ou de uma busca na Internet. Outras propostas que contemplem o cenário delineado nesta dissertação não foram encontradas na literatura.

Palavras-chave: Processamento de Sinais de Áudio. Modulação de Fase. Notação ABC. Ocultação de Dados.

Abstract

In the last decades, techniques devoted to data hiding have been performed an important role in the context of multimedia security. In this scenario, where steganographic and watermarking schemes applicable to digital audio, video and image are remarkable, interesting methods extensible to applications correlated to the mentioned ones have been employed. In this dissertation, one proposes the use of one of such methods, which is based on the discrete Fourier transform and on the phase modulation operation, to hide musical scores in audio signals. Essentially, given a music, one considers its score and writes it using the ABC notation, which employs ASCII characters only; these characters are represented by means of a binary sequence, which is embedded in the respective audio signal. We perform computational experiments whose results indicate that the degradation due to the embedment of bits in the original audio is tolerable from the point of view of auditory perception. Moreover, the binary sequence can be extracted with acceptable error rate, even if the audio is submitted to certain compression techniques. From the referred sequence, we then recover the score in the ABC notation and, consequently, in its original form, that is, with all graphical symbols usually employed in the musical writing. The approach introduced in the current work allows, for example, a person interested in playing a music obtained in a standard audio file format to simultaneously obtain its score, without the need of a complimentary file or an Internet search. Other approaches fitting the scenario outlined in this dissertation were not found in the literature.

Keywords: Audio Signal Processing. Phase Modulation. ABC Notation. Data Hiding.

Lista de ilustrações

Figura 1 – Sistema esteganográfico genérico	19
Figura 2 – Procedimento para codificação de paridade	24
Figura 3 – Esquema básico de inserção de uma marca d'água e de sua extração	27
Figura 4 – Processamento de tempo discreto de sinal de tempo contínuo	34
Figura 5 – Amostragem com trem de impulsos	35
Figura 6 – Amostragem com retentor de zero	35
Figura 7 – Diagrama de bloco de um típico codificador MPEG-1 Layer-3	41
Figura 8 – Partitura da notação ABC escrita acima.	44
Figura 9 – Diagrama de blocos do processo de inserção de bits em um sinal de áudio.	46
Figura 10 – Representação das modificações sofridas por uma componente de um vetor, originalmente de amplitude ρ_0 e fase φ , em função da inserção de um bit via modulação de fase, resultando em uma amplitude de ρ e um acréscimo de δ na fase.	47
Figura 11 – Parte imaginária da DFT de um vetor real, com ênfase para a simetria ímpar entre as componentes.	49
Figura 12 – Diagrama de blocos do processo de extração dos bits.	51
Figura 13 – Partitura da canção <i>All my loving</i> na notação ABC, recuperada após a sequência de bits ter sido extraída com 1 bit errado.	60
Figura 14 – Partitura da canção <i>All my loving</i> na notação ABC, recuperada após a sequência de bits ter sido extraída com 11 bits errados.	61
Figura 15 – Partitura da canção <i>All my loving</i> na notação ABC, recuperada após a sequência de bits ter sido extraída com 33 bits errados.	62
Figura 16 – Partitura da canção <i>All my loving</i> na notação usual, recuperada após a sequência de bits ter sido extraída com 1 bit errado.	63
Figura 17 – Partitura da canção <i>All my loving</i> na notação usual, recuperada após a sequência de bits ter sido extraída com 33 bits errados.	64
Figura 18 – Representação gráfica das posições dos bits extraídos com erro, ao longo da sequência de bits que representa a partitura em notação ABC: (a) 1 erro, (b) 11 erros e (c) 33 erros.	65
Figura 19 – Partitura da canção <i>All my loving</i> na notação ABC.	75
Figura 20 – Partitura da canção <i>All my loving</i> na notação usual.	76
Figura 21 – Partitura da canção <i>Here, there and everywhere</i> na notação ABC.	77
Figura 22 – Partitura da canção <i>Here, there and everywhere</i> na notação usual.	77
Figura 23 – Primeira parte da partitura da canção <i>Yesterday</i> na notação ABC.	78
Figura 24 – Segunda parte da partitura da canção <i>Yesterday</i> na notação ABC.	78
Figura 25 – Partitura da canção <i>Yesterday</i> na notação usual.	79

Lista de tabelas

Tabela 1	– Números de bits extraídos com erro para diversos formatos de áudio mono e diferentes combinações dos parâmetros $step$ ($\pi/6, \pi/7, \pi/8$) e γ , para a música <i>All my loving</i>	57
Tabela 2	– Números de bits extraídos com erro para diversos formatos de áudio mono e diferentes combinações dos parâmetros $step$ ($\pi/6, \pi/7, \pi/8$) e γ , para a música <i>Here, there and everywhere</i>	58
Tabela 3	– Números de bits extraídos com erro para diversos formatos de áudio mono e diferentes combinações dos parâmetros $step$ ($\pi/6, \pi/7, \pi/8$) e γ , para a música <i>Yesterday</i>	58
Tabela 4	– Números de bits extraídos com erro para diversos formatos de áudio estéreo e diferentes combinações dos parâmetros $step$ ($\pi/6, \pi/7, \pi/8$) e γ , para a música <i>All my loving</i>	58
Tabela 5	– Números de bits extraídos com erro para diversos formatos de áudio estéreo e diferentes combinações dos parâmetros $step$ ($\pi/6, \pi/7, \pi/8$) e γ , para a música <i>Here, there and everywhere</i>	59
Tabela 6	– Números de bits extraídos com erro para diversos formatos de áudio estéreo e diferentes combinações dos parâmetros $step$ ($\pi/6, \pi/7, \pi/8$) e γ , para a música <i>Yesterday</i>	59

Sumário

1	INTRODUÇÃO	15
1.1	Justificativa e Caracterização da Proposta	16
1.2	Objetivos	17
1.3	Estrutura e Contribuições da Dissertação	18
2	OCULTAÇÃO DE DADOS EM CONTEÚDO MULTIMÍDIA DIGITAL	19
2.1	Esteganografia	19
2.1.1	Esteganografia em arquivos de imagem	20
2.1.1.1	Domínio espacial	21
2.1.1.2	Domínio da transformada	21
2.1.2	Esteganografia em arquivos de vídeo	22
2.1.3	Esteganografia em arquivos de áudio	22
2.1.3.1	Codificação do LSB	23
2.1.3.2	Codificação de Paridade	23
2.1.3.3	Codificação de Fase	23
2.1.3.4	Espalhamento espectral	25
2.1.3.5	Ocultação de eco	25
2.2	Marca D'Água	26
2.2.1	Propriedades da marca d'água digital	27
2.2.2	Marca d'água em arquivos de imagem	28
2.2.3	Marca d'água em arquivos de áudio	29
2.2.3.1	Requisitos para marca d'água em áudio	29
2.2.3.2	Aplicações para a marca d'água em sinais de áudio	30
2.2.3.3	Técnicas de Marca D'água em Áudio	31
3	ÁUDIO DIGITAL: REPRESENTAÇÃO E PADRÃO MP3	33
3.1	Amostragem	33
3.1.1	Representação de sinais de tempo contínuo a partir de suas amostras	34
3.1.2	Reconstrução de sinais contínuos a partir de suas amostras	35
3.2	Quantização de Sinais de áudio	36
3.3	Codificação de áudio	37
3.3.1	Representação de Áudio Baseada em Amostragem	37
3.3.1.1	Pulse Code Modulation - PCM	37
3.3.1.2	Compact Disc Digital Audio - CD-DA	37
3.3.1.3	Differential Pulse Code Modulation - DPCM	38
3.3.2	Representação de Áudio Baseada em Processamento Digital de Sinais	38

3.3.2.1	AIFF - Audio Interchange File Format	38
3.3.2.2	WAV - Waveform Audio Data	38
3.3.2.3	MPEG-1, MPEG-2 E MPEG-4	39
3.3.2.4	AAC - Advanced Audio Coding	39
3.3.2.5	Ogg Vorbis	39
3.3.2.6	WMA - Windows Audio Media	39
3.4	O Padrão de Compressão MP3	40
3.4.1	Modo de operação	40
3.4.2	Frequência de Amostragem	40
3.4.3	<i>Bitrate</i> (taxa de bits)	40
3.4.4	Normativo <i>versus</i> Informativo	41
3.4.5	Descrição do Algoritmo	41
4	OCULTAÇÃO DE PARTITURAS EM SINAIS DE ÁUDIO DIGITAL	43
4.1	A Notação ABC para Escrita de Partituras Musicais	43
4.1.1	Representação em Bits de Partituras Escritas em Notação ABC	45
4.2	Inserção dos Bits	45
4.2.1	Adaptação para Áudio Mono	49
4.2.2	Adaptação para Áudio Estéreo	50
4.3	Extração dos Bits	51
5	SIMULAÇÕES E RESULTADOS	53
5.1	Definição dos Parâmetros Iniciais	53
5.1.1	Áudios Utilizados	53
5.1.2	Transcrição da Partitura em Notação Musical para Notação ABC	53
5.2	Inserção de Bits	54
5.2.1	Adaptação para Áudio Mono	54
5.2.2	Adaptação para Áudio Estéreo	55
5.3	Extração dos bits	56
5.4	Avaliação Preliminar dos Efeitos da Extração Incorreta dos Bits	58
6	CONCLUSÕES	67
6.1	Trabalhos Futuros	67
6.2	Artigos Publicados	67
	REFERÊNCIAS	69

APÊNDICES **73**

	APÊNDICE A – PARTITURAS UTILIZADAS NAS SIMULAÇÕES	75
A.1	Faixa 1 - <i>All my loving</i>	75
A.2	Faixa 2 - <i>Here, there and everywhere</i>	76
A.3	Faixa 3 - <i>Yesterday</i>	76

1 Introdução

Nas últimas décadas, as redes de comunicação de dados têm experimentado mudanças que vem trazendo uma série de impactos na sociedade. Tais mudanças, as quais são movidas principalmente pela introdução de novas técnicas implementadas em circuitos eletrônicos e em *software*, têm reflexo direto na expansão dessas redes, tanto do ponto de vista de abrangência geográfica quanto do de capacidade de transmissão (STALLINGS, 2013; BARABÁSI; PÓSFAL, 2016; KUROSE; ROSS, 2016). Essa evolução tem sido acompanhada pela popularização de dispositivos que permitem a conexão a essas redes, como *tablets* e *smartphones*, os quais possuem custo relativamente baixo e possibilitam, com mobilidade e autonomia aceitáveis, o compartilhamento de informações, o envio de mensagens, o acesso a sítios da *web* etc. (CHOWDHURY; BISWAS, 2017; OSSEIRAN et al., 2016)

Diferentemente do que se tinha há poucos anos, as taxas de transmissão de dados que, atualmente, alcançam o usuário final de uma rede de comunicação permitem, com certa facilidade, a troca de mensagens com conteúdo multimídia de alta qualidade; isso inclui, basicamente, imagens, vídeos e áudio digitais, produzidos das mais diversas formas e, quase sempre, deixados à disposição de quem os queira manipular. Tal possibilidade dá margem, por exemplo, para que uma imagem com conteúdo falso seja criada ou para que, num arquivo de áudio, seja feita uma edição que distorça o diálogo que tenha sido originalmente registrado (GONZALEZ; WOODS, 2017; TEKALP, 2015; KAMENOV, 2014).

Diante do exposto, o uso de mecanismos voltados à segurança de informação multimídia tem se tornado essencial nos mais diferentes cenários de aplicação. Considerando o escopo em que se encontra o presente trabalho, três desses mecanismos podem ser destacados: a cifragem, a marca d'água e a esteganografia (SHIH, 2012; FURHT; KIROVSKI, 2006; HE, 2011; GHORBANI; AMIRI, 2016). O número de métodos arquivados na literatura e nos quais esses mecanismos se baseiam é de ordem elevada, sendo impraticável elaborar uma revisão que os contemple a todos. De qualquer modo, definir concisamente em que consiste cada um desses mecanismos é algo que colabora para a contextualização apresentada nesta introdução.

A cifragem de conteúdo multimídia possui o mesmo objetivo da cifragem realizada por criptossistemas de chave secreta usuais, isto é, modificar um *texto claro* num *texto cifrado*, de forma que a informação seja inacessível à parte não-autorizada. A questão é que, quando se trata de conteúdo multimídia, algumas especificidades precisam ser consideradas; um mecanismo de cifragem usual poderia falhar, por exemplo, no “embaralhamento” dos

pixels de uma imagem, caso essa tivesse regiões muito extensas com a mesma intensidade, como um fundo preto numa imagem de exame médico ou um céu predominantemente azul na fotografia de uma paisagem. Fragilidades desse tipo, que poderiam ser exploradas em algumas técnicas de criptanálise, são levadas em consideração quando da concepção de técnicas voltadas à cifragem de conteúdo multimídia.

Técnicas de marca d'água e de esteganografia têm por objetivo ocultar, de maneira estatística e visualmente invisível, dados num arquivo com conteúdo multimídia. Essas técnicas se utilizam, em muitos casos, de métodos e ferramentas matemáticas similares. Porém, no caso da marca d'água, os dados a serem ocultados têm o propósito de servir como uma espécie de identificador a ser empregado na proteção do arquivo multimídia; isso pode permitir à parte receptora do arquivo a detecção de manipulações não autorizadas que ele tenha sofrido e a confirmação da identidade do remetente ou do detentor dos direitos autorais sobre ele. No caso da esteganografia, o interesse é propriamente nos dados a serem ocultados, cujo conteúdo se pretende comunicar ou armazenar de forma secreta. Isso significa que o arquivo em que os referidos dados são escondidos não possui importância aparente; este fato pode ser usado para “despistar” entidades interessadas em realizar ataques criptográficos.

Cifragem, marca d'água e esteganografia são aplicáveis a arquivos de áudio digital e, para isso, precisam levar em consideração as peculiaridades desse tipo de conteúdo. Tais peculiaridades incluem, por exemplo, questões relacionadas a codificação do áudio, que, por sua vez, têm relação com o número de níveis de quantização e com o número de bits empregados para representar cada amostra; o uso de mecanismos de compressão com perda e o fato de se ter um ou mais canais (áudio mono, estéreo etc.) também precisam ser levados em conta. Quando se trata especificamente da ocultação de dados em áudio digital (marca d'água ou esteganografia), precisa-se considerar, ainda, os efeitos que a técnica empregada provoca no que diz respeito à percepção auditiva. Idealmente, o inserção dos bits com a mensagem (marca d'água propriamente dita ou mensagem secreta na esteganografia) não deve degradar a qualidade do áudio original, isto é, antes da ocultação.

Uma revisão concisa das principais técnicas relacionadas à ocultação de dados em arquivos com conteúdo multimídia é apresentada no Capítulo 2 desta dissertação.

1.1 Justificativa e Caracterização da Proposta

O presente trabalho se enquadra num contexto bastante peculiar, pois aborda o tema de ocultação de dados em áudio digital num cenário em que não há interesse em preencher certos requisitos criptográficos; não se deseja confirmar a identidade do emissor de uma mensagem, nem transmitir dados de forma secreta. O que se pretende é explorar técnicas de ocultação de dados para embutir num sinal de áudio e, de modo

ainda mais particular, num arquivo de música, informação extra a respeito do conteúdo desse arquivo. Tal informação consiste na partitura da própria música contida no arquivo, a qual, inicialmente escrita numa notação denominada ABC, é facilmente convertida numa sequência binária. Essa sequência é, então, inserida no sinal de áudio empregando uma versão adaptada de um método denominado “modulação de fase”. Posteriormente, pode-se extrair a referida sequência binária e remontar a partitura da música tanto na notação ABC quanto empregando os símbolos musicais usuais. O desenvolvimento desta ideia se justifica, principalmente, pela sua originalidade, uma vez que abordagens similares não foram encontradas na literatura, e pelos avanços que os resultados obtidos podem representar no tocante à disponibilização de novas ferramentas voltadas ao entretenimento e à educação musical. A proposta introduzida neste trabalho permite, por exemplo, que alguém que se interesse por tocar uma música obtida num dos formatos de arquivo de áudio padrão, obtenha simultaneamente a sua partitura, sem a necessidade de um arquivo anexo ou de uma busca na Internet.

1.2 Objetivos

O objetivo geral desta dissertação é a proposição de uma técnica para ocultação de partituras musicais em sinais de áudio digital, empregando operações que têm sido predominantemente aplicadas a estes sinais para inserção e extração de marcas d’água.

Como objetivos específicos deste trabalho, podem ser elencados os seguintes:

1. Conhecer os aspectos distintivos das diversas técnicas para ocultação de dados em arquivos com conteúdo multimídia;
2. Identificar as questões mais relevantes no que diz respeito ao áudio digital, com ênfase nas técnicas empregadas em sua representação;
3. Reconhecer as principais características da notação musical ABC, propondo uma metodologia para conversão sistemática para binário dos símbolos empregados nesta notação;
4. Introduzir, empregando operações como a transformada discreta de Fourier e a modulação de fase, um método para ocultar a partitura de uma música (sequência binária associada à partitura na notação ABC) no respectivo arquivo de áudio digital;
5. Propor adaptações do método de ocultação mencionado em função do formato do arquivo de áudio considerado (mono ou estéreo);
6. Realizar simulações computacionais com vista à avaliação do método proposto, sobretudo com relação à sua robustez contra erros na extração da partitura ocultada quando da compressão mp3 do arquivo de áudio.

1.3 Estrutura e Contribuições da Dissertação

Esta dissertação se encontra organizada da seguinte forma:

- No Capítulo 1, é realizada uma contextualização do trabalho, sendo apresentada, de forma concisa, a ideia desenvolvida e sendo colocados os principais motivos para a sua realização e os objetivos inicialmente delineados.
- O Capítulo 2 apresenta uma revisão bibliográfica sobre ocultação de dados em conteúdo multimídia digital. São apresentados, basicamente, os principais conceitos relacionados à esteganografia e à marca d'água, e descritas algumas técnicas importantes nesse contexto.
- No Capítulo 3, é apresentada uma revisão sobre áudio digital, com foco nas técnicas empregadas em sua representação e no padrão de compressão mp3. O conhecimento desse conteúdo é de suma importância para que, no desenvolvimento da ideia proposta nesta dissertação, seja possível selecionar a metodologia mais adequada para ocultar dados num sinal de áudio e avaliar os resultados obtidos sobretudo quando este sinal é submetido a perdas.
- No Capítulo 4, é descrita a metodologia para ocultação de partituras musicais proposta neste trabalho. Tal metodologia, a qual constitui a principal contribuição desta dissertação, emprega fundamentalmente a transformada discreta de Fourier e a operação de modulação de fase. Esta última permite que, sem que qualquer informação adicional seja armazenada, seja extraída do arquivo de áudio (música) a sequência de bits que representa sua partitura, a qual pode, posteriormente, ser convertida para o formato ABC ou mesmo para a notação musical convencional.
- O Capítulo 5 contém uma descrição a respeito das simulações computacionais realizadas e uma análise dos resultados obtidos. Para os arquivos de áudio considerados nos experimentos, verifica-se que, ainda que o arquivo com a partitura inserida seja submetido à compressão mp3, pode-se recuperar os bits que representam a partitura com um número aceitável de erros, isto é, com um número de erros que permite a reconstrução da referida partitura.
- O Capítulo 6 traz as principais conclusões do trabalho e indica sugestões para a continuidade da pesquisa.

2 Ocultação de Dados em Conteúdo Multimídia Digital

Quando se deseja usar alguma técnica para ocultar dados em conteúdo multimídia digital, tem-se como principais premissas a imperceptibilidade desses dados, após a sua ocultação, e a fidelidade do referido conteúdo multimídia em comparação à sua versão original. As técnicas utilizadas para esconder informação são principalmente duas: esteganografia e marca d'água, as quais são abordadas nos tópicos a seguir (AL-OTTHMANI; MANAF; ZEKI, 2012), (ALSALAMI; AL-AKAIDI, 2003), (CHAUHAN; RIZVI, 2013), (PROVOS; HONEYMAN, 2003), (FRIDRICH; GOLJAN; DU, 2001).

2.1 Esteganografia

O termo esteganografia é derivado do grego, a partir das palavras *stegos*, que significa *disfarce*, e *grafia* que significa *escrita*, significando em contexto geral *escrita disfarçada* (NOSRATI; KARIMI; HARIRI, 2012). Falar em “proteção de dados” traz à mente o conceito de Criptografia, que consiste em modificar uma mensagem que se deseja proteger, misturando-a e desordenando-a antes de sua transmissão por um canal inseguro (AL-OTTHMANI; MANAF; ZEKI, 2012), (NISSAR; MIR, 2010). A esteganografia, por sua vez, tem o objetivo de ocultar uma determinada mensagem inserida em outra mensagem, de mesmo formato ou não, no processo esteganográfico.



Figura 1 – Sistema esteganográfico genérico

Em sistemas esteganográficos modernos, além da estrutura apresentada, são empregadas chaves secretas (AL-OTTHMANI; MANAF; ZEKI, 2012), (PROVOS; HONEYMAN, 2003). Na fig. 1 é possível observar um sistema esteganográfico genérico, onde são vistos o arquivo de disfarce, a mensagem secreta a ser escondida, a chave secreta, as figuras do emissor e do receptor, o processo de ocultação da mensagem secreta, o arquivo estego que contém a mensagem secreta, o processo de extração da mensagem e, por fim, a mensagem secreta extraída e o arquivo de disfarce (JAYARAM; RANGANATHA; ANUPAMA, 2011).

É possível definir uma estrutura padrão para esteganografia e especificar todos elementos que a compõem (AL-OTHMANI; MANAF; ZEKI, 2012):

1. Mensagem secreta: consiste de um arquivo digital de qualquer formato (imagem, vídeo, texto, áudio, etc) a ser inserido (escondido) em um arquivo digital também de qualquer formato;
2. Arquivo de disfarce: é o arquivo digital (áudio, vídeo, imagem, etc) no qual será inserida a mensagem secreta;
3. Chave: é a chave que garante maior segurança a mensagem secreta, de conhecimento do emissor e do receptor legítimos; também é inserida no arquivo de disfarce;
4. Arquivo estego: é o arquivo resultante do processo de esteganografia.
5. Processo de ocultação: processo por meio do qual o emissor esconde a mensagem secreta no arquivo de disfarce.
6. Processo de extração: processo por meio do qual o receptor pode recuperar a mensagem secreta.

Ainda sobre a inserção da mensagem secreta, (PROVOS; HONEYMAN, 2003) são definidas três características diferentes e importantes: capacidade, segurança e robustez. Capacidade diz respeito a quantidade de informação que pode ser inserida em um dado arquivo; segurança se refere ao grau de dificuldade em se descobrir a mensagem secreta embutida; robustez é entendida como sendo a capacidade de que mesmo que o arquivo estego seja modificado não haja distorção da mensagem secreta inserida nele.

A seguir são apresentados alguns métodos mais utilizados para esconder dados em arquivos de imagem, vídeo e, mais detalhadamente, áudio.

2.1.1 Esteganografia em arquivos de imagem

Imagens são os arquivos mais comumente utilizados para esconder informação, principalmente pelo seu largo uso e, normalmente, por possuírem menor tamanho em comparação a outros formatos, como vídeo e áudio. Dentro da categoria de imagens, existem diversos formatos com usos específicos e cada um destes estão relacionados com técnicas esteganográficas próprias (MORKEL; ELOFF; OLIVIER, 2005).

Numa imagem, são exibidas diferentes cores e intensidades de luz em diferentes regiões de um plano. Essas diferentes cores e intensidades de luz são representadas por números, cuja representação numérica forma pontos individuais conhecidos por pixels (NARAYANA; PRASAD, 2010) (MORKEL; ELOFF; OLIVIER, 2005). O número de bits em um esquema de cores (ou profundidade de bit) se refere à quantidade de bits utilizados

para representar cada pixel. Imagens em escala de cinza, por exemplo, podem ter cada pixel representado por uma palavra de 8 bits, possibilitando a reprodução de $2^8 = 256$ tons de cinza (MORKEL; ELOFF; OLIVIER, 2005). Empregando o modelo RGB, os pixels de uma imagem colorida são representados pela soma de 3 cores (vermelho, verde e azul) e cada uma destas cores é comumente representada por 8 bits, logo são imagens de 24 bits por pixel (OWENS, 2002) (CUNHA; LIMA; BEZERRA, 2010).

As técnicas de esteganografia usadas em imagens podem ser aplicadas no domínio espacial e no domínio da transformada. A distinção entre essas duas classes de esteganografia é feita a seguir.

2.1.1.1 Domínio espacial

A técnica mais utilizada no domínio espacial é a substituição do bit menos significativo (LSB, do inglês *least significant bit*) de um pixel por um bit pertencente à mensagem que se deseja ocultar. Este procedimento não provoca distorções drásticas na imagem e é imperceptível à visão humana (HUSSAIN; HUSSAIN, 2013). As principais vantagens da utilização da técnica de modificação do LSB no domínio espacial são:

- Pouca deformação da imagem original utilizada como arquivo disfarce.
- É possível armanezar uma quantidade maior de mensagem na imagem original.

Em contraponto, as desvantagens do uso desta técnica são:

- Pouca robustez, uma vez que a mensagem secreta pode ser perdida através de manipulações na imagem.
- A mensagem secreta pode ser facilmente destruída pelas mais simples formas de ataque.

Uma forma de tornar este método mais seguro é o compartilhamento de uma chave secreta, entre o emissor e receptor, que determine exatamente que pixels foram alterados (MORKEL; ELOFF; OLIVIER, 2005).

2.1.1.2 Domínio da transformada

O processo de esconder informação no domínio da transformada é mais efetivo que o utilizado no domínio espacial. A principal vantagem é que a informação é ocultada em regiões da imagem menos expostas a compressão, recorte e processamento. As transformadas comumente utilizadas nestes métodos são a transformada discreta do cosseno (DCT, do inglês *discrete cosine transform*), a transformada discreta de Fourier (DFT, do inglês

discrete Fourier transform) e a transformada discreta de wavelet (DWT, do inglês *discrete wavelet transform*).

Utilizando a DCT, por exemplo, a qual é empregada também no popular padrão de compressão JPEG (do inglês *Joint Photographics Experts Group*), cada um dos 64 coeficientes $F(u, v)$ da DCT de um bloco de 8×8 pixels de imagem $f(x, y)$ é dado por (PROVOS; HONEYMAN, 2003)

$$F(u, v) = \frac{1}{4}C(u)C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cdot \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right], \quad (2.1)$$

em que $C(x) = \frac{1}{\sqrt{2}}$, quando $x = 0$, e $C(x) = 0$, caso contrário. Em seguida, os coeficientes $F(u, v)$ são quantizados, produzindo coeficientes $F^Q(u, v)$ dados por

$$F^Q(u, v) = \left\lfloor \frac{F(u, v)}{Q(u, v)} \right\rfloor, \quad (2.2)$$

em que $Q(u, v)$ é uma tabela de quantização de 64 elementos (SMITH et al., 1997). Como os coeficientes F^Q são todos inteiros, torna-se mais simples representá-los em binário e considerar o bit menos significativo de cada coeficiente para inserção de uma mensagem esteganográfica. O processo de substituição dos referidos LSB por bits da mensagem que se deseja ocultar é análogo àquele implementado no domínio espacial, com a exceção de que, caso o coeficiente quantizado possua valor 0 ou 1, este não é utilizado.

2.1.2 Esteganografia em arquivos de vídeo

A utilização de vídeos como arquivos de disfarce pode ser mais interessante do que outros formatos de arquivos por conta dos seus tamanhos e exigências de memória. Os vídeos são compostos de milhares de imagens que são exibidas a taxas que variam de 24 a 30 quadros por segundo. Assim, considerando esta dinâmica, é possível explorar o fato da visão humana não ter a capacidade de notar diferenças pequenas nas imagens por conta da inserção dos bits da mensagem secreta (CARVALHO; GOULARTE, 2005). Em arquivos de vídeo, pode-se utilizar, por exemplo, a esteganografia LSB. A técnica funciona de modo análogo ao explicado anteriormente para imagens. O mesmo acontece com técnicas que atuam no domínio da transformada, que utilizam principalmente a transformada discreta do cosseno, a transformada discreta de Fourier e a transformada discreta de wavelet.

2.1.3 Esteganografia em arquivos de áudio

Esconder alguma informação em sinais de áudio é mais desafiador do que em sinais de vídeo e imagem, por conta do amplo alcance dinâmico do sistema auditivo humano (HAS, do inglês *human auditory system*) em relação ao sistema visual humano

(HVS, do inglês *human visual system*) (NOSRATI; KARIMI; HARIRI, 2012). De qualquer forma, esquemas esteganográficos utilizados em arquivos de áudio possuem estrutura geral similar à apresentada na Figura 1. Alguns dos métodos mais comuns para esteganografia em sinais de áudio são discutidos a seguir (JAYARAM; RANGANATHA; ANUPAMA, 2011) (BENDER et al., 1996).

2.1.3.1 Codificação do LSB

O mesmo método utilizado para a codificação do LSB apresentado para imagens e vídeos pode ser aplicado para sinais de áudio. Inicialmente, é preciso converter o sinal de áudio em *bit-stream*. Em seguida, a coluna com os bits menos significativos do áudio é modificada pela coluna com os bits da mensagem secreta, a fim de que o arquivo estego seja obtido. Conforme já se mencionou anteriormente, os dois bits menos significativos poderiam ser usados para esconder bits da mensagem secreta. Embora isso aumente a capacidade do arquivo de disfarce, permitindo a ocultação de mais bits da mensagem secreta, tais modificações poderiam comprometer a qualidade sonora do áudio, despertando, inclusive, a desconfiança de que o arquivo original teria passado por alguma espécie de manipulação.

2.1.3.2 Codificação de Paridade

Na codificação de paridade, o sinal de áudio é segmentado em áreas separadas de amostras e a mensagem secreta é escondida no bit de paridade de cada uma dessas áreas. Quando o bit de paridade de uma área não corresponder ao bit da mensagem secreta a ser escondida, o bit menos significativo de alguma amostra desta área será invertido, de forma a ser igual ao bit da mensagem secreta. Assim, o emissor terá um aumento nas possibilidades de inserção do bit da mensagem secreta, mantendo a mudança no sinal menos perceptível. A Figura 2 apresenta um procedimento de codificação de paridade (DUTTA; BHATTACHARYA; KIM, 2009).

2.1.3.3 Codificação de Fase

A codificação de fase baseia-se no fato de que as componentes de fase de um áudio não são perceptíveis ao ouvido humano, diferentemente do que acontece com os ruídos. Nesse sentido, a codificação de fase leva vantagem em relação às técnicas apresentadas até agora pelo fato de não inserir ruídos no sinal original. Ao invés disso, esta técnica realiza a modificação de fase no segmento de áudio inicial por uma fase de referência que representa a mensagem secreta, proporcionando uma codificação inaudível em termos da relação sinal-ruído.

Os segmentos restantes são reajustados afim de manter a fase relativa entre os segmentos. A codificação de fase é conhecida como sendo um dos métodos de esteganografia de áudio mais eficientes em termos da relação sinal-ruído. Se a relação de fase entre as

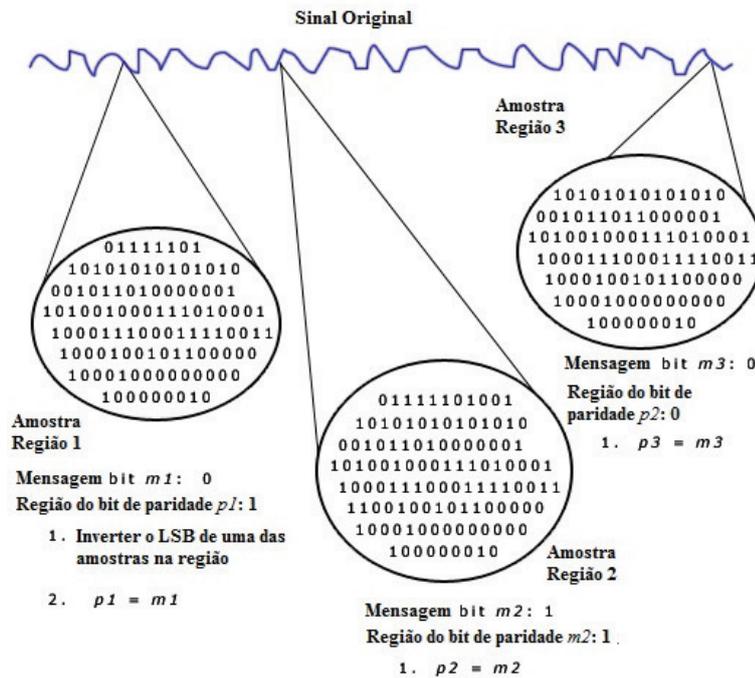


Figura 2 – Procedimento para codificação de paridade

componentes de frequência sofrerem mudanças drásticas, será ocasionada uma dispersão de fase bastante notável. Por outro lado, caso a alteração de fase seja pequena o suficiente, o processo esteganográfico pode ser realizado de maneira imperceptível.

Os procedimentos para a realização da codificação de fase são listados a seguir:

- Divide-se o sinal de áudio original, $A(i)$, ($0 \leq i \leq I - 1$), em N pequenos segmentos, $A_n(i)$, onde ($0 \leq n \leq N - 1$), em que os comprimentos dos segmentos devem ser iguais entre si e a quantidade destes segmentos deve ser igual ao tamanho da mensagem secreta, em bits, a ser escondida.
- Aplica-se uma transformada discreta de Fourier de K -pontos para os N segmentos, onde $K = \frac{I}{N}$, e cria-se uma matriz de fase, $\phi_n(\omega_k)$, e uma de magnitude, $A_n(\omega_k)$, para ($0 \leq k \leq K - 1$).
- Calcula-se a diferença de fase entre os segmentos adjacentes para ($0 \leq n \leq N - 1$):

$$\Delta\phi_{n+1}(\omega_k) = \phi_{n+1}(\omega_k) - \phi_n(\omega_k).$$

- O desvio de fase entre segmentos adjacentes é detectável de maneira fácil e rápida, assim é possível modificar qualquer fase absoluta do segmento, contanto que seja mantida a diferença relativa entre as fases adjacentes. Nesse caso, a mensagem secreta

será escondida somente no vetor de fase do primeiro segmento do sinal conforme

$$\text{nova fase} = \begin{cases} \frac{\pi}{2}, & \text{se o bit de mensagem} = 0, \\ -\frac{\pi}{2}, & \text{se o bit de mensagem} = 1. \end{cases} \quad (2.3)$$

5. Usando a nova fase do primeiro segmento, $\phi'_0(\omega_k)$, é criada uma matriz com a nova fase e a fase original é ajustada, para $n > 0$, utilizando as diferenças de fase:

$$\begin{bmatrix} \phi'_1(\omega_k) = \phi'_0(\omega_k) + \Delta\phi_1(\omega_k) \\ \dots \\ \phi'_n(\omega_k) = \phi'_{n-1}(\omega_k) + \Delta\phi_n(\omega_k) \\ \dots \\ \phi'_N(\omega_k) = \phi'_{N-1}(\omega_k) + \Delta\phi_N(\omega_k) \end{bmatrix}$$

6. O sinal de áudio pode ser reconstruído no final através da aplicação da DFT inversa utilizando a matriz de fase modificada, $\phi'_n(\omega_k)$, e a matriz com a magnitude original, $A_n(\omega_k)$ e, em seguida, unindo todos os segmentos do sinal de áudio.

Para a extração da mensagem contida no arquivo de áudio é necessário que o receptor conheça o comprimento do segmento e a quantidade de pontos da DFT utilizada.

2.1.3.4 Espalhamento espectral

O método de espalhamento espectral básico consiste em distribuir os bits da mensagem secreta ao longo de todo o espectro de frequência do sinal de áudio. Como consequência disso, o sinal resultante utilizará uma largura de banda superior à que é realmente necessária para transmissão destes dados. A vantagem de utilizar este método é que, em relação aos métodos apresentados anteriormente, ele apresenta uma taxa de transmissão de dados relativamente moderada e possui um alto nível de robustez contra técnicas de quebra de informação. Por outro lado, ele tem a desvantagem de produzir ruído no arquivo de áudio.

2.1.3.5 Ocultação de eco

Essa técnica insere dados secretos em um arquivo de áudio a partir da inserção de um eco neste sinal, oferecendo um aumento na taxa de transmissão de dados e elevando o nível de robustez em comparação a alguns dos métodos anteriores. A quantidade de bits da mensagem secreta a serem inseridos depende diretamente da quantidade de ecos produzidos no sinal de áudio original. Existem três parâmetros relacionados ao eco que devem ser analisados e eventualmente alterados: amplitude, *offset* e tempo de decaimento. Todos estes parâmetros devem ser mantidos abaixo do limiar da audição humana (20 Hz) para garantir a imperceptibilidade do eco. Os valores de *offset* são alterados de acordo com valores específicos que representam os bits 1 e 0.

2.2 Marca D'Água

Marca d'água digital é o processo de ocultação de uma informação (oculta ou não) em algum arquivo multimídia digital, como áudio, vídeo e imagem, por exemplo, e que pode ser extraída ou detectada por muitos métodos com os mais diversos objetivos (CHAN-DRAMOULI; MEMON; RABBANI, 2002). A Figura 3 apresenta um esquema básico do processo de inserção e do processo de extração da marca d'água. Diferentemente da ideia de esteganografia, em que tanto o método de inserção da mensagem quanto a mensagem em si devem ser secretos, o processo de inserção de marca d'água é conhecido e a mensagem não deve ser secreta, a menos que se utilize alguma chave secreta (PODILCHUK; DELP, 2001). Uma marca d'água pode ser visível ou invisível. Quando a marca é imperceptível, após a sua inserção, o arquivo de disfarce não sofre alterações significativas. Nesse caso, algoritmos de extração e detecção são necessários para descobrir a presença de informação secreta nos arquivos de disfarce e também se utilizam chaves secretas para elevar o nível de segurança. O processo de marca d'água possui uma noção adicional de robustez contra ataques e manipulações de dados usuais, como conversão digital-analógico e conversão de formato digital (HARTUNG; KUTTER, 1999).

Ainda sobre marcas d'água visíveis, estas podem ser frágeis, robustas ou semi-frágeis, enquanto as marcas d'água não visíveis podem ser apenas frágeis e robustas. Uma marca d'água frágil não suporta transformações com perda no sinal original e, assim, tem como propósito a detecção de alteração no arquivo original. Uma técnica para inserir uma marca d'água frágil mantendo a sua imperceptibilidade é alocando os dados da informação nas partes perceptualmente insignificantes dos sinais a serem marcados (PODILCHUK; DELP, 2001). A marca d'água semi-frágil possui uma importante função em autenticação de conteúdo para multimídia e é capaz de distinguir mudanças não maliciosas através de operações de processamento de sinais comuns de modificações maliciosas. Para isso é necessário que essa marca tenha um certo grau de robustez e fragilidade (ZHAO; SHEN, 2009). Como a robustez é uma das características mais importantes em marca d'água ela será abordada nos tópicos seguintes.

Os seguintes cenários de aplicação das marcas d'água digitais podem ser destacados:

- *Impressão digital e marcação*: Estes se relacionam com aplicações de proteção de direitos autorais quando as informações sobre o gerador e receptor do conteúdo são inseridas como marca d'água (HARTUNG; KUTTER, 1999). Se em um certo momento forem encontradas cópias não autorizadas do arquivo, a sua origem pode ser detectada pela recuperação da impressão digital. Para sua utilização a marca d'água deve ser invisível e deve ser invulnerável a tentativas de constituição, remoção ou invalidação.
- *Prevenção de cópia ou controle*: Um exemplo para esse tipo de aplicação é quando em

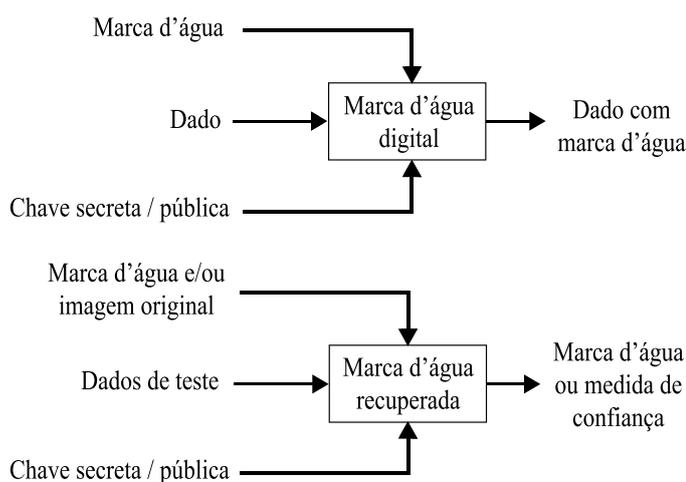


Figura 3 – Esquema básico de inserção de uma marca d'água e de sua extração

um sistema fechado, um *hardware* específico é necessário para copiar ou visualizar um conteúdo. Esse sistema utiliza uma marca d'água para contabilizar quantas cópias foram realizadas e, baseado numa pré-definição da quantidade de cópias permitidas, não realizará mais estas cópias, informando ao usuário que a quantidade máxima de cópias foi atingida.

- *Detecção de fraude e sabotagem*: Quando se trabalha com informação cujo conteúdo é algo de efeito legal, como aplicações em ciências médicas e transações comerciais, é importante garantir que o conteúdo transmitido tem origem confiável e livre de qualquer manipulação ou falsificação. Garantir essa segurança é possível inserindo uma marca d'água nesse conteúdo e, nesse caso, é importante a utilização de uma chave secreta, associada à fonte do conteúdo, que possibilite a correta extração da marca d'água. Daí verifica-se a integridade do conteúdo de interesse através da integridade da marca d'água extraída.

2.2.1 Propriedades da marca d'água digital

Os sistemas que utilizam a marca d'água como forma de inserção de informação apresentam várias propriedades cuja importância varia de acordo com o objetivo pretendido. Abaixo são listadas algumas destas propriedades (CHANDRAMOULI; MEMON; RABBANI, 2002), (KATZENBEISSER; PETITCOLAS, 2000) e (COX et al., 2007).

- *Eficácia da inserção*: A eficácia de um sistema de inserção de marca d'água é a probabilidade de que um arquivo de disfarce após ter sido submetido a esse processo de inserção seja exatamente um arquivo marcado, ou seja, a probabilidade de detecção positiva imediatamente após inserida a marca d'água.

- *Fidelidade*: Refere-se à semelhança percentual entre o arquivo original utilizado e o arquivo marcado.
- *Carga de dados*: Refere-se à quantidade de bits que são inseridos por marca d'água em uma unidade de tempo. Para arquivos de áudio, por exemplo, carga de dados se refere à quantidade de bits inseridos por segundo.
- *Taxa de falso positivo*: É a detecção de uma marca d'água em um arquivo de disfarce que não possui uma. É o número de falsos positivos estimado em um determinado número de varreduras realizadas por um detector.
- *Robustez, segurança e custo*: A primeira é a capacidade de detectar uma marca d'água mesmo após o arquivo marcado ter sido manipulado por operações de processamento de sinais. A *segurança* de uma marca d'água se refere à capacidade do método resistir a ataques hostis. Para elevar o grau de segurança de um processo de marca d'água, são utilizadas chaves secretas partindo dos conceitos conhecidos em criptografia. O *custo* de um processo de marca d'água se refere à velocidade em que inserção e detecção devem ser realizadas e o número de insersores e detectores necessários.
- *Imperceptibilidade*: O processo de inserção de dados não deve introduzir nenhum conteúdo perceptível no arquivo a ser marcado. Contudo, para que se tenha um alto desempenho em relação a robustez é necessário que a amplitude da marca d'água seja tão grande quanto possível. Sendo assim, os métodos de inserção de marca d'água apresentam uma linha bastante tênue entre robustez e imperceptibilidade. O ideal nesse caso seria poder inserir dados abaixo do limiar da percepção, os quais são de difícil definição quando se pensa em áudio, imagem e vídeo do mundo real (HARTUNG; KUTTER, 1999).

Como acontece na esteganografia, que pode ser realizada nos mais diversos arquivos digitais, a marca d'água também pode ser inserida em arquivos de texto, imagem, vídeo e áudio, conforme descrito a seguir.

2.2.2 Marca d'água em arquivos de imagem

A marca d'água digital de imagens se utiliza do fato de imagens digitais conterem uma redundância de dados que, além de ser usada nas técnicas de compressão de imagem, possibilita a sua utilização para a inserção de dados, nesse caso, a marca d'água (HARTUNG; KUTTER, 1999; KOSTOPOULOS et al., 2003).

Em (GUPTA; BARVE, 2014) são citados os dois tipos mais comuns de marcas d'água que podem ser inseridos em uma imagem. O primeiro é a *Sequência Gaussiana Pseudo-Aleatória*, que é formada pelos números 1 e -1, possuindo média zero e variância

unitária. Para a detecção dos dados originais são utilizadas medidas de correlação destas marcas d'água. O segundo emprega como marcas imagens binárias ou em escala de cinza.

Inicialmente, os esquemas de marca d'água foram introduzidos no domínio espacial (SINGH; SINGHAL; CHAUDHARY, 2013). A partição de cores pode ser usada para estas marcas d'água desde que a marca d'água esteja em apenas uma banda de cor. Esses processos envolvem a adição de um pseudo-ruído na imagem. Isso provoca uma modificação no bit menos significativo do conteúdo original, o qual é visualmente irrelevante.

A marca d'água no domínio da transformada é utilizada por trazer vantagens no processo de inserção e para o desenvolvimento de técnicas que sejam mais robustas contra vários ataques geométricos, quando comparadas às técnicas aplicadas no domínio espacial. As transformadas normalmente utilizadas são a DFT, a DCT e a DWT. O algoritmo baseado na DCT, por exemplo, é robusto contra operações de processamento de imagem como filtragem passa-baixas, ajustes de contraste e brilho e etc.

Boa parte das técnicas empregadas para marca d'água de imagens pode ser empregada para marca d'água de vídeos digitais. De qualquer forma, particularidades que só são observadas em vídeos, como, por exemplo, o grande volume de dados de redundância entre os frames, o desequilíbrio entre as regiões móveis e imóveis, requisitos de tempo real em um *broadcasting* de vídeo e a possibilidade de certos tipos de ataques precisam ser levadas em conta.

2.2.3 Marca d'água em arquivos de áudio

A quantidade de trabalhos que abordam técnicas de marca d'água em arquivos de áudio ainda é inferior em relação as que abordam marca d'água em vídeo e imagem. O processo de inserção de marca d'água em sinais de áudio é mais desafiador do que para sequências de imagem e vídeo devido a sensibilidade do sistema auditivo humano (SAH) em comparação à do sistema visual humano (SVH). O SAH é sensível num alcance de potência maior do que $10^9:1$ e um alcance de frequência maior que $10^3:1$; além disso, é bastante sensível ao ruído aditivo gaussiano branco; esse ruído pode ser detectado tão baixo quanto 70 dB abaixo do nível do ambiente em um arquivo de áudio. Por outro lado, o SAH tem um pequeno alcance diferencial, ou seja, um som mais forte tende a cobrir um som mais fraco (CHAUHAN; RIZVI, 2013), (ALSALAMI; AL-AKAIDI, 2003).

2.2.3.1 Requisitos para marca d'água em áudio

Para o desenvolvimento de sistemas de marca d'água eficientes, é necessário que alguns requisitos, como os apresentados a seguir, sejam levados em conta (CHAUHAN; RIZVI, 2013).

- *Transparência perceptual*: Diz respeito ao fato de se desejar que a informação que

contém a marca d'água seja imperceptível e não cause nenhuma deterioração (ou degradação) do sinal original, ou seja, não deve ser ouvido pelo ouvido humano.

- *Robustez*: Representa a habilidade que a marca d'água tem ao ponto de se manter intacta mesmo após sofridos diferentes tipos de ataques (conversão D/A e A/D, filtragem linear e não linear, compressão e transformação geométrica do arquivo que hospeda a marca d'água).
- *Segurança*: Depende do uso ou não de chave secreta. A marca d'água tem de ser fortemente resistente a detecção não autorizada e também agentes indesejáveis que visam piratear informação.
- *Taxa de dados*: Refere-se ao número de bits de marca d'água que devem ser inseridos num sinal de áudio de forma confiável por unidade de tempo.
- *Verificação e confiabilidade*: Capacidade da marca d'água fornecer informações completas e confiáveis para comprovar as propriedades de produtos protegidos por direitos autorais.

2.2.3.2 Aplicações para a marca d'água em sinais de áudio

Existem várias aplicações para a marca d'água em sinais de áudio. A seguir são apresentadas algumas dessas aplicações.

- *Proteção de copyright*: A marca d'água inserida carrega uma prova de propriedade, devendo ser robusta e segura, para garantir ao proprietário do áudio a demonstração de autenticidade do arquivo.
- *Detecção de alteração*: O principal objetivo aqui é detectar a modificação no arquivo original. Nestas técnicas normalmente se utilizam informações secundárias para inserir uma marca d'água no sinal original para posteriormente verificar se houve alguma modificação no arquivo. Para um bom resultado pode ser usada marca d'água frágil, que oferece menos robustez contra algumas alterações.
- *Proteção de cópia*: Representa um controle de cópia e um controle na política de privacidade. Usualmente, é empregada para evitar cópias ilegais e não autorizadas de CDs e DVDs, por exemplo. As políticas de controle de acesso e de controle de cópia detectam a marca d'água e impõem operações particulares que inibem a continuidade destas cópias.
- *Impressão digital*: A impressão digital permite ao proprietário do conteúdo rastrear a origem da ilegalidade através da inserção de uma única marca d'água nas cópias do sinal que são fornecidas tanto para clientes quanto para distribuidores.

2.2.3.3 Técnicas de Marca D'água em Áudio

Nesta seção serão apresentadas algumas das técnicas mais populares para inserção de marca d'água em sinais de áudio (CHAUHAN; RIZVI, 2013):

- *Técnicas de marca d'água no domínio do tempo*

As técnicas de marca d'água no domínio do tempo utilizam o áudio em sua forma original para realizar a inserção da marca d'água. Apesar de possuir uma implementação simples e computacionalmente não custosa, essas técnicas são fracas em robustez contra simples filtragens passa-baixa, por exemplo (CHAUHAN; RIZVI, 2013):

Uma marca d'água pode ser inserido no domínio do tempo empregando a já discutida modificação do LSB. A robustez deste método depende da quantidade de bits modificados no áudio original. O método possui vantagens e desvantagens similares às que já foram descritas anteriormente, neste trabalho, para outros cenários e tipos de mídia.

Outra possibilidade é inserir uma marca d'água através da adição de eco nos sinais de áudio, o que provoca um “enriquecimento” na qualidade do som, pois um eco provoca uma ressonância no áudio. Após a adição do eco, o sinal de áudio continua com as mesmas características perceptuais e estatísticas. Os três parâmetros do eco utilizados para a inserção da marca d'água são: amplitude inicial, taxa de decaimento e *offset*. Com a diminuição do *offset*, o sinal original e o eco se misturam, fazendo com que o sistema auditivo humano não seja capaz de distingui-los. Este fato torna imperceptível a marca inserida por meio desta técnica. Uma desvantagem deste método é a sua alta complexidade devido à necessidade de calcular parâmetros cepstrais.

- *Técnicas de marca d'água no domínio da frequência*

As técnicas no domínio da frequência se baseiam predominantemente em operações de quantização. O método de inserção de marca d'água baseado na Modulação do Índice de Quantização (QIM) é um dos melhores por ser muito simples e possuir uma boa robustez em relação a taxa de distorção. O sinal de áudio original é quantizado utilizando uma ou duas sequências de níveis de quantização; a escolha do quantizador a ser utilizado depende de cada unidade de informação da marca que se deseja inserir.

Também se pode inserir uma marca d'água de áudio baseada em quantização média. A abordagem é simples e menos susceptível à ocorrência de erros, quando comparada aos métodos baseados em quantização usuais. Pode-se mencionar, ainda, a Marca d'água baseada em quantização de modulação dinâmica, que apresenta bons resultados em relação a robustez, carga de dados e imperceptibilidade. Quando combinada com outras técnicas obtém-se uma melhoria no desempenho do processo de marca d'água (CHAUHAN; RIZVI,

2013), (GOENKA; PATIL, 2005). Também é possível inserir e extrair marcas d'água em sinais de áudio empregando quantização vetorial.

- *Marca d'água de áudio no domínio comprimido*

O padrão de compressão MPEG estabelece técnicas de codificação para conteúdos audiovisuais devido a sua capacidade de compressão (ALSALAMI; AL-AKAIDI, 2003). Uma vantagem em sua utilização combina o fato de que no domínio comprimido (NEUBAUER; HERRE, 2000) a marca d'água é inserida diretamente no *stream* de bits do áudio MPEG ao invés do domínio não comprimido, além de garantir que não é preciso alteração na sintaxe do bit stream original. Isso se dá pelo fato do algoritmo de compressão de áudio MPEG possuir perdas e utilizar a natureza especial do sistema auditivo humano (SAH) para a sua realização, pois remove as partes perceptualmente irrelevantes do áudio, tornando a distorção do sinal de áudio inaudível ao ouvido humano. Esse tipo de sistema é bastante interessante para o cenário de “áudio pago” em que o proprietário armaneza conteúdos de áudio em formato comprimido (NEUBAUER; HERRE, 1998).

- *Marca d'água de áudio no domínio combinado*

A combinação de abordagens multidimensionais define a marca d'água no domínio combinado. Inclui-se, nesse caso, o uso de modelos psicoacústicos e propagação do espectro (LACY et al., 1998), mascaramento psicoacústico e método de bit stream (BONEY; TEWFIK; HAMDY, 1996), etc. Quando se combinam as técnicas citadas, tem-se mais garantia de que a marca d'água será inaudível para qualquer taxa de bits das tecnologias de compressão de áudio conhecidas atualmente. Possui uma baixa complexidade computacional, o que é positivo para sua implementação e não degrada a qualidade do áudio que recebe a marca d'água.

A técnica proposta nesta dissertação não contempla de forma rigorosa conceitos esteganográficos ou de marca d'água, pois o objetivo aqui é embutir informação que não é secreta, neste caso as partituras musicais. De modo mais específico, conforme se apresentará no Capítulo 4 desta dissertação, a técnica proposta consiste na adaptação de um método de modulação de fase para inserção de bits que compõem a partitura, quando esta, escrita utilizando notação ABC, for convertida para números binários.

3 Áudio Digital: Representação e Padrão MP3

ANTES de se iniciar uma apresentação dos formatos digitais aos quais arquivos de áudio são submetidos, faz-se necessário uma rápida análise da generalidade que existe por trás da digitalização de sinais, ou seja, um estudo sucinto sobre amostragem de sinais de tempo contínuo e/ou dos próprios sinais de tempo discreto, que podem ser submetidos, por exemplo, à *subamostragem*.

3.1 Amostragem

Em (HSU, 2008) define-se um sinal como uma função que representa alguma quantidade ou variável física que contenha informações sobre o seu comportamento ou natureza do fenômeno e apresenta-se uma série de classificações para os diversos tipos de sinais. Os fenômenos observados na natureza são normalmente analógicos (sinais de tempo contínuo). Contudo, neste trabalho se restringem as discussões sobre os sinais digitais. Para o processamento digital de sinais é necessário utilizar uma técnica conhecida como *amostragem* para representar um sinal de tempo contínuo por meio de um sinal de tempo discreto. Essa amostragem é realizada para facilitar a manipulação desse sinal por computadores ou microprocessadores (HAYKIN; VEEN, 2001), os quais experimentaram, nas últimas décadas, um desenvolvimento bastante expressivo de tecnologias. Um sinal contínuo, sob certas condições, pode ser completamente representado por amostras espaçadas uniformemente no tempo (OPPENHEIM; WILLSKY, 2010).

O conceito de amostragem pode ser ilustrado, por exemplo, considerando imagens impressas que são formadas por uma grade de pontos, em que cada ponto representa uma amostra da imagem contínua que se deseja representar. Essas amostras, quando agrupadas de forma muito próxima, transmitem a impressão de que a imagem é espacialmente contínua. Noutro caso, o som pode ser representado em função do tempo, contudo tanto a amplitude do som como o eixo temporal são originalmente contínuos. Representar esse som em formato digital requer amostragem (*conversão* do eixo temporal para um eixo de tempo discreto) e, posteriormente, quantização (*mapeamento* dos valores de amplitude contínua em valores de amplitude discreta) (BOSI; GOLDBERG, 2003).

Logo, o conceito de amostragem se destaca pelo fato de permitir empregar as técnicas desenvolvidas para sinais de tempo discreto como forma de manipulação dos sinais originalmente de tempo contínuo. Na Figura 4, é apresentado um diagrama que

contém, além de um sistema discreto, blocos associados aos processos de conversão A/D (analógico/digital) e conversão D/A (digital/analógico), os quais se relacionam com a amostragem. O processo de gravação e reprodução de um ou vários instrumentos em um estúdio musical ilustra bem esses processos. A gravação acontece através de ferramentas analógicas, por conta da natureza analógica dos instrumentos, em seguida, ocorre a digitalização desses sinais utilizando ferramenta específicas de áudio. A partir daí realiza-se a manipulação através de *softwares* específicos e a finaliza-se registrando o áudio em mídia digital. O processo de reprodução converte os sinais de tempo discreto para o tempo contínuo e através de alto-falantes reproduz-se o áudio originalmente gravado. A seguir serão apresentadas de maneira sucinta e objetiva alguns conceitos de amostragem de sinais.

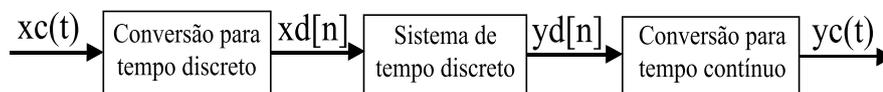


Figura 4 – Processamento de tempo discreto de sinal de tempo contínuo

3.1.1 Representação de sinais de tempo contínuo a partir de suas amostras

O *Teorema da Amostragem de Nyquist-Shannon*, ou simplesmente *Teorema da Amostragem*, afirma que um sinal $x(t)$ limitado em banda com $X(j\omega) = 0$ para $|\omega| > \omega_M$, é determinado apenas por suas amostras $x(nT)$; $n = 0, \pm 1, \pm 2, \dots$, se

$$\omega_s \geq 2 \cdot \omega_M, \quad (3.1)$$

em que

$$\omega_s = \frac{2 \cdot \pi}{T} \quad (3.2)$$

em que ω_M é a largura de banda do sinal $x(t)$ e ω_s é a frequência de amostragem. A partir dessas amostras é possível reconstruir $x(t)$ gerando um trem de impulsos periódicos, processando-o através de um filtro passa-baixas com ganho T e frequência de corte maior que ω_M e menor que $\omega_s - \omega_M$; se uma amostragem tiver sido realizada a uma frequência menor que a frequência de Nyquist, $|\omega| < \omega_M$, ocorre um fenômeno conhecido como *aliasing*, ou *subamostragem*, em que o sinal $x(t)$ não é reconstruído perfeitamente.

Entre as técnicas mais conhecidas de amostragem de sinais, merece destaque a que emprega um trem de impulsos periódicos, multiplicando-o pelo sinal de tempo contínuo $x(t)$ que se deseja amostrar, conforme ilustrado na Figura 5. Multiplicando um sinal $x(t)$ por um impulso unitário, retira-se como amostra do sinal exatamente o seu valor no instante em que o impulso está localizado, isto é, $x(t) \cdot \delta(t - t_0) = x(t_0) \cdot \delta(t - t_0)$ (OPPENHEIM; WILLSKY, 2010).

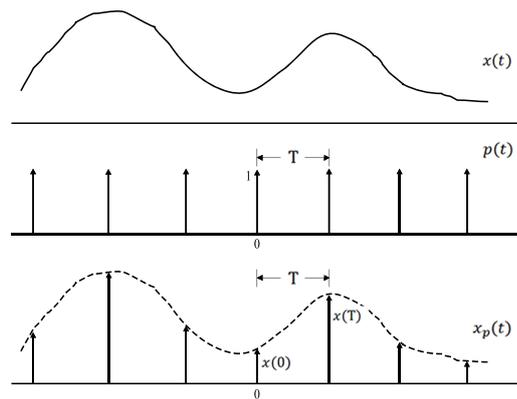


Figura 5 – Amostragem com trem de impulsos

Na prática, o método de amostragem de um sinal utilizando trem de impulsos é relativamente difícil de ser realizado, pela dificuldade que existe em gerar um pulso estreito e de amplitude elevada. O que mais convém, diante disso, é gerar o sinal amostrado na forma conhecida como retentor de ordem zero, ou *zero-order hold*. Esse sistema amostra o sinal $x(t)$ em um instante e mantém o seu valor até um próximo instante de recolhimento de outra amostra, como ilustrado na Figura 6 (OPPENHEIM; WILLSKY, 2010).

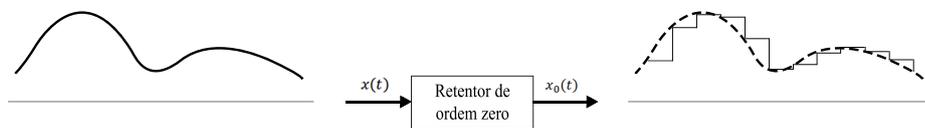


Figura 6 – Amostragem com retentor de zero

3.1.2 Reconstrução de sinais contínuos a partir de suas amostras

A reconstrução de um sinal contínuo a partir de suas amostras é comumente realizada empregando interpolação. Nesse contexto, pode-se usar, por exemplo, a *interpolação ideal* e a *interpolação linear*, em que pontos de amostra adjacentes são ligados através de uma reta ou, em alguns casos, por polinômios de ordem mais alta. Mais especificamente, a reconstrução de um sinal $x(t)$, a partir das amostras $x(nT)$ e do filtro passa-baixas ideal $H(j\omega)$, pode ser feita por

$$x_r(t) = \sum_{n=-\infty}^{+\infty} x(nT) \frac{\omega_c T}{\pi} \frac{\text{sen}(\omega_c(t - nT))}{\omega_c(t - nT)} \quad (3.3)$$

em que ω_c é a frequência de corte.

A interpolação apresentada na Equação 3.3, que utiliza a resposta ao impulso de um filtro passa-baixas ideal, é normalmente chamada de *interpolação de banda limitada*,

por implementar a reconstrução perfeita de $x(t)$ se esse sinal for limitado e se as condições do Teorema da Amostragem tiverem sido satisfeitas.

Todos os pontos discutidos até agora são procedimentos genéricos para a amostragem de qualquer formato de sinal, seja vídeo, imagem ou áudio, por exemplo. A partir deste ponto, são destacadas apenas questões referentes aos sinais de áudio, que constituem o principal objeto desta dissertação.

3.2 Quantização de Sinais de áudio

Como citado anteriormente, para representar um sinal em formato digital é necessário convertê-lo de sinal contínuo para sinal discreto para que possa ser armazenado em um computador, por exemplo. Para realizar essa conversão é necessário, além de amostrar o sinal, quantizá-lo, o que corresponde a aproximar os valores de suas amostras por valores discretos pertencentes a um conjunto finito e previamente definido. A quantização provoca uma distorção na qualidade do áudio original, influenciando assim o processo de codificação deste áudio.

A quantização pode ser escalar e ou vetorial. Na quantização escalar, cada amostra é quantizada e codificada isoladamente; na quantização vetorial, um grupo de amostras é quantizado e associado a um único código. Além disso, os quantizadores podem ser classificados em quantizadores uniformes e não uniformes. Os quantizadores uniformes utilizam níveis de quantização equiespaçados e são bastante utilizados em sinais que não possuem uma faixa dinâmica alta. Esses quantizadores são divididos em *meio piso*, que são capazes de produzir uma nível de saída de valor zero e possuem um número par de saídas, e *meio degrau* que, ao contrário, não produzem saída de valor zero e possuem um número ímpar de saídas; para N bits o quantizador *meio piso* permite $2^N - 1$ códigos diferentes contra os 2^N códigos permitidos no quantizador *meio degrau*. Como o quantizador *meio piso* apresenta maior comprimento do range de entrada, ele é o quantizador uniforme mais utilizado.

Os quantizadores não-uniformes, em que as diferenças entre dois níveis de quantização subsequentes variam com a amplitude do sinal de entrada, são mais utilizados para sinais que apresentam um range dinâmico alto. Por essa razão, a construção de quantizadores não-uniformes requer o conhecimento de valores adequados de quantização para cada região do sinal, ou seja, quantizadores diferentes para entradas diferentes; isso implica em um grande custo de implementação. Uma solução para este caso é a compressão do sinal de entrada e, em seguida, a aplicação de um quantizador uniforme neste sinal, que produz o mesmo efeito de um quantizador não-uniforme.

3.3 Codificação de áudio

No início dos anos 80, com o surgimento do CD (*compact disc*), teve-se, pela primeira vez, a comercialização de mídia com formato digital. A partir daí, esse formato iniciou um processo de substituição quase que completo em relação ao formato analógico, até então comercializado através dos LPs (*Long-Play*) de vinil e as fitas cassetes. O aumento da capacidade de armazenamento e processamento dos computadores, em conjunto com o desenvolvimento de técnicas de manipulação de sinais discretos, também proporcionou maior interesse no desenvolvimento de formatos digitais. A isso, juntaram-se os avanços das técnicas de codificação de áudio (ou compressão), pois a digitalização dos sinais de áudio produzia uma grande quantidade de dados, dificultando a sua transmissão e o seu armazenamento. Inicialmente, foram desenvolvidas técnicas sem perdas, como, por exemplo, o código de Huffman e o código de Lempel-Ziv, que reduziam pela metade o tamanho do arquivo, viabilizando o armazenamento em CDs. Em seguida, técnicas de compressão que reduziam em até 90% o tamanho do arquivo sem perda significativa de qualidade foram desenvolvidas (MORKEN, 2007; LIU; CHANG, 2001).

As técnicas de representação de áudio digital podem ser divididas em dois grupos: baseadas em amostragem e baseadas em processamento digital de sinal (OLIVEIRA, 2006). Tal subdivisão é discutida sucintamente a seguir.

3.3.1 Representação de Áudio Baseada em Amostragem

Existe um número elevado de técnicas de codificação por amostragem, as quais não são específicas para certos tipos de áudio e, assim, podem ser utilizadas para voz e música, por exemplo. A seguir, essas técnicas são apresentadas.

3.3.1.1 Pulse Code Modulation - PCM

É uma técnica desenvolvida para codificação de voz para linhas telefônicas, as quais têm, normalmente, largura de banda limitada entre 300 Hz e 3,4 kHz. Embora uma frequência de amostragem de 6,8 kHz fosse suficiente para reconstrução do sinal, a fim de tornar prática a implementação dos filtros empregados, utiliza-se uma frequência de 8 kHz. Cada valor de amostra é normalmente representado por 8 bits (PCM-30).

3.3.1.2 Compact Disc Digital Audio - CD-DA

Como a música possui uma faixa de frequência de 20 Hz a 20 kHz, para satisfazer a condição do teorema da amostragem, uma frequência de amostragem de 40 kHz seria suficiente. Porém, na prática, utiliza-se uma taxa de 44,1 kHz a fim de acomodar discrepância de filtro, ou seja, eventuais frequências harmônicas de ordem superior à 40 Hz que venham a ocorrer. Cada amostra é quantizada de forma linear e representada por 16 bits.

Em áudio estéreo, são 16 bits por canal. A taxa produzida é de 705,6 kbps para áudio mono e de 1411 kbps para áudio estéreo.

3.3.1.3 Differential Pulse Code Modulation - DPCM

Essa técnica de codificação se baseia na diferença entre os valores de duas amostras consecutivas de uma forma de onda de um sinal de áudio; como o sinal é fortemente correlacionado, tal diferença tende a ser muito pequena, o que permite que um amostra possa ser utilizada na predição da amostra seguinte. Com isso, torna-se necessário apenas codificar a já mencionada diferença, ao invés de codificar cada amostra individualmente.

3.3.2 Representação de Áudio Baseada em Processamento Digital de Sinais

Nesta seção, são apresentadas algumas das principais técnicas para representação de áudio baseadas em processamento digital de sinais. Dentre as diversas características das técnicas abordadas, enfatiza-se, para cada uma delas, se a codificação realizada envolve ou não perdas no sinal ([MORKEN, 2007](#)).

3.3.2.1 AIFF - Audio Interchange File Format

O formato AIFF foi desenvolvido e publicado pela Apple em 1988 e suporta diferentes taxas de amostragem e comprimento de bits por amostra. Comumente, é utilizado para armazenamento de áudio em CD com taxa de amostragem de 44,1 kHz e 16 bits por amostra.

3.3.2.2 WAV - Waveform Audio Data

O formato WAV foi desenvolvido pela Microsoft e pela IBM. Assim como o formato AIFF, também suporta variação na taxa de amostragem, contudo o mais utilizado é o padrão para qualidade de CD de áudio. Esse formato usa inteiros de 32 bits para determinar o tamanho do arquivo em seu início, o que significa que arquivos no formato WAV não podem ter tamanho maior do que 4 GB. Para solucionar esse problema a Microsoft desenvolveu o formato W64.

Os formatos de compressão com perdas usam versões modificadas da DCT para sucessivos grupos de valores de amostra, analisa esses resultados e os perturbam a partir de modelos psicoacústicos. Os valores perturbados são convertidos para formatos de números adequados e codificados com algum código sem perdas, como o código de Huffman. Quando se deseja tocar o áudio o processo todo deve ser revertido e os dados trazidos de volta para valores de amostras perturbadas em taxa de amostragem apropriada.

3.3.2.3 MPEG-1, MPEG-2 E MPEG-4

O MPEG (do inglês *Moving Pictures Experts Group*), como parte da Organização Internacional para Padronização (ISO, do inglês *International Organization Standardization*), desenvolve vários padrões de codificação de áudio com objetivo de armazenamento e transmissão de dados digitais (LIU; CHANG, 2001). Os primeiros padrões de compressão de áudio digital de alta qualidade são o MPEG-1 e o MPEG-2 (NOLL, 1997), mas já foi desenvolvido, também, o MPEG-4. O MPEG-1 é o nome da primeira fase do trabalho da MPEG dividido em Layer-I, Layer-II e Layer-III (BRANDENBURG; POPP, 2000) (o popularmente conhecido MP3) e opera em um canal (mono) ou em dois canais (estéreo) com frequências de amostragem de 32, 44,1 e 48 kHz (LIU; CHANG, 2001). O desenvolvimento do MPEG-2, o qual se divide em MPEG-2 BC e MPEG-2 AAC, teve foco no suporte a multicanais e também na diminuição das taxas de amostragem (16, 22,05 e 24 kHz) e de bits (reduzida para 64 kbps) (NOLL, 1997).

3.3.2.4 AAC - Advanced Audio Coding

Esse formato, desenvolvido pela MPEG e classificado como MPEG-2, tem sido indicado como o sucessor do formato MP3 por um dos principais desenvolvedores do MP3, a Fraunhofer Society. O formato AAC pode ter melhor qualidade sonora em comparação ao MP3, principalmente para taxas de bits abaixo dos 192 kbps. O AAC suporta diferentes taxas de amostragem (variando dos 8 kHz aos 96 kHz) e até 48 canais. Assim como o MP3, o AAC utiliza a transformada discreta do cosseno modificada (MDCT, do inglês *modified discrete cosine transform*), mas realiza processamento de 1024 amostras no tempo e utiliza o Código de Huffman para codificação eficiente dos valores da MDCT.

3.3.2.5 Ogg Vorbis

O Vorbis é um formato de código-livre e possui muitas semelhanças com os formatos MP3 e AAC: utiliza MDCT para transformação dos grupos de amostra para o domínio da frequência, aplica um modelo psicoacústico e, por fim, codifica os dados finais usando Código de Huffman. Contudo, diferentemente dos formatos citados, ele sempre usa taxas de bits variáveis. São utilizados inteiros entre -1 e 10 para indicar a qualidade do áudio e suportadas taxas de amostragens que variam de 8 kHz a 192 kHz, com até 255 canais.

3.3.2.6 WMA - Windows Audio Media

O formato WMA foi desenvolvido pela Microsoft e também se baseia na MDCT e na codificação de Huffman. Esse formato permite taxa de amostragem de até 48 kHz e dois canais. Existe uma versão avançada desse formato, WMA Professional, que suporta 96 kHz de taxa de amostragem e 24 bits por amostra. A versão sem perdas, *WMA Lossless*, em baixas taxas de bits apresenta melhores resultados do que o MP3.

3.4 O Padrão de Compressão MP3

O formato de compressão MP3, nome popular para MPEG-1 Layer-3, é talvez o formato de áudio mais conhecido e difundido entre usuários interessados em realizar *download* ou armazenar arquivos de áudio altamente comprimidos e com boa qualidade sonora (BRANDENBURG; POPP, 2000; MORKEN, 2007). Esse formato foi desenvolvido pela Philips, CCETT, IRT e Fraunhofer Society, tornando-se um padrão internacional em 1991. Em 1995, o padrão MPEG-1 Layer-3 teve seu primeiro momento de destaque para o cenário comercial, quando foi utilizado para um sistema de *broadcasting* de satélite digital desenvolvido pela World-Space. O seu segundo momento, e talvez crucial para o seu sucesso, foi devido ao uso da Internet para a distribuição eletrônica de música; em 1999, o formato MP3 foi o termo de busca mais procurado na internet, em 1999. A seguir são apresentadas algumas características básicas para o entendimento do MP3.

3.4.1 Modo de operação

Os padrões MPEG-1 trabalham da mesma forma para sinais mono e estéreo. Existe uma técnica conhecida por *Codificação Joint Stereo* que pode ser usada para realizar uma codificação combinada mais eficiente para os canais esquerdo e direito em um sinal estéreo. O MP3 permite tanto a codificação estéreo *mid / side* quanto a codificação estéreo em intensidade. Essa última é bastante interessante para baixas taxas de bit, mas carrega o risco de alterar a imagem do som, que pode ser pensada como o desenho espacial dos instrumentos ou o palco sonoro criado pela disposição deles no ambiente de gravação, por exemplo). Os modos de operação são: canal único, canal duplo (dois canais independentes que contém, por exemplo, versões de linguagem diferentes do áudio), estéreo e *joint stereo* (BRANDENBURG, 1999).

3.4.2 Frequência de Amostragem

Os padrões de compressão MPEG trabalham com várias frequências de amostragem diferentes. No caso da MPEG-1, é possível empregar valores como 32 kHz, 44,1 kHz e 48 kHz.

3.4.3 *Bitrate* (taxa de bits)

Assim como na frequência de amostragem, o padrão MPEG também permite uma variedade de valores no *bitrate*. A seleção da taxa de bits do áudio comprimido pode ser escolhida por quem faz uso do arquivo de áudio. Especificamente, o padrão MP3 permite *bitrates* que variam de 8 kbps a 320 kbps. Além disso, o decodificador MP3 deve suportar o chaveamento da taxa de bits na transição entre dois frames de áudio.

3.4.4 Normativo *versus* Informativo

Uma das principais propriedades do padrão MPEG, citado em (BRANDENBURG; POPP, 2000), é o princípio de minimizar a quantidade de elementos normativos no padrão. Para o caso de áudio, o MPEG considera apenas a representação de dados e o decodificador como normativos.

O decodificador não é especificado a partir de uma forma exata de bits, mas através de fórmulas dadas para a maioria das partes do algoritmo e que definem a conformidade pelo desvio máximo do sinal codificado a partir de uma referência codificadora implementando fórmulas com dupla precisão aritmética. Esse fator permite aos decodificadores trabalharem tanto para arquiteturas de ponto flutuante quanto para arquiteturas de ponto fixo. O codificador de áudio MPEG é totalmente deixado para o implementador do padrão. Alguns exemplos de codificadores são descritos em ISO/IEC IS 11172-3.

3.4.5 Descrição do Algoritmo

A Figura 7 traz um diagrama de blocos que representa um típico codificador de MPEG-1 Layer-3, MP3 (BRANDENBURG, 1999; BOSI; GOLDBERG, 2003). São utilizados dois bancos de filtros em cascata: o primeiro é um banco de filtros polifásicos, seguido por uma MDCT adicional, caracterizando-o na classe de banco de filtros híbrido. Realiza-se a subdivisão de cada banda de frequência polifásica em 18 sub-bandas de ajuste mais fino, melhorando assim o desempenho na codificação de sinais tonais e, adicionalmente, torna controlável o sinal de erro permitindo um monitoramento mais fino do limiar de mascaramento. A saída da etapa filtro híbrido é quantizada não-uniformemente por um quantizador *meio piso*, cujo *step* é determinado pela saída do modelo psicoacústico. A saída da etapa de quantização é codificada por um código de Huffman e a saída é formatada em bitstream. A saída do bitstream é o dado de áudio codificado.

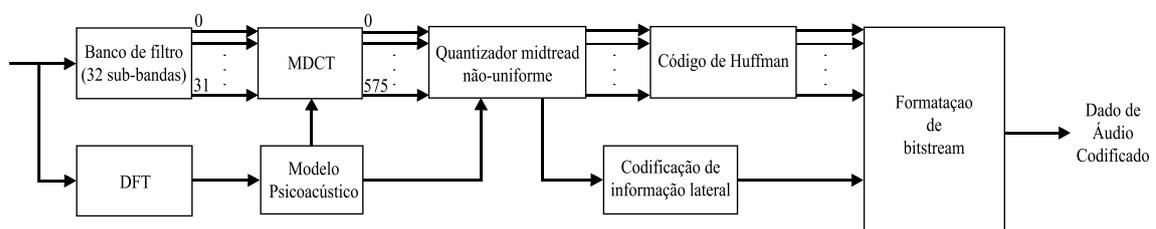


Figura 7 – Diagrama de bloco de um típico codificador MPEG-1 Layer-3

O modelo perceptual é quem determina a qualidade da implementação de um codificador. Em sua saída, são obtidos os valores para o limiar de mascaramento ou ruído permitido em cada partição do codificador. Para o padrão MP3, essas partições assemelham-se, de forma grosseira, às bandas críticas da capacidade auditiva do ser humano. Sendo possível manter-se a quantização do ruído abaixo do limiar de mascaramento para

cada partição do codificador, o resultado da compressão será imperceptível em relação ao áudio original.

Um sistema de dois *loops* de iteração aninhados é a solução mais comum para a quantização e a codificação em codificadores MP3. A quantização é realizada por um quantizador *power-law*, em que muitos valores são codificados automaticamente, com pouca precisão, e alguma modelagem de ruído é feita durante o processo de quantização.

Esses valores quantizados são codificados a partir da codificação de Huffman, que normalmente trabalha em pares ou quádruplos, no caso da codificação de números muito pequenos. Ainda, para melhor adaptação das estatísticas do sinal, utilizam-se diferentes tabelas de código de Huffman em diferentes partes do espectro. Antes da quantização, devem ser aplicados fatores de escala e um valor de ganho global a fim de manter o ruído de modelagem abaixo do limiar de mascaramento. Dois modelos de iteração são utilizados para encontrar o ganho ótimo e os fatores de escalas: o laço de iteração interno e o laço de iteração externo. Ambos são mais detalhados em ([BRANDENBURG, 1999](#)).

4 Ocultação de Partituras em Sinais de Áudio Digital

Neste capítulo, é descrita em detalhes a metodologia proposta neste trabalho. Inicialmente, é apresentada a notação ABC e, em seguida, um método utilizado para a representação em bits da partitura escrita em notação ABC é proposto. Posteriormente, descreve-se o processo de inserção dos bits produzidos com a escrita da notação ABC no sinal de áudio, o que constitui o objetivo principal do trabalho. Neste cenário, são discutidos aspectos específicos sobre a utilização do método em áudio mono e a sua adaptação para áudio estéreo. Por fim, é apresentado o procedimento para extração dos bits bem como os passos necessários à leitura da informação extraída, a qual se refere à partitura propriamente dita, em linguagem totalmente musical.

4.1 A Notação ABC para Escrita de Partituras Musicais

A notação ABC é uma notação textual que permite escrever partituras musicais sem o emprego de *softwares* específicos e que geralmente contém uma interface gráfica elaborada (CHAMBERS, 2009). Na notação ABC, caracteres ASCII são usados para representar notas e outros símbolos, e uma estrutura puramente textual é gerada. Embora a utilização de ferramentas gráficas para a escrita musical possa ser considerada mais intuitiva e de entendimento mais fácil por parte de músicos iniciantes, a utilização de uma notação textual permite rapidez e praticidade, uma vez que qualquer aplicativo de bloco de notas é suficiente para iniciar a escrita.

De qualquer modo, também existem *softwares* específicos para a escrita em notação ABC. Em relação aos demais *softwares* de composição (notação gráfica), esses *softwares* possuem, dentre outras vantagens, as seguintes:

- Praticamente todos os programas para notação ABC são gratuitos;
- Os programas, em geral, são compactos e possíveis de se utilizar em computadores de baixo rendimento ou mesmo *smartphones*;
- A qualidade das folhas de música que se obtém usando programas para notação ABC são, algumas vezes, melhores do que as obtidas nos programas tradicionais;
- As partituras são criadas em arquivos PostScript ou pdf, e não em formatos reservados aos programas tradicionais, os quais dificultam o compartilhamento.

Uma música escrita em notação ABC é formada basicamente por duas partes: o cabeçalho, que contém informações sobre a música como o seu título, compositor, tonalidade, etc; e o corpo, que contém a música. Além da escrita de toda a notação musical através da notação ABC, existem muitos comandos disponíveis para a especificação do formato da página, como formato do papel, margens, tipo de letra, entre outros. É possível encontrar toda a notação ABC para escrita de partitura em formato textual em (GONZATO, 2007). Como exemplo, considere o pequeno trecho de partitura escrito na notação ABC:

```
X: 1 % início do cabeçalho
K: C % tonalidade: dó maior
C D E F G A B c | c d e f g a b
c' |
```

A partitura produzida a partir do exemplo dado é apresentada na Figura 8.



Figura 8 – Partitura da notação ABC escrita acima.

Para a leitura da notação textual existem muitos *softwares* e páginas na internet que oferecem suporte a notação ABC. Dentre os *softwares* destaca-se o JedABC, um editor que torna a escrita em notação mais simples e rápida. Para a conversão de músicas em ABC para folhas de música, pode-se usar o *abcm2ps*, um *software* livre que lê arquivos ABC e os converte para o formato *PostScript*, que pode ser visualizado e impresso pela aplicação *GhostScript*. O *GhostScript* permite a conversão de arquivos *PostScript* em vários formatos de leitura, entre eles o PDF. Dentre os *softwares*, um que é bastante interessante é o *abc2midi*, pois ele converte arquivos ABC em arquivos MIDI, o que permite uma percepção da partitura que está sendo escrita.

Entre as páginas da internet que permitem algum tipo de interação com a notação, cabe destacar o *website* www.mandolintab.net/abcconverter.php, que oferece a possibilidade de escrita da notação ABC, sem todas as vantagens apresentadas no JedABC, mas que, ao término da escrita, possibilita ao usuário a geração da partitura musical em *pdf* e também a produção de um arquivo MIDI correspondente ao que foi escrito usando a notação ABC; a essas funcionalidades juntam-se outras possibilidades não mencionadas aqui, mas que são todas gratuitas.

Nos tópicos futuros, serão apresentados os textos na notações ABC correspondentes às partituras utilizadas nos experimentos. Como não é objetivo deste trabalho aprofundar-se nos detalhes da notação ABC, recomenda-se ao leitor que possua dúvidas ou questões mais específicas sobre a notação recorrer às referências citadas.

4.1.1 Representação em Bits de Partituras Escritas em Notação ABC

Nesta etapa do método, o objetivo é, ao obter-se a partitura escrita em notação ABC, realizar a conversão dessa notação textual para o formato binário, através de ferramentas conhecidas. Como em toda a notação são utilizados caracteres ASCII para a representação das notas e toda estrutura textual, então, é possível, a partir do conhecimento da tabela ASCII, que relaciona cada caractere a uma codificação binária representada através de 8 bits, converter toda a partitura escrita em notação ABC em bits a serem inseridos utilizando técnicas apresentadas posteriormente.

Na prática, o que se faz é importar o arquivo .txt contendo determinada partitura em notação ABC (disponíveis no apêndice A) e armazená-lo num vetor, para que possam ser realizados os devidos processamentos para sua conversão em bits. Inicialmente, realiza-se a conversão do texto para o formato *string*, o qual é, então, convertido para binário e, em seguida, para decimal. Essa conversão para decimal, seguida do cálculo da transposta do vetor que representa o texto da notação ABC, tem por objetivo identificar todas as vezes em que ao longo do vetor aparece o caractere “@”, que, na convenção assumida neste trabalho, servirá como um delimitador que indica que deve ser criada uma nova linha no momento da extração dos bits e geração do arquivo .txt recuperado. Ao fim desse processo, o vetor nomeado *name1* passará por mais uma conversão decimal-binário utilizando a função *dec2bin* do Matlab. Com isso, o vetor *mensagem_bin* é a representação em bits da partitura musical escrita em notação ABC. Essas etapas são executadas empregando o seguinte código de Matlab:

```
arquivo = input('Digite o nome do arquivo .txt: '); % Digita-se o nome do arquivo
.txt colocando o seu nome entre apóstrofos.
arquivo = char(importdata(arquivo)); % Importa-se o arquivo .txt e convertemos
para decimal usando char.
name1 = bin2dec(dec2bin(num2str(arquivo))); % Converte-se 'arquivo' para string,
depois para binário e, em seguida, decimal.
name1 = name1'; % Calcula-se a transposta de name1 (diferente do caso 1).
mensagem_bin = dec2bin(name1); % Retorna-se o vetor 'name1' para binário.
```

4.2 Inserção dos Bits

Nesta seção, serão apresentadas de forma detalhada todas as etapas do método utilizado para a inserção dos bits referentes ao texto em notação ABC em determinado sinal de áudio. Cada bit é inserido, por meio de uma regra de ajuste de fase (FAN; WANG, 2009), (BRAIN; GREGORY, 2001), (DONG; BOCKO; IGNJATOVIC, 2004), em

componentes específicas da DFT de segmentos do respectivo sinal de áudio. A Figura 9 apresenta um diagrama de blocos do processo de inserção dos bits no sinal de áudio.

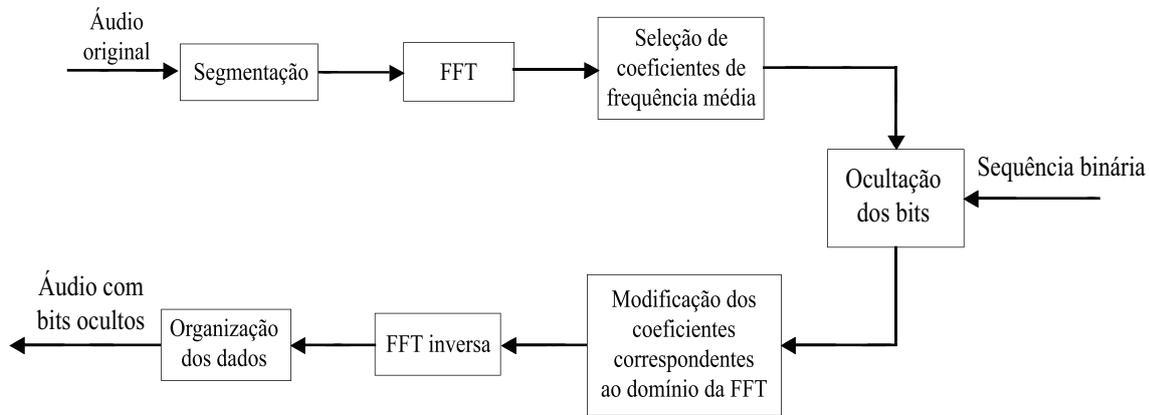


Figura 9 – Diagrama de blocos do processo de inserção de bits em um sinal de áudio.

Inicialmente, considera-se um sinal de áudio digital em formato não-comprimido (*wave*) representado como

$$A = \{A(i) | 1 \leq i \leq L\}. \quad (4.1)$$

Em seguida, considera-se o vetor contendo os bits que representam a partitura em notação ABC e denota-se esse vetor por

$$w = \{w(i) | 1 \leq i \leq M\}. \quad (4.2)$$

A inserção propriamente dita dos bits no sinal de áudio é realizada mediante a execução dos passos a seguir:

Passo 1. *Segmentação do sinal de áudio original:* o vetor A que representa o áudio original deve ser segmentado em, no mínimo, M segmentos. Pode-se segmentar todo o vetor ou apenas parte dele; tal escolha depende da relação entre os comprimentos L e M . A seguir, considera-se que M divide L e que todo o vetor A é segmentado. Os segmentos obtidos são denotados por

$$A_s = \left\{ A_s(h, l) | 1 \leq h \leq M, 1 \leq l \leq \frac{L}{M} \right\}. \quad (4.3)$$

Passo 2. *Cálculo da DFT:* calcula-se a transformada discreta de Fourier de cada segmento de A_s . O resultado obtido é denotado por

$$A'_s = \{A'_s(h, l)\}. \quad (4.4)$$

Passo 3. *Seleção dos coeficientes de frequências médias:* em cada segmento de A'_s , são selecionadas C componentes consecutivas na faixa de frequências médias. O objetivo é obter uma boa imperceptibilidade, isto é, a música, após a inserção dos bits,

não deve apresentar diferença quando comparada fa música original. Os novos segmentos (C componentes de cada segmento de A'_s) são denotados por A_c .

Passo 4. Inserção dos bits da partitura: em cada segmento de A_c , é escolhida a componente com maior amplitude para inserção de cada bit de w . Tais componentes são seleccionadas a fim de proporcionar robustez ao método, contra manipulações como a compressão mp3. Uma análise mais formal acerca desse aspecto pode ser desenvolvida considerando que a componente com amplitude ρ_0 e fase φ utilizada para inserção do bit da partitura, após alguma manipulação, tem sua amplitude modificada para ρ e sua fase incrementada de δ . Com base na Figura 10, pode-se escrever

$$\begin{cases} \rho \cdot \cos \delta - \rho_0 = \Delta_1, \\ \rho \cdot \sin \delta = \Delta_2, \end{cases} \quad (4.5)$$

em que Δ_1 e Δ_2 são, supostamente, dois ruídos unidimensionais com alguma distribuição. A partir das equações acima, tem-se

$$\begin{cases} \tan \delta = \frac{\Delta_2}{\Delta_1 + \rho_0}, \\ \rho^2 = (\rho_0 + \Delta_1)^2 + \Delta_2^2. \end{cases} \quad (4.6)$$

Com base nas duas últimas equações, é possível concluir que um aumento em ρ_0 provoca um acréscimo em ρ e um decréscimo em $|\delta|$. Dessa forma, a fase de um elemento com a maior amplitude será menos sensível a algum tipo de manipulação.

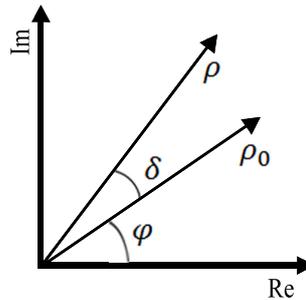


Figura 10 – Representação das modificações sofridas por uma componente de um vetor, originalmente de amplitude ρ_0 e fase φ , em função da inserção de um bit via modulação de fase, resultando em uma amplitude de ρ e um acréscimo de δ na fase.

A inserção propriamente dita dos bits é feita conforme uma técnica de modulação de fase (DONG; BOCKO; IGNJATOVIC, 2004). Suponha que, em dado segmento de A_c , a componente de maior amplitude A_m possua fase denotada por α , $0 \leq \alpha \leq 2\pi$. A inserção do bit w da partitura, correspondente a esse segmento, modifica A_m para

$$A'_m = A_m + \gamma, \quad (4.7)$$

em que γ é um número real positivo utilizado para distinguir a componente que recebeu o bit das demais componentes, e modifica α para

$$\alpha' = \begin{cases} (temp + 0.5) \times step, & \text{se } w = 0 \text{ e } \text{mod}(temp, 2) = 0 \\ (temp - 0.5) \times step, & \text{se } w = 0 \text{ e } \text{mod}(temp, 2) = 1 \\ & \text{e } \alpha < (temp + 0.5) \times step \\ (temp + 1.5) \times step, & \text{se } w = 0 \text{ e } \text{mod}(temp, 2) = 1 \\ & \text{e } \alpha > (temp + 0.5) \times step \\ (temp + 0.5) \times step, & \text{se } w = 1 \text{ e } \text{mod}(temp, 2) = 1 \\ (temp - 0.5) \times step, & \text{se } w = 1 \text{ e } \text{mod}(temp, 2) = 0 \\ & \text{e } \alpha < (temp + 0.5) \times step \\ (temp + 1.5) \times step, & \text{se } w = 0 \text{ e } \text{mod}(temp, 2) = 0 \\ & \text{e } \alpha > (temp + 0.5) \times step; \end{cases} \quad (4.8)$$

o parâmetro $step$ representa um passo de quantização ímpar-par e $temp = \lfloor \frac{\alpha}{step} \rfloor$ (BRAIN; GREGORY, 2001). Um aumento no valor do $step$ proporciona também um aumento na robustez do esquema de inserção. Entretanto, como tal incremento é acompanhado de distorções no sinal de áudio processado, é importante selecionar o valor do $step$ de forma equilibrada.

Passo 5. *Cálculo da DFT inversa:* esta etapa consiste na aplicação aos segmentos do sinal de áudio, após a inserção dos bits, do procedimento inverso ao realizado no **Passo 2**.

Passo 6. *Reorganização dos segmentos:* neste último passo do processo de inserção dos bits, os segmentos são reorganizados, a fim de que o vetor correspondente ao sinal de áudio seja remontado e o sinal de áudio completo, com os bits da partitura inseridos, seja obtido.

Após a inserção dos bits de uma partitura no sinal de áudio, algumas componentes do vetor resultante adquirem uma parte imaginária residual, tornando-se números complexos; tal resíduo é proveniente do ajuste de fase implementado quando da inserção de cada bit da partitura. Dessa forma, a inscrição dessas componentes como amostras de um sinal de áudio, que deveriam ser puramente reais, fica comprometida. A solução adotada para essa questão é, então, desprezar a referida parte imaginária residual e inscrever nas amostras do sinal de áudio produzido apenas a parte real das componentes do vetor. Embora este *truncamento* aparentemente não prejudique as propriedades sonoras do sinal, podem ocorrer erros na extração dos bits, mesmo sem que tenha sido feita qualquer manipulação do áudio; isso depende fundamentalmente dos parâmetros empregados no processo de inserção (γ e $step$).

4.2.1 Adaptação para Áudio Mono

A proposta apresentada na sessão anterior se destina a inserção de bits em sinais de áudio mono, ou seja, em sinais de áudio que apresentam apenas um canal para reprodução de um determinado áudio. Áudios mono não são os mais populares comercialmente e também apresentam uma relativa perda na qualidade sonora, pelo fato de não permitir um bom palco sonoro, ou seja, não permite a criação de um ambiente sonoro para os ouvintes, diferentemente do que acontece com sinais de áudio estéreo.

Como apresentado na seção anterior, quando os bits referentes a partitura musical são ocultados num sinal de áudio mono, é possível verificar o surgimento de um resíduo imaginário após o cálculo da DFT inversa por conta do ajuste de fase realizado no processo de inserção dos bits, destacado no Passo 4. Contudo, conforme também se comentou na seção anterior, a realização de uma simples *eliminação* desta parte imaginária pode acarretar erros no processo de extração dos bits.

Neste caso, a solução pode ser obtida lançando mão do fato de que, ao se calcular a DFT de um vetor puramente real, obtém-se como resultado um vetor de conteúdo complexo cuja parte imaginária possui simetria ímpar, como o exemplo ilustrado na Figura 11, que considera a transformada do vetor

$$v[n] = [2, 3, 6, 7, 6, 5, 4]. \quad (4.9)$$

Ao se calcular a DFT deste vetor v , obtém-se o vetor

$$V[\omega] = [33 + j \cdot 0, -7.7959 - j \cdot 0.627, -1.3629 + j \cdot 2.1906, -0.3412 + j \cdot 0.2408, -0.3412 - j \cdot 0.2408, -1.3629 - j \cdot 2.1906, -7.7959 + j \cdot 0.627]. \quad (4.10)$$

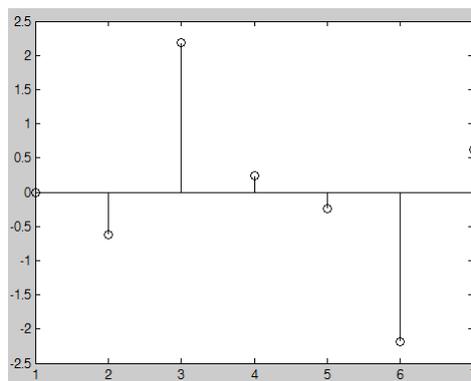


Figura 11 – Parte imaginária da DFT de um vetor real, com ênfase para a simetria ímpar entre as componentes.

Considerando o fato exposto, propõe-se o seguinte método como substituição do simples *truncamento* da referida parte imaginária residual: após a realização do procedimento de inserção dos bits em cada segmento do vetor que representa o áudio original a ser processado, ajustam-se as partes imaginárias das componentes apropriadas do mesmo

segmento, a fim de forçar o aparecimento de uma simetria ímpar; com isso, quando do cálculo da DFT inversa, não se verificará a presença de qualquer resíduo imaginário.

As linhas de programação abaixo representam a implementação da técnica que se acabou de descrever:

```

for p = 1:comp_mensagem_bin
vector_position(p) = position(p) + 49; % Cria um vetor com as posições onde cada
bit foi inserido referente ao tamanho do áudio original.
novop(p,vector_position(p)) = complexo_m(p); % para que possa ser atribuído a esta
posição (no áudio) o sinal com o bit embutido.
novop(p,seg1 - vector_position(p) + 2) = conj(complexo_m(p)); % Atribui-se o
conjugado do elemento simétrico com a fase modificada.

```

4.2.2 Adaptação para Áudio Estéreo

A técnica descrita na seção anterior pode ser adaptada para inserção dos bits da partitura em sinais de áudio estéreo, os quais contêm dois canais; como citado anteriormente, este tipo de áudio possibilita a criação de um palco sonoro mais rico e que permite um maior aprofundamento e uma melhoria na qualidade sonora de um áudio em relação ao áudio mono. Os princípios de inserção e de extração dos bits permanecem os mesmos, porém, a representação do sinal de áudio estéreo por meio de um vetor é realizada conforme descrição a seguir. Como um sinal de áudio estéreo possui dois canais, o que se faz é criar um vetor

$$A = \left\{ A(i) = A_l(i) + j \cdot A_r(i) \mid 1 \leq i \leq L, j = \sqrt{-1} \right\} \quad (4.11)$$

com L componentes complexas, em que $A_l(i)$ e $A_r(i)$ são, respectivamente, a i -ésima componente do canal esquerdo e a i -ésima componente do canal direito do sinal de áudio correspondente. Daí, os passos 1 a 6 descritos para sinais de áudio do tipo mono podem ser aplicados, bem como as etapas envolvidas no processo de extração dos bits, apresentadas no próximo tópico.

Além de os sinais de áudio estéreo serem mais difundidos entre os usuários, outro aspecto motivou a adaptação do método. No caso do áudio estéreo, como os dois canais são interpretados como parte real e parte imaginária, e considerados tanto no processo de inserção dos bits quanto no de extração, a limitação relacionada ao aparecimento de resíduos imaginários, observada para áudio mono, não ocorre.

4.3 Extração dos Bits

O processo de extração dos bits para obtenção da partitura é praticamente o inverso do processo de inserção e independe do fato de o áudio utilizado ser mono ou estéreo. A Figura 12 apresenta um diagrama da sequência de ações do processo de recuperação dos bits.

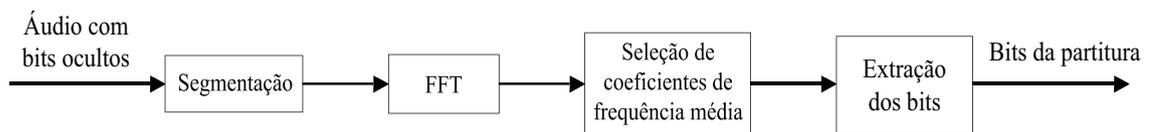


Figura 12 – Diagrama de blocos do processo de extração dos bits.

Num passo inicial, é realizada uma segmentação do sinal de áudio. Após isso, é efetuado o cálculo da DFT de cada segmento e isoladas as componentes associadas às frequências médias. Em cada segmento resultante, seleciona-se a componente de maior amplitude, a qual possui fase denotada por α' , $0 \leq \alpha' < 2\pi$. O bit \hat{w} de w , que, sendo extraído de forma correta, coincide com o bit w inserido em determinado segmento do áudio original, é recuperado segundo a regra:

$$\hat{w} = \begin{cases} 0, & \text{se } \text{mod}(\lfloor \frac{\alpha'}{\text{step}} \rfloor, 2) = 0, \\ 1, & \text{se } \text{mod}(\lfloor \frac{\alpha'}{\text{step}} \rfloor, 2) = 1. \end{cases} \quad (4.12)$$

Finalmente, os bits extraídos são organizados convenientemente e convertidos para caracteres ASCII, permitindo a reprodução de um arquivo no formato .txt que corresponde à partitura da música em questão.

Conforme indicado na Seção 4.1 deste trabalho, a geração de um arquivo no formato pdf ou em formato de imagem, apresentando a partitura da música escrita com os símbolos musicais convencionais, pode ser feita utilizando serviços gratuitos. É possível realizar essa conversão copiando o conteúdo do arquivo de texto resultante da extração dos bits e colando, por exemplo, na interface web oferecida em <http://www.mandolintab.net/abcconverter.php>. Na mesma interface, o usuário também pode ouvir o resultado (em formato *midi*) da execução da partitura gerada.

5 Simulações e Resultados

NESTE capítulo, são apresentadas as simulações realizadas utilizando os métodos de inserção e extração de bits de partitura descritos no Capítulo 4. Para isso foram selecionados três áudios diferentes, todos eles em suas versões mono e estéreo. As simulações foram realizadas a partir de códigos escritos no Matlab®, o qual fornece uma série de comandos que permitem ler, manipular e escrever arquivos de áudio.

5.1 Definição dos Parâmetros Iniciais

Antes de se iniciar o procedimento de inserção dos bits que representam a partitura musical em notação ABC é necessário que haja a definição de todos os parâmetros a serem utilizados na realização das simulações, como, por exemplo, definição dos áudios utilizados e escrita de suas respectivas partituras em notação ABC, como mostrado a seguir.

5.1.1 Áudios Utilizados

O primeiro passo para a realização das simulações é a seleção de quais músicas serão utilizadas como os sinais de áudio. Para este trabalho foram tomadas como material de análise as seguintes canções:

1. *All my loving*, com duração de 02 minutos e 08 segundos.
2. *Here, there and everywhere*, com duração de 02 minutos e 25 segundos.
3. *Yesterday*, com duração de 02 de minutos e 06 segundos.

Todas essas canções pertencem aos compositores *John Lennon* e *Paul McCartney*, do conjunto britânico *The Beatles*. Foram tomadas as versões de áudio mono e estéreo para cada música, todas de arquivo em formato *wave* e com frequência de amostragem $f_s = 44.100$ Hz. O detalhamento sobre cada áudio (mono e estéreo) para cada música é mostrado nas próximas seções.

5.1.2 Transcrição da Partitura em Notação Musical para Notação ABC

O passo seguinte, a partir da definição das faixas utilizadas, é a transcrição da partitura em notação musical de todas as faixas para a notação ABC, usando as ferramentas disponíveis em (GONZATO, 2007). As partituras escritas em notação ABC bem como as versões correspondentes em notação musical usual estão disponíveis no Apêndice A.

Cada representação das partituras em notação ABC teve diferentes abordagens. Por exemplo, na canção *Yesterday* foram utilizados os parâmetros que definem a formatação da página na qual será exibida / impressa a partitura em notação musical, enquanto que, nas canções *All my loving* e *Here, there and everywhere*, foi utilizada apenas a notação ABC puramente com cabeçalho e corpo, sem definições de formatação da página.

Vale lembrar a utilização do caractere "@" desempenhando função de delimitador de linhas quando for realizado o processo de extração dos bits e escrita destes em arquivo .txt, ou seja, sempre que identificado um "@", deve-se iniciar uma nova linha para prosseguimento da leitura. Toda a conversão dos caracteres textuais da notação ABC é realizada dentro da linha de código desenvolvida no Matlab e apresentada no capítulo anterior.

5.2 Inserção de Bits

O procedimento de inserção dos bits em cada áudio segue o algoritmo apresentado na Seção 4.2. Conforme mencionado, foram utilizados áudios mono e estéreo para cada uma das canções. A partir deste ponto, são discutidos os resultados obtidos em cada um desses tipos de áudio.

5.2.1 Adaptação para Áudio Mono

Como foi citado anteriormente, os áudios mono utilizados estão em arquivos no formato *wave*. A seguir são listados os comprimentos de cada sinal de áudio, denotados por L , bem como o comprimento do vetor binário que representa a notação ABC, denotado por M , utilizando a definição

$$2^{\lceil \log_2 \lfloor \frac{L}{M} \rfloor \rceil} \quad (5.1)$$

para que estes comprimentos sejam representados por múltiplos de 2.

- *All my loving* $\rightarrow L = 5656320$ e $M = 6020$ bits

$$\text{Comprimento do segmento: } 2^{\lceil \log_2 \lfloor \frac{5656320}{6020} \rfloor \rceil} = 512$$

- *Here, there and everywhere* $\rightarrow L = 6407424$ e $M = 4235$ bits

$$\text{Comprimento do segmento: } 2^{\lceil \log_2 \lfloor \frac{6407424}{4235} \rfloor \rceil} = 512$$

- *Yesterday* $\rightarrow L = 5559552$ e $M = 10997$ bits

Comprimento do segmento: $2^{\lceil \log_2 \lfloor \frac{5559552}{10997} \rfloor \rceil} = 256$

Nos parâmetros listados, observa-se que, com o propósito de se obter um aumento na capacidade de inserção de bits, o comprimento do segmento de cada áudio é variável de acordo com a necessidade existente, podendo assumir os valores 256 e 512. Ao se calcular a DFT com 256 (ou 512) pontos de cada segmento, são selecionadas 8 componentes da posição 50 a 57 e, dentre essas amostras, a de maior amplitude é a escolhida para a inserção do bit, baseado na técnica de modulação de fase apresentada no Passo 4 da Seção 4.2. Os valores assumidos para γ são $\gamma = 0,05$ e $\gamma = 0,15$ e os valores assumidos para o *step* são $step = \frac{\pi}{6}$, $step = \frac{\pi}{7}$ e $step = \frac{\pi}{8}$.

Em seguida, esses componentes que foram submetidos à inserção de bits são retornados aos seus segmentos de origem e então é realizada a modificação do elemento simétrico ao elemento cuja fase foi alterada para o seu conjugado, permitindo que, quando calculada a DFT inversa, o resultado seja um vetor puramente real, eliminando o resíduo imaginário que fora verificado anteriormente. Após o cálculo da DFT inversa, todos os segmentos do vetor do áudio são unidos de maneira a obter como resultado o áudio com os bits inseridos. É relevante mencionar que, do ponto de vista de percepção auditiva, após a inserção dos bits, não foram identificadas degradações nos áudios dos arquivos usados nos experimentos. Uma avaliação completa desse aspecto pode ser conduzida na verificação do valor quadrático médio da diferença entre os sinais.

5.2.2 Adaptação para Áudio Estéreo

Os áudios estéreo são mais presentes comercialmente do que os áudios mono e se tornaram um formato padrão para as músicas gravadas a partir dos anos 60. Por este motivo esse tipo de áudio também foi utilizado para a realização das simulações. Os áudios aqui utilizados também estão em formato *wave*. A linha de código exibida abaixo é responsável pela “junção” dos sinais dos dois canais isolados (esquerdo e direito), mapeando-os num único sinal na forma complexa:

```
audio_original_complex = audio_original(:, 1) + i * audio_original(:, 2);
```

Na linha acima, *audio_original* representa o áudio propriamente dito, que, no Matlab, é lido de um arquivo externo utilizando a função *audioread*, e *audio_original_complex* representa o vetor de áudio na forma de número complexo. Daqui por diante esse vetor pode ser tratado como um sinal de áudio com canal único e todos os passos citados anteriormente são realizados para este sem nenhuma objeção. O comprimento de cada sinal de áudio estéreo é exatamente igual a sua versão mono, logo não se faz necessário aqui reescrever todas as informações referentes ao comprimento dos áudios nem da notação ABC.

No caso de os sinais de áudio utilizados não terem capacidade (comprimento) suficiente para inserção dos bits, existem duas possibilidades de elevação dessa capacidade. A primeira é a implementação da redução do comprimento dos segmentos para o tamanho de 128, o que duplicaria a capacidade do áudio; a outra alternativa válida seria considerar os dois canais de forma independente e inserir os bits como se esses canais fossem dois sinais de áudio mono. Este procedimento também duplicaria o número de blocos para a inserção de bits.

5.3 Extração dos bits

Após a inserção dos bits correspondentes às partituras na notação ABC de cada canção considerada nas simulações, foram elaborados códigos para realizar o procedimento de extração explicado no Capítulo 4. Além dos experimentos realizados para arquivos no formato *wave*, também foram realizados experimentos de extração dos bits das partituras após a conversão dos arquivos de áudio para o formato ISO/MPEG-1 Audio Layer III (ou mp3); este é, de fato, um dos tipos de arquivo de áudio mais difundidos nas redes de comunicação. A conversão para mp3 proporciona a compressão do áudio original (formato *wave*) e pode ser realizada por meio de uma gama bastante ampla de aplicativos. Neste trabalho todos os arquivos de áudio no formato *wave* foram convertidos para o formato mp3 utilizando o *software* MediaHuman Audio Converter.

Nos experimentos, foram consideradas quatro taxas de compressão na conversão do sinal para mp3: 320 kbps, 256 kbps, 192 kbps e 128 kbps. Com essas taxas, o decréscimo na qualidade do áudio original não é perceptível e os tamanhos dos arquivos produzidos são relativamente pequenos, considerando o espaço requerido para armazenamento e o tempo para *upload* / *download* usando uma conexão de banda larga doméstica. O objetivo é verificar quantos bits são extraídos com erro em cada um desses casos e formar uma ideia qualitativa a respeito da robustez do método a este tipo de manipulação.

Do procedimento de inserção de bits apresentado na seção anterior, deixa-se registrado o comprimento da notação ABC na forma binária afim de utilizar essa informação no momento da segmentação do áudio com os bits ocultados; no ato da extração desses bits, não há como conseguir esta informação de maneira cega. Todo o processo descrito na Seção 4.3 é realizado e os resultados para ambos os tipos de áudio (mono e estéreo) e formatos (*wave* e mp3) são listados nas Tabelas 1 a 6.

As tabelas 1, 2 e 3 apresentam o números de bits errados extraídos dos áudios mono. Para os áudios no formato *wave* observa-se que todos os bits foram extraídos corretamente, contudo os resultados obtidos da extração dos bits para os áudios sob compressão mp3 merecem uma análise mais detalhada.

Observando as três primeiras tabelas, para áudios mono, é possível perceber que,

embora sejam utilizadas as faixas completas, a medida que os áudio são submetidos a taxas de compressão maiores, a quantidade de bits errados extraídos aumenta consideravelmente. De qualquer forma, um áudio à taxa de compressão de 192 kbps, por exemplo, é um áudio de boa qualidade sonora e que não possui tamanho grande e, assim, também é um arquivo que permite a sua difusão por usuários sem qualquer restrição. É válido esclarecer que não foram utilizados os dados obtidos da faixa *Yesterday* para a taxa de compressão mp3 de 128 kbps, em virtude de o texto em notação ABC para esta faixa ter, aproximadamente, o dobro do comprimento dos demais textos, diminuindo assim a sua capacidade de inserção.

É importante notar que, para todas as faixas e em quaisquer taxas de compressão, uma mudança em γ de 0,05 para 0,15 faz com que as taxas de bits extraídos com erro diminuam. Por exemplo, na Tabela 3, para uma taxa de compressão de 192 kbps, a diferença entre a taxa de bits extraídos com erro para $\gamma = 0,05$, $step = \frac{\pi}{7}$ e $\gamma = 0,15$, $step = \frac{\pi}{7}$ é de 24 vezes mais para o primeiro caso. Naturalmente, isso acontece ao preço de uma maior distorção do sinal, visto que representa um incremento direto na amplitude da componente que recebe o bit.

As Tabelas 4, 5 e 6 apresentam os números de bits errados extraídos dos áudios estéreo. Para os áudios no formato *wave* observa-se que todos os bits foram extraídos corretamente. Porém, os resultados obtidos da extração dos bits para os áudio sob compressão mp3 não apresentam o mesmo resultado, contendo, em números absolutos, uma quantidade maior de erros que aquela observada para os áudios mono correspondentes.

Da mesma forma que para áudios mono, a mudança de γ de 0,05 para 0,15 traz uma redução na taxa de bits extraídos com erro nos áudios estéreo. Contudo, observa-se que quando esses áudios são submetidos a uma taxa de compressão de 256 kbps não existe a extração total de bits corretos para quaisquer parâmetros adotados para γ e $step$. Em geral, o número de erros na extração de bits de sinais mono é menor, o que sugere que o método, quando aplicado a sinais estéreo, possui maior sensibilidade à compressão mp3.

Tabela 1 – Números de bits extraídos com erro para diversos formatos de áudio mono e diferentes combinações dos parâmetros $step$ ($\pi/6, \pi/7, \pi/8$) e γ , para a música *All my loving*.

Formato	$\gamma = 0,05$			$\gamma = 0,15$		
	$\pi/6$	$\pi/7$	$\pi/8$	$\pi/6$	$\pi/7$	$\pi/8$
wave	0	0	0	0	0	0
mp3 (320 kbps)	0	0	0	0	0	0
mp3 (256 kbps)	0	0	0	0	0	0
mp3 (192 kbps)	11	8	17	1	0	0
mp3 (128 kbps)	111	98	111	38	33	40

Tabela 2 – Números de bits extraídos com erro para diversos formatos de áudio mono e diferentes combinações dos parâmetros $step$ ($\pi/6, \pi/7, \pi/8$) e γ , para a música *Here, there and everywhere*.

Formato	$\gamma = 0,05$			$\gamma = 0,15$		
	$\pi/6$	$\pi/7$	$\pi/8$	$\pi/6$	$\pi/7$	$\pi/8$
wave	0	0	0	0	0	0
mp3 (320 kbps)	0	0	0	0	0	0
mp3 (256 kbps)	0	0	0	0	0	0
mp3 (192 kbps)	0	0	0	0	0	0
mp3 (128 kbps)	13	15	16	1	5	3

Tabela 3 – Números de bits extraídos com erro para diversos formatos de áudio mono e diferentes combinações dos parâmetros $step$ ($\pi/6, \pi/7, \pi/8$) e γ , para a música *Yesterday*.

Formato	$\gamma = 0,05$			$\gamma = 0,15$		
	$\pi/6$	$\pi/7$	$\pi/8$	$\pi/6$	$\pi/7$	$\pi/8$
wave	0	0	0	0	0	0
mp3 (320 kbps)	1	1	2	0	0	0
mp3 (256 kbps)	4	2	9	0	1	0
mp3 (192 kbps)	47	48	91	7	2	9

Tabela 4 – Números de bits extraídos com erro para diversos formatos de áudio estéreo e diferentes combinações dos parâmetros $step$ ($\pi/6, \pi/7, \pi/8$) e γ , para a música *All my loving*.

Formato	$\gamma = 0,05$			$\gamma = 0,15$		
	$\pi/6$	$\pi/7$	$\pi/8$	$\pi/6$	$\pi/7$	$\pi/8$
wave	0	0	0	0	0	0
mp3 (320 kbps)	6	9	8	1	0	1
mp3 (256 kbps)	39	51	45	4	2	7
mp3 (192 kbps)	341	402	462	48	32	73

5.4 Avaliação Preliminar dos Efeitos da Extração Incorreta dos Bits

Diante dos resultados apresentados, torna-se importante avaliar quais os impactos de uma extração errônea dos bits que representam a partitura na notação ABC de uma música. Tal avaliação é realizada tomando por base a canção *All my loving* em formato mono. Para isso, são consideradas três combinações dos parâmetros γ e $step$. A primeira é (i) $\gamma = 0,15$ e $step = \pi/6$, em que a sequência binária é recuperada com 1 erro; a segunda é (ii) $\gamma = 0,05$ e $step = \pi/6$, em que a sequência binária é recuperada com 11 erros; a terceira é (iii) $\gamma = 0,15$ e $step = \pi/7$, em que a sequência binária é recuperada com 33 erros (vide Tabela 1).

Tabela 5 – Números de bits extraídos com erro para diversos formatos de áudio estéreo e diferentes combinações dos parâmetros *step* ($\pi/6, \pi/7, \pi/8$) e γ , para a música *Here, there and everywhere*.

Formato	$\gamma = 0,05$			$\gamma = 0,15$		
	$\pi/6$	$\pi/7$	$\pi/8$	$\pi/6$	$\pi/7$	$\pi/8$
wave	0	0	0	0	0	0
mp3 (320 kbps)	5	8	6	0	1	1
mp3 (256 kbps)	42	45	40	3	5	7
mp3 (192 kbps)	322	380	450	43	38	66

Tabela 6 – Números de bits extraídos com erro para diversos formatos de áudio estéreo e diferentes combinações dos parâmetros *step* ($\pi/6, \pi/7, \pi/8$) e γ , para a música *Yesterday*.

Formato	$\gamma = 0,05$			$\gamma = 0,15$		
	$\pi/6$	$\pi/7$	$\pi/8$	$\pi/6$	$\pi/7$	$\pi/8$
wave	0	0	0	0	0	0
mp3 (320 kbps)	14	9	3	1	0	1
mp3 (256 kbps)	45	37	54	9	5	12
mp3 (192 kbps)	273	357	425	64	95	161

Nas Figuras 13, 14 e 15, são apresentadas as partituras da canção *All my loving* na notação ABC, recuperadas após a sequência de bits ter sido extraída com 1, 11 e 33 erros, respectivamente. Os caracteres com erro, em comparação à partitura original correta (vide Figura 19 no apêndice) são apresentados de forma sublinhada. Na Figura 13, por exemplo, o único bit errado leva a um erro no caráter “e” da palavra “when” da letra da música, que é trocado pelo caráter “u” (vide linha 17 da figura). Nas outras partituras, o número de caracteres sublinhados é maior. Nesse contexto, é relevante verificar que impacto esses erros nos caracteres provocam quando se tenta recuperar automaticamente a partitura na notação musical usual a partir da partitura na notação ABC. Isso é discutido a seguir.

Na Figura 16, é apresentada a partitura na notação usual, recuperada automaticamente após a sequência de bits ter sido extraída com 1 bit errado. Conforme se observou na partitura em notação ABC correspondente, a única diferença dessa partitura em relação à partitura correta (vide Figura 20) é na palavra “when” da letra, em que o caráter “e” é trocado por “u”.

A recuperação da partitura na notação usual, após a sequência de bits ter sido extraída com 11 bits errados, não foi possível, porque, conforme comentado, alguns desses erros corromperam informações fundamentais de cabeçalho, necessárias à “inicialização” da escrita musical. Por outro lado, na Figura 17, é apresentada a partitura na notação usual, recuperada após a sequência de bits ter sido extraída com 33 bits errados. Neste

```

X:02
T: All my loving
T: From Album 'With The Beatles' 1964 UK
O: Lennon/McCartney
M:4/4
L:1/4
Q:1/4=120
K: E
%
A G |: F2 G A | B c2 d | e2 e d | c A2 c |
w: Close your eyes and I'll kiss you, to-mor-row I'll miss you. Re-
w: |tend that I'm kiss-ing the lips I am miss-ing and
c2 c B | A2 F E | F4- | E z A G |
w: mem-ber I'll al-ways be true._ And then
w: hope that my dreams will come true._ And then
F2 G A | B2 c d | e2 e d | c G-c/2c/2 |
w: whun I'm a-way I'll write home ev-ery day,_ and I'll
w: while I'm a-way I'll write home ev-rey day,_ and I'll
c2 c B | A G2 F | E4- |[1 E z A G :|| [2 E z e d |
w: send all my lov-ing to you._ I'll pre - All my
w: send all my lov-ing to you._
c G2> | G2 F E F | G4- | G z e d |
w: lov-ing, I will send to you._ All my
c G2> | G2 F E E | E4- | E z z2 ||
w: lov-ing, dar-ling I'll be true._

```

Figura 13 – Partitura da canção *All my loving* na notação ABC, recuperada após a sequência de bits ter sido extraída com 1 bit errado.

caso, o número de erros relativamente maior que os dos casos anteriores, corrompeu significativamente a disposição dos símbolos musicais, a própria letra da música e os demais elementos textuais. Isso pode ser verificado em quase todas as regiões da imagem.

Embora as consequências dos erros na extração dos bits possam ter impactos visuais consideráveis (como apresentado na Figura 17, por exemplo), comprometendo a escrita e a leitura da partitura, cabe observar que a taxa de erro de bit é relativamente pequena, considerando o cenário de aplicação proposto. No último caso, são 33 bits errados em 6020 bits ocultados, o que corresponde a 1 bit errado a cada 182 bits extraídos. Isso sugere que

```

X:02
T: All my loving
T: From Album 'With The Beatles' 1964 UK
O: Lennon/McCartney
M:4/4
L:1/4
Q:1/4=120\K: E
%
A G |: F2 G A } B c2 d | e2 e d | c A2 c |
w: Close your eyeq and I'll kiss you, to-mor-row I'll miss you. Re-
w: |tend that I'm kiss
ing the lips I am miSs-ing and
c2 c B | A2 F E | F4- | E z A G |
w: mem-ber I'll al-ways be true._ And then
w: hope that my dreams will come true._ And thel
F2 G A | B2 c d | e2 e d | c G-c/2c/2 |
w: whun I'm a-way I'll write home ev-ery day,_ and I'll
w: while I'm a-way I'll write home ev-rey day,_ and I'll
c2 c B | A G2 F | E4- |[1 E z A G :|| [2 E z e d |
w: send all my lov-ig to yoq._ I'll pre - All my
w: send all my lov-ing to you._
c G2> | G2 F E F | G4- | G z e d |
w: lov-ing, I will send to you._ All my
c G2> | G2 F E E | E4- | E z z2 ||
w: lov-ing, dar-ling I'll be true._

```

Figura 14 – Partitura da canção *All my loving* na notação ABC, recuperada após a sequência de bits ter sido extraída com 11 bits errados.

se observe a forma como esses bits errados se distribuem ao longo da sequência de bits associada à partitura na notação ABC. Tal distribuição, para os três casos considerados, é apresentada na Figura 18.

Os padrões visualizados nas subfiguras da Figura 18 sugerem que os bits extraídos com erro ocupam posições aleatórias ao longo da sequência binária; no presente contexto, o uso do termo “aleatórias” tem o intuito de fazer um contraponto a um cenário em que os erros ocorreriam de forma semelhante a um “surto”, corrompendo bits em várias posições consecutivas. Isso permite concluir que, para taxas de erro de bit de ordens

```

X:02
T: All my$lo~ing
T: From Album 'With The Baatles' 1964 UK
O: Lennon/McCartney
M:4/4
L:1/4
Q:1/4=122
K: A
%
A G|: F2 G A | B c2 d | e2 eQd | c A2 c |
wz Close {our eyes and I'll kiss$qou, to-mor-row I'll miss iou. Re-
7: tteNd that I'm kiss-ing the lips I am eiss-ing and
c2 c B |(C2 F E | F4- | E z A G |
w: mem,ber I'll al-ways0be true. _ And then
w: hope thap my dreams0will come true. _ Ind then
F2 G A | @2 c d | e2 e d | c G-c/2c/2 |
w: when I'm a-uay I'll write home ev-ery day, _ and I'll
w: while I'm a-way I'll wvite home ev-rmy eay, _ and I'll
c2 c B | A G2 F | E4- |[1 E z A G :|| [2 E z e d |
w: send all my lov-ing to {ou. _ I'll pre - All my
w: send all my lov-ing to you. _
c G2> | G2 F"E F | G4- | G z e d |
w: lov-ing, I will send to you. _OAll my
c G2> | G2 F E E | E4- | E z z2 ||
w: lov-ing, far-ling I'll be true. _

```

Figura 15 – Partitura da canção *All my loving* na notação ABC, recuperada após a sequência de bits ter sido extraída com 33 bits errados.

próximas às consideradas nas combinações de parâmetros (i), (ii) e (iii), o emprego de um código corretor de erros bastante simples levaria à recuperação correta de todos os bits da partitura na notação ABC. Isso contemplaria, também, outras combinações de parâmetros com números maiores de bits errados na extração e arquivos de áudio diferentes daquele considerado nesta seção. Um aspecto importante a ser levado em conta em tal possibilidade é a capacidade do arquivo de áudio de absorver não apenas os bits resultantes da partitura propriamente dita, mas também os bits de redundância do código. O que se espera é que esse aspecto não seja problemático, visto que a maior parte das músicas tem partituras

All my loving
From Album 'With The Beatles' 1964 UK
Lennon/McCartney

$\text{♩} = 120$

Close your eyes and I'll kiss you, to - mor - row I'll miss you. Re -
mem - - ber I'll al - - ways be true. And then
hope that my dreams will come true. And then

when I'm a - - way I'll write home ev - ery day, and I'll
while I'm a - - way I'll write home ev - rey day, and I'll

send all my lov - ing to you. I'll pre - - - All my
send all my lov - ing to you.

lov - ing, I will send to you. All my
lov - - ing, dar - - ling I'll be true.

Figura 16 – Partitura da canção *All my loving* na notação usual, recuperada após a sequência de bits ter sido extraída com 1 bit errado.

(em números de bits) relativamente curtas, em comparação à sua duração. A eficácia dessa ideia deve ser investigada em trabalhos futuros.

All my loving
From Album 'With The Beatles' 1964 UK
Lennon/McCartney

$\text{♩} = 122$

| F2 G A | B c2 d | e2 e0 d | c A2 c |

mem,ber I'll al - - ways be true. And then then
hope that my dreams will come true. And then then

when I'm a - - way I'll write home ev - - ery day, and
while I'm a - - way I'll write home ev - - ery day, and

send all my lov - ing to you. I'll pre - - All my
send all my lov - ing to you.

lov - - ing, I will

EF | G4 | G2 e d |

lov - - ing, far - - ling I'll be true.

Figura 17 – Partitura da canção *All my loving* na notação usual, recuperada após a sequência de bits ter sido extraída com 33 bits errados.

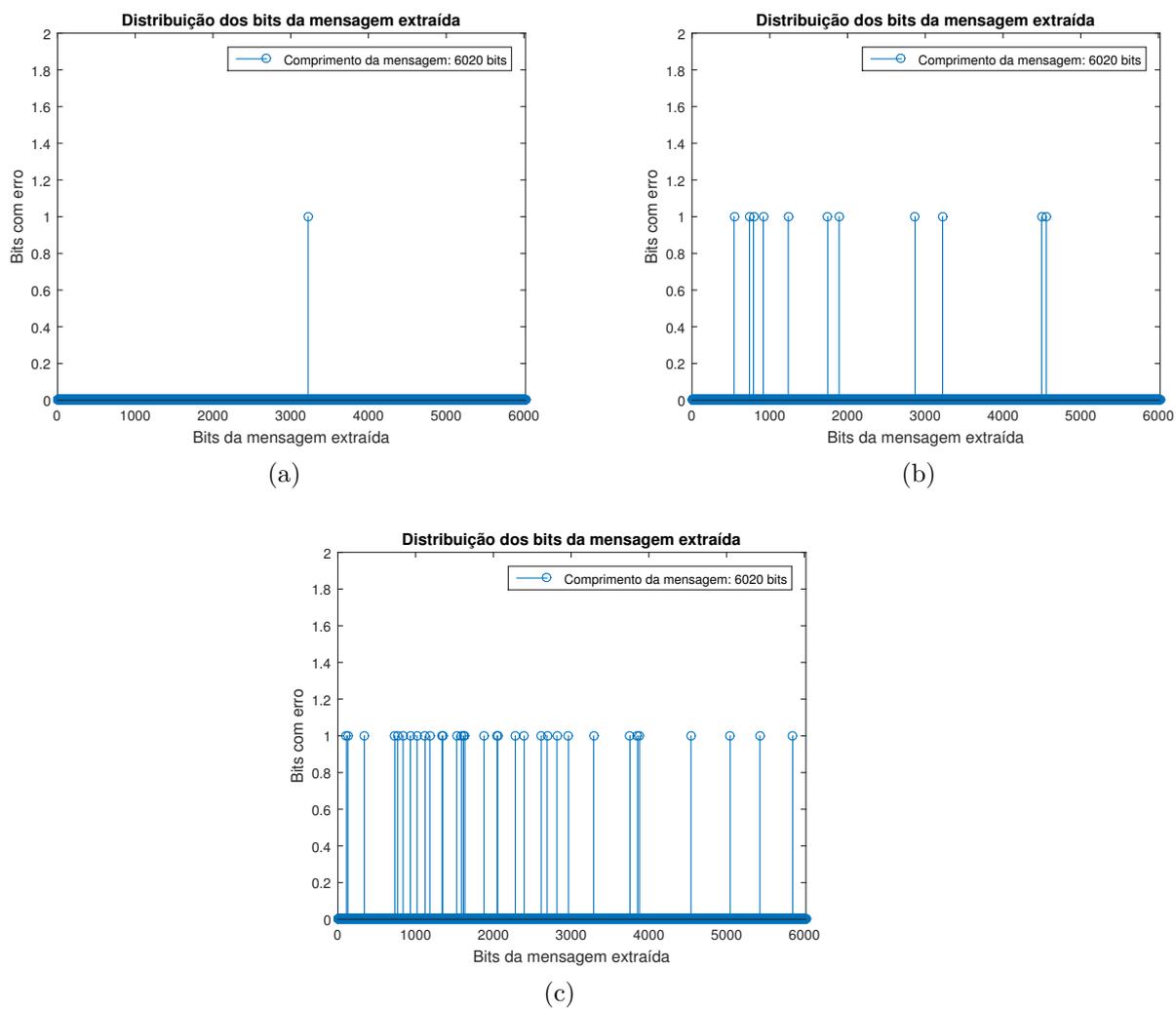


Figura 18 – Representação gráfica das posições dos bits extraídos com erro, ao longo da sequência de bits que representa a partitura em notação ABC: (a) 1 erro, (b) 11 erros e (c) 33 erros.

6 Conclusões

NESTA dissertação, foi proposta uma técnica para ocultação em sinais de áudio de partituras musicais escritas na notação ABC. Embora faça uso de ferramentas e operações conhecidas, como a transformada discreta de Fourier e a modulação de fase, o método introduzido agrega, de forma original, uma funcionalidade interessante às músicas que podem ser obtidas das diversas redes de comunicação: a disponibilização de uma informação que é guardada de maneira “invisível” e que pode ser de importância para determinada classe de usuários, aqueles interessados em tocar a música. Particularidades e ajustes necessários à aplicação da técnica proposta em arquivos de áudio mono e estéreo foram discutidos e ilustrados por meio de exemplos e simulações. Como resultados dessas simulações, vale a pena destacar que, ainda que o arquivo de áudio com a partitura oculta seja submetido à compressão mp3 sob diferentes taxas, vislumbra-se a possibilidade de recuperação sem erros da sequência de bits que representa essa partitura.

6.1 Trabalhos Futuros

Como trabalhos relacionados a esta dissertação e que podem ser desenvolvidos em oportunidades futuras, podem ser elencados os seguintes:

- Avaliar os resultados do método para um número maior de arquivos de áudio, identificando que propriedades desses arquivos podem influenciar nos procedimentos de ocultação e extração dos bits;
- Selecionar métricas para medição da degradação provocada pela ocultação da partitura em determinado arquivo de áudio;
- Incorporar ao método um código corretor de erros, a fim de avaliar a robustez do procedimento de extração de bits a compressões com taxas mais agressivas;
- Converter os códigos desenvolvidos para implementação do método num aplicativo com interface simples e funcionamento automático.

6.2 Artigos Publicados

Os resultados obtidos nesta dissertação foram publicados no artigo listado a seguir e aceito para apresentação num evento nacional:

- Mário C. L. da Cruz Jr., Juliano B. Lima. Uma Técnica para Ocultação de Partituras Musicais em Sinais de Áudio. In: XXXIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (Juiz de Fora, Minas Gerais, Setembro de 2015).

Referências

- AL-OTHMANI, A. Z.; MANAF, A. A.; ZEKI, A. M. A survey on steganography techniques in real time audio signals and evaluation. *International Journal of Computer Science Issues (IJCSI)*, Citeseer, v. 9, 2012. Citado 2 vezes nas páginas 19 e 20.
- ALSALAMI, M. A.; AL-AKAIDI, M. M. Digital audio watermarking: survey. *School of Engineering and Technology, De Montfort University, UK*, 2003. Citado 3 vezes nas páginas 19, 29 e 32.
- BARABÁSI, A.-L.; PÓSFAI, M. *Network Science*. 1st. ed. [S.l.]: Cambridge University Press, 2016. Citado na página 15.
- BENDER, W. et al. Techniques for data hiding. *IBM systems journal*, IBM, v. 35, n. 3.4, p. 313–336, 1996. Citado na página 23.
- BONEY, L.; TEWFIK, A. H.; HAMDY, K. N. Digital watermarks for audio signals. In: IEEE. *Multimedia Computing and Systems, 1996., Proceedings of the Third IEEE International Conference on*. [S.l.], 1996. p. 473–480. Citado na página 32.
- BOSI, M.; GOLDBERG, R. *Introduction to Digital Audio Coding and Standards*. [S.l.]: Kluwer Academic Publisher, 2003. v. 1. Citado 2 vezes nas páginas 33 e 41.
- BRAIN, C.; GREGORY, W. Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. *IEEE Trans Inform Theory*, v. 47, n. 5, p. 1423–1443, 2001. Citado 2 vezes nas páginas 45 e 48.
- BRANDENBURG, K. Mp3 and aac explained. In: AUDIO ENGINEERING SOCIETY. *Audio Engineering Society Conference: 17th International Conference: High-Quality Audio Coding*. [S.l.], 1999. Citado 3 vezes nas páginas 40, 41 e 42.
- BRANDENBURG, K.; POPP, H. Mpeg layer-3. *EBU Technical review*, p. 1–15, 2000. Citado 3 vezes nas páginas 39, 40 e 41.
- CARVALHO, D. F. de; GOULARTE, R. Esteganografia digital: Uma abordagem baseada em vídeos. 2005. Citado na página 22.
- CHAMBERS, J. *ABC Music Notation*. [S.l.: s.n.], 2009. Citado na página 43.
- CHANDRAMOULI, R.; MEMON, N.; RABBANI, M. Digital watermarking. *Encyclopedia of Imaging Science and Technology*, Wiley Online Library, 2002. Citado 2 vezes nas páginas 26 e 27.
- CHAUHAN, S. P. S.; RIZVI, S. A survey: Digital audio watermarking techniques and applications. In: *2013 4th International Conference on Computer and Communication Technology (ICCCCT)*. [S.l.: s.n.], 2013. Citado 4 vezes nas páginas 19, 29, 31 e 32.
- CHOWDHURY, M.; BISWAS, A. *Wireless Communication: Theory and Applications*. 1st. ed. [S.l.]: Cambridge University Press, 2017. Citado na página 15.

- COX, I. et al. *Digital watermarking and steganography*. [S.l.]: Morgan Kaufmann, 2007. Citado na página 27.
- CUNHA, J.; LIMA, M.; BEZERRA, J. Transmissão de informação codificada em imagens digitais utilizando substituição de lsb. *Engenharia de Computação em Revista*, v. 1, n. 5, 2010. Citado na página 21.
- DONG, X.; BOCKO, M. F.; IGNJATOVIC, Z. Data hiding via phase manipulation of audio signals. In: IEEE. *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP'04). IEEE International Conference on*. [S.l.], 2004. v. 5, p. V-377. Citado 2 vezes nas páginas 45 e 47.
- DUTTA, P.; BHATTACHARYYA, D.; KIM, T.-h. Data hiding in audio signal: A review. *International journal of database theory and application*, v. 2, n. 2, p. 1-8, 2009. Citado na página 23.
- FAN, M.; WANG, H. Chaos-based discrete fractional sine transform domain audio watermarking scheme. *Computers & Electrical Engineering*, Elsevier, v. 35, n. 3, p. 506-516, 2009. Citado na página 45.
- FRIDRICH, J.; GOLJAN, M.; DU, R. Reliable detection of lsb steganography in color and grayscale images. In: ACM. *Proceedings of the 2001 workshop on Multimedia and security: new challenges*. [S.l.], 2001. p. 27-30. Citado na página 19.
- FURHT, B.; KIROVSKI, D. *Multimedia Encryption and Authentication Techniques and Applications*. 1st. ed. [S.l.]: Auerbach Publications, 2006. Citado na página 15.
- GHORBANI, S.; AMIRI, S. *Audio Content Security: Attack Analysis on Audio Watermarking*. 1st. ed. [S.l.]: Syngress, 2016. Citado na página 15.
- GOENKA, M. K. V.; PATIL, M. P. K. Overview of audio watermarking techniques. *International Journal of Emerging Technology and Advanced Engineering, ISSN*, Citeseer, p. 2250-2459, 2005. Citado na página 32.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 4th. ed. [S.l.]: Pearson, 2017. Citado na página 15.
- GONZATO, G. *Making Music With Abc Plus: A guide to the notation and its applications*. 2007. Disponível em: <http://http://natura.di.uminho.pt/~jj/ipm/abcplus_en.pdf>. Citado 2 vezes nas páginas 44 e 53.
- GUPTA, V.; BARVE, M. A. A review on image watermarking and its techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*, v. 4, n. 1, p. 92-97, 2014. Citado na página 28.
- HARTUNG, F.; KUTTER, M. Multimedia watermarking techniques. *Proceedings of the IEEE, IEEE*, v. 87, n. 7, p. 1079-1107, 1999. Citado 2 vezes nas páginas 26 e 28.
- HAYKIN, S.; VEEN, B. V. *Sinais e Sistemas*. [S.l.]: Bookman, 2001. v. 1. Citado na página 33.
- HE, X. *Signal Processing, Perceptual Coding and Watermarking of Digital Audio: Advanced Technologies and Models*. 1st. ed. [S.l.]: IGI Global, 2011. Citado na página 15.

- HSU, H. P. *Sinais e Sistemas*. [S.l.]: Bookman, 2008. v. 2. Citado na página 33.
- HUSSAIN, M.; HUSSAIN, M. A survey of image steganography techniques. Citeseer, 2013. Citado na página 21.
- JAYARAM, P.; RANGANATHA, H.; ANUPAMA, H. Information hiding using audio steganography—a survey. *The International Journal of Multimedia & Its Applications (IJMA) Vol*, v. 3, p. 86–96, 2011. Citado 2 vezes nas páginas 19 e 23.
- KAMENOV, A. *Digital Signal Processing for Audio Applications*. 2nd. ed. [S.l.]: Prentice Hall, 2014. Citado na página 15.
- KATZENBEISSER, S.; PETITCOLAS, F. *Information hiding techniques for steganography and digital watermarking*. [S.l.]: Artech house, 2000. Citado na página 27.
- KOSTOPOULOS, H. et al. A digital image watermarking technique using modulated pascal’s triangles. In: *International Conference Signal Processing, Pattern Recognition & Applications*. [S.l.: s.n.], 2003. p. 82–86. Citado na página 28.
- KUROSE, J.; ROSS, K. *Computer Networking: A Top-Down Approach*. 7th. ed. [S.l.]: Pearson, 2016. Citado na página 15.
- LACY, J. et al. On combining watermarking with perceptual coding. In: IEEE. *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*. [S.l.], 1998. v. 6, p. 3725–3728. Citado na página 32.
- LIU, C.-M.; CHANG, W.-W. Audio coding standards. *Multimedia communications: directions and innovations*, Academic Pr, p. 45, 2001. Citado 2 vezes nas páginas 37 e 39.
- MORKEL, T.; ELOFF, J. H.; OLIVIER, M. S. An overview of image steganography. In: ISSA. [S.l.: s.n.], 2005. p. 1–11. Citado 2 vezes nas páginas 20 e 21.
- MORKEN, K. Audio compression in practice. Universitetet i Oslo, p. 103–116, 2007. Citado 3 vezes nas páginas 37, 38 e 40.
- NARAYANA, S.; PRASAD, G. Two new approaches for secured image steganography using cryptographic techniques and type conversions. *Signal & Image Processing: An International Journal (SIPIJ) Vol*, v. 1, n. 2, p. 60–73, 2010. Citado na página 20.
- NEUBAUER, C.; HERRE, J. Digital watermarking and its influence on audio quality. In: AUDIO ENGINEERING SOCIETY. *Audio Engineering Society Convention 105*. [S.l.], 1998. Citado na página 32.
- NEUBAUER, C.; HERRE, J. Audio watermarking of mpeg-2 aac bit streams. In: AUDIO ENGINEERING SOCIETY. *Audio Engineering Society Convention 108*. [S.l.], 2000. Citado na página 32.
- NISSAR, A.; MIR, A. Classification of steganalysis techniques: A study. *Digital Signal Processing*, Elsevier, v. 20, n. 6, p. 1758–1770, 2010. Citado na página 19.
- NOLL, P. Mpeg digital audio coding. *Signal Processing Magazine, IEEE*, IEEE, v. 14, n. 5, p. 59–81, 1997. Citado na página 39.

- NOSRATI, M.; KARIMI, R.; HARIRI, M. Audio steganography: a survey on recent approaches. *World Applied Programming*, v. 2, n. 3, p. 202–205, 2012. Citado 2 vezes nas páginas 19 e 23.
- OLIVEIRA, J. C. de. Audio compression and coding techniques. *Encyclopedia of Multimedia*, Springer US, p. 12–21, 2006. Citado na página 37.
- OPPENHEIM, A. V.; WILLSKY, A. S. *Sinais e Sistemas*. [S.l.]: Pearson, 2010. v. 2. Citado 3 vezes nas páginas 33, 34 e 35.
- OSSEIRAN, A. et al. *5G Mobile and Wireless Communications Technology*. 1st. ed. [S.l.]: Cambridge University Press, 2016. Citado na página 15.
- OWENS, M. A discussion of covert channels and steganography. *SANS institute*, v. 1, p. 1–18, 2002. Citado na página 21.
- PODILCHUK, C. I.; DELP, E. J. Digital watermarking: algorithms and applications. *Signal Processing Magazine, IEEE*, IEEE, v. 18, n. 4, p. 33–46, 2001. Citado na página 26.
- PROVOS, N.; HONEYMAN, P. Hide and seek: An introduction to steganography. *IEEE Security & Privacy*, IEEE, v. 1, n. 3, p. 32–44, 2003. Citado 3 vezes nas páginas 19, 20 e 22.
- SHIH, F. Y. *Multimedia Security: Watermarking, Steganography, and Forensics*. 1st. ed. [S.l.]: CRC Press, 2012. Citado na página 15.
- SINGH, M.; SINGHAL, A.; CHAUDHARY, A. Digital image watermarking techniques: A survey. *International Journal of Computer Science and Telecommunications*, v. 4, n. 6, 2013. Citado na página 29.
- SMITH, S. W. et al. *The scientist and engineer's guide to digital signal processing*. [S.l.]: California Technical Pub. San Diego, 1997. Citado na página 22.
- STALLINGS, W. *Data and Computer Communications*. 10th. ed. [S.l.]: Pearson, 2013. Citado na página 15.
- TEKALP, A. M. *Digital Video Processing*. 2nd. ed. [S.l.]: Prentice Hall, 2015. Citado na página 15.
- ZHAO, H.; SHEN, D.-s. A new semi-fragile watermarking for audio authentication. In: IEEE. *Artificial Intelligence and Computational Intelligence, 2009. AICI'09. International Conference on*. [S.l.], 2009. v. 3, p. 299–302. Citado na página 26.

Apêndices

APÊNDICE A – Partituras Utilizadas nas Simulações

Neste apêndice, são apresentadas as partituras das músicas utilizadas para demonstração do método de inserção proposto. Nas figuras que se seguem, essas partituras são apresentadas tanto na notação ABC quanto na notação musical usual.

A.1 Faixa 1 - *All my loving*

A notação apresentada na Figura 19 inclui somente a partitura propriamente dita, não sendo utilizada a parte de formatação da partitura. Assim, tem-se basicamente a letra e as notas que compoem a canção. A partitura correspondente na notação musical usual é apresentada na Figura 20.

```
X:02
T: All my loving
T: From Album 'With The Beatles' 1964 UK
O: Lennon/McCartney
M:4/4
L:1/4
Q:1/4=120
K: E
%
A G |: F2 G A | B c2 d | e2 e d | c A2 c |
w: Close your eyes and I'll kiss you, to-mor-row I'll miss you.
Re-
w: |tend that I'm kiss-ing the lips I am miss-ing and
c2 c B | A2 F E | F4- | E z A G |
w: mem-ber I'll al-ways be true._ And then
w: hope that my dreams will come true._ And then
F2 G A | B2 c d | e2 e d | c G-c/2c/2 |
w: when I'm a-way I'll write home ev-ery day,_ and I'll
w: while I'm a-way I'll write home ev-rey day,_ and I'll
c2 c B | A G2 F | E4- |[1 E z A G :|| [2 E z e d |
w: send all my lov-ing to you._ I'll pre - All my
w: send all my lov-ing to you._
c G2> | G2 F E F | G4- | G z e d |
w: lov-ing, I will send to you._ All my
c G2> | G2 F E E | E4- | E z z2 ||
w: lov-ing, dar-ling I'll be true._
```

Figura 19 – Partitura da canção *All my loving* na notação ABC.

All my loving
From Album 'With The Beatles' 1964 UK
Lennon/McCartney

♩ = 120

Close your eyes and I'll kiss you, to - - mor - row I'll miss you. Re -
 tend that I'm kiss - ing the lips I am miss - ing and
 mem - - ber I'll al - - ways be true. And then
 hope that my dreams will come true. And then
 when I'm a - - way I'll write home ev - ery day, and I'll
 while I'm a - - way I'll write home ev - ery day, and I'll
 send all my lov - ing to you. I'll pre - - - All my
 send all my lov - ing to you.
 lov - ing, I will send to you. All my
 lov - - ing, dar - - ling I'll be true.

Figura 20 – Partitura da canção *All my loving* na notação usual.

A.2 Faixa 2 - *Here, there and everywhere*

A notação apresentada na Figura 21 apresenta uma escrita simplificada da notação ABC, omitindo detalhes referentes à formatação da partitura impressa. Assim, o arquivo produzido possui formatação padronizada. Além disso, neste exemplo, optou-se por omitir a letra da canção. A partitura correspondente na notação musical usual é apresentada na Figura 22.

A.3 Faixa 3 - *Yesterday*

Diferentemente do que foi utilizado nas notações das partituras das duas primeiras canções, na escrita a seguir, utiliza-se todo o detalhamento referente à formatação do arquivo impresso, bem como a adição da letra da canção. Assim, o texto obtido pelo uso da notação ABC é mais extenso e, para melhor visualização, é apresentado em duas figuras, a 23 e a 24; a primeira contém uma série de linhas relacionadas à formatação e a segunda contempla a partitura e a letra propriamente ditas. A partitura correspondente na notação musical usual é apresentada na Figura 25.

```

X:01
T: Here, There and Everywhere
T: From Album 'Revolver' 1966 UK
O: Lennon/McCartney
M:4/4
L:1/4
Q:1/4=90
K: G
%
z/ D/G/B/ d/d/ d | z/ =F/ _B/d/ =f/_e/d/d/- | d2 z2 ||
|: G4 | B/d/a/g/- g/c/e/d- | d4 | B/c/d/e/- e/G/A/B/- |
B/B/B/B/-B z |
B/c/B/B/-A A/G/- | G/A/B/f/- f/ e A/- | [1 A2> z :|2 A2 z/2
A/2_B/2c/2 |
|: d/2=f/2 d2 c/2_B/2 | c/d/_e/c/ A/A_B/c/ | d/_B/ G2> |
cd/_e/c/ A/A=B/c/ | d/ =B G/-G2 || B/d/a/g/- g/c/e/d/> | d8 |
B/c/d/e/- e/G/A/B/- | B/B/B/B/- B z | B/c/B/B/- B A/G/- |
G/A/B/f/-f e A/-
|[1 A2 z/ A/_B/c :|2 A2 z D/G/G/ ]| G2 z z/2 B/ |
d B/4-A/4G/2-G2| G2 G> B/2 | d f/g/-g2 | G4 ||

```

Figura 21 – Partitura da canção *Here, there and everywhere* na notação ABC.

Here, There and Everywhere
From Album 'Revolver' 1966 UK

Lennon/McCartney

$\text{♩} = 90$

Figura 22 – Partitura da canção *Here, there and everywhere* na notação usual.

```

%%pageheight 29.7cm
%%pagewidth 21cm
%%topmargin 1cm
%%botmargin 1cm
%%leftmargin 1cm
%%rightmargin 1cm
%%topspace 2cm %%titlespace 2cm %%subtitlespace 0.3cm
%%composerspace 0.5cm %%musicspace 0.5cm %%vocalspace 1.5cm
%%sysstaffsep 1cm %%staffsep 3cm %%titlefont Times-Bold 30
%%subtitlefont Times-Bold 16
%%composerfont Times-Roman 12
%%vocalfont Helvetica 14 %%gchordfont Times-Bold 14 %%measurebox
true %%measurenb 0 %%exprabove true %%scale 0.65

```

Figura 23 – Primeira parte da partitura da canção *Yesterday* na notação ABC.

```

% X:02
T: Yesterday
T: From Album 'Help!' 1965 UK
O: Lennon/McCartney vocal by Paul McCartney
M:4/4
L:1/4
Q:1/4=90
K: F
%
G/2F/2 F2> | z2 A/2=B/2 ċ/2d/2e/2f/2 | e> d d2 |
w: Yes-ter-day, All my troub-les seemed so far a-way
z d/2d/2 c/2B/2A/2G/2 | B A/2A/2-A G | F A/2G/2-G D |
w: Now it looks as though they're here to stay Oh, I be-lieve in
F A/2A/2-A2 | G/2F/2 F2> | z2 A/2=B/2 ċ/2d/2e/2f/2 | e> d d2 |
w: yes-ter-day. Sud-den-ly, I'm not half the man I used to be,
z d/2d/2 c/2B/2A/2G/2 | B A/2A/2-A G | F A/2G/2-G D | F
A/2A/2-A2 |
w: There's a sha-dow han-ging o-ver me, Oh, yes-ter-day came
sud-den-ly.
A2 A2 | d e f e/2d/2 | e> d c d | A4 | A2 A2 |
w: Why she had to go I don't know. She would not say, I said
d e f e/2d/2 | e> d c e | (f c B A) | G/2F/2 F2> |
w: Some-thing wrong, Now I long for yes-ter-day. |Yes-ter-day,
z2 A/2=B/2 ċ/2d/2e/2f/2 | e> d d2 | z d/2d/2 c/2B/2A/2G/2 | B
A/2A/2-A G |
w: Love was such an ea-sy game to play, Now I need a place to
hide a-way, Oh,
F A/2G/2-G D | F A/2A/2-A2 | F A G D | F A/2A/2-A2 ||
w: I be-lieve in yesterday. Mm mm mm mm mm mm mm

```

Figura 24 – Segunda parte da partitura da canção *Yesterday* na notação ABC.

Yesterday
From Album 'Help!' 1965 UK

Lennon/McCartney vocal by Paul McCartney

$\text{♩} = 90$

Yes - ter - day, All my trou - ble - seemed so far a - - way

Now it looks as though they're here to stay Oh, I be - - lieve in

yes - - ter - day. Sud - - den - ly, I'm not half the man I used to be,

There's a sha - dow han - ging o - ver me, Oh, yes - - ter - day came sud - den - - ly.

Why she had to go I don't know. She would not say, I said

Some - thing wrong, Now I long for yes - - ter - - day. Yes - ter - day,

Love was such an ea - sy game to play, Now I need a place to hide a - way, Oh,

I be - lieve in yesterday. Mm mm mm mm mm mm mm

Figura 25 – Partitura da canção *Yesterday* na notação usual.