



# **IMPLEMENTAÇÃO DE SISTEMAS DE CONTROLE ATIVO DE RUÍDO ACÚSTICO PARA AMBIENTES AUTOMOTIVOS**

**Trabalho de Conclusão de Curso**

**Engenharia da Computação**

**Gabriel Alves de Lima**

**Orientador: Prof. Carmelo José Albanez Bastos Filho**

**Co-orientador: Rodrigo de Paula Monteiro**

**Universidade de Pernambuco  
Escola Politécnica de Pernambuco  
Graduação em Engenharia de Computação**

**GABRIEL ALVES DE LIMA**

**IMPLEMENTAÇÃO DE SISTEMAS DE  
CONTROLE ATIVO DE RUÍDO  
ACÚSTICO PARA AMBIENTES  
AUTOMOTIVOS**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

**Recife, maio de 2018.**

### MONOGRAFIA DE FINAL DE CURSO

#### Avaliação Final (para o presidente da banca)\*

No dia 22 de maio de 2018, às 14:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente **GABRIEL ALVES DE LIMA**, orientado pelo professor **Carmelo José Albanez Bastos Filho**, sob título **IMPLEMENTAÇÃO DE SISTEMAS DE CONTROLE ATIVO DE RUÍDO ACÚSTICO PARA AMBIENTES AUTOMOTIVOS**, a banca composta pelos professores:

**Sérgio Campello Oliveira**

**Carmelo José Albanez Bastos Filho**

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

Aprovada       Aprovada com Restrições\*       Reprovada

e foi-lhe atribuída nota: 10,0 ( Dez )

\*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 07 dias para entrega da versão final da monografia a contar da data deste documento.

\_\_\_\_\_  
**SÉRGIO CAMPELLO OLIVEIRA**

\_\_\_\_\_  
**CARMELO JOSÉ ALBANEZ BASTOS FILHO**

\* Este documento deverá ser encadernado juntamente com a monografia em versão final.

*O conhecimento é uma ferramenta, e como todas as ferramentas, o seu impacto está nas mãos de quem o usa. - **Dan Brown***

# Agradecimentos

Primeiramente, a Deus por tudo. Também à minha família, que, de perto, me acompanhou durante todo o meu trajeto e tenho certeza de que nunca vão me abandonar. Aos amigos, Mariane Magalhães, Ruamberg Vasconcelos, Petrus Vinicius e Mariana Lira, pois sem eles eu não teria chegado até aqui.

Ao corpo docente da Graduação em Engenharia da Computação da POLI/UPE, que incansavelmente deram suporte para a minha formação. E, em especial, aos professores Carmelo Bastos, Sérgio Campello, José Paulo Gonçalves, Sérgio Murilo Maciel, Mêuser Valença, Bruno Fernandes, Daniel Ribeiro, Byron Dantas e Diego Rátiva, que direta ou indiretamente contribuíram com este trabalho.

Também a colegas que surgiram durante toda esta trajetória, Jacinaldo Medeiros, Daniel Cardoso, Hilton Lima, Cristóvão Rufino, Kewin Lima, Maria Clara Ferraz e tantos outros que fazem parte de uma lista sem fim.

Por fim, novamente ao professor Carmelo José Albanez Bastos Filho e a Rodrigo de Paula Monteiro, meus orientadores, por me guiarem em meio a tantos desafios.

A todos estes, eu expresso aqui o meu mais profundo agradecimento.

# Resumo

A exposição contínua a ruídos acústicos tende a prejudicar a saúde física e mental de um indivíduo ao longo do tempo, principalmente em grandes centros urbanos, que apresentam grandes quantidades de automóveis trafegando diariamente. Sabendo disto, este trabalho teve como objetivo desenvolver um sistema de controle ativo capaz de atenuar os ruídos na parte interna de um automóvel e descrever os problemas na implementação deste tipo de artefato e suas respectivas soluções. Foi visto que é inviável executar algoritmos desta natureza em sistemas de uso geral, sendo assim necessário um equipamento específico. A solução proposta neste trabalho utilizou o processador TMS320C5515 para execução do algoritmo FBFxLMS. Nesta implementação, foram utilizadas listas circulares e operações matemáticas em ponto fixo para melhoria da performance do sistema de modo a alcançar a atenuação de ruídos complexos e variáveis, tais como os gerados pelo sistema motor de um automóvel. O protótipo resultante deste trabalho é capaz de controlar este tipo de ruído com redução da frequência principal em cerca de 10 vezes.

**Palavras-Chave:** Controle Ativo de Ruído, Filtros adaptativos, DSP, Sistema de uso geral, FBFxLMS, antirruído.

# Abstract

*Continuous exposure to acoustic noise tends to harm physical and mental health of an individual over time, especially in large urban centers, which presents large quantities of cars traveling on a daily basis. Then, this work had as objective to develop an active control system able to attenuate the noise inside of a car and to describe the problems that occur when implementing this type of artifact and their respective solutions. It has been seen that it is not feasible to perform such algorithms using general purpose systems, therefore, a specific equipment is needed. The solution proposed in this work used the processor TMS230C5515 to execute the FBFxLMS algorithm. In this implementation, circular lists and fixed point arithmetic operations were used to improve the performance in order to achieve complex and variable noise attenuation, such as those generated by the engine. The resulting prototype of this work is able to control this type of noise reducing the main frequency by about 10 times.*

**Keywords:** *Active noise control, adaptive filters, DSP, general-purpose system, FBFxLMS, antinnoise.*

# Sumário

<b>Capítulo 1</b>	
<b>Introdução</b>	<b>14</b>
Motivação e caracterização do problema	14
Objetivos e metas	15
Objetivos específicos	16
Estrutura da monografia	16
<b>Capítulo 2</b>	
<b>Referencial teórico</b>	<b>17</b>
Filtros adaptativos	17
Filtros de resposta finita e infinita ao impulso	17
Ordem do filtro	18
Taxa de aprendizagem	18
Algoritmos de adaptação dos coeficientes	19
Método dos mínimos quadrados (LMS)	19
Mínimos quadrados com fator de esquecimento (Leaky LMS)	20
Controle ativo de ruído	20
Estrutura de um sistema ANC	23
Estrutura feedforward	23
Estrutura feedback	23
Elementos de um sistema de controle ativo de ruído	23
Caminhos primário e secundário	24
Feedback acústico	24
Região de atenuação	24
Compensação dos efeitos de feedback acústico e caminho secundário	25
<b>Capítulo 3</b>	
<b>Implementação em um sistema de uso geral</b>	<b>26</b>
Configuração e ferramentas do sistema	26
Estrutura	26
Sensores	27
Unidade de processamento	27
Atuador	28

Implementação	28
Configuração de acesso ao driver de áudio	29
Inicialização dos filtros	30
Estimativa dos efeitos de caminho secundário e feedback acústico	30
Execução	31
Resultados	31
<b>Capítulo 4</b>	
<b>Implementação em um sistema DSP</b>	<b>36</b>
Configuração e ferramentas do sistema	36
Implementação	36
Configuração de acesso ao áudio	36
Primeira implementação	37
Operações em ponto fixo	38
Listas circulares	39
Implementação final	41
Resultados	41
<b>Capítulo 5</b>	
<b>Conclusões e trabalhos futuros</b>	<b>47</b>
<b>Bibliografia</b>	<b>49</b>
<b>Anexo A</b>	
<b>Números em ponto fixo</b>	<b>51</b>

# Índice de Figuras

<b>Figura 1.</b> Diagrama de bloco das estruturas FIR (1) e IIR (2)	18
<b>Figura 2.</b> Diagrama de blocos de um filtro FIR adaptativo	19
<b>Figura 3.</b> Interação entre ondas senoidais de 100Hz exatamente em fase	21
<b>Figura 4.</b> Interação entre ondas senoidais de 100Hz exatamente fora de fase	21
<b>Figura 5.</b> Interação entre duas ondas senoidais de 100Hz defasadas em $162^\circ$	22
<b>Figura 6.</b> Diagrama de blocos de um sistema ANC	25
<b>Figura 7.</b> Diagrama de blocos do FBFxLMS, um sistema ANC	28
<b>Figura 8.</b> Diagrama de blocos da estimativa dos efeitos de caminho secundário <i>feedback</i> acústico	31
<b>Figura 9.</b> Resultado da simulação para $k = 1$	33
<b>Figura 10.</b> Resultado da simulação para $k = 2$	34
<b>Figura 11.</b> Resultado da simulação para $k = 3$	34
<b>Figura 12.</b> Resultado da simulação para $k = 4$	35
<b>Figura 13.</b> Representação do Q15	39
<b>Figura 14.</b> Fluxo das amostras através da conversão para ponto fixo	39
<b>Figura 15.</b> <i>Shifting</i> em vetores	40
<b>Figura 16.</b> Lista circular após a inserção de quatro valores ( $V_1$ , $V_2$ , $V_3$ e $V_4$ , respectivamente), mostrando os ponteiros de início e fim	40
<b>Figura 17.</b> <i>Setup</i> de experimentação	42
<b>Figura 18.</b> Ruído senoidal de 133 Hz com o sistema desligado	43
<b>Figura 19.</b> Ruído senoidal de 133 Hz com o sistema ligado	43
<b>Figura 20.</b> Ruído gerado pelo sistema motor de um automóvel a 4000 rpm com o sistema desligado	44

- Figura 21.** Ruído gerado pelo sistema motor de um automóvel a 4000 rpm com o sistema ligado 45
- Figura 22.** Ruído gerado pelo sistema motor de um automóvel acelerando de 3000 rpm para 4000 rpm com o sistema desligado 45
- Figura 23.** Ruído gerado pelo sistema motor de um automóvel acelerando de 3000 rpm para 4000 rpm com o sistema ligado 46
- Figura 24.** Resposta do sistema ao ser ligado em um momento aleatório 46

# Índice de Tabelas

**Algoritmo 1.** Implementação do FBFxLMS utilizando o ALSA

29

# Tabela de Símbolos e Siglas

ALSA – *Advanced Linux Sound Architecture* (Arquitetura avançada de som para Linux)

ANC – *Active Noise Control* (Controle ativo de ruído)

DDR – *Double Data Rate* (Taxa de dados dupla)

DSP – *Digital Signal Processor* (Processador digital de sinais)

EPA – *Environmental Protection Agency* (Agência de proteção ambiental)

EVM – *Evaluation Module* (Módulo de validação)

FBFxLMS – *Feedback Filtered-x Least Mean Squares*

FFT – *Fast Fourier Transform* (Transformada Rápida de Fourier)

FIR – *Finite Impulse Response* (Resposta finita ao impulso)]

FPU – *Floating Point Unit* (Unidade de ponto flutuante)

FxLMS – *Filtered-x Least Mean Squares*

IIR – *Infinite Impulse Response* (Resposta infinita ao impulso)

LMS – *Least Mean Squares* (Mínimo da média quadrática)

NLMS – *Normalized Least Mean Squares* (Mínimo da média quadrática normalizado)

Q15 – Número em ponto fixo com 15 bits para a parte fracionária

RAM – *Random Access Memory* (Memória de acesso aleatório)

# Capítulo 1

## Introdução

Este capítulo se encontra dividido em três seções. Primeiramente, é discutido o problema que motivou este trabalho. Em seguida, são expostos os objetivos principais. A estrutura dos demais capítulos, por fim, também será apresentada.

### 1.1 Motivação e caracterização do problema

Ruído acústico é definido, segundo a Agência de Proteção Ambiental dos Estados Unidos (EPA, do inglês *Environmental Protection Agency*), como qualquer som indesejado ou que cause algum tipo de perturbação [1]. O som não tem cheiro nem mesmo pode ser visto e, por isso, este muitas vezes passa despercebido, mas, especialmente em centros metropolitanos, a presença de ruído é constante. É apontado cientificamente que a exposição prolongada a ruídos pode levar ao esgotamento físico e a distúrbios químicos, metabólicos e mecânicos do sistema auditivo, podendo ser a causa de diversos outros problemas de saúde [2].

No dia a dia dos centros urbanos, a emissão de ruídos é advinda principalmente da utilização de automóveis [3]. Usuários destes estão continuamente submetidos a sinais sonoros indesejados diversos, como o do motor a combustão, do atrito com a superfície do asfalto, o ruído proveniente de outros carros, etc. Visando resolver este problema, construtoras de automóveis empregam forte esforço para reduzir o ruído acústico no ambiente interno do veículo, tanto como forma de prover para seus usuários uma experiência de conforto melhorada, quanto para cumprir leis instituídas para manter a saúde da população [3]. O que torna o problema ainda mais interessante na fase de projeto de um automóvel é o fato de que a tendência da produção automobilística aponta para a produção de veículos mais potentes e mais leves. Isto torna ainda mais impactante aos usuários o ruído gerado, principalmente pelo sistema motor.

Pode-se definir controle de ruído como estudo e aplicação de métodos para redução de ruído em um espaço específico. Quanto aos métodos aplicáveis a redução de ruído acústico, existem duas grandes classes: os métodos passivos e ativos. Estes se diferenciam, basicamente, pela forma como a atenuação do ruído é promovida e no tipo de conhecimento aplicado.

A classe das técnicas de controle de ruído passivo se refere a maneiras de realizar algum tipo de isolamento acústico. Isto é realizado através do emprego de materiais ou invólucros que conseguem absorver parte significativa do som. Desta forma é possível atenuar uma banda de frequência relativamente larga (i.e. de 200 Hz até 10 kHz) [4-7].

Para ruídos mais graves, porém, estas técnicas tendem a perder a eficácia, pois, em baixas frequências, o controle de ruído apresenta problemas específicos, devido a um maior comprimento de onda reduzindo a absorção do som. Para estas situações, é necessário aumentar a quantidade de material para o invólucro [8]. Desta maneira, as soluções passivas, à medida que o ruído a ser controlado se

torna mais grave, tendem a se tornar caras, pesadas e passam a ocupar muito espaço.

Em relação à indústria automobilística, há uma forte inadequação dos métodos passivos com a tendência de construção dos carros por diversos motivos. Primeiramente, os ruídos gerados dentro do ambiente automotivo tendem a ser de baixa frequência, tornando as técnicas passivas inadequadas, visto que o motor funcionando a 4000 rpm apresenta como componente principal um tom de 133 Hz. Por fim, existe forte necessidade de um projeto ser competitivo no mercado de modo que este pode retornar lucro para a construtora. Essa competitividade, entre outros pontos, se dá pelo espaço interno disponível do veículo e pela economia de combustível, logo, aplicar técnicas passivas para controle de ruído dentro de um automóvel pode ser causa de prejuízo financeiro aos investidores.

As técnicas de controle ativo de ruído ou ANC (do inglês *Active Noise Control*), por sua vez, são implementadas através do emprego de sistemas de controle que atuam sobre o ambiente com o objetivo de alcançar o cancelamento parcial ou, idealmente, total do ruído [5]. Estes sistemas podem ser implementações eletroacústicas (reduzindo o ruído diretamente) ou eletromecânicas (reduzindo o ruído através da eliminação de vibração) [5].

Estes sistemas funcionam através da superposição de dois sinais sonoros opostos entre si, causando interferência destrutiva no ruído. Em outras palavras, o sistema ANC deve emitir uma onda sonora de mesma amplitude e fase oposta (ruído secundário) em relação ao sinal indesejado (ruído primário). Estes sistemas normalmente ocupam pouco espaço e alcançam melhores resultados na atenuação de frequências abaixo dos 600 Hz se comparados com métodos passivos [5,9]. Isto faz deste tipo de controle de ruído a melhor opção para aplicação em ambientes automotivos.

Porém, apesar de representar uma boa resposta para problemas de atenuação de sons indesejados, soluções que implementam estas técnicas não são tão frequentes de se encontrar no mercado, exceto em alguns nichos específicos como *headphones*. Neste sentido, este trabalho realizou a proposta de discutir a implementação do algoritmo FBFxLMS (do inglês *Feedback Filtered-x Least Mean Squares*) em sistemas reais e relatar as dificuldades e limitações encontradas no desenvolvimento de um protótipo funcional capaz de atenuar bem ruídos complexos (tais como o ruído do sistema motor de um automóvel). Esta discussão se inicia de forma a mostrar a inviabilidade prática de realização de atenuação de ruído ativo com as técnicas existentes através do uso de um sistema computacional de uso geral (computadores *desktop*, *laptops*, etc), também citar o emprego de estruturas de dados abstratas e algoritmos de modo a obter um sistema eficiente.

## 1.2 Objetivos e metas

Este trabalho teve como objetivo geral delinear os problemas encontrados durante a implementação de um algoritmo de cancelamento ativo de ruídos para sistemas DSP (do inglês, *Digital Signal Processor*) e propor soluções para os mesmos.

### 2.1.1 Objetivos específicos

Em termos específicos, este trabalho buscou:

- Discutir a escolha entre a utilização de um computador de uso geral (pessoal) e um *hardware* específico para processamento de sinais;
- Indicar os efeitos que um ambiente real causa na performance do cancelamento ativo;
- Propor o emprego de algoritmos e estruturas de dados específicos de modo a alcançar uma maior velocidade de execução do processamento do ruído.

## 1.3 Estrutura da monografia

Os demais capítulos estão estruturados da seguinte forma:

- **Capítulo 2 - Referencial teórico:** Nesse capítulo será exposta a teoria que fundamenta todas as etapas realizadas no desenvolvimento do trabalho;
- **Capítulo 3 - Revisão bibliográfica:** Esse expõe trabalhos relacionados ao tema desenvolvidos pela comunidade científica;
- **Capítulo 4 - Implementação em um sistema de uso geral:** Nesse será descrito o trabalho desenvolvido para implementação de um sistema ANC utilizando um sistema de uso geral, com o objetivo de mostrar a inviabilidade destes em realizar este tipo de tarefa;
- **Capítulo 5 - Implementação em um sistema DSP:** Esse capítulo, por sua vez, demonstra as configurações realizadas em um sistema DSP para realização do controle ativo;
- **Capítulo 6 - Resultados obtidos:** Aqui será exposto a análise dos resultados obtidos durante as fases de desenvolvimento do trabalho;
- **Capítulo 7 - Conclusões e trabalhos futuros:** Por fim, serão descritas todas as conclusões obtidas com estes resultados e indicação para trabalhos futuros.

# Capítulo 2

## Referencial teórico

Este capítulo tem como propósito expor de forma sucinta a fundamentação teórica deste trabalho com a finalidade de prover o seu entendimento pleno. As seções deste trabalho estão divididas de acordo com seus temas.

### 2.1 Filtros adaptativos

Filtros adaptativos são filtros que aprendem sobre o sinal que processam através de algum algoritmo implementado. Este aprendizado é armazenado nos coeficientes do filtro, que, por sua vez, são alterados no decorrer do tempo [10]. Em comparação com os filtros digitais tradicionais, os filtros adaptativos são importantes para um número vasto de aplicações no campo de processamento de sinais, pois estes são aplicados quando informações sobre as características do sinal a ser processado não podem ser especificadas *a priori*. Isto torna estes filtros aplicáveis em problemas como cancelamento de eco em sistemas de comunicação, posicionamento de antenas, identificação de sistemas, etc [10].

Sobre estes filtros, existem parâmetros a serem configurados de modo a alterar o seu funcionamento, tais como o algoritmo de adaptação, estrutura do filtro, etc. Nesta seção, serão explanadas estas possíveis configurações para prover melhor entendimento do funcionamento destes.

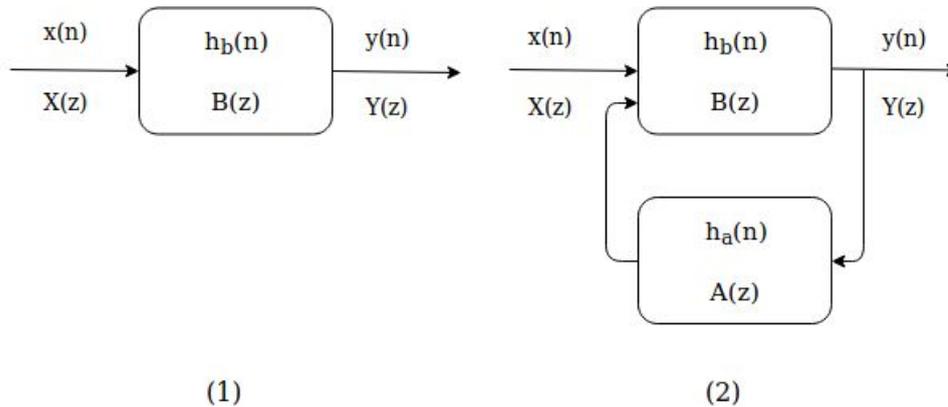
#### 2.1.1 Filtros de resposta finita e infinita ao impulso

Estes podem ser encontrados em várias configurações, porém as implementações mais comuns são em configuração de resposta finita ao impulso (FIR, do inglês *Finite Impulse Response*) ou em resposta infinita ao impulso (IIR, do inglês *Infinite Impulse Response*). A escolha de uma dessas estruturas pode afetar diretamente na qualidade do filtro, pois estas duas formas estão intimamente ligadas com a velocidade de convergência dos coeficientes do filtro e com a complexidade computacional do sistema.

O nome dado a estas formas estruturais diz respeito a sua função de transferência. Os filtros FIR são construídos como sistemas sem realimentação, e, assim, não apresentam polos. A resposta de um filtro nessa estrutura, portanto, é a combinação linear de seus coeficientes. Já os filtros IIR apresentam uma malha de realimentação, resultando em uma forma tal que a sua resposta tem influência de instantes de tempo anteriores ao da observação. Sua função de transferência passa a apresentar polos por causa desta característica. O diagrama de blocos ilustrando estas estruturas está apresentado na Figura 1.

Para controle de ruídos, ambas as estruturas são consideradas viáveis. A estrutura FIR, porém, é a mais comum e simples de implementar na prática [5]. O motivo desta preferência é o fato de que esta estrutura possui apenas zeros a serem ajustados. Logo, o filtros FIR não estão sujeitos às condições de instabilidade que os

filtros IIR estão. Contudo, não é possível afirmar que um filtro FIR sempre será estável, pois isto também depende do algoritmo de ajuste dos coeficientes. Os filtros IIR, por sua vez, por apresentarem pólos ajustáveis, podem demonstrar bons resultados com menos coeficientes, tornando estes sistemas que requerem menos operações aritméticas por ciclo se comparados com filtros FIR.



**Figura 1.** Diagrama de bloco das estruturas FIR (1) e IIR (2)

### 2.1.2 Ordem do filtro

Os filtros digitais, por serem limitados a funcionar num ambiente discreto, dependem da quantidade de seus coeficientes para representar seu comportamento no domínio da frequência de maneira adequada. E por este motivo, determinar a quantidade destes elementos que compõem o filtro é importante, pois isto afeta diretamente a capacidade de representação do modelo [9]. A este parâmetro, dá-se o nome de ordem do filtro, que é diretamente proporcional ao número de coeficientes envolvidos para realizar a filtragem. Esta não deve ser nem tão pequena a ponto de não representar bem o sistema, nem tão grande de modo a tornar o filtro muito complexo. É importante definir a ordem suficientemente alta para que o sistema funcione adequadamente [9].

### 2.1.3 Taxa de aprendizagem $\mu$

Frequentemente chamado de passo (ou *step*), é um parâmetro muito relevante em algoritmos de aprendizagem (tais como redes neurais artificiais) que também é utilizado para parametrizar o ajuste dos coeficientes de um filtro digital adaptativo. Este parâmetro é o que altera a velocidade com a qual os coeficientes tendem a mudar de valor ao longo da operação do sistema. Em outras palavras, valores muito pequenos de  $\mu$  podem resultar em um processo de ajuste lento, enquanto valores muito grandes de  $\mu$  podem encaminhar o sistema para uma situação instável. O valor adequado deste valor está intrinsecamente ligado com os níveis de ruído capturados pelo sistema, pois este serve como balanço para a influência do sinal na adaptação dos coeficientes do filtro [9].

### 2.1.4 Algoritmos de adaptação dos coeficientes

Com o rápido desenvolvimento de sistemas computacionais cada vez mais eficientes, filtros adaptativos cada vez mais robustos passam a ser estruturas viáveis de serem implementadas. Tendo isto em vista, os algoritmos de adaptação dos coeficientes foram bastante estudados por cientistas ao redor do mundo e, como resultado deste esforço, existem muitos algoritmos viáveis quanto a implantação.

A Figura 2 mostra como estes algoritmos podem ser embutidos dentro do processamento de um filtro. É comum considerar como entrada para estes algoritmos algum tipo de informação sobre o sinal a ser processado, de modo que a adaptação dos coeficientes alcance o ponto de convergência, ou seja, um estado alcançável do filtro tal que este consiga processar adequadamente o sinal de entrada (visto na Figura 2 como *Input*) através de sua resposta (visto na Figura 2 como *Output*).

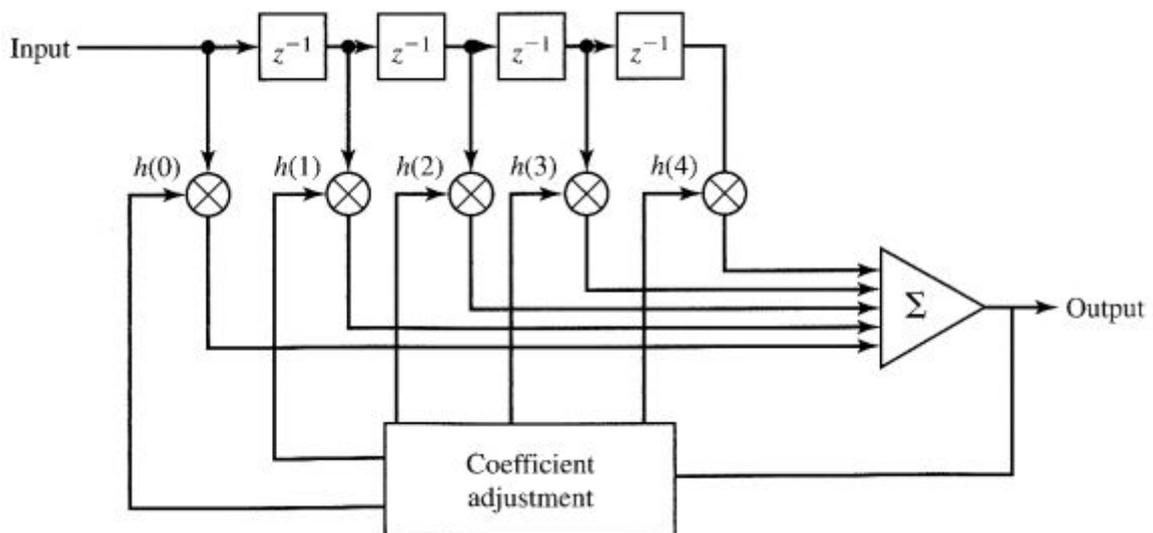


Figura 2. Diagrama de blocos de um filtro FIR adaptativo [10]

### Método dos mínimos quadrados (LMS)

Este método é aplicado em diversas áreas da engenharia para adaptação dos coeficientes de filtros adaptativos. Por ser bastante simples, existem várias outras modificações deste modelo com a finalidade de obter melhores resultados. É vantajoso, portanto, entender este método para conhecer melhor o funcionamento de outros algoritmos mais robustos.

O LMS (do inglês *Least Mean Squares*) é baseado no algoritmo do gradiente descendente. Este é iniciado com coeficientes próximos ou iguais a zero e, para cada iteração do sistema, este algoritmo busca atualizar os parâmetros do filtro com o conhecimento de um sinal de erro. Assim é possível obter o gradiente do erro quadrático médio. O LMS pode ser executado através da Equação (2.1)

$$W_n(t + 1) = W_n(t) - \mu x_n(t)e(t), \quad (2.1)$$

na qual  $W_n(t)$  é o coeficiente  $n$  do filtro num instante de tempo  $t$  (tempo é comumente medido em termos de iterações do sistema);  $x_n(t)$  é a amostra de entrada  $n$  armazenada no vetor regressor  $x$  num instante de tempo  $t$ ;  $e(t)$  é o erro num instante de tempo  $t$  e  $\mu$  é a taxa de aprendizagem. Esta, porém, não é a única abordagem para aplicação em filtros adaptativos. Existem aplicações, por exemplo, que fazem experimentos em sinais utilizando algoritmos genéticos [11].

### Mínimos quadrados com fator de esquecimento (Leaky LMS)

Para este trabalho, considerando o comparativo entre diversas técnicas realizado em [9], outros algoritmos como o NLMS (do inglês *Normalized LMS*) não se tornam interessantes visto que técnicas baseadas no LMS podem fornecer melhores resultados. Porém, tendo em vista a redução de erros numéricos causados pela quantização dos valores e o aumento de robustez do sistema quando exposto a ruídos de amplitude muito grande, uma variação interessante do LMS é o chamado *Leaky LMS* [5]. Este considera um limite para os coeficientes do filtro de modo a limitar o valor da saída. Este é executado segundo a Equação (2.2)

$$W_n(t + 1) = \gamma W_n(t) - \mu x_n(t)e(t), \quad (2.2)$$

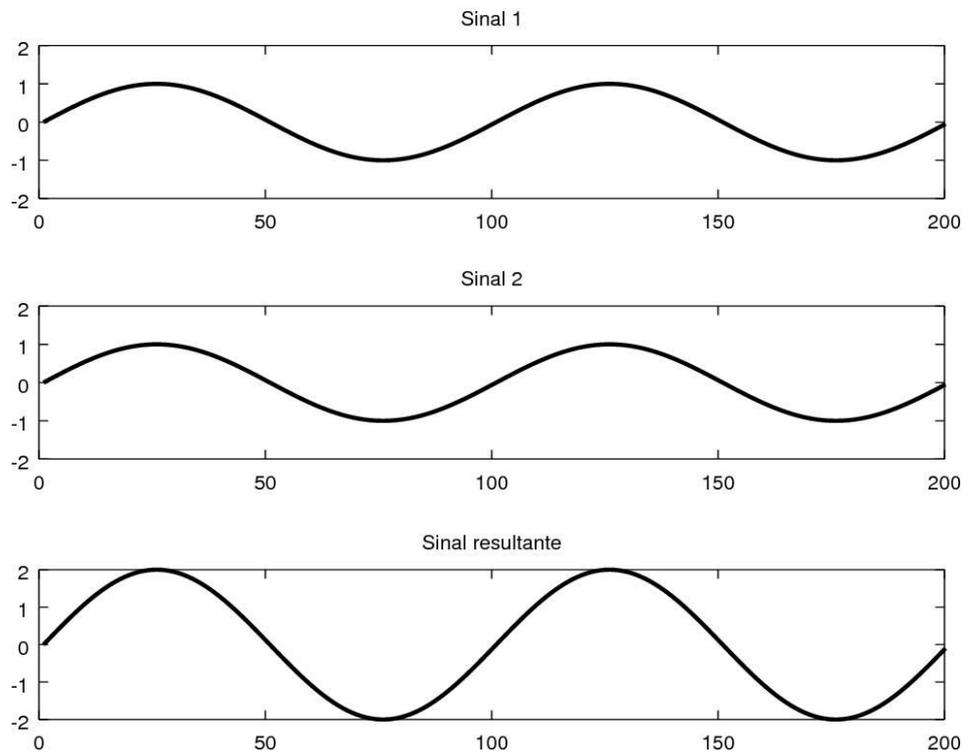
onde  $\gamma$  é o fator de esquecimento (*leakage factor* em inglês), escolhido tal que  $0 < \gamma < 1$ .

## 2.2 Controle ativo de ruído

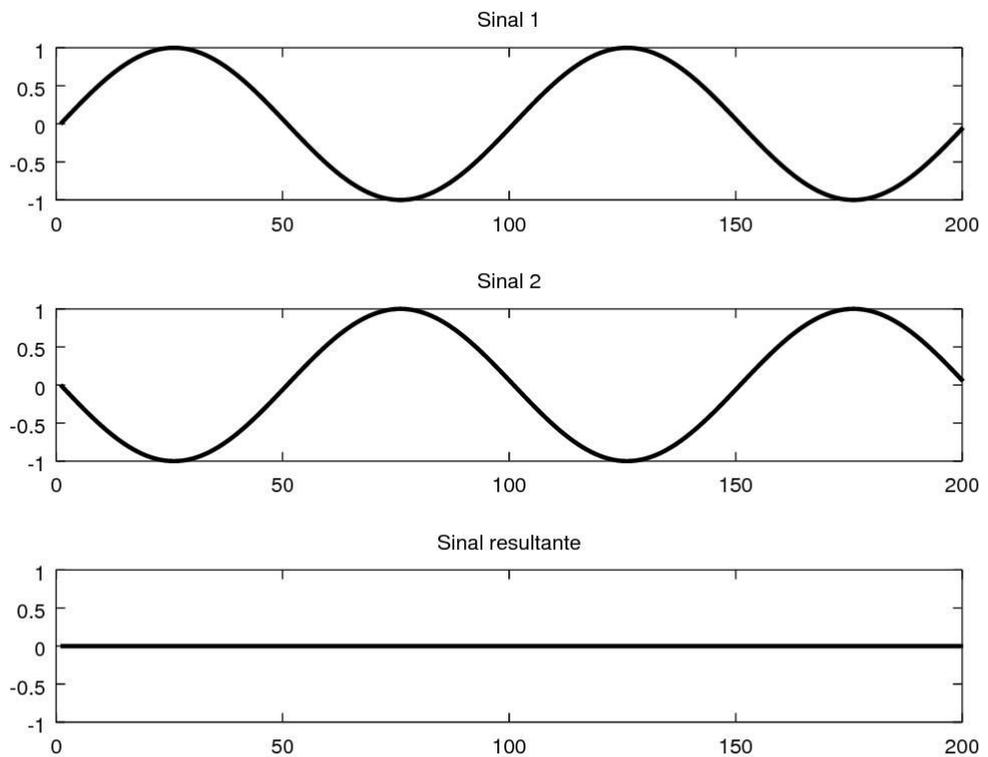
O estudo de controle ativo de ruído busca compreender os diversos cenários de aplicação e projeção de sistemas capazes de atuar sobre um sinal indesejado. Estes projetos podem representar artefatos eletroacústicos, que buscam atuar sobre o ruído diretamente, ou eletromecânicos, os quais funcionam com a eliminação da vibração de materiais. A aplicação destes sistemas depende do resultado da interação entre ondas.

O princípio da superposição de ondas indica que onde a há a manifestação de duas ou mais ondas, o deslocamento de cada ponto do domínio da função resultante é igual a soma do deslocamento no mesmo ponto de suas ondas componentes [12]. Em outras palavras, a interação entre dois ou mais sinais periódicos resulta no somatório destes elementos. Deste princípio, existem dois casos especiais: as interferências construtiva e destrutiva.

O caso de interferência construtiva acontece quando as cristas dos sinais que estão interagindo entre si se combinam. Neste caso, é dito que os sinais estão exatamente em fase, pois a informação de fase destes componentes é a mesma. A Figura 3 representa esta situação, mostrando duas ondas senoidais em fase de mesma amplitude e frequência. O caso contrário acontece quando a crista de um dos sinais interage com o vale de outro sinal. Neste caso, é dito que os sinais estão exatamente fora de fase, pois a informação de fase destes componentes difere de  $180^\circ$ . A Figura 4 exemplifica esta situação, mostrando duas ondas senoidais fora de fase de mesma amplitude e frequência.

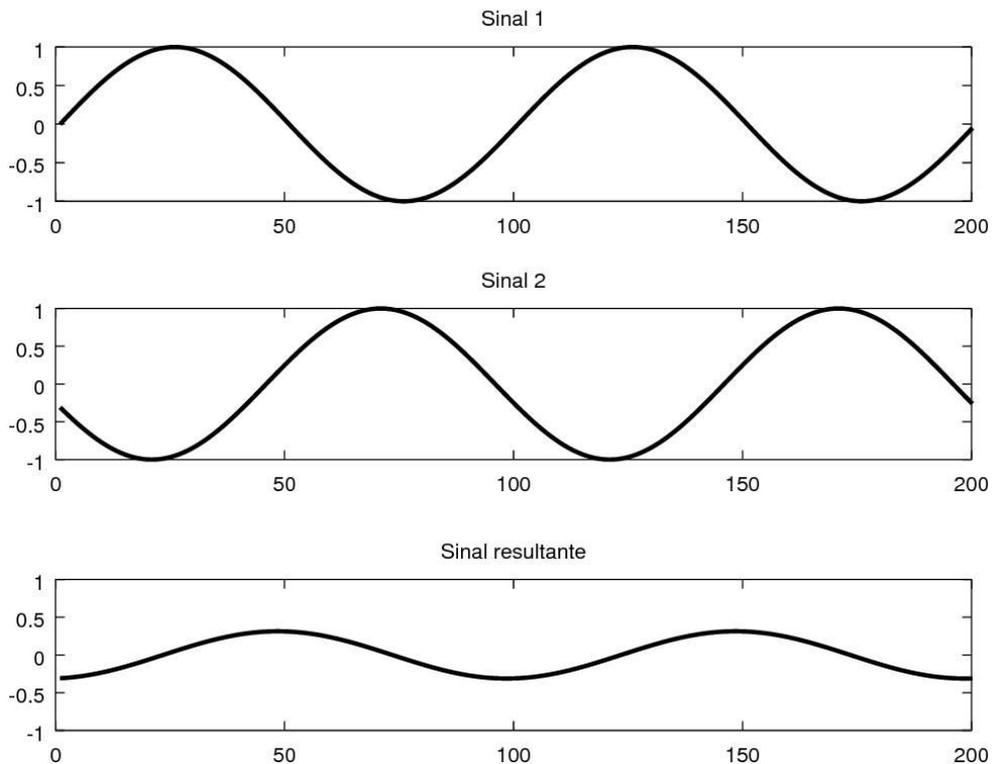


**Figura 3.** Interação entre ondas senoidais de 100Hz exatamente em fase



**Figura 4.** Interação entre ondas senoidais de 100Hz exatamente fora de fase

Os sistemas de controle ativo se baseiam neste último caso, pois este oferece uma modelagem matemática que permite a um algoritmo, atuar destrutivamente sobre um sinal indesejado. Ou seja, seu funcionamento consiste em emitir uma onda de mesma amplitude e frequência que o som indesejado, porém, defasado em exatamente  $180^\circ$  em uma região do espaço. Se for possível emitir esse sinal, então o cancelamento será perfeito. Esta exatidão, no entanto, é improvável de acontecer dado a restrição temporal forte do problema. Por isto, o que os sistemas de controle ativo normalmente alcançam é uma atenuação satisfatória do ruído. Isto pode ser visto na Figura 5.



**Figura 5.** Interação entre duas ondas senoidais de 100Hz defasadas em  $162^\circ$

Uma consequência importante de notar é a prevalência do princípio da conservação de energia, que tem como premissa a constância da quantidade total da energia em um sistema. É fácil de verificar esse princípio no fenômeno de interferência construtiva, pois a potência associada ao sinal resultante desta interação é claramente composta pela soma da energia dos sinais de entrada. Já na demonstração do fenômeno de interferência destrutiva, a energia permanece constante, porém esta é deslocada para algum outro ponto no espaço. Dentro de um ambiente fechado, como um automóvel, é difícil afirmar como essa energia deve se manifestar, mas muito provavelmente este efeito deve provocar vibração a ser absorvida pelo material dos limites físicos do ambiente. Isto deve resultar na conversão desta energia cinética em térmica.

Para a implementação destes sistemas, é muito comum a utilização de filtros adaptativos, pois estes, utilizando um algoritmo adequado, são capazes de encontrar uma configuração que consegue distorcer o sinal primário de forma a alcançar a atenuação. Além disso, este tipo de filtro tem a capacidade de se ajustar

constantemente a um sinal de referência. Portanto, o sistema se torna genérico o suficiente para controlar formas de onda distintas em ambientes variáveis.

### **2.2.1 Estrutura de um sistema ANC**

Como visto anteriormente, algoritmos baseados no LMS para a adaptação dos coeficientes de um filtro se utilizam de uma informação de entrada  $x(t)$  e um sinal de erro  $e(t)$ . Baseado na forma de obter estes sinais, existem duas formas principais de realizar a implementação de um sistema ANC. Estas se chamam *feedback* e *feedforward* e são estas estruturas diferentes entre si pela forma de captar informação sobre o ruído [5].

#### **Estrutura *feedforward***

Os sistemas na configuração *feedforward* são construídos utilizando dois sensores com finalidades distintas. O primeiro deles, chamado de sensor de referência, tem como objetivo captar o ruído com maior proximidade de sua origem, se distanciando da região de atenuação (sinal de referência). Este coleta para o sistema amostras do ruído. O segundo sensor, chamado de sensor de erro, é colocado na região de atenuação, onde este capta o sinal de erro e indica esta informação para o sistema. Estes sinais representam respectivamente as variáveis  $x(t)$  e  $e(t)$ .

O benefício desta estrutura está no fato de que o sistema tem maior quantidade de informação sobre o sinal a ser processado e o ruído é capturado antes de que este chegue na região espacial de atenuação. Isto resulta em um sistema que apresenta melhores resultados do que a estrutura *feedback*, porém com complexidade e custo maiores.

Vale notar que o sensor de referência pode ser substituído por um sensor não acústico, desde que seja possível extrair informação sobre o ruído deste.

#### **Estrutura *feedback***

Os sistemas na configuração *feedback* são construídos utilizando apenas um sensor acústico. Este único sensor é responsável por coletar o sinal de erro e, através deste, realizar uma estimativa do sinal de referência de acordo com o histórico de amostras da saída e informações do caminho secundário. Estes sinais representam respectivamente as variáveis  $e(t)$  e  $x(t)$ .

A vantagem desta estrutura está no fato de que o sistema é mais barato em termos de construção, porém exibe taxas de atenuação piores que sistemas na estrutura *feedforward* [13].

### **2.2.2 Elementos de um sistema de controle ativo de ruído**

É convencional chamar o ruído que se pretende cancelar como ruído primário,  $d(n)$ , que tem como origem a fonte primária. Já o som emitido pelo sistema de cancelamento ativo é comumente chamando de antirruído ou ruído secundário,  $y'(n)$ , que tem como origem a fonte secundária [6]. Ambos os sinais percorrem caminhos distintos até chegarem ao ponto de interação, nomeados respectivamente

de caminho primário  $P(z)$  e secundário  $H(z)$  [6]. Ainda existem o efeito de *feedback* acústico denominado por  $F(z)$  [6]. Estes elementos podem ser modelados como apresentados na Figura 6.

### **Caminhos primário e secundário**

Os caminhos primário e secundário devem ser considerados no modelo, pois ambos distorcem os sinais respectivamente envolvidos. O caminho  $P(z)$  se refere ao espaço compreendido desde a fonte do ruído até a região de atenuação. Por diversos fatores, tais como a presença de corpos sólidos e formato do ambiente, este provoca distorção no som [5]. Isto causa uma alteração no formato de onda do sinal a ser atenuado, portanto, afetando a performance do sistema. Este, porém, não é considerado na implementação do modelo real, pois seu efeito é identificado pelo algoritmo adaptativo empregado.

Já o  $H(z)$  compreende um conjunto de elementos eletrônicos empregados no sistema para emissão do antirruído, tais como cabos e sensores [5]. Estes também distorcem o sinal emitido pelo sistema, pois existem parâmetros eletrônicos como a impedância que influenciam no formato de onda do som, também implicando na capacidade de controle de ruído. Este deve ser compensado no sistema de modo a tornar o algoritmo robusto ao efeito deste caminho.

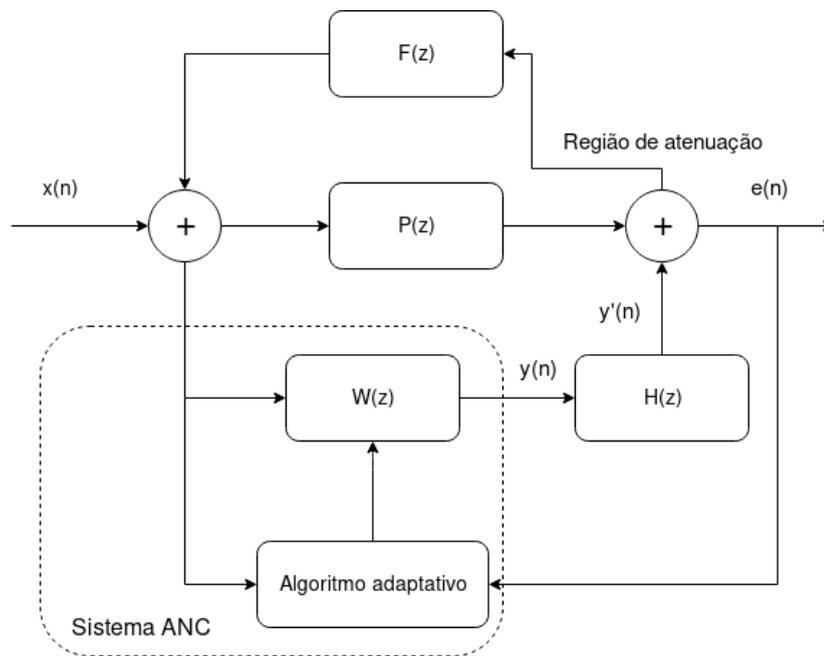
### **Feedback acústico**

Este é um efeito que acontece apenas com a implementação de uma estrutura *feedforward* com sensores acústicos, consistindo na interferência que o ruído secundário causa no sensor de referência [6]. Dependendo da disposição espacial e amplitude dos sinais envolvidos, uma fração do antirruído pode ser adicionada ao sinal de referência, causando dados inválidos para a entrada do sistema. Se este efeito for verificado, deve ser implementada uma compensação de modo a eliminar ou reduzi-lo.

### **Região de atenuação**

É nomeada como região de atenuação uma porção espacial onde ocorre os maiores níveis de redução do ruído. É objetivo de um sistema de controle ativo de ruído fazer com que o espaço destinado ao usuário esteja contido na região de atenuação. Por este motivo, é importante determinar onde é esta região.

Considerando que, como visto, os sistemas ANC alcançam maiores níveis de atenuação à medida que a amplitude do sinal de erro é reduzido, é possível afirmar que o controle de ruído tem como resultado a demonstração de redução deste sinal. Logo, como este é capturado pelo sensor de erro, a região com maior probabilidade de apresentar o maior grau de redução do som indesejado por atuação do sistema é a região nas proximidades desse sensor. Em outros termos, é para esta porção do espaço que o sistema se adapta para realizar a defasagem do ruído primário. Isto implica na produção de um ruído secundário adaptado para as características do ruído a ser cancelado nesta região.



**Figura 6.** Diagrama de blocos de um sistema ANC

### Compensação dos efeitos de *feedback* acústico e caminho secundário

Dado que estes efeitos foram modelados, é possível compensá-los de modo a tornar o sistema robusto. A compensação é realizada por meio da adição de informação ao sinal de entrada, fazendo com que o processo de adaptação dos coeficientes não precise realizar esta compensação *on-line*, ou seja, durante a execução do algoritmo de atenuação.

Para o caminho secundário, é feita a filtragem do sinal de referência utilizando a modelagem deste efeito. Isto acelera a convergência do algoritmo e adiciona robustez a este [6]. Trabalhos realizados por Morgan demonstram que na modelagem do caminho secundário pode constar uma boa disparidade do sistema real e ainda assim apresentar bons resultados [5]. Seus resultados mostram que, dentro de um limite de velocidade de adaptação, o algoritmo com compensação tende a convergir mesmo que a estimativa do caminho secundário apresente diferenças de fase de aproximadamente  $90^\circ$  do efeito real [5].

Já a retribuição do *feedback* acústico ocorre pela subtração do sinal de referência pelo sinal de saída convoluído por sua estimativa, pois este efeito é produzido de forma aditiva da resposta do sistema sobre o sensor de referência. Em outras palavras, a modelagem deste efeito é utilizada para reduzir a realimentação do sistema.

Estas compensações podem ser aplicadas independentemente, dando origem a sistemas de controle ativo de ruídos distintos, tais como o FxLMS (do inglês *Filtered-x Least Mean Squares*) e o FBFxLMS. O primeiro apresenta apenas a retribuição do caminho secundário, enquanto o segundo é composto pela compensação dos dois efeitos.

# Capítulo 3

## Implementação em um sistema de uso geral

A atenuação ativa de ruídos é, como já visto, uma tarefa de forte restrição temporal. A hipótese é de que sistemas de uso geral, como computadores pessoais, *smartphones* e *tablets*, são impróprios para realizar esta função adequadamente por causa da necessidade de defasagem exata ou muito próxima entre os sinais primário e secundário. Neste trabalho foi iniciada a experimentação dos algoritmos de ANC neste tipo de dispositivo. Este processo é adequado à indústria automobilística, pois os sistemas de consumo midiático presente nos automóveis atuais (estes atualmente denominados por sistemas de *infotainment*) são comumente aplicados neste tipo de equipamento.

Neste capítulo será descrito todo processo de implementação de um *software* ANC utilizando filtros adaptativos para um sistema operacional de uso geral. Será exposto também resultados de simulações que mostram a inviabilidade de execução de atenuação ativa através deste tipo de sistema.

### 3.1 Configuração e ferramentas do sistema

Nesta seção, são expostas especificações dos materiais utilizados e escolhas feitas na construção do sistema. Todas as especificações feitas são justificadas.

#### 3.1.1 Estrutura

Como já visto, de acordo com a literatura científica, os sistemas ANC *feedforward* acrescentam custo ao protótipo final, pois estes contam com dois sensores, um de referência e um de erro, em troca de ter uma melhor eficiência se comparados com a estrutura *feedback*. É conhecido que implementar um sistema ANC com o sensor de referência possibilita taxas de atenuação muito maiores do que sistemas sem este sensor [13].

A estrutura *feedforward* assume que a proximidade da fonte do ruído é conhecida e é possível de captar o som indesejado próximo desta origem [5]. Dentro de um automóvel, focando no sistema motor do mesmo, é possível definir espacialmente esta fonte, tornando esta estrutura aplicável dentro de carros.

Portanto, tendo em vista as premissas citadas anteriormente, neste trabalho foi adotada a estrutura *feedforward* com o objetivo de atenuar bem ruídos complexos dentro de um automóvel. Assim, o algoritmo funcionará sempre utilizando sinais de referência (captando amostras nas proximidades da fonte do ruído) e de erro (captando amostras na região de atenuação).

### 3.1.2 Sensores

Foram utilizados dois microfones condensadores Behringer ECM8000 [14] como sensores acústicos, apresentando resposta de frequência na faixa entre 15 Hz e 20 kHz. Não há distinção perceptível entre os dois equipamentos.

Como estes devem ser alimentados por uma fonte *Phantom Power* 48V, foi utilizado uma mesa de mistura Behringer Eurorack UB1002FX [15] como forma de captação do som dos sensores. Deste equipamento, é possível controlar a amplitude dos sinais captados pelos sensores e obter uma resposta estéreo através de um cabo a ser conectado na unidade de processamento. Esta mesa foi configurada de modo a emitir os sinais de erro e referência pelos canais direito e esquerdo, respectivamente.

### 3.1.3 Unidade de processamento

Nesta fase, foi utilizado um computador *desktop* para execução de um algoritmo ANC. O equipamento utilizado para esta finalidade era composto por um processador Intel I5-3570 e memória RAM DDR3 4GB. O sistema operacional utilizado foi o Ubuntu Mate 15.10, que é uma distribuição Linux. O acesso ao sistema de captura e reprodução de áudio foi realizado por meio da biblioteca ALSA, utilizando o pacote *libasound2-dev*, disponível nos repositórios oficiais do módulo gerenciador de programas do sistema operacional. A biblioteca ALSA foi escolhida por fornecer chamadas ao sistema através das linguagens C e C++ e meios de configuração de baixo nível.

A escolha do sistema operacional foi baseada em princípios de performance, licença, pois distribuições Linux são comumente mais baratas em termos de recursos computacionais e muitas delas são *softwares* livres, reduzindo o custo associado ao desenvolvimento. Além disto, este trabalho foi desenvolvido como parte integrante de um projeto *infotainment* para automóveis e este foi desenvolvido em um sistema Linux, logo, para prover integração entre os módulos integrantes do projeto, a escolha deste sistema operacional foi mais adequada. Os demais módulos do projeto não fazem parte do escopo deste trabalho e, portanto, não serão tratados seus detalhes.

Dos algoritmos disponíveis para implementação, é conhecido que o FxLMS é um método eficiente e robusto em relação aos efeitos do caminho secundário [9]. Visto que o *feedback* acústico, porém, é um problema presente nas implementações reais, é importante considerar este efeito para obter um sistema robusto. Portanto, foi escolhido o algoritmo FBFxLMS como técnica de controle ativo. Este algoritmo está representado na Figura 7 no formato de diagrama de blocos, no qual  $F(z)$  é o efeito de *feedback* acústico,  $H(z)$  representa o caminho primário,  $D(n)$  a estimativa do *feedback* acústico usada para compensação deste efeito.  $C(z)$  representa a estimativa do efeito causado pelo caminho secundário usada para compensação do mesmo. Por fim,  $W(z)$  representa o filtro adaptativo usado para gerar o ruído secundário, seguindo o algoritmo LMS para adaptação dos coeficientes utilizando como entrada o sinal  $x'(n)$  e como sinal de erro a entrada  $e(n)$ . Os sinais  $u(n)$  e  $y(n)$  são os ruídos primário e secundário, respectivamente.

### 3.1.4 Atuador

Tendo em vista que o ruído proveniente do sistema motor é predominantemente composto de baixas frequências (abaixo dos 1500 Hz) [9], foi utilizado um alto-falante *subwoofer* Multilaser SP097 [16] para emissão do antirruído. Este equipamento recebe diretamente a saída do sistema ANC através de conexão por cabo.

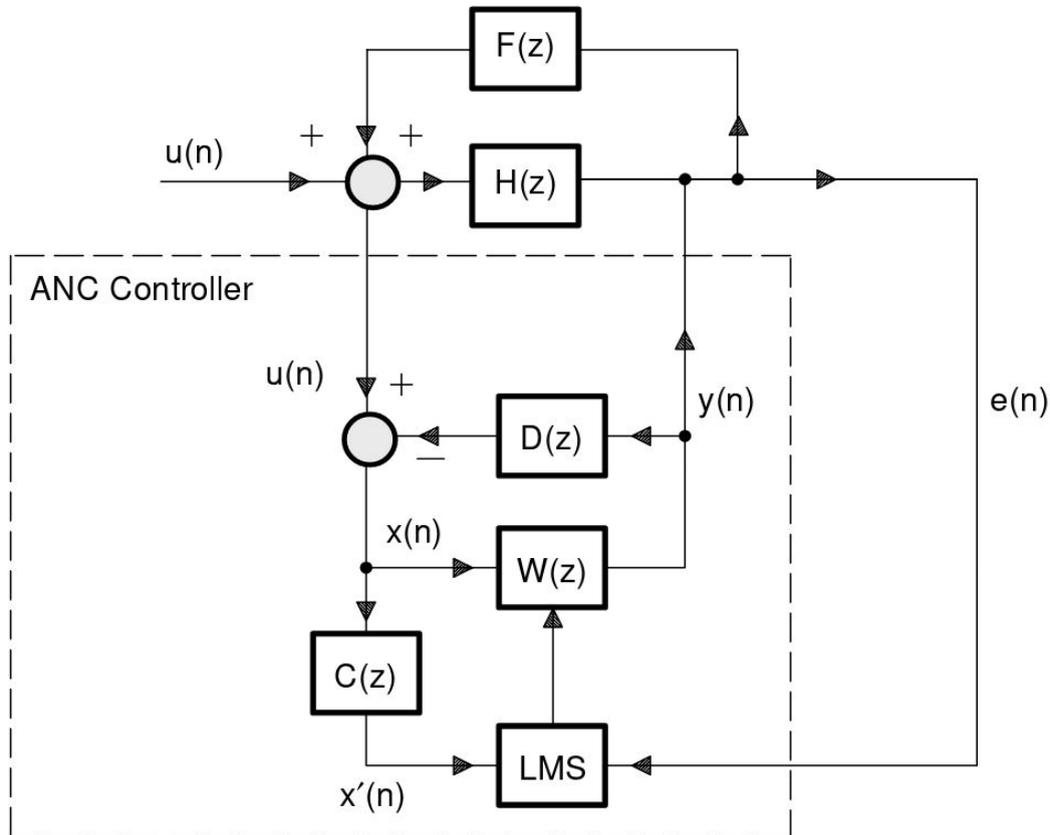


Figura 7. Diagrama de blocos do FBFxLMS, um sistema ANC [6]

## 3.2 Implementação

A implementação do sistema foi realizada segundo o Algoritmo 1. O primeiro ponto a ser tomado são as configurações de acesso ao *driver* de áudio, seguido de inicialização das estruturas computacionais associadas ao processo de filtragem. Por se tratar de uma implementação com compensação dos efeitos de *feedback* acústico e caminho secundário, é necessário realizar uma estimativa destes componentes de forma a considerá-los durante a adaptação dos coeficientes do filtro. Após este passo, o sistema entra em um laço de repetição que executa a captura, processamento e emissão dos sinais primário e secundário, respectivamente. Para o experimento realizado, foi utilizado um laço de repetição infinito para manter o foco da análise especificamente na execução do controle ativo.

### 3.2.1 Configuração de acesso ao *driver* de áudio

A biblioteca ALSA permite o acesso aos dispositivos de áudio através de chamadas ao sistema operacional. Como o núcleo Linux abstrai os periféricos conectados de maneira similar aos arquivos do sistema, as operações com dispositivos de áudio nesta arquitetura se parecem muito com operações de leitura e escrita em arquivos. Assim, através da biblioteca, o sistema fornece ao programa *handlers* que permitem a utilização privilegiada destes componentes. Para cada dispositivo a ser usado, deve ser requisitado um *handler*. Neste trabalho, como o sistema será composto por um único dispositivo de entrada com dois sensores e uma saída, serão necessários dois *handlers*.

Ao fazer a aquisição destes *handlers*, é possível definir uma série de parâmetros, tais como taxa de amostragem e número de canais. Foi definido nesta implementação que seriam utilizados dois canais, levando em consideração as configurações feitas para os sensores descritas na Seção 3.1.2. A taxa de amostragem foi inicialmente definida em 5 kHz baseado no teorema da amostragem de Nyquist-Shannon [9].

Existe, porém, uma peculiaridade ao realizar captura e reprodução de áudio em sistemas de uso geral. Como os sistemas operacionais (incluindo o Linux) trabalham com políticas de preempção para o gerenciamento de processos, não é possível garantir que as chamadas ao sistema vão conseguir obter/reproduzir amostras de som deterministicamente. Ou seja, não é possível garantir que o sinal sonoro de entrada/saída serão formados segundo a taxa de amostragem definida. Por este motivo, as implementações de acesso ao sistema de som (incluindo a ALSA) funcionam com *buffers*, pequenos blocos de memória que armazenam  $n$  amostras de entrada/saída. É possível configurar o valor de  $n$  para cada *handler* em termos de *frames*, que são conjuntos de uma amostra para cada canal. Para este trabalho, portanto, cada *frame* será composto por duas amostras. Assim, em um curto intervalo de tempo, um processo com acesso ao driver de áudio será capaz de obter  $p$  *frames* de um intervalo de tempo passado e estabelecer  $q$  *frames* a serem reproduzidos a partir de um instante de tempo posterior. Inicialmente, foi definido um tamanho de 16 *frames*, tanto para o *buffer* de entrada quanto para o de saída.

---

#### Algoritmo 1: Implementação do FBFxLMS utilizando o ALSA

---

1. Definição dos parâmetros de configuração do *driver* de áudio (e.g. taxa de amostragem, quantidade de canais, etc)
  2. Inicialização dos filtros envolvidos (e.g. passo de aprendizagem, ordem, etc)
  3. Execução de estimativa dos efeitos de caminho secundário e *feedback* acústico
  4. Enquanto o sistema estiver ativo, faça:
    5.  $saidaTotal = \text{array vazio de } n \text{ frames}$
    6. Captura de  $n$  *frames*.
    7. Para cada *frame* capturado, faça:
      8.  $saida = \text{Execução do FBFxLMS para este frame}$
      9. Insere *saida* em *saidaTotal*
    10. Envia *saidaTotal* para a emissão de áudio
-

### 3.2.2 Inicialização dos filtros

São três os filtros envolvidos no processo de controle ativo para a implementação do FBFxLMS. O primeiro deles é o filtro adaptativo responsável pela geração do ruído secundário, sendo os coeficientes deste atualizados durante a execução da atenuação de som indesejado (treinamento *on-line*). Os outros dois filtros foram utilizados para compensação dos efeitos de caminho secundário e *feedback* acústico, respectivamente, sendo os coeficientes deste atualizados antes da atenuação de som indesejado. Ou seja, nestes dois filtros, não haverá adaptação dos coeficientes durante a execução do controle ativo (treinamento *off-line*).

Para todos os filtros, foram empregados os mesmos parâmetros, sendo estes o passo de aprendizagem, ordem dos filtros e fator de esquecimento. Estes parâmetros, no entanto, não tiveram nenhuma definição quanto aos valores, que foram alterados durante a fase de coleta dos resultados do algoritmo. Os coeficientes dos filtros foram inicializados como zero, visto que esta definição evita uma possível instabilidade causada por coeficientes aleatórios [9].

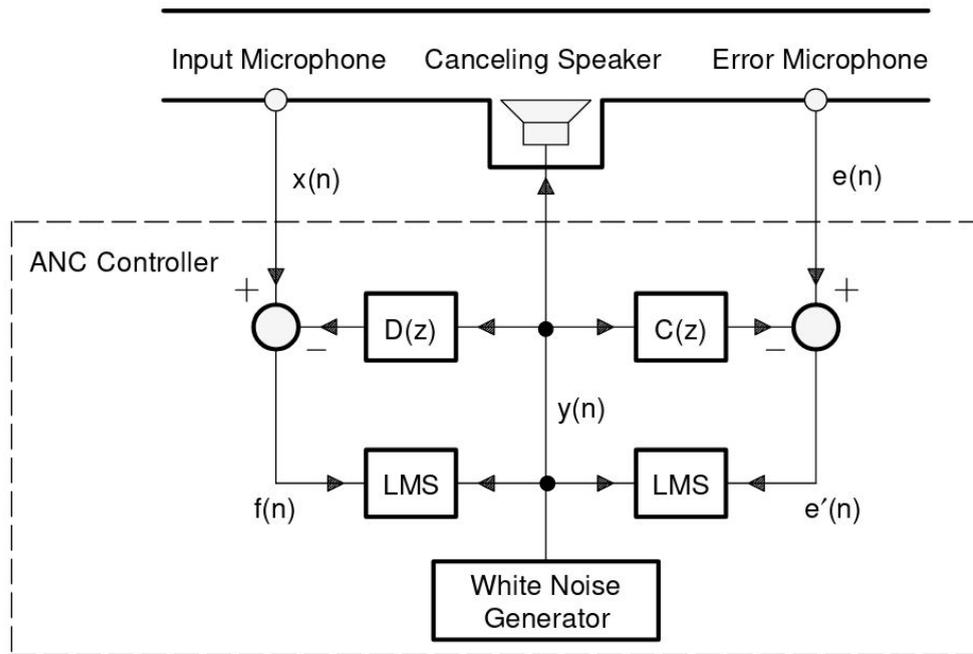
### 3.2.3 Estimativa dos efeitos de caminho secundário e *feedback* acústico

A estimativa destes efeitos é feita antes do processo de controle ativo, ou seja, é realizado um treinamento *off-line* do sistema quanto a estas propriedades do sistema. Esta fase consiste na utilização de dois filtros adaptativos, um para cada efeito, de modo a modelar sistemas desconhecidos que causam estes fenômenos.

O efeito causado pelo caminho secundário envolve as propriedades eletromecânicas que causam a distorção do sinal de saída do sistema. Esta alteração implica na modificação do formato de onda esperado pelo sensor de erro, pois é nesta seção do sistema que o ruído secundário estará presente.

Já o *feedback* acústico é formado pela interação de uma porção possivelmente distorcida do ruído secundário com o ruído primário na região em que o sensor de referência está contido. Logo, o resultado desta interação é dado para o sistema como sinal contaminado de referência. Esta malha de realimentação insere pólos no sistema resultante, o que pode causar instabilidade [6].

É possível perceber que ambos os efeitos podem ser percebidos a partir da ação da fonte secundária. Ou seja, esta fonte pode ser utilizada de modo a modelar ambos os efeitos para serem considerados durante a execução do sistema. O processo a ser realizado está descrito graficamente na Figura 8, sendo  $D(z)$  e  $C(z)$  os filtros de modelagem dos efeitos de *feedback* acústico e caminho secundário, respectivamente. Neste processo é utilizado um gerador de ruído branco para modelagem destes sistemas, pois este tipo de ruído é formado por todas as frequências de acordo com a taxa de amostragem, obedecendo o teorema de amostragem de Nyquist-Shannon. Desta forma é possível adaptar os elementos  $D(z)$  e  $C(z)$  para incorporar informação de todo o espectro de frequência possível.



**Figura 8.** Diagrama de blocos da estimativa dos efeitos de caminho secundário *feedback* acústico [6]

### 3.2.4 Execução

O fluxo de execução assume que os *handlers* dos dispositivos já foram adquiridos e as estimativas de caminho secundário e *feedback* acústico realizadas. A execução do sistema basicamente consiste em iterar sobre o *buffer* de *frames* e obter um conjunto de *frames* de saída e, ao final disto, enviar para reprodução.

### 3.2.5 Resultados

Os resultados não foram satisfatórios, pois, mesmo revisitando a literatura acerca de informações sobre a implementação e escolha dos parâmetros, não foi possível alcançar a convergência do algoritmo. Na prática isto significa que o sistema, depois de um período de tempo proporcional ao passo de aprendizagem, chegava a um ponto de instabilidade tal que o sistema passava a emitir ruídos de amplitude muito grandes.

Ao realizar uma investigação mais minuciosa, foi percebido que existia um efeito que o tamanho dos *buffers* de acesso ao *driver* de áudio causava na resposta do sistema, alterando o sinal secundário. Porém, ao tentar reduzir o tamanho destes *buffers* a 1 *frame* por chamada ao sistema, a resposta era de criação de ruídos pela ocorrência de *underrun*. Este erro consiste na insuficiência de amostras para reprodução. Em outras palavras, dado um *buffer* de um *frame* enviado para reprodução, o programa suspenso pelo sistema operacional passará um período de tempo maior do que o necessário para executar a reprodução destas duas amostras. Logo, o sistema de som terá consumido as amostras disponíveis e o programa não

será capaz de gerar as amostras restantes a tempo de prosseguir enviando as demais amostras. O problema deste efeito é que, durante o tempo em que o programa não envia amostras, o dispositivo de áudio pode inserir outros níveis de som (tais como zero). O consequência disto é a geração de frequências muito altas, gerando estalos na saída e alterando o formato de onda desejado. Além disto, a utilização de *buffers* interfere também na capacidade de atenuação de um algoritmo de filtro adaptativo, pois este também fere as restrições temporais do algoritmo. Foi verificado através de testes que existe uma limitação prática para o tamanho mínimo deste *buffer* de 12 *frames*. Ou seja, não é possível diminuir absolutamente este valor, assim como verificado em outros trabalhos [17].

A fim de verificar este problema, foi realizada uma simulação do *setup* de controle ativo de ruídos considerando a utilização de *buffers* para captura e reprodução de amostras de áudio. O ambiente de execução utilizado para esta simulação foi o *Octave*, considerando os seguintes parâmetros:

- Velocidade do som: 340 *m/s*
- Distância entre sensor de referência e sensor de erro: 2 *m*
- Distância entre sensor de erro e fonte secundária: 15 *cm*
- Taxa de amostragem: 5 *kHz*
- Estrutura do controlador ANC: *Leaky LMS*
- Influência do ambiente na amplitude de propagação do som: linear, em função da distância, sendo  $y(d) = k(d)y(0)$ , onde  $k(d) = 1 - 0,05d$
- Ordem do filtro: 60
- Passo de aprendizagem: 0,002
- Fator de esquecimento: 0,9999

Os efeitos de caminho secundário e *feedback* acústico foram desconsiderados. Isto é o mesmo que dizer que a modelagem destes efeitos alcançou a igualdade, ou seja, foi muito boa. A parametrização da simulação consiste na definição do tamanho dos *buffers* em termos de *frames*. Por se tratar de uma simulação, a limitação prática citada acima pode ser superada. A seguir, verificam-se resultados desta simulação para *buffers*  $k$  *frames*, sendo  $k$  um número inteiro tal que  $k \in \{1, 2, 3, 4\}$ . Foi considerada também a seguinte nomenclatura:

- $x(t)$  é o ruído primário, em uma região muito próxima da fonte.
- $d(t)$  é o ruído a ser cancelado. Ou seja,  $d(t) = x(t) * P(t)$ , sendo  $P(t)$  o caminho primário e a operação '\*' a convolução.
- $y(t)$  é a saída do sistema, ou seja, o ruído secundário.
- $e(t)$  é o sinal de erro, o que significa dizer que  $e(t) = d(t) + y(t)$ .

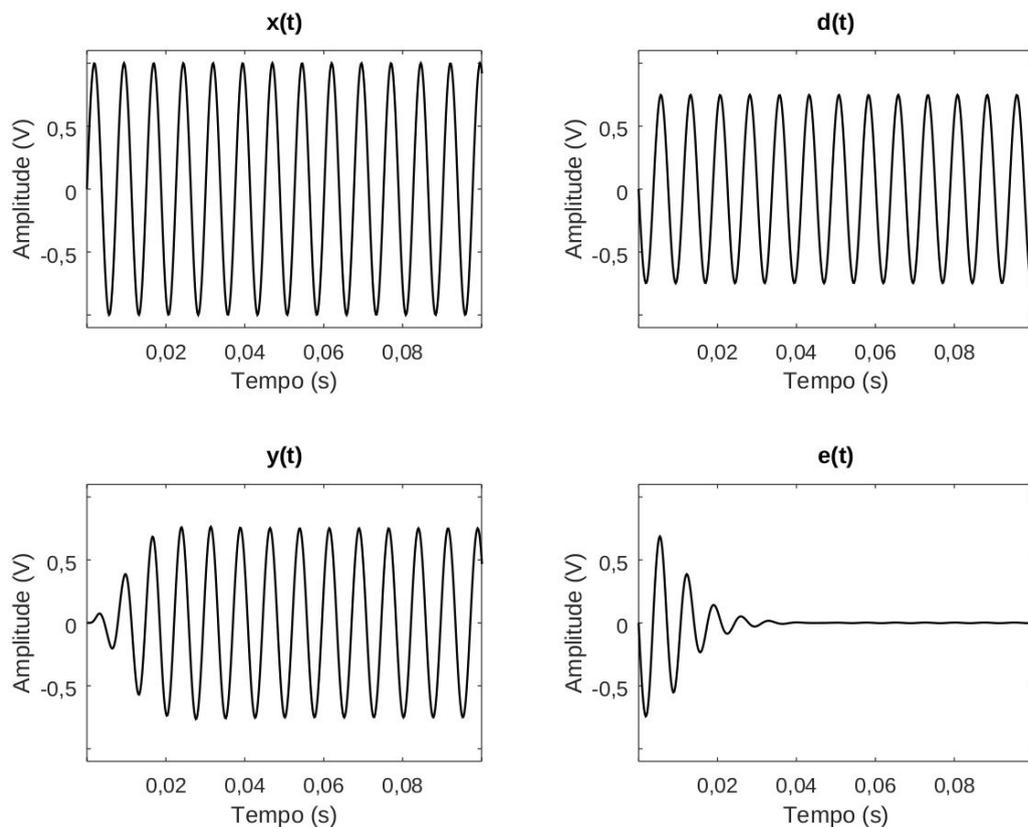
O resultado exposto pela Figura 9 significa o funcionamento do algoritmo para um sistema de captura e reprodução de amostras de áudio com  $k = 1$ , ou seja, não utilizam memória de *buffers*. Este caso é indicado pela literatura científica para aplicações em sistemas do mundo real [5,6]. É notável a redução da amplitude do sinal de erro, sendo isto a representação da atenuação do ruído primário. Portanto, é possível afirmar que o algoritmo empregado nesta simulação é capaz de realizar o controle ativo de um ruído sob estas circunstâncias. É vista também a necessidade

de um intervalo de tempo para convergência do algoritmo. Isto é previsto pela fundamentação teórica e existem diversos trabalhos que investigam formas de diminuir este tempo, resultando em sistemas mais sofisticados [9].

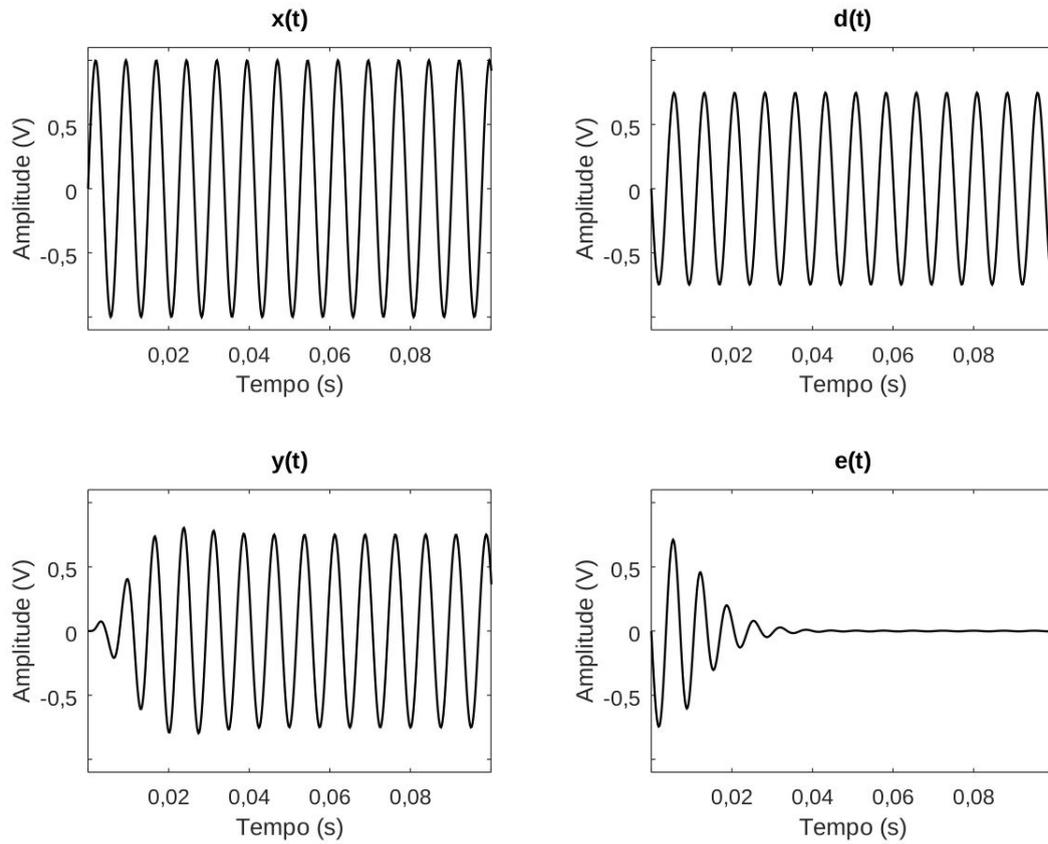
A Figura 10, por sua vez, representa o resultado da simulação quando  $k = 2$ . Isto significa que existe o uso de memória para um único *frame* para captura e reprodução de amostras de som. Fica nítido que, para esta configuração, também é possível alcançar a atenuação do ruído. Existe apenas um prolongamento do tempo de convergência muitíssimo pequeno, praticamente imperceptível por pura inspeção visual. Por este motivo, existe uma semelhança muito grande entre este resultado e o demonstrado pela Figura 9. A diferença entre estes resultados, porém, deve aumentar para ruídos mais complexos.

Quando  $k = 3$ , como pode ser visto pela Figura 11, o sinal de erro tem uma leve diminuição na amplitude se comparado com  $d(t)$ . É visto, porém, uma alteração brusca no formato de onda do sinal. Levando em consideração a definição de ruído apresentada no Capítulo 1, este resultado pode ser compreendido como uma amplificação do ruído do aspecto qualitativo. Isto é, sob estas configurações, o sistema tornou-se incapaz de atenuar o ruído apresentado e passou a amplificar o mesmo, o que é uma resposta contrária à desejada. Fica evidente, portanto, a influência do tamanho dos *buffers* na atuação do sistema.

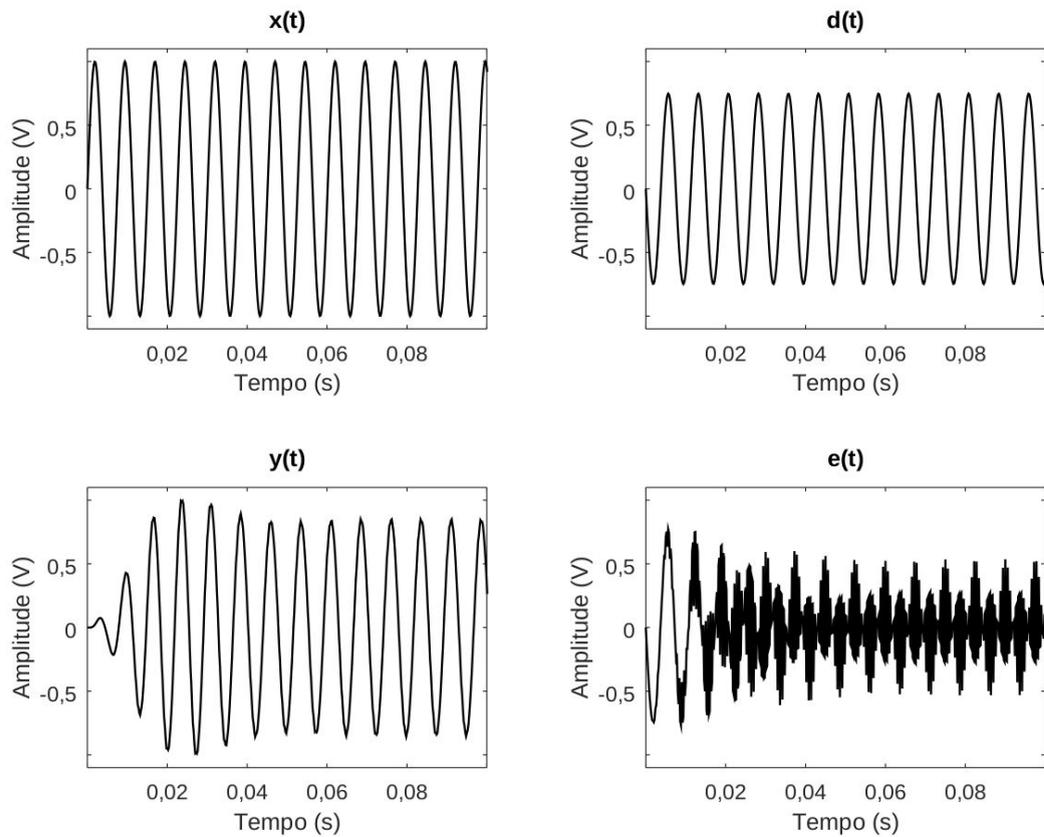
A Figura 12 mostra a mesma resposta quando  $k = 4$ . Percebe-se que o resultado tende a piorar quando o valor de  $k$  cresce. Vale ressaltar que a queda na qualidade da atenuação passa a piorar também no aspecto quantitativo, pois a amplitude do sinal de erro tende a aumentar.



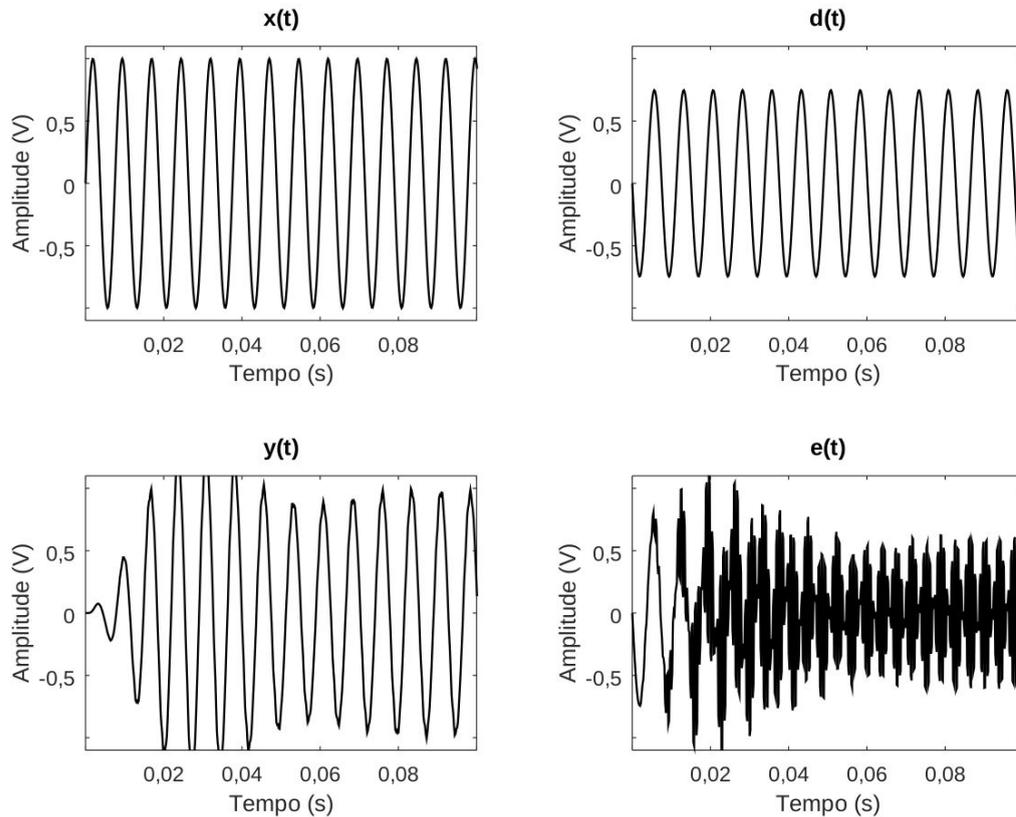
**Figura 9.** Resultado da simulação para  $k = 1$



**Figura 10.** Resultado da simulação para  $k = 2$



**Figura 11.** Resultado da simulação para  $k = 3$



**Figura 12.** Resultado da simulação para  $k = 4$

Em resumo, é possível perceber que a qualidade do resultado do algoritmo se torna pior a medida que o *buffer* aumenta, não só quantitativamente, mas também qualitativamente. Logo, com estes resultados, é possível afirmar a inviabilidade da utilização de sistemas operacionais de uso geral para execução das técnicas de controle ativo de ruído existentes, exceto para casos onde estas técnicas são adaptadas para ruídos conhecidos e pouco variáveis [17].

Não é possível, porém, afirmar sobre a implementação realizada para sistemas operacionais dedicados a operações com restrições temporais, tais como aqueles com núcleos de tempo real ou modificados para tal. O mesmo pode ser generalizado para o *hardware* de uso geral, como computadores pessoais, que por apresentarem grande poder computacional, pode também, dedicado a esta aplicação, mostrar sua viabilidade. A utilização deste tipo de sistema para este tipo de aplicação necessita de investigação detalhada que não faz parte do escopo deste trabalho.

# Capítulo 4

## Implementação em um sistema DSP

Segundo a conclusão obtida de que é inviável a execução de algoritmos de cancelamento ativo utilizando captura e reprodução de amostras em blocos (*buffers*), é necessário utilizar um sistema que não ofereça tal limitação. Em outras palavras, o funcionamento destas técnicas de ANC devem acontecer, baseado nos resultados obtidos anteriormente, quando as amostras de entrada e saída puderem ser manipuladas uma a uma. Logo, para a criação de um sistema ANC, será implementado este sistema utilizando um processador DSP.

A grande vantagem de utilização de um processador DSP está no fato de que o projeto deste tipo de equipamento considera que estes devem executar instruções com tempo previsível. Isto acontece pois a tarefa de processar sinais tem a característica de lidar com fluxos de dados contínuos, como ondas sonoras. Além disto, são processadores otimizados para execução de operações matemáticas.

Os processadores DSP, portanto, são úteis e apropriados para a implementação de sistemas ANC pois a utilização destes elimina o problema de captura e reprodução de amostras verificado nos sistemas de uso geral. Neste capítulo será abordado o trabalho realizado para a realização deste desenvolvimento.

### 4.1 Configuração e ferramentas do sistema

O processador escolhido para implementação desta fase foi o TMS320C5515, projetado e produzido pela Texas Instruments, embarcado na placa de desenvolvimento TMS320C5515 EVM, produzido Pela Spectrum Digital. Esta escolha foi baseada na frequência de *clock* máxima de processamento e no consumo de energia, parâmetros com valores interessantes se comparados com a média do mercado. Este equipamento utiliza o *codec* de áudio TLV320AIC3204, também produzido pela Texas Instruments, que vai oferecer uma maneira simplificada de configurar a captura e reprodução de amostras de áudio.

Nesta fase do projeto, foi possível utilizar um osciloscópio para análise do sistema. A estrutura do filtro, os sensores e atuadores utilizados foram os mesmos descritos na seção 3.1.

### 4.2 Implementação

#### 4.2.1 Configuração de acesso ao áudio

Como já visto, as frequências mais notáveis do ruído gerado pelo sistema motor de um automóvel estão abaixo dos 1500 Hz. Por este motivo, com o objetivo de diminuir a quantidade de coeficientes do filtro e disponibilizar maior tempo disponível para a geração do ruído secundário, buscou-se configurar o *codec* de

áudio para funcionar a uma taxa de amostragem de 3000 Hz. Para isto, é necessário configurar alguns parâmetros como descrito na documentação oficial seguindo as equações presentes em [18] e [19]. Mais especificamente, em [18], são vistas as equações que configuram o *clock* do periférico I2S, responsável pela amostragem e roteamento da entrada e saída de áudio, tal que:

$$I2SnCLK = SystemClock / (2^{CLKDIV + 1}) \quad (5.1)$$

$$I2SnFS = I2SnCLK / (2^{FS DIV + 3}) \quad (5.2)$$

Interessa que *I2SnFS* seja igual a taxa de amostragem desejada e *I2SnCLK* será utilizado pelo TLV320AIC3204. Porém, visto que o *SystemClock* pode apenas assumir um conjunto de valores discretos e este valor deve ser dividido por uma potência de base 2, é fácil deduzir que não é possível definir *I2SnFS* para qualquer valor desejado.

Quanto a configuração do TLV320AIC3204, como visto em [19], é necessário que:

$$ADCfs = \frac{CODECCLKIN}{NADC \times MADC \times AOSR} \quad (5.3)$$

$$DACfs = \frac{CODECCLKIN}{NDAC \times MDAC \times DOSR} \quad (5.4)$$

O *CODECCLKIN* é o *I2SnCLK* e os valores de *DACfs* e *ADCfs* devem ser iguais ao *I2SnFS* de modo que entrada e saída devem ser configuradas para operarem a mesma taxa. Estes valores foram definidos com o objetivo de obter uma taxa de amostragem próxima a 3000 Hz. Com as configurações realizadas, o valor obtido foi 3661,75 Hz.

#### 4.2.2 Primeira implementação

A primeira implementação foi obtida pela tradução do código gerado para o sistema de uso geral, fazendo as devidas adaptações e definindo o número de coeficientes dos filtros como 20. Esta obteve pouco sucesso por funcionar com pouca qualidade apenas para ruídos senoidais simples. Não foram feitos registros destes resultados por não representar uma situação satisfatória. Foi visto que a baixa qualidade do ruído secundário era consequência do baixo poder de descrição do filtro por apresentar poucos coeficientes. Ao aumentar o número de coeficientes, porém, a qualidade da resposta do sistema era ainda menor, caracterizando assim como insuficiência de performance do algoritmo empregado. Logo, para melhorar estes resultados, foi necessário refatorar o código com o objetivo de substituir operações de alto custo computacional por alternativas rápidas. Logo, foi visto que existiam duas operações bastante recorrentes e que estavam tornando o algoritmo lento: operações matemáticas em ponto flutuante e atualização dos regressores dos filtros envolvidos. Cada um destes problemas identificados foram investigados individualmente.

### 4.2.3 Operações em ponto fixo

Valores de ponto fixo são representações de números decimais com uma quantidade de *bits* fixa dedicada para a parte fracionária do valor, sendo os demais *bits* do valor inteiro para a parte inteira [20]. Isto difere das implementações de ponto flutuante, pois estas contam com variação da posição do ponto decimal. Enquanto os valores de ponto flutuante possibilitam a representação de valores com maior precisão em números pequenos, estes demandam muitas operações para conversão. Estão disponíveis no Anexo A mais detalhes sobre o tema.

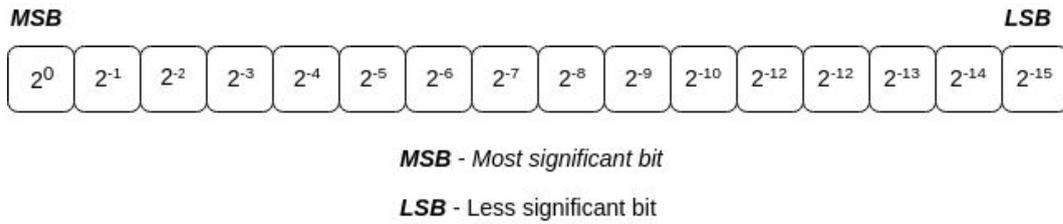
É visto através de uma pesquisa rápida sobre os produtos disponíveis no mercado que a grande maioria dos processadores DSP (incluindo o TMS320C5515) não são fabricados com uma unidade de ponto flutuante (FPU, do inglês *Floating Point Unit*) por motivos de redução de custo e consumo de energia. Estas unidades são responsáveis pelo tratamento de valores em ponto flutuante, como sua formatação para armazenamento. Portanto, é comum notar em projetos de processamento digital de sinais a presença de operações em valores de ponto fixo, que permitem executar operações matemáticas com valores decimais utilizando apenas valores inteiros.

Para definir a divisão de *bits* para as partes inteira e fracionária, é importante levar em consideração o total de *bits* disponíveis em um valor. Isto é respondido ao analisar os parâmetros dos conversores analógico-digital, que fornecem valores de 16 *bits*. Portanto, para acelerar o algoritmo atual, foi implementada a representação Q15, que significa reservar o *bit* mais significativo para a parte inteira e os demais para a parte fracionária, sendo assim possível representar valores entre  $-1$  e  $0,9999694824$ . É possível ver esta representação graficamente na Figura 13. Para operar com estes valores foram desenvolvidas as operações de soma com saturação e multiplicação, visto que eram as únicas operações matemáticas necessárias para a realização do controle ativo.

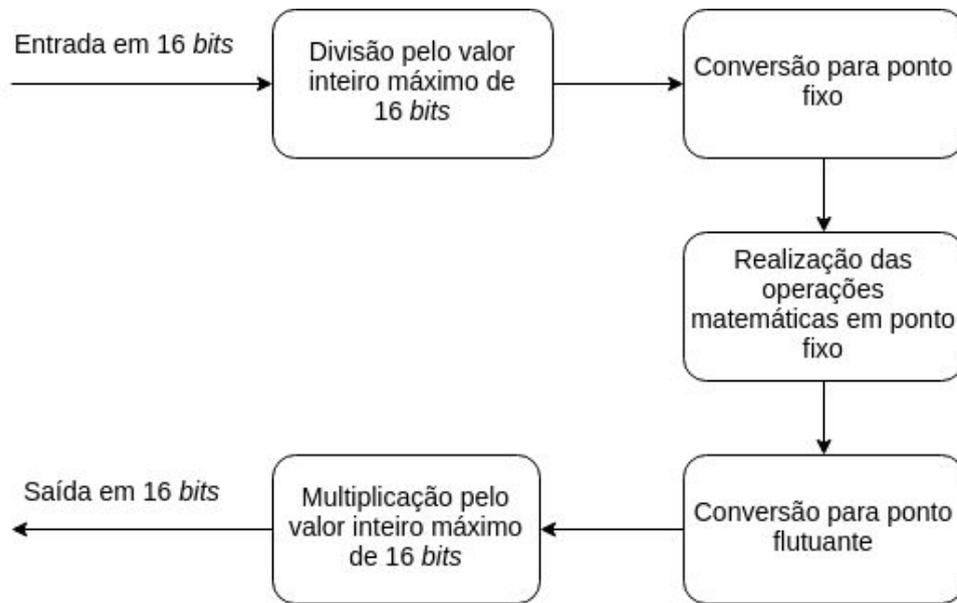
A utilização destes valores em Q15 é possível após uma conversão de um valor em ponto flutuante de mesma magnitude. Logo, é necessário normalizar o valor de entrada e, para isso, foi empregada uma divisão pelo valor inteiro máximo de 16 *bits*. Após isto, estes valores eram utilizados pelas operações de geração do ruído secundário. Após isto, é necessário converter de volta para ponto flutuante e então, multiplicado pelo valor inteiro máximo de 16 *bits*. Este valor resultante é dado como saída do algoritmo. Este processo está representado na Figura 14.

A implementação deste foi realizada em um computador *desktop* (mais especificamente o mesmo utilizado para execução do sistema ANC descrito no Capítulo 4) para simplificar a validação de corretude das conversões e operações matemáticas implementadas. Após esta validação, o algoritmo foi cuidadosamente traduzido para execução no sistema DSP, fazendo as devidas adaptações.

Para validar a corretude desta implementação foram criados programas com instruções para realizar conversões e operações arbitrárias utilizando diversos valores. Já para verificação da performance, foi desenvolvido uma rotina com muitas repetições para realizar operações matemáticas com ponto fixo e ponto flutuante. O programa com ponto flutuante durou cerca de 12 segundos, enquanto o programa com ponto fixo durou cerca de 3,9 segundos, caracterizando um aumento de velocidade de, aproximadamente, 3 vezes.



**Figura 13.** Representação do Q15



**Figura 14.** Fluxo das amostras através da conversão para ponto fixo

#### 4.2.4 Listas circulares

Uma operação muito recorrente em algoritmos de filtragem, seja adaptativa ou não, é a operação de atualização dos regressores, que consiste na inserção de uma amostra no início do filtro, remoção da amostra mais antiga e deslocamento das demais. Esta operação pode ser vista na Figura 15 e é conhecida como *shifting*.

O *shifting*, quando realizado com vetores, é uma operação de complexidade  $O(n)$ , ou seja, o custo computacional relacionado a esta cresce linearmente de acordo com o tamanho do vetor associado. O problema é que esta operação acontece pelo menos uma vez para cada filtro associado no processo por iteração. Logo, a implementação do FBFxLMS, que conta com três filtros, deve ter pelo menos 3 *shifts*. Isto significa que a complexidade do algoritmo aumenta com o tamanho do vetor com uma proporção de 3. De forma a obter um algoritmo escalável, é desejável substituir esta operação por uma similar, porém de complexidade constante. Isto é possível pela aplicação de listas circulares, pois a alteração de elementos da estrutura pode ser realizado em tempo constante.



Tendo em vista o mesmo ganho de performance obtido pela aplicação de listas circulares comentada anteriormente, é comum encontrar no conjunto de instruções de microcontroladores RISC uma operação para *shifting* em vetores. A disposição desta instrução, por ser implementada em *hardware*, fornece o mesmo resultado em um único ciclo de execução. Por este motivo, a implementação com listas circulares pode ser substituída pela instrução referida nos processadores que esta estiver disponível. A vantagem, porém, da implementação com listas circulares é construída pelo fato de que esta deve funcionar mesmo em processadores que não apresentem esta instrução dentro do seu conjunto de instruções. Por este motivo esta foi a solução escolhida para este desenvolvimento.

#### 4.2.5 Implementação final

Após a aplicação das técnicas descritas nas seções anteriores, tornou-se possível incrementar a quantidade de coeficientes dos filtros envolvidos no sistema. Através da realização de testes, foi possível definir valores dos parâmetros que produziram resultados satisfatórios em atenuar ruídos mais complexos, tais como o ruído gerado pelo sistema motor de um automóvel. Foi adicionado ao algoritmo também um parâmetro de tolerância em ponto fixo, que este indica um valor de erro máximo ao qual o sistema pode se considerar correto e não realiza a adaptação dos coeficientes do filtro. Esta medida foi adotada para evitar instabilidade por *overfitting*. Os parâmetros encontrados nos testes foram:

- Número de coeficientes: 60
- Tolerância: 1000 (em ponto fixo)
- Passo de aprendizagem: 0,01
- Leakage factor: 0,98

A Figura 17 retrata o *setup* de experimentação elaborado para execução dos testes. Nessa estão destacados os elementos que compõem o protótipo final, tais como os sensores utilizados e o atuador do sistema. É possível verificar também os componentes do ambiente que interferem na atuação do algoritmo.

#### 4.2.6 Resultados

A seguir são mostrados os resultados obtidos com a implementação final. Os ruídos foram selecionados de uma base de gravações de 20 segundos cada coletadas de um único automóvel, sendo gravações em rotações constantes (1000 rpm, 2000 rpm, 3000 rpm e 4000 rpm) e rotações variáveis (entre 1000 rpm e 2000 rpm, entre 2000 rpm e 3000 rpm e entre 3000 rpm e 4000 rpm).

Visto que as técnicas de controle ativo funcionam melhor em baixas frequências [6,9], os testes iniciais foram feitos com ruídos em maior rotação, para verificação da qualidade do algoritmo. Porém, nem todas as gravações da base foram utilizadas, pois não foi possível submeter o sistema a ruídos do sistema motor em baixas rotações. Isto aconteceu por que o sistema, quando na presença de gravações abaixo de 3000 rpm demonstrou a captura de sinais de erro

insignificantes, ou seja, muito próximo do sinal nulo. Isso foi verificado empiricamente, dado que uma pessoa submetida a estes ruídos na região de atenuação durante os testes com o sistema desligado foi capaz de ouvir o som indesejado. Mas o sensor de erro demonstrou, sob as mesmas circunstâncias, sinais de amplitude desprezíveis. Como o algoritmo proposto realiza a adaptação dos coeficientes do filtro utilizando as amostras de erro, o sistema não foi capaz de apresentar nenhum nível de atenuação ou aprendizado. Portanto, pelo fato de que tornou-se impossível testar o algoritmo nestas condições, os únicos resultados demonstrados a seguir são da interação do sistema utilizando gravações do motor a 4000 rpm e senóides de 133 Hz (que tem relação com o ruído do motor a 4000 rpm). É nítido verificar este problema ao analisar a Figura 22.

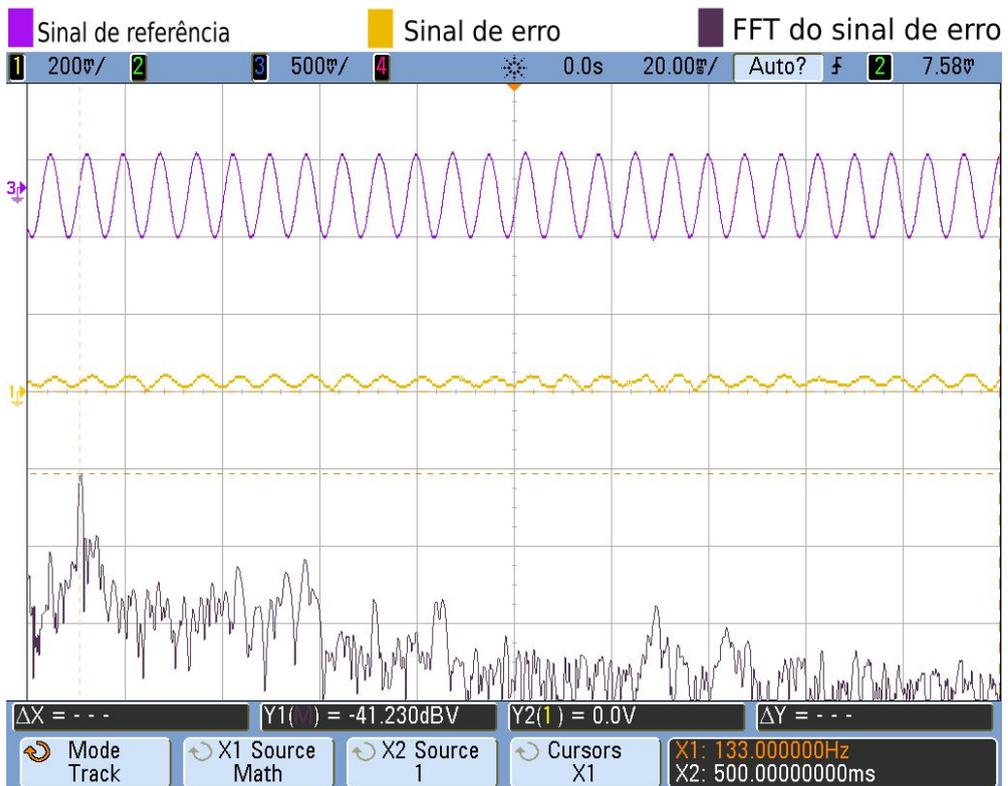
A Figura 18 mostra a captura dos sinais de referência e erro sem a atuação do sistema sobre o ruído. Vale notar que a amplitude de ambos os sinais não estão apresentados na mesma escala para melhor análise visual. Por este motivo, o sinal de erro pode parecer ter amplitude maior que o de referência, o que não faria sentido na maioria das ambientes. A FFT (do inglês, *fast fourier transform*) do sinal de erro mostra a predominância de um tom em 133 Hz. A Figura 19, por sua vez, mostra a atuação do sistema sobre as mesmas circunstâncias apresentadas na Figura 18. É possível constatar que o tom predominante foi reduzido em cerca de 20 dBV, o que significa reduzir a amplitude do sinal em cerca de 10 vezes.



**Figura 17.** Setup de experimentação



**Figura 18.** Ruído senoidal de 133 Hz com o sistema desligado

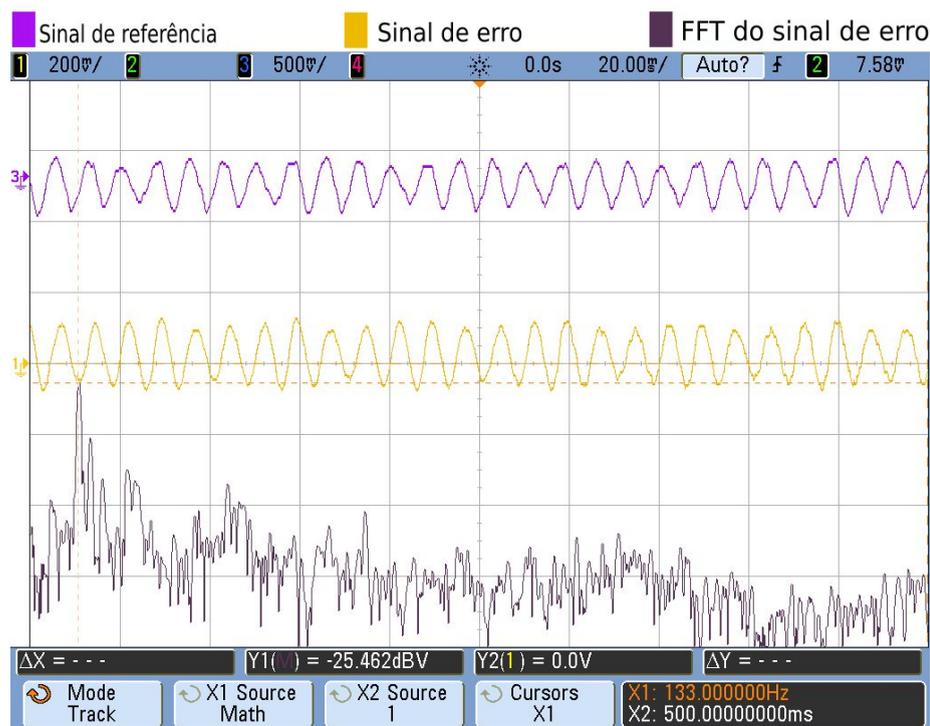


**Figura 19.** Ruído senoidal de 133 Hz com o sistema ligado

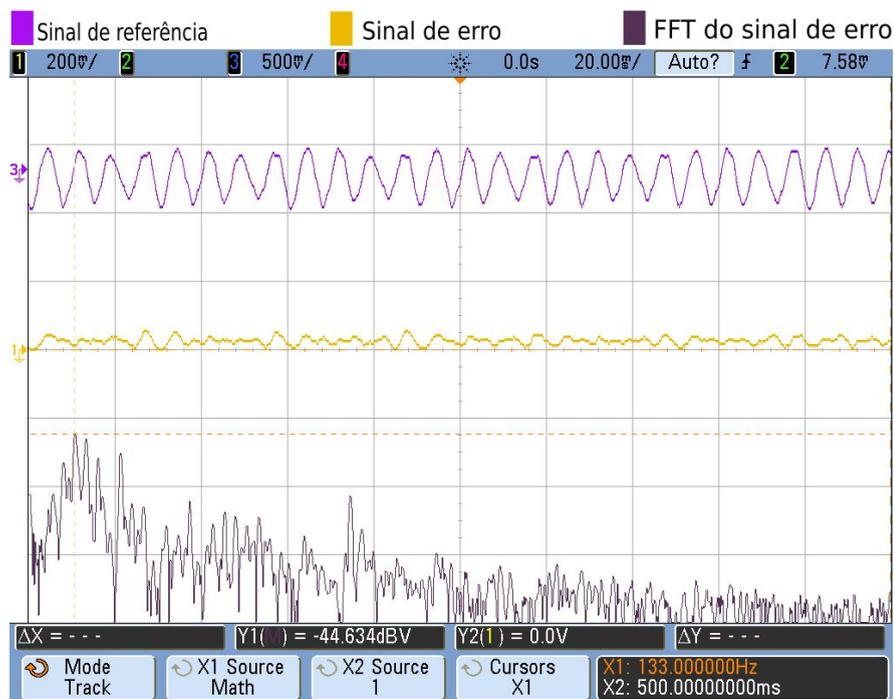
Os próximos resultados foram obtidos ao apresentar ao sistema a gravação de um sistema motor funcionando a 4000 rpm. Ao analisar a FFT do sinal de erro na Figura 20, é possível constatar a maior complexidade deste sinal por notar que este é composto por diversos tons. Nesta Figura, o sistema foi mantido inativo, enquanto na Figura 21 o algoritmo passou a ser executado. Estes resultados são importantes, pois demonstram a eficiência do protótipo para estes ruídos de frequência relativamente alta. De acordo com a literatura científica, é provável que o sistema funcione para frequências mais baixas nesta mesma configuração [9].

A atenuação realizada sobre a componente principal, como é possível ver pelo uso dos cursores do osciloscópio, foi de magnitude similar ao ruído puramente senoidal. É interessante notar este fato, pois assim é verificado que o sistema mostra robustez quanto a complexidade do ruído apresentado, já que com a mesma configuração, foi possível controlar sinais indesejados distintos e com a mesma taxa de redução.

É possível também verificar que o sistema foi capaz de atenuar outras componentes do ruído, porém, em níveis muito pequenos, praticamente desprezíveis. Isto pode revelar a dificuldade do algoritmo de encontrar a melhor solução possível. Ou seja, as técnicas baseadas em mínimos quadrados podem ficar presas em mínimos locais [21].

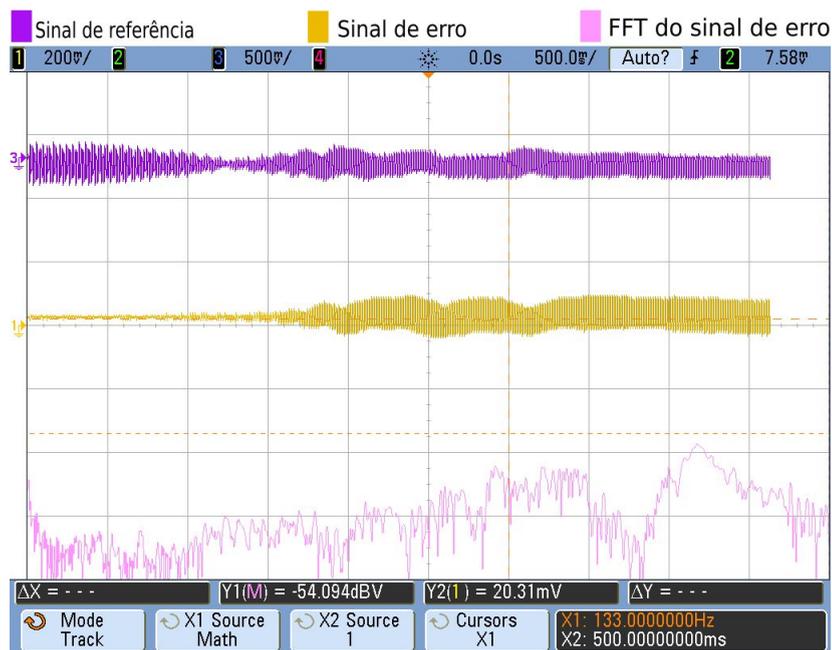


**Figura 20.** Ruído gerado pelo sistema motor de um automóvel a 4000 rpm com o sistema desligado

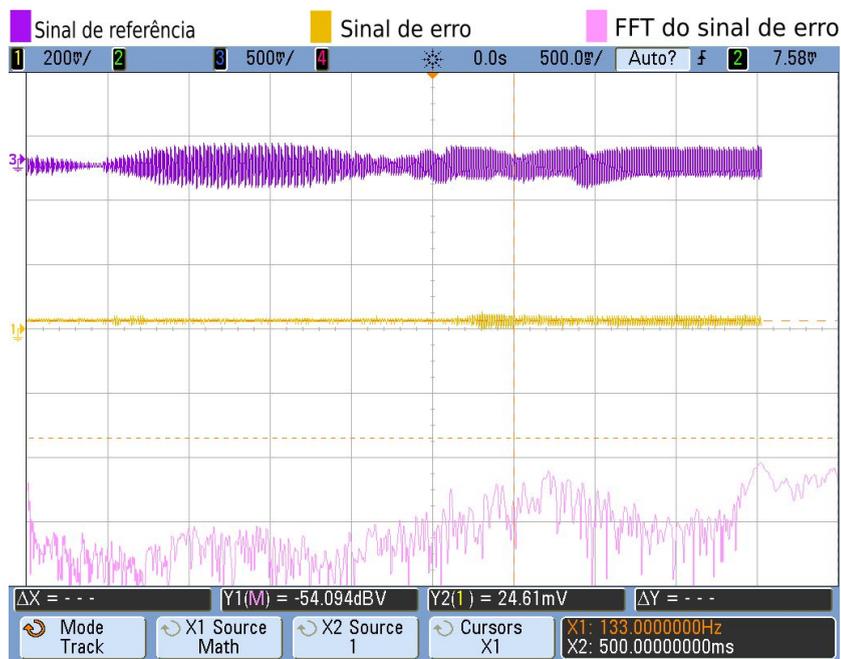


**Figura 21.** Ruído gerado pelo sistema motor de um automóvel a 4000 rpm com sistema ligado

Os resultados mostrados nas Figuras 22 e 23 mostram a atuação do sistema quando o ruído é variável. Para esta situação, existe a necessidade de que o algoritmo adapte os coeficientes do filtro rápido o suficiente durante a variação das características do ruído [9].

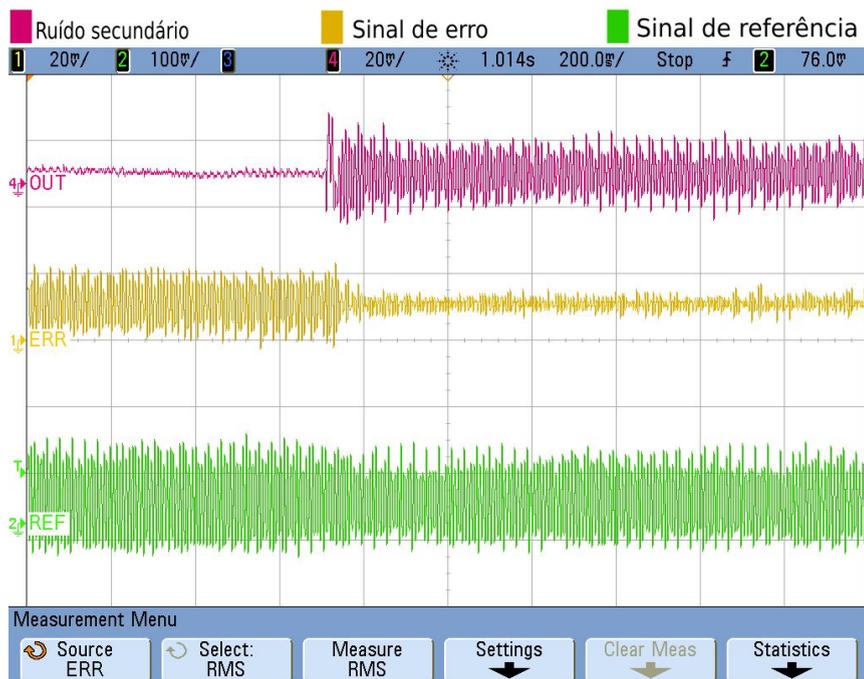


**Figura 22.** Ruído gerado pelo sistema motor de um automóvel acelerando de 3000 rpm para 4000 rpm com o sistema desligado



**Figura 23.** Ruído gerado pelo sistema motor de um automóvel acelerando de 3000 rpm para 4000 rpm com o sistema ligado

A Figura 24 exemplifica a ativação do sistema em um instante de tempo aleatório durante a apresentação do ruído. Este resultado demonstra que o sistema é capaz de atenuar o som indesejado e manter a instabilidade sem a inserção de novas componentes indesejadas.



**Figura 24.** Resposta do sistema ao ser ligado em um momento aleatório

# Capítulo 5

## Conclusões e trabalhos futuros

A presença de ruído acústico no ambiente é inegável, principalmente em grandes centros urbanos. Isto é um fato perigoso, pois é comprovado que a exposição contínua a som indesejado pode causar doenças físicas e psicológicas. O controle de ruído, portanto, torna-se uma aplicação de grande importância. Para a fabricação de automóveis, que existe a tendência natural pelo aumento do espaço interno e redução do peso, as técnicas ativas se mostram mais interessantes do que as passivas, visto a natureza do ruído gerado pelo sistema motor. Logo, é importante o estudo e a implementação deste tipo de sistema computacional.

Este trabalho se propôs a discutir as dificuldades encontradas e descrever as soluções propostas para os mesmos. Assim, torna-se documentado o trabalho que foi realizado para obter uma implementação eficiente de um sistema ANC. Portanto, é fácil verificar a importância deste desenvolvimento, pois este descreve problemas da transição dos modelos teóricos oferecidos na literatura acadêmica para o sistema em funcionamento.

Em resumo, os resultados obtidos com este trabalho descrevem aspectos necessários para o desenvolvimento de sistemas de controle ativo de ruídos acústicos. É importante notar que não é possível utilizar sistemas operacionais de uso geral para execução de sistemas ANC sem a realização de considerações prévias sobre o ruído. Esta deficiência acontece devido a rotina de execução de processos sem ordem ou tempo determinístico. Para a aplicação em ambientes automotivos, que o ruído é complexo e variável, isto se torna impraticável. Já para a implementação destes sistemas utilizando processadores DSP, foi demonstrada a atenuação de cerca de 10 vezes na amplitude do sinal. Também foi verificado que a implementação obtida é capaz de tratar ruídos variáveis, pois este apresenta velocidade de adaptação suficiente. Também é útil entender as necessidades de cada *hardware* e empregar soluções específicas para estes. A utilização deste tipo de equipamento, apesar de necessária, é bastante custosa, pois demanda a produção de um sistema exclusivo para as tarefas de ANC. Para aplicações de propósito específico, porém, como dispositivos embarcados, a aplicação de um DSP é viável. Por fim, é possível verificar que a aplicação da teoria se mostra bastante robusta e confiável.

Das dificuldades enfrentadas para o desenvolvimento deste trabalho, é possível citar a inaptidão de sistemas de uso geral para tarefas de controle ativo de ruídos acústicos, pois a investigação deste problema demandou bastante tempo. Também foi bastante custoso realizar a configuração do TMS320C5515 e seus periféricos, já que todas as definições são realizadas através de chamadas de funções implementadas em baixo nível de abstração.

É possível estender estes resultados em trabalhos futuros pela realização de novos testes qualitativos baseados em avaliações de voluntários sobre o funcionamento do sistema. Assim será possível analisar a qualidade do sistema do ponto de vista dos usuários. Também recomenda-se investigações com outras abordagens que não sejam baseadas em filtros adaptativos, tais como algoritmos de

enxames. Também existe a possibilidade de empregar outras técnicas que consigam descrever o antirruído mesmo a partir da coleta dos sinais de erro e referência em *buffers*, assim como nos sistemas de uso geral. Assim, seria possível a execução de controle ativo neste tipo de sistema, como *smartphones*, etc.

É possível também para futuras contribuições, através de testes, determinar características da implantação deste sistema dentro de um automóvel. Não é possível afirmar qual a melhor forma de posicionar os elementos que compõem o sistema sem a realização de testes específicos para tal, mas compreender como o ambiente afeta a performance do algoritmo é importante. Inclusive essa forma pode variar de modelo para modelo de automóvel devido a variação de espaço interno e material utilizado. Neste *setup* apresentado na Figura 17, está presente no ambiente o efeito do caminho primário, previsto pelos modelos teóricos e é compreendido como a alteração provocada sobre o ruído pelo espaço entre os sensores de referência e erro. Também é verificado visualmente o efeito de *feedback* acústico, criado pelo sinal secundário ao ser propagado até o sensor de referência.

O objetivo é fazer com que os usuários do automóvel tenham numa região próxima ao posicionamento típico da cabeça os maiores níveis de atenuação. Como visto na Seção 2.2.2, isso pode ser obtido por colocar o sensor de erro próximo ao encosto da cabeça dos usuários a serem beneficiados pelo sistema ANC. Já o posicionamento do sensor de referência depende de testes, mas é proposto a verificação na região de repouso dos pés do usuário e dentro do compartimento do sistema motor. Estas duas variações afetam diretamente o efeito do caminho primário e do *feedback* acústico.

A implantação também depende do orçamento disponível para tal. Para que todos os usuários de um automóvel tenham a melhor percepção quanto a redução do ruído proveniente do sistema motor, é interessante a aplicação de um sistema para cada assento disponível. Esta configuração, apesar de ser custosa, consegue adaptar filtros dedicados para cada usuário, resultando em sinais secundários mais adaptados a cada posição possível dentro de um carro. Um possível problema para esta proposta é o fato de que o ruído secundário de um artefato poderia interferir no sinal de erro de outro, prejudicando assim a convergência. Outra possibilidade de implantação consiste no emprego de dois sistemas ANC, um para os assentos dianteiros e outro para os traseiros. Esta solução é mais econômica, porém tende a mostrar resultados de menor qualidade para cada indivíduo e, de acordo com a Seção 2.2, o princípio de conservação de energia tende a apresentar efeitos colaterais indesejados. Isto pode ser afirmado já que, visto que o sistema deve atuar por uma porção espacial maior, o ruído secundário emitido deve ter maior potência.

# Bibliografia

- [1] *Environmental Protection Agency, Clean Air Act Title IV - Noise Pollution*. Disponível em <http://www.epa.gov/clean-air-act-overview/clean-air-act-title-iv-noise-pollution>. Acesso em 2 de março de 2018.
- [2] L. I. C. do Carmo, **Efeitos do ruído ambiental no organismo humano e suas manifestações auditivas**. Disponível em [http://www.farmacia.ufrj.br/consumo/vidaurbana/Monografia\\_goiania.pdf](http://www.farmacia.ufrj.br/consumo/vidaurbana/Monografia_goiania.pdf). Acesso em 2 de março de 2018.
- [3] CONAMA, **Resolução Nº 252 de 01 de fevereiro de 1999**. Disponível em <http://www.mma.gov.br/port/conama/res/res99/res25299.html>. Acesso em 2 de março de 2018.
- [4] S. J. Elliott e P. A. Nelson, **Active noise control**, *IEEE signal processing magazine*, vol. 10, nº. 4, pp. 12–35, 1993.
- [5] S. M. Kuo e D. R. Morgan, **Active noise control: a tutorial review**, *Proceedings of the IEEE*, vol. 87, nº. 6, pp. 943-973, 1999.
- [6] S. M. Kuo, I. Panahi, K. M. Chung, T. Horner, M. Nadeski e J. Chyan, **Design of active noise control systems with the TMS320 family**, *Texas Instruments*, 1996.
- [7] T. M. Johnson, C. E. Hanson, J. C. Ross, A. K. Zaouk, e Q. N. America, **Development of passive and active noise control for next generation locomotive cabs**, *Inter noise*, 2009.
- [8] S. Wise e G. Leventhall, **Active noise control as a solution to low frequency noise problems**, *Journal of Low Frequency Noise Vibration and Active Control*, pp. 129-137, 2010.
- [9] R. P. Monteiro, **Improving the Active Noise Control in Automotive Vehicles Using Swarm Intelligence and Neural Computing**. Dissertação (Mestrado em Engenharia de Sistemas) - Universidade de Pernambuco. Pernambuco, 2017.
- [10] J. G. Proakis e D. K. Manolakis, **Digital Signal Processing: Principles, Algorithms and Applications**. 4. ed. New Jersey: Pearson, 2006.
- [11] A. Neubauer, **Non-linear adaptive filters based on genetic algorithms with applications to digital signal processing**. *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, Perth, WA, 1995, pp. 527-532 vol.2.
- [12] M. R. Srinivasan, **Physics for Engineers**. 2. ed. New Delhi. New Age International, 2001.
- [13] Wu Lifu, Qiu Xiaojun, Ma Jin e Guo Yecai, **Comparison of feedforward, feedback and hybrid structures on an ANC headphone**, *23 International Congress on Sound And Vibration*, Atenas, Grécia, 2016

- [14] Behringer, **ECM8000, Ultra-linear measurement condenser microphone**. Disponível em <http://www.musictri.be/Categories/Behringer/Microphones/Condenser/ECM8000/p/P0118>. Acesso em 22 de maio de 2018.
- [15] Behringer, **Eurorack UB1002FX, Ultra-low noise design 10-input 2-bus mic/line mixer with premium mic preamplifiers and multi-fx processor**. Disponível em <http://www.musictri.be/Categories/Behringer/Mixers/Analog/UB1002FX/p/P0300>. Acesso em 22 de maio de 2018.
- [16] Multilaser, **Caixa de som 2.1 20 W RMS gamer bivolt - SP097**. Disponível em <http://suporte.multilaser.com.br/downloads/detalhe/SP097/>. Acesso em 22 de maio de 2018.
- [17] H. A. Cordourier-Maruri, F. Orduña-Bustamante, **Active control of periodic fan noise in laptops: spectral width requirements in a delayed buffer implementation**. *Journal of Applied Research and Technology*, vol.7, nº.2, 2009.
- [18] Texas Instruments, **TMS320C5515/14/05/04 DSP Inter-IC Sound (I2S) Bus User's Guide**. Disponível em <http://www.ti.com/lit/ug/sprufx4b/sprufx4b.pdf>. Acesso em 2 de março de 2018.
- [19] Texas Instruments, **TLV320AIC3204 Application Reference Guide**. Disponível em <http://www.ti.com/lit/ml/slaa557/slaa557.pdf>. Acesso em 2 de março de 2018.
- [20] Randy Yates, **Fixed-Point Arithmetic: An Introduction**. Disponível em <https://courses.cs.washington.edu/courses/cse467/08au/labs/l5/fp.pdf>. Acesso em 22 de maio de 2018.
- [21] S. Tripathi e M. A. Ikbal, **Optimization of LMS Algorithm for Adaptive Filtering using Global Optimization Techniques**. *International Journal of Computer Applications*, vol 132, nº.10, 2015.

# Anexo A

## Números em ponto fixo

Esta Seção está dedicada a explanação de maiores detalhes sobre a representação e aritmética de números em ponto fixo. O texto “*Fixed-Point Arithmetic: An Introduction*”, indicado em [20] é a fonte que embasa o conteúdo a seguir. Pelo fato, porém de este ser escrito em uma língua estrangeira, os principais pontos serão traduzidos.

### Definição de ponto fixo

Uma coleção de  $N$  (um inteiro positivo) dígitos binários apresentam  $2^N$  possíveis estados. Isto é observável diretamente da teoria fundamental da contagem, que afirma que existem duas possibilidades para o primeiro *bit*, outras duas para o próximo *bit*, repetindo o mesmo raciocínio até o último *bit*, resultando em

$$2 \times 2 \times \dots \times 2 = 2^N$$

possibilidades. Em termos mais gerais, é possível utilizar estes estados para representar qualquer tipo de contagem.

A partir da teoria dos conjuntos e de álgebra elementar, é possível verificar uma representação como um mapeamento entre estados binários e elementos no conjunto representacional. O aspecto importante nisso é que não existe nenhum significado associado a uma palavra binária. Apesar de serem comumente associadas à números inteiros positivos, o significado de uma palavra binária de comprimento  $N$  depende inteiramente na interpretação associada a esta, isto é, depende do conjunto representacional e do mapeamento escolhido.

Considerando uma representação na qual o conjunto representacional é um subconjunto dos números racionais. Vale reforçar que números racionais são parte de um conjunto de números que podem ser escritos na forma  $a/b$ , onde  $a, b \in \mathbb{Z}$ ,  $b \neq 0$  ( $\mathbb{Z}$  é o conjunto dos números inteiros). O subconjunto anteriormente referido é formado pelos números racionais para os quais  $b = 2^N$ . Outra restrição aplicado a este conjunto de representação define que todos os elementos contidos nesse têm a mesma quantidade de dígitos binários e o ponto binário tem a mesma posição, isto é, o ponto binário é fixo. Esta representação é chamada de *ponto fixo*.

### Números racionais de ponto fixo sem sinal

Uma palavra binária de  $N$  *bits*, quando interpretada como um número racional de ponto fixo sem sinal pode assumir valores de um subconjunto  $P$  de números não negativos dados por

$$P = \{p/2^b \mid 0 \leq p \leq 2^N - 1, p \in \mathbb{Z}\}.$$

Vale notar que  $P$  contém  $2^N$  elementos. É denominada tal representação como  $U(a, b)$ , onde  $a = N - b$ .

Para  $U(a, b)$ , o  $n$ -ésimo *bit*, contando da direita para a esquerda, tem um peso de  $2^n / 2^b = 2^{n-b}$ , partindo de  $n = 0$ . Vale notar que quando  $n = b$  o peso é exatamente 1. Assim como na notação em base decimal, o ponto binário fica situado entre este *bit* e o próximo *bit* a direita. A esta posição é dado o nome de *ponto binário implícito*. Isto significa que  $U(a, b)$  tem  $a$  *bits* dedicados a parte inteira e  $b$  *bits* para a parte fracionária.

O valor de um número binário específico  $x$  utilizando  $N$  *bits* com uma representação  $U(a, b)$  é dado pela expressão

$$x = (1 / 2^b) \sum_{n=0}^{N-1} 2^n x_n$$

na qual  $x_n$  representa o *bit*  $n$  de  $x$ . O intervalo de uma representação  $U(a, b)$  abrange os valores entre 0 e  $(2^N - 1) / 2^b = 2^a - 2^{-b}$ . Por exemplo, considerando um número racional representado com uma palavra binária de comprimento 8 seguindo a representação  $U(6, 2)$  tem a forma

$$b_5 b_4 b_3 b_2 b_1 b_0 b_{-1} b_{-2},$$

onde o *bit*  $b_k$  tem o peso  $2^k$ . Vale notar que desde que  $b = 2$ , o ponto binário está a esquerda do segundo *bit* da direita para a esquerda, e assim, o número tem seis *bits* inteiros e dois fracionários. Esta representação abrange um intervalo de valores entre 0 e  $2^6 - 2^{-2}$ .

### Números racionais em ponto fixo sinalizados por complemento a dois

Uma palavra binária de comprimento  $N$ , quando interpretado como um valor sinalizado por complemento a dois, pode assumir valores de um subconjunto  $P$  de racionais dados por

$$P = \{p / 2^b \mid -2^{N-1} \leq p \leq 2^{N-1} - 1, p \in Z\}.$$

Vale notar que  $P$  contém  $2^N$  elementos. É denominada tal representação como  $A(a, b)$ , onde  $a = N - b - 1$ . O valor de um número binário específico  $x$  utilizando  $N$  *bits* com uma representação  $A(a, b)$  é dado pela expressão

$$x = (1 / 2^b) \left[ -2^{N-1} x_{N-1} + \sum_{n=0}^{N-2} 2^n x_n \right]$$

na qual  $x_n$  representa o *bit*  $n$  de  $x$ . O intervalo de uma representação  $A(a, b)$  abrange os valores entre  $-2^{N-1-b}$  e  $2^{N-1-b} - 1 / 2^b$ . O *bit* mais significativo nesta representação é usualmente referido como o sinal.

### Regras fundamentais da aritmética de ponto fixo

A seguir são listadas regras práticas da aritmética de ponto fixo. Para estas regras é importante notar que quando um operando pode ser tanto sinalizado quanto não sinalizado, é utilizada a notação  $X(a, b)$ .

- **Adição:** dois números binários precisam estar na mesma representação e na mesma ordem para serem somados. Isto é, dado que existem dois operandos  $X(c, d)$  e  $Y(e, f)$ , a soma será válida apenas quando  $X = Y$ ,  $c = e$  e  $d = f$ . A soma de dois números  $X(e, f)$  é dada por  $X(e + 1, f)$ , ou seja, o resultado da soma de duas palavras binárias de comprimento  $M$  requer  $M + 1$  bits.
- **Multiplicação sem sinal:**  $U(a, b) \times U(c, d) = U(a + c, b + d)$ .
- **Multiplicação com sinal:**  $A(a, b) \times A(c, d) = A(a + c + 1, b + d)$ .