

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

LUCAS DAMBROS MENIN

PROJETO DE DIPLOMAÇÃO
REDUÇÃO ATIVA DO RUÍDO ACÚSTICO

Porto Alegre
2011

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

REDUÇÃO ATIVA DO RUÍDO ACÚSTICO

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para Graduação em Engenharia Elétrica.

ORIENTADOR: Tiaraju Vasconcellos Wagner

Porto Alegre

2011

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

LUCAS DAMBROS MENIN

REDUÇÃO ATIVA DO RUÍDO ACÚSTICO

Este projeto foi julgado adequado para fazer jus aos créditos da Disciplina de “Projeto de Diplomação”, do Departamento de Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. MSc. Tiaraju Vasconcellos Wagner, UFRGS

Mestre pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Banca Examinadora:

Prof. Dr. Luís Fernando Alves Pereira, UFRGS

Doutor pelo Instituto Tecnológico de Aeronáutica – São José dos Campos, Brasil

Prof. Dr. Marcelo Götz, UFRGS

Doutor pela Universität Paderborn – Paderborn, Alemanha

Prof. MSc. Tiaraju Vasconcellos Wagner, UFRGS

Mestre pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Porto Alegre, julho de 2011

AGRADECIMENTOS

Eu agradeço à minha família, meus pais e meus irmãos, por todo o apoio, compreensão e incentivo que sempre recebi.

Agradeço também ao Professor Tiaraju, que aceitou orientar este trabalho, e que sempre esteve disponível para ajudar.

Agradeço também aos Professores da UFRGS, e a todos aqueles que contribuíram com a minha formação em Engenharia.

Obrigado a todos.

RESUMO

A redução ativa do ruído consiste em gerar um sinal sonoro com a mesma amplitude, porém em oposição de fase ao ruído que se quer eliminar. Neste trabalho foi desenvolvido o projeto de um dispositivo que suprime o ruído na saída de um tubo, simulando instalações industriais de ventilação.

Palavras-chaves: Engenharia Elétrica. Processamento de Sinais. Controle Ativo do Ruído. Eletrônica e Instrumentação.

ABSTRACT

Active noise cancellation is a technique of noise suppression in which unwanted noise is countered by generating another signal with similar properties, except the phase which is inverted. In this project an active noise canceller was devised, that attenuates noise propagating down a tube simulating an industrial ventilation piping.

Keywords: Electrical Engineering. Signal Processing. Active Noise Cancellation. Electronic and Instrumentation.

SUMÁRIO

| | | |
|----------|---|-----------|
| 1 | INTRODUÇÃO | 11 |
| 1.1 | TEMA E OBJETIVO | 12 |
| 1.2 | JUSTIFICATIVA | 13 |
| 1.3 | MÉTODO DE PESQUISA | 13 |
| 1.4 | ESTRUTURA DO TRABALHO | 13 |
| 2 | CONTEXTO DO PROJETO | 14 |
| 2.1 | DELIMITAÇÃO DO TRABALHO | 14 |
| 2.2 | ESPECIFICAÇÕES | 14 |
| 3 | O SOM | 15 |
| 3.1 | EQUAÇÕES DE PROPAGAÇÃO SONORA | 15 |
| 3.1.1 | Verificação do nível de compressibilidade do ar | 15 |
| 3.1.2 | A equação de onda | 16 |
| 3.2 | RUÍDO | 17 |
| 4 | ANÁLISE DE ALTERNATIVAS | 19 |
| 4.1 | BLOQUEIO DA TRANSMISSÃO SONORA | 19 |
| 4.2 | ABSORÇÃO DA ENERGIA SONORA | 20 |
| 4.3 | CONTROLE ATIVO DO RUÍDO | 21 |
| 5 | REDUÇÃO ATIVA DO RUÍDO | 23 |
| 5.1 | CARACTERÍSTICAS DO RUÍDO | 25 |
| 5.2 | ESTRUTURAS DE CONTROLE | 25 |
| 5.2.1 | Microfones e alto-falantes direcionais | 25 |
| 5.2.2 | Sensor insensível às ondas sonoras | 26 |
| 5.2.3 | Utilização de dois microfones | 26 |
| 5.2.4 | Utilização de apenas um microfone | 27 |
| 5.2.5 | Outras estruturas de controle | 28 |
| 6 | ALGORITMOS | 29 |
| 6.1 | DEFINIÇÕES | 29 |
| 6.2 | ALGORITMO DO GRADIENTE | 30 |
| 6.3 | ALGORITMO LEAST-MEAN-SQUARE (LMS) | 32 |
| 6.4 | ALGORITMO LMS A REFERÊNCIAS FILTRADAS (FXLMS) | 34 |
| 7 | MÉTODOS, PROCESSOS E DISPOSITIVOS | 37 |
| 7.1 | ALGORITMOS UTILIZADOS | 37 |
| 7.2 | FLUXOGRAMA | 39 |
| 7.3 | HARDWARE | 40 |
| 7.3.1 | Pré-amplificação | 41 |
| 7.3.2 | Conversão A/D e D/A | 42 |
| 7.3.3 | Amplificador de potência | 45 |
| 7.3.4 | Geração do sinal anti-ruído | 46 |
| 8 | RESULTADOS ALCANÇADOS | 49 |
| 8.1 | RESULTADOS SIMULINK | 49 |
| 8.2 | RESULTADO DA SÍNTESE DE CÓDIGO VHDL | 51 |
| 9 | CONCLUSÃO | 52 |

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 1 Demandas de compensação à ACC por perda auditiva | 12 |
| Figura 2 Espectro do ruído emitido por uma ventoinha | 18 |
| Figura 3 Bloqueio da transmissão sonora..... | 19 |
| Figura 4 Silencioso | 20 |
| Figura 5 Bloqueio da transmissão sonora com material dissipativo | 20 |
| Figura 6 Silencioso com material dissipativo..... | 21 |
| Figura 7 Revestimento..... | 21 |
| Figura 8 Patente de Paul Lueg, 1936..... | 22 |
| Figura 9 Fone de ouvido BOSE QuietComfort 15, GMC Terrain 2011 e ULTRA ANC..... | 22 |
| Figura 10 Esquema básico de controle acústico ativo..... | 23 |
| Figura 11 Múltiplos microfones e fontes de sinal anti-ruído | 24 |
| Figura 12 Microfones e auto-falantes direcionais | 26 |
| Figura 13 Sensores insensíveis às ondas sonoras | 26 |
| Figura 14 Utilização de 2 microfones | 27 |
| Figura 15 Utilização de 1 microfone | 28 |
| Figura 16 Utilização do algoritmo LMS | 33 |
| Figura 17 Impossibilidade de atualização direta dos coeficientes do filtro $W(z,t)$ | 34 |
| Figura 18 Utilização do algoritmo FxLMS | 35 |
| Figura 19 Estimação das funções de transferência no sistema..... | 38 |
| Figura 20 Cancelamento do ruído | 39 |
| Figura 21 Diagrama de estados | 39 |
| Figura 22 Demonstrador..... | 41 |
| Figura 23 Esquemático dos pré-amplificadores | 42 |
| Figura 24 Placa de circuito impresso dos pré-amplificadores..... | 42 |
| Figura 25 Esquemático da seção de filtragem anti-aliasing | 43 |
| Figura 26 Esquemático da placa de interfaces A/D e D/A | 44 |
| Figura 27 Placa de circuito impresso com interfaces A/D e D/A..... | 45 |
| Figura 28 Esquemático dos amplificadores de potência | 46 |
| Figura 29 Placa de circuito impresso dos amplificadores de potência..... | 46 |
| Figura 30 Simulação do algoritmo FxLMS..... | 50 |
| Figura 31 Resultados após 20s de simulação do algoritmo FxLMS | 50 |
| Figura 32 Resultados após 20s de simulação do algoritmo FxLMS | 50 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 Parâmetros do circuito sintetizado..... | 51 |
|--|----|

LISTA DE ABREVIATURAS

A/D: Analog-to-Digital

ANC: Active Noise Cancellation

D/A: Digital-to-Analog

DELET: Departamento de Engenharia Elétrica

FIR: Finite Impulse Response

FPGA: Field-Programmable Gate Array

FxLMS: Filtered-x Least-Mean-Square

LMS: Least-Mean-Square

SAR: Successive Approximation Register

UFRGS: Universidade Federal do Rio Grande do Sul

VHDL: VHSIC Hardware Description Language

1 INTRODUÇÃO

Este relatório apresenta o trabalho realizado durante o primeiro semestre de 2011, na área de redução do ruído acústico. A busca de novas alternativas que contribuam para evitar ou reduzir a surdez justifica-se pela gravidade do problema: atualmente, a perda de audição devido à exposição a altos níveis de ruído é uma das doenças ocupacionais mais comuns segundo (NIOSH, 2001), afetando (antes dos 50 anos) mais de 40% dos trabalhadores em diversas áreas de atuação como mineração, carpintaria e construção civil.

Uma grande parte dos custos ocasionados perda auditiva recai direta e indiretamente sobre a indústria: em um levantamento realizado há 15 anos, (GIRARD, 2007 apud NZ-OSHS, 1996) concluiu que o custo anual médio resultante da exposição de um trabalhador a níveis demasiados de ruído (maiores do que 90 dB) seria de 2364 NZD por trabalhador, equivalentes a cerca de 4,3 mil reais em 2011. Na ocasião, foi feita uma estimativa não só dos custos prontamente mensuráveis devido a indenizações, aquisição de próteses auditivas, ausências no trabalho e queda na produtividade, mas também de custos indiretos como efeitos nefastos sobre os empregados, aumento do risco de acidentes, deterioração da imagem de marca e necessidade de contratação de pessoal com maior frequência do que o normal.

Apesar de que todo o trabalhador possa acabar exposto a níveis excessivos de ruído em seu local de trabalho, há áreas de atividade nas quais tradicionalmente esta exposição é maior do que o normal. Segundo (THORNE, 2008), o maior número de demanda de reparações por perda auditiva na Nova Zelândia, no período de 1995 a 2006, partiu de agricultores, pescadores, operadores de máquina e trabalhadores da construção civil.

Como se pode observar na figura 1, o número de pessoas que receberam compensação financeira pela perda auditiva na Nova Zelândia aumentou nos últimos anos (gráfico publicado pela Accident Compensation Corporation of New Zealand – ACC). Isto pode ser a

consequência tanto de uma maior exposição ao ruído do que no passado, quanto de uma população melhor informada ou de um melhor diagnóstico da perda auditiva.

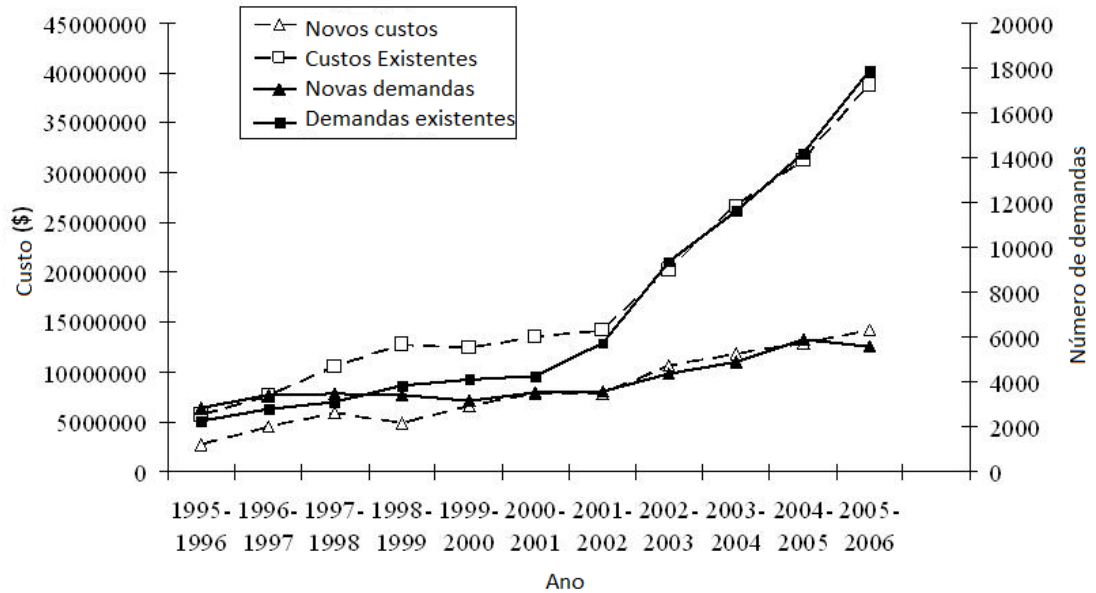


Figura 1 Demandas de compensação à ACC por perda auditiva

Independentemente da razão pela qual estão aumentando os processos judiciais relacionados à perda auditiva na Nova Zelândia, os dados apontam os altos custos para a sociedade da surdez induzida pela exposição ao ruído, um problema de saúde que, em teoria, poderia ser completamente evitável.

Além da motivação social e econômica para reduzir o ruído nas instalações fabris, para as empresas os avanços na redução de ruído são importantes na medida em que possibilitam novos produtos (carros, barcos, sistemas de ventilação, por exemplo) com maior conforto.

1.1 TEMA E OBJETIVO

O objetivo do trabalho desenvolvido é pesquisar as vantagens e limitações da redução ativa do ruído acústico em relação aos métodos passivos.

1.2 JUSTIFICATIVA

Entre as fontes de ruído sonoro intenso às quais a população humana é exposta com mais frequência estão as máquinas e processos na indústria em geral, e os meios de transporte.

Como a poluição sonora tem um custo elevado para a sociedade, justifica-se a pesquisa da redução ativa do ruído especialmente nas aplicações em que nenhuma, ou muito pouca, redução de ruído é utilizada atualmente devido ao custo, volume, ou massa necessária.

1.3 MÉTODO DE PESQUISA

A pesquisa foi iniciada com um levantamento das soluções já existentes para redução do ruído, tanto passivas quanto ativas. Em seguida, os algoritmos para implementação dos métodos ativos foram estudados para se compreender o princípio de funcionamento.

Após a fase inicial de pesquisas na bibliografia, um demonstrador para realizar a atenuação do ruído ao longo de um tubo (uma direção de propagação) foi projetado, incluindo os sistemas de aquisição e processamento do sinal de ruído, e de geração do sinal anti-ruído.

1.4 ESTRUTURA DO TRABALHO

O trabalho, cujo escopo é delimitado no capítulo 2, inicia com uma revisão dos conceitos básicos que regem a propagação sonora (capítulo 3).

No capítulo 4, algumas possibilidades de redução de ruído são brevemente expostas, enquanto que no capítulo 5 as arquiteturas mais simples de redução ativa do ruído são apresentadas. Em seguida, no capítulo 6, são vistos os algoritmos adaptativos que serviram de base para o projeto realizado.

A descrição do projeto realizado, um demonstrador da redução ativa do ruído acústico, pode ser encontrada no capítulo 7, seguida no capítulo 8 dos resultados obtidos.

2 CONTEXTO DO PROJETO

2.1 DELIMITAÇÃO DO TRABALHO

O trabalho trata da redução do ruído acústico proveniente de uma única fonte, com propagação ao longo de uma única direção. Não serão abordadas com profundidade as situações onde possa haver múltiplas fontes de ruído, nem a atenuação do ruído em várias direções.

Ao término das pesquisas para familiarização com os temas da redução de ruído e da filtragem adaptativa, um demonstrador será projetado de modo que possa atenuar ruídos periódicos de baixa frequência em um pequeno duto, simulando a redução do ruído de uma máquina ao longo de um duto de ventilação.

2.2 ESPECIFICAÇÕES

O trabalho inclui o projeto de um protótipo de demonstrador, dispositivo em escala reduzida baseado em sistemas projetados para a redução do ruído de grandes ventiladores em dutos de ventilação.

O sistema deve atenuar ruídos periódicos de frequência compreendida desde 50 Hz até 200 Hz. O sistema deve ainda automaticamente seguir a variação de frequência do ruído incidente, só devendo passar por uma nova regulagem quando as propriedades do ambiente (como a distância dos microfones à fonte de anti-ruído) forem alteradas.

3 O SOM

O som é um fenômeno mecânico, onde ondas longitudinais de pressão se propagam através de um meio gasoso, líquido ou sólido. Neste trabalho apenas o som se propagando através do ar e em condições normais de temperatura e pressão será considerado.

A seguir, algumas características importantes do som serão analisadas com o objetivo de guiar o desenvolvimento de estratégias para minimizar o nível de ruído.

3.1 EQUAÇÕES DE PROPAGAÇÃO SONORA

3.1.1 Verificação do nível de compressibilidade do ar

Considerando a unidade padrão para medida da pressão sonora, o decibel (dB), constata-se que mesmo um ruído muito intenso, como 94 dB, gera no ambiente uma flutuação de pressão muito pequena se comparada à pressão atmosférica (equação 1) (GALLAND, 2009):

$$\frac{p'}{p_0} = \frac{10^{\left(\frac{94\text{dB}}{20\text{dB}}\right)} \times 20 \mu\text{Pa}}{101325 \text{ Pa}} = \frac{1,0024 \text{ Pa}}{101325 \text{ Pa}} \approx 10^{-5} \quad (1)$$

Para se caracterizar o comportamento do ar em presença deste som de 94 dB, por exemplo, deve-se primeiramente calcular o número de Mach. Ele terá um valor muito pequeno (resultado da equação 2), indicando que os efeitos da viscosidade são desprezíveis:

$$M = \frac{v'}{c_0} = \frac{\left(\frac{p'}{\rho_0 c_0}\right)}{c_0} = \frac{p'}{\rho_0 c_0^2} = \frac{p'}{\rho_0 \cdot \left(\gamma \frac{p_0}{\rho_0}\right)} = \frac{p'}{\gamma p_0} = \frac{p'}{1,4 p_0} \approx 10^{-5} \ll 1 \quad (2)$$

O parâmetro γ , coeficiente adiabático, tem o valor de $7/5$ para as moléculas diatômicas como os gases N_2 e O_2 .

De uma forma geral, em acústica sempre se tem $M \ll 1$ (GALLAND, 2009), portanto, os efeitos térmicos e da viscosidade são desprezíveis e o ar comporta-se como um fluido perfeito (incompressível e sem viscosidade). Isto significa que, em condições normais de pressão atmosférica e de nível de pressão sonora, a acústica é um fenômeno essencialmente linear.

3.1.2 A equação de onda

Para se calcular as equações que caracterizam a propagação de ondas sonoras, pode-se inicialmente considerar o ar ambiente (na ausência de sons) como sendo estacionário, homogêneo e em repouso. Neste caso, tem-se que a pressão e a densidade do fluido são constantes, e a sua velocidade é nula (equações 3):

$$\begin{aligned} p_0 &= cte \\ \rho_0 &= cte \\ \vec{v}_0 &= \vec{0} \end{aligned} \tag{3}$$

Em presença de uma onda acústica, ocorre uma ligeira perturbação no fluido, resultando no sistema de equações 4:

$$\begin{aligned} p(\vec{x}, t) &= p_0 + p'(\vec{x}, t) \\ \rho(\vec{x}, t) &= \rho_0 + \rho'(\vec{x}, t) \\ s(\vec{x}, t) &= s_0 + s'(\vec{x}, t) \\ \vec{v}(\vec{x}, t) &= \vec{0} + \vec{v}'(\vec{x}, t) \end{aligned} \tag{4}$$

Considerando apenas os termos lineares (o som é um fenômeno linear) e desprezando os termos produto, as equações 4 podem ser manipuladas para dar origem a um sistema de três equações (equações 5):

$$\begin{aligned}
\frac{\partial \rho'}{\partial t} + \rho_0 \operatorname{div}(\vec{v}') &= 0 \\
\rho_0 \frac{\partial \vec{v}'}{\partial t} &= -\vec{\nabla} p' \\
p' &= c_0^2 \rho'
\end{aligned} \tag{5}$$

Este sistema pode ser resumido em uma equação geral para as ondas acústicas (equação 6):

$$\Delta p' - \frac{1}{c_0^2} \frac{\partial^2 p'}{\partial t^2} = 0 \tag{6}$$

Se tratando de ondas unidimensionais, a equação 6 pode ser escrita como

$$\frac{\partial^2 p'}{\partial x_1^2} - \frac{1}{c_0^2} \frac{\partial^2 p'}{\partial t^2} = 0, \text{ cuja solução é } p'(\vec{x}_1, t) = f\left(t - \frac{x_1}{c_0}\right) + g\left(t + \frac{x_1}{c_0}\right).$$

As funções $f(t, x_1)$ e $g(t, x_1)$ são funções arbitrárias, correspondentes à parte progressiva (propagação no sentido de $+x_1$) e à parte regressiva (propagação no sentido de $-x_1$) da onda, e determinadas pelas condições iniciais específicas de cada problema a tratar.

Sendo a acústica um fenômeno linear, em teoria é possível utilizar a superposição de uma ou mais ondas sonoras, precisamente em oposição de fase ao ruído, para silenciar o ambiente. Esta idéia foi originalmente proposta por (PAUL LUEG, 1936), e após grandes avanços trazidos pela aplicação dos conceitos de filtragem adaptativa, atualmente (2011) já existem veículos de série onde o controle ativo do ruído é empregado para aumentar o conforto dos passageiros (GMC, 2011).

3.2 RUÍDO

Usualmente se caracteriza o ruído acústico como sendo o conjunto dos sons indesejáveis, que possuem caráter desagradável ou perturbador. No Brasil, a norma ABNT

NBR 10151 define as medições a serem realizadas para se fazer a caracterização do nível de ruído em edificações.

Para atender à legislação, e também para obter vantagens competitivas face à concorrência (disponibilizar um veículo com cabine mais silenciosa, por exemplo), é importante que o nível de ruído seja levado em conta desde a fase de concepção de qualquer veículo, máquina ou instalação fabril.

De acordo com seu conteúdo espectral de potência, o ruído acústico pode ser considerado de banda estreita (como ruídos periódicos gerados por máquinas rotativas), de larga banda (como o ruído do contato pneu/asfalto em um automóvel), ou ainda combinar as duas características, como no caso do ruído emitido por uma ventoinha de servidores em um data-center (figura 2) (MINOGUE, 2000):

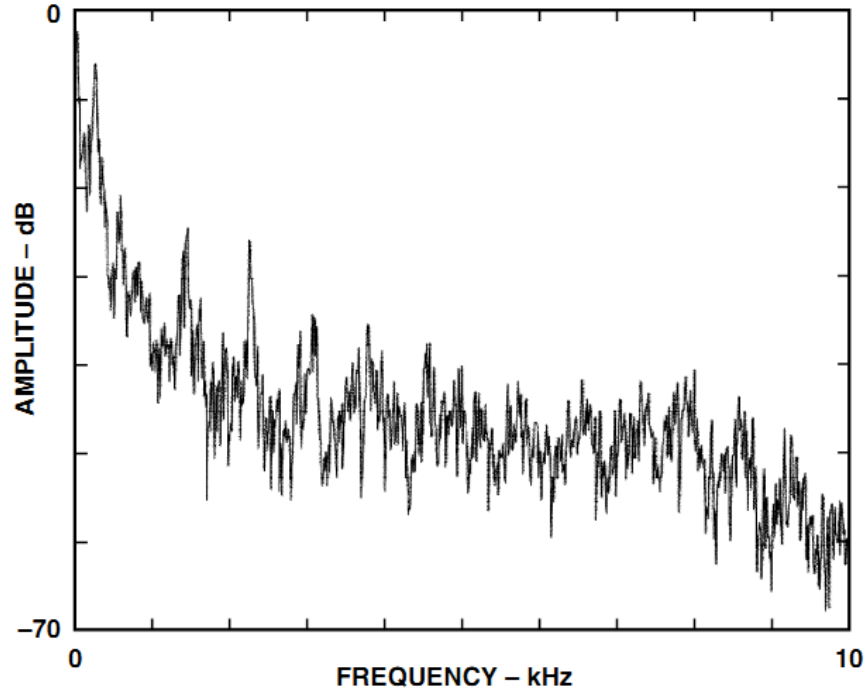


Figura 2 Espectro do ruído emitido por uma ventoinha

4 ANÁLISE DE ALTERNATIVAS

A seguir serão apresentados alguns métodos usualmente empregados para reduzir o ruído acústico, de modo que se possa conhecer as suas vantagens e desvantagens frente às outras opções.

4.1 BLOQUEIO DA TRANSMISSÃO SONORA

O isolamento das fontes de ruído através da utilização de painéis e coberturas, impedindo a propagação direta das ondas sonoras, é o modo mais simples de redução do ruído. Não é um método eficaz para bloquear os ruídos de baixa frequência (tipicamente as dimensões e a massa dos elementos envolvidos tornam-se impraticáveis), além de exigir o emprego de um material denso e de poder induzir vibrações na estrutura utilizada para o isolamento acústico.

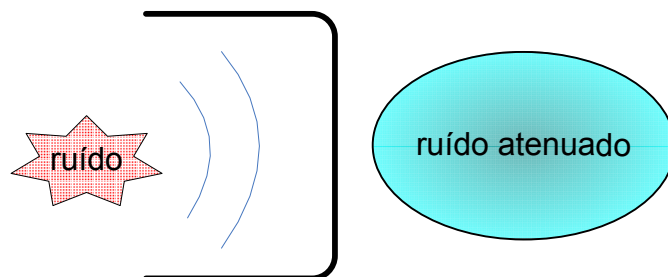


Figura 3 Bloqueio da transmissão sonora

É possível, também, eliminar a propagação de uma única frequência sonora e de suas harmônicas em dutos através do emprego de silenciosos (figura 4). O comprimento do silencioso define a frequência fundamental a ser suprimida, e a razão entre as seções do silencioso define a atenuação que pode ser obtida.

Apêndices perpendiculares aos dutos e calculados para eliminar uma frequência fundamental e suas múltiplas também podem ser empregados mas, como no caso dos silenciosos, uma vez instalados geralmente é muito difícil modificar as frequências onde a

função de transferência é anulada (para se adaptar à instalação de uma nova máquina que funcione em uma velocidade diferente, por exemplo) (MÖSER, 2009).

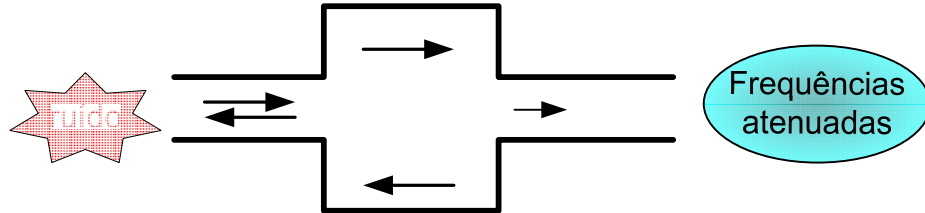


Figura 4 Silencioso

4.2 ABSORÇÃO DA ENERGIA SONORA

A adição de materiais dissipativos à estrutura de painéis e coberturas (figura 5) permite uma atenuação maior do ruído, especialmente em altas frequências. O ruído de baixa frequência normalmente não é atenuado através deste método, pois isto exigiria uma espessura da ordem de grandeza dos comprimentos de onda envolvidos, grande demais para a maioria das aplicações.

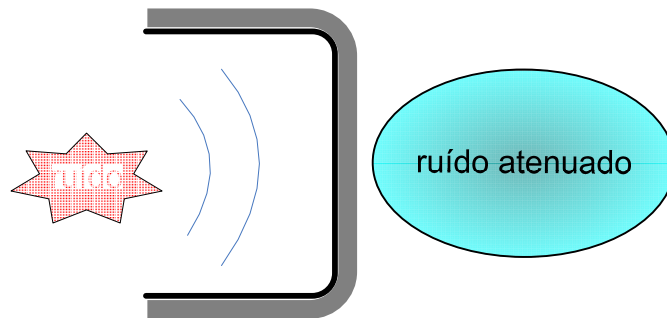


Figura 5 Bloqueio da transmissão sonora com material dissipativo

Em dutos é possível atenuar gradualmente o som com o emprego de material dissipativo ao longo da parede interna (figura 6).

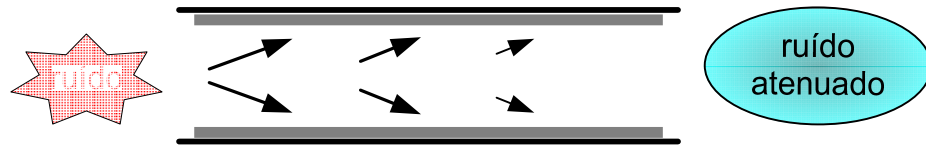


Figura 6 Silencioso com material dissipativo

Quando há envolvimento total da fonte de ruído como mostrado na figura 7, o material dissipativo contribui não só para a redução do nível de ruído propagado diretamente, mas também ajuda a atenuar a reverberação.

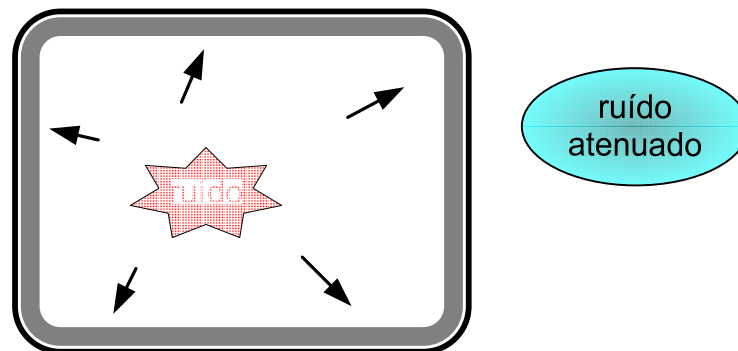


Figura 7 Revestimento

A maioria dos métodos de atenuação passivos do ruído exige adição de material (que muitas vezes não terá outra função senão a redução de ruído) ocasionando um aumento nos custos de produção e de transporte devido ao maior volume e à maior massa.

Os métodos passivos de redução do ruído não são adequados para atenuar as baixas frequências (com grande comprimento de onda) por causa da necessidade de se utilizar espessuras muito grandes para que a absorção do som seja significativa, mas em contrapartida podem ser muito efetivos em altas frequências se os materiais adequados forem empregados (MÖSER, 2009).

4.3 CONTROLE ATIVO DO RUÍDO

Para eliminar os ruídos indesejados, busca-se gerar um sinal artificial (anti-ruído) de mesma frequência e amplitude, porém em oposição de fase ao ruído original, causando assim

interferência destrutiva das ondas sonoras e conseqüente atenuação do ruído percebido. Este método foi originalmente proposto por (PAUL LUEG, 1936) e é particularmente útil na atenuação de ruídos de baixa frequência, porque assim a realização prática da oposição de fase é facilitada pelo maior comprimento de onda, além de o volume onde ocorre a redução de ruído ser maior. Este será o método de atenuação do ruído empregado neste trabalho.

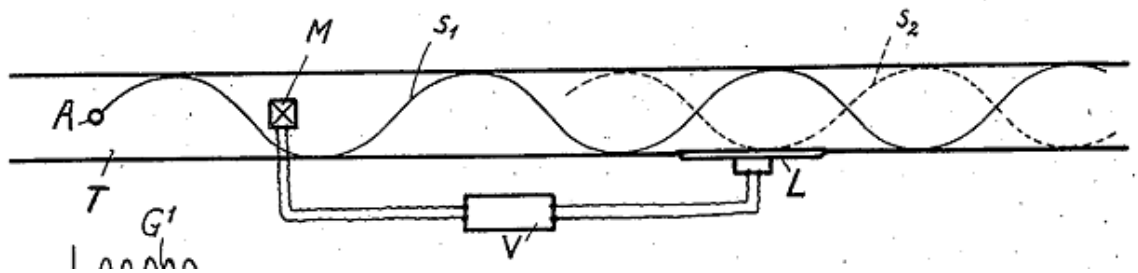


Figura 8 Patente de Paul Lueg, 1936

Alguns exemplos de aplicações comerciais do cancelamento ativo de ruído são fones auriculares (BOSE, 2011), veículos de passeio onde os auto-falantes na cabine cancelam o ruído do motor quando em baixa rotação (GMC, 2011), e sistemas adaptados a aeronaves com múltiplos microfones e atuadores (alto-falantes e/ou shakers na fuselagem) (ULTRA, 2011). Na figura 9 pode-se observar parte do material de divulgação desses três sistemas.

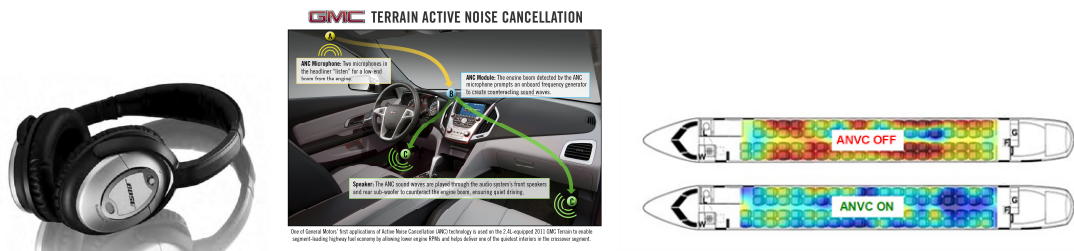


Figura 9 Fone de ouvido BOSE QuietComfort 15, GMC Terrain 2011 e ULTRA ANC

5 REDUÇÃO ATIVA DO RUÍDO

O esquema básico de controle acústico ativo está delineado na figura 10, consistindo na medida do ruído e na geração de um sinal de anti-ruído.

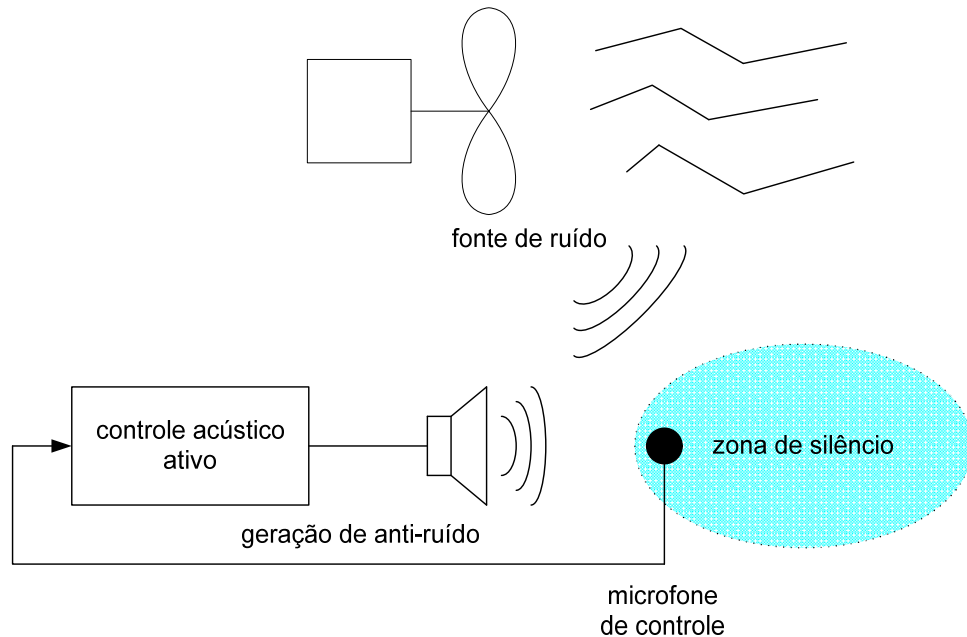


Figura 10 Esquema básico de controle acústico ativo

Através da medição do ruído (e, em função da aplicação, também dos sons que devem ser preservados), ondas sonoras de anti-ruído são geradas de modo a interferir destrutivamente com as ondas sonoras de ruído em certo volume do espaço, estabelecendo uma zona de silêncio. O anti-ruído pode ser gerado através do processamento do sinal do próprio ruído, ou sintetizado através de um sinal de referência com a mesma frequência e fase que o ruído.

Como o anti-ruído não poderá interferir destrutivamente com o ruído em todos os pontos do espaço, deve-se observar que em algumas regiões fora da zona de controle o ruído percebido aumentará devido às interferências construtivas, podendo chegar ao dobro da amplitude do ruído inicial (quando não estava em funcionamento o controle ativo).

É importante observar que o volume da zona de controle será função do comprimento de onda do ruído: para ruídos de baixa frequência, a zona de silêncio pode se estender por vários metros, ao passo que para ruídos de frequência elevada os limites da zona de silêncio situam-se tão próximos do microfone de erro que o emprego de controle ativo do ruído deixa de fazer sentido, sendo mais eficaz recorrer às técnicas passivas para a atenuação dos sons indesejados. Uma das possibilidades para diminuir os problemas do controle ativo do ruído em altas frequências é a estimação do ruído através de microfones fora da zona de silêncio. Dessa maneira, uma zona de silêncio pequena pode ser aceitável desde que seja posicionável no espaço (acompanhando a cabeça do usuário com ajuda de um sistema de câmeras de vídeo, por exemplo). (KESTELL, 2000)

Para aumentar o volume da zona de silêncio, é necessário o emprego de vários microfones de controle e de várias fontes secundárias de sinal anti-ruído. Atualmente as aplicações comerciais de redução do ruído em aeronaves utilizam vários microfones e auto-falantes simultaneamente, em um arranjo similar à figura 11. (GALLAND, 2009).

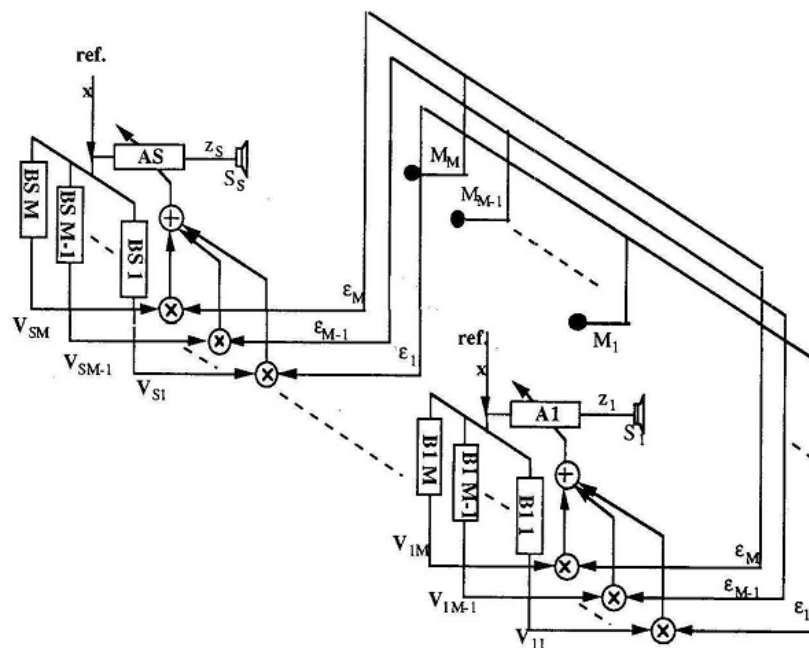


Figura 11 Múltiplos microfones e fontes de sinal anti-ruído

5.1 CARACTERÍSTICAS DO RUÍDO

Uma característica comum a muitos sistemas de controle ativo do ruído é a falta de certeza, sobre as características (potência e distribuição espectral, por exemplo) que o ruído poderá ter durante a operação do sistema (ABRANTES, 2000). Além disso, é muito provável que o ruído acústico seja estatisticamente variável no tempo, o que significa que mesmo que um filtro ideal fosse regulado para perfeitamente cancelar o ruído acústico em alguma aplicação específica, seria impossível garantir que este dispositivo continuasse a operar de maneira ideal sem poder se adaptar às variações do ambiente e do sinal de ruído.

Estas constatações exigem o emprego de uma família de filtros capazes de se ajustar ao ambiente de operação, ditos filtros adaptativos. Através de filtros adaptativos, o dispositivo de redução de ruído poderá se ajustar continuamente às variações do ambiente e do ruído indesejável, buscando sempre o desempenho ótimo em relação a algum critério pré-determinado.

5.2 ESTRUTURAS DE CONTROLE

Proposto inicialmente por (PAUL LUEG 1936), o controle ativo do ruído é hoje geralmente realizado através de filtros digitais. O emprego de filtros analógicos é somente recomendável quando as dimensões do sistema exigem um tempo muito curto de propagação dos sinais através do filtro (como a supressão de ruído de larga banda em fones auriculares). Quando realizado digitalmente, o controle ativo do ruído existe em quatro formas principais, apresentadas a seguir.

5.2.1 Microfones e alto-falantes direcionais

Microfones e/ou auto-falantes altamente direcionais (figura 12) permitem atenuar a realimentação acústica. Esta técnica segue diretamente as idéias propostas por (PAUL LUEG 1936), porém é de custo muito elevado e freqüentemente oferece uma performance é limitada (UOSUKAINEN, 1998).

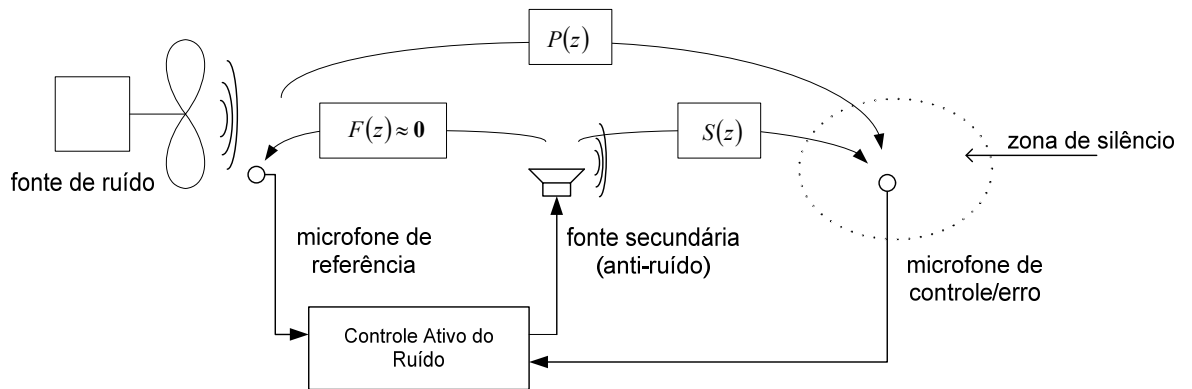


Figura 12 Microfones e auto-falantes direcionais

5.2.2 Sensor insensível às ondas sonoras

Nos casos em que um sinal elétrico correlacionado com o ruído estiver prontamente disponível, ele poderá ser utilizado como referência, evitando assim o uso de microfones e a realimentação acústica (figura 13) (GALLAND, 2009).

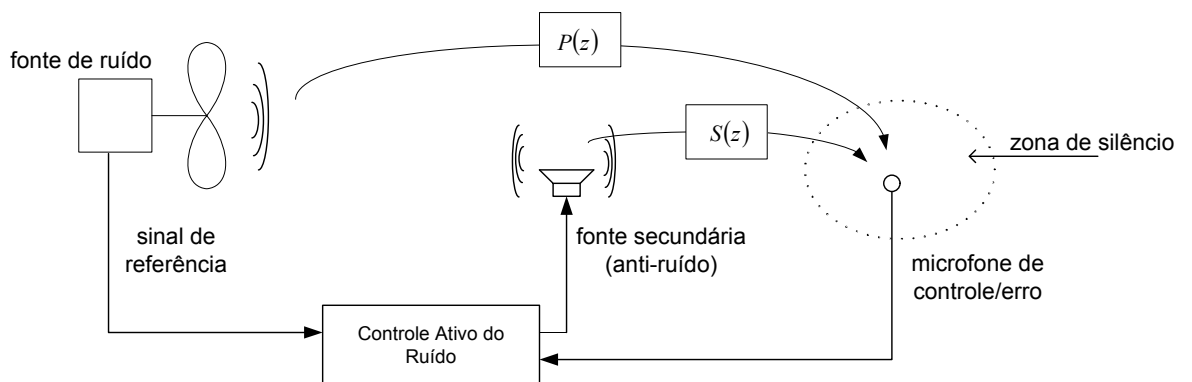


Figura 13 Sensores insensíveis às ondas sonoras

5.2.3 Utilização de dois microfones

É possível utilizar dois microfones – um para obter o sinal de referência, outro para obter o sinal de erro – e, a partir de filtros previamente determinados para estimar o caminho de retorno $\hat{F}(z)$ e o caminho secundário $\hat{S}(z)$, eliminar o sinal anti-ruído que chegar ao microfone de referência após ter atravessado $F(z)$ e se misturando com o ruído original $r(n)$, obtendo-se assim uma medida do ruído $r(n)$, como apresentado na figura 14 (AKHTAR, 2007; GALLAND, 2009).

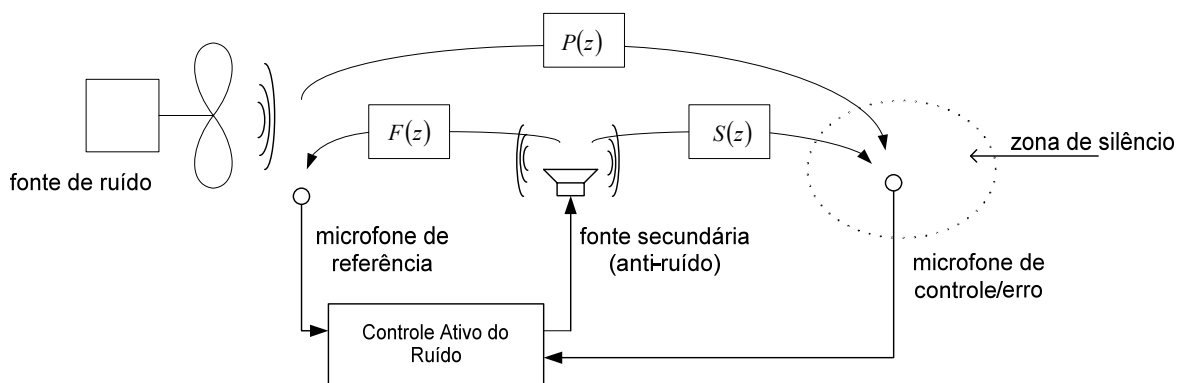


Figura 14 Utilização de 2 microfones

Neste trabalho, esta foi a alternativa escolhida para o desenvolvimento do protótipo demonstrador, pois é mais flexível do que as outras técnicas, permitindo eliminar ruídos periódicos e não-periódicos. Como desvantagem pode-se citar o maior poder de processamento exigido.

5.2.4 Utilização de apenas um microfone

Utilizando um único microfone situado na zona de silêncio, o microfone de erro, é possível eliminar apenas ruídos periódicos (figura 15). Este é o método com implementação mais compacta, útil para aplicações em fones de ouvido (CHANG, 2011). Continua sendo necessária a identificação prévia do caminho secundário, o que pode ser feito ainda na fábrica.

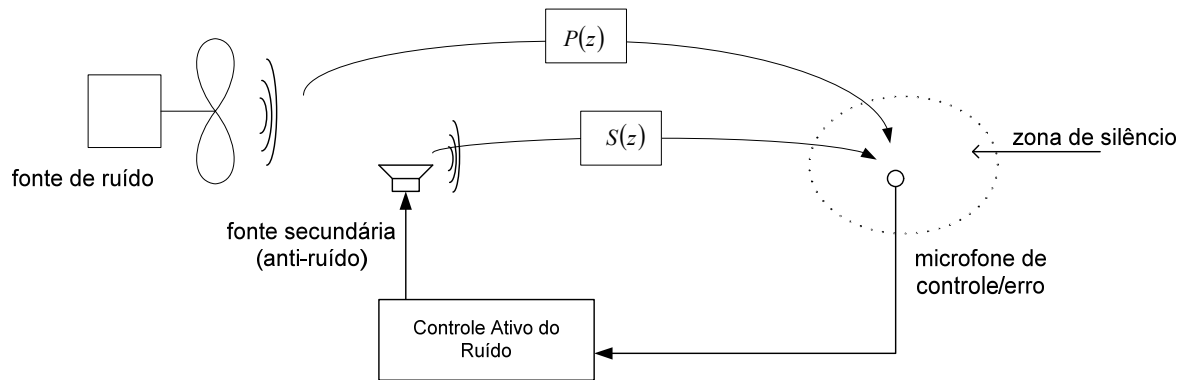


Figura 15 Utilização de 1 microfone

5.2.5 Outras estruturas de controle

Existem aplicações do controle ativo do ruído em que há dois laços de controle trabalhando em paralelo: um filtro digital adaptativo encarrega-se de eliminar os ruídos periódicos, ao passo que um filtro analógico realiza a atenuação dos componentes aleatórios do ruído. (MINOGUE, 2000)

6 ALGORITMOS

Neste capítulo serão apresentados o algoritmo do gradiente, o algoritmo Least-Mean-Square (LMS) e o algoritmo LMS a referências filtradas (FxLMS). As deduções são baseadas em (ABRANTES, 2000; WIDROW, 1985)

O algoritmo LMS foi utilizado neste trabalho para atualizar o filtro de resposta ao impulso finita (FIR) que estima a resposta do caminho secundário $\hat{S}(z)$ e para atualizar o filtro FIR que estima a resposta do caminho de realimentação acústica $\hat{F}(z)$. O algoritmo FxLMS foi utilizado para atualizar o filtro adaptativo $W(z)$. Ambos os algoritmos, LMS e FxLMS, são baseados no algoritmo do gradiente.

6.1 DEFINIÇÕES

Considerando um filtro FIR transversal com N coeficientes, pode-se definir o vetor dos dados de entrada como sendo $\mathbf{a}(n) = [a(n) \ a(n-1) \ a(n-2) \ \dots \ a(n-N+1)]^T$.

Supondo então que todos os elementos de $\mathbf{a}(n)$ sejam reais, a matriz de autocorrelação de entrada, $\mathbf{R}(n)$, poderá definida como

$$\mathbf{R}(n) = E[\mathbf{a}(n)\mathbf{a}(n)^T] = E \begin{bmatrix} a^2(n) & a(n)a(n-1) & \dots & a(n)a(n-N+1) \\ a(n-1)a(n) & a^2(n) & \dots & a(n-1)a(n-N+1) \\ \vdots & \vdots & \ddots & \vdots \\ a(n-N+1)a(n) & a(n-N+1)a(n-1) & \dots & a^2(n-N+1) \end{bmatrix}$$

O vetor dos coeficientes do filtro FIR adaptativo no instante n pode ser escrito como $\mathbf{c}(n) = [c(n) \ c(n-1) \ c(n-2) \ \dots \ c(n-N+1)]^T$ e, por fim, o vetor de correlação cruzada entre a resposta desejada $d(n)$ e a seqüência $\mathbf{a}(n)$ é $\mathbf{p}(n) = E[d(n)\mathbf{a}(n)]$.

Observa-se que a matriz $\mathbf{R}(n)$ e o vetor $\mathbf{p}(n)$, apesar de variáveis no tempo, são comumente designados como apenas \mathbf{R} e \mathbf{p} na literatura (ABRANTES, 2000), deixando implícita sua dependência em relação ao instante n .

A saída do filtro FIR será, a cada instante n , $y(n) = \mathbf{c}^T \mathbf{a}(n)$.

Caso se considere o filtro que minimize o erro quadrático médio entre a resposta desejada $d(n)$ e a saída $y(n)$, a ele se dá o nome de filtro de Wiener. Este é o filtro ótimo no sentido do erro quadrático médio quando os sinais de entrada são estacionários (ou seja, quando as características estatísticas de $\mathbf{a}(n)$ não mudam) (WIDROW, 1985).

6.2 ALGORITMO DO GRADIENTE

O algoritmo do gradiente busca a minimização do erro quadrático médio entre a saída atual do filtro e a saída desejada do filtro.

Supondo que possa existir o filtro de Wiener com dados coeficientes fixos $\mathbf{c} = [c_0 \ c_1 \ \dots \ c_{N-1}]^T$, busca-se evoluir os coeficientes do filtro FIR na direção que mais reduza o erro quadrático médio.

Sendo ε o erro quadrático médio e $e(n) = d(n) - \mathbf{c}^T \mathbf{a}(n)$ a diferença entre o valor desejado e a saída do filtro, tem-se que ε é dado pela equação 7:

$$\begin{aligned} \varepsilon &= E[e^2(n)] = E\{[d(n) - y(n)]^2\} = E[d^2(n)] + E[y^2(n)] - 2E[d(n)y(n)] \\ \varepsilon &= E[d^2(n)] + \mathbf{c}^T \mathbf{R} \mathbf{c} - 2\mathbf{c}^T \mathbf{p} \end{aligned} \quad (7)$$

Se a resposta desejada possuir média nula, como os sinais de áudio, a parcela $E[d^2(n)]$ representará a sua variância.

A equação (de ε) é uma superfície hiperparabolóide de $N + 1$ dimensões, sem mínimos locais, comumente chamada de superfície de performance do erro. Caso a entrada

seja estacionária (com estatísticas invariantes no tempo), a superfície de performance será fixa. Se a entrada tiver os seus parâmetros estatísticos variantes no tempo, então a superfície de performance se deslocará também.

No fundo da superfície de performance, onde o erro quadrático médio é nulo, sabe-se que o gradiente de ε também é nulo (pois a superfície tem o formato de um hiperparabolóide). O gradiente de ε é dado pela equação (8):

$$\nabla = \text{grad}[\varepsilon] = \text{grad} E[e^2(n)] = \frac{\partial E[e^2(n)]}{\partial \mathbf{c}} = -2\mathbf{p} + 2\mathbf{R}\mathbf{c} \quad (8)$$

O valor ótimo de \mathbf{c} ocorre quando o gradiente de ε for nulo (equação 9):

$$\nabla = -2\mathbf{p} + 2\mathbf{R}\mathbf{c} = 0 \quad (9)$$

ou seja, o vetor dos coeficientes ótimos é $\mathbf{c}_{opt} = \mathbf{R}^{-1}\mathbf{p}$, equação que é conhecida como a equação de Wiener-Hopf.

Para evitar o cálculo da matriz inversa \mathbf{R}^{-1} e do vetor de correlação cruzada \mathbf{p} , pode-se empregar um procedimento iterativo. Partindo de qualquer ponto na superfície de performance, os coeficientes do filtro FIR, $\mathbf{c}(n)$, são atualizados conforme a equação 10:

$$\mathbf{c}(n+1) = \mathbf{c}(n) - \mu \nabla(n) \quad (10)$$

Visto que $e(n) = d(n) - \mathbf{c}^T \mathbf{a}(n)$, o gradiente é (equação 11):

$$\nabla(n) = \frac{\partial \varepsilon(n)}{\partial \mathbf{c}(n)} = \frac{\partial E[e^2(n)]}{\partial \mathbf{c}(n)} = E \left[\frac{\partial e^2(n)}{\partial \mathbf{c}(n)} \right] = E \left[2e(n) \frac{\partial e(n)}{\partial \mathbf{c}(n)} \right] = -2E[e(n)\mathbf{a}(n)] \quad (11)$$

Logo, a equação 12 define como atualizar os coeficientes do filtro FIR:

$$\mathbf{c}(n+1) = \mathbf{c}(n) + 2\mu E[e(n)\mathbf{a}(n)] \quad (12)$$

A constante μ , presente na equação 12, é uma constante de proporcionalidade positiva, chamada de passo de adaptação. Dela depende o desempenho do algoritmo: uma constante μ “pequena” levará a uma convergência lenta em direção ao vetor de coeficientes ótimo \mathbf{c}_{opt} . Por outro lado, o valor de μ deve sempre satisfazer a condição necessária e suficiente para a convergência do algoritmo:

$$0 < \mu < \frac{1}{\lambda_{\max}}, \text{ onde } \lambda_{\max} \text{ é o maior autovalor da matriz } \mathbf{R}.$$

Como \mathbf{R} é uma matriz positiva definida, uma condição mais simples de satisfazer na prática é fazer com que μ seja menor do que o inverso do traço da matriz \mathbf{R} (equação 14):

$$0 < \mu < \frac{1}{\text{tr}[\mathbf{R}]} \quad (14)$$

6.3 ALGORITMO LEAST-MEAN-SQUARE (LMS)

Uma simplificação dos cálculos necessários no algoritmo do gradiente pode ser obtida se forem feitas algumas hipóteses simplificativas:

Hipótese 1: os vetores de dados de entrada a cada instante ($\mathbf{a}(1), \mathbf{a}(2), \dots, \mathbf{a}(n)$) formam uma seqüência de valores estatisticamente independentes.

Hipótese 2: no instante n a resposta desejada $d(n)$ é estatisticamente independente de todas as suas amostras prévias, dependendo somente do vetor de dados de entrada atual $\mathbf{a}(n)$.

Hipótese 3: $d(n)$ e $\mathbf{a}(n)$ são variáveis aleatórias conjuntas gaussianas.

Estas hipóteses não são rigorosamente verdadeiras, porém simplificam o tratamento matemático permitindo obter uma estimativa, $\hat{\mathbf{v}}(n)$, mais simples para o gradiente $\nabla(n)$ dando origem ao algoritmo Least-Mean-Square (LMS) (equações 15):

$$\hat{\nabla}(n) = \frac{\partial e^2(n)}{\partial \mathbf{c}(n)} = \begin{bmatrix} \frac{\partial e^2(n)}{\partial c_0(n)} \\ \frac{\partial e^2(n)}{\partial c_1(n)} \\ \vdots \\ \frac{\partial e^2(n)}{\partial c_{N-1}(n)} \end{bmatrix} = 2e(n) \begin{bmatrix} \frac{\partial e(n)}{\partial c_0(n)} \\ \frac{\partial e(n)}{\partial c_1(n)} \\ \vdots \\ \frac{\partial e(n)}{\partial c_{N-1}(n)} \end{bmatrix} = -2e(n)\mathbf{a}(n) \quad (15)$$

Logo, o vetor de coeficientes do filtro FIR pode ser atualizado de acordo com a equação 16:

$$\mathbf{c}(n+1) = \mathbf{c}(n) + 2\mu e(n)\mathbf{a}(n) \quad (16)$$

Para garantir a convergência, o passo de adaptação μ deve continuar a satisfazer a condição $0 < \mu < \frac{1}{\text{tr}[\mathbf{R}]}$.

Devido à estimativa inexata $\hat{\nabla}(n)$ do gradiente $\nabla(n)$, os coeficientes $\mathbf{c}(n)$ nunca irão se tornar os coeficientes do filtro de Wiener \mathbf{c}_{opt} : inicialmente convergirão na direção de \mathbf{c}_{opt} , e então passarão a oscilar em torno dos valores de \mathbf{c}_{opt} . É possível provar que quanto maior for μ , maior será o “ruído dos coeficientes” $\mathbf{c}(n)$ após a convergência (em contrapartida, um valor maior para μ geralmente resulta em uma convergência mais rápida). (ABRANTES, 2000).

Na figura 16 está representado o funcionamento do algoritmo LMS.

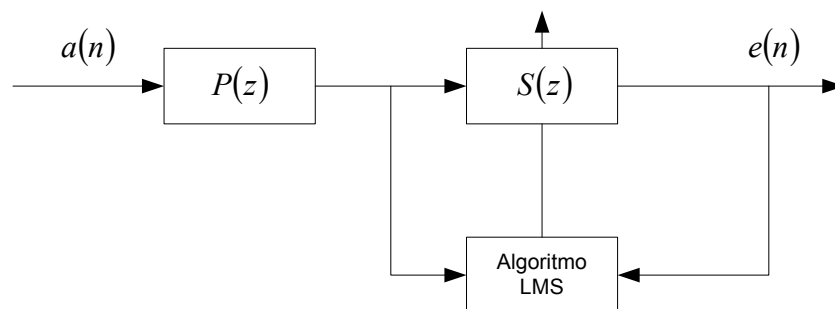


Figura 16 Utilização do algoritmo LMS

6.4 ALGORITMO LMS A REFERÊNCIAS FILTRADAS (FXLMS)

Em aplicações de cancelamento ativo do ruído acústico, o algoritmo LMS não pode ser utilizado diretamente para atualizar o filtro FIR gerador do sinal anti-ruído (figura 17). Isto ocorre porque o sinal de erro $e(n)$ não é captado logo após a saída do filtro adaptativo, mas somente após ser modificado pelo caminho secundário $S(z)$ (o caminho secundário compreende no mínimo a conversão D/A, pré-amplificação, amplificação, irradiação por um transdutor, propagação até o microfone de erro, captação pelo microfone de erro, pré-amplificação e conversão A/D). Se o sinal no microfone de erro fosse utilizado diretamente para atualizar os coeficientes do filtro adaptativo, isso provavelmente faria com que os coeficientes divergissem – e mesmo que não diverjam, convergiriam para valores diferentes dos coeficientes ótimos \mathbf{c}_{opt} (do filtro de Wiener). (WIDROW, 1985)

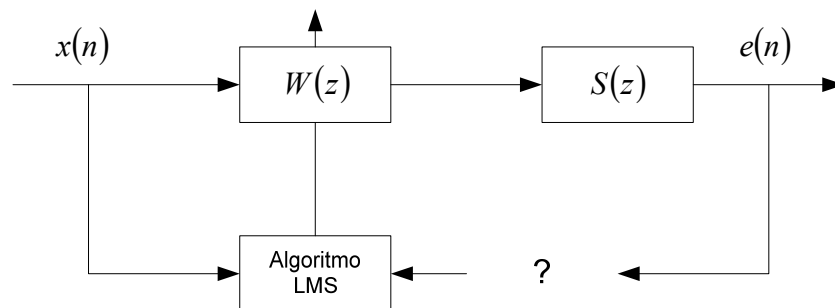


Figura 17 Impossibilidade de atualização direta dos coeficientes do filtro $W(z,t)$

É possível, segundo (WIDROW, 1985), utilizar o sinal no microfone de erro para atualizar os coeficientes do filtro adaptativo, desde que algumas condições sejam satisfeitas:

Condição 1: a planta $S(z)$ deve ser linear e invariante no tempo.

Condição 2: a adaptação do filtro adaptativo $W(z,t)$ deve ser suficientemente lenta em comparação com as constantes de tempo do próprio filtro e da planta $S(z)$.

Normalmente a primeira condição pode ser considerada como satisfeita pela planta, devido à linearidade do fenômeno de propagação do som em condições normais de pressão atmosférica e sonora.

O filtro adaptativo não é linear nem invariante no tempo, logo não atende à primeira condição. No entanto, se a constante de tempo de adaptação for lenta em comparação com as constantes de tempo da planta e do filtro combinadas, então a sua resposta impulsiva vai variar com uma constante de tempo longa em comparação com a constante de tempo combinada da planta e do filtro. Neste caso, basta que o filtro adaptativo $W(z, t)$ atenda à segunda condição (adaptação lenta) para que seja considerado linear e comutável com o caminho secundário $S(z)$.

Uma vez que $W(z, t)$ e $S(z)$ são considerados comutáveis, pode-se modificar o sinal de referência para que também seja modificado pelo caminho secundário – neste caso, uma estimativa $\hat{S}(z)$ do caminho secundário (figura 18):

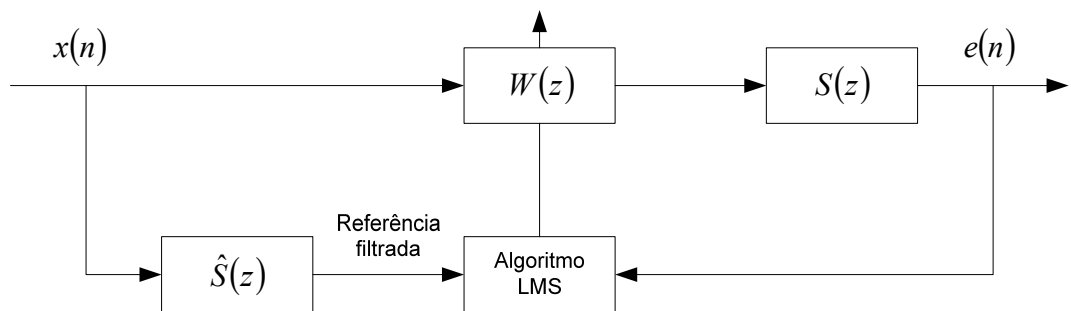


Figura 18 Utilização do algoritmo FxLMS

Neste trabalho, o algoritmo FxLMS será utilizado para calcular as atualizações dos coeficientes do filtro $W(z, t)$, visto que o seu sinal de erro $e(t)$ será captado pelo microfone de erro apenas após o anti-ruído ter atravessado todo o caminho secundário $S(z)$.

Na adaptação dos coeficientes dos filtros para estimar o caminho secundário $\hat{S}(z)$ e a realimentação acústica $\hat{F}(z)$ poderá ser utilizado diretamente o algoritmo LMS: nestes dois casos o sinal de erro estará disponível para o algoritmo logo após a saída dos filtros $\hat{S}(z)$ e $\hat{F}(z)$ ter sido calculada.

A estimação do caminho secundário $\hat{S}(z)$ não precisa ser exata (WIDROW, 1985), pois eventuais diferenças em relação à verdadeira $S(z)$ serão compensadas pela adaptação do filtro $W(z)$.

7 MÉTODOS, PROCESSOS E DISPOSITIVOS

Com o objetivo de ilustrar a aplicação do controle ativo do ruído, um sistema de demonstração foi projetado. Este sistema é composto de um duto (por onde há ruído se propagando), com uma abertura lateral para a introdução do sinal anti-ruído. Microfones captam o ruído e o anti-ruído, servindo como referência para filtros digitais adaptativos.

7.1 ALGORITMOS UTILIZADOS

É necessário eliminar a contribuição do sinal anti-ruído presente no microfone de referência ($y_f(n) = y(n) * f(n)$) e no microfone de erro ($y_s(n) = y(n) * s(n)$), caso contrário o filtro adaptativo $W(z)$ (que processa o ruído para gerar o sinal anti-ruído) não poderá convergir. Para tanto, deve-se estimar $F(z)$ e $S(z)$, o que é feito injetando ruído branco na fonte secundária e identificando as respectivas funções de transferência através de dois filtros adaptativos, $\hat{F}(z)$ e $\hat{S}(z)$.

A adaptação de $\hat{F}(z)$ e $\hat{S}(z)$ é feita utilizando-se o algoritmo LMS. Durante este processo, o filtro $W(z)$ permanece desativado (figura 19). Idealmente a identificação da realimentação acústica deve ser feita sem a presença do ruído externo $r(n)$, visto que ele não é atenuado no microfone de referência e perturba a convergência do filtro $\hat{F}(z)$.

É possível modificar o sistema para que $\hat{F}(z)$ e $\hat{S}(z)$ convirjam mais rapidamente através do emprego de filtros preditores (AKHTAR, 2007 apud KUO, 2002), inclusive na presença de forte ruído $r(n)$, porém esta possibilidade não foi explorada no presente trabalho.

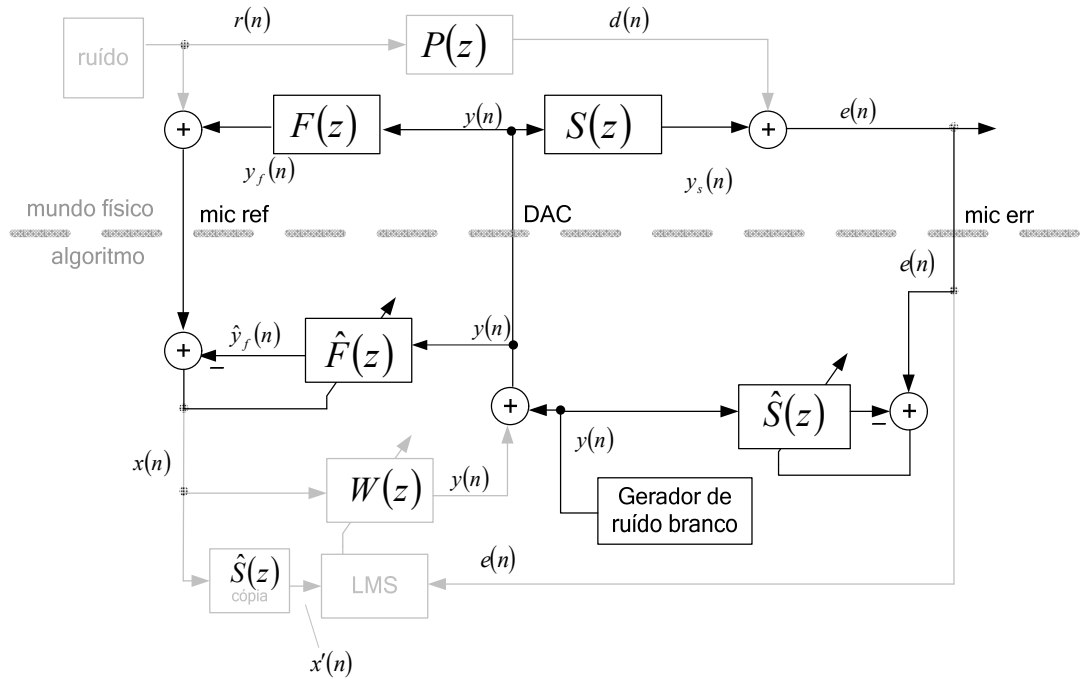


Figura 19 Estimação das funções de transferência no sistema

Após a convergência dos filtros $\hat{F}(z)$ e $\hat{S}(z)$, a fonte de ruído branco é desativada e o filtro adaptativo $W(z)$ inicia a sua adaptação. Enquanto as funções de transferência do sistema não mudarem significativamente, os coeficientes dos filtros $\hat{F}(z)$ e $\hat{S}(z)$ poderão ser mantidos fixos. Após iterações suficientes, e se o passo de adaptação estiver correto, o filtro $W(z)$ convergirá para uma resposta tal que fará com que o ruído $r(n)$ seja cancelado na região próxima ao microfone de erro.

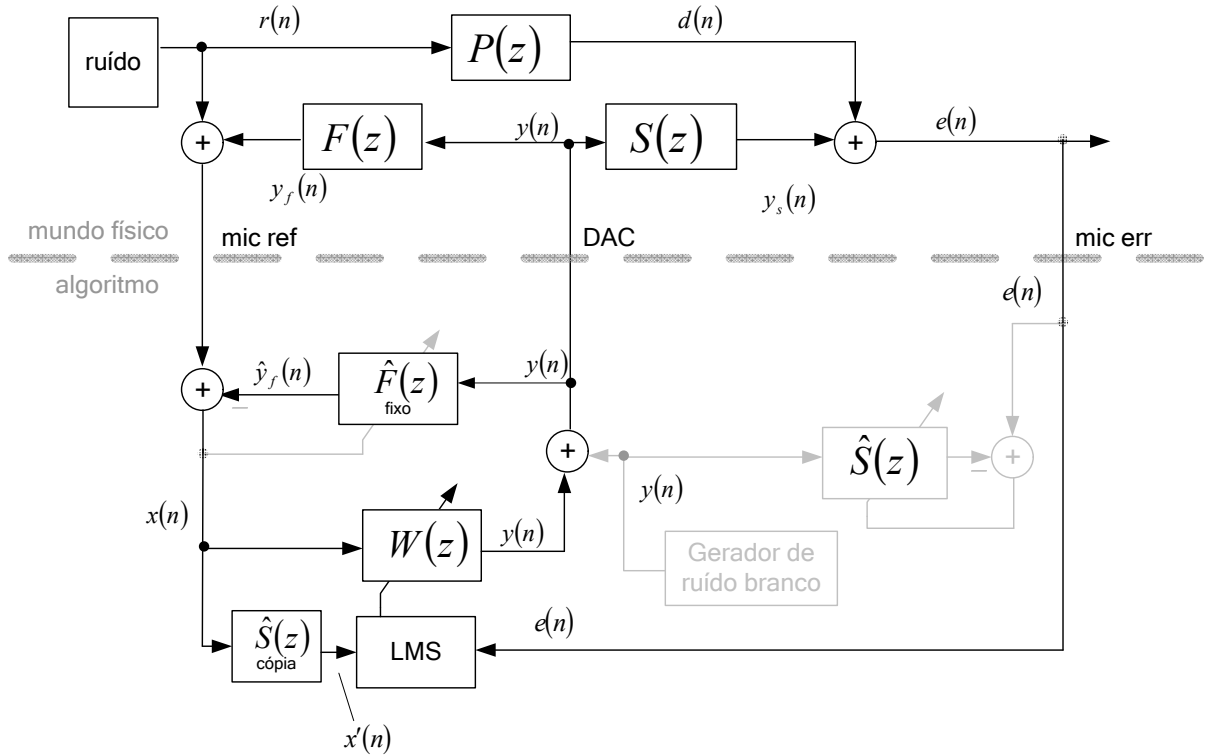


Figura 20 Cancelamento do ruído

7.2 FLUXOGRAMA

O fluxograma geral do algoritmo programado pode ser visualizado na figura 21, sob forma de diagrama de estados. Há 3 modos de funcionamento: Reset, Adaptação de $\hat{F}(z)$ e $\hat{S}(z)$, e Cancelamento do ruído.

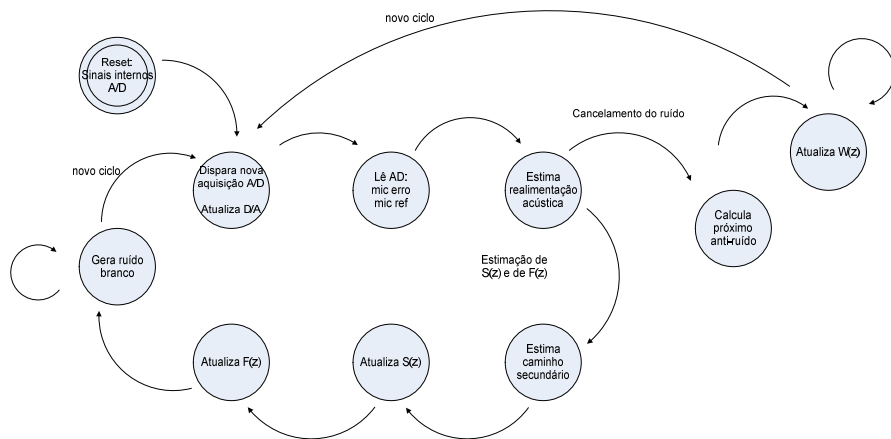


Figura 21 Diagrama de estados

Para controlar a interação entre os diversos eventos, optou-se pela utilização de uma grande máquina de estados e de vários sinais de relógio com frequências diferentes: as operações aritméticas são realizadas a 100 MHz, ao passo que a máquina de estados troca de estado a 1 MHz e repete todas as operações a 2 kHz. Isto permite que o resultado das operações matemáticas tenha tempo para estabilizar-se: o caminho crítico compreende uma operação de multiplicação (latência de 5 ciclos) e 10 operações de soma (latência de 7 ciclos). Portanto, dos 100 ciclos disponíveis apenas no máximo 75 são utilizados por estado.

O sinal de relógio utilizado pelo conversor A/D tem frequência de 2 MHz, garantindo que os comandos enviados a cada transição da máquina de estados (1 MHz) sejam adequadamente recebidos.

Por fim, um sinal de relógio de 2 kHz estabelece o período de amostragem dos sinais analógicos, reiniciando o processo de aquisição dos sinais de áudio, processamento e geração do sinal de anti-ruído.

Uma alternativa à solução adotada (com vários sinais de relógio controlando a operação do circuito) seria a implementação de sinais de handshake e a divisão do circuito em partes menores, que sinalizariam às demais o seu estado. Esta solução poderia utilizar melhor os recursos disponíveis no FPGA, porém requer mais tempo para ser implementada e depurada.

A listagem da programação do FPGA Cyclone III EP3C16F484 em VHDL pode ser encontrada no Apêndice, ao final deste trabalho.

7.3 HARDWARE

O hardware utilizado é composto de dois microfones (para a aquisição do som em dois pontos diferentes), um auto-falante (para a geração do sinal anti-ruído) e um FPGA e circuitos associados (para a implementação dos filtros FIR adaptativos e demais circuitos de controle).

Tudo deve ser montado em um tubo rígido de PVC conforme o diagrama da figura 22.

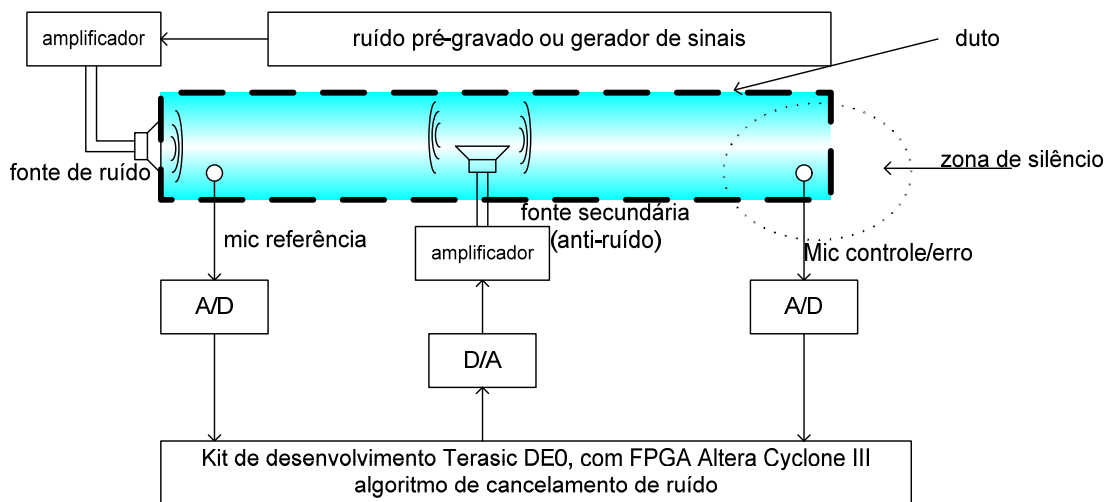


Figura 22 Demonstrador

7.3.1 Pré-amplificação

Os sinais serão adquiridos através de dois microfones de eletreto, e serão pré-amplificados antes do envio à placa de captura. Entre os diversos microfones com resposta em frequência incluindo a faixa de 50 Hz a 200 Hz, os microfones de eletreto foram selecionados devido ao baixo custo de aquisição. Para maior flexibilidade do sistema, o pré-amplificador pode ter seu ganho ajustável desde 11 V/V até 111 V/V. O amplificador operacional selecionado é o modelo OPA2134 (destinado a ser utilizado para áudio) por introduzir baixo ruído no sinal amplificado (característica importante, pois o ganho de tensão do sinal pode ser

elevado). Na figura 23 pode-se ver o esquemático de cada pré-amplificador, e na figura 24 a sua placa de circuito impresso.

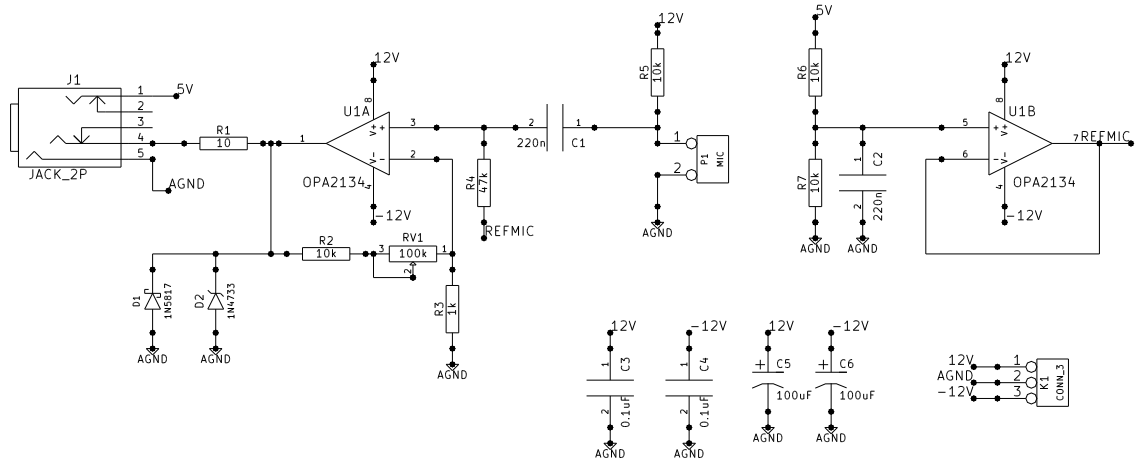


Figura 23 Esquemático dos pré-amplificadores

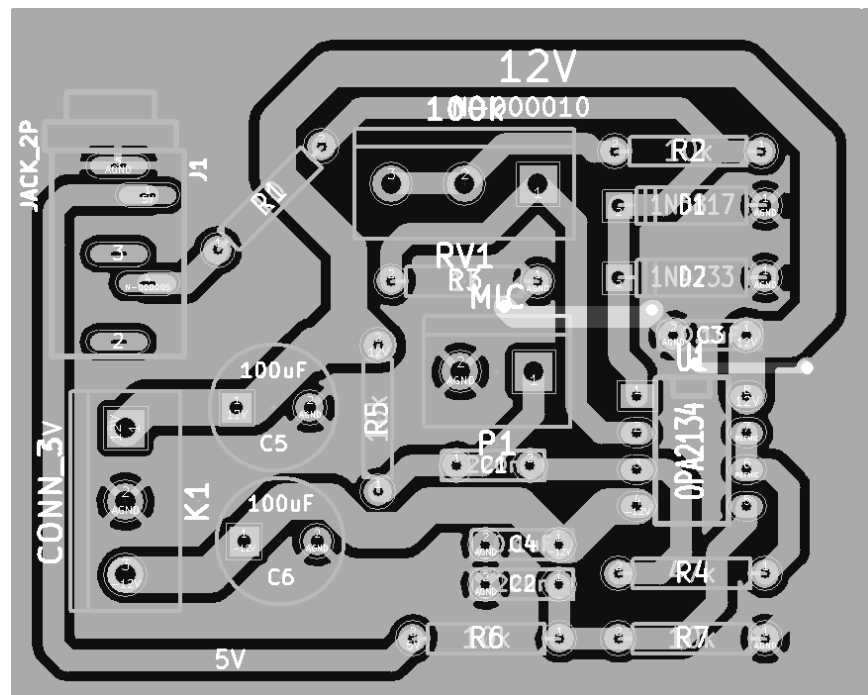


Figura 24 Placa de circuito impresso dos pré-amplificadores

7.3.2 Conversão A/D e D/A

Após ser pré-amplificado, o sinal de áudio passa por um filtro passa-baixas de segunda ordem, ajustado para a frequência de corte de 280 Hz. A sua função de transferência

Butterworth é maximamente plana na banda de passagem e foi escolhida por apresentar um bom compromisso entre a atenuação na banda de rejeição e o atraso de grupo introduzido pelo filtro.

Estes filtros anti-aliasing foram projetados de acordo com a técnica dos resistores e capacitores iguais (LANCASTER, 1975) para filtros Sallen-Key, que consiste em forçar o mesmo valor para os dois resistores que definem a frequência, e o mesmo valor para os dois capacitores. O ganho não é definido pelo projetista, pois passa a ser função do coeficiente de amortecimento (que é por sua vez função do tipo de resposta desejada – neste caso Butterworth). Na figura 25 pode-se ver a seção do esquemático da placa de interfaces A/D e D/A correspondente aos filtros anti-aliasing.

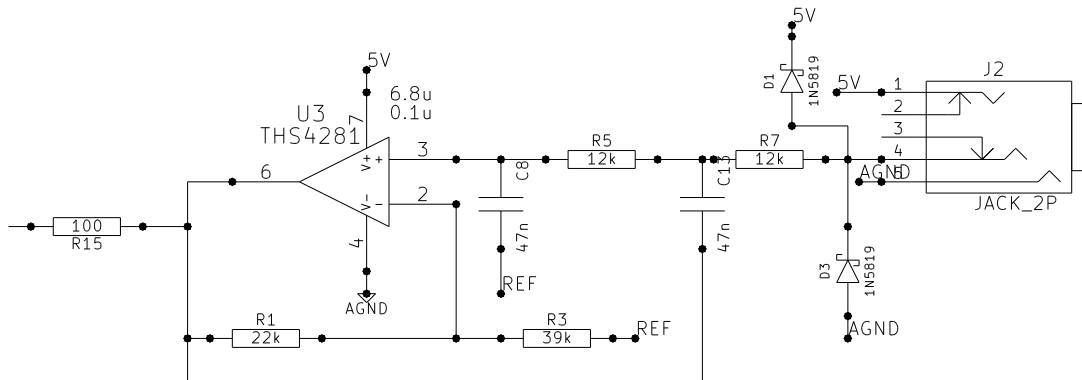


Figura 25 Esquemático da seção de filtragem anti-aliasing

Para realizar a aquisição dos sinais foi selecionado o conversor A/D ADS8365, pois tem um custo razoável e pode amostrar diversos canais simultaneamente. Este conversor funciona por aproximações sucessivas (SAR) e tem um tempo de conversão rápido para a aplicação de áudio, de apenas 8 μ s quando operado com um sinal de relógio de 2 MHz. Além disso, pode ser diretamente conectado aos pinos de I/O do FPGA empregado, já que utiliza níveis lógicos LVCMOS, de 3.3 V, para a sua interface digital de comunicação.

O conversor D/A utilizado é o modelo THS5641A, que possui interface simples e velocidade suficiente para a aplicação. Sua interface consiste em apenas de um sinal de relógio e do dado a converter em formato paralelo.

A saída do conversor D/A é em corrente e pode ser diferencial ou single-ended. O circuito projetado utiliza o modo single-ended para simplificar o projeto do amplificador de potência.

Abaixo, na figura 26, pode-se ver o esquemático da placa de aquisição com o conversor A/D ao centro, o conversor D/A no topo, o conector com a placa de desenvolvimento à esquerda e os filtros anti-aliasing à direita. Na figura 27 está a placa de circuito impresso para interfaces A/D e D/A.

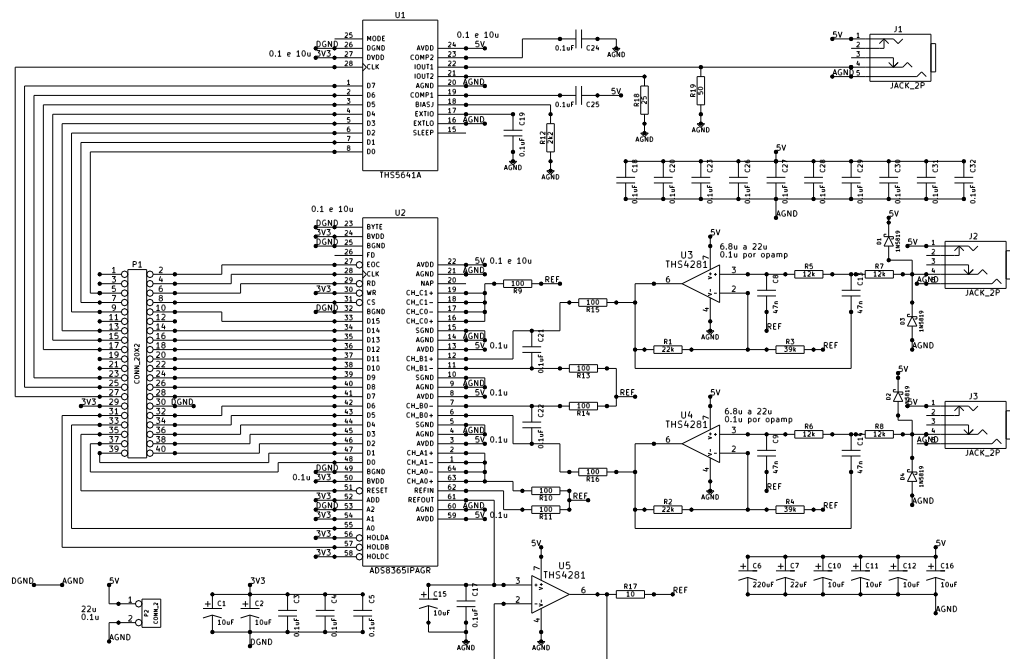


Figura 26 Esquemático da placa de interfaces A/D e D/A

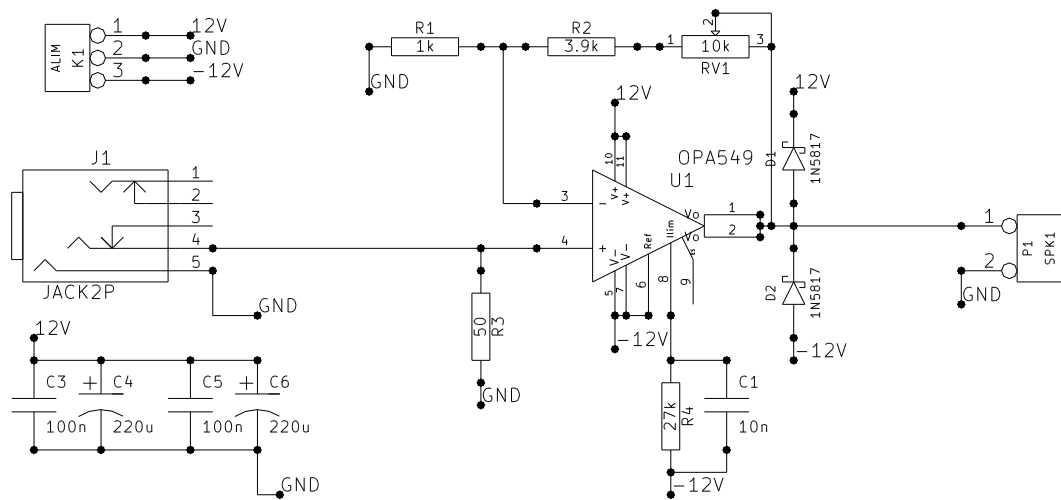


Figura 28 Esquemático dos amplificadores de potência

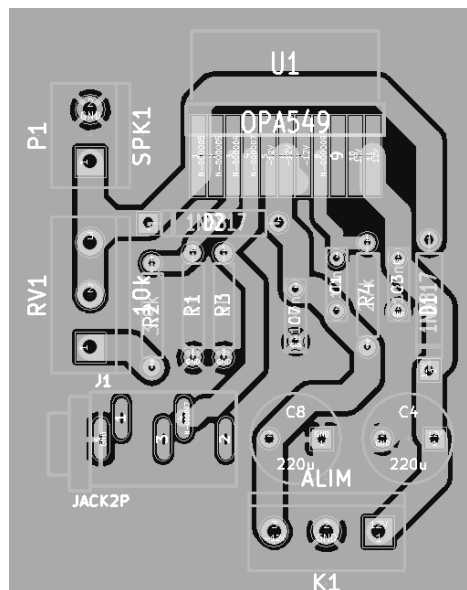


Figura 29 Placa de circuito impresso dos amplificadores de potência

7.3.4 Geração do sinal anti-ruído

Após a aquisição dos sinais de ruído (captados pelo microfone de erro e pelo microfone de referência), os sinais são inicialmente convertidos em notação de ponto flutuante para processamento dentro do FPGA.

A maior parte do processamento a ser realizado é seqüencial como, por exemplo, as operações de cálculo da próxima saída de um filtro FIR e a atualização de seus coeficientes. A interface com o conversor A/D também precisa de um processamento seqüencial: os dois canais são lidos um por vez (a amostragem, no entanto, é realizada pelo conversor A/D quando o sinal HOLD vai a zero, e ocorre simultaneamente para ambos os canais).

Como todo os sinais estão representados em ponto flutuante, é necessário instanciar as unidades de soma e multiplicação de ponto flutuante para serem sintetizadas pela ferramenta de concepção. Um filtro FIR de ordem N (contendo $N-1$ atrasos) precisará de N multiplicações e $N-1$ somas para computar a saída, porém a atualização de seus coeficientes precisará de N somadores e de N multiplicadores. Deste modo, o processamento dos dados em ponto flutuante fica a cargo de unidades com 3 entradas e uma saída: as 2 primeiras entradas são multiplicadas e somadas à terceira para calcular o resultado na saída.

Toda programação do FPGA, em VHDL, foi realizada no ambiente de desenvolvimento Quartus II 9.1sp2 do fabricante Altera. Este sistema gera, para o FPGA Cyclone III utilizado, somadores e multiplicadores de ponto flutuante que consomem aproximadamente 1000 elementos lógicos por conjunto (1 multiplicador e 1 somador). Como o FPGA Cyclone III EP3C16 possui 15408 elementos lógicos, optou-se por instanciar 10 somadores e 10 multiplicadores de ponto flutuante, em precisão simples (32 bits: 1 bit de sinal, 8 bits de expoente e 24 bits de mantissa dos quais apenas 23 são armazenados).

Uma máquina de estados controla todas as funções que o filtro adaptativo pode realizar. Inicialmente, aguarda-se um sinal de reset do usuário, o que permite reiniciar o conversor A/D e, em seguida, iniciar a adaptação dos filtros $\hat{F}(z)$ e $\hat{S}(z)$. A adaptação continua até que o operador sinalize a passagem para o modo de cancelamento do ruído,

quando então os filtros $\hat{F}(z)$ e $\hat{S}(z)$ param de ser adaptados e o filtro $W(z)$ começa a ser adaptado.

Durante a fase de adaptação um gerador de números pseudo-aleatórios permanece ativo, ao passo que na fase de cancelamento do ruído o filtro adaptativo $W(z)$ gera o sinal de anti-ruído. Tanto o ruído branco quanto o anti-ruído são convertidos para o formato de números inteiros antes de serem enviados ao conversor D/A.

8 RESULTADOS ALCANÇADOS

8.1 RESULTADOS SIMULINK

Antes da implementação física do demonstrador, o algoritmo LMS foi simulado com o software MATLAB/Simulink para que se obtivesse uma estimativa do tamanho necessário para os filtros adaptativos. Concluiu-se que para cada componente freqüencial do ruído (caso este seja periódico) bastam 2 coeficientes no filtro adaptativo. Caso o filtro deva também atenuar um ruído branco, então quanto mais componentes tiver o filtro adaptativo, melhor.

Um dos aspectos a considerar quando se projeta o sistema são os atrasos introduzidos pelas diversas funções de transferência, e pelo processamento dos sinais. Se o caminho primário tiver um atraso maior do que o caminho secundário, então será possível atenuar ruídos não-periódicos: há tempo para o sistema anular a onda sonora de ruído. Se o caminho secundário for mais lento que o caminho primário, apenas ruídos repetitivos (componentes senoidais) poderão ser atenuadas.

Na figura 30 pode-se ver uma simulação do algoritmo FxLMS cujos resultados após 20 s, também apresentados na figura 31, foram a atenuação do ruído gaussiano na entrada do sistema em aproximadamente 10 dB, e a atenuação da senóide em mais de 40 dB. Na figura 31 à esquerda, a linha superior representa o ruído caso não houvesse controle ativo, e a linha inferior representa o ruído residual após 20 s de início do sistema.

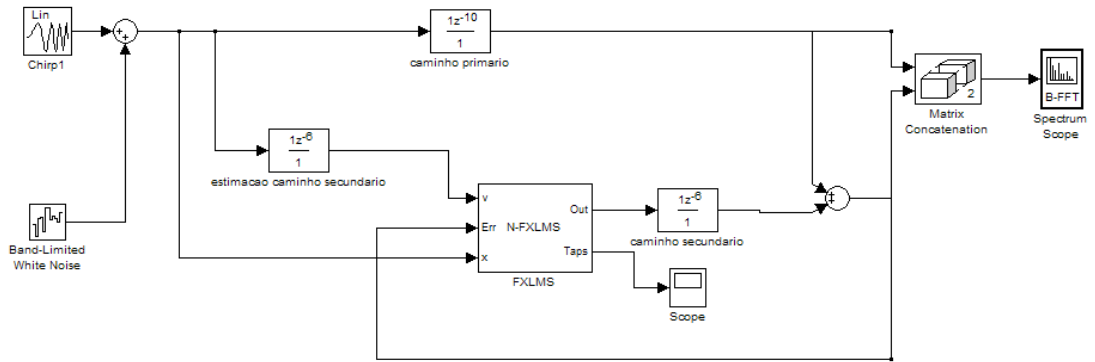


Figura 30 Simulação do algoritmo FxLMS

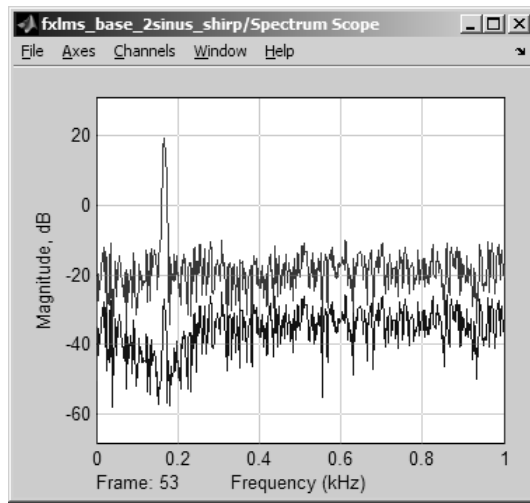


Figura 31 Resultados após 20s de simulação do algoritmo FxLMS

Na figura 32 pode-se observar a evolução dos coeficientes do filtro adaptativo.

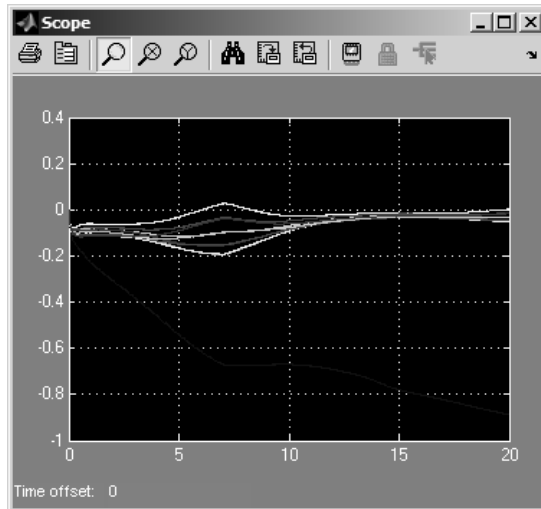


Figura 32 Resultados após 20s de simulação do algoritmo FxLMS

8.2 RESULTADO DA SÍNTESE DE CÓDIGO VHDL

O código VHDL listado no Apêndice foi sintetizado em um circuito FPGA Cyclone III EP3C16F484C6N, presente na placa de desenvolvimento DE0. A tabela 1 lista a utilização dos elementos lógicos necessária para a implementação do circuito, bem como os parâmetros de consumo previstos.

Tabela 1 Parâmetros do circuito sintetizado

| Recurso | Utilizados | Disponíveis | Utilização (%) |
|-----------------------------------|------------|-------------|----------------|
| Elementos lógicos totais | 15153 | 15408 | 98% |
| Funções combinacionais | 12731 | 15408 | 83% |
| Registradores dedicados | 9842 | 15408 | 64% |
| Bits de memória | 644 | 516096 | <1% |
| Multiplicadores dedicados (9-bit) | 63 | 112 | 56% |
| PLLs | 1 | 4 | 25% |
| | | | |
| Frequência máxima | 116,75 MHz | | |
| Consumo total | 97,41 mW | | |

9 CONCLUSÃO

Os métodos de filtragem adaptativa podem ser empregados para se realizar dispositivos de controle ativo do ruído que possam seguir as variações nos parâmetros estatísticos da entrada e ajustar-se em conseqüência. Para se cancelar um ruído periódico, é suficiente que o controlador gere um sinal de anti-ruído com precisão (em fase e em amplitude). Para se cancelar um ruído não-periódico, além de realizar a oposição de fase e amplitude com precisão, o controlador deverá ser capaz de realizar todo o processamento do sinal captado em um tempo menor do que aquele que o ruído leva para chegar à zona de silêncio, caso contrário não haverá convergência.

O controle ativo do ruído adéqua-se bem às situações onde o ruído a ser eliminado é de baixa freqüência, complementando as técnicas tradicionais de redução passiva do ruído (que são mais eficazes em altas freqüências). A aplicação de controle ativo do ruído pode contribuir para reduzir prejuízos econômicos e sociais que são atualmente causados por uma exposição demasiada ao ruído.

Apesar de trazer muitas possibilidades novas de controle do ruído, o controle ativo do ruído não é uma solução que resolve todos os problemas da área de acústica, e sua implementação prática ainda pode ser trabalhosa.

Como sugestão para melhoramentos futuros, pode-se investigar novas formas de adquirir o sinal de referência e de gerar o sinal de anti-ruído. Uma possibilidade seria emitir o sinal de anti-ruído através de um arranjo de auto-falantes capaz de direcionar o som emitido, contribuindo para a redução do ruído em uma região do espaço sem aumentá-lo na direção oposta, e, eliminando a realimentação acústica, facilitando a aquisição do sinal de referência.

REFERÊNCIAS

ABRANTES, S. A. **Processamento adaptativo de sinais**. Lisboa: Fundação Caluste Gulbenkian, 2000. 301 p. ISBN: 972-31-0896-8.

AKHTAR, M. T.; ABE, M.; KAWAMATA, M. On Active Noise Control Systems With Online Acoustic Feedback Path Modeling. **IEEE Transactions on Audio, Speech, and Language Processing**, v. 15, n. 2, p. 593-600, fev. 2007.

BOSE. **QuietComfort 15: Owner's Guide**. 2009. 10 p. Disponível em: <http://products.bose.com/pdf/customer_service/owners/og_qc15.pdf>. Acesso em: 19 maio 2011.

CHANG, C. Y.; LI, S. T. Active Noise Control in Headsets by Using a Low-Cost Microcontroller. **IEEE Transactions on Industrial Electronics**, v. 58, n. 5, p. 1936-1942, maio 2011.

GALLAND, M. A. **Filtrage adaptatif – Application au contrôle actif**. U. E. Génie Electrique. École Centrale de Lyon, Écully, 2009.

GIRARD, S. et al. **Le Bruit en Milieu de Travail: Une Analyse des Coûts pour le Régime d'Indemnisation**. Direction Systèmes de Soins et Services. 43 p., out 2007. ISBN: 978-2-550-51295-0.

GMC. **High-End Audio Technology Helps Terrain Get Top MPGs**. 2011. Disponível em: <http://media.gm.com/content/media/us/en/gmc/news.detail.html/content/Pages/news/us/en/2011/Feb/0222_Terrain>. Acesso em: 20 maio 2011.

KESTELL, C. D. **Active control of sound in a small engine aircraft cabin with virtual error sensors**. 2000. 227 p. Tese (Doutorado em engenharia) – Department of Mechanical Engineering, Adelaide University, South Australia, 2000.

LANCASTER, D. **Active-Filter Cookbook**. Indianapolis: Howard W. Sams & Co., 1975. 240 p. ISBN: 0-672-21168-8.

MINOGUE, P.; RANKIN, N.; RYAN, J. Adaptivey Cancelling Server Fan Noise: Principles and Experiments with a Short Duct and the AD73522 dspConverter. **Analog Dialogue** v. 34, n. 2, p. 1-6, 2000.

MÖSER, M. **Engineering Acoustics: An Introduction to Noise Control**. 2 ed. Berlin: Springer-Verlag, 2009. 536 p. ISBN: 978-3-540-92722-8.

NIOSH. **Work Related Hearing Loss**. DHHS (NIOSH) Publication, n. 103, 2001. Disponível em: <<http://www.cdc.gov/niosh/docs/2001-103/>>. Acesso em: 2 jul 2011.

PAUL LUEG. **Process of silencing sound oscillations**. US n. PI 2,043,416, 8 mar 1934 (depósito), 9 jun 1936.

THORNE, P. et al. Epidemiology of noise-induced hearing loss in New Zealand. **Journal of the New Zealand Medical Association**, v. 121, n. 128022, ago 2008. Disponível em: <<http://www.nzma.org.nz/journal/121-1280/3211>>. Acesso em: 19 jun 2011.

ULTRA Electronics. **Active Noise and Vibration Control**. <<http://www.ultra-controls.com/productdetails.php?productID=15>>. Acesso em: 20 maio 2011.

UOSUKAINEN, S. Unidirectional JMC actuators and their approximations in the active attenuation of noise in ducts. **VTT Publications** v. 341, 100p. Espoo, 1998.

WIDROW, B.; STEARNS, S. D. **Adaptive signal processing**. Upper Saddle River, Prentice-Hall PTR, 1985. 474 p. ISBN: 0-13-004029-0.

APÊNDICE:

Listagem em VHDL para a programação do FPGA Cyclone III

```

1      LIBRARY ieee;
2      USE ieee.std_logic_1164.all;
3
4      ENTITY anc IS
5          GENERIC(n : INTEGER := 32);
6          PORT (      CLOCK_50   : IN STD_LOGIC;
7                  KEY           : IN STD_LOGIC_VECTOR(2 DOWNTO 0);
8                  SW            : IN STD_LOGIC_VECTOR(9 DOWNTO 0);
9                  AD_EOC        : IN STD_LOGIC;
10                 AD_CLOCK       : OUT STD_LOGIC;
11                 AD_RD          : OUT STD_LOGIC;
12                 AD_CS          : OUT STD_LOGIC;
13                 AD_DATA        : IN STD_LOGIC_VECTOR(15 DOWNTO 0);
14                 AD_RESET       : OUT STD_LOGIC;
15                 AD_CANAL       : OUT STD_LOGIC;
16                 AD_HOLD        : OUT STD_LOGIC;
17                 DA_CLOCK       : OUT STD_LOGIC;
18                 DA_DATA        : OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
19      END anc;
20
21      ARCHITECTURE ancarch OF anc IS
22
23          TYPE sinal_fpx10 IS ARRAY(9 DOWNTO 0) OF STD_LOGIC_VECTOR(n-
24          1 DOWNTO 0);
25
26          TYPE tipo_estado IS (st1, st2, st3, st4, st6, st7, st8, st9, st10, st11, st12, st13,
27          st14, st15, st16, st17, st18, st19, st20, st21, st22, st23, st24, st25, st26, st27, st28,
28          st30, st31, st32, st33, st34, st35);
29
30          SIGNAL estado           : tipo_estado;
31          SIGNAL pr_estado        : tipo_estado;
32          SIGNAL push_rst         : STD_LOGIC;
33          SIGNAL chave_run        : STD_LOGIC;
34          SIGNAL espera           : STD_LOGIC;
35          SIGNAL clk_100          : STD_LOGIC;
36          SIGNAL clk_1            : STD_LOGIC;
37          SIGNAL clk_2            : STD_LOGIC;

```



```

34     SIGNAL clk_2K           : STD_LOGIC;
35     SIGNAL ad0_in          : STD_LOGIC_VECTOR(15 DOWNTO 0);
36     SIGNAL ad1_in          : STD_LOGIC_VECTOR(15 DOWNTO 0);
37     SIGNAL da_buffer       : STD_LOGIC_VECTOR(7 DOWNTO 0);
38     SIGNAL mic_erro        : STD_LOGIC_VECTOR(n-1 DOWNTO 0);
39     SIGNAL mic_ref         : STD_LOGIC_VECTOR(n-1 DOWNTO 0);
40     SIGNAL y_fb            : STD_LOGIC_VECTOR(n-1 DOWNTO 0);
41     SIGNAL y_sec           : STD_LOGIC_VECTOR(n-1 DOWNTO 0);
42     SIGNAL eus2           : STD_LOGIC_VECTOR(n-1 DOWNTO 0);
43     SIGNAL xuf2           : STD_LOGIC_VECTOR(n-1 DOWNTO 0);
44     SIGNAL euw2           : STD_LOGIC_VECTOR(n-1 DOWNTO 0);
45     SIGNAL ynn            : sinal_fpx10;
46     SIGNAL pr_ynn         : sinal_fpx10;
47     SIGNAL yn             : sinal_fpx10;
48     SIGNAL xn             : sinal_fpx10;
49     SIGNAL xln           : sinal_fpx10;
50     SIGNAL s_coef        : sinal_fpx10;
51     SIGNAL f_coef        : sinal_fpx10;
52     SIGNAL w_coef        : sinal_fpx10;
53     SIGNAL pr_yn         : sinal_fpx10;
54     SIGNAL pr_xn         : sinal_fpx10;
55     SIGNAL pr_xln        : sinal_fpx10;
56     SIGNAL pr_s_coef     : sinal_fpx10;
57     SIGNAL pr_f_coef     : sinal_fpx10;
58     SIGNAL pr_w_coef     : sinal_fpx10;
59     SIGNAL intfp_in      : STD_LOGIC_VECTOR(15 DOWNTO 0);
60     SIGNAL intfp_out     : STD_LOGIC_VECTOR(n-1 DOWNTO 0);
61     SIGNAL fpint_in      : STD_LOGIC_VECTOR(n-1 DOWNTO 0);
62     SIGNAL fpint_out     : STD_LOGIC_VECTOR(7 DOWNTO 0);
63     SIGNAL multa_in      : sinal_fpx10;
64     SIGNAL multb_in      : sinal_fpx10;
65     SIGNAL somar_in      : sinal_fpx10;
66     SIGNAL mms_out       : sinal_fpx10;
67     SIGNAL randn         : STD_LOGIC_VECTOR(7 DOWNTO 0);
68     SIGNAL randm         : STD_LOGIC_VECTOR(7 DOWNTO 0);

```

```

69
70     CONSTANT zeron      : STD_LOGIC_VECTOR(n-1 DOWNTO 0)
       := "00000000000000000000000000000000"; -- 0.0
71     CONSTANT um        : STD_LOGIC_VECTOR(n-1 DOWNTO 0)
       := "00111111110000000000000000000000"; -- 1.0
72     CONSTANT menosum   : STD_LOGIC_VECTOR(n-1 DOWNTO 0)
       := "10111111110000000000000000000000"; -- -1.0
73     CONSTANT us2       : STD_LOGIC_VECTOR(n-1 DOWNTO 0)
       := x"3A83126F";-- 0.001
74     CONSTANT uf2       : STD_LOGIC_VECTOR(n-1 DOWNTO 0)
       := x"3A83126F";-- 0.001
75     CONSTANT uw2       : STD_LOGIC_VECTOR(n-1 DOWNTO 0)
       := x"3A83126F";-- 0.001
76     CONSTANT lp_coef :          sinal_fpx10          :=
       ((x"3AF05FDC"),(x"3CAC08F9"),(x"3D9ED333"),(x"3E28E85D"),(x"3E704C2
       B"),(x"3E704C2B"),(x"3E28E85D"),(x"3D9ED333"),(x"3CAC08F9"),(x"3AF05
       FDC")); -- Fc = 0.3*Fs, Chebyshev
77
78     COMPONENT fp_int8_conv
79         PORT (dataa : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
80              clock  : IN STD_LOGIC ;
81              result : OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
82     END COMPONENT;
83
84     COMPONENT int16_fp_conv
85         PORT (dataa : IN STD_LOGIC_VECTOR(15 DOWNTO 0);
86              clock  : IN STD_LOGIC ;
87              result : OUT STD_LOGIC_VECTOR(31 DOWNTO 0));
88     END COMPONENT;
89
90     COMPONENT fp_mms
91         PORT (clock : IN STD_LOGIC ;
92              multa  : IN STD_LOGIC_VECTOR(n-1 DOWNTO 0);
93              multb  : IN STD_LOGIC_VECTOR(n-1 DOWNTO 0);
94              somar  : IN STD_LOGIC_VECTOR(n-1 DOWNTO 0);
95              resultado : OUT STD_LOGIC_VECTOR(n-1 DOWNTO 0));
96     END COMPONENT;
97

```

```

98     COMPONENT pll_ctrl
99         PORT (inclk0 : IN STD_LOGIC := '0';
100             c0      : OUT STD_LOGIC;
101             c1      : OUT STD_LOGIC;
102             c2      : OUT STD_LOGIC;
103             c3      : OUT STD_LOGIC);
104     END COMPONENT;
105
106     BEGIN
107
108     fp_int : fp_int8_conv PORT MAP (
109         dataa => fpint_in,
110         clock => clk_100,
111         result => fpint_out);
112
113     int_fp : int16_fp_conv PORT MAP (
114         dataa => intfp_in,
115         clock => clk_100,
116         result => intfp_out);
117
118     generate_filtro : FOR i IN 1 TO 9 GENERATE
119     BEGIN
120     filtro: fp_mms PORT MAP (
121         clock => clk_100,
122         multa => multa_in(i),
123         multb => multb_in(i),
124         somar => somar_in(i),
125         resultado => mms_out(i));
126     END GENERATE;
127
128     pll_ctrl_inst : pll_ctrl PORT MAP (
129         inclk0 => CLOCK_50,      -- 50 MHz
130         c0     => clk_100,      -- 100 MHz
131         c1     => clk_2,        -- 2 MHz
132         c2     => clk_1,        -- 1 MHz

```

```
133         c3      => clk_2k);  -- 2 kHz
134
135
136     push_rst <= key(0);
137     chave_run <= sw(0);
138     AD_CLOCK <= clk_2;
139     DA_CLOCK <= clk_1;
140
141
142     PROCESS(clk_1)
143     BEGIN
144         IF (clk_1'EVENT AND clk_1='1') THEN
145             IF (push_rst='1') THEN
146                 estado <= st1;
147             ELSE
148                 estado <= pr_estado;
149             END IF;
150         END IF;
151     END PROCESS;
152
153
154     PROCESS(clk_2k)
155     BEGIN
156         IF (clk_2k'EVENT AND clk_2k='1') THEN
157             IF ((estado = st28) OR (estado = st25)) THEN
158                 espera <= '0';
159             ELSE
160                 espera <= '1';
161             END IF;
162         END IF;
163     END PROCESS;
164
165
166     PROCESS(clk_100)
167     BEGIN
```

```

168     IF (clk_100'EVENT AND clk_100='1') THEN
169         CASE estado IS
170             WHEN st1=>                                     --reset
171                 pr_estado    <= st2;
172                 AD_RD        <= '1';
173                 AD_CS        <= '1';
174                 AD_RESET    <= '0';
175                 AD_CANAL    <= '0';
176                 AD_HOLD    <= '1';
177                 DA_DATA    <= (OTHERS => '0');
178                 mic_erro    <= (OTHERS => '0');
179                 mic_ref     <= (OTHERS => '0');
180                 y_fb        <= (OTHERS => '0');
181                 y_sec       <= (OTHERS => '0');
182                 yn          <= ((OTHERS=> (OTHERS=>'0')));
183                 xn          <= ((OTHERS=> (OTHERS=>'0')));
184                 xln         <= ((OTHERS=> (OTHERS=>'0')));
185                 s_coef      <= ((OTHERS=> (OTHERS=>'0')));
186                 f_coef      <= ((OTHERS=> (OTHERS=>'0')));
187                 w_coef      <= ((OTHERS=> (OTHERS=>'0')));
188                 intfp_in    <= (OTHERS => '0');
189                 fpint_in    <= (OTHERS => '0');
190                 multa_in    <= ((OTHERS=> (OTHERS=>'0')));
191                 multb_in    <= ((OTHERS=> (OTHERS=>'0')));
192                 somar_in    <= ((OTHERS=> (OTHERS=>'0')));
193                 randm <= "10110101";
194         -- lê A/D, escreve D/A
195             WHEN st2=>
196                 pr_estado <= st3;
197                 AD_RESET <= '1';
198                 AD_CS <= '1';
199                 AD_HOLD <= '1';
200                 DA_DATA <= da_buffer;
201
202             WHEN st3=>

```

```
203             pr_estado <= st4;
204
205     WHEN st4=>
206             pr_estado <= st6;
207             AD_CS <= '0';
208             AD_HOLD <= '0';
209
210     WHEN st6=>
211             pr_estado <= st7;
212     WHEN st7=>
213             pr_estado <= st8;
214     WHEN st8=>
215             pr_estado <= st9;
216     WHEN st9=>
217             pr_estado <= st10;
218     WHEN st10=>
219             pr_estado <= st11;
220     WHEN st11=>
221             pr_estado <= st12;
222     WHEN st12=>
223             pr_estado <= st13;
224
225     WHEN st13=>
226             pr_estado <= st14;
227             AD_CS <= '0';
228             AD_RD <= '0';
229
230     WHEN st14=>
231             pr_estado <= st15;
232             ad0_in <= AD_DATA;
233
234     WHEN st15=>
235             pr_estado <= st16;
236             AD_CS <= '1';
237             AD_RD <= '1';
```

```

238             AD_CANAL <= '1';
239             intfp_in <= ad0_in;
240
241         WHEN st16=>
242             pr_estado <= st17;
243             AD_CS <= '0';
244             AD_RD <= '0';
245
246         WHEN st17=>
247             pr_estado <= st18;
248             ad1_in <= AD_DATA;
249
250         WHEN st18=>
251             pr_estado <= st19;
252             AD_CS <= '1';
253             AD_RD <= '1';
254             AD_CANAL <= '0';
255             mic_erro <= intfp_out;
256             intfp_in <= ad1_in;
257         -- começa identificação dos caminhos secundário e de realimentação acústica, ou
258         -- vai para a filtragem adaptativa do ruído
259         WHEN st19=>
260             IF (chave_run = '1') THEN
261                 pr_estado <= st20;
262             ELSE
263                 pr_estado <= st30;
264             END IF;
265
266             mic_ref <= intfp_out;
267             multa_in <= yn;
268             multb_in <= f_coef;
269             somar_in(0) <= zeron;
270             somar_in(1) <= mms_out(0);
271             somar_in(2) <= mms_out(1);
272             somar_in(3) <= mms_out(2);

```

```
272          somar_in(4) <= mms_out(3);
273          somar_in(5) <= mms_out(4);
274          somar_in(6) <= mms_out(5);
275          somar_in(7) <= mms_out(6);
276          somar_in(8) <= mms_out(7);
277          somar_in(9) <= mms_out(8);
278          y_fb <= mms_out(9);
279
280      WHEN st20=>
281          pr_estado <= st21;
282          multa_in <= yn;
283          multb_in <= s_coef;
284          somar_in(0) <= zeron;
285          somar_in(1) <= mms_out(0);
286          somar_in(2) <= mms_out(1);
287          somar_in(3) <= mms_out(2);
288          somar_in(4) <= mms_out(3);
289          somar_in(5) <= mms_out(4);
290          somar_in(6) <= mms_out(5);
291          somar_in(7) <= mms_out(6);
292          somar_in(8) <= mms_out(7);
293          somar_in(9) <= mms_out(8);
294          y_sec <= mms_out(9);
295
296      WHEN st21=>
297          pr_estado <= st22;
298          multa_in(0) <= y_sec;
299          multb_in(0) <= menosum;
300          somar_in(0) <= mic_erro;
301          multa_in(1) <= mms_out(0);
302          multb_in(1) <= us2;
303          somar_in(1) <= zeron;
304          eus2 <= mms_out(1);
305
306          multa_in(2) <= y_fb;
```



```

307         multb_in(2) <= menosum;
308         somar_in(2) <= mic_ref;
309         multa_in(3) <= mms_out(2); -- x
310         multb_in(3) <= uf2;
311         somar_in(3) <= zeron;
312         xuf2 <= mms_out(3);
313         randn <= randm;
314     -- atualiza o filtro adaptativo que representa o caminho secundário
315     WHEN st22=>
316         pr_estado <= st23;
317         multa_in <= yn;
318         multb_in(0) <= eus2;
319         multb_in(1) <= eus2;
320         multb_in(2) <= eus2;
321         multb_in(3) <= eus2;
322         multb_in(4) <= eus2;
323         multb_in(5) <= eus2;
324         multb_in(6) <= eus2;
325         multb_in(7) <= eus2;
326         multb_in(8) <= eus2;
327         multb_in(9) <= eus2;
328         somar_in <= s_coef;
329         pr_s_coef <= mms_out;
330
331         randm(0) <= '0';
332         randm(1) <= randn(0);
333         randm(2) <= randn(1);
334         randm(3) <= randn(2);
335         randm(4) <= randn(3);
336         randm(5) <= randn(4);
337         randm(6) <= randn(5);
338         randm(7) <= randn(6);
339     -- atualiza o filtro adaptativo que representa o caminho de realimentação acústica
340     WHEN st23=>
341         pr_estado <= st24;

```

```

342         s_coef <= pr_s_coef;
343         multa_in <= yn;
344         multb_in(0) <= xuf2;
345         multb_in(1) <= xuf2;
346         multb_in(2) <= xuf2;
347         multb_in(3) <= xuf2;
348         multb_in(4) <= xuf2;
349         multb_in(5) <= xuf2;
350         multb_in(6) <= xuf2;
351         multb_in(7) <= xuf2;
352         multb_in(8) <= xuf2;
353         multb_in(9) <= xuf2;
354         somar_in <= f_coef;
355         pr_f_coef <= mms_out;
356     -- gera um número pseudo-aleatório
357         IF (randm(7) = '1') THEN
358             randn <= "10110100" XOR randm;
359         ELSE
360             randn <= randm;
361         END IF;
362
363     WHEN st24=>
364         pr_estado <= st25;
365         f_coef <= pr_f_coef;
366
367         randm <= randn;
368         intfp_in <= (randm & "00000000");
369
370     WHEN st25=>
371         pr_estado <= st26;
372         pr_ynn(0) <= intfp_out;
373         pr_ynn(1) <= ynn(0);
374         pr_ynn(2) <= ynn(1);
375         pr_ynn(3) <= ynn(2);
376         pr_ynn(4) <= ynn(3);

```

```

377         pr_ynn(5) <= ynn(4);
378         pr_ynn(6) <= ynn(5);
379         pr_ynn(7) <= ynn(6);
380         pr_ynn(8) <= ynn(7);
381         pr_ynn(9) <= ynn(8);
382         -- apenas filtra o sinal de ruído gerado internamente com um filtro passa-baixas
          (fixo) antes de colocá-lo na saída
383         WHEN st26=>
384             pr_estado <= st27;
385             ynn <= pr_ynn;
386             multa_in <= ynn;
387             multb_in <= lp_coef;
388             somar_in(0) <= zeron;
389             somar_in(1) <= mms_out(0);
390             somar_in(2) <= mms_out(1);
391             somar_in(3) <= mms_out(2);
392             somar_in(4) <= mms_out(3);
393             somar_in(5) <= mms_out(4);
394             somar_in(6) <= mms_out(5);
395             somar_in(7) <= mms_out(6);
396             somar_in(8) <= mms_out(7);
397             somar_in(9) <= mms_out(8);
398             pr_yn(0) <= mms_out(9);
399             pr_yn(1) <= yn(0);
400             pr_yn(2) <= yn(1);
401             pr_yn(3) <= yn(2);
402             pr_yn(4) <= yn(3);
403             pr_yn(5) <= yn(4);
404             pr_yn(6) <= yn(5);
405             pr_yn(7) <= yn(6);
406             pr_yn(8) <= yn(7);
407             pr_yn(9) <= yn(8);
408
409         WHEN st27=>
410             pr_estado <= st28;

```

```

411             yn <= pr_yn;
412             fpint_in <= yn(0);
413     -- espera por um novo ciclo de clk_2k, o clock que controla a amostragem
414     WHEN st28=>
415         IF (espera='0') THEN
416             pr_estado <= st2;
417         ELSE
418             pr_estado <= st28;
419         END IF;
420
421     da_buffer <= fpint_out;
422
423     -- modo de filtragem adaptativa:
424     -- estado anterior: st19
425
426     WHEN st30=>
427         pr_estado <= st31;
428         multa_in(2) <= y_fb;
429         multb_in(2) <= menosum;
430         somar_in(2) <= mic_ref;
431         multa_in(3) <= mms_out(2); --x
432         multb_in(3) <= uf2;
433         somar_in(3) <= zeron;
434         xuf2 <= mms_out(3);
435
436         pr_xn(0) <= mms_out(2); -- x
437         pr_xn(1) <= xn(0);
438         pr_xn(2) <= xn(1);
439         pr_xn(3) <= xn(2);
440         pr_xn(4) <= xn(3);
441         pr_xn(5) <= xn(4);
442         pr_xn(6) <= xn(5);
443         pr_xn(7) <= xn(6);
444         pr_xn(8) <= xn(7);
445         pr_xn(9) <= xn(8);

```

```
446
447     WHEN st31=>
448         pr_estado <= st32;
449         xn <= pr_xn;
450         multa_in <= xn;
451         multb_in <= w_coef;
452         somar_in(0) <= zeron;
453         somar_in(1) <= mms_out(0);
454         somar_in(2) <= mms_out(1);
455         somar_in(3) <= mms_out(2);
456         somar_in(4) <= mms_out(3);
457         somar_in(5) <= mms_out(4);
458         somar_in(6) <= mms_out(5);
459         somar_in(7) <= mms_out(6);
460         somar_in(8) <= mms_out(7);
461         somar_in(9) <= mms_out(8);
462         pr_yn(0) <= mms_out(9);
463         pr_yn(1) <= yn(0);
464         pr_yn(2) <= yn(1);
465         pr_yn(3) <= yn(2);
466         pr_yn(4) <= yn(3);
467         pr_yn(5) <= yn(4);
468         pr_yn(6) <= yn(5);
469         pr_yn(7) <= yn(6);
470         pr_yn(8) <= yn(7);
471         pr_yn(9) <= yn(8);
472
473     WHEN st32=>
474         pr_estado <= st33;
475         yn <= pr_yn;
476         fpint_in <= yn(0);
477         multa_in <= xn;
478         multb_in <= s_coef;
479         somar_in(0) <= zeron;
480         somar_in(1) <= mms_out(0);
```

```

481          somar_in(2) <= mms_out(1);
482          somar_in(3) <= mms_out(2);
483          somar_in(4) <= mms_out(3);
484          somar_in(5) <= mms_out(4);
485          somar_in(6) <= mms_out(5);
486          somar_in(7) <= mms_out(6);
487          somar_in(8) <= mms_out(7);
488          somar_in(9) <= mms_out(8);
489          pr_xln(0) <= mms_out(9);
490          pr_xln(1) <= xln(0);
491          pr_xln(2) <= xln(1);
492          pr_xln(3) <= xln(2);
493          pr_xln(4) <= xln(3);
494          pr_xln(5) <= xln(4);
495          pr_xln(6) <= xln(5);
496          pr_xln(7) <= xln(6);
497          pr_xln(8) <= xln(7);
498          pr_xln(9) <= xln(8);
499
500          WHEN st33=>
501              pr_estado <= st34;
502              xln <= pr_xln;
503              multa_in(0) <= mic_erro;
504              multb_in(0) <= uw2;
505              somar_in(0) <= zeron;
506              euw2 <= mms_out(0);
507
508              da_buffer <= fpint_out;
509          -- calcula os novos coeficientes do filtro W(z)
510          WHEN st34=>
511              pr_estado <= st35;
512              multa_in <= xln;
513              multb_in(0) <= euw2;
514              multb_in(1) <= euw2;
515              multb_in(2) <= euw2;

```

```
516         multb_in(3) <= euw2;
517         multb_in(4) <= euw2;
518         multb_in(5) <= euw2;
519         multb_in(6) <= euw2;
520         multb_in(7) <= euw2;
521         multb_in(8) <= euw2;
522         multb_in(9) <= euw2;
523         somar_in <= w_coef;
524         pr_w_coef <= mms_out;
525     -- atualiza os coeficientes do filtro adaptativo
526     WHEN st35=>
527         pr_estado <= st35;
528         w_coef <= pr_w_coef;
529
530     IF (espera='0') THEN
531         pr_estado <= st2;
532     ELSE
533         pr_estado <= st28;
534     END IF;
535
536     END CASE;
537 END IF;
538 END PROCESS;
539 END ancarch;
```