

UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FERNANDO HAGE CANTERUCCIO

**Estudo de Desempenho *versus* Custo computacional de Algoritmos
Adaptativos Aplicados a Redução Ativa de Ruído Acústico**

VIÇOSA
2016

FERNANDO HAGE CANTERUCCIO

**ESTUDO DE DESEMPENHO *VERSUS* CUSTO COMPUTACIONAL DE
ALGORITMOS ADAPTATIVOS APLICADOS A REDUÇÃO ATIVA DE
RUÍDO ACÚSTICO**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Tarcísio de Assunção Pizziolo

VIÇOSA
2016

FERNANDO HAGE CANTERUCCIO

ESTUDO DE DESEMPENHO *VERSUS* CUSTO COMPUTACIONAL DE ALGORITMOS ADAPTATIVOS APLICADOS A REDUÇÃO ATIVA DE RÚIDO ACÚSTICO

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 15 de Dezembro de 2016.

COMISSÃO EXAMINADORA

Prof. Dr. Tarcísio de Assunção Pizziolo
Universidade Federal de Viçosa

Prof. Dr. André Gomes Tôrres - Membro
Universidade Federal de Viçosa

Prof. Dr. Mauro de Oliveira Prates - Membro
Universidade Federal de Viçosa

“Quod enim est erit semper”

H. T.’

Aos meus pais, Suzel e Alexandre, e aos meus irmãos incríveis

Agradecimentos

Agradeço sinceramente a minha mãe, Suzel, e ao meu pai, Alexandre, por me proporcionarem a oportunidade de tamanho crescimento e inspiração para trilhar a senda, pois nenhum outro caminho me traria ao mesmo ponto onde me encontro hoje, e ao meu irmão e irmã por serem fonte de orgulho constante.

Agradeço a todo o apoio recebido dos amigos quando caí e não tive força suficiente para, sozinho, me levantar.

Agradeço a todo homem que, querendo ou não, me é professor, e a cada dia me abre os olhos e me dá a capacidade de enxergar a beleza pura da manifestação.

Por fim, agradeço a Deus por me proporcionar a dádiva de viver o eterno presente.

Resumo

Uma teoria matemática para redução ativa de ruído acústico no ar já é bem desenvolvida desde os anos oitenta, porém apenas recentemente, com o desenvolvimento de processadores digitais mais rápidos e baratos, foi que a implementação deste tipo de redução de ruído se tornou tecnológica e economicamente viável

Neste trabalho de monografia será feito um estudo comparativo sobre o desempenho de diferentes algoritmos adaptativos candidatos a serem aplicados em um dispositivo redutor ativo de ruído acústico, através de simulações em ambiente virtual, assim como sobre suas complexidades computacionais. Este estudo visa viabilizar o desenvolvimento de um dispositivo redutor de ruído próprio para ser aplicado a cabines de aeronaves do tipo ultra-leves avançados.

Abstract

A mathematical theory about active noise cancellation is very well known since the eighties, however only recently, with the development of faster and cheaper digital processors, the implementation of this type of noise reduction became technologically and economically viable.

In this paper, a comparative study will be done on the performance of different adaptive algorithms to be applied in an active noise reduction device, by means of simulations in a virtual environment, as well as on its computational complexities. This study aims to make possible the development of a proper noise reduction device to be applied to advanced ultra-light aircraft cabins.

Sumário

1	Introdução.....	15
1.1	Objetivos.....	16
1.2	Organização do Trabalho.....	16
2	Conceitos Básicos.....	17
2.1	Introdução.....	17
2.1.1	Atenuação Passiva.....	17
2.1.2	Atenuação Ativa.....	18
2.2	Filtros Digitais.....	19
2.2.1	Introdução.....	19
2.2.2	Filtro IIR.....	19
2.2.3	Filtro FIR.....	20
2.2.4	Projeto de um Filtro Digital.....	20
2.3	Filtragem Adaptativa.....	21
2.3.1	O Filtro de Wiener.....	21
2.3.2	Solução de Wiener para Filtros Transversais.....	22
2.3.3	O Princípio da Ortogonalidade.....	23
2.3.4	Determinação do Erro Quadrático Médio Mínimo.....	24
2.3.5	O Algoritmo Steepest Descent.....	25
2.3.6	O Algoritmo Least-Mean-Square.....	26
2.3.7	O Algoritmo NLMS.....	27
2.3.8	O Efeito do Caminho Secundário.....	28
2.3.9	O Algoritmo Leaky FxLMS.....	29
2.3.10	Algoritmo Híbrido VSSLMS/FxLMS.....	31
2.3.11	Algoritmo LMS de Passo Variável para Ambientes com Razão Sinal-Ruído Baixa (FxNLMSVS).....	33
2.4	Complexidade de Algoritmo.....	34
2.4.1	Introdução.....	34
2.4.2	Análise Assintótica.....	35
3	Simulação em Ambiente Virtual.....	36
3.1	Modelo Acústico.....	37
3.2	Algoritmos Utilizados.....	40
3.2.1	Treinamento Off-line do Modelo do Caminho Secundário.....	40
3.2.2	Algoritmos de Controle.....	41
3.2.2.1	Algoritmo FxNLMS de Passo Adaptativo e Ganho de Realimentação Variável (FxNLMSVSAG).....	42

3.2.2.2 Erro Quadrático Médio	42
3.2.2.3 Custo computacional dos Algoritmos Utilizados	43
4 Resultados e Discussão	45
5 Conclusão	61
Referências Bibliográficas	62

Lista de Figuras

Figura 1: Interferência destrutiva e ruído residual resultado da combinação imperfeita de magnitude e/ou fase [2].	18
Figura 2: Processo de filtragem digital de sinal.	19
Figura 3: Filtro transversal de Wiener.	21
Figura 4: Modelagem do sistema com efeito do caminho secundário $H(z)$ [2].	29
Figura 5: Topologia do algoritmo FxLMS [2].	30
Figura 6: Algoritmo de decisão para adaptação do step-size de acordo com parâmetro thd. ..	33
Figura 7: Duto Acústico	37
Figura 8: Diagrama de bode da função transferência do caminho primário.	39
Figura 9: Diagrama de bode da função transferência do caminho secundário.	39
Figura 10: Modelagem off-line do caminho secundário, $C(Z)$ [2].	41
Figura 11: Ganho de desempenho computacional entre os algoritmos VSSLMS/LMS e LMS convencional [13].	45
Figura 12: Resultados da simulação dos algoritmos FxLMS e FxNLMS. De cima para baixo tem-se o ruído primário contaminado com ruído branco aditivo, o antiruído gerado pelo filtro e o ruído residual após a superposição das ondas sonoras.	46
Figura 13: Resultados da simulação dos algoritmos FxLMS e FxNLMS. De cima para baixo tem-se o passo de adaptação, a soma das correlações cruzadas quadráticas do sinal de erro com seus lags, o ganho no feedback do algoritmo e a amplitude do ruído residual em dB.	47
Figura 14: Espectrograma do sinal de ruído residual na zona silenciosa para simulação com os algoritmos FxLMS e FxNLMS.	48
Figura 15: MSE dos Algoritmo FxLMS e FxNLMS.	49
Figura 16: Resultados da simulação do algoritmo FxNLMSVS. De cima para baixo tem-se o ruído primário contaminado com ruído branco aditivo, o antiruído gerado pelo filtro e o ruído residual após a superposição das ondas sonoras.	50
Figura 17: Resultados da simulação do algoritmo FxNLMSVS. De cima para baixo tem-se o passo de adaptação, a soma das correlações cruzadas quadráticas do sinal de erro com seus lags, o ganho no feedback do algoritmo e a amplitude do ruído residual em dB.	51
Figura 18: Espectrograma do sinal de ruído residual na zona silenciosa para simulação com o algoritmo FxNLMSVS.	52
Figura 19: MSE do algoritmo FxNLMSVS.	53
Figura 20: Resultados da simulação do algoritmo FxNLMSVSAG. De cima para baixo tem-se o ruído primário contaminado com ruído branco aditivo, o antiruído gerado pelo filtro e o ruído residual após a superposição das ondas sonoras.	54
Figura 21: Resultados da simulação do algoritmo FxNLMSVSAG. De cima para baixo tem-se o passo de adaptação, a soma das correlações cruzadas quadráticas do sinal de erro com seus lags, o ganho no feedback do algoritmo e a amplitude do ruído residual em dB.	55

Figura 22: Espectrograma do sinal de ruído residual na zona silenciosa para simulação com o algoritmo FxNLMSVSAS.	56
Figura 23: Resultados da simulação do algoritmo FxNLMSVSAG. De cima para baixo tem-se o ruído primário, o antiruído gerado pelo filtro e o ruído residual após a superposição das ondas sonoras.	57
Figura 24: Resultados da simulação do algoritmo FxNLMSVSAG. De cima para baixo tem-se o passo de adaptação, a soma das correlações cruzadas quadráticas do sinal de erro com seus lags, o ganho no feedback do algoritmo e a amplitude do ruído residual em dB.	57
Figura 25: Espectrograma do sinal de ruído residual na zona silenciosa para simulação com o algoritmo FxNLMSVSAS sem a presença do ruído branco contaminante.	58
Figura 26: MSE do algoritmo FxNLMSVSAG.....	59

Lista de Tabelas

Tabela 1: Custos computacionais comparados dos algoritmos estudados.	43
Tabela 2: Resultados comparados das simulações versus custo computacional dos algoritmos testados.	60

1 Introdução

Com o aumento na utilização de máquinas cíclicas ruidosas, crescente densidade populacional e demanda por menores níveis de ruídos acústicos contaminantes no ambiente, cresce a procura por soluções de baixo custo e eficientes para redução de ruído acústico.

A redução ativa de ruído acústico vem solucionar o problema da contaminação do ambiente por ruídos de baixa frequência de forma mais eficiente e de baixo custo se comparada a formas de redução passiva de ruído, como pela utilização de materiais resistivos e arranjos de dutos reativos.

Uma destas áreas de aplicação da redução ativa de ruído acústico é na aviação leve, onde o piloto e passageiros ficam expostos a elevados níveis de ruído acústico por muitas horas, fazendo-se necessária a utilização de proteção auricular apropriada para redução do ruído. Estes equipamentos são projetados com conchas acústicas isoladas, geralmente por gel, e causam grande grau de desconforto ao usuário, principalmente quando utilizados por tempos longos em viagens de longa distância. Alternativamente a este tipo de protetores auriculares, existem atualmente auscultadores com redutores ativos de ruído. Estes tratam de obter um sinal de referência do ruído contaminante, utilizam algoritmos que fazem o casamento de amplitudes e a inversão de fase do sinal capturado e injetam o sinal tratado no transdutor interno, afim de realizar o cancelamento acústico do ruído por interferência destrutiva, porém este tipo de equipamento possui alto custo e ainda causam certo grau de desconforto.

Com este problema como motivação, a utilização da redução ativa de ruído acústico apresenta elevada viabilidade de aplicação, fazendo uso de pequenos processadores embarcados, microfones e transdutores de saída leves, como auto-falantes pequenos, para gerar um antiruído proporcional e de fase invertida ao ruído contaminante, objetivando sua anulação por sobreposição das ondas sonoras no ar [1].

1.1 Objetivos

Este trabalho tem como objetivo fazer uma análise de desempenho de diferentes algoritmos adaptativos aplicados a um dispositivo redutor ativo de ruído acústico destinado a cabines de aviões ultra-leves, como parte do desenvolvimento e projeto do produto.

Dado o objetivo geral, têm-se como objetivos específicos:

1. estudar a topologia apropriada do sistema a ser aplicado.
2. realizar a montagem em ambiente virtual e simulação de tal topologia.
3. coletar dados sobre o desempenho em estado estacionário e transiente dos filtros.
4. estudar o custo computacional e viabilidade de implementação de cada combinação de algoritmos.
5. realizar a comparação desempenho versus complexidade entre as diferentes combinações de algoritmos afim de se escolher a mais apropriada à aplicação neste caso específico.

1.2 Organização do Trabalho

Este trabalho é dividido em cinco ênfases dadas a seguir.

1. Introdução.
2. Conceitos básicos, fazendo uma introdução a matemática de filtragem adaptativa.
3. Simulação, onde são apresentados os dados e modelos utilizados na coleta na simulação em ambiente virtual.
4. Resultados, onde são coletados e discutidos os dados resultantes da simulação.
5. Conclusão

2 *Conceitos Básicos*

Nesta seção serão abordados tópicos importantes para a compreensão e dará base para a simulação e análise dos resultados.

2.1 *Introdução*

Problemas de ruído acústico tem se tornado mais notáveis pelas razões dadas a seguir.

- Aumento no número de grandes máquinas cíclicas em ambientes industriais, como motores, ventiladores, exaustores, compressores, bombas elétricas, etc.
- Aumento na densidade de população de grandes centros urbanos trazendo o crescimento da exposição média a ruído sofrida por seus moradores, proveniente dos vizinhos ou tráfego de veículos próximos às moradias.
- O uso de materiais mais leves e de baixo custo na construção civil, causando uma diminuição na capacidade de atenuação de ruídos das paredes e limites das casas e apartamentos.

Dois tipos de ruídos acústicos podem existir no ambiente. Um deles é causado por turbulências no ar, e este é totalmente aleatório. Fontes comuns para este tipo de ruído são explosões, turbinas, ruídos causados pelo vento, etc. Este tipo de ruído distribui sua energia igualmente por todo o espectro de frequências. Este é chamado ruído de banda larga, ou *broadband*. O segundo tipo de contaminador é o ruído de banda estreita, ou *narrowband*, e este concentra sua energia em frequências específicas dentro do espectro, sendo estas a frequência fundamental e harmônicas. Ruídos deste tipo são gerados por máquinas cíclicas, como motores a explosão, máquinas elétricas, bombas, compressores, etc [2].

2.1.1 *Atenuação Passiva*

Pode-se classificar um redutor de ruído acústico como passivo ou ativo. O redutor passivo de ruído utiliza de materiais absorventes, como concreto, espuma ou cortiça, que são colocados entre a fonte de ruído e o interlocutor afim de se atenuar a amplitude do ruído que chega ao ouvinte. Dentro dos redutores passivos ainda pode-se classificar um sistema entre

resistivo ou reativo. Sistemas resistivos são aqueles que se baseiam na perda de energia sofrida por uma onda propagando-se em um duto coberto por material absorvente, como em chaminés e dutos de escape e ventilação. Sistemas reativos são combinações de dutos e barreiras, comumente utilizados em escapes de motores a combustão interna como silenciadores de automóveis, e exploram o conceito de mudança de impedância acústica para dado arranjo de dutos. Sistemas de redução de ruído passivos são utilizados quando o peso e tamanho não são um problema, e tem características de atenuação sobre uma ampla faixa de frequências, porém são normalmente de alto custo e ainda causam uma pressão não desejada enquanto o fluido escoar pelos dutos [2].

2.1.2 Atenuação Ativa

O redutor ativo de ruído acústico tem como princípio de funcionamento a interação entre diferentes ondas de pressão no ar (ondas sonoras) de forma a obter-se interferência destrutiva e assim mitigar a potência do ruído não desejado. O dispositivo redutor de ruído toma uma entrada de ruído de referência, esta pode ser acústica ou não, e a partir desta gera um antirruído afim de causar interferência destrutiva. Para que isto ocorra, na zona silenciosa desejada, ambos ruído e antirruído devem ter a mesma magnitude e devem ter uma defasagem de 180° entre um e outro. Qualquer desvio destes requerimentos gera como resultado um ruído residual, sendo este a soma do ruído original com o antirruído. A Figura 1 exibe o processo físico de sobreposição das ondas sonoras no meio [2].

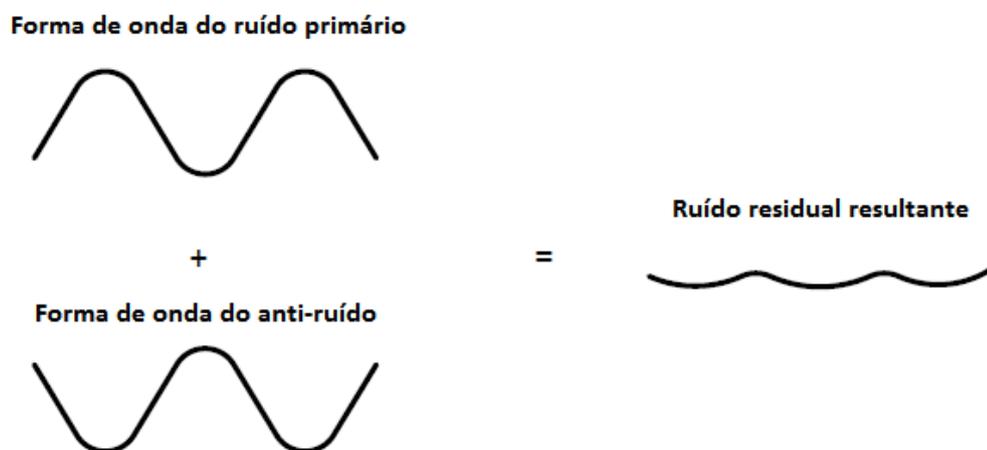


Figura 1: Interferência destrutiva e ruído residual resultante da combinação imperfeita de magnitude e/ou fase [2].

O dispositivo utilizado para fazer o controle de magnitude e defasagem do sinal antiruído pode ser classificado como *narrowband* ou *broadband* dependendo do tipo de ruído que este visa atenuar, e ainda em *feedforward* ou *feedback* dependendo do tipo de malha de controle empregado.

2.2 Filtros Digitais

2.2.1 Introdução

Filtros digitais são empregados para fazer o tratamento e filtragem de sinais amostrados, afim de se controlar suas características de magnitude e fase. O processo de filtragem de sinais digitais é exibido na Figura 2.

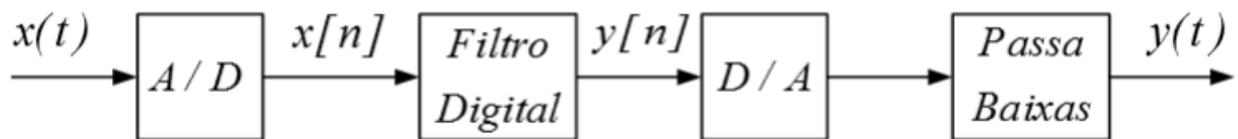


Figura 2: Processo de filtragem digital de sinal.

O bloco A/D representa um conversor analógico/digital o qual é usado para fazer a amostragem do sinal $x(t)$. O sinal amostrado $x(n)$ é então processado pelo filtro digital e obtém-se a saída $y(n)$ como sinal filtrado. O bloco D/A representa um conversor digital/analógico que transforma $y(n)$ em um sinal contínuo e este é reconstruído com um filtro passa-baixas, obtendo-se a saída $y(t)$ [3].

2.2.2 Filtro IIR

São Filtros Digitais cuja resposta ao impulso apresenta duração infinita (*IIR – Infinite Impulse Response*). As saídas do filtro IIR podem ser calculadas pela equação (1) [4].

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \quad (1)$$

O filtro IIR tem suas características de entrada e saída definidas por equações lineares de diferenças, com coeficientes de natureza recursiva, sendo estas [4]:

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) = \sum_{k=0}^N b_k x(n-k) - \sum_{k=1}^M a_k y(n-k) \quad (2)$$

Aplicando-se a transformada z , obtêm-se a sua função transferência:

$$H(z) = \frac{\sum_{k=0}^N b_k z^{-k}}{1 + \sum_{k=1}^M a_k z^{-k}} \quad (3)$$

Tem-se que a função transferência do filtro IIR é uma fração de polinômios de potências negativas de z , sendo a_k e b_k os coeficientes do filtro.

2.2.3 Filtro FIR

São filtros digitais cuja resposta ao impulso apresenta duração finita (*FIR – Finite Impulse Response*). As saídas do filtro FIR podem ser calculadas pela equação (2), tomando-se $a_k = 0$ para todo valor de k .

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) = \sum_{k=0}^N b_k x(n-k) \quad (4)$$

Filtros FIR tem como função transferência a equação (5) [4].

$$H(z) = \sum_{k=0}^{N-1} H(k)z^{-k} \quad (5)$$

Nota-se que o filtro FIR tem sua função transferência como uma função polinomial de potências negativas de z . Os filtros FIR ainda apresentam características desejáveis em processamento digital de sinais como memória finita, o que força transitórios a terem duração limitada, e ainda apresentam resposta em fase linear [4].

2.2.4 Projeto de um Filtro Digital

Para determinar o projeto adequado de um filtro digital a ser utilizado em dada aplicação, seguem-se os seguintes passos [5]:

- especificação das características do filtro;
- cálculo dos coeficientes da função transferência;
- teste e análise dos efeitos do comprimento finito da palavra digital no desempenho do filtro;
- implementação do filtro em software.

A escolha do tipo de filtro depende da aplicação. Deve-se considerar a resposta em fase do filtro FIR, implicando em ausência de distorção de fase no sinal filtrado. Essa característica é importante em áreas como tratamento digital de áudio e imagem, transmissão de dados, entre outras. Além disso, filtros FIR são não recursivos, e desta forma sempre estáveis e tem os efeitos da precisão finita e erros de quantização mitigados quando comparados aos filtros IIR. Pelo lado positivo, filtros IIR necessitam de menor ordem para atender os mesmos requisitos de filtros FIR e podem ser facilmente convertidos de filtros analógicos equivalentes [3].

2.3 Filtragem Adaptativa

2.3.1 O Filtro de Wiener

Como conceito introdutório para filtragem adaptativa, é conveniente entender inicialmente o filtro de Wiener. Este filtro é utilizado para produzir uma estimativa de um processo aleatório através da filtragem linear invariante no tempo (LTI) de um processo ruidoso, dado o espectro de sinal e o ruído conhecidos. Este filtro visa minimizar o erro quadrático entre as saídas do processo aleatório em questão e as saídas do filtro estimado.

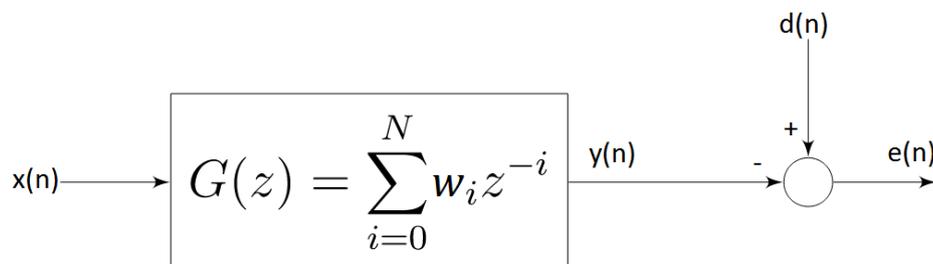


Figura 3: Filtro transversal de Wiener.

Consideremos que o bloco contendo $G(z)$ na Figura 3 é um filtro linear discreto com resposta ao impulso $[a_0, a_1, a_2, \dots, a_{N-1}]$ e cuja entrada é dada por $x(n)$. Dado que a saída do filtro seja $y(n)$, obtém-se o sinal de erro $e(n)$ a partir da subtração do sinal desejado $d(n)$ por $y(n)$ como mostrado na equação (6).

$$e(n) = d(n) - y(n) \quad (6)$$

Dado o diagrama de blocos da Figura 3, nota-se que para se minimizar o sinal de erro $e(n)$, deve-se escolher os coeficientes do filtro afim de igualar-se $y(n)$ a $d(n)$. Para satisfazer a condição de obter-se apenas uma resposta ótima para o filtro, toma-se a função a ser minimizada como o erro quadrático médio, dado da forma:

$$J = E[e(n)^2] = E[e(n)e^*(n)] \quad (7)$$

Sendo $E[.]$ o operador de esperança matemática, ou valor esperado, e $*$ representando o complexo conjugado. J é chamada função desempenho, ou função custo, a ser minimizada. Em particular, no caso de filtros FIR, a função J é um parabolóide com concavidade voltada para cima contendo um único ponto de mínimo [6].

2.3.2 Solução de Wiener para Filtros Transversais

Dado o filtro transversal da Figura 3, cuja entrada a cada instante n é caracterizada pelo vetor $\mathbf{u}(n) = [u(n), u(n-1), u(n-2), \dots, u(n-N+1)]^T$, assume-se que tanto a entrada $u(n)$ como sua resposta desejada $d(n)$ são processos aleatórios estacionários e, em geral, complexos, assim como são complexos os coeficientes do filtro.

A solução de Wiener para os parâmetros ótimos do filtro transversal que minimiza a função custo J é dada tal que se tem um vetor $\mathbf{w} = [w_0, w_1, \dots, w_{N-1}]^T$ que satisfaz a equação (7):

$$y(n) = \sum_{k=0}^{N-1} w_k^* u(n-k) = \mathbf{w}^H \mathbf{u}(n) \quad (8)$$

Sendo \mathbf{w}^H o Hermitiano da matriz \mathbf{w} .

Da equação (7), temos o sinal de erro expresso vetorialmente como:

$$e(n) = d(n) - \mathbf{w}^H \mathbf{u}(n) \quad (9)$$

Substituindo (9) em (7), obtém-se:

$$J = E\{[d(n) - \mathbf{w}^H \mathbf{u}(n)][d^*(n) - \mathbf{u}^H(n)\mathbf{w}]\} \quad (10)$$

$$J = E[|d(n)|^2] - \mathbf{p}^H \mathbf{w} - \mathbf{w}^H \mathbf{p} + \mathbf{w}^H \mathbf{R} \mathbf{w} \quad (11)$$

Sendo \mathbf{R} e \mathbf{p} a matriz de autocorrelação de $\mathbf{u}(n)$ e o vetor correlação cruzada entre $\mathbf{u}(n)$ e $d(n)$, respectivamente.

Define-se então \mathbf{R} e \mathbf{p} , omitindo-se o argumento (n) , como:

$$\mathbf{p} = E[\mathbf{u}(n)d^*(n)] \quad (12)$$

$$\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)] = \begin{bmatrix} r(0) & r(1) & \cdots & r(N-1) \\ r^*(1) & r(0) & \cdots & r(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ r^*(N-1) & r^*(N-2) & \cdots & r(0) \end{bmatrix} \quad (13)$$

A equação (11) descreve uma função quadrática dependente dos parâmetros do filtro transversal, ou seja, J é mínimo para \mathbf{w} sendo o vetor de coeficientes ótimos do filtro.

Para determinar então os coeficientes do filtro \mathbf{w} calcula-se o gradiente de J e então iguala-se este gradiente a zero.

$$\Delta J = \frac{\partial J}{\partial \mathbf{w}} = 2\mathbf{p} - 2\mathbf{R}\mathbf{w} \quad (14)$$

Igualando-se (14) a zero, obtém-se:

$$\mathbf{R}\mathbf{w}_0 = \mathbf{p} \quad (15)$$

$$\mathbf{w}_0 = \mathbf{R}^{-1}\mathbf{p} \quad (16)$$

Sendo \mathbf{w}_0 o vetor de coeficientes ótimos para o filtro de Wiener.

O sistema de equações descrito por (15) e (16) é chamado *equações de Wiener-Hopf*. A solução deste sistema assume que \mathbf{R} é uma matriz não-singular.

2.3.3 O Princípio da Ortogonalidade

O princípio da ortogonalidade resulta do cálculo do gradiente de J , obtido através da derivação parcial de J em relação às partes real e imaginária do filtro. O gradiente de J é exibido na equação (17) [7]:

$$\Delta_k J = \frac{\partial e(n)}{\partial \mathbf{a}_k} e^*(n) + \frac{\partial e^*(n)}{\partial \mathbf{a}_k} e(n) + \frac{\partial e(n)}{\partial \mathbf{b}_k} j e^*(n) + \frac{\partial e^*(n)}{\partial \mathbf{b}_k} j e(n) \quad (17)$$

Sendo \mathbf{a}_k e \mathbf{b}_k as partes reais e imaginária de \mathbf{w}_k , respectivamente. Sendo assim:

$$\mathbf{w}_k = \mathbf{a}_k + j\mathbf{b}_k \quad (18)$$

De (9), obtém-se:

$$\frac{\partial e(n)}{\partial \mathbf{a}_k} = -\mathbf{u}(n-k) \quad (19)$$

$$\frac{\partial e(n)}{\partial \mathbf{b}_k} = j\mathbf{u}(n-k) \quad (20)$$

$$\frac{\partial e^*(n)}{\partial \mathbf{a}_k} = -\mathbf{u}^*(n-k) \quad (21)$$

$$\frac{\partial e^*(n)}{\partial \mathbf{b}_k} = -j\mathbf{u}^*(n-k) \quad (22)$$

Daí, utilizando a equação (17), obtém-se:

$$\Delta_k J = -2E[\mathbf{u}(n-k)\mathbf{e}^*(n)] \quad (23)$$

Para obter a equação que descreve o princípio da ortogonalidade, iguala-se (23) a zero, desta forma:

$$E[\mathbf{u}(n-k)\mathbf{e}^*(n)] = \mathbf{0}, \quad k = 0, 1, 2, \dots \quad (24)$$

A equação (24) demonstra que a condição necessária e suficiente para que a função custo J atinja seu valor mínimo é a não correlação do sinal de erro com o sinal de entrada, ou a ortogonalidade entre estes sinais [7].

Dada (24), pode-se perceber que quando o filtro atinge os valores ótimos, tem-se:

$$E[\mathbf{y}(n-k)\mathbf{e}^*(n)] = \mathbf{0}, \quad k = 0, 1, 2, \dots \quad (25)$$

Logo a saída estimada para o filtro de Wiener é ortogonal ao sinal de erro.

2.3.4 Determinação do Erro Quadrático Médio Mínimo

A partir da equação (11), obtém-se J_{min} da forma:

$$J_{min} = E[|d(n)|^2] - \mathbf{p}^H \mathbf{w}_0 - \mathbf{w}_0^H \mathbf{p} + \mathbf{w}_0^H \mathbf{R} \mathbf{w}_0 \quad (26)$$

Substituindo o \mathbf{w}_0 pela equação (16), obtém-se:

$$J_{min} = E[|d(n)|^2] - \mathbf{p}^H \mathbf{R}^{-1} \mathbf{p} - (\mathbf{R}^{-1} \mathbf{p})^H \mathbf{p} + (\mathbf{R}^{-1} \mathbf{p})^H \mathbf{R} \mathbf{R}^{-1} \mathbf{p} = \sigma_d^2 - \mathbf{p}^H \mathbf{R}^{-1} \mathbf{p} \quad (27)$$

Sendo σ_d^2 a variância da resposta desejada $d(n)$.

2.3.5 O Algoritmo Steepest Descent

Dada a alta capacidade computacional demandada para resolver o sistema de equações de Wiener-Hopf (16), devido a necessidade de inversão da matriz \mathbf{R} , um método alternativo foi sugerido para determinar os coeficientes ótimos do filtro de Wiener iterativamente. O algoritmo *Steepest Descent*, é uma alternativa a resolução explícita das equações de Wiener-Hopf para encontrar os coeficientes ótimos do filtro transversal utilizado na estimação da função desejada, consistindo em uma estratégia de atualização iterativa dos parâmetros do filtro utilizando o gradiente da função desempenho.

O algoritmo Steepest Descent utiliza os seguintes passos.

1. Inicialmente atribui-se pesos aleatórios ou nulos para os valores de $w(0)$. Uma alternativa a este passo, caso se tenha conhecimento de valores aproximados ou estimados anteriormente para os coeficientes do filtro, é iniciar $w(0)$ com estes coeficientes. Isso resultará em uma convergência mais rápida no caso da entrada $u(n)$ ser invariante no tempo.

2. Calcula-se o vetor gradiente da função custo na n -ésima iteração $\Delta J(n)$, para $n = 1, 2, 3 \dots$, utilizando-se a equação (14).

$$\Delta J(n) = -2\mathbf{p} + 2\mathbf{R}w(n) \quad (28)$$

3. Dado que o gradiente da função custo calculado no passo 2 aponta na direção e sentido de maior crescimento da função custo, atualiza-se a função no sentido oposto do gradiente obtido, de forma que:

$$w(n+1) = w(n) + \frac{\mu}{2} (-\Delta J(n)) \quad (29)$$

Sendo μ o passo de adaptação. Esta é uma constante que define a velocidade de convergência do filtro, bem como sua estabilidade e erro em estado estacionário.

4. Repete-se os passos 2 a 4 até que o critério de convergência seja alcançado.

A priori, para que se garanta a estabilidade do filtro de parâmetros $w(n)$, o passo de adaptação μ deve obedecer a desigualdade (30) [8]:

$$1 < \mu < 2/\lambda_{max} \quad (30)$$

Sendo λ_{max} o maior autovalor da matriz auto-correlação R .

2.3.6 O Algoritmo Least-Mean-Square

O algoritmo *least-mean-square* e suas variações são preferidos em várias áreas de estudos nos campos da Engenharia Elétrica, Ciências da Computação, Engenharia Biomédica e outras áreas, quando há necessidade de se aplicar filtragem adaptativa ou fazer modelagem de sistemas dinâmicos, por sua baixa complexidade computacional, robustez a ruídos e mudanças súbitas do sinal de entrada, e ainda por sua facilidade de aprendizagem e aplicação. O algoritmo *least-mean-square* se enquadra muito bem no contexto da filtragem adaptativa, atualizando os coeficientes do filtro transversal variantes no tempo. Quando os sinais de entrada se tornam estacionários, o algoritmo LMS tende para a solução ótima de Wiener [8]. A aplicação de tal algoritmo supera a do algoritmo *Steepest Descent* no quesito de baixo custo computacional pela substituição explícita do cálculo do gradiente da função custo por uma estimativa do vetor gradiente obtida a partir de uma única amostra do vetor de entrada. Essa substituição tem como consequência o erro residual na estimação dos parâmetros, e assim na saída do filtro, porém aplicando-se este filtro recursivamente tem-se a gradual diminuição do erro do vetor gradiente, aproximando os coeficientes do filtro com precisão satisfatória para muitas aplicações.

A equação que define a adaptação dos pesos dos filtros é da forma:

$$w(n+1) = w(n) - \frac{\mu}{2} \hat{\Delta} \mathbf{J}(n) \quad (31)$$

Sendo $\hat{\Delta} \mathbf{J}(n)$ a estimativa do vetor gradiente da função custo, dado por:

$$\hat{\Delta} \mathbf{J}(n) = -2\mathbf{u}(n)\mathbf{u}^H(n)w(n) + 2\mathbf{u}(n)d^*(n) \quad (31)$$

Obtida da equação (14), substituindo-se $\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)]$ por $\hat{\mathbf{R}} = \mathbf{u}(n)\mathbf{u}^H(n)$ e $\mathbf{p} = E[\mathbf{u}(n)d^*(n)]$ por $\hat{\mathbf{p}} = \mathbf{u}(n)d^*(n)$.

Observa-se que:

$$-2\mathbf{u}(n)\mathbf{u}^H(n)w(n) + 2\mathbf{u}(n)d^*(n) = \quad (32)$$

$$2\mathbf{u}(n)(d^*(n) - \mathbf{u}^H(n)\mathbf{w}(n)) = \quad (33)$$

$$\hat{\Delta}\mathbf{J}(n) = 2\mathbf{u}(n)e^*(n) \quad (34)$$

Substituindo (34) em (31), obtém-se:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\mathbf{u}^T(n)e^*(n) \quad (35)$$

As três equações que regem o algoritmo LMS são:

1 - A resposta do filtro:

$$y(n) = \mathbf{w}^H(n)\mathbf{u}(n) \quad (36)$$

2 - O desvio da resposta desejada, ou erro:

$$e(n) = d(n) - y(n) \quad (37)$$

3 - A função de adaptação dos pesos do filtro

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\mathbf{u}(n)e^*(n) \quad (38)$$

O passo de aprendizagem μ dita a velocidade de convergência do filtro, o erro estacionário do filtro e a estabilidade do processo de adaptação [9]. A restrição dos valores de μ para que seja assegurada a convergência do filtro são dadas pela equação (39):

$$0 < \mu < \frac{2}{\sum_{k=0}^n \frac{\mathbf{u}^2(n-k)}{n}} \quad (39)$$

Sendo que o somatório no denominador do limite superior de μ representa a potência do sinal de entrada.

2.3.7 O Algoritmo NLMS

No algoritmo LMS, a adaptação dos coeficientes do filtro depende diretamente da potência do sinal de entrada. Quando os valores do vetor de entrada $\mathbf{u}(n)$ são elevados, o algoritmo apresenta um problema de amplificação do ruído do gradiente [7]. Este problema não existe no algoritmo NLMS pois o passo de adaptação é normalizado pela potência do vetor de entrada.

A equação que define o passo de adaptação normalizado é:

$$\mu = \frac{\tilde{\mu}}{a + \sum_{k=0}^{N-1} u^2(n-k)} \quad (40)$$

Sendo $\tilde{\mu}$ a nova constante do passo de adaptação e a uma constante muito pequena que assegura que o passo de adaptação μ não se torne infinito caso a potência do vetor de entrada vá para zero.

Para assegurar a convergência do algoritmo NLMS a constante $\tilde{\mu}$ deve satisfazer a inequação (41) [7].

$$0 < \tilde{\mu} < 2 \quad (41)$$

Pode-se perceber que no algoritmo NLMS o passo de adaptação é variante no tempo, e um novo valor de μ é obtido para cada amostra do vetor de entrada.

Substituindo o novo μ na equação da adaptação do filtro no algoritmo LMS obtém-se:

$$w(n+1) = w(n) - \frac{\tilde{\mu}}{a + \sum_{k=0}^{N-1} u^2(n-k)} \mathbf{u}(n)e^*(n) \quad (42)$$

O algoritmo NLMS é superior ao algoritmo LMS na velocidade de convergência e na robustez a ruídos de potência no sinal de entrada, porém com um custo computacional um pouco mais elevado.

2.3.8 O Efeito do Caminho Secundário

A robustez da aplicação de algoritmos LMS para modelagem de sistemas dinâmicos depende diretamente da aquisição precisa do sinal de erro e de seu feedback na função de adaptação dos coeficientes do filtro. Caso este processo esteja comprometido, é gerado um erro cumulativo nos coeficientes do filtro de forma a desestabilizar o sistema. Existindo função transferência $H(z)$ que corresponde a resposta ao impulso do sistema que modifica o sinal $y(n)$, pode-se modelar o sistema de acordo com a Figura 4.

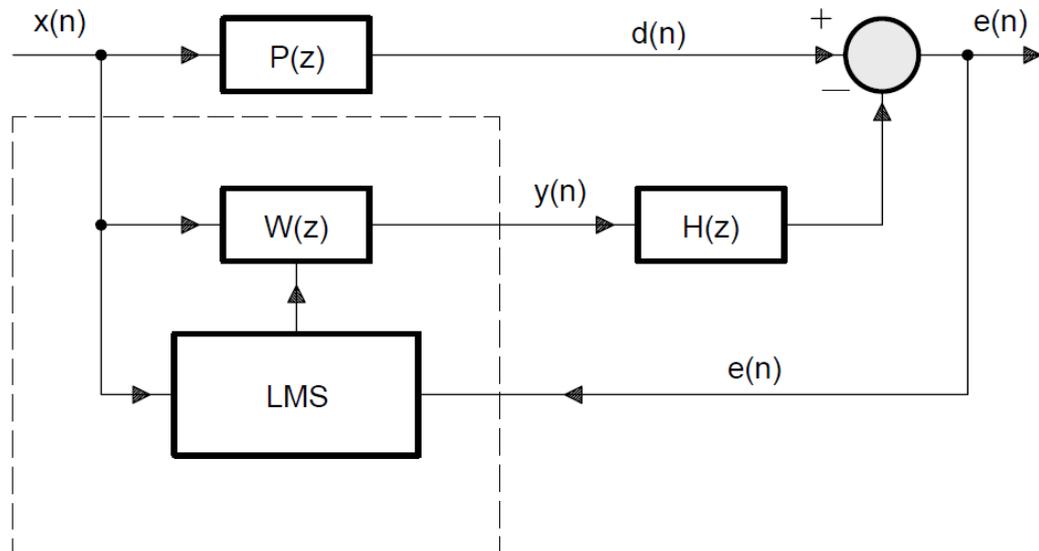


Figura 4: Modelagem do sistema com efeito do caminho secundário $H(z)$ [2].

Daí, pode-se obter o a transformada z do sinal de erro:

$$E(z) = X(z)P(z) + X(z)W(z)H(z) \quad (43)$$

Assumindo $W(z)$ de ordem suficientemente grande, após a convergência do filtro, o sinal de erro deve ser igual a zero, isto é, $E(z) = 0$. Para a função transferência ótima, tem-se então:

$$W(z) = -\frac{P(z)}{H(z)} \quad (44)$$

Desta forma, o filtro adaptativo $W(z)$ deve modelar o caminho primário $P(z)$ e o inverso do caminho secundário $H(z)$. Contudo, é impossível inverter o atraso inerente causado por $H(z)$ caso o atraso do caminho primário não seja pelo menos igual ou superior a este. Além disso, observa-se da equação (43) que o sistema é ineficaz caso exista frequência ω tal que $P(\omega) = 0$. Ainda, caso exista ω tal que $H(\omega) = 0$, o sistema se torna instável [2].

Dados estes fatos, tem-se que é necessária a compensação do efeito do caminho secundário para a aplicação eficiente do algoritmo LMS.

2.3.9 O Algoritmo Leaky FxLMS

Para compensar os efeitos do caminho secundário $H(z)$ na eficácia do filtro LMS, a entrada do dispositivo de aquisição do sinal de erro é filtrada por uma estimativa do caminho

secundário $C(z)$. Isso resulta no algoritmo *Filtered-X LMS*, ou *FXLMS*, proposto por Morgan [10]. A utilização do algoritmo *FxLMS* para compensar os efeitos do caminho secundário foi proposta por Burgess [11]. A Figura 5 apresenta a topologia do algoritmo *FxLMS*.

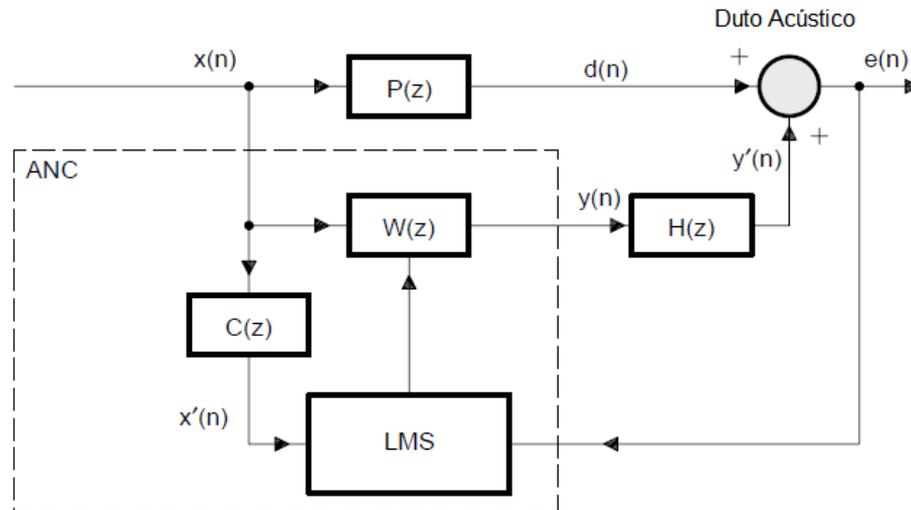


Figura 5: Topologia do algoritmo FxLMS [2].

Além disso, quando um filtro adaptativo é implementado em um processador de sinais com precisão finita, problemas com o arredondamento causam um erro cumulativo nos coeficientes do filtro. Isso faz com que os coeficientes causem *overflow no processador*, o que resulta na degradação do desempenho do filtro. Uma solução para este problema é a multiplicação de cada coeficiente do filtro por uma constante próxima, porém menor que a unidade. Isso gera um pequeno *offset* na função estimada do filtro em relação a resposta ótima, porém atenua o problema de *overflow*.

A função de adaptação dos coeficientes do filtro $w(n)$ é a seguinte:

$$w(n+1) = \alpha w(n) - \mu e(n) \mathbf{c}^T(n) \mathbf{x}(n) \quad (45)$$

Sendo α a constante que multiplica os coeficientes do filtro para controlar o efeito da precisão finita sobre a convergência do filtro, e $\mathbf{c}^T(n)$ o vetor que representa os coeficientes da estimativa do caminho secundário $C(z)$ [10].

Este algoritmo é de rápida convergência e bastante tolerante a erros na estimativa do caminho secundário. Como mostrado por Morgan [10], o algoritmo *FxLMS* com erro de fase entre $C(z)$ e $H(z)$ de quase 90° .

As características da função transferência $H(z)$ são geralmente desconhecidas e variam com o tempo. Várias técnicas de modelagem *on-line* do caminho secundário foram apresentadas por Eriksson [12]. Assumindo-se que as características de $H(z)$ são invariantes no intervalo de tempo de interesse, é possível realizar a modelagem *off-line* do caminho secundário em uma fase prévia de treinamento. Após o termino desta operação, o modelo resultante é utilizado para a filtragem do sinal de entrada na adaptação do filtro como a função transferência $C(z)$.

2.3.10 Algoritmo Híbrido VSSLMS/FxLMS

Uma das principais decisões na aplicação de um algoritmo *LMS* é o valor do *step-size* μ . Esta constante dita a velocidade de convergência do algoritmo, ficando responsável pelo desempenho em transientes, assim como o erro médio de estado estacionário. Valores mais elevados de μ geram melhores tempos de resposta, enquanto valores menores ganham em desempenho no estado estacionário.

Para aprimorar o desempenho tanto em estado estacionário quanto como rastreador, é sugerido o algoritmo *VSSLMS*, que faz a adaptação do valor do *step-size* de forma a obter a melhor resposta em sistemas variantes no tempo.

A função de adaptação do algoritmo *VSSLMS* é dada na equação (46) como:

$$w_k(n+1) = \alpha w_k(n) - 2\mu_k(n)e(n)x(n-k), \quad k = 0, 1, 2, \dots, N-1 \quad (46)$$

A função de adaptação do *step-size* é relacionada ao gradiente estocástico $g_k(n) = e(n)x(n-1)$, este é obtido durante um número determinado de amostras e iterações do algoritmo, de forma que é incrementado caso o gradiente g_k retorne valores persistentemente positivos ou negativos. Isso ocorre quando os coeficientes do filtro ainda estão longe de convergir para os coeficientes ótimos, contudo quando o algoritmo está próximo da zona de convergência, os valores de g_k oscilam rapidamente entre positivos e negativos e o *step-size* é então decrementado até um limite pré-estabelecido de forma a obter o melhor desempenho de estado permanente [13].

A especificação dos limites do parâmetro μ_k que garantem a convergência do algoritmo *VSSLMS* não é tao simples como no caso do passo de adaptação fixo μ do algoritmo *LMS*.

Este é mostrado na equação (47) [13].

$$0 < \mu < \frac{1}{3\text{tr}[\mathbf{R}]} \quad (47)$$

Sendo $\mathbf{R} = E[x(n)x^T(n)]$ a matriz de autocorrelação do vetor de entrada $x(n)$. Esta dificuldade é causada pela natureza variante no tempo do *step-size*, sendo que o vetor contendo os coeficientes do filtro para dada iteração é escrito na forma:

$$\mathbf{w}(n+1) = \alpha\mathbf{w}(n) - 2\mu(n)e(n)\mathbf{x}(n) \quad (48)$$

Sendo $\mu(n)$ a matriz diagonal contendo os parâmetros para o *step-size* $\mu_0(n)$, $\mu_1(n)$, ..., $\mu_{N-1}(n)$. Para garantir a convergência do filtro, o parâmetro linear μ é substituído pela matriz diagonal $\mu(n)$, o que leva a inequação:

$$\text{tr}[\mu(n)\mathbf{R}] < \frac{1}{3} \quad (49)$$

A equação (48) define o limite que garante a operação estável do filtro *VSSLMS*. Apesar de esta equação ditar limites dinâmicos para o passo de adaptação $\mu_k(n)$, este procedimento levaria a grande aumento da complexidade computacional. Ao invés disso, utiliza-se (47) para limitar todos os valores de μ_k a um mesmo limite superior, μ_{max} . O limite inferior, μ_{min} , pode ser tão baixo quanto zero, porém existem vantagens práticas de se utilizar um limite inferior maior do que zero, como garantir a adaptação do filtro para qualquer mudança momentânea no sinal de entrada com o menor atraso possível. Além disso, como descrito a seguir, utilizar limites superiores e inferiores múltiplos de dois garante que todos os valores de μ_k serão múltiplos de dois, diminuindo muito a complexidade computacional do algoritmo, visto que operações com potências de dois são processadas como uma operação de translação ao invés de uma operação de multiplicação comum. Este pequeno ajuste faz com que o algoritmo híbrido *VSSLMS/FxLMS* tenha um custo computacional menor do que o algoritmo *FxLMS* por si só. Desta forma, o parâmetro $\mu(n)$ é adaptado de acordo com a equação (50):

$$\mu(n) = 2^p \mu(n-1) \quad (50)$$

Sendo $p = 1, -1$ ou 0 dependendo dos sinais de $e = e(n)$, $e_i = e(n-1)$ e $e_{ii} = e(n-2)$, de acordo com um parâmetro de decisão *thd*, como apresentado no fluxograma da Figura 6, sendo $\mu_{min} = \mu_{LMS}$ o mínimo passo de adaptação do algoritmo e μ_{max} o limite superior da mesma variável dado por:

$$\mu_{max} = \frac{1}{3tr[R]} \quad (51)$$

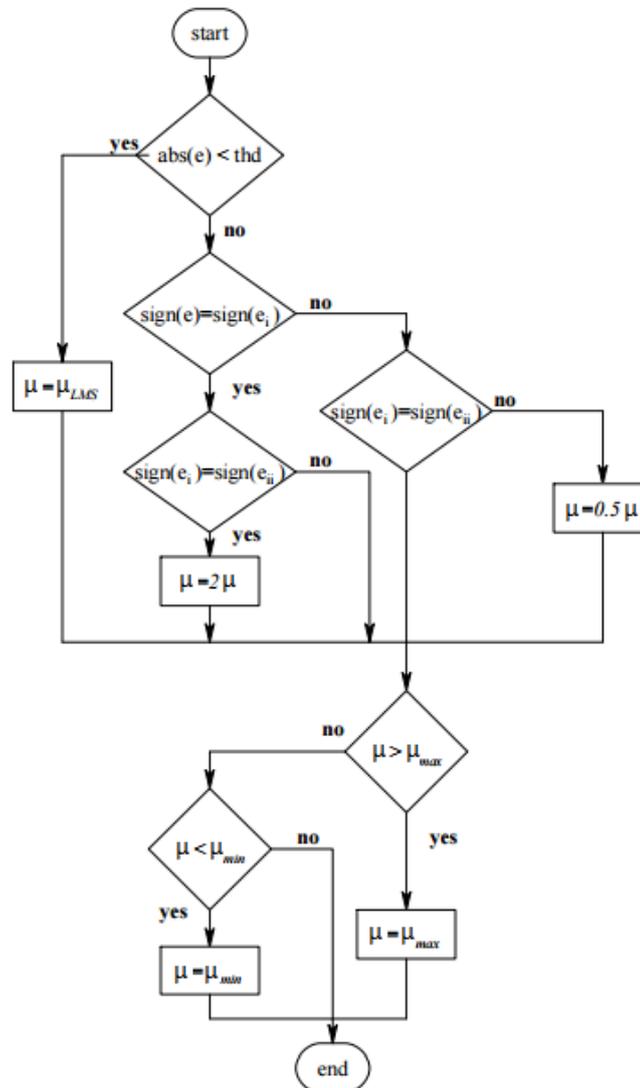


Figura 6: Algoritmo de decisão para adaptação do step-size de acordo com parâmetro thd.

2.3.11 Algoritmo LMS de Passo Variável para Ambientes com Razão Sinal-Ruído Baixa (FxNLMSVS)

Em aplicações onde a razão sinal-ruído do sinal com o qual é alimentado o algoritmo LMS é baixa, o desempenho do mesmo não é satisfatório. Contudo, na aplicação proposta para este algoritmo no presente trabalho, sinais de entrada com razão sinal-ruído baixas são comuns em estado estacionário, e inclusive desejáveis, já que se tem como alvo a atenuação

do sinal de entrada no ambiente, enquanto ruídos não correlacionados como o do vento, música e fala dos passageiros não devem ser modificados.

Um algoritmo de passo variável como o descrito no item 2.3.10 utiliza o gradiente estocástico para determinar o passo de adaptação a ser utilizado no processamento da amostra subsequente. Contudo, este tipo de algoritmo sofre em ambientes com baixa relação sinal-ruído. Para solucionar este problema é proposto em [14] um algoritmo de passo variável baseado na medida da auto-correlação de várias amostras atrasadas do sinal de erro, $lag(1), lag(2), \dots, lag(M)$, sendo M a ordem do filtro adaptativo. É importante ressaltar que o uso de várias amostras de atraso aumenta apenas marginalmente a complexidade computacional do algoritmo [14].

Neste algoritmo, considera-se $p(n)$ como o valor médio da soma dos quadrados das funções de autocorrelação entre $e(n)$ e $e(n-1), e(n-2), \dots, e(n-N)$, estimando através da seguinte expressão:

$$p(n) = \beta p(n-1) + (1-\beta) \sum_{i=1}^M [e(n)e(n-i)]^2 \quad (52)$$

Sendo $\beta > 0$ o parâmetro de controle de filtragem.

A equação de adaptação do passo é dada por:

$$\mu(n+1) = \alpha \mu(n) + \gamma p(n) \quad (53)$$

Com α e γ sendo parâmetros de controle do ajuste do passo de adaptação.

2.4 Complexidade de Algoritmo

2.4.1 Introdução

Algoritmo é a palavra que define um processo ou um conjunto de regras para realização de cálculos. Dado um algoritmo proposto para a solução de determinado problema, pode-se analisar este em dois quesitos principais em função da resposta, são eles:

- corretude;
- eficiência.

Garantir a corretude do algoritmo não é trivial, e este procedimento é geralmente realizado utilizando-se indução matemática. A corretude do algoritmo quantifica a proximidade entre respostas esperadas e saídas do algoritmo para determinado conjunto de entradas. A eficiência do algoritmo, por outro lado, descreve o desempenho versus a quantidade de recursos utilizado para a resolução do problema em questão. Neste contexto, a complexidade do algoritmo mede a quantidade de trabalho requerido para solucionar um determinado problema [15]. Critérios de medidas de complexidade são geralmente: espaço em memória e tempo de execução. Nesta seção será dada ênfase à complexidade de tempo.

2.4.2 Análise Assintótica

A análise assintótica de um algoritmo é realizada assumindo-se valores de n muito grandes, sendo n o tamanho do vetor de entrada ou conjunto de dados de entrada do algoritmo. Esta análise é dividida em duas partes, a saber, a ordem O e a ordem Ω [15].

Diz-se $f(n) = O(g(n))$ quando a ordem da função $g(n)$ é superior a ordem da função $f(n)$ para valores elevados de n , ou segundo a equação (54), sendo:

$$f(n) \leq c \cdot g(n) \quad (54)$$

Sendo c uma constante positiva.

Diz-se que $f(n) = \Omega(g(n))$ quando a ordem da função $g(n)$ é inferior a ordem da função $f(n)$ para valores elevados de n , ou segundo a equação (55):

$$f(n) \geq c \cdot g(n) \quad (55)$$

Sendo c uma constante positiva.

Desta forma, observa-se que a análise assintótica da complexidade é baseada nestas definições e estabelece uma ordem relativa entre o crescimento das funções para valores elevados de n .

Para realizar a análise de tempo do algoritmo, utiliza-se a notação $O(\cdot)$, ou *Big-Oh*, percorrendo os passos a seguir [16].

1. Separar $f(n)$ em duas partes, a saber, o termo dominante e os termos de ordem inferior.
2. Ignora-se os termos de ordem inferior.

3. Ignora-se constantes de proporcionalidade.

Para encontrar o tempo de execução do algoritmo, considera-se que as instruções são executadas sequencialmente, e que cada conjunto de instruções simples é executado em uma unidade de tempo. Em estruturas de instruções mais complexas, o tempo de execução é definido como a seguir.

- Laços: O tempo de execução do laço pé no máximo o tempo das execuções das instruções contidas no laço vezes o número de iterações.
- Aninhamento de laços: Analisa-se primeiro os laços internos, sendo o tempo de execução total de um grupo de laços aninhados igual ao tempo de execução da instrução vezes o produto dos tamanhos de todos os laços.
- Instruções consecutivas: Efetua-se a soma dos tempos para cada instrução.
- Chamada de funções: Calcula-se a complexidade da função separadamente, e então utiliza-se esta medida para computar a complexidade do algoritmo como se a função fosse apenas uma operação de tempo igual o tempo que a função leva para ser completada.
- Recursão: Faz-se a linearização do laço recursivo através da substituição da chamada recursiva por laços aninhados. Quando esta operação não é possível, deve-se calcular o tempo de execução através de relações de recorrência.

Deve-se ter em mente que a análise assintótica funciona para casos onde o tamanho do vetor de entrada n é muito grande. Assim, uma função com custo computacional n^3 operações, por exemplo, será computada mais rapidamente do que uma com o custo $n^2 + 500000$ operações se o tamanho n do vetor de entrada não for superior a $n = 79$.

Diz-se ainda que o algoritmo tem complexidade exata $\Theta(f(n))$ (*Big theta*) se ele é ao mesmo tempo $f(n) = O(g(n))$ e $f(n) = \Omega(g(n))$ [17].

3 Simulação em Ambiente Virtual

Nessa seção serão descritos os procedimentos e modelos utilizados na simulação dos algoritmos em ambiente virtual.

3.1 Modelo Acústico

Visando obter validação dos resultados das simulações de desempenho, utilizou-se funções transferência para os modelos dos dutos acústicos dos caminhos primário e secundário obtidos e validados experimentalmente [18] em dados em (56) e (57), respectivamente:

$$\frac{P(x_m, s)}{Q_a(s)} = \frac{\rho_0 c_0 \cosh s(L - x_m)/c_0}{s \sinh sL/c_0} \quad (56)$$

$$\frac{P(x_m, s)}{Q_b(s)} = \frac{\rho_0 c_0 \cosh s(L - x_s)/c_0}{s \sinh sL/c_0} \quad (57)$$

Sendo $\rho_0 = 1,21 \text{ kg/m}^3$ a densidade do ar, $c_0 = 343 \text{ m/s}$ a velocidade do som no ar, L o comprimento do duto, x_s a posição do microfone de referência em relação ao autofalante Q_b , x_m a posição do microfone em relação ao início do duto e $\beta = s/c_0$ o módulo do volume volumétrico adiabático, considerando-se que a fonte de ruído está localizada no ponto de referência $x = 0$ no início do duto [18], como exibido na Figura 7.

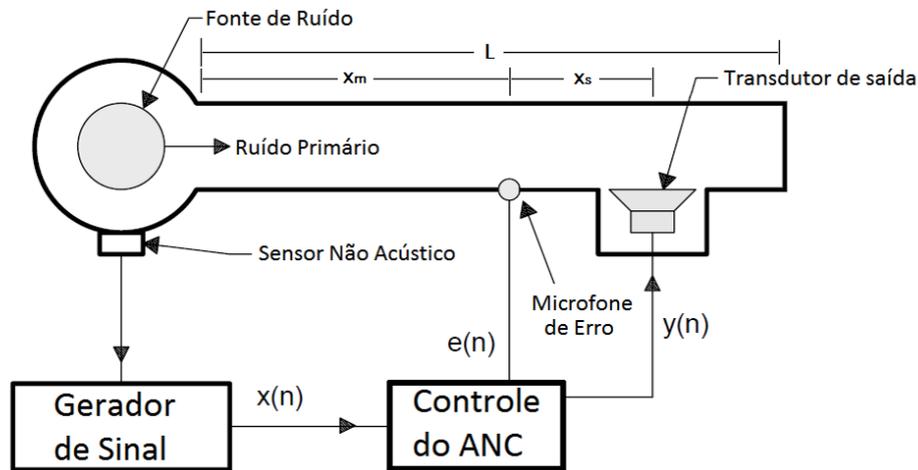


Figura 7: Duto Acústico

Além disso, as equações que modelam as respostas acústicas, na faixa de frequências de interesse, do autofalante e do microfone utilizados no sistema são dadas por (58).

$$\frac{V_{x_m}(x)}{sU(x_m, s)} = 0,0056 \text{ N/m}^2 \quad (58)$$

Este modelo é válido considerando meio fluido contínuo e homogêneo, perfeitamente elástico, processo isentrópico e ainda invariante no tempo. Desta forma, a função transferência do caminho secundário é dada por (59).

$$H(s) = -0,0056 \cdot \frac{\rho_0 c_0 \cosh s(L - x_m)/c_0}{s \sinh sL/c_0} \quad (59)$$

Para que estas funções possam ser inseridas no modelo para simulação, utiliza-se a série de Taylor para expandir as funções trigonométricas de s [19], sendo:

$$\sinh(s) = \sum_{n=0}^{\infty} \frac{s^{2n+1}}{(2n+1)!} = s + \frac{s^3}{3!} + \frac{s^5}{5!} + \frac{s^7}{7!} + \dots \quad (60)$$

$$\cosh(s) = \sum_{n=0}^{\infty} \frac{s^{2n}}{(2n)!} = 1 + \frac{s^2}{2!} + \frac{s^4}{4!} + \frac{s^6}{6!} + \dots \quad (61)$$

De posse dos coeficientes da função transferência representando cada caminho, utiliza-se filtros IIR para implementá-las na simulação. Os diagramas de Bode representando a resposta em frequências dos caminhos primário e secundário podem ser observados nas Figuras a seguir.

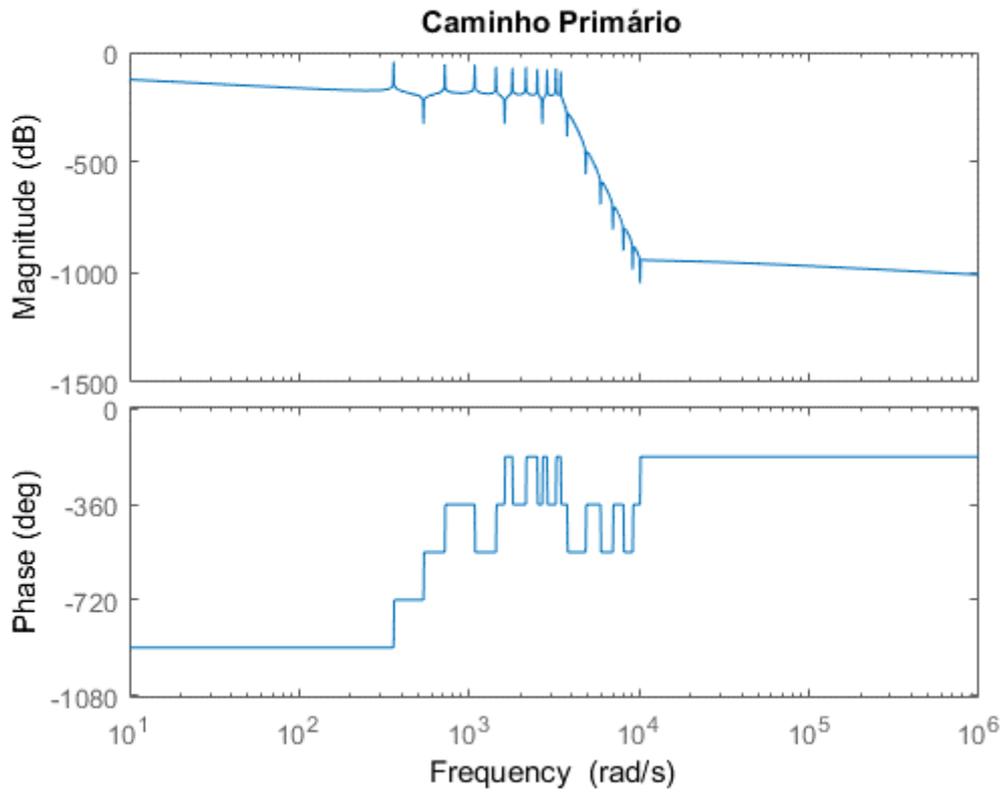


Figura 8: Diagrama de bode da função transferência do caminho primário.

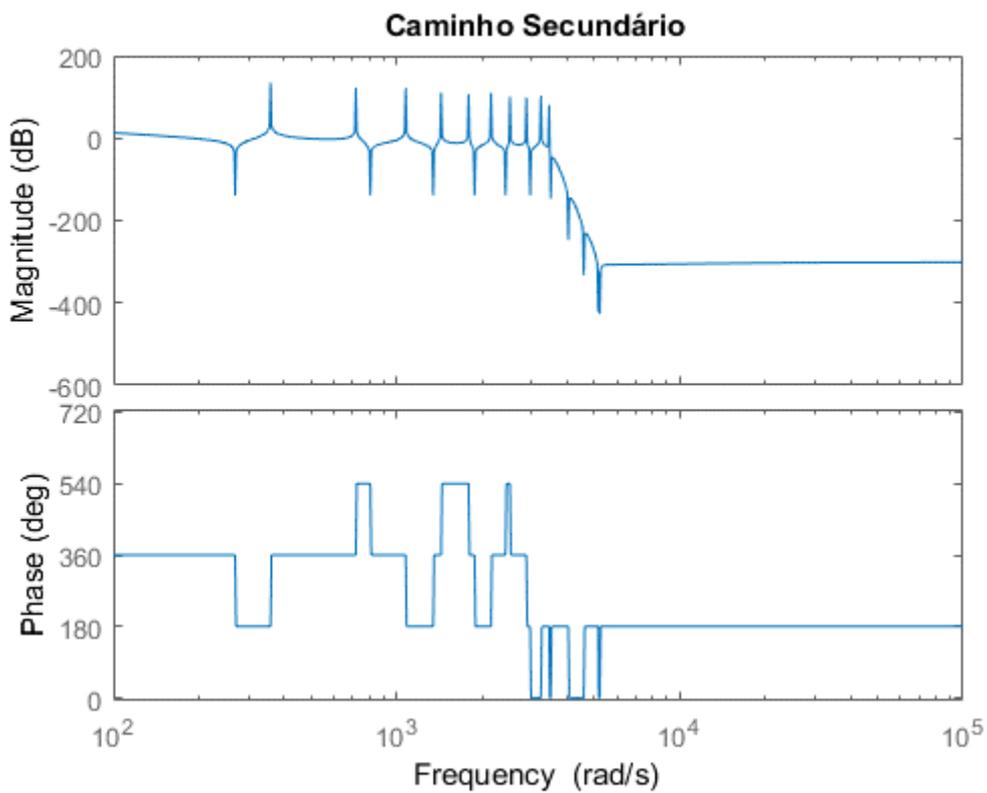


Figura 9: Diagrama de bode da função transferência do caminho secundário.

Nota-se a não linearidade da função transferência da pressão no espaço em função da frequência dentro do duto fechado, isto se dá devido a formação de ondas estacionárias dentro do tubo, criando regiões de pressão características em pontos específicos do duto. As regiões de maior e menor pressão sonora em determinada frequência dentro do duto equivalem as regiões de formação de cristas e vales das componentes fundamentais e harmônicas naquela dada frequência. Pode-se observar também a ampla gama de deslocamento de fase para determinadas frequências, chegando a ocorrer deslocamentos de até 720° em regiões de frequência relativamente pequenas. Este efeito constitui a maior dificuldade neste tipo de aplicação específica, forçando o algoritmo a responder de forma rápida a mudanças abruptas na fase do ruído a ser cancelado.

Os diagramas da Figura 8 e da Figura 9 são obtidos fazendo-se $x_m = 2\text{ m}$ e $x_s = 1\text{ m}$, sendo assim o microfone de referência encontra-se posicionado a 2 metros da extremidade do duto e o transdutor de saída encontra-se no fim do duto com a face virada para dentro.

3.2 Algoritmos Utilizados

3.2.1 Treinamento *Off-line* do Modelo do Caminho Secundário

Como descrito no item 2.3.8, o caminho secundário tem efeito significativo sobre o desempenho do sistema de controle ativo de ruído. Nesta análise, a estimativa do caminho secundário é feita de forma *off-line*. Isto quer dizer que o processo de modelagem e controle da planta acústica é feito em duas etapas.

1 - Inicialmente, com o sistema de controle principal desligado, é injetado no sistema, através do transdutor de controle, um ruído branco. Este ruído é captado pelo microfone de referência, passa por um conversor analógico-digital, e é então somado eletricamente ao mesmo ruído branco modificado por um filtro $C(z)$. Este filtro é a estimativa do caminho secundário. O resultado desta soma é o sinal de referência para um algoritmo adaptativo LMS que faz a adaptação dos coeficientes do filtro a cada amostra, de forma a minimizar o erro residual presente, neste caso a soma dos sinais captado pelo microfone e filtrado por $C(z)$. Desta forma, ao reduzir o sinal de erro a um nível abaixo de um limiar aceitável, obtém-se a estimativa final do caminho secundário a ser utilizada no algoritmo principal de redução de ruído acústico [2].

2 - Após o término da fase de treinamento, o filtro $C(z)$ é utilizado para tratar o sinal de entrada do algoritmo FX-LMS, afim de compensar os efeitos do caminho secundário $H(z)$. Como este algoritmo apresenta boa robustez a erros na estimativa do caminho secundário, não há necessidade de uma fase de treinamento demasiadamente longa, bastando poucas dezenas de segundos para obter-se estimativa satisfatória para o funcionamento do sistema [2].

A Figura 10 apresenta o sistema descrito nos itens anteriores.

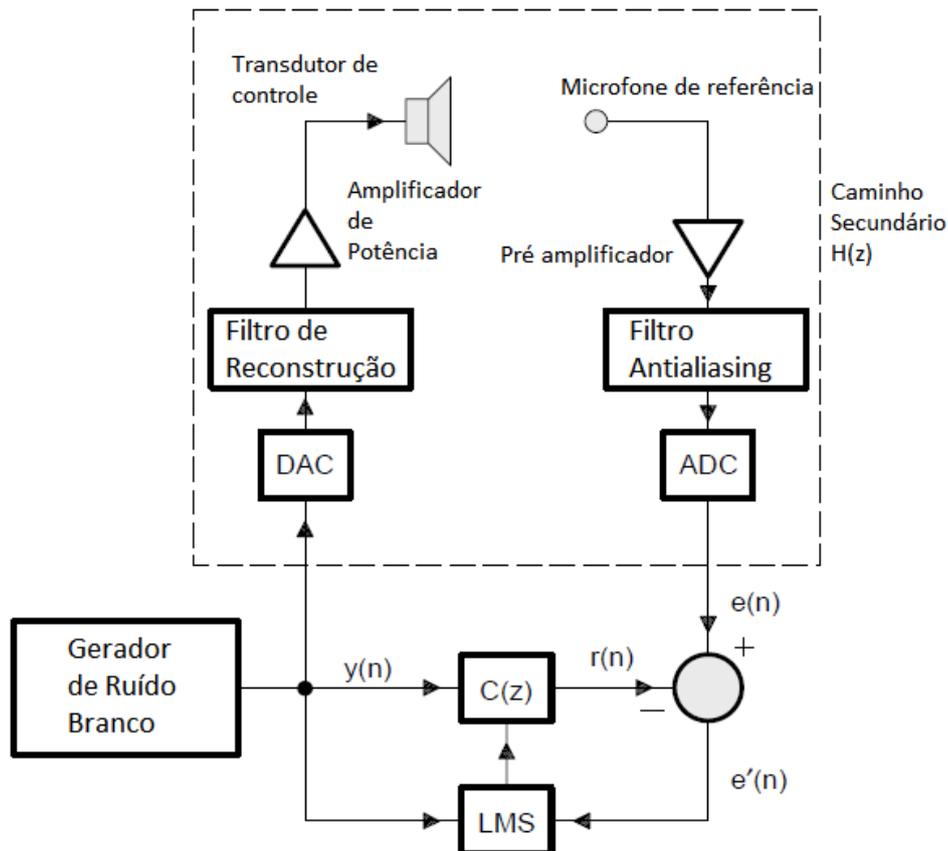


Figura 10: Modelagem *off-line* do caminho secundário, $C(Z)$ [2].

3.2.2 Algoritmos de Controle

No processo de desenvolvimento e teste de desempenho do controlador ativo de ruído foram avaliados quatro algoritmos principais baseando-se em resultados prévios em aplicações similares com crescente nível de complexidade de um para outro, são eles:

- O algoritmo FxLMS com passo fixo
- O algoritmo FxNLMS com passo fixo
- O Algoritmo FxNLMS de passo adaptativo para ambientes com baixa razão sinal-ruído

- O Algoritmo FxNLMS de passo adaptativo para ambientes com baixa razão sinal ruído e ganho de realimentação variável.

3.2.2.1 Algoritmo FxNLMS de Passo Adaptativo e Ganho de Realimentação Variável (FxNLMSVSAG)

O algoritmo final proposto nesta monografia utiliza passo variável estimado a partir de N sinais de correlação cruzada do sinal de referência com seus respectivos $lag(1), lag(2), \dots, lag(N)$, de forma similar ao algoritmo apresentado no item 2.3.11, porém o valor final do passo de adaptação é dividido por metade do valor do ganho utilizado no sinal de referência que alimenta o algoritmo de adaptação dos pesos do filtro, de acordo com a equação (62):

$$\mu(n+1) = \frac{\alpha \mu(n) + \gamma p(n)}{\tau K_{fb}} \quad (62)$$

Sendo K_{fb} o valor do ganho do sinal de realimentação do algoritmo LMS, este ajustável pelo usuário por meio de interface gráfica, e τ uma variável de controle do algoritmo. Este ganho trata de ajustar o valor final do ruído residual após a convergência do algoritmo. A possibilidade de ajuste deste valor é um dos requisitos para o produto final. O motivo é a necessidade da existência de um som proveniente do motor, ainda que atenuado, para auxílio no diagnóstico do piloto caso ocorra alguma situação fora do comum com o motor. Assim, o sinal de referência de entrada do algoritmo LMS é definido por:

$$e(n) = K_{fb}d(n) - y(n) \quad (63)$$

A adição deste ganho ao sinal de referência age de forma a acelerar a convergência do algoritmo e possibilita um aumento real no desempenho de atenuação quando comparado ao algoritmo sem controle de ganho.

3.2.2.2 Erro Quadrático Médio

Em análise estatística de modelos e na avaliação de modelos matemáticos, um indicador comumente utilizado para medir o desempenho do modelo é o erro quadrático médio (MSE) e sua evolução no tempo. O MSE é exibido pela equação (64) [20]:

$$MSE = E\{|\varepsilon(n)|^2\} \quad (64)$$

Assim como indicador de desempenho, o MSE pode ser utilizado no processo de escolhas dos valores das variáveis que compõe o filtro como critério de escolha em algoritmo de decisão [21].

3.2.2.3 Custo computacional dos Algoritmos Utilizados

A complexidade do algoritmo é um fator crucial e inviabilizador em processos de tempo real. Na estrutura de cancelamento de ruído acústico *feedforward*, três grandes operações são executadas no sinal antes que este possa ser mandado para o transdutor de saída. São estas: a geração do sinal de controle, a aplicação do filtro FIR referente a modelagem do caminho secundário ao sinal e a atualização dos pesos do controlador.

As operações necessárias para cada ciclo de cada algoritmo utilizado encontram-se descritas na Tabela 1.

Tabela 1: Custos computacionais comparados dos algoritmos estudados.

Algoritmo	Op.	Geração do Sinal de Controle	Op. Fx do Caminho Secundário	Atualização dos Pesos do Filtro	Total
FxLMS	*	N	L	$N + 1$	$2N + L + 1$
	+	$N - 1$	$L - 1$	N	$2N + L - 2$
FxNLMS	*	N	L	$2N + 2$	$3N + L + 2$
	+	$N - 1$	$L - 1$	$2N$	$3N + L - 2$
HYBRIDO	*	N	L	N	$2N + L$
VSSLMS/LMS	+	$N - 1$	$L - 1$	$N + 1$	$2N + L - 1$
FxNLMSVS	*	N	L	$2N + 2M + 6$	$3N + 2M + L + 6$
	+	$N - 1$	$L - 1$	$2N + M + 3$	$3N + M + L + 1$
FxNLMSVSAG	*	N	L	$2N + 2M + 8$	$3N + 2M + L + 8$
	+	$N - 1$	$L - 1$	$2N + M + 3$	$2N + M + 3$

Observa-se a crescente complexidade de algoritmo para versões mais robustas do código, ainda assim todos os algoritmos descritos se enquadram na categoria $O(n)$, considerados de fácil manuseio. Tomando-se apenas o número de multiplicações, este o fator dominante no tempo de processador gasto para computar determinada operação, observa-se que para $N = 128$, $L = 128$ e $M = 32$, são necessárias 584 operações para realizar um ciclo do algoritmo de passo e ganho variável proposto, 582 operações para o algoritmo FxNLMSVS, 514 para o algoritmo FxNLMS, enquanto para o algoritmo FxLMS são necessárias apenas 385 operações. Observa-se que para crescentes complexidades de planta a serem modeladas, maiores deverão ser os valores de N , L e M , muito embora deva-se evitar a sobre-parametrização. Valores ideais para os tamanhos dos filtros devem ser estudados para cada caso de aplicação, e aumentados quando necessário. Técnicas para a escolha dos valores dos tamanhos dos filtros são apresentadas em [22], [23] e [24].

Nota-se que para obtenção de controle de ganho e de adaptação de passo do algoritmo foram adicionadas 199 operações ao ciclo de processamento quando comparado o algoritmo FxNLMSVSAG mais robusto ao mais simples FxLMS. Esta soma corresponde a uma adição de carga computacional de 51%.

Caso a complexidade computacional seja um fator limitante para viabilidade do produto, pode-se ainda obter-se o processamento completo de um ciclo do algoritmo VSSLMS com apenas 384 operações de multiplicação, conservando-se o benefício do passo de adaptação do algoritmo variável. A Figura 11 [13] apresenta a relação de ganho de velocidade de processamento entre os filtros VSSLMS/LMS e LMS convencional em porcentagem.

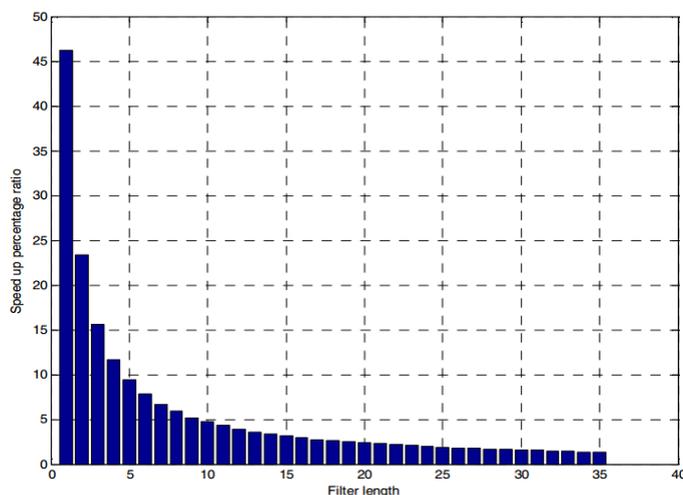


Figura 11: Ganho de desempenho computacional entre os algoritmos VSSLMS/LMS e LMS convencional [13].

4 Resultados e Discussão

Os resultados simulados obtidos neste trabalho tiveram como frequência de amostragem da simulação 44,1KHz, suficiente para que não ocorra *aliasing* dentro da faixa de frequências de interesse no projeto. As simulações do treinamento do modelo do caminho secundário e do algoritmo de controle do ANC foram realizadas no ambiente virtual SIMULINK, e as simulações e estudo da planta acústica foram realizadas em MATLAB *script* e então seus coeficientes importados para o ambiente virtual SIMULINK para serem utilizadas em filtros FIR nos modelos estudados.

A faixa de frequências estudada foi de 0Hz a 4 KHz para a componente fundamental de entrada do sistema e seus respectivos 3º e 4º harmônicos. Estas componentes de frequência concentram a maior parte da distribuição de potência do ruído a ser anulado, dado que estes são produzidas pelo giro principal do eixo do motor, pelo ruído do giro da hélice tripá WARPDRIVE, e pela combustão nos quatro pistões dos motores ROTAX escolhidos para teste [25].

Para estimar a robustez do filtro, o sinal de referência é contaminado com um ruído branco com distribuição normal e potência coerente com a contaminação na aplicação.

Todos os testes foram realizados partindo do estado inicial desligado e cada simulação teve um tempo total de dois segundos, suficiente para que todos os modelos pudessem ser avaliados em relação a seu desempenho e velocidade de convergência. Os resultados serão avaliados com base na atenuação do sinal de interesse, bem como na velocidade e robustez em transientes dos algoritmos.

Os tamanhos dos vetores utilizados para a filtragem dos sinais em todos os testes foram $N = 128$, sendo este o tamanho do filtro do algoritmo de controle, e $L = 128$ o tamanho do filtro que modela o caminho secundário em treinamento *off-line*. Estes valores foram definidos de acordo com os critérios descritos em [22] e [24], sendo os valores encontrados os menores valores possíveis de números de pesos dos filtros de controle que

modelam eficientemente o sistema sem ocorrer sobre-parametrização. O valor da variável α foi mantida unitária para todos os testes.

Os filtros FxLMS e FxNLMS obtiveram respostas muito semelhantes conforme o esperado e seus resultados foram condensados, apresentados na Figura 12, afim de evitar redundâncias nas análises. As figuras a seguir exibem o comportamento de cada filtro para frequências de excitação sobrepostas de 3.5KHz, 10Kz e 15KHz. Estas frequências representam o pior caso a qual os filtros serão submetidos [25], e em todas as frequências inferiores os resultados foram superiores aos obtidos nestes testes.

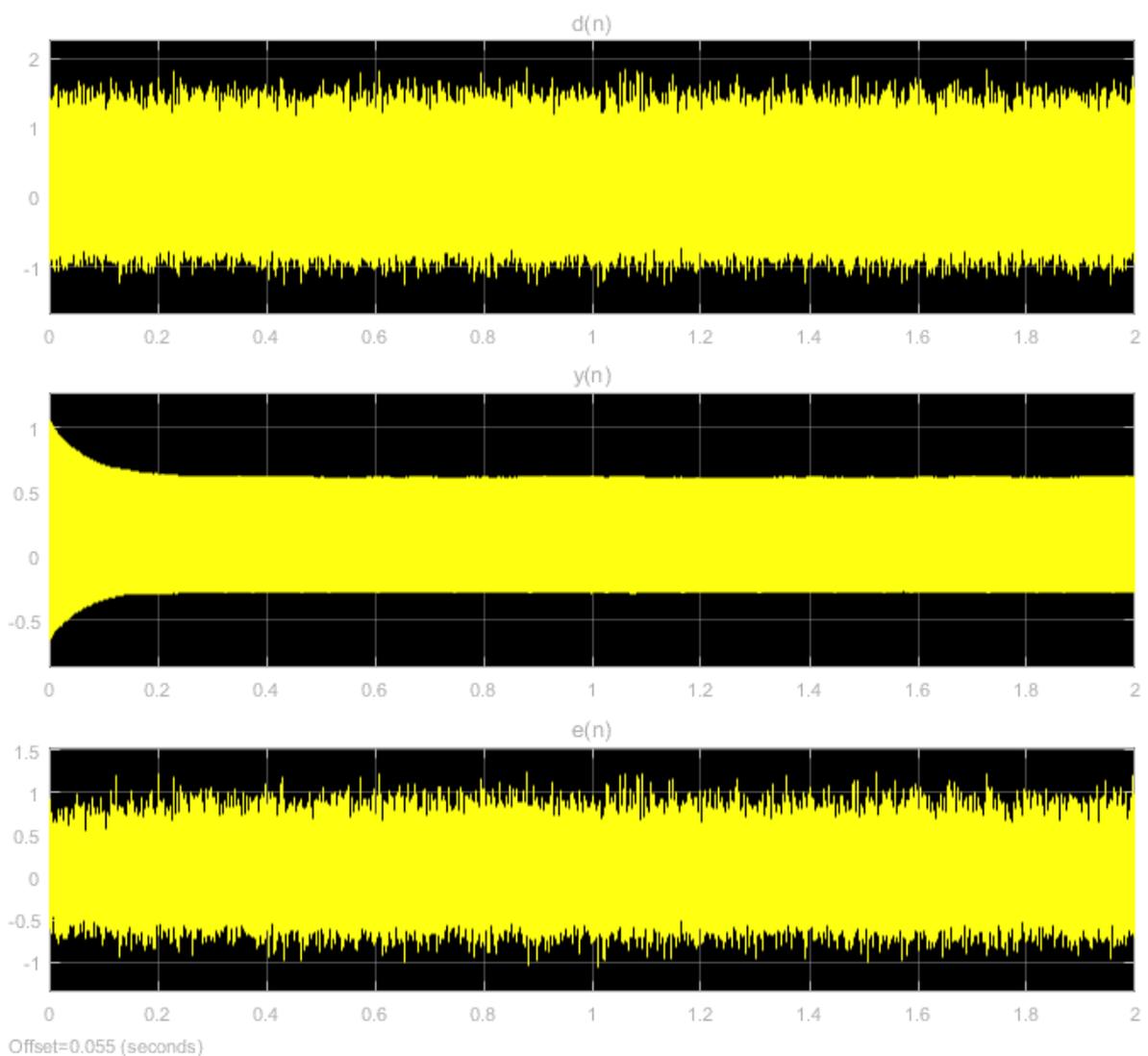


Figura 12: Resultados da simulação dos algoritmos FxLMS e FxNLMS. De cima para baixo tem-se o ruído primário contaminado com ruído branco aditivo, o antiruído gerado pelo filtro e o ruído residual após a superposição das ondas sonoras.

Avalia-se que os modelos FxLMS e FxNLMS obtiveram convergência razoavelmente rápida e nível de atenuação do ruído correlacionado na zona silenciosa de 6 dB. O passo de adaptação dos pesos do filtro utilizado foi de $\mu = 10^{-3}$. Este algoritmo respondeu de forma semelhante para uma faixa de valores de passo de $\mu = 10^{-5}$ á $\mu = 5 * 10^{-3}$. Para valores menores de μ observou-se um aumento demasiado no tempo de convergência, e para valores maiores do que $\mu = 5 * 10^{-3}$ o algoritmo LMS se torna instável enquanto o algoritmo FxNLMS começa a apresentar oscilações na resposta do filtro. Pode-se verificar também como o modelo trata de cancelar apenas os ruídos correlacionados, mantendo intacto o ruído branco aditivo. Este requisito é importante visto que ruídos não correlacionados como o vento ou ruídos provenientes de falhas servem de referência para o piloto durante o voo.

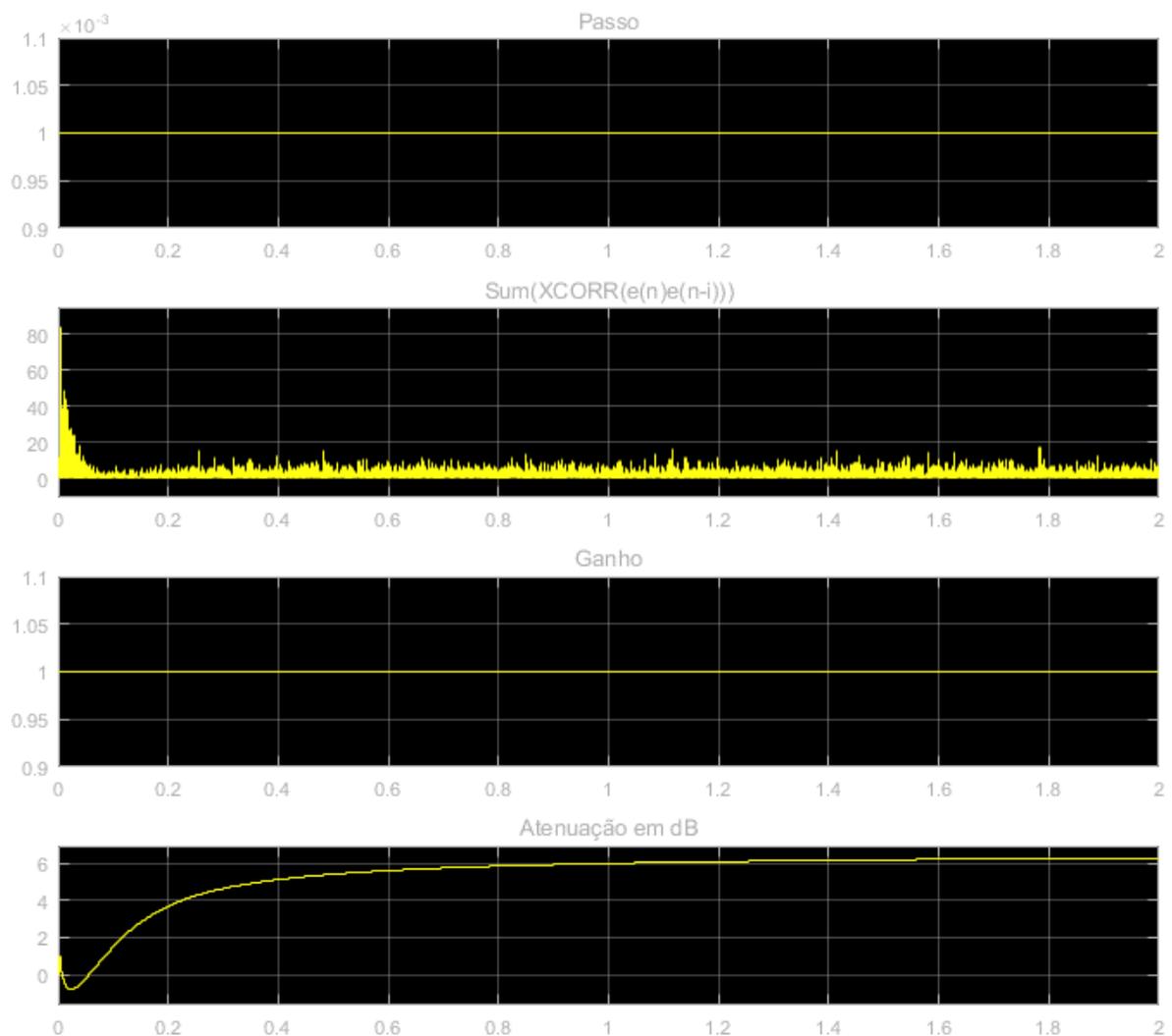


Figura 13: Resultados da simulação dos algoritmos FxLMS e FxNLMS. De cima para baixo tem-se o passo de adaptação, a soma das correlações cruzadas quadráticas do sinal de erro com seus *lags*, o ganho no *feedback* do algoritmo e a amplitude do ruído residual em dB.

Fazendo uma análise da resposta da soma das correlações do ruído residual com seus consecutivos *lags*, pode-se observar a manutenção de um nível médio de sinal durante todo o teste. Isto indica que o algoritmo não obtém melhora significativa no nível de atenuação de ruído com o passar do tempo depois dos primeiros 1.4 segundos.

Na Figura 14 pode-se observar o espectrograma do ruído primário contaminado com ruído branco, em amarelo, juntamente com o espectrograma do ruído residual na zona silenciosa, na cor azul.

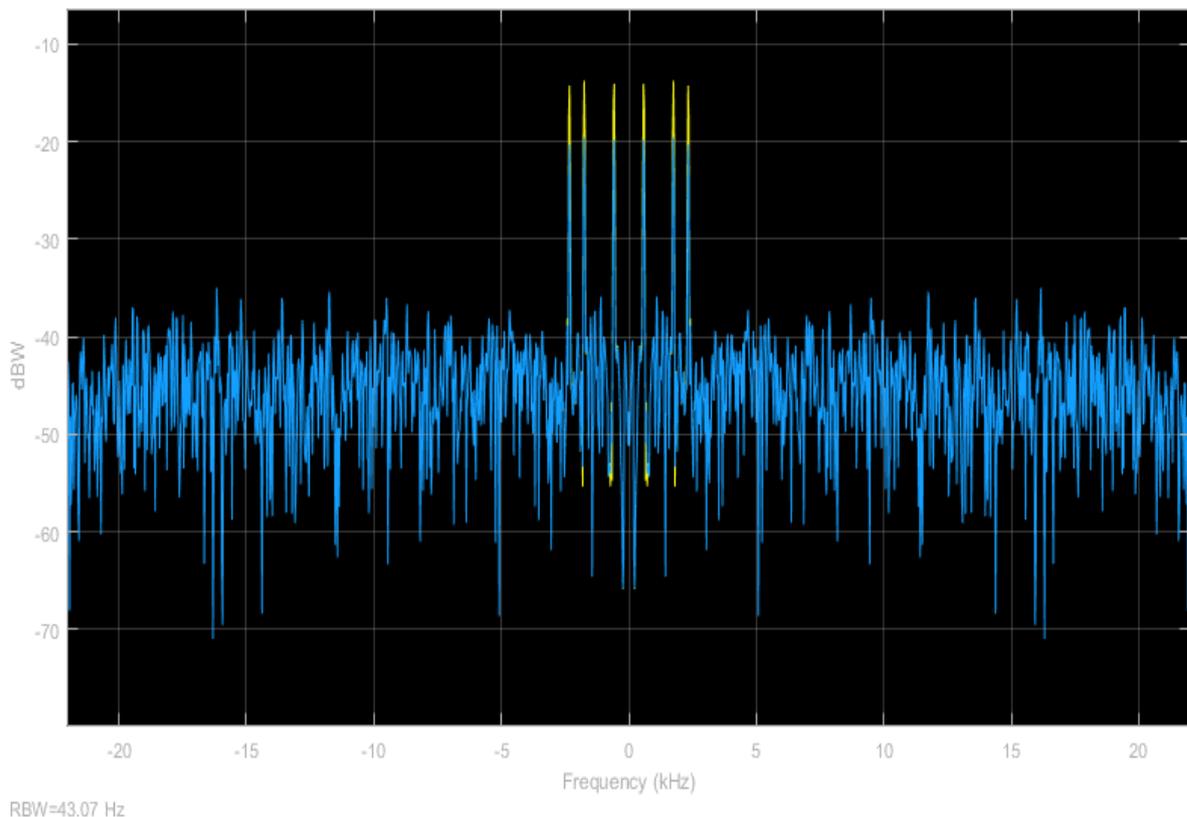


Figura 14: Espectrograma do sinal de ruído residual na zona silenciosa para simulação com os algoritmos FxLMS e FxNLMS.

Pode-se constatar que grande parte da potência do ruído residual ainda se encontra concentrada nas faixas de frequências do ruído primário. As excitações em banda larga observadas são de responsabilidade do ruído branco aditivo que espalha sua potência por todo o espectro.

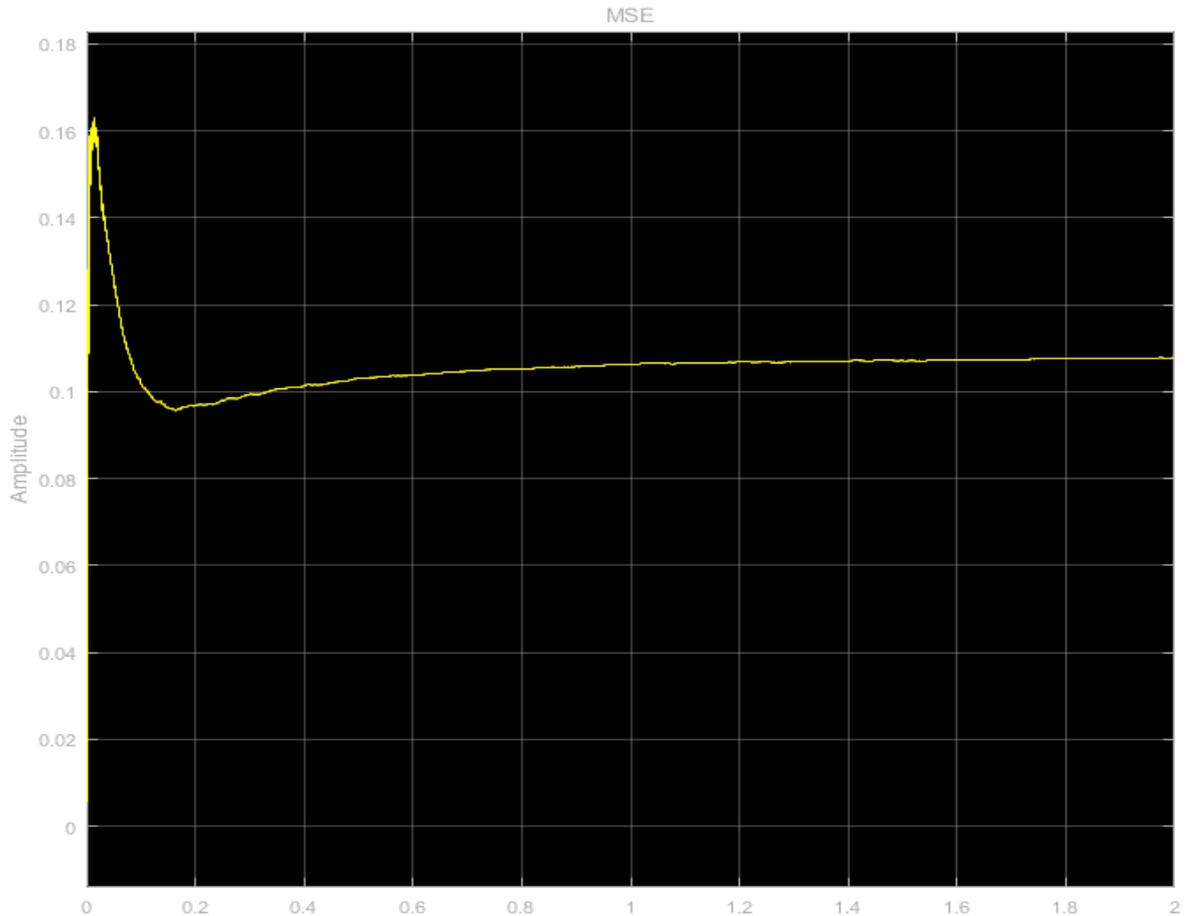


Figura 15: MSE dos Algoritmo FxLMS e FxNLMS.

A evolução do MSE no tempo de testes com os algoritmos FxLMS e FxNLMS é exibida na Figura 15. Este diagrama foi adquirido sem a presença do ruído branco contaminante para que pudesse ser demonstrado o tempo de convergência e o MSE em estado estacionário dos algoritmos.

Para o algoritmo FxNLMSVS utilizou-se 32 amostras de $lag(n)$ para realizar a adaptação do passo do algoritmo. Além disso, foram empregados $\alpha = 0.99$, $\beta = 0.99$ e $\gamma = 10^{-4}$. Estes hiperparâmetros foram escolhidos de forma a garantir a convergência do algoritmo para toda a faixa de frequências testada, e ainda robustez a mudanças súbitas na potência do sinal de entrada causadas pela variação da frequência do sinal.

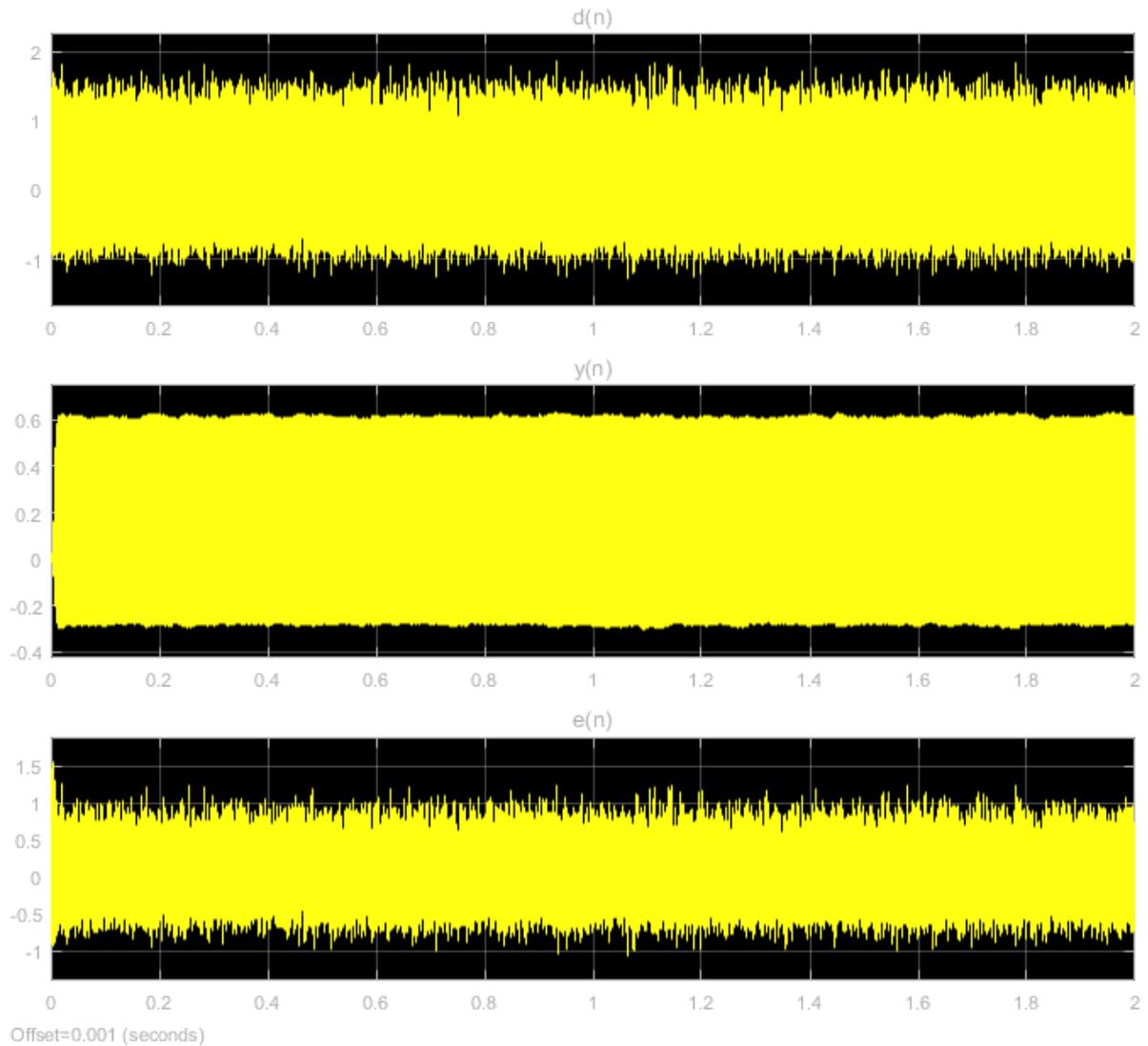


Figura 16: Resultados da simulação do algoritmo FxNLMSVS. De cima para baixo tem-se o ruído primário contaminado com ruído branco aditivo, o antiruído gerado pelo filtro e o ruído residual após a superposição das ondas sonoras.

Nota-se a rápida convergência do algoritmo, diminuindo o tempo de resposta em relação ao algoritmo FxNLMS de aproximadamente 1.4 segundos para menos de 0.5 segundos. Apesar deste benefício, a eficiência da atenuação não foi superior à do algoritmo mais simples, com o nível de atenuação na zona silenciosa ficando na casa dos 6 dB.

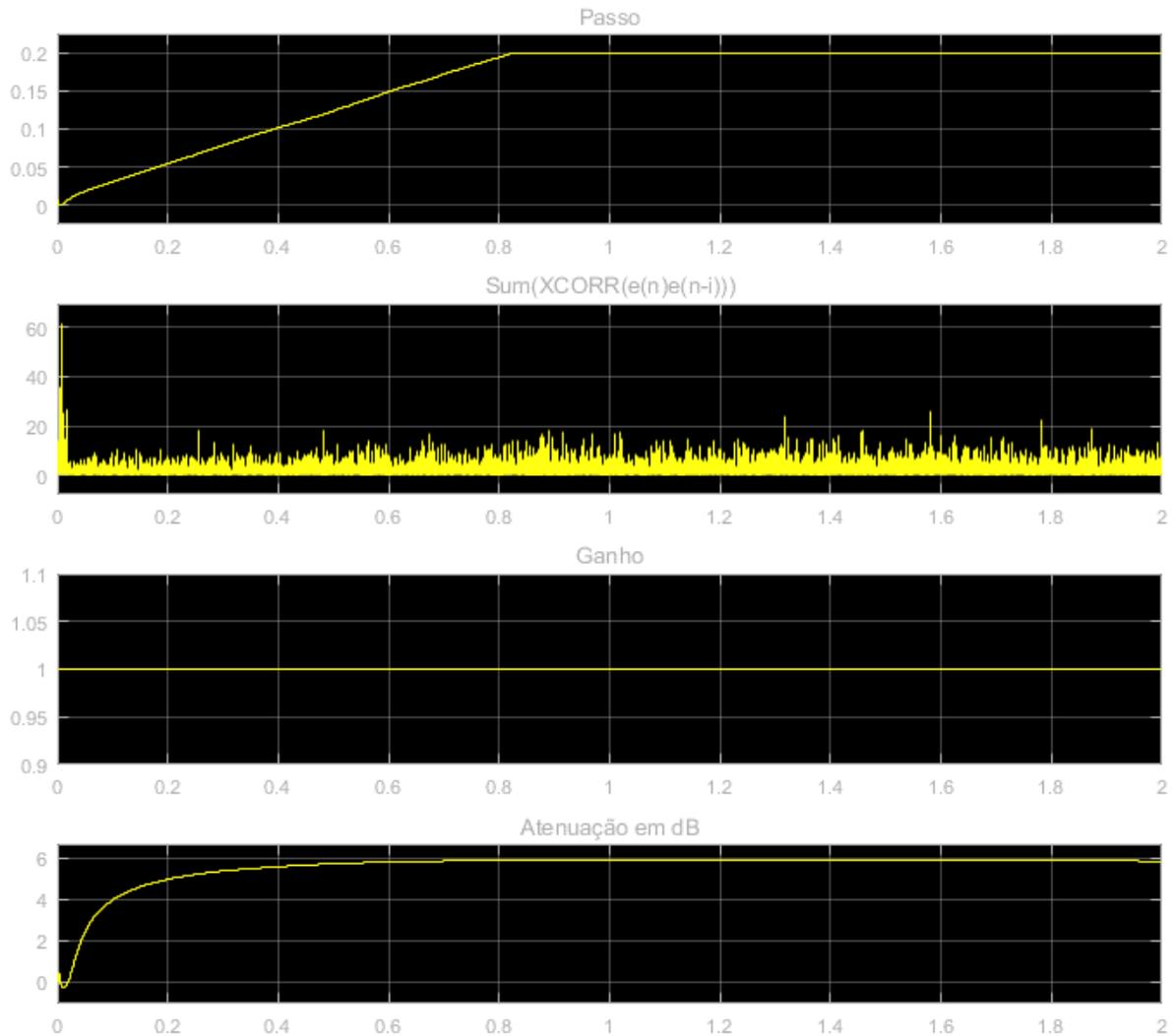


Figura 17: Resultados da simulação do algoritmo FxNLMSVS. De cima para baixo tem-se o passo de adaptação, a soma das correlações cruzadas quadráticas do sinal de erro com seus lags, o ganho no feedback do algoritmo e a amplitude do ruído residual em dB.

Na Figura 17 pode-se observar a ação do algoritmo que varia o passo de adaptação do estimador LMS que atualiza os coeficientes do filtro de controle, sendo que este obteve desempenho satisfatório para apenas 32 valores de *lag* diferentes na entrada da função de autocorrelação. Valores maiores do vetor de *lag* para a correlação cruzada do erro poderiam ser escolhidos, isto resultaria em maior robustez na estimação do passo de adaptação a ser utilizado, evitando que este seja diminuído de maneira prematura afetando o desempenho do algoritmo [14].

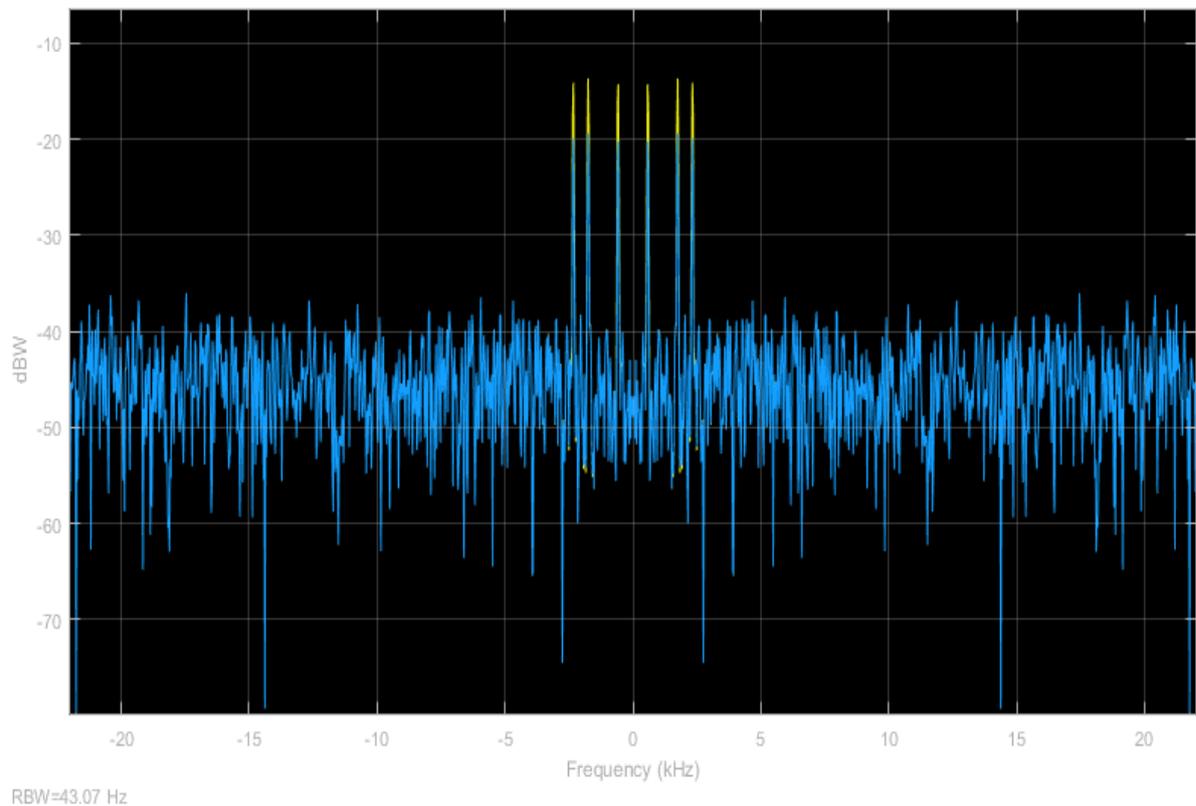


Figura 18: Espectrograma do sinal de ruído residual na zona silenciosa para simulação com o algoritmo FxNLMSVS.

Testes com variados níveis de ruído aditivo demonstraram a robustez do algoritmo FxNLMSVS em ambientes com baixa razão sinal/ruído, com rápida convergência e grande estabilidade. Na Figura 18 ainda se pode notar a presença de uma grande densidade de potência espectral concentrada dentro do espectro de frequências de interesse a serem atenuadas.

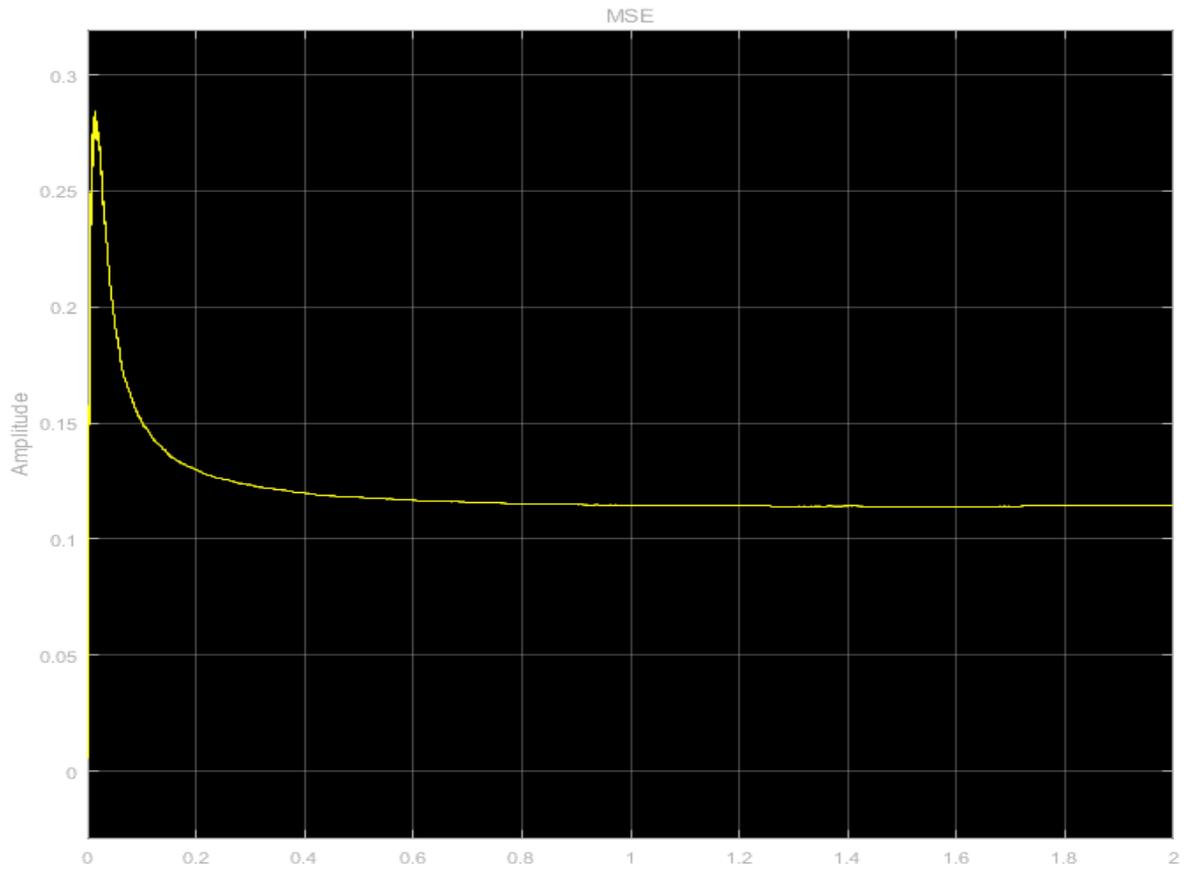


Figura 19: MSE do algoritmo FxNLMSVS.

Da Figura 19, verifica-se o ganho na velocidade de convergência do algoritmo quando comparado a seus concorrentes de passo de adaptação fixo, ainda que o MSE em estado estacionário permaneça praticamente o mesmo.

Para o teste do algoritmo FxNLMSVSVG proposto, utilizou-se 32 amostras de *lag* para estimar o passo de adaptação. Além disso, foram empregados $\alpha = 0.99$, $\beta = 0.9999$ e $\gamma = 10^{-4}$. Estes parâmetros foram escolhidos baseando-se na robustez do algoritmo contra oscilações durante mudanças na faixa de ganho e desempenho em estado estacionário.

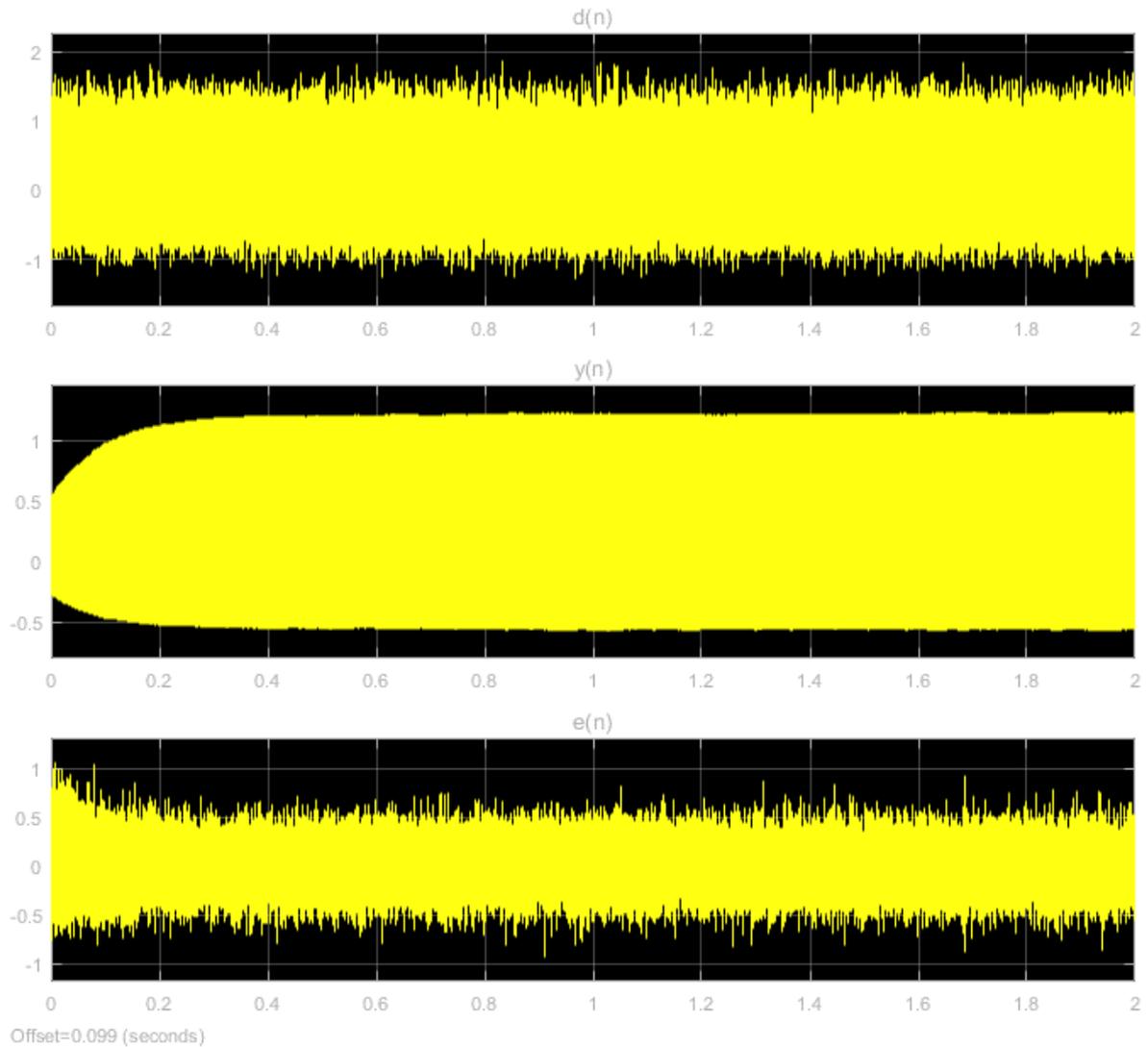


Figura 20: Resultados da simulação do algoritmo FxNLMSVSAG. De cima para baixo tem-se o ruído primário contaminado com ruído branco aditivo, o anti-ruído gerado pelo filtro e o ruído residual após a superposição das ondas sonoras.

Verifica-se o desempenho elevado quando comparado o algoritmo proposto aos algoritmos mais simples. O nível do ruído atenuação obtido foi estimado em 40 dB após dois segundos de simulação, superando os 6 dB alcançados pelos algoritmos sem controle de ganho em aproximadamente 0.1 segundos. A velocidade de convergência deste algoritmo pode ser aumentada tornando a rampa de subida do ganho mais inclinada. O valor do ganho do sinal de referência nesta simulação foi estipulado em 500 e com limite de taxa de subida e descida de $\pm 100\text{ unidade/segundo}$. Este limite de subida e descida favorece a estabilidade do filtro em troca de velocidade de convergência, evitando que variações bruscas na frequência do motor causem sinais de erro demasiadamente grandes que podem causar a divergência do algoritmo durante o transiente.

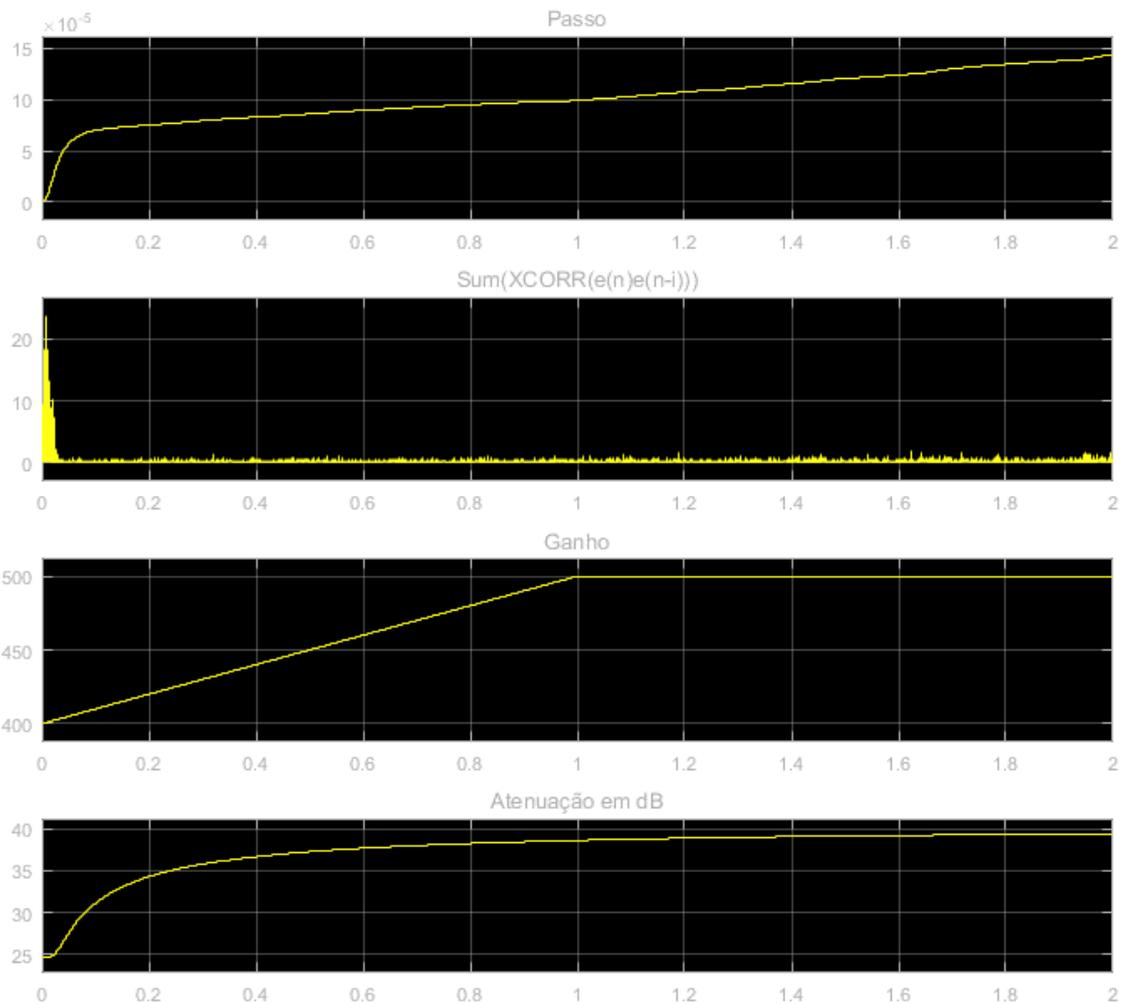


Figura 21: Resultados da simulação do algoritmo FxNLMSVSAG. De cima para baixo tem-se o passo de adaptação, a soma das correlações cruzadas quadráticas do sinal de erro com seus lags, o ganho no feedback do algoritmo e a amplitude do ruído residual em dB.

Verifica-se a diminuição rápida no sinal da soma das correlações do ruído residual com seus sinais de *lag*, demonstrando a capacidade do algoritmo proposto em aproximar a forma de onda e fase corretas durante o ciclo de operação, visto que este diminui a presença de sinal correlacionado na zona silenciosa. Observa-se ainda que o valor do ganho do sinal de realimentação é um fator crucial no desempenho do algoritmo. Dado ganho muito baixo, o algoritmo converge com baixo desempenho de atenuação, casando a fase do sinal porém obtendo um sinal de saída com amplitude mais baixa do que a do sinal de entrada, do contrário, para valores elevados de ganho, o algoritmo se torna instável, dado que o valor do gradiente calculado vezes o passo de adaptação se torna muito grande, prejudicando a capacidade do algoritmo de ajustar os pesos do filtro de maneira a diminuir o erro quadrático médio.

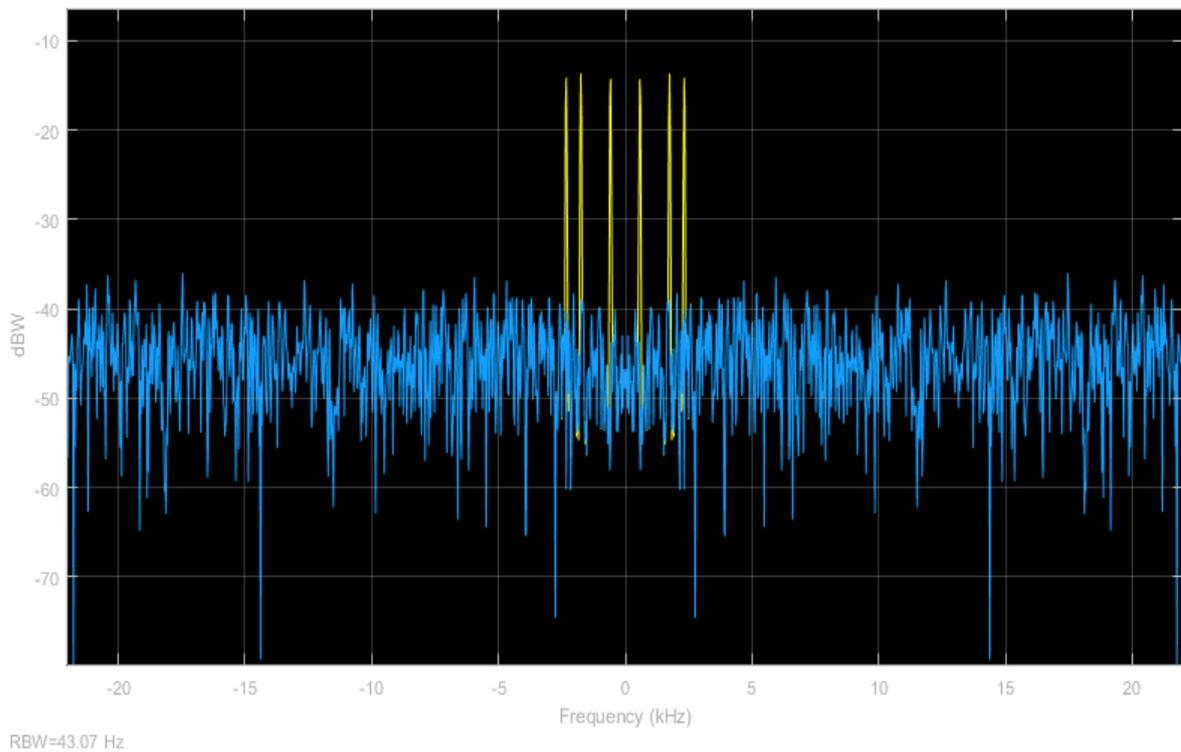


Figura 22: Espectrograma do sinal de ruído residual na zona silenciosa para simulação com o algoritmo FxNLMSVSAS.

Pode-se observar na Figura 22, no entanto, que a totalidade da distribuição de potência do sinal residual encontra-se espalhada por toda a faixa de frequências, sem grandes concentrações de potência nas faixas de frequência de interesse. Esta distribuição de potência é característica do ruído branco aditivo. Claramente todo o sinal de ruído dentro das frequências alvo foi atenuado até o nível do ruído de fundo. De fato, fazendo-se o mesmo teste no algoritmo FxNLMSVSVG para um ruído primário não contaminado, pode-se notar o desempenho superior deste algoritmo em relação ao tempo de convergência e ao nível de atenuação do ruído alvo. As Figuras Figura 23 e Figura 24 apresentam os resultados do algoritmo proposto nos testes supracitados.

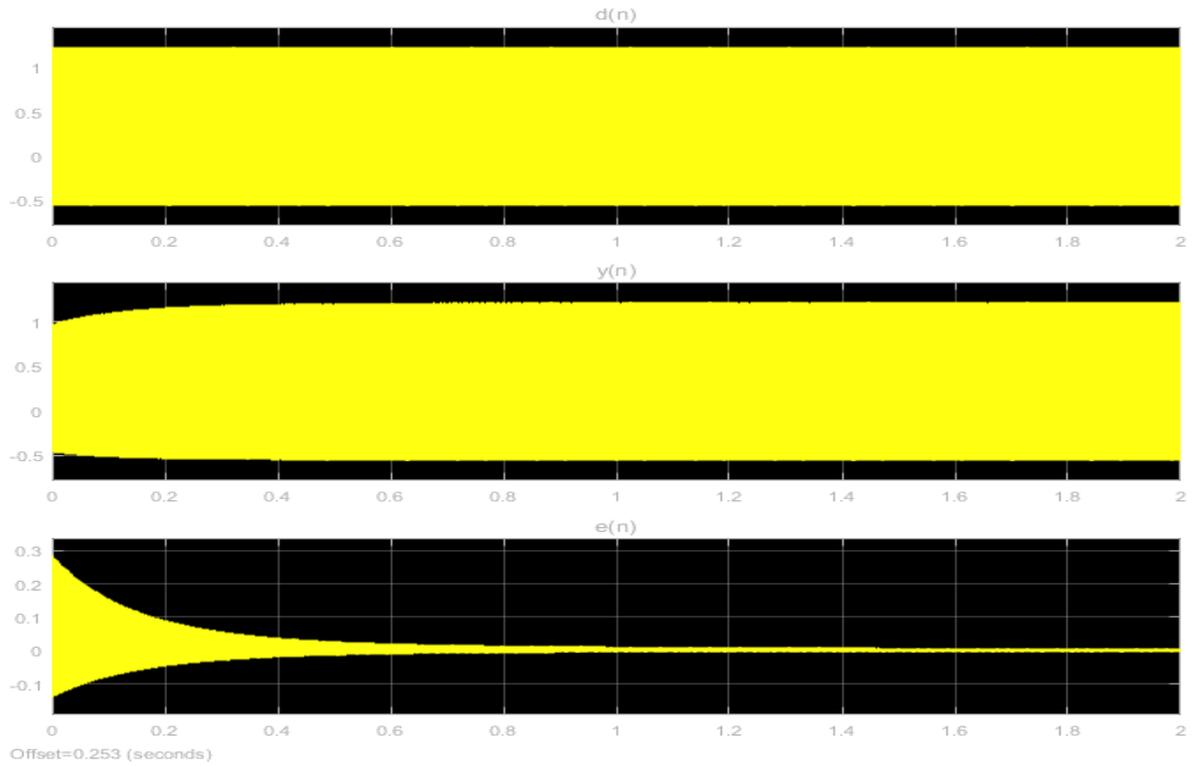


Figura 23: Resultados da simulação do algoritmo FxNLMSVSAG. De cima para baixo tem-se o ruído primário, o antiruído gerado pelo filtro e o ruído residual após a superposição das ondas sonoras.

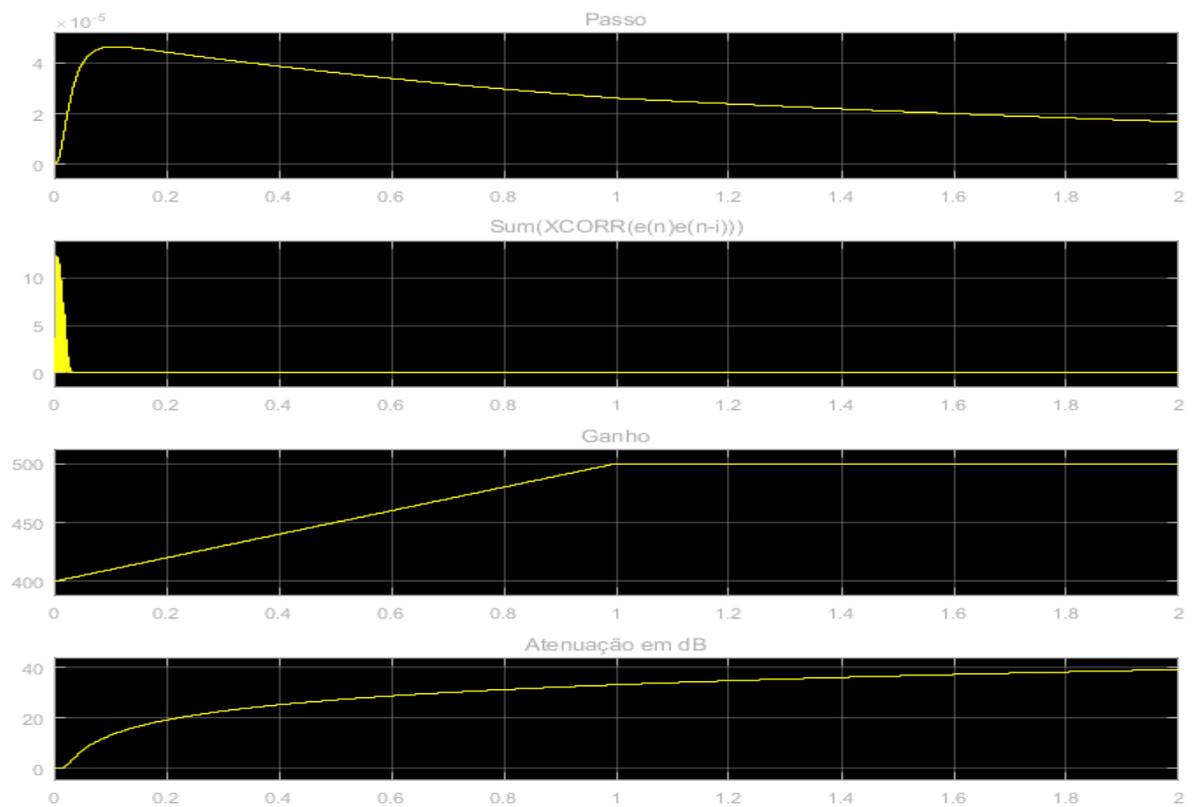


Figura 24: Resultados da simulação do algoritmo FxNLMSVSAG. De cima para baixo tem-se o passo de adaptação, a soma das correlações cruzadas quadráticas do sinal de erro com seus *lags*, o ganho no feedback do algoritmo e a amplitude do ruído residual em dB.

Verifica-se na Figura 23 o desempenho superior do algoritmo proposto para sinais com autocorrelação não nula. Este teste foi feito utilizando-se ainda hiperparâmetros conservadores em relação a robustez do algoritmo, apesar da utilização de apenas ruídos correlacionados. Para parâmetros mais agressivos a velocidade de convergência do algoritmo aumenta, porém a custo de estabilidade.

Em estado estacionário esta formulação obtém aproximadamente $45dB$ de atenuação no ruído correlacionado, valor satisfatório para os requisitos do projeto.

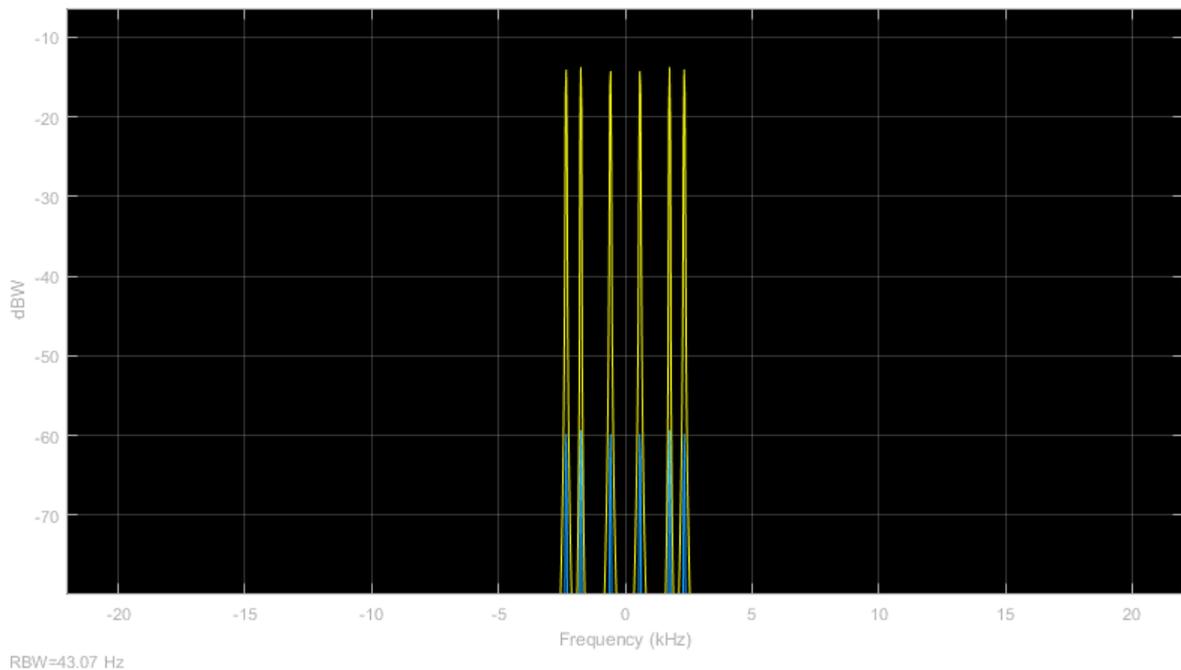


Figura 25: Espectrograma do sinal de ruído residual na zona silenciosa para simulação com o algoritmo FxNLMSVSAS sem a presença do ruído branco contaminante.

É importante observar que para valores de ganho mais elevados, há um aumento considerável de desempenho no nível de atenuação do algoritmo FxNLMSVSAG, porém a custo de robustez a transientes de níveis elevados durante a mudança o tempo de acomodação do ganho. Este efeito pode ser mitigado ajustando-se o valor de τ de forma a controlar o valor final do passo de adaptação, afim de não permitir que a potência de entrada elevada do sinal de referência façam os pesos do algoritmo adquiriram valores muito elevados.

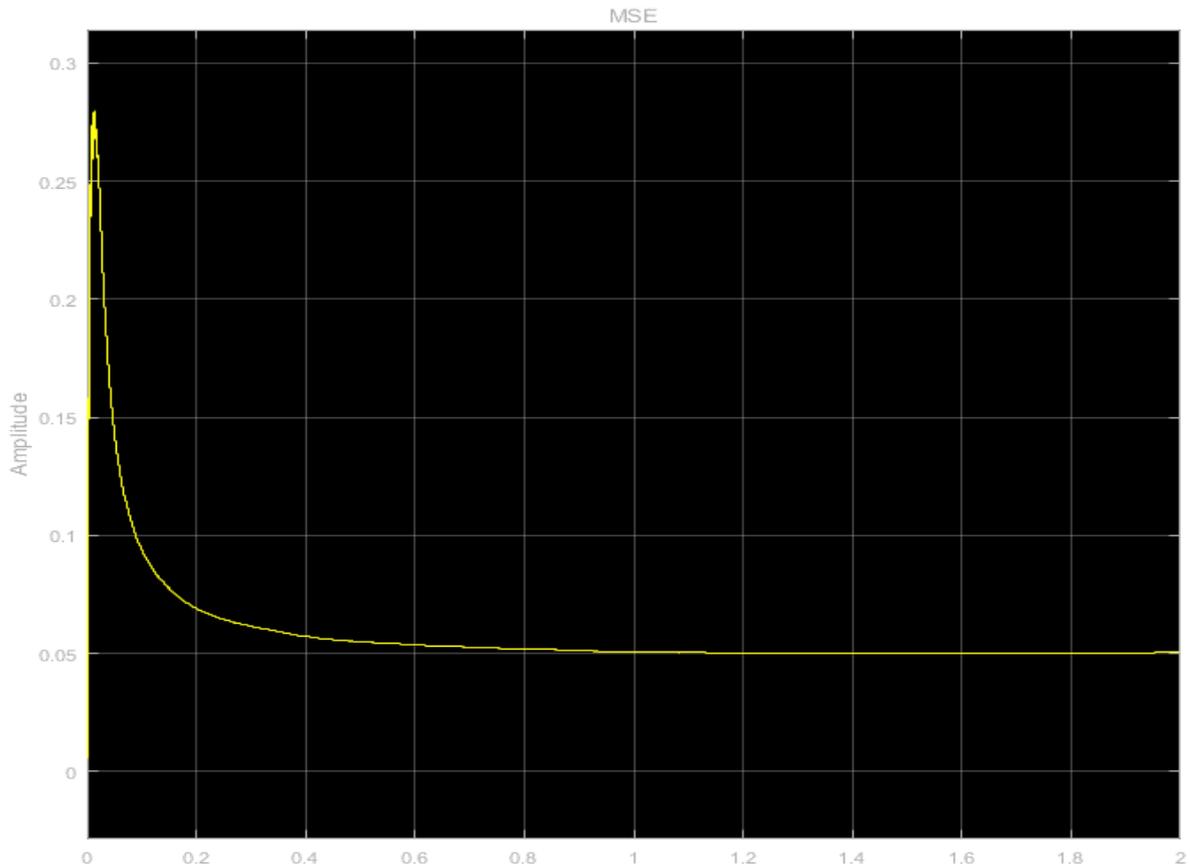


Figura 26: MSE do algoritmo FxNLMSVSAG.

A evolução do MSE pelo tempo do algoritmo FxNLMSVSAG confirma o desempenho superior do algoritmo após a aplicação do ganho de realimentação.

Dado o desempenho satisfatório do algoritmo proposto, considera-se viável sua aplicação, baseando-se puramente em seu desempenho em atenuação de ruído, para o caso estudado em dispositivo redutor ativo de ruído acústico.

O próximo passo para determinar a viabilidade dos algoritmos e auxiliar na escolha do algoritmo final a ser aplicado no dispositivo é realizar a comparação de desempenho comparada a custo computacional.

A Tabela 2 exhibe a comparação dos resultados de desempenho *versus* custo computacional dos algoritmos analisados.

Tabela 2: Resultados comparados das simulações versus custo computacional dos algoritmos testados.

Algoritmo	Hiper- Parametros	Nível de Atenuação	Velocidade de Convergência	Custo computacional
FxLMS/ FxNLMS	$\mu = 10^{-3}$ $N = 128$ $L = 128$	15dB	1 <i>segundo</i>	385 <i>operações</i> <i>/ciclo</i> 514 <i>operações</i> <i>/ciclo</i>
FxNLMSVS	$N = 128$ $L = 128$ $M = 32$ $\alpha = 0.99$ $\beta = 0.99$ $\gamma = 10^{-4}$	15dB	0.1 <i>segundo</i>	582 <i>operações</i> <i>/ciclo</i>
FxNLMSVSVG	$N = 128$ $L = 128$ $M = 32$ $\alpha = 0.99$ $\beta = 0.9999$ $\gamma = 10^{-4}$	45dB	2 <i>segundos</i>	584 <i>operações</i> <i>/ciclo</i>

Verifica-se que o algoritmo FxNLMSVSVG é viável para aplicação na redução ativa de ruído acústico por seu elevado desempenho em atenuação de ruído em uma ampla faixa de frequências, obtendo desempenho 600% maior do que seus concorrentes mais simples, grande robustez mesmo em ambientes com baixa razão sinal/ruído, e elevação da complexidade computacional aceitável quando comparado a algoritmos com desempenho muito menor.

5 Conclusão

No desenvolver deste trabalho foram obtidos conhecimentos de grande valia na área da filtragem adaptativa, inclusive a obtenção de conceitos importantes para a implementação do projeto em trabalhos futuros. Constatou-se que a redução ativa de ruído é viável dada uma faixa de frequências, e pode-se estudar modelos variados para serem aplicados na resolução do problema em questão. Neste sentido, pode-se verificar a viabilidade da aplicação de filtragem adaptativa ao problema de redução ativa de ruído acústico.

Notou-se que a compensação do caminho secundário na entrada do estimador LMS que adapta os parâmetros do filtro de controle é de crucial importância para a convergência do algoritmo, para que este possa assim obter desempenho satisfatório.

Pode-se constatar também que nem sempre com o aumento do número de pesos do filtro de controle é obtido aumento real no desempenho, e que a utilização de valores muito elevados de números de parâmetros leva a uma sobreparametrização com resultados maléficos sobre o desempenho do filtro.

Foi realizado ainda um estudo sobre a complexidade computacional e constatou-se a não linearidade da relação entre desempenho e custo computacional. Por esta razão, a escolha correta do algoritmo a ser empregado é fator crucial na viabilização do projeto, dado que processadores mais velozes possuem elevado valor financeiro e conseqüente custo de aquisição.

Durante a realização deste trabalho, pode-se observar ainda que a potência de entrada do sinal de referência é um fator crucial para a convergência dos algoritmos, devendo-se destinar atenção elevada a este ponto, realizando-se o tratamento do sinal antes que este chegue a entrada do algoritmo adaptativo.

Trabalhos futuros envolvem o aprimoramento e implementação do algoritmo escolhido em processador embarcado para prototipagem, desenvolvimento de uma interface homem-máquina para modificar e controlar os hiperparâmetros do controlador *on-line*, viabilização da produção do produto em escala e treinamento de pessoal qualificado para realizar a montagem do mesmo nas cabines de aviões ultraleves avançados.

Referências Bibliográficas

- [1] N. V. George e G. Panda, “Advances in active noise control: A survey, with emphasis on recent nonlinear techniques”, *Signal Processing*, vol. 93, n° 2, p. 363–377, 2013.
- [2] S. M. Kuo, I. Panahi, K. M. Chung, T. Horner, M. Nadaski, e J. Chyan, “Design of active noise control systems with the TMS320 family”, 1996.
- [3] B. Haykin, S. and Van Veen, *Signals and Systems*, 2nd ed. New York, 2002.
- [4] A. V Oppenheim, A. S. Willsky, e S. H. Nawab, *Signals and Systems*, vol. 2nd ed. 1997.
- [5] B. S. Kamen, E. W. and Heck, *Fundamentals of Signals and Systems*, 2nd ed. Prentice Hall, 1999.
- [6] A. V Oppenheim, R. W. Schafer, e J. R. Buck, *Discrete Time Signal Processing*, vol. 1999. 1999.
- [7] S. Haykin, *Adaptive Filter Theory*. Prentice-Hall.
- [8] B. Farhang-Boroujerny, *Adaptive Filter Theory*. John Wiley and Sons.
- [9] B. W. and S. D. Stearns, *Adaptive Signal Processing*. Prentice-Hall.
- [10] D. R. Morgan, “An Analysis Of Multiple Correlations Cancellation Loops With A Filter In The Auxiliary Path”, *IEEE Trans. Acoust.*, vol. 28, n° 4, p. 454–467, 1980.
- [11] J. C. Burgess, “Active adaptive sound control in a duct: A computer simulation.”, *J. Acoust. Soc. Am.*, vol. 70, n° 3, p. 715–726, 1981.
- [12] L. J. Eriksson, “Development of the Filtered-U Algorithm for Active Noise Control”, *J. Acoust. Soc. Am.*, vol. Vol. 89, n° No. 1, p. 257–265.
- [13] F. Y. Farhan, “Improved Hybrid Variable and Fixed Step Size Least Mean Square Adaptive Filter Algorithm with Application to Time Varying System Identification”, vol. 0, n° 2, p. 94–98, 2015.
- [14] J. G. F. Zipf e O. J. Tobias, “Algoritmo LMS de Passo Variável para Ambientes com Razão Sinal-Ruído Baixa”, n° 1, p. 2–5, 2008.
- [15] M. A. G. De Carvalho, “ST067 - Tópicos Especiais em Informática”, 2004.
- [16] A. A. da Rocha, *Análise da Complexidade de Algoritmos*. FCA, 2014.
- [17] D. David, “Alise de Programas”, *DIMAp/UFRN*, p. 1–10, 2003.
- [18] T. D. E. Laplace e S. H. Cardoso, “Modelagem da Propagação Sonora em Dutos: Abordagem Matemática no Domínio da Frequência e do Tempo Utilizando Transformada de Laplace”, *Univ. Brasília Fac. unB gama/ Fac. Tecnol. Programa pós-graduação em integridade Mater. da Eng.*, 2016.
- [19] I. Jorge e C. Nuñez, “MODELO DE DUTOS ACÚSTICOS TENDO EM VISTA O PROJETO DE CONTROLADORES ATIVOS DE RUÍDO”, 1999.
- [20] V. K. Madisetti, Org., *Digital Signal Processing Fundamentals*, 2° ed. CRC Press, Taylor & Francis Group.
- [21] D. M. Allen, “Mean Square Error of Prediction as a Criterion for Selecting Variables”, *Technometrics*, vol. 13, n° 3, p. 469–475, 1971.
- [22] S. C. Douglas e M. Rupp, “Convergence Issues in the LMS Adaptive Filter”, *Digit. Signal Process. Fundam.*, p. 19-1-19–21, 2010.
- [23] H. Akaike, “A new look at the statistical model identification”, *IEEE Trans. Autom. Control*, vol. 19, n° 6, p. 716–723, 1974.
- [24] J. Rissanen, “Modelling by the shortest data description”, *Automatica*, vol. 14, p. 465–471, 1978.

- [25] ROTAX, “Operator’s Manual”, n° June, 2011.