Algorithms and implementations to overcome practical issues in active noise control systems

Shi, Dongyuan

2020

Shi, D. (2020). Algorithms and implementations to overcome practical issues in active noise control systems. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/137095>

<https://doi.org/10.32657/10356/137095>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 10:28:49 SGT

Algorithms and implementations to overcome practical issues in active noise control systems

SHI DONGYUAN

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

1.Aug. 2019

.....

Date

Shi Dongyuan

.....

SHI DONGYUAN

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

1. Aug. 2019

.....

Date

.....

Prof. GAN WOON SENG

Authorship Attribution Statement

This thesis contains material from 9 paper(s) published and 3 paper(s) under review in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Chapter 3 is published as Shi, DongYuan, Woon-Seng Gan, Bhan Lam, and Chuang Shi. “Two-gradient direction FXLMS: An adaptive active noise control algorithm with output constraint.” *Mechanical Systems and Signal Processing* 116 (2019): 651-667.

The contributions of the co-authors are as follows:

- Prof Gan provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Prof Gan, Prof Shi, and Dr. Lam.
- I co-designed the study with Prof Gan and performed all the laboratory work at the School of Electrical and Electronic Engineering. I also analyzed the data.
- Prof Shi assisted in the interpretation of the experiments.
- Dr. Lam gave the suggestions about paper’s modification and the numerical simulation.

Chapter 3 is published as Shi, Dongyuan, Bhan Lam, Woon-Seng Gan, and Shulin Wen. “Optimal Leak Factor Selection for the Output-Constrained Leaky Filtered-Input Least Mean Square Algorithm.” *IEEE Signal Processing Letters* 26, no. 5 (2019): 670-674.

The contributions of the co-authors are as follows:

- Prof Gan provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Prof Gan, Dr. Lam, and Dr. Wen.
- I conducted the literature survey and drew all the graphics.
- I designed the study and performed all the laboratory work at the school of electrical and electronic engineering. I also analyzed the data.
- Dr. Lam gave the suggestions about paper’s modification and the numerical simulation.
- Dr. Wen assisted in the experiments.

Chapter 3 is published as Shi, Dongyuan, Chuang Shi, and Woon-Seng Gan. “Effect of the audio amplifier’s distortion on feedforward active noise control.” In 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 469-473. IEEE, 2017.

The contributions of the co-authors are as follows:

- Prof Gan provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Prof Gan and Prof Shi.
- I conducted the literature survey and drew all the graphics.
- I designed the study and performed all the laboratory work at the school of electrical and electronic engineering. I also analyzed the data.
- Prof Shi assisted in the interpretation of the experiments.

Chapter 3 is published as Shi, Dongyuan, Bhan Lam, Woon-seng Gan and Shulin Wen. “Practical Consideration and Implementation for Avoiding Saturation of Large Amplitude Active Noise Control.” In 23th Intl. Congress on Acoustics, Aachen, Germany, 2019.

The contributions of the co-authors are as follows:

- Prof Gan provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Prof Gan, Dr. Lam, and Dr. Wen.
- I conducted the literature survey and drew all the graphics.
- I designed the study and performed all the laboratory work at the school of electrical and electronic engineering. I also analyzed the data.
- Dr. Lam gave the suggestions about paper’s modification and the numerical simulation.
- Dr. Wen assisted in the experiments and simulations.

Chapter 4 is published as Shi, Dongyuan, Bhan Lam, and Woon-seng Gan. “Analysis of Multichannel Virtual Sensing Active Noise Control to Overcome Spatial Correlation and Causality Constraints.” In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8499-8503. IEEE, 2019.

and Shi Dongyuan, Woon-seng Gan, and Bhan Lam. “Feedforward Multichannel Virtual-sensing Active Noise Control Technique: Analysis on causal and sensor-actuator constraints.” The Journal of the Acoustical Society of America, (Under review).

The contributions of the co-authors are as follows:

- Prof Gan provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Prof Gan and Dr. Lam.
- I conducted the literature survey and drew all the graphics.
- I designed the study and performed all the laboratory work at the school of electrical and electronic engineering. I also analyzed the data.
- Dr. Lam gave the suggestions about paper's modification and the numerical simulation.

Chapter 5 is published as Shi, Dong Yuan, Bhan Lam, and Woon-Seng Gan. "A novel selective active noise control algorithm to overcome practical implementation issue." In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1130-1134. IEEE, 2018.

and Shi Dongyuan, Woon-seng Gan, Bhan Lam, and Wen Shulin. "Feedforward Selective Active Noise Control: Algorithm and Implementation." IEEE Transactions on audio, speech, and language processing.

The contributions of the co-authors are as follows:

- Prof Gan provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Prof Gan, Dr. Lam, and Dr. Wen.
- I conducted the literature survey and drew all the graphics.
- I designed the study and performed all the laboratory work at the school of electrical and electronic engineering. I also analyzed the data.
- Dr. Lam gave the suggestions about paper's modification and the numerical simulation.
- Dr. Wen assisted in the experiments and simulations.

Chapter 6 is published as Shi, Dong Yuan, and Woon-Seng Gan. "Comparison of different development kits and its suitability in signal processing education." In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6280-6284. IEEE, 2016.

The contributions of the co-authors are as follows:

- Prof Gan provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts.
- I conducted the literature survey and drew all the graphics.

Chapter 6 is published as Shi, Dongyuan, Chuang Shi, and Woon-Seng Gan. "A systolic FxLMS structure for implementation of feedforward active noise control

on FPGA.” In 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), pp. 1-6. IEEE, 2016.

The contributions of the co-authors are as follows:

- Prof Gan provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Prof Gan and Prof Shi.
- I conducted the literature survey and drew all the graphics.
- I designed the study and performed all the laboratory work at the school of electrical and electronic engineering. I also analyzed the data.
- Prof Shi assisted in the interpretation of the experiments.

Chapter 6 is published as Shi, Dongyuan, Jianjun He, Chuang Shi, Tatsuya Murao, and Woon-Seng Gan. “Multiple parallel branch with folding architecture for multichannel filtered-x least mean square algorithm.” In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1188-1192. IEEE, 2017.

and Shi, Dongyuan, Woon-Seng Gan, Jianjun He, and Bhan Lam. “Practical Implementation of Multichannel Filtered-x Least Mean Square Algorithm based on the Multiple-Parallel-Branch with Folding Architecture for Large-Scale Active Noise Control.” IEEE Transactions on Very Large Scale Integration (VLSI) Systems.

The contributions of the co-authors are as follows:

- Prof Gan provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Prof Gan, Dr. He, Prof Shi, and Prof Murao.
- I conducted the literature survey and drew all the graphics.
- I designed the study and performed all the laboratory work at the school of electrical and electronic engineering. I also analyzed the data.
- Prof Shi assisted in the interpretation of the experiments.
- Dr. He gave the suggestions about paper’s modification and the numerical simulation.
- Dr. Lam helped to build the experimental setup.
- Prof Murao helped to carry out the real-time experiments and measurements.

1. Aug. 2019

.....

Date

Shi Dongyuan

.....

SHI DONGYUAN

Acknowledgements

I wish to express my most enormous gratitude to my supervisor, Professor Gan Woon Seng, for his patient and support in overcoming numerous obstacles throughout my Ph.D. His enthusiasm has cultivated a vibrant and conducive research environment, which provides us plenty of opportunities to advance my research.

Numerous people have helped me throughout the pursuit of my Ph.D., and I would like to recognize their contributions and thanks to them as follows:

- Professor Shi Chuang for his guidance and assistance in the conduct of academic research and technical writing.
- Dr. Lam Bhan for his help in experiments and technical writing.
- Dr. He Jianjun for his assistance in the initial setup of the test chamber and the experiment.
- Professor Tatsuya Murao for his effort in establishing the experiment setups and the test chamber.
- All members and alumni of the lab that have been a significant part of Ph.D. life, in no particular order- Mu Hao (Dr), Yi yang, Valiastin, Wen Shulin (Dr), Rina, Santi, Duy Hai, Xiaoyi, Rishabh, and Zhen Ting.

Last but not least, I would like to appreciate my wife, Li Yanlin, and my family for their encouragement and support throughout.

“If I had one hour to save the world, I would spend 55 minutes defining the problem and only five minutes finding the solution.”

—Einstein, Albert

Contents

Acknowledgements	xiii
Summary	xxiii
List of Figures	xxiv
List of Tables	xxxi
Symbols and Acronyms	xxxiii
1 Introduction	1
1.1 Urban noise	1
1.2 Traditional noise control approaches	2
1.3 Introduction to active noise control	4
1.3.1 Feedforward structure	6
1.3.2 Feedback structure	7
1.3.3 Multichannel structure	7
1.4 Objective	9
1.5 Major contributions of this thesis	10
1.6 Outline of the thesis	11
2 Adaptive algorithms used in feedforward active noise control	13
2.1 Principle of the adaptive filter	13
2.2 Least mean square (LMS) algorithm	16
2.3 Filtered-x least mean square (FxLMS) algorithm	16
2.3.1 Single-channel FxLMS algorithm	16
2.3.2 Multichannel FxLMS algorithm	18
2.4 Modified active noise control algorithm	21
2.4.1 Normalized FxLMS algorithm	21
2.4.2 Sign FxLMS algorithm	22
2.4.3 Moment FxLMS algorithm	23
2.4.4 Frequency domain adaptive filter used in ANC	24
2.4.5 Filtered-x recursive least square (FxRLS) algorithm	25
2.5 Conclusion	28

3	Effect of output nonlinearity on active noise control	29
3.1	Influence of output distortions in active noise control	29
3.1.1	Amplitude distortion	30
3.1.2	Frequency distortion	31
3.1.3	Phase distortion	32
3.2	Effect of the audio amplifier distortion on feedforward ANC	32
3.2.1	Influence of the amplitude distortion on ANC performance	32
3.2.1.1	Case A: Moderate overdriving	33
3.2.1.2	Case B: Severe Overdriving	36
3.2.1.3	Simulation: Effect of amplitude distortion	37
3.2.2	Influence of the phase distortion on ANC performance	39
3.2.2.1	Simulation: Effect of phase distortion	40
3.3	Modified adaptive filter algorithms to overcome the effect of output saturation	42
3.3.1	Nonlinear filtered-x LMS (NLFxLMS) algorithm	43
3.3.2	Leaky FxLMS algorithm	44
3.3.3	Rescaling algorithm	46
3.4	Optimal control filter for output constraint	47
3.4.1	The objective function under output constraint	47
3.4.2	The optimal solution of the adaptive algorithm with output constraint	47
3.4.3	Minimum mean square error (MMSE) of the adaptive algorithm with output constraint	50
3.5	Two-gradient direction FxLMS (2GD-FxLMS) algorithm	52
3.5.1	Simulation: Performance of the 2GD-FxLMS algorithm	54
3.5.1.1	Control of simulated tonal noise	54
3.5.1.2	Control of simulated random noise	57
3.5.1.3	Tonal control in a single-channel practical active control system	60
3.5.2	Implementation of 2GD-FxLMS algorithm	64
3.5.2.1	An analysis on the computational complexity of the algorithm	64
3.5.2.2	Experiment results of 2GD-FxLMS	65
3.6	Optimal leak factor selection for Leaky FxLMS algorithm	67
3.6.1	Derivation of the optimal leaky factor	69
3.6.2	Simulation: LFxLMS algorithm with the optimal leak factor	73
3.7	Conclusions	76
4	Virtual sensing feedforward active noise control	79
4.1	Virtual sensing approaches	80
4.1.1	Class I: Virtual signal prediction based on an acoustic model	80
4.1.2	Class II: Preliminary training method	81
4.2	Single-channel virtual sensing ANC	83
4.3	Multichannel virtual sensing ANC	86

4.4	Analysis on multichannel virtual sensing ANC	90
4.4.1	Frequency domain analysis of the MVANC technique	90
4.4.1.1	Tuning stage	91
4.4.1.2	Control stage	96
4.4.2	Time domain analysis on the MVANC technique	98
4.5	Effect of alternative sensor-actuator configurations	100
4.5.1	Case i: Over- or fully-determined tuning and control stages.	101
4.5.2	Case ii: Under-determined tuning stage and an over-determined control stage.	102
4.5.3	Case iii: Over-determined tuning stage and an under-determined control stage.	103
4.5.4	Case iv: Under-determined tuning and control stage.	103
4.6	Simulation: MVANC with measured transfer functions	104
4.7	Experiments of the 4-channel MVANC	109
4.7.1	Experiment 1: four secondary sources, four physical microphones, and four virtual microphones.	110
4.7.2	Experiment 2: Swapping physical microphones and virtual microphones.	112
4.7.3	Experiment 3: Three secondary sources, four physical microphones, and four virtual microphones.	115
4.7.4	Experiment 4; three secondary sources, three physical microphones, and four virtual microphones.	115
4.7.5	Experiment 5: three secondary sources, two physical microphones, and four virtual microphones.	118
4.7.6	Experiment 6: three secondary sources, four physical microphones, and two virtual microphones.	120
4.8	Conclusions	123
5	Selective fixed-filter feedforward active noise control technique	125
5.1	Selective active noise control approach	126
5.1.1	Offline pre-training stage of control filters	127
5.1.2	Online control: Feature extraction and control filter selection	127
5.2	Frequency-Band-Match algorithm for SFANC	128
5.3	Proposed selective fixed-filter active noise control algorithm	132
5.3.1	Pre-training stage of the control filters	132
5.3.2	Selective active noise control algorithm	138
5.4	Computational complexity analysis	140
5.5	Numerical simulation results	142
5.5.1	Simulation A: Noise reduction as a function of similarity	143
5.5.2	Simulation B: Active control of stationary broadband noise	144
5.5.3	Simulation C: Active control of time-varying broadband noise	146
5.5.4	Simulation D: Active control of recorded real-world noise	146
5.5.5	Simulation E: The effect of FFT accumulation over L iterations on the noise reduction performance	148

5.6	Conclusions	149
6	The efficient implementation architectures of the feedforward active noise control on FPGA	151
6.1	Background: Programmable logic device	152
6.2	Optimization of digital signal processing structure on programmable logic devices	152
6.2.1	Latency and Throughput	153
6.2.2	Critical path	153
6.2.3	Pipelining technique	154
6.2.4	Graph transposition approach	155
6.2.5	Parallel technique	156
6.2.6	Folding technique	156
6.2.7	Systolic structure	159
6.3	Systolic FxLMS algorithm for feedforward ANC implementation	161
6.3.1	Derivation of systolic FxLMS Structure	163
6.3.2	Theoretical analysis on the systolic FxLMS algorithm	166
6.3.3	Implementation of systolic FxLMS algorithm	168
6.3.4	Simulation: systolic FxLMS algorithm	169
6.3.5	Implementation result of systolic FxLMS algorithm	171
6.4	Practical implementation of feedforward multichannel active noise control	173
6.4.1	Multiple-Parallel-Branch with folding Architecture	176
6.4.1.1	Parallel Branch	176
6.4.1.2	Recursive formula derivation	176
6.4.1.3	Folded structure derivation	179
6.4.2	Implementation of MPBF architecture on FPGA.	182
6.4.2.1	Finite-state machine to control the switches	183
6.4.2.2	Data communication between iterations	185
6.4.2.3	Clock divider	186
6.4.3	Implementation results	188
6.4.4	Real-time active control of noise through the open window	189
6.4.5	Real-time experimental result	191
6.5	Conclusions	196
7	Conclusions and future work	199
7.1	Conclusions	199
7.2	Future works	201
A	Proofs of part I	203
A.1	Optimal solution of extended leaky FxLMS algorithm	203
B	Proofs for Part II	205
B.1	Optimal control filter out the frequency band	205

B.2	Power Spectral Density of the Residual Error	206
B.3	Filtered reference com-partitioning frequency domain adaptive filter	208
B.4	Binary distance or similarity ratio	210
C	Proofs of part III	211
C.1	Root locus analysis of systolic FxLMS algorithm	211
	List of Author's Awards, Patents, and Publications	215
	Bibliography	221

Summary

Noise control has garnered increasing attention, in the face of growing urbanization. To deal with the noise issue, the active noise control (ANC) technique is revisited in the thesis. The ANC technique utilizes a sensor to acquire the noise information and synthesizes the anti-noise by driving the actuator to cancel the noise. Compared to the traditional passive noise control approach, it shows the apparent advantages of the cost, size, and easy deployment. Hence, ANC is widely used in active control headphone and in automotive interiors. However, the ANC technique still possesses many practical challenges, such as the nonlinearity of the secondary path, the placement of the error sensor, and the soaring computational requirements with the increasing scale of multichannel active noise control (MCANC) systems. To effectively solve these practical issues, the thesis firstly reviews existing optimization techniques and subsequently proposes unique algorithms and methods.

The distortions caused by the saturation of the audio amplifier play the central role in the nonlinearity of ANC. These distortions seriously influence noise reduction performance and often result in the divergence of the adaptive algorithm. One of the most effective ways is to constrain the power of the output signal to ensure that the audio amplifier operates linearly and avoids saturation. To guarantee optimal noise reduction performance of the ANC system under output constraint, a two-gradient direction FxLMS algorithm is proposed and validates. The algorithm automatically switches the gradient direction to reduce the power gain of the control filter when the ANC system exceeds the output-power constraint. An alternative method is to utilize the leaky FxLMS algorithm adaptively restraining the magnitude of the control filter. With the proposed optimal leak factor suggested in the thesis, the leaky FxLMS algorithm gains optimal control under output constraint. The numerical simulations and the experiments verify the performance of both algorithms.

In some ANC applications, it is impractical to place the error sensor in the desired location. To overcome this issue, the virtual sensing ANC technique that is capable of projecting the quiet zone to the desired position from the physical microphone is reexamined. In the thesis, the virtual sensing technique is extended from single-channel ANC to the multichannel ANC. The multichannel virtual sensing ANC (MVANC) is shown to achieve the same noise attenuation performance as MCANC, which places the real error sensors at the desired positions. Furthermore, the physical limitation of the MVANC approach is determined by a sensor-actuator analysis and experimentally validated in a small chamber.

To alleviate the computational complexity of FxLMS-based approaches, selective fixed-filter active noise control (SFANC) approach is proposed in the thesis that obtains a series of filters and stores into a database in the preliminary stage. When dealing with a specific type of primary noise, the SFANC algorithm selects a suitable pre-trained filter from the database based on the frequency-band-match method. The simulations show that SFANC obtains satisfactory noise reduction performance in different applications. Moreover, since SFANC does not contain the adaptive update process, it saves a considerable amount of computations and achieves higher robustness.

Finally, the thesis explores a series of approaches, which can optimize the structure of the digital processing algorithm implemented on the FPGA platform. Firstly, the systolic FxLMS, which significantly increase the sampling rate and throughput rate of the ANC system, is proposed in the thesis. For realizing a large channel count MCANC on FPGA, a multiple-parallel-branch with folding structure is proposed. This architecture is used to implement a 24-channel MCANC and successfully balances the usage of the computation resource and the maximum operating speed. This 24-channel MCANC is deployed in an open window whereby the noise reduction performance sufficiently validates the effectiveness of the proposed architecture.

List of Figures

1.1	(a) Destructive interference (b) Constructive interference.	4
1.2	Active control of plane wave in an infinite duct using single secondary source.	5
1.3	Single-channel feedforward ANC system in an air duct.	6
1.4	Single-channel feedforward ANC without feedback noise.	7
1.5	Single-channel feedback ANC system in an air duct.	8
1.6	Block diagram of the MCANC system.	8
2.1	Block diagram of adaptive filter algorithm.	14
2.2	Block diagram of the transversal filter.	15
2.3	Block diagram of the FxLMS algorithm applied in a single-channel feedforward ANC system. (1) Green color denotes the acoustic field, and (2) Blue color denotes the digital field.	17
2.4	Block diagram of the multichannel FxLMS algorithm.	19
2.5	Block diagram of the frequency-domain adaptive filter used in ANC.	24
3.1	The output magnitude as a function of the input magnitude.	30
3.2	The clipped output signal of the amplifier.	31
3.3	Single-frequency adaptive noise canceller.	33
3.4	The equivalent diagram of the single-frequency adaptive noise canceller for harmonics.	35
3.5	Spectrum of the disturbance and attenuated noise at 500 Hz.	38
3.6	The curves of weight $w_1(n)$ for disturbances with different power.	38
3.7	Spectrum of the disturbance and attenuated noise at the phase error of 108°	41
3.8	The coefficient $w_1(n)$ of the control filter with a phase error of 108°	41
3.9	Block diagram of an ANC system with the saturation effect	43
3.10	Block diagram of ANC with the saturation model.	45
3.11	Mean square error surface of the adaptive algorithm with output constraint when the threshold is (a) less than the power of the disturbance, (b) equals to the power of the disturbance, and (c) larger than the power of the disturbance. The case without constraint is shown in (d). (Note: The gain of the secondary path is assumed to be one.)	51
3.12	The output signal of (a) FxLMS algorithm, (b) leaky FxLMS algorithm, (c) rescaling algorithm, and (d) 2GD-FxLMS algorithm.	55

3.13	The weight convergence path of the (a) FxLMS algorithm, (b) leaky FxLMS, (c) rescaling algorithm, and (d) 2GD-FxLMS algorithm against contours of the mean square error $\mathbb{E}[e^2(n)]$, with the red curve denoting the constraint $\mathbb{E}[\mathbf{w}^T(n)\mathbf{x}(n) ^2] = 0.28125$	56
3.14	The residual error of the FxLMS algorithm, leaky FxLMS algorithm, rescaling algorithm, and 2GD-FxLMS algorithm.	57
3.15	The control filter output with the (a) leaky FxLMS, (b) rescaling, and (c) 2GD-FxLMS algorithms with a constraint of 0.75, as indicated by the dashed line.	58
3.16	The control filter weights of the (a) leaky FxLMS, (b) rescaling, and (c) 2GD-FxLMS algorithm, as a function of the number of iterations; and (d) the distribution ratio of the two gradient directions in 2GD-FxLMS.	59
3.17	The error signal of the (a) leaky FxLMS, (b) rescaling, and (c) 2GD-FxLMS algorithms, as a function of the number of iterations.	60
3.18	The impulse response of the (a) primary path, and (b) secondary path, of a real ANC system, and (c) output value vs. the input value of the amplifier TDA2003.	61
3.19	The output signal of the (a) FxLMS algorithm, (b) leaky FxLMS algorithm, (c) rescaling algorithm, and (d) the 2GD-FxLMS algorithm, all as a function of the number of iterations.	62
3.20	The spectrum of the error signal for a real ANC system, (a) without control, (b) with FxLMS algorithm, (c) with leaky FxLMS algorithm, (d) with the rescaling algorithm, and (e) with the 2GD-FxLMS algorithm, all as a function of normalized frequency. The distribution ratio of two gradient directions in 2GD-FxLMS algorithm is shown in (f).	63
3.21	The detailed block diagram of the 2GD-FxLMS algorithm.	65
3.22	Error signals' power spectrum of FxLMS and 2GD-FxLMS for the tonal noise.	67
3.23	Error signal power spectrum of FxLMS and 2GD-FxLMS as a function of frequency for the broadband noise.	68
3.24	The flow chart of the LFxLMS algorithm with the optimal leak factor.	73
3.25	The convergence paths of the leaky FxLMS algorithm with the optimal leak factor, for different primary paths with a sinusoid as the reference signal.	74
3.26	The convergence paths of the leaky FxLMS algorithm with the optimal leak factor, for different primary paths with white noise as the reference signal.	75
3.27	The spectrum of the error signal and the control signal of different algorithms.	76
4.1	A single-channel ANC duct with the virtual microphone.	80
4.2	Block diagram of the feedforward active control algorithm with the remote microphone	82

4.3	Block diagram of the single-channel virtual sensing ANC algorithm in the tuning stage.	84
4.4	Block diagram of the single-channel virtual sensing ANC algorithm in the control stage.	85
4.5	A simplified schematic of the MVANC algorithm.	87
4.6	Training the control filter in the tuning stage of the MVANC technique.	88
4.7	Training the auxiliary filter in the tuning stage of the MVANC technique.	89
4.8	Training the control filter in the control stage of the MVANC technique.	90
4.9	Block diagram of the tuning stage in the MVANC technique, where $\sum$ denotes acoustic summation and $+$ denotes digital summation.	91
4.10	Block diagram of the control stage in the MVANC technique.	96
4.11	Schematic diagram of the four channel MVANC system with different configurations: (a) $K = 4$ (secondary sources), $M = 4$ (Physical MICs), $N_v = 4$ (Virtual MICs); (b) $K = 4$, $N_v = 4$, $M = 4$; (c) $K = 3$, $M = 4$, $N_v = 4$; (d) $K = 3$, $M = 3$, $N_v = 4$; (e) $K = 3$, $M = 2$, $N_v = 4$; (f) $K = 3$, $M = 4$, $N_v = 2$	104
4.12	Coefficients of the four steady-state control filters in simulation (a).	105
4.13	Noise reduction in dB at the virtual microphone positions of the six simulation conditions.	106
4.14	Coefficients of the four steady-state control filters in simulation (b).	107
4.15	Steady-state control filters in the tuning (blue line) and control (yellow line) stages in simulation (e).	108
4.16	Physical configuration of the $(4 \times 4 \times 4)$ MCANC with 4 virtual microphones in the far field in Experiment 1.	110
4.17	The power spectrum of physical error signals in Experiment 1.	111
4.18	The power spectrum of virtual error signals in Experiment 1.	111
4.19	Noise reduction levels of microphones in Experiment 1.	112
4.20	Physical configuration of the $(4 \times 4 \times 4)$ MCANC with 4 virtual microphones in the near field in Experiment 2.	113
4.21	The power spectrum of physical error signals in Experiment 2.	113
4.22	Power spectrum of virtual error signals in Experiment 2.	114
4.23	Noise reduction levels of microphones in Experiment 2.	114
4.24	Physical configuration of the $(4 \times 3 \times 4)$ MCANC with 4 virtual microphones In Experiment 3.	115
4.25	The power spectrum of physical error signals in Experiment 3.	116
4.26	The power spectrum of virtual error signals in Experiment 3.	116
4.27	Noise reduction levels of microphones in Experiment 3.	117
4.28	Physical configuration of the $(4 \times 3 \times 3)$ MCANC with 4 virtual microphones in Experiment 4.	117
4.29	The power spectrum of physical error signals in Experiment 4.	118
4.30	The power spectrum of virtual error signals in Experiment 4.	119
4.31	Noise reduction levels of microphones in Experiment 4.	119

4.32	Physic configuration of the $(4 \times 3 \times 2)$ MCANC with 4 virtual microphones in Experiment 5.	120
4.33	The power spectrum of physical error signals in Experiment 5.	120
4.34	The power spectrum of virtual error signals in Experiment 5.	121
4.35	Noise reduction levels of microphones in Experiment 5.	121
4.36	Physical configuration of the $(4 \times 3 \times 2)$ MCANC with 2 virtual microphones in Experiment 6.	122
4.37	The power spectrum of physical error signals in Experiment 6.	122
4.38	The power spectrum of virtual error signals in Experiment 6.	123
4.39	Noise reduction levels of microphones in Experiment 6.	123
5.1	The processing block of the SFANC method, which features the first stage of deriving a set of control filters based on commonly encountered noise types and noise paths. Subsequently, selective fixed-filter algorithm extracts the closest control filter to filter out the actual noise in real-time.	127
5.2	Block diagram of single channel feedforward ANC. The primary noise disturbance is denoted as $d(n)$ and the residual error signal $e(n)$ is acoustically measured from microphone at the control region.	129
5.3	Partitioned structure of the filter in K combs. The input signal is down-sampled at K times and then finishes the FFT operation to generate M frequency bins. The frequency bins are input to the M filter branches, and the summation of these branches is the output of the partitioned filter.	133
5.4	Block diagram of the filtered reference comb-partitioning frequency domain adaptive filter (FxCFDAF), which adopts the filtered reference signal based on the secondary path estimate $\hat{S}(z)$	135
5.5	Block diagram of the m th branch in FxCFDAF algorithm.	136
5.6	The construction of coefficient library, which is used to store the pre-trained filters in training stage. The approximate power spectrum $\mathbf{S}_x$ is used to create a bit array A as the address in the library for storing the corresponding control filter coefficients.	137
5.7	Overall block diagram of the SFANC algorithm. It composed of three parts: (1). the FFT transform part of the partitioned reference signal, (2). $M/2$ frequency filter branches, and (3). the noise classification and filter weight update part.	138
5.8	The classification and weight update part in SFANC. The bit vector A created from the approximate power spectrum $\mathbf{S}_x$ that is utilized as the index or address to fetch the corresponding control filter for the specific primary noise.	139

5.9	Implementation architecture of the modified classifier. The index A created from the reference signal is input into the classifier. The first layer computes the binary distance or similarity between input A and the indexes of all the stored pre-trained filters (I pre-trained filters stored in the coefficient library), where ‘&’ stands for the logic AND operation. The index, which owns the maximum similarity ratio with the input A , is used to fetch the corresponding filter in the second layer.	141
5.10	Frequency and phase responses of the primary (upper row) and secondary (lower row) paths.	142
5.11	Noise reduction as a function of similarity ratios and the bandwidth of the training noise B_{train} in simulation A, where the bandwidth of the primary noise B_{primary} is set to 500 Hz. The noise reduction peaks as the similarity ratio approaches 1.	143
5.12	The power spectrum of the primary noise, and the frequency band of the training signal of the control filter chosen by SFANC algorithm.	144
5.13	The mean square error (MSE) of FxLMS and SFANC as a function of time in Simulation B. The SFANC algorithm achieves faster convergence speed and similar steady-state noise reduction performance as the FxLMS algorithm whose step size is less than 0.001. When the FxLMS algorithm chooses the step size of 0.002, it obtains a faster convergence but worse steady-state error than the SFANC algorithm.	145
5.14	The time history of the FxLMS and SFANC algorithm when dealing with the varying primary noise in Simulation C. The primary noise changes every 3.5 seconds. An updated control filter is invoked by the SFANC algorithm upon the onset of a different disturbance.	147
5.15	From top to bottom, the disturbance: without control, after control with FxLMS, and after control with SFANC. The SFANC algorithm shows a faster convergence speed when dealing with the whistle sound than the FxLMS algorithm, without a specific step size selection strategy.	148
5.16	Noise reduction level as a function of FFT accumulation iterations L of the SFANC algorithm in Simulation E. A small value of L ($L < 900$) results in the degradation of the noise reduction performance.	149
6.1	The data flow graph (DFG) of a 3-order FIR filter.	153
6.2	DFG of a 3-order pipelining FIR filter.	154
6.3	Transposed structure of a 3-order pipelining FIR filter.	155
6.4	(a) A DSP program with two addition operations; (b) a folded structure of the two addition operations, where $l \in \mathbb{Z}^+$, and $2l + i$ denotes the $(2l + i)$ th cycle.	157
6.5	(a) The data path $U \rightarrow V$ with $w(e)$ delays; (b) The corresponding new path in the folded structure	157
6.6	Folding a 3-order FIR filter.	158

6.7	Folded structure of the 3-order FIR filter.	159
6.8	Modified folded structure of the 3-order FIR filter with forward-back register allocation.	159
6.9	The dependency graph (DG) of a 3-order FIR filter.	160
6.10	The systolic structure of a 3-order FIR filter.	161
6.11	Block diagram of the conventional FxLMS structure.	163
6.12	Block diagram of the FxLMS algorithm with a N taps control filter.	165
6.13	Block diagram of the systolic FxLMS structure with pipelining of processing modules.	169
6.14	Block diagram of the processing module.	170
6.15	Primary and secondary paths used in the simulation.	171
6.16	Convergence curves of the FxLMS algorithm, delay FxLMS algorithm, and systolic FxLMS structure, when the tap length is 40, using the step size of (a) 0.0025; (b) 0.02; (c) 0.023; (d) 0.03.	172
6.17	Convergence curves of the FxLMS algorithm, delay FxLMS algorithm, and systolic FxLMS structure, when the tap length is 200, using the step size of (a) 0.00155; (b) 0.0025; (c) 0.0045; (d) 0.006.	173
6.18	Processing time of the FxLMS algorithm and systolic FxLMS structure on an Xilinx XC7Z010 FPGA.	174
6.19	A $J \times J \times M$ MCANC using the FxLMS algorithm	175
6.20	Block diagram of the j th control filter.	177
6.21	Block diagram of the reference signal filtering.	178
6.22	Block diagram of the FFMCFxLMS based on J branches.	179
6.23	Block diagram of the FFMCFxLMS based on J branches.	181
6.24	The folded structure of the j th branch.	182
6.25	Proposed architecture of the FFMCFxLMS algorithm.	183
6.26	Statechart diagram of the finite state machine.	184
6.27	Read and write timing of the ports.	185
6.28	Read and write timing of BRAM.	186
6.29	The schematic diagram of the divider.	187
6.30	The front view of the 24-channel ANC window.	189
6.31	The schematic diagram of the 24-channel ANC window.	190
6.32	The cross-sectional view of the ANC unit.	190
6.33	The positions of the observation microphones.	191
6.34	The implementation diagram of the 24-channel ANC system.	192
6.35	The time-averaged SPL of the monitoring microphone A as a function of frequency in the active control of tonal disturbances.	193
6.36	The SPL of monitoring microphone B, C, D, E for sine tones.	194
6.37	The energy-average noise reduction level of microphone A, B, C, D, and E.	195
6.38	The power spectrum of the broadband noise at the monitoring microphone A.	195
C.1	The root locus curve when $\mu \rightarrow \infty$, where $i = 3$ and $\lambda_k = 1$	212

List of Tables

1.1	Health effects linked to exposure at different SPLs [1]	2
3.1	Comparison on the computational complexity of different algorithms	64
3.2	Pseudocode of the 2GD-FxLMS algorithm	66
4.1	System characteristics of alternative sensor-actuator configurations in the tuning and control stages. Here, K , N_v , and M denote the number of secondary sources, virtual microphones, and physical microphones, respectively.	100
5.1	The number of additions and multiplications used in FxLMS, FDAF, SFANC algorithms, when the control filter has 1024 taps.	142
5.2	Frequency band and amplitude of the time-varying broadband noise	146
6.1	Comparison of implementation platforms of noise canceling headset.	162
6.2	The new delay on each data path.	182
6.3	Device utilization of the multichannel FxLMS algorithm on Kintex-7 7K235T.	188
6.4	Broadband noise reduction level at the locations of the monitoring microphones.	196
B.1	The binary vector expressions of two 4-bit binary vectors	210

Symbols and Acronyms

Symbols

p_{p+}	the instantaneous acoustic pressure of the primary noise
p_{s+}	the wave traveling in the positive x direction
p_{s-}	the wave traveling in the negative x direction
ω	the frequency
$e^{j\omega t}$	The complex exponential
k_v	the wave number
λ_v	the acoustic wavelength
λ_{opt}	the optimal leak factor
$\mathbb{R}$	the real number
$\mathbb{C}$	the complex number
$\mathbb{E}[\cdot]$	the expectation operation
$\text{tr}[\cdot]$	the trace operation
$\mathbb{L}(\cdot)$	the Lagrange function
∇J	the gradient vector
$[\cdot]^T$	the transposition of the matrix
$[\cdot]^H$	the conjugate transposition of the matrix
$[\cdot]^{-1}$	the inverse matrix
$*$	the linear convolution
$(\cdot)^*$	the conjugate of the argument
$\text{sgn}(\cdot)$	the sign function
$\text{FFT}[\cdot]$	the fast Fourier transform
$\text{IFFT}[\cdot]$	the inverse fast Fourier transform
$\mathcal{F}^{-1}$	the inverse Fourier transform
∂	the partial differential equation
$\dot{f}$	the derivation operation

$\sum$	the summation
$\neq$	the inequality sign
$\approx$	the approximate equality sign
$\ \cdot\ $	the 2-norm of a vector or matrix in Euclidean space

Acronyms

ANC	Active Noise Control
ASIC	Application-Specific Integrated Circuit
ADC	Analog to Digital Converter
Comb-PFDAF	Comb-partitioning Frequency Domain Adaptive Filter
CNN	Convolutional Neural Network
DAC	Digital to Analog Converter
DFT	Discrete Fourier Transform
DSP	Digital Signal Processor
FDADF	Frequency Domain Adaptive filter
FFT	Fast Fourier Transform
FIR	Finite Impulse Filter
FPGA	Field-Programmable Gate Array
FLANN	Functional Link Artificial Neural Network
FxLMS	Filtered-x Least Mean Square
FxRLS	Filtered-x Recursive Least Square
FxCFDAF	Filtered-x Comb-partitioning Frequency Domain Adaptive Filter
LMS	Least Mean Square
LFxLMS	Leaky FxLMS algorithm
MCANC	Multichannel Active Noise Control
MCFxLMS	Multichannel FxLMS
MMSE	Minimum Mean Square Error
MSE	Mean Square Error

MVANC	Multichannel Virtual Sensing Active Noise Control
NLFxLMS	Nonlinear FxLMS algorithm
PLD	Programmable Logic Device
PSD	Power Spectral Density
SPL	Sound Press Level
SFANC	Selective Fixed-filter Active Noise Control
2GD-FxLMS	Two-gradient Direction FxLMS
THD	Total Harmonic Distortion
VLSI	Very Large Scale Integration
VS-ANC	Virtual Sensing Active Noise Control

Chapter 1

Introduction

1.1 Urban noise

With the rapid pace of urbanization, our living space is becoming more and more congested and surrounded by intricate transportation systems, various construction sites, and busy industrial areas. Unfortunately, the by-product of urbanization, urban acoustic noise from various sources, such as motor vehicles, jet planes, trains, air conditioners, and even human activities [2], regularly impacts our daily lives [3].

How does the acoustic noise affect humans? Our hearing system is a sensitive receiver. Sound waves propagate into our inner ear via the pinna, through the ear canal and vibrate the eardrum. Subsequently, this vibration is transmitted to the cochlea by the tiny chain of bones in the middle ear and drives the fluid in the cochlea to move. The fluid's movement stimulates our nerves creating the sensation of hearing. Hence, the acoustic pressure of sound has a crucial influence on our hearing system. If the acoustic pressure is too high, it will cause human to feel annoyed, uncomfortable, and even undermine the physical mechanism of our ears, leading to hearing loss.

Besides hearing loss, chronic exposure to high levels of noise is also linked to other serious health issues, such as cardiovascular diseases, cognitive impairment, sleep disturbance, and tinnitus [4, 5]. Table 1.1 lists the health effects of different noise levels, measured in decibels (dB), on human health.

TABLE 1.1: Health effects linked to exposure at different SPLs [1]

Average night noise level over a year	Health effects observed in the population
Up to 30 dB	No substantial biological effects are observed (subject to individual differences).
30 to 40 dB	Several effects on sleep observed but seem modest even in the worst cases. Vulnerable groups (e.g. children, elderly) are more susceptible.
40 to 55 dB	Adverse health effects observed among the exposed population. Many people adapt their lives to cope with the noise at night. Vulnerable groups are more severely affected.
Above 55 dB	Considered increasingly dangerous for public health. Frequent occurrence of adverse health effects, with a sizeable proportion of the population highly annoyed and sleep-disturbed. There is evidence that the risk of cardiovascular disease increases.

Therefore, mitigating urban noise is widely regarded as an urgent and necessary issue. Nowadays, there are numerous measures utilized to decrease acoustic noise: at the noise sources, along the noise propagation path, and at the receiver end. Furthermore, many countries even impose strict laws against the factories and car manufacturers that generated excessive noise. Nevertheless, few of these methods can gain good noise cancellation at low frequencies cost-effectively.

1.2 Traditional noise control approaches

One simple but effective traditional method to control noise is to build noise barriers, which line the noise propagation path. These barriers are prevalent along the highways, around airports, and construction sites. The effectiveness of a noise barrier depends on its (1) height (how high to shield from the noise source), (2) thickness (how much energy to block), (3) length (what is the size of the shadow zone), (4) placement (proximity to noise sources or receivers). In the shadow zone, noise barriers can archive 3–7 dB reduction for broadband noise, above 1.5 kHz [6]. However, receivers not in the shadow zone do not benefit from the noise barriers, such as the high-rise buildings.

The noise barrier built in the transmission path of noise usually occludes. Furthermore, due to the scattering of the sound waves (the sound wave can circumvent the object whose size is smaller than the sound's wavelength), the barrier must be huge and of sufficient density to attenuate low-frequency noise. According to the Washington State Department of Transportation in the United States of America, noise barriers cost about US\$730 per square meter. For example, a 5-meter-high barrier will cost \$3.65 million per km. Therefore, the high construction expenditure and the vast size make them hard to be deployed in densely populated urban areas where low-frequency urban noises are problematic.

Another popular passive approach is facade noise insulation, in which the receiver is isolated from the noise environment by sound-absorbing material. The insulation can achieve a good noise attenuation performance because it is usually close to the receiver and absorbs a large part of noise around the receiver. Facade noise insulation hence becomes the natural choice for the buildings, such as windows [7]. In optimal scenarios, partially-opened windows ($<$ hand's breadth) can provide a 10 – 15 dB noise attenuation, whereas fully-closed single layer windows have the insulation of less than 24 dB and double- and triple- glazed windows can provide around 30 – 40 dB attenuation.

A survey about facade noise insulation from 2500 dwellings in Norway shows, when an average noise reduction of 7 dB was achieved, the reported complaints reduced from 42% (before the program) to 16% (after program). However, according to a Swedish study, a lack of ventilation results in sleep disturbance. In the tropical climate, windows cannot be entirely closed to allow natural ventilation into the room.

Furthermore, having all the windows shut with air-conditioning is not in line with sustainability efforts and so future housing should be built for natural ventilation. Many countries (especially in the tropics) with a high percentage of the population in public housing will benefit from economic noise cancellation approaches that do not impede natural ventilation.

An alternative technique, which effectively tackles low-frequency noise while requiring minimal space as compared to other passive methods, is the active noise control (ANC) technique. It only requires a secondary source to generate anti-noise to mitigate the disturbance [8–10]. Hence, compared to traditional approaches, ANC

FIGURE 1.1: (a) Destructive interference (b) Constructive interference.

technique appears to be a relatively low-cost and compact solution that could be massively deployed in residential buildings.

1.3 Introduction to active noise control

The mechanism of ANC, first proposed by Lueg in 1936 [11], is artificially generating an anti-noise wave to interfere with the disturbance destructively, as illustrated in Fig. 1.1. Conventional ANC systems consist of a microphone, an electronic circuit, and a loudspeaker. The microphone acquires the primary noise signal, and the electronic circuit shifts this signal in a specific phase. Subsequently, the loudspeaker converts the signal into the anti-noise wave to attenuate the disturbance.

Since Lueg, many theoretical investigations have been carried out in different applications [8, 10]. The following instance explains how the acoustic noise is canceled by using the ANC method. In an acoustic duct application, there is an incident plane wave (the primary noise) traveling in the positive x direction. According to the wave equation [8, 12, 13], the instantaneous acoustic pressure of the primary noise at x position and t time is given by

$$p_{p+} = Ae^{j(\omega_0 t - k_v x)}, \quad x \in \mathbb{R}. \quad (1.1)$$

The parameters A and ω_0 denote the amplitude and frequency of the primary noise, respectively. k_v presents the acoustic wave number ($k_v = \frac{2\pi}{\lambda_v}$, λ_v denotes the wavelength). An acoustic secondary source (such as the loudspeaker) is mounted

FIGURE 1.2: Active control of plane wave in an infinite duct using single secondary source.

midway in the duct, as illustrated in Fig. 1.2. The duct is assumed to be infinite in length, with an inner rigid wall and a constant cross section.

The secondary source creates an incident wave of the anti-noise with the same frequency as the primary noise. This incident wave propagates in two opposing directions through the duct. p_{s+} denotes the wave traveling in the positive x direction:

$$p_{s+} = B e^{j(\omega_0 t - k_v x)}, \quad x \in [0, \infty); \quad (1.2)$$

p_{s-} denotes the wave traveling in the negative x direction:

$$p_{s-} = B e^{j(\omega_0 t + k_v x)}, \quad x \in (-\infty, 0) \quad (1.3)$$

where B represents the amplitude of the incident wave. If we carefully adjust the amplitude and phase of anti-noise to make $B = -A$, the total sound pressure at the downstream will be

$$p_{p+}(t, x) + p_{s+}(t, x) = 0, \quad x \in [0, \infty). \quad (1.4)$$

Hence, the disturbance noise is entirely canceled by the anti-noise wave in the downstream direction. Meanwhile, the sound pressure in the upstream section is derived as

$$p_{p+}(t, x) + p_{s+}(t, x) = -2jA \sin(k_v x) e^{j\omega_0 t}, \quad x \in (-\infty, 0). \quad (1.5)$$

From (1.5), it can be seen that the synthetic sound in the upstream is a standing wave, whose amplitude harmonically oscillates with frequency ω_0 . Therefore, there is no acoustic energy that being transferred to the downstream.

FIGURE 1.3: Single-channel feedforward ANC system in an air duct.

In general, the ANC system can be classified as feedforward structure and feedback structure. Moreover, the ANC system also can be referred to as single-channel ANC or multichannel based on the number of secondary sources used.

1.3.1 Feedforward structure

Figure 1.3 shows the block diagram of a single-channel feedforward ANC system in an air duct [8]. The single-channel ANC system is composed of two microphones, one loudspeaker, and an electronic controller. The microphone located at the upstream of the duct is the aforementioned reference microphone; the microphone located in the downstream is the error microphone. The reference signal $x(n)$ sampled from the primary noise is passed into the controller to create the control signal $y(n)$ emitted by the secondary source. Ideally, the control signal goes through the secondary path (anti-noise) and entirely cancels the primary noise at the location of the error microphone. At the same time, the residual error signal sensed by the error microphone is used to adjust the control signal of the controller to achieve maximum noise reduction. Hence, with the aid of the reference microphone and the error microphone, the generated anti-noise can dynamically match with the variation of the primary noise, which allows it to deal with many noise types, such as low-frequency broadband noise. Therefore, feedforward ANC is commonly utilized in various industrial applications (e.g., duct noise reduction) [14, 15].

An unfavorable factor of the feedforward ANC implementation is the interference of the reference microphone signal by the feedback noise from the secondary source,

FIGURE 1.4: Single-channel feedforward ANC without feedback noise.

which dramatically impairs the robustness of the system [9]. To avoid the interference of the feedback noise, a simple but effective method [16] is using non-acoustic sensors, as illustrated in Fig. 1.4. Sampling primary noise by non-acoustic sensors also increases the estimate accuracy of the primary noise, which improves the coherence between the reference signal and the primary noise so that the ANC system can achieve a higher noise reduction [8].

1.3.2 Feedback structure

Another prevalent ANC architecture is the single-channel feedback ANC system, as shown in Fig. 1.5. Compared to the feedforward ANC, feedback ANC only requires one microphone, which is used to monitor the residual error. Hence, feedback ANC must estimate (or predict) the information of the primary noise hidden in the residual error. Due to the limitation of system causality, it is much harder to predict random or broadband signals than a periodic signal. Therefore, feedback ANC can only deal with narrow-band noise and periodic noise [9]. However, its simple structure and compact size have the advantage of easy implementation in portable commercial devices, such as in ANC headphones [17, 18].

1.3.3 Multichannel structure

The above ANC setups, as illustrated in Fig. 1.3 and Fig. 1.5, belong to the single-channel structure, which usually achieves the local noise control and only realize a quiet zone around the error microphone in a free-field [19]. For instance, in the

FIGURE 1.5: Single-channel feedback ANC system in an air duct.

cancellation of the high-frequency vehicle noise, the single-channel ANC can only create quiet zones around the headrest rather than to attenuate unwanted sounds in the whole cabin [20]. Conversely, the global approach requires the attenuation of a large free space (e.g., the entire room, large dimension duct). In this case, multichannel ANC is implemented to gain a larger quiet zone through multiple secondary sources and error microphones [13, 21–30]. The block diagram of a multichannel ANC system is shown in Fig. 1.6. The multichannel ANC (MCANC) has J ($J > 0$) reference microphones, K ($K > 0$) secondary sources, and M ($M > 0$) error microphones.

FIGURE 1.6: Block diagram of the MCANC system.

1.4 Objective

With the rapid development of digital signal processors and micro-controllers, digital signal processing has gradually taken center stage in the implementation of the active noise control. Especially when digital ANC adopts adaptive filtering, it achieves improved noise reduction performance and stability compared to previous analog techniques. Nevertheless, there are still some challenges that remain to be solved in the practical implementation of ANC.

The actuator that generates the anti-noise wave to cancel the disturbance is a critical component of the ANC system. However, it faces a big issue—nonlinearity. When the output power of the system exceeds its rated power, the actuator enters the saturation state, which results in many unwanted distortions and reduces its operational longevity. Furthermore, the saturation distortion of the actuator plays the main role in the nonlinearity of the secondary path and may even result in the divergence of the adaptive algorithm. Hence, it is necessary to develop a practical algorithm, which can significantly prohibit the saturation distortion of the actuator while maintaining a satisfactory noise-reduction performance and computational complexity.

In some applications, it is impractical to place the error microphone at the desired position, such as at the eardrums. The virtual sensing ANC method was previously shown to be a feasible solution to solve this problem. It effectively projects the quiet zone to the desired location by placing microphones at more convenient locations. To obtain a larger control zone, the virtual sensing ANC technique must be extended to multichannel ANC (MCANC).

Since current digital ANC systems adopt the adaptive algorithms, they are bounded by the high computational complexity. Moreover, for MCANC systems, the computations soar with the increasing channel count. Hence, the ANC system usually requires a powerful processor, which hugely increases the cost and undermines the adoption of the ANC system. So, it is necessary to modify the current ANC algorithm to reduce its computational requirements so that the modified algorithm can run on low-cost processors without sacrificing too much performance.

An alternative solution to reduce the cost of the ANC system uses the application-specific integrated circuit (ASIC). Moreover, the ASIC technique can help the ANC

system achieve a higher sampling rate to deal with the high-frequency disturbance and obtain a compact size. However, there are few efficient algorithms or architectures for the single-channel ANC implementation on ASIC or FPGA, let alone for the MCANC.

1.5 Major contributions of this thesis

This thesis focuses on using signal processing methods to deal with the practical issues mentioned in the previous section.

I. To tackle the nonlinearity of the actuator in ANC applications, this thesis proposes the two-gradient FxLMS algorithm (2GD-FxLMS) and the optimal leaky FxLMS algorithm. The two-gradient FxLMS algorithm takes two different directions to update the coefficient. When the output signal exceeds the rated power, the algorithm chooses the negative direction that reduces the gain of the control filter. In that, the output power of the system is restrained. Furthermore, 2GD-FxLMS possesses the same computation complexity as the conventional FxLMS, and the implementation result validates its effectiveness. The thesis also suggests an optimal leak factor selection for the leaky FxLMS algorithm without taking consideration of any actuator model. This optimal leak factor can help the leaky FxLMS achieve the optimal noise control under the output constraint.

II. This thesis revisits the single-channel virtual ANC technique and extends it to the multichannel ANC application. The frequency and time domain analysis show that the multichannel virtual sensing ANC can achieve the best noise reduction performance in the desired position without the real error microphones. Moreover, the thesis also investigates the configuration limitation of the actuators and sensors. All of these theoretical results are verified in a 4-channel MVANC system.

III. This thesis proposes a novel selective fixed-filter active noise control (SFANC) algorithm, which selects different pre-trained control filters to cancel the different types of noise. Hence, compared to the conventional adaptive algorithm, it saves a large number of computations that are usually consumed in the adaptive process. It allows the algorithm to be deployed in a low-cost processor. Moreover, without requiring an error signal to update the coefficients, SFANC increases the robustness of the ANC system.

IV. This thesis also proposes a systolic FxLMS algorithm to implement the single-channel feedforward ANC on FPGA. The algorithm integrates the systolic and pipelining structure and hence, obtains a higher sampling rate. Furthermore, the conducted simulation also proves that the systolic FxLMS algorithm possesses a better convergence behavior than the delay FxLMS algorithm.

V. Since there are few dedicated structures for the multichannel ANC implementation on FPGA, the thesis proposes a multiple-parallel-branch and folding architecture that balances the amount of the hardware usage and processing speed. This architecture is used to realize a 24-channel MCANC on FPGA. This 24-channel MCANC system is deployed in an open window, and the experiment results validate the effectiveness of the 24-channel MCANC.

1.6 Outline of the thesis

The thesis is organized as following: Chapter 2 revisits the adaptive algorithms used in active noise control. Section 2.1 and Section 2.2 briefly introduce the basic principle of the adaptive filtering theory and the typical algorithm, the least mean square (LMS) algorithm. To compensate the effects of the secondary path, Section 2.3 retrospects the filtered-x least mean square (FxLMS) algorithm used in single-channel and multichannel feedforward ANC application. Section 2.4 shows some modified active noise control algorithms that improve the performance of the FxLMS algorithm in terms of convergence speed and computational complexity.

Chapter 3 discusses the influence of output nonlinearity on the active noise control system. Section 3.1 and Section 3.2 introduce the distortions of the audio amplifier, which play the main part in the output nonlinearity of ANC, and how the distortions influence the performance of the adaptive algorithm. To solve this issue, Section 3.3 revisits some modified adaptive filter algorithms. Section 3.4 to 3.6 propose the two-gradient direction FxLMS (2GD-FxLMS) and Optimal leaky FxLMS algorithm, which significantly prohibit the output saturation and force the system to operate linearly.

Chapter 4 introduces the virtual sensing technique used in the active noise control application. Section 4.1 to 4.3 review the virtual sensing active noise control (VS-ANC) and extend the technique from single-channel to multichannel applications.

The frequency domain and time domain analysis on multichannel virtual sensing ANC (MVANC) are carried out in Section 4.4. Moreover, Section 4.5 gives the physical limitation of MVANC on the sensor-actuator configuration. Section 4.6 and Section 4.7 exhibit the simulations and experiments of MVANC to validate the analysis.

Chapter 5 proposes the selective fixed-filter feedforward active noise control (SFANC) technique. Section 5.1 briefly introduces the selective fixed-filter active noise control approach. One of the practical filter selection strategy, the frequency-band-match method, is discussed in detail in Section 5.2. Based on the frequency-band-match selection method, Section 5.3 and Section 5.4 propose a novel SANC algorithm. To verify its effectiveness, Section 5.5 carries out a series of numerical simulations on the algorithm.

Chapter 6 discusses the implementation of multichannel ANC (MCANC) on FPGA. Section 6.1 introduces the basic knowledge about the programmable logic device (PLD), and Section 6.2 reviews some optimization methods used to modify the digital processing structure. Based on these techniques, Section 6.3 proposes the systolic FxLMS algorithm that can achieve a high throughput rate. Furthermore, Section 6.4 suggests the multiple-parallel-branch with folding architecture for the implementation of MCANC on an actual system.

Chapter 2

Adaptive algorithms used in feedforward active noise control

As mentioned above, feedforward ANC is good at dealing with broadband noise and thus, widely utilized in many applications. However, it is usually awkward to implement feedforward ANC by analog circuits. In contrast, adaptive filter algorithms play a critical role in current feedforward active noise control systems since their excellent performance and easy implementation in digital processors. This chapter briefly introduces the basic principle of the adaptive filtering theory, and some adaptive algorithms used in active noise control applications.

2.1 Principle of the adaptive filter

Before the adaptive filter was first proposed [31], active noise control (ANC) technique had experienced a long period of stagnation because of difficulty in its implementation. During this period, even though many researchers have realized the signal-channel ANC system with analog circuits, its limited noise reduction performance and instability did little to convince the public that ANC systems are practical and feasible [8]. However, with the rapid development of adaptive filtering theory, many adaptive control structures were proposed and applied in ANC [9]. Furthermore, the commercialization of powerful microprocessors accelerated the applications of compact ANC system [10, 16, 32].

FIGURE 2.1: Block diagram of adaptive filter algorithm.

From then on, the adaptive filter undoubtedly became the kernel of the modern ANC technique, owing to its robustness, optimal noise control performance, and easy realization [33]. To briefly introduce the adaptive algorithm, we consider an adaptive filter used in the context of this section [31], as shown in Fig. 2.1. In this diagram, a transversal filter creates the signal $y(n)$ to cancel the noise $d(n)$. The block diagram of the transversal filter is shown in Fig. 2.2. The input signal vector of the filter is given by

$$\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-N+1)]^T \quad (2.1)$$

Moreover, the weight vector of the filter is written as

$$\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{N-1}(n)]^T \quad (2.2)$$

where $[\cdot]^T$ denotes the transpose operation. Hence, the output signal of the filter is obtained as

$$y(n) = \mathbf{w}^T(n)\mathbf{x}(n). \quad (2.3)$$

The residual error of the adaptive filter is given by

$$\begin{aligned} e(n) &= d(n) - y(n) \\ &= d(n) - \mathbf{w}^T(n)\mathbf{x}(n). \end{aligned} \quad (2.4)$$

FIGURE 2.2: Block diagram of the transversal filter.

The cost function $J(n)$ is the mean square error (MSE) of (2.4) given by

$$\begin{aligned} J(n) &= \mathbb{E} [e^2(n)] \\ &= \mathbb{E} [d^2(n)] - 2\mathbf{p}^T \mathbf{w}(n) + \mathbf{w}^T(n) \mathbf{R} \mathbf{w}(n) \end{aligned} \quad (2.5)$$

where $\mathbb{E} [\cdot]$ denotes the expectation operator. $\mathbf{p}$ denotes the cross-correlation vector given by

$$\mathbf{p} = \mathbb{E} [d(n)\mathbf{x}(n)] \quad (2.6)$$

and $\mathbf{R}$ represents the auto-correlation matrix of the input signal as

$$\mathbf{R} = \mathbb{E} [\mathbf{x}(n)\mathbf{x}^T(n)] . \quad (2.7)$$

The optimal filter is obtained by minimizing (2.5), where the gradient of (2.5) is set to zero, giving

$$\mathbf{w}_o = \mathbf{R}^{-1} \mathbf{p} \quad (2.8)$$

where $\mathbf{R}$ is assumed to be invertible. Substituting (2.8) into (2.5) yields the minimum mean square error (MMSE)

$$J_{\min} = \mathbb{E} [d^2(n)] - \mathbf{p}^T \mathbf{w}_o. \quad (2.9)$$

Combined with (2.9), (2.5) is rewritten as

$$\begin{aligned} J(n) &= J_{\min} + [\mathbf{w}(n) - \mathbf{w}_o]^T \mathbf{R} [\mathbf{w}(n) - \mathbf{w}_o] \\ &= J_{\min} + \mathbf{v}^T \mathbf{R} \mathbf{v} \end{aligned} \quad (2.10)$$

where

$$\mathbf{v} = \mathbf{w}(n) - \mathbf{w}_o. \quad (2.11)$$

2.2 Least mean square (LMS) algorithm

The previous section exhibits the optimal solution of the adaptive algorithm. However, using (2.8) to achieve the optimal filter is impractical, as the cross-correlation vector (2.6) and the auto-correlation matrix (2.7) is usually unavailable in practice. To solve this issue, we replace the mean square error (2.5) by its instantaneous value as

$$J(n) = e^2(n). \quad (2.12)$$

The gradient of (2.12) with respect to $\mathbf{w}(n)$ is given by

$$\nabla J(n) = -2\mathbf{x}(n)e(n). \quad (2.13)$$

Hence, the recursive formula of the filter weight becomes

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{x}(n)e(n) \quad (2.14)$$

where μ denotes the step size, which is a positive value and used to control the update rate of the coefficients. Equation (2.14) is the so-called least mean square error (LMS) algorithm. Compared to other adaptive algorithms, the LMS filter is most widely implemented due to its relatively low computational load and effectiveness.

2.3 Filtered-x least mean square (FxLMS) algorithm

The filtered-x least mean square (FxLMS) is among the most practical adaptive algorithms proposed to compensate for the influence of the secondary path in an ANC system. This section introduces the FxLMS algorithm used in feedforward single- and multichannel ANC applications.

2.3.1 Single-channel FxLMS algorithm

Although the LMS filter is widely used for its low computational load, when it is applied in the ANC system, the effect of the secondary path its convergence

FIGURE 2.3: Block diagram of the FxLMS algorithm applied in a single-channel feedforward ANC system. (1) Green color denotes the acoustic field, and (2) Blue color denotes the digital field.

behavior and stability [34]. In this situation, the FxLMS algorithm is proposed to compensate for the effect of the secondary path while retaining low computational complexity [31, 33]. The block diagram of FxLMS algorithm is illustrated in Fig. 2.3. In the figure, $P(z)$ denotes the transfer function of the primary path from the reference microphone to the error microphone; $W(z)$ represents the control filter; $S(z)$ stands for the transfer function of the secondary path from the secondary source to the error microphone, and its estimation is $\hat{S}(z)$.

The reference signal vector $\mathbf{x}(n)$ is $[x(n), x(n-1), \dots, x(n-N+1)]^T$ and N is the length of the control filter. So, the output of the control filter is

$$y(n) = \mathbf{w}^T(n)\mathbf{x}(n) \quad (2.15)$$

where, $\mathbf{w}(n)$ represents the coefficient vector of the control filter. The residual error signal can be expressed as

$$e(n) = d(n) - s(n) * [\mathbf{w}^T(n)\mathbf{x}(n)] \quad (2.16)$$

where, $d(n)$ and $s(n)$ denote the primary disturbance and the impulse response of secondary path, respectively; $*$ is the linear convolution. To minimize the instantaneous squared error, $J(n) = e^2(n)$, the most widely used method to achieve this objective is the gradient descent method, which updates the coefficient vector in

the negative gradient direction with step size μ :

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{2} \nabla J(n) \quad (2.17)$$

where $\nabla J(n)$ denotes the instantaneous estimate of the MSE gradient at time n , and can be expressed as

$$\nabla J(n) = \nabla e^2(n) = -2\mathbf{x}'(n)e(n). \quad (2.18)$$

The filtered reference signal vector is given by

$$\mathbf{x}'(n) = \hat{s}(n) * \mathbf{x}(n) \quad (2.19)$$

where we replace the impulse response $s(n)$ with its estimate $\hat{s}(n)$. Hence, substituting (2.18) and (2.19) into (2.17) yields the update equation of the FxLMS algorithm

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}'(n). \quad (2.20)$$

If the group delay of the secondary path is assumed to be D_s , the bound of the step size μ in the FxLMS algorithm is given by [33]

$$0 < \mu < \frac{1}{\lambda_{\max} D_s} \quad (2.21)$$

where $\lambda_{\max}$ represents the maximum eigenvalue of the auto-correlation matrix $R_{\mathbf{x}'} = \mathbb{E}[\mathbf{x}'^T(n)\mathbf{x}'(n)]$ of the filtered reference signal.

As described above, a single-channel feedforward FxLMS algorithm is composed of three parts: (1) the control filter, as in (2.15); (2) the secondary path modeling, as in (2.19); (3) the coefficients updating, as in (2.20). Hence, a single channel feedforward FxLMS must complete $3N+1$ multiply operations and $3N-2$ additions in a cycle, assuming that the length of the control filter and secondary path estimate are both N .

2.3.2 Multichannel FxLMS algorithm

Figure. 2.4 shows the block diagram of the multichannel FxLMS (MCFxLMS) algorithm, which has J reference microphones, K secondary sources, and M error

FIGURE 2.4: Block diagram of the multichannel FxLMS algorithm.

microphones [19, 33]. The control filter matrix is given by

$$\mathbf{w}(n) = \begin{bmatrix} \mathbf{w}_{11}^T(n) & \mathbf{w}_{12}^T(n) & \cdots & \mathbf{w}_{1J}^T(n) \\ \mathbf{w}_{21}^T(n) & \mathbf{w}_{22}^T(n) & \cdots & \mathbf{w}_{2J}^T(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{w}_{K1}^T(n) & \mathbf{w}_{K2}^T(n) & \cdots & \mathbf{w}_{KJ}^T(n) \end{bmatrix} \in \mathbb{R}^{K \times JN} \quad (2.22)$$

where $\mathbf{w}_{kj}(n) = [w_{kj,0}(n), w_{kj,1}(n), \dots, w_{kj,N-1}(n)]^T \in \mathbb{R}^{JN \times 1}$ is the control filter from the j th input to the k th output, and N denotes the length of the control filter. The reference vector is given by

$$\mathbf{x}(n) = [\mathbf{x}_1^T(n), \mathbf{x}_2^T(n), \dots, \mathbf{x}_J^T(n)]^T \in \mathbb{R}^{JN \times 1} \quad (2.23)$$

where $\mathbf{x}_j = [x_j(n), x_j(n-1), \dots, x_j(n-N+1)]^T \in \mathbb{R}^{N \times 1}$. The output of the control filter is given by

$$\mathbf{y}(n) = \mathbf{w}(n)\mathbf{x}(n) \in \mathbb{R}^{K \times 1}. \quad (2.24)$$

The anti-noise vector at the error microphones, as illustrated in Fig. 2.4, is given by

$$\mathbf{y}'(n) = \mathbf{s} * \mathbf{y}(n) \in \mathbb{R}^{M \times 1} \quad (2.25)$$

where $\mathbf{s}$ represents the impulse response matrix of secondary paths

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_{11}(n) & \mathbf{s}_{12}(n) & \cdots & \mathbf{s}_{1K}(n) \\ \mathbf{s}_{21}(n) & \mathbf{s}_{22}(n) & \cdots & \mathbf{s}_{2K}(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}_{M1}(n) & \mathbf{s}_{M2}(n) & \cdots & \mathbf{s}_{MK}(n) \end{bmatrix} \in \mathbb{R}^{M \times K}. \quad (2.26)$$

The error signal vector $\mathbf{e}(n)$ measured by M error microphones can be expressed as

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{y}'(n) \in \mathbb{R}^{M \times 1} \quad (2.27)$$

where $\mathbf{d}(n) = [d_1(n), d_2(n), \dots, d_M(n)]^T \in \mathbb{R}^{M \times 1}$ stands for the disturbance vector, with $d_m(n)$ denoting the disturbance at the m th ($m = 1, 2, \dots, M$) error microphone. The cost function of the adaptive filter is defined as

$$J(n) = \sum_{m=1}^M e_m^2(n). \quad (2.28)$$

Hence, the gradient of the cost function is

$$\nabla J(n) = \begin{bmatrix} \sum_{m=1}^M e_m [s_{m1} * \mathbf{x}_1(n)] & \sum_{m=1}^M e_m [s_{m1} * \mathbf{x}_2(n)] & \cdots & \sum_{m=1}^M e_m [s_{m1} * \mathbf{x}_J(n)] \\ \sum_{m=1}^M e_m [s_{m2} * \mathbf{x}_1(n)] & \sum_{m=1}^M e_m [s_{m2} * \mathbf{x}_2(n)] & \cdots & \sum_{m=1}^M e_m [s_{m2} * \mathbf{x}_J(n)] \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{m=1}^M e_m [s_{mK} * \mathbf{x}_1(n)] & \sum_{m=1}^M e_m [s_{mK} * \mathbf{x}_2(n)] & \cdots & \sum_{m=1}^M e_m [s_{mK} * \mathbf{x}_J(n)] \end{bmatrix}$$

where we replace the impulse response $s_{mk}(n)$ with its estimate $\hat{s}_{mk}(n)$ to give

$$x'_{jkm}(n) = \hat{s}_{mk}(n) * x_j(n) \quad (2.29)$$

The control filter weight vector $\mathbf{w}_{kj}(n)$ is updated in the direction of the negative gradient with step size μ based on the minimization of the estimated cost function of (2.28):

$$\mathbf{w}_{kj}(n+1) = \mathbf{w}_{kj}(n) + \mu \sum_{m=1}^M e_m(n) \mathbf{x}'_{jkm}(n) \quad (2.30)$$

where the filtered reference vector is

$$\mathbf{x}'_{jkm}(n) = [x'_{jkm}(n), x'_{jkm}(n-1), \dots, x'_{jkm}(n-N+1)]^T.$$

From the above analysis, the number of multiplications and additions in MCFxLMS algorithm for one cycle is given by

$$N_0 = JKM(N + L_s) + JKN \quad (2.31)$$

where L_s denotes the length of the secondary path estimate. If $J = K = M = L$ and $L_s = N$, the number of multiplications and additions becomes

$$N_0 = L^2N(2L + 1). \quad (2.32)$$

From (2.32), it becomes clear that the computational load exponentially increases with an increasing number of MCFxLMS channels.

2.4 Modified active noise control algorithm

This section briefly introduces some modified active noise control algorithms, which seek to alleviate computational complexity and improve convergence speed.

2.4.1 Normalized FxLMS algorithm

The step size μ is critical to the performance of the FxLMS algorithm. A large step size can accelerate the convergence speed of the algorithm but increases the steady-state error and vice versa. Hence, the normalized FxLMS algorithm has been proposed to achieve a balance between the convergence speed and steady-state error. Moreover, it also leads to the convergence speed independent of the input power [35, 36]. In the algorithm, the step size is a variable to time n . Hence, the update equation is given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu(n)e(n)\mathbf{x}'(n). \quad (2.33)$$

The time-variable step size is expressed as

$$\mu(n) = \frac{2\mu_0}{N\hat{P}_x(n)} \quad (2.34)$$

where μ_0 is a constant within $(0, 1]$; N denotes the length of the control filter, and $\hat{P}_x(n)$ represents the power estimate of the input signal at time n .

According to the moving-average method, the input power estimate of M samples is obtained as

$$\hat{P}_x(n) = \frac{1}{M} \sum_{m=0}^{M-1} = \hat{P}_x(n-1) + \frac{x^2(n) - x^2(n-M+1)}{M}. \quad (2.35)$$

If $M = N$, (2.35) can be rewritten as

$$\hat{P}_x(n) = \frac{1}{N} \mathbf{x}^T(n) \mathbf{x}(n). \quad (2.36)$$

Hence, the update equation of the Normalized FxLMS algorithm is derived as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu_0}{\mathbf{x}^T(n) \mathbf{x}(n)} e(n) \mathbf{x}'(n). \quad (2.37)$$

2.4.2 Sign FxLMS algorithm

In the conventional FxLMS algorithm, the update equation consumes the bulk of the computations. To reduce the computational complexity, the cost function can be defined as

$$J(n) = 2 |e(n)| \quad (2.38)$$

where $|\cdot|$ denotes the absolute value of the argument. Hence, the new update equation is derived as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \cdot \text{sgn}[e(n)] \mathbf{x}'(n) \quad (2.39)$$

where $\text{sgn}[\cdot]$ represents the sign function given by

$$\text{sgn}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \quad (2.40)$$

Another variant of the sign FxLMS algorithm is given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \cdot e(n) \text{sgn}[\mathbf{x}'(n)]. \quad (2.41)$$

and a simpler double-sign FxLMS algorithm is given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \cdot \text{sgn}[e(n)] \text{sgn}[\mathbf{x}'(n)]. \quad (2.42)$$

From (2.39), (2.41), and (2.42), it is evident of that the sign FxLMS algorithm significantly lightens the computational load during update. However, these algorithms substantially sacrifice convergence performance and steady-state error [37].

2.4.3 Moment FxLMS algorithm

The moment FxLMS algorithm originates from the moment LMS algorithm, which is widely applied in the adaptive signal enhancement and neural networks. The moment LMS involves a term of weight variation, which accelerates the convergence of the algorithm while maintaining a satisfactory steady-state error. The moment FxLMS algorithm is derived as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \cdot e(n) \mathbf{x}'(n) + \alpha [\mathbf{w}(n) - \mathbf{w}(n-1)] \quad (2.43)$$

where α denotes the moment factor, and $|\alpha| < 1$.

If the group delay of the secondary path is assumed to be D_s , the bound of the step size of the moment FxLMS algorithm is given by

$$0 < \mu < \frac{1 + \alpha}{\lambda_{\max}} \sin \frac{\pi}{2(2D_s + 1)} \quad (2.44)$$

where $\lambda_{\max}$ represents the maximum eigenvalue of the auto-correlation matrix of the filtered reference signal. When $D_s \gg 1$, (2.44) can be rewritten as

$$0 < \mu < \frac{1 + \alpha}{\lambda_{\max} D_s}. \quad (2.45)$$

Compared to (2.21), the moment FxLMS algorithm has a relaxed step size bound, which means this algorithm has the ability to ease the influence of the eigenvalue spread of the input auto-correlation matrix on the convergence of the control filter.

FIGURE 2.5: Block diagram of the frequency-domain adaptive filter used in ANC.

2.4.4 Frequency domain adaptive filter used in ANC

For the conventional FxLMS algorithm, a larger ratio of the eigenvalues in the input auto-correlation matrix will result in a slower convergence speed. The frequency-domain adaptive filter (FDAF) is one of the effective solutions to overcome this issue [38–41].

The FDAF algorithm deals with the input data by block, which is assumed to contain N samples. As shown in Fig. 2.5, the reference signal, and error signal are transformed into their frequency-domain formats as

$$\mathbf{X}(n) = [X_0(n), X_1(n), \dots, X_{N-1}(n)]^T = \text{FFT}[\mathbf{x}(n)] \quad (2.46)$$

$$\mathbf{E}(n) = [E_0(n), E_1(n), \dots, E_{N-1}(n)]^T = \text{FFT}[\mathbf{e}(n)] \quad (2.47)$$

where $\text{FFT}[\cdot]$ denotes the fast Fourier transform, and the reference and error signal vectors are given by

$$\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-N+1)]^T \quad (2.48)$$

$$\mathbf{e}(n) = [e(n), e(n-1), \dots, e(n-N+1)]^T \quad (2.49)$$

The l th output signal of the control filter is stated to

$$Y_l(n) = W_l(n)X_l(n) \quad (2.50)$$

$$\mathbf{y}(n) = [y(n), y(n-1), \dots, y(n-N+1)]^T = \text{IFFT}[\mathbf{Y}(n)] \quad (2.51)$$

and IFFT denotes the inverse fast Fourier transform, and

$$\mathbf{Y}(n) = [Y_0(n), Y_1(n), \dots, Y_{N-1}(n)]^T. \quad (2.52)$$

The error signal in Fig. 2.5 is expressed as

$$e(n) = d(n) + s(n) * y(n) \quad (2.53)$$

where $s(n)$ represents the impulse response of the secondary path. To minimize the mean square error (MSE) of (2.53), we can derive the update equation of the FDAF algorithm as

$$\mathbf{W}_1(n+N) = \mathbf{W}_1(n) + \mu \mathbf{R}'^H \mathbf{E}(n) \quad (2.54)$$

where

$$\mathbf{R}' = \text{FFT}[\mathbf{x}'(n)] \quad (2.55)$$

$$\mathbf{x}'(n) = \mathbf{x}(n) * \hat{s}(n). \quad (2.56)$$

The $\hat{s}(n)$ denotes the impulse response of the secondary path estimate, and μ stands for the step size. Compared to the conventional FxLMS algorithm, the FDAF algorithm requires fewer computations. However, the delay time coming from the block processing deteriorates the noise cancelation performance for the broadband disturbance.

2.4.5 Filtered-x recursive least square (FxRLS) algorithm

When dealing with broadband interference, the conventional FxLMS algorithm usually faces the slow convergence issue. In contrast, the filtered-x recursive least square (FxRLS) algorithm behaves a faster convergence, which is not sensitive to the eigenvalue ratio of the input auto-correlation matrix, and it also achieves a better steady-state error [42].

The cost function is defined as

$$J(n) = \sum_{k=1}^n \lambda^{n-k} e^2(k). \quad (2.57)$$

where $e(k)$ denotes the error signal at time k ; λ represents the forgetting factor, and $0 < \lambda < 1$.

We assume that the weights of the control filter do not change with L_s time, and the secondary path estimate has L_s taps. Hence, the error signal of FxRLS is given by

$$e(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}'(n). \quad (2.58)$$

$d(n)$ stands for the disturbance, and the filtered reference signal vector is stated to

$$\mathbf{x}'(n) = \mathbf{x}(n) * s(n) \quad (2.59)$$

where $s(n)$ denotes the impulse response of the secondary path. Substituting (2.58) into (2.57) yields

$$J(n) = \sum_{k=1}^n \lambda^{n-k} [d(k) - \mathbf{w}^T(k)\mathbf{x}'(k)]^2. \quad (2.60)$$

By setting

$$\frac{\partial J(n)}{\partial \mathbf{w}(n)} = 0 \quad (2.61)$$

We get

$$\sum_{k=1}^n \lambda^{n-k} \mathbf{x}'(k) \mathbf{x}'^T(k) \mathbf{w}_o = \sum_{k=1}^n \lambda^{n-k} d(k) \mathbf{x}'(k). \quad (2.62)$$

By defining

$$\mathbf{R}'(n) = \sum_{k=1}^n \lambda^{n-k} \mathbf{x}'(k) \mathbf{x}'^T(k) \quad (2.63)$$

$$\mathbf{P}(n) = \sum_{k=1}^n \lambda^{n-k} d(k) \mathbf{x}'(k). \quad (2.64)$$

the (2.62) is rewritten as

$$\mathbf{R}'(n) \mathbf{w}_o = \mathbf{P}(n). \quad (2.65)$$

$\mathbf{R}'(n)$ is assumed to be invertible, and the optimal solution of the least square (LS) method is derived as

$$\mathbf{w}_o = \mathbf{R}'^{-1}(n) \mathbf{P}(n). \quad (2.66)$$

the (2.63) is rewritten as

$$\mathbf{R}'(n) = \lambda \mathbf{R}'(n-1) + \mathbf{x}'(n) \mathbf{x}'^T(n) \quad (2.67)$$

According to Woodbury theorem, the inversion of (2.67) is derived as

$$\mathbf{R}'^{-1}(n) = \frac{1}{\lambda} \left[\mathbf{R}'^{-1}(n-1) - \frac{\mathbf{R}'^{-1}(n-1) \mathbf{x}'(n) \mathbf{x}'^T(n) \mathbf{R}'^{-1}(n-1)}{\lambda + \mu(n)} \right] \quad (2.68)$$

where

$$\mu(n) = \mathbf{x}'^T(n) \mathbf{R}'^{-1}(n-1) \mathbf{x}'(n). \quad (2.69)$$

For convenience, we define

$$\mathbf{C}(n) = \mathbf{R}'^{-1}(n) \quad (2.70)$$

and

$$\mathbf{g}(n) = \frac{\mathbf{C}(n-1) \mathbf{x}'(n)}{\lambda + \mu(n)}. \quad (2.71)$$

Hence, (2.68) is rewritten as

$$\mathbf{C}(n) = \frac{1}{\lambda} \left[\mathbf{C}(n-1) - \mathbf{g}(n) \mathbf{x}'^T(n) \mathbf{C}(n-1) \right]. \quad (2.72)$$

Extending (2.71) yields

$$\mathbf{g}(n) = \mathbf{C}(n) \mathbf{x}'(n). \quad (2.73)$$

Furthermore, (2.64) also can be written as

$$\mathbf{P}(n) = \lambda \mathbf{P}(n-1) + d(n) \mathbf{x}'(n). \quad (2.74)$$

Combing (2.66), (2.72), and (2.74), the new control filter is derived as

$$\begin{aligned} \mathbf{w}(n) &= \mathbf{C}(n) \mathbf{P}(n) \\ &= \frac{1}{\lambda} \left[\mathbf{C}(n-1) - \mathbf{g}(n) \mathbf{x}'^T(n) \mathbf{C}(n-1) \right] \lambda \mathbf{P}(n-1) + d(n) \mathbf{C}(n) \mathbf{x}'(n) \\ &= \mathbf{C}(n-1) \mathbf{P}(n-1) - \mathbf{g}(n) \mathbf{x}'^T(n) \mathbf{C}(n-1) \mathbf{P}(n-1) \\ &\quad + d(n) \mathbf{C}(n) \mathbf{x}'(n). \end{aligned} \quad (2.75)$$

By simplifying (2.75), we get

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \mathbf{g}(n) \mathbf{x}'^T(n) \mathbf{w}(n-1) + \mathbf{g}(n) d(n). \quad (2.76)$$

Therefore, the update equation of FxRLS algorithm is derived from (2.76) as

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{g}(n)e(n|n-1) \quad (2.77)$$

where $e(n|n-1)$ denotes the estimated error by using the optimal control filter of $n-1$ time

$$e(n|n-1) = d(n) - \mathbf{w}^T(n-1)\mathbf{x}'(n). \quad (2.78)$$

The FxRLS algorithm achieves better performance in terms of the convergence compared with other algorithms. However, it also uses a large number of computations. For instance, to implement an N taps control filter, the FxRLS algorithm will need $3N^2 + N$ multiplications and additions.

2.5 Conclusion

In this chapter, Section 2.1 and Section 2.2 retrospect the principle of adaptive algorithm and its typical instance: least mean squares (LMS) algorithm, respectively.

Section 2.3 introduces the filtered-x least square (FxLMS) algorithm that originates from LMS and compensates for the effect of the secondary path in the active noise control (ANC) application. Since its simplicity and effectiveness, the FxLMS algorithm has been the critical benchmark in the adaptive algorithms used in ANC. To further increase the convergence speed or reduce the computational complexity, some modified ANC adaptive algorithms are mentioned in Section 2.4.

Chapter 3

Effect of output nonlinearity on active noise control

The output components comprise the loudspeaker and the amplifier. However, only the amplifier belongs to the active ingredients, it, thus, plays a central role in the output nonlinearity of the ANC system. This nonlinearity caused by the amplifier usually influences the noise reduction performance and stability of the active noise control system in practice. Hence, in this chapter, we analyze how this output nonlinearity affects the performance of the adaptive ANC algorithm and propose two effective ways to overcome this issue.

3.1 Influence of output distortions in active noise control

The audio amplifier, one of the most critical components of an ANC system, provides power amplification for the control signal so that it can drive the loudspeaker to generate the anti-noise wave. In terms of the structure, audio amplifiers are usually classified into three categories; Class A, Class AB, and Class D. Among these three types, Class A and Class AB amplifiers have low latency but low power efficiency; Class D has high power efficiency but long latency. However, no matter what kind of amplifier, they face the same limitation: distortion. There are three common distortions: amplitude distortion, frequency distortion, and phase

FIGURE 3.1: The output magnitude as a function of the input magnitude.

distortion. Since the amplifier is the only active electronic device in the output components of ANC system, its distortion mainly dictates the output nonlinearity of ANC.

3.1.1 Amplitude distortion

Two causes lead to amplitude distortion. Firstly, due to the incorrect biasing level of the amplifier, only part of the signal in the whole cycle is amplified, and the rest is clipped. So, the output signal of the amplifier will be clipped in the positive or negative half. Fortunately, this kind of distortion usually happens to the discrete component amplifier, because it is the only type of amplifier that is sensitive to the biasing level chosen by designers. The integrated audio amplifier avoids this drawback.

The second cause is the large input amplitude. When the amplitude of the input signal exceeds the limitation of the amplifier, the signal will be truncated and cause distortions. For instance, this effect is illustrated by plotting the output magnitude as a function of input amplitude of an amplifier in Fig. 3.1, where V_{thr} denotes the threshold of the amplifier. The gain of the amplifier is assumed to be 1, and the input signal of the amplifier is a sine tone given by

$$y(t) = C \sin(\omega_0 t) \quad (3.1)$$

where ω_0 and C represent the frequency and amplitude of the sine tone, respectively. If C exceeds V_{thr} , the output signal of the amplifier will be distorted and

FIGURE 3.2: The clipped output signal of the amplifier.

truncated, as shown in Fig. 3.2. This clipped signal can be expressed as

$$\bar{y}(t) = \begin{cases} -V_{\text{thr}} & \omega_0 t + 2k\pi \in [\frac{\pi}{2} - \Delta t, \frac{\pi}{2} + \Delta t] \\ C \sin(\omega_0 t) & \text{otherwise} \\ V_{\text{thr}} & \omega_0 t + 2k\pi \in [\frac{\pi}{2} - \Delta t, \frac{\pi}{2} + \Delta t] \end{cases} \quad (3.2)$$

where, $\Delta t = \frac{\pi}{2} - \arcsin\left(\frac{V_{\text{thr}}}{C}\right)$ and $k \in \mathbb{Z}$. It can be written in Fourier series as

$$\bar{y}(t) = \frac{-2\Delta t + \pi + \sin(2\Delta t)}{\pi} C \sin(\omega_0 t) + \frac{4 \cos(\Delta t) \sin^3(\Delta t)}{3\pi} C \sin(3\omega_0 t) + \dots \quad (3.3)$$

From (3.3), the clipped waveform consists of the odd order harmonics. Furthermore, when an audio amplifier exhibits amplitude distortion, it can be regarded as a memoryless system, because the output $\bar{y}(n)$ relies only on the latest input $y(n)$.

3.1.2 Frequency distortion

Frequency distortion is another type of distortion which occurs in power amplifiers, when the level of amplification varies with the frequency. It is a common problem for the discrete component amplifier and the high-frequency amplifier because it involves many inductors and capacitors, which frequently create ripples in the amplitude-frequency response of the amplifier. However, for most audio amplifiers, they have a flatter amplitude-frequency response in the low-frequency because of its

simple structure with few inductors and capacitors. Hence, the frequency distortion has little effect on the audio amplifier.

3.1.3 Phase distortion

When an amplifier operates under saturation or nonlinearly, an unexpected time delay between the input and the output will be present. In this case, the current output of the amplifier does not only rely on the current information but the past input signal. Hence, the amplifier becomes a non-linear memory filter. This time delay is embodied as the phase error in the output signal, which will not only worsen the noise reduction performance but also undermine the stability of the ANC system. It is worth noting that most of the practical audio amplifiers suffer from both the amplitude and phase distortion when they are overdriven.

3.2 Effect of the audio amplifier distortion on feedforward ANC

The distortion of the audio amplifier plays the main role in the output nonlinearity of the active noise control system. This section analyses how the output nonlinearity affects the performance of the adaptive ANC system through amplitude and phase distortions.

3.2.1 Influence of the amplitude distortion on ANC performance

Figure 3.3 shows the block diagram of a single-frequency feedforward ANC controller cascaded with a memoryless amplifier model [31]. $\hat{H}_s(z)$ is the estimate of the secondary path $H_s(z)$. For simplifying the analysis on the effect of the amplitude distortion on the ANC controller, this estimate is assumed to be equal to the real secondary path: $\hat{H}_s(z) = H_s(z) = A_s e^{-j\theta_s}$. The primary disturbance signal $d(n)$ is defined as

$$d(n) = D \sin(\omega_0 n) \quad (3.4)$$

FIGURE 3.3: Single-frequency adaptive noise canceller.

where D denotes the amplitude of the disturbance signal of frequency ω_0 . The reference signal is given by

$$x_0(n) = \sin(\omega_0 n + \theta_r) \quad (3.5)$$

$$x_1(n) = \cos(\omega_0 n + \theta_r) \quad (3.6)$$

where θ_r is the initial phase of the reference signal. Based on the FxLMS algorithm, the coefficient updating equation of the single-frequency adaptive noise canceller is given by [31]

$$w_0(n+1) = w_0(n) + \mu e(n)[x_0(n) * h_s(n)] \quad (3.7)$$

$$w_1(n+1) = w_1(n) + \mu e(n)[x_1(n) * h_s(n)] \quad (3.8)$$

where μ and $e(n)$ denote the step size of the coefficient updating and the residual error signal, respectively. When the coefficient of the control filter reaches its optimal solution, the anti-noise $y(n)$ should be equal to the disturbance noise $d(n)$, i.e.

$$y'(n) = d(n) = D \sin(\omega_0 n). \quad (3.9)$$

3.2.1.1 Case A: Moderate overdriving

When the amplitude of the disturbance signal exceeds the threshold of the audio amplifier, the output signal $\bar{y}(n)$ is distorted and able to be expressed by (3.3). As the overdriving signal is moderate, the fundamental and third harmonics are

dominating in the distorted output, which are written as

$$\bar{y}_{\text{first}}(n) = AC(n) \sin(\omega_0 n + \theta_s) \quad (3.10)$$

$$\bar{y}_{\text{third}}(n) = BC(n) \sin(3\omega_0 n + 3\theta_s) \quad (3.11)$$

where

$$A = \frac{-2\Delta t + \pi + \sin(2\Delta t)}{\pi}, \text{ and } B = \frac{4 \cos(\Delta t) \sin^3(\Delta t)}{3\pi}$$

when the output $y(n)$ of the control filter is actually given by

$$y(n) = C(n) \sin(\omega_0 n + \theta_s). \quad (3.12)$$

The corresponding dominant harmonics in the anti-noise becomes

$$\bar{y}'_{\text{first}}(n) = A_s AC(n) \sin(\omega_0 n) \quad (3.13)$$

$$\bar{y}'_{\text{third}}(n) = A_s BC(n) \sin(3\omega_0 n + 2\theta_s) \quad (3.14)$$

where the amplitude A is much greater than B . When the amplitude of the $y(n)$ slightly exceeds the threshold V_{thr} of the amplifier. The error signal is further expressed as

$$e(n) = d(n) - \bar{y}'_{\text{first}}(n) - \bar{y}'_{\text{third}}(n) \cdots \quad (3.15)$$

where the higher order harmonics of the $\bar{y}'(n)$ are omitted. The new coefficient updating equation is given by

$$w'_0(n+1) = w'_0(n) + \mu e(n) x'_0(n) \quad (3.16)$$

$$w'_1(n+1) = w'_1(n) + \mu e(n) x'_1(n). \quad (3.17)$$

$w'_0(n)$ and $w'_1(n)$ are the coefficients of the controller when the amplifier generates amplitude distortion. Because $x'_0(n)$ and $x'_1(n)$ are orthogonal to the harmonics in anti-noise, after calculating the expectation of (3.16) and (3.17), the coefficient updating equation becomes

$$w'_0(n+1) = w'_0(n) + \mu E\left\{[d(n) - \frac{AC(n)}{D} y'(n)] x'_0(n)\right\} \quad (3.18)$$

$$w'_1(n+1) = w'_1(n) + \mu E\left\{[d(n) - \frac{AC(n)}{D} y'(n)] x'_1(n)\right\} \quad (3.19)$$

where the $C(n)$ is independent of the reference signal $x'_0(n)$ and $x'_1(n)$.

FIGURE 3.4: The equivalent diagram of the single-frequency adaptive noise canceller for harmonics.

Therefore, the optimal coefficients of the controller with the amplitude distortion $w_0'^o$ and $w_1'^o$ are derived as

$$w_0'^o = \frac{1}{A} w_0^o \quad (3.20)$$

$$w_1'^o = \frac{1}{A} w_1^o \quad (3.21)$$

where w_0^o and w_1^o are the optimal coefficients of the controller without the amplitude distortion. It confirms that the FxLMS algorithm has the optimal solution in this situation. The disturbance $d(n)$ is finally cancelled by the fundamental harmonic $\bar{y}_{\text{first}}(n)$ but harmonics are introduced as additional high-frequency annoyance.

The error transfer function of the single-frequency ANC controller is given by [31]

$$G(z) = \frac{Y(z)}{E(z)} \quad (3.22)$$

where $Y(z)$ and $E(z)$ are the z-domain output and error signals, respectively. Based on the simplified structure shown in Fig. 3.4, the system transfer function for harmonics is written as

$$H_e(z) = \frac{1}{1 - G(z)H_s(z)}. \quad (3.23)$$

As (3.23) shows, the transfer function for the harmonics $\bar{y}_{\text{third}}(n)$ forms a positive feedback and hence, introduces instability. Moreover, the amplitude of the harmonics will increase until the gradient of the FxLMS algorithm decreases to 0.

In the moderate overdriving case, we assume that the anti-noise subsequently cancels the disturbance, and hence, $A_s AC(n) = D$. From (3.3), the third harmonic is derived to be constant. Furthermore, the other higher frequency harmonics also can be found to be constant when the control filter reaches the optimal solution.

3.2.1.2 Case B: Severe Overdriving

When the amplitude of disturbance largely exceeds the threshold of the audio amplifier, the distorted output would be similar to a bipolar rectangular wave expressed as

$$\bar{y}(n) \approx \begin{cases} V_{thr} & \omega_0 n + \theta_s \in (2k\pi, (2k+1)\pi] \\ -V_{thr} & \omega_0 n + \theta_s \in ((2k+1)\pi, 2(k+1)\pi], \end{cases} \quad (3.24)$$

where $k \in \mathbb{Z}$, and the output signal $y(n) = C(n) \sin(\omega_0 n + \theta_s)$. The Fourier series of the anti-noise is given by

$$\bar{y}'(n) \approx \frac{4A_s V_{thr}}{\pi} \sin(\omega_0 n) + \frac{4A_s V_{thr}}{3\pi} \sin(3\omega_0 n) \dots \quad (3.25)$$

If the amplitude D of the disturbance $d(n)$ is greater than the amplitude of the fundamental harmonic

$$D > \frac{4A_s V_{thr}}{\pi}, \quad (3.26)$$

the controller would fail to cancel the disturbance. In this situation, the power of the residual error signal is constant and leads to a constant gradient in the FxLMS algorithm, i.e.

$$\nabla \mathbb{E}\{\|e\|^2\} = \frac{1}{2} \left(\frac{4A_s V_{thr}}{\pi} - D \right) \neq 0 \quad (3.27)$$

which will drive the coefficients of the control filter to increase infinitely bound and hence, result in divergence of the algorithm.

Therefore, the single-frequency ANC controller is stable and far from high-frequency annoyance, when the amplitude of the disturbance is $D \in [0, A_s V_{thr}]$. The high-frequency annoyance arises, when moderate overdriving happens, i.e. $D \in (A_s V_{thr}, \frac{4A_s V_{thr}}{\pi}]$. The ANC controller becomes unstable and diverges, when $D \in (\frac{4A_s V_{thr}}{\pi}, +\infty)$. In this case, the fundamental harmonic cannot entirely cancel the disturbance. Hence, the residual error signal will drive the magnitude of the control filter to increase until its coefficients overflow.

3.2.1.3 Simulation: Effect of amplitude distortion

To simulate the effect of the amplitude distortion, the output function of the audio amplifier is configured as

$$\bar{y}(n) = \begin{cases} -0.75 & y(n) < -0.75 \\ y(n) & \text{otherwise} \\ 0.75 & y(n) > 0.75 \end{cases} \quad (3.28)$$

where $y(n)$ and $\bar{y}(n)$ are the input and output signal of the audio amplifier. The primary noise is given by

$$x(n) = C \sin(0.1\pi n). \quad (3.29)$$

The disturbance signal is generated by a simple primary path as

$$d(n) = 0.01x(n-1) + 0.99x(n-2) + v(n) \quad (3.30)$$

where $v(n)$ is a Gaussian white noise with a variance of 0.0323. The secondary path is assumed to be a simple delay with a unit gain ($A_s = 1$). The sampling frequency is set to 10 kHz, and D denotes the amplitude of $d(n)$ ($D \approx C$).

Using (3.26) with a threshold $V_{\text{thr}} = 0.75$, the threshold of the disturbance is calculated as $\frac{4A_s V_{\text{thr}}}{\pi} = 0.9549$. When D equals 0.94, the spectrum of the attenuated noise is shown in Fig. 3.5, whereby the disturbance is sufficiently canceled but the high-order harmonics are observed in the attenuated noise.

Subsequently, D is adjusted from 0.74 to 0.97. The curves of the coefficient $w_1(n)$ are drawn in Fig. 3.6 for different values of D . When D is slightly less than the threshold, the coefficient $w_1(n)$ is close the optimal solution; when the $D \in [0.75, 0.9549)$, the controller converges to a different optimal solution; when $D > 0.9549$, the FxLMS algorithm fails to converge. Therefore, the simulation results agree with the analysis in the previous section.

FIGURE 3.5: Spectrum of the disturbance and attenuated noise at 500 Hz.

FIGURE 3.6: The curves of weight $w_1(n)$ for disturbances with different power.

3.2.2 Influence of the phase distortion on ANC performance

In practical applications, the audio amplifier is not memoryless once it enters into the saturation or the overdriven state. At that time, the output of the amplifier can be described as

$$\bar{y}(n) = \begin{cases} \sum_{i=0}^{\infty} a_i y(n-i) & E\{\|y(n)\|^2\} > \|V_{\text{thr}}\|^2 \\ y(n) & E\{\|y(n)\|^2\} \leq \|V_{\text{thr}}\|^2 \end{cases} \quad (3.31)$$

From (3.31), it becomes clear that additional delays will be brought into the output signal $\bar{y}(n)$ when the output power of the amplifier surpasses its threshold, which brings about a phase error in the secondary path [13].

In this situation, the disturbance can be rewritten as its Fourier series

$$d(n) = \sum_{k=-\infty}^{\infty} \bar{D}_k = \sum_{k=-\infty}^{\infty} D_k e^{jk\omega_0 n} \quad (3.32)$$

where D_k represents the amplitude of the k th harmonic. The output signal of the control filter is given by

$$y(n) = \sum_{k=-\infty}^{\infty} \bar{Y}_k = \sum_{k=-\infty}^{\infty} Y_k e^{jk\omega_0 n}. \quad (3.33)$$

Hence, the squared error of the noise canceller is stated to

$$J = \mathbb{E} \left(\sum_{k=-\infty}^{\infty} \|E_k\|^2 \right) = \mathbb{E} \left[\sum_{k=-\infty}^{\infty} \|\bar{D}_k - H_s(e^{jk\omega_0}) \bar{Y}_k\|^2 \right] \quad (3.34)$$

where $\mathbb{E}(\cdot)$ denotes the expectation operation. Based on the least mean square (LMS) error method, the new output is derived as

$$\bar{Y}_k(n+1) = \bar{Y}_k(n) + \mu H_s^* (e^{jk\omega_0}) E_k(n) \quad (3.35)$$

where μ is the step size. When the amplifier enters saturation, a phase error θ_a is brought into the secondary path. Hence, (3.35) is rewritten as

$$\bar{Y}_k(n+1) = \bar{Y}_k(n) + \mu H_s^* (e^{jk\omega_0}) e^{j\theta_a} E_k(n). \quad (3.36)$$

Substituting the optimal solution $\bar{Y}_k^o$ into (3.36) yields

$$\bar{Y}_k(n+1) - \bar{Y}_k^o = [1 - \mu e^{j\theta_a} H_s^* (e^{jk\omega_0}) H_s (e^{jk\omega_0})] [\bar{Y}_k(n) - \bar{Y}_k^o]. \quad (3.37)$$

Hence, to guarantee the convergence of (3.37), the phase error should be

$$|\theta_a| < \arccos \left[\frac{\mu}{2} H_s^* (e^{jk\omega_0}) H_s (e^{jk\omega_0}) \right]. \quad (3.38)$$

If we assume that the step size μ exists, $\cos \theta_a$ should be positive. Hence, the bound of the step size is derived as

$$|\theta_a| < \frac{\pi}{2} \quad (3.39)$$

Once the phase error, caused by the amplifier's saturation, exceeds this bound, it may deteriorate the system's stability.

3.2.2.1 Simulation: Effect of phase distortion

To simulate the effect of phase distortion, a practical amplifier mode was considered, and its output function was defined as

$$\bar{y}(n) = \begin{cases} -0.75y(n-m) & y(n) < -0.75 \\ y(n) & \text{otherwise} \\ 0.75y(n-m) & y(n) > 0.75 \end{cases} \quad (3.40)$$

where the m denotes the additional delay when the amplifier enters the overdriven state. The other conditions in the simulation were as defined in Subsection. 3.2.1.3.

Figure 3.7 exhibits the spectrum of the attenuated noise at 30s when the phase distortion brought in a 108° phase error, and the D was set to 0.94. The amplitude of the fundamental harmonic gradually exceeded that of the disturbance. As more time elapsed, the FxLMS algorithm finally diverged, as shown in the Fig. 3.8.

Compared to the simulation result of Section 3.2.1.3, even though the amplitude of the disturbance is within the $\frac{4A_s V_{thr}}{\pi} = 0.9549$, the FxLMS algorithm still diverges. Therefore, the phase error created by the overdriven amplifier further deteriorates the robustness of the adaptive algorithm ANC.

FIGURE 3.7: Spectrum of the disturbance and attenuated noise at the phase error of 108° .

FIGURE 3.8: The coefficient $w_1(n)$ of the control filter with a phase error of 108° .

3.3 Modified adaptive filter algorithms to overcome the effect of output saturation

Conventional unconstrained adaptive algorithms are widely used in ANC systems [31]. The steady-state output of these algorithms usually drives actuators (i.e., the combination of loudspeakers and amplifiers) to minimize the residual sound pressure level in a specified area [33]. The amplifier, however, has limited output capacity, reducing the effectiveness of conventional adaptive algorithms in handling large amplitude output signals. In the case of output saturation, the mean square error (MSE) surface of the linear adaptive algorithms severely deforms as the degree of nonlinearity increases [43]. As mentioned in Section 3.2, the output saturation will cause the amplitude and phase distortion of the audio amplifier, which may lead to the divergence of the adaptive algorithm. Therefore, the saturation problem often limits the practical implementation of ANC solutions for suppressing loud noises.

On the surface, the nonlinear adaptive algorithm seems to be an effective solution to overcome the nonlinearity incurred in an ANC system [44]. When the amplifiers are partially overdriven, a typical nonlinear ANC algorithm either uses a truncated Volterra series [45, 46] or a functional link artificial neural network (FLANN) strategy [47] [15,16] to improve the noise reduction performance and convergence speed [44]. Some other algorithms, such as the fuzzy adaptive algorithm, due to their self-tuning capability of the free parameters during the updating progress, also shows promise to handle the distortion of sensors and amplifiers [48]. However, the huge computational expense of the nonlinear ANC algorithms undermines their practicality. More importantly, when the power of the optimum control signal is greater than the output threshold of the amplifier, neither the linear nor the nonlinear ANC algorithms can provide sufficient power to cancel the disturbance. If the anti-noise signal incompletely cancels the disturbance, the residual error accumulates in the weights of the control filter and may result in overflow [49]. Therefore, the nonlinear adaptive algorithm without any constraints may not be the desired solution for the saturation problem of ANC systems, in the case when amplifiers are severely overdriven.

FIGURE 3.9: Block diagram of an ANC system with the saturation effect

3.3.1 Nonlinear filtered-x LMS (NLFxLMS) algorithm

To describe the behaviors of feedforward ANC under the output saturation, a nonlinear model that approximates the saturation curve of the amplifier is cascaded with the secondary path [50], as shown in Fig. 3.9. $f(\cdot)$ denotes the nonlinear model, which is usually defined as the error function

$$f(y) = \int_0^y e^{-\frac{t^2}{2\eta^2}} dt \quad (3.41)$$

where η^2 represents the degree of nonlinearity. The signal $x(n)$ is the reference signal directly retrieved from the primary source, and the residual error signal $e(n)$ is the acoustic superposition in the air of the distorted anti-noise and the disturbance $d(n)$. Hence,

$$e(n) = d(n) - f[y'(n)]. \quad (3.42)$$

The anti-noise is given by

$$y'(n) = \sum_{l=0}^{L_s} s_l \mathbf{w}^T(n-l) \mathbf{x}(n-l) \quad (3.43)$$

where s_l represents the l th coefficient of the impulse response $\mathbf{s}$ of the secondary path with L_s taps. The control filter and reference signal vector are given by

$$\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{N-1}(n)]^T \quad (3.44)$$

$$\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-N+1)]^T \quad (3.45)$$

The cost function is defined as

$$J(n) = e^2(n). \quad (3.46)$$

Hence, the gradient of (3.46) with respect to $\mathbf{w}(n)$ is derived to

$$\nabla J_{\mathbf{w}}(n) = \frac{\partial e^2(n)}{\partial \mathbf{w}(n)} = -2e(n)\dot{f}[y'(n)] \frac{\partial y'(n)}{\partial \mathbf{w}(n)}. \quad (3.47)$$

where $\dot{f}$ stands for the differentiation of $f(\cdot)$. Based on the LMS method, the weight update equation of the nonlinear ANC, by using the negative gradient (3.47), is given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\dot{f}[y'(n)] \mathbf{x}'(n) \quad (3.48)$$

where μ denotes the step size. The filtered reference signal vector is given by

$$\mathbf{x}'(n) = \sum_{l=0}^{L_s} \hat{s}_l \mathbf{x}(n-l) \quad (3.49)$$

where $\hat{s}_l$ denotes the l th coefficient of the secondary path estimate. The differentiation term in (3.48) is expressed as

$$\dot{f}[y'(n)] = e^{-\frac{y'^2(n)}{2\eta^2}}. \quad (3.50)$$

Equation (3.48) represents the weight update equation of the nonlinear filtered-x least mean square (NLFxLMS) algorithm [51]. The NLFxLMS algorithm outperforms the conventional FxLMS algorithm in dealing with the output saturation of the amplifier [50]. However, the exponential function in (3.50) increases its computational complexity. Furthermore, the algorithm also requires the prior knowledge of the amplifier model, which is arguably cumbersome in practical implementation.

3.3.2 Leaky FxLMS algorithm

In practical applications, where the amplifier comes before the secondary path, as shown in Fig 3.10, NLFxLMS may not achieve the optimal solution due to the effect of the secondary path and an inaccurate saturation model. In this situation, the other effective strategy is to constrain the output amplitude of the control filter,

FIGURE 3.10: Block diagram of ANC with the saturation model.

which limits the maximum output level and prevents the amplifiers from being overdriven. The first method is to modify the cost function of the mean square error, as exemplified by the leaky FxLMS algorithm,

$$J(n) = e^2(n) + \gamma \mathbf{w}^T(n) \mathbf{w}(n) \quad (3.51)$$

where γ is the leak factor (to mitigate the coefficient overflow problem, $\gamma \in (0, 0.1)$; otherwise to constraint the output power, $\gamma > 0.1$) [52]. The vector of the control filter is expressed by $\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{N-1}(n)]^T$, where N is the length of the control filter. The coefficient updating equation can be obtained by using the gradient descent method

$$\mathbf{w}(n+1) = (1 - \mu\gamma) \mathbf{w}(n) + \mu e(n) \mathbf{x}'(n) \quad (3.52)$$

where μ is the step size of adaptation, and $\mathbf{x}'(n) = [x'_0(n), x'_1(n), \dots, x'_{N-1}(n)]^T$ is the vector of the filtered reference signal in (3.49).

In (3.51) and (3.52), the leak factor controls the “tightness” of the penalty for the cost function. If the selected leak factor is significant, the control output may not be sufficiently large for satisfactory noise reduction. In contrast, if the selected leak is trivial, the control output may violate the constraint and result in a saturation problem. Hence, the steady-state performances of these leaky FxLMS algorithms are easily affected by the empirical choice of the leak factor or penalty coefficient [53]. Moreover, there is no guarantee that the output signal adheres to the imposed constraints [54].

Furthermore, the noise reduction of the leaky FxLMS algorithm occurs across all

frequencies, resulting in a large, steady-state convergence error [55]. Improved leaky-type FxLMS algorithms introduce constraints into the frequency-domain implementation of the LMS algorithms, where only specific bins that violate constraints are penalized without severely affecting other frequencies [53, 55]. However, these modified algorithms based on penalty method incur higher computational complexity [56–59].

3.3.3 Rescaling algorithm

Some other algorithms search for the optimal solution along the constraints or search between the constraints and the feasible region [60–63]. A typical example based on this strategy is the rescaling algorithm, which is given by

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n+1) + \mu e(n) \mathbf{x}'(n) \\ y(n+1) &= [x(n+1), x(n), \dots, x(n-N+2)]^T \mathbf{w}(n+1)\end{aligned}\tag{3.53}$$

If $|y(n+1)| > C$, then

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n+1) [C/|y(n+1)|] \\ y(n+1) &= y(n+1) [C/|y(n+1)|]\end{aligned}\tag{3.54}$$

where $|\cdot|$ denotes the absolute value, and C is the maximum output magnitude of the amplifier that can be obtained from its specification, circuit simulation or real measurement.

The rescaling algorithm adopts a gradient projection method to rescale the weights of the control filter and the output signal when the filter output violates the constraint. However, in contrast to the conventional FxLMS algorithm, the rescaling algorithm requires more multiplications and divisions, which increases with the length of the control filter and the control channels leading to a lower system sampling rate in the practical implementation of ANC.

3.4 Optimal control filter for output constraint

The alternative way to overcome the output nonlinearity caused by the audio amplifier is to restrain the output power of the system. Specifically, the audio amplifier is forced to work under the linear model at the expense of the noise reduction level. The section introduces the optimal solution of the adaptive filter under the output constraint and analyses its minimum mean square error.

3.4.1 The objective function under output constraint

To obtain the optimal solution of the output-constrained adaptive algorithm, we reconstruct an optimization problem with inequality constraints given by

$$\begin{aligned} \min_{\mathbf{w}} J(\mathbf{w}) &= \mathbb{E} \left[\left| d(n) - \sum_{l=0}^{L_s} s_l \mathbf{w}^T(n-l) \mathbf{x}(n-l) \right|^2 \right] \\ \text{s.t. } g(\mathbf{w}) &= \mathbb{E} \left[|\mathbf{w}^T(n) \mathbf{x}(n)|^2 \right] \leq \rho^2 \end{aligned} \quad (3.55)$$

where ρ^2 denotes the maximum average output power or the rated power of the amplifier and can be measured from experiments or directly refer to the datasheet of the output amplifier. $\mathbb{E}[\cdot]$ represents the expectation operation. The solution of (3.55) achieves the minimum mean square error (MMSE) under the condition that the output power of the control filter does not exceed the constraint ρ^2 . Since the constraint forces the ANC to operate in the linear region, the nonlinearity function $f(\cdot)$ can be ignored.

3.4.2 The optimal solution of the adaptive algorithm with output constraint

To solve the objective (3.55), we first construct a slack variable θ^2 ,

$$\theta^2 = \rho^2 - g(\mathbf{w}). \quad (3.56)$$

Next, we define a Lagrangian function based on (3.55) and slack variable as defined in (3.56) as

$$\mathbb{L}(\mathbf{w}, \lambda, \theta) = J(\mathbf{w}) + \lambda [g(\mathbf{w}) + \theta^2 - \rho^2] \quad (3.57)$$

where λ ($\lambda \in \mathbb{R}$) is the Lagrangian factor. By using the Lagrangian necessary conditions, (3.57) is differentiated with respect to $\mathbf{w}(n)$, λ and θ , respectively, and equated to 0.

$$\frac{\partial \mathbb{L}(\mathbf{w}, \lambda, \theta)}{\partial \mathbf{w}} = \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} + \lambda \frac{\partial g(\mathbf{w})}{\partial \mathbf{w}} = 0 \quad (3.58a)$$

$$\frac{\partial \mathbb{L}(\mathbf{w}, \lambda, \theta)}{\partial \lambda} = g(\mathbf{w}) + \theta^2 - \rho^2 = 0 \quad (3.58b)$$

$$\frac{\partial \mathbb{L}(\mathbf{w}, \lambda, \theta)}{\partial \theta} = 2\lambda\theta = 0. \quad (3.58c)$$

Next, the gradient of the cost function is given by

$$\begin{aligned} \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} &= \frac{\partial}{\partial \mathbf{w}(n)} \mathbb{E} \left[|d(n) - \mathbf{w}^T(n) \mathbf{X}(n) \mathbf{s}(n)|^2 \right] \\ &= \frac{\partial}{\partial \mathbf{w}(n)} \mathbb{E} \left[d^2(n) - 2d(n) \mathbf{w}_o^T \mathbf{X}(n) \mathbf{x}(n) \right. \\ &\quad \left. + \mathbf{w}_o^T \mathbf{X}(n) \mathbf{s}(n) \mathbf{s}^T(n) \mathbf{X}^T(n) \mathbf{w}(n) \right] \\ &= -2\mathbf{P}_{\text{dx}'} + 2\mathbf{R}_{\text{x}'} \mathbf{w}(n) \end{aligned} \quad (3.59)$$

where the vector $\mathbf{P}'_{\text{dx}}$ is $\mathbb{E}[d(n) \mathbf{X}(n) \mathbf{s}]$, and the vector $\mathbf{R}_{\text{x}'}$ is $\mathbb{E}[\mathbf{X}(n) \mathbf{s} \mathbf{s}^T \mathbf{X}^T(n)]$. The input signal matrix, of order $N \times L_s$, is stated as

$$\mathbf{X}(n) = [\mathbf{x}(n), \mathbf{x}(n-1), \dots, \mathbf{x}(n-L_s+1)].$$

The coefficient vector is $\mathbf{s} = [s_0, s_1, \dots, s_{L_s}]^T$, where s_l is the l th coefficient of the secondary path. The gradient of the output constraint can be written as

$$\begin{aligned} \frac{\partial g(\mathbf{w})}{\partial \mathbf{w}} &= \frac{\partial}{\partial \mathbf{w}(n)} \mathbb{E} [\mathbf{w}^T(n) \mathbf{x}(n) \mathbf{x}^T(n) \mathbf{w}(n)] \\ &= 2\mathbb{E} [\mathbf{x}(n) \mathbf{x}^T(n) \mathbf{w}(n)] \\ &= 2\mathbf{R}_{\text{x}} \mathbf{w}(n), \end{aligned} \quad (3.60)$$

where $\mathbf{R}_{\text{x}} = \mathbb{E} [\mathbf{x}(n) \mathbf{x}^T(n)]$. By substituting (3.59) and (3.60) into (3.58a), (3.58b), and (3.58c), the final Lagrangian factor is obtained:

$$\lambda_o = \frac{\mathbf{w}_o^T \mathbf{P}_{\text{dx}'} - \mathbf{w}_o^T \mathbf{R}_{\text{x}'} \mathbf{w}_o}{\rho^2} \text{ or } 0 \quad (3.61)$$

where $\mathbf{w}_o$ is the final optimal solution of the objective function in (3.55). If $\lambda_o \neq 0$, (3.61) can be rewritten as

$$\begin{aligned}\lambda_o &= \frac{\mathbf{w}_o^T \mathbf{P}_{\mathbf{d}\mathbf{x}'} - \mathbf{w}_o^T \mathbf{R}_{\mathbf{x}'} \mathbf{w}_o}{\rho^2} \\ &= \frac{\mathbb{E}\{\mathbf{w}_o^T \mathbf{x}'(n) d(n)\} - \mathbb{E}\{\mathbf{w}_o^T \mathbf{x}'(n) [\mathbf{x}'(n)]^T \mathbf{w}_o\}}{\rho^2} \\ &= \frac{\mathbb{E}\{y'_o(n) d(n)\} - \mathbb{E}\{y'_o(n)^2\}}{\rho^2}.\end{aligned}\tag{3.62}$$

where $y'_o(n) = \mathbf{w}_o^T \mathbf{X}(n) \mathbf{s}(n)$. As the final anti-noise $y'_o(n)$ has the same phase as $d(n)$ (residual error $e(n) = d(n) - y'_o(n)$), and that the amplitude of $d(n)$ is greater than $y'_o(n)$, the expectation $\mathbb{E}[y'_o(n) d(n)] - \mathbb{E}[y'_o(n)^2]$ will become greater than 0. Hence, the range of Lagrangian factor should be:

$$\lambda_o \geq 0.\tag{3.63}$$

Rearranging (3.58a), (3.58b), (3.58c), and (3.63) yields

$$\begin{cases} \partial \mathbb{L}(\mathbf{w}_o, \lambda_o) / \partial \mathbf{w}_o = 0 \\ g(\mathbf{w}_o) - \rho^2 \leq 0 \\ \lambda_o [g(\mathbf{w}_o) - \rho^2] = 0 \\ \lambda_o \geq 0 \end{cases}\tag{3.64}$$

Equation (3.64) satisfies the Kuhn-Tucker conditions, which is the necessary condition for (3.55) to have an optimal solution. Because the reference signal $x(n)$ inherently incorporates measurement noise, the Hessian metric of $J(\mathbf{w})$ and $g(\mathbf{w})$ exist and are positive definite for every $\mathbf{w}(n) \in \mathbb{R}$, and hence they are strictly convex on $\mathbf{w}(n)$. Therefore, (3.55) has only one optimal solution. The optimal solution can be derived from (3.64) as:

$$\mathbf{w}_o = (\lambda_o \mathbf{R}_{\mathbf{x}} + \mathbf{R}_{\mathbf{x}'})^{-1} \mathbf{P}_{\mathbf{d}\mathbf{x}'}.\tag{3.65}$$

Specifically, when λ_o equals to 0, the optimal solution of the control filter reduces to

$$\mathbf{w}_o|_{\lambda_o=0} = \mathbf{R}_{\mathbf{x}'}^{-1} \mathbf{P}_{\mathbf{d}\mathbf{x}'}\tag{3.66}$$

which is the optimal weight vector of the Wiener-Hopf equations. The optimal solution indicates that the anti-noise signal completely cancels the disturbance $d(n)$ within the output constraint. Therefore, the solution of the constrained optimization in (3.55) would reduce to the Wiener-Hopf solution, if the output power remains in the constraint.

3.4.3 Minimum mean square error (MMSE) of the adaptive algorithm with output constraint

Since the error signal $e(n)$ of an active control system is given by

$$e(n) = d(n) - \sum_{l=0}^{L_s} s_l \mathbf{w}^T(n-1) \mathbf{x}(n-l). \quad (3.67)$$

The mean square error can be expressed as

$$\begin{aligned} J(\mathbf{w}) &= \mathbb{E} [|e(n)|^2] \\ &= \mathbb{E} [d(n)^2] - \mathbf{P}_{\text{dx}'}^T \mathbf{w}(n) - \mathbf{w}^T(n) \mathbf{P}_{\text{dx}'} + \mathbf{w}^T(n) \mathbf{R}_{\text{x}'} \mathbf{w}(n) \\ &= \mathbb{E} [d(n)^2] - 2\mathbf{w}^T(n) \mathbf{P}_{\text{dx}'} + \mathbf{w}^T(n) \mathbf{R}_{\text{x}'} \mathbf{w}(n). \end{aligned} \quad (3.68)$$

Also, if $\lambda_o \neq 0$, the third term of (3.64) can be rewritten as

$$\begin{aligned} g(\mathbf{w}_o) &= \mathbf{w}_o^T \mathbf{R}_x \mathbf{w}_o = \mathbb{E} [y_o^2(n)] \\ &= \rho^2. \end{aligned} \quad (3.69)$$

When the $\mathbf{w}(n) = \mathbf{w}_o$, the second term in (3.68) can be expanded to

$$\mathbf{w}_o^T \mathbf{P}_{\text{dx}'} = \mathbf{w}_o^T \lambda_o \mathbf{R}_x \mathbf{w}_o + \mathbf{w}_o^T \mathbf{R}_{\text{x}'} \mathbf{w}_o \quad (3.70)$$

and the third term in (3.68) can be derived as

$$\mathbf{w}_o^T \mathbf{R}_{\text{x}'} \mathbf{w}_o = \sum_{i=0}^{L_s-1} s_i^2 \rho^2. \quad (3.71)$$

where $y(n)$ is assumed to be a white noise, and $\mathbf{s} = [s_0, s_1, \dots, s_{L_s}]^T$ denotes the impulse response of the secondary path. Substituting (3.69), (3.70), and (3.71)

FIGURE 3.11: Mean square error surface of the adaptive algorithm with output constraint when the threshold is (a) less than the power of the disturbance, (b) equals to the power of the disturbance, and (c) larger than the power of the disturbance. The case without constraint is shown in (d). (Note: The gain of the secondary path is assumed to be one.)

into (3.68), the minimal mean square error should be

$$\begin{aligned}
 J_{\min}(\mathbf{w}_o) &= \mathbb{E}[d^2(n)] - 2\mathbf{w}_o^T \mathbf{P}_{dx'} + \mathbf{w}_o^T \mathbf{R}_{x'} \mathbf{w}_o \\
 &= \mathbb{E}[d^2(n)] - 2\lambda_o \mathbf{w}_o^T \mathbf{R}_x \mathbf{w}_o - \mathbf{w}_o^T \mathbf{R}_{x'} \mathbf{w}_o \\
 &= \mathbb{E}[d^2(n)] - \left(2\lambda_o + \sum_{l=0}^{L_s} s_l^2 \right) \rho^2.
 \end{aligned} \tag{3.72}$$

In (3.72), when $\lambda_o > 0$, a bigger threshold ρ^2 can lead to a smaller MSE. If $\lambda_o = 0$, substituting (3.66) into (3.68) yields

$$J_{\min}(\mathbf{w}_o) = \mathbb{E}[d(n)^2] - \mathbf{P}_{dx'}^T \mathbf{w}_o \tag{3.73}$$

which is as the same as the MSE of the conventional FxLMS algorithm.

From (3.72), it is obvious that the threshold ρ^2 relates to the MMSE, which is shown in Fig. 3.11, where the gain of the secondary path is assumed to be unit. If ρ^2 is smaller than the power of the noise disturbance $d(n)$, $\mathbb{E}[d^2(n)]$, the MSE cannot achieve the minimum solution of the algorithm without constraint as depicted in Fig. 3.11 (a). The MSE performance is improved as ρ^2 increases. When ρ^2 equals $\mathbb{E}[d^2(n)]$, the MMSE is as the same as the algorithm without constraint and locates the minimum point on the bound of the MSE surface, as shown in Fig. 3.11 (b). If ρ^2 surpasses the power of the disturbance noise, the adaptive algorithm with constraint can gain the same noise reduction as the conventional FxLMS algorithm, as shown in Fig. 3.11 (c). However, this threshold is decided by the driving capability of the secondary source, which is measured by experiments. Fig. 3.11 (d) illustrates the optimal solution of FxLMS algorithm without constraint.

3.5 Two-gradient direction FxLMS (2GD-FxLMS) algorithm

Although the optimal solution of the output-constrained control filter has been derived in (3.65), the real-time implementation incurs high computation load due to matrix inversion. Hence, we proposed a time-iterative algorithm based on the hem-stitching method [63]. The algorithm searches for the optimal solution along the negative gradient direction of the cost function or the output constraint, depending on the condition, which is described in this section.

When the average output power of the control filter is within the constraint ($g(\mathbf{w}) = \mathbb{E}[\mathbf{w}^T(n)\mathbf{x}(n)] \leq \rho^2$), the weight update is given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{1}{2}\mu \frac{\nabla J(\mathbf{w})}{\|\nabla J(\mathbf{w})\|}. \quad (3.74)$$

In contrast, when the average output power exceeds the constraint ($\mathbb{E}[\mathbf{w}^T(n)\mathbf{x}(n)] > \rho^2$), the weight update changes to

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{1}{2}\mu \frac{\nabla g(\mathbf{w})}{\|\nabla g(\mathbf{w})\|} \quad (3.75)$$

where ∇ and $\|\cdot\|$ denote the differentiation operator and the Euclidean distance, respectively. Since the statistics of the auto-correlation matrix of the reference and the filtered reference signal cannot be directly obtained in practical applications, we approximate the gradient of the cost function and the constraint by their instantaneous value, as

$$\begin{aligned}\nabla J(\mathbf{w}) &\approx \frac{\partial [d(n) - \sum_{l=0}^{L_s} s_l \mathbf{w}^T(n-l) \mathbf{x}(n-l)]^2}{\partial \mathbf{w}(n)} \\ &= -2e(n) \sum_{l=0}^{L_s} s_l \mathbf{x}(n-l) \\ &= -2e(n) \mathbf{x}'(n)\end{aligned}\tag{3.76}$$

and

$$\begin{aligned}\nabla g(\mathbf{w}) &\approx \frac{\partial [\mathbf{w}^T(n) \mathbf{x}(n) \mathbf{x}^T(n) \mathbf{w}(n)]}{\partial \mathbf{w}(n)} \\ &= 2y(n) \mathbf{x}(n).\end{aligned}\tag{3.77}$$

respectively. By substituting (3.76) and (3.77) into (3.74) and (3.75), respectively, and by replacing the normalized gradient and average power constraint with the instantaneous gradient and amplitude constraint, the update equation can now be expressed as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n) \mathbf{x}'(n)\tag{3.78}$$

when the output signal $|y(n)| = |\mathbf{w}^T(n) \mathbf{x}(n)| \leq C$, while

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu y(n) \mathbf{x}(n)\tag{3.79}$$

when the output signal $|y(n)| = |\mathbf{w}^T(n) \mathbf{x}(n)| > C$. C is the equivalent constraint of the output amplitude and can be measured by experiments or obtained from the technical specifications of the amplifier and amplifier. If the output signal is a sine tone, C equals $\rho/\sqrt{2}$. It is noteworthy that, because we replaced the normalized step size $\mu/2\|g(\mathbf{w})\|$ with a constant step size μ in (3.79), the convergence speed and stability of the algorithm will be sensitive to the value of μ [64], which depends on the correlation statistics of the input data [65] and the constraint. If the μ is too small, then (3.79) cannot promptly adjust the control filter back to the feasible area after it violates the constraint, which will lead the distortion in the output signal; However, if μ is large, it will cause a bigger steady-state error in the algorithm.

One potential solution is to use two different step sizes in (3.78) and (3.79), but this requires an in-depth theoretical analysis and practical verifications, and is thus out of the research scope in this section. Henceforth, the same step size will be used for both (3.78) and (3.79).

As indicated in (3.78) and (3.79), the weight of the control filter varies along the same direction as the FxLMS algorithm if the output signal is within its constraint. Once the output signal violates the constraint, it switches the gradient direction of the constraint so that the weight rebounds from the boundary, which reduces the output power. The weight update continues in this manner until it reaches the optimal solution. As the proposed algorithm updates along two gradient directions, it is named the two-gradient direction FxLMS (2GD-FxLMS). Since the constraint in (3.55) restricts the average output power, there is no guarantee that the instantaneous output power falls within the constraint, which results in saturation distortion. To solve this problem, the output signal is clipped when its amplitude is above the constraint. Hence, if $|y(n)| \leq C$ the output signal is given by

$$y_{\text{out}}(n) = y(n) \quad (3.80)$$

If $|y(n)| > C$, the output signal will become

$$y_{\text{out}}(n) = \begin{cases} C, & y(n) > 0 \\ -C, & y(n) < 0 \end{cases} \quad (3.81)$$

where $y_{\text{out}}(n)$ is the clipped output signal.

3.5.1 Simulation: Performance of the 2GD-FxLMS algorithm

The section describes three different simulations to verify the performance of the 2GD-FxLMS algorithm.

3.5.1.1 Control of simulated tonal noise

In the first simulation, we carried out a comparative study on the performance of FxLMS algorithm, leaky FxLMS algorithm, rescaling algorithm, and the proposed

FIGURE 3.12: The output signal of (a) FxLMS algorithm, (b) leaky FxLMS algorithm, (c) rescaling algorithm, and (d) 2GD-FxLMS algorithm.

2GD-FxLMS algorithm in a time-invariant system, through Figs. 3.12 to 3.14. The sampling frequency f_s is set to 10 kHz, and the reference signal $x(n)$ is $2.15 \sin(0.16\pi n)$. The primary and secondary path is chosen as $\mathbf{p} = [0.3, 0.3]^T$ and $\mathbf{s} = [1, 0]^T$, respectively. The step size μ of the adaptive algorithm is 0.05, and the output constraint C is 0.75.

As illustrated in Fig. 3.12, the FxLMS algorithm (or the leaky FxLMS algorithm with leakage factor $\gamma = 0$) failed to control the amplitude of the output signal, while the leaky FxLMS algorithm with $\gamma = 0.25$, rescaling algorithm, and 2GD-FxLMS algorithm successfully confined the output signal within the output constraint. It is worth noting that the leak factor is obtained based on the selection strategy in Section 3.6

In Fig. 3.13, the dashed red curve denotes the average power given by

$$\mathbb{E} [|\mathbf{w}^T(n)\mathbf{x}(n)|^2] = \frac{C^2}{2} = 0.28125 \quad (3.82)$$

FIGURE 3.13: The weight convergence path of the (a) FxLMS algorithm, (b) leaky FxLMS, (c) rescaling algorithm, and (d) 2GD-FxLMS algorithm against contours of the mean square error $\mathbb{E}[e^2(n)]$, with the red curve denoting the constraint $\mathbb{E}[\|\mathbf{w}^T(n)\mathbf{x}(n)\|^2] = 0.28125$.

which divides the solution surface into two parts: (1) The feasible region, where the average output power of the control filter is less than the value $C^2/2$, and contains the origin $(0, 0)$; and (2) the infeasible region, where the average output power of the control filter exceeds the value $C^2/\sqrt{2}$. Although the FxLMS algorithm converges to the optimal solution, as illustrated in Fig 3.13 (a), it is out of the feasible region, and hence, results in the average output power exceeding the power limitation. In contrast, the final solutions of the other algorithms are close to this average power constraint curve. Furthermore, the optimal solution with output constraint was derived from (3.65) and (3.62) as

$$\mathbf{w}_o = \begin{pmatrix} 0.172 \\ 0.206 \end{pmatrix}.$$

In Fig. 3.13, the final solutions of these algorithms are $\mathbf{w}_{\text{FxLMS}} = [0.3, 0.3]^T$, $\mathbf{w}_{\text{leaky}} = [0.162, 0.214]^T$, $\mathbf{w}_{\text{rescaling}} = [0.183, 0.243]^T$, and $\mathbf{w}_{\text{2GD-FxLMS}} = [0.169, 0.204]^T$. Therefore, 2GD-FxLMS is closest to the Kuhn-Tucker's solution. It is evident that

the Leaky FxLMS and proposed algorithm successfully restrain around the average power constraint.

FIGURE 3.14: The residual error of the FxLMS algorithm, leaky FxLMS algorithm, rescaling algorithm, and 2GD-FxLMS algorithm.

Since the leaky FxLMS, rescaling, and 2GD-FxLMS algorithms restricted the output signal in the presence of the constraint, it can be observed that the FxLMS algorithm has the smallest residual error, as shown in Fig. 3.14. The noise reduction after convergence was 8.1 dB, 8.9 dB, and 8.5 dB for leaky FxLMS, rescaling, and 2GD-FxLMS algorithms, respectively. The rescaling algorithm has the better noise reduction at the expense of slightly exceeding the average power limitation, as shown in Fig. 3.13 (c). As one would expect, the FxLMS algorithm almost entirely canceled the noise at the expense of letting the output signal exceed the constraint by a large margin, as illustrated in Fig. 3.12 (a).

3.5.1.2 Control of simulated random noise

The performance of the three algorithms under a time-varying system using random noise as the reference signal is shown through Figs. 3.15 to 3.17. The mean of the

FIGURE 3.15: The control filter output with the (a) leaky FxLMS, (b) rescaling, and (c) 2GD-FxLMS algorithms with a constraint of 0.75, as indicated by the dashed line.

random noise used in the simulations is 0, and its standard deviation is 0.9. In the simulation, the primary path is a time-varying system with $\mathbf{p} = [0.25, 0.35]^T$ for the first 1.5 seconds and then changes to $\mathbf{p} = [0.5, 0]^T$ in the next 1.5 s, and $\mathbf{p} = [0, 0.5]^T$ for the last 1.5 s, totalling 4.5 s. The secondary path remains unchanged with $\mathbf{s} = [1, 0]^T$. The step size of the three algorithms is set to $\mu = 0.001$, and the leakage factor γ in the leaky FxLMS algorithm is set to 0.5.

In Fig. 3.15 (a), it is noted that the leakage factor $\gamma = 0.5$ can limit the amplitude of the output signal with the constraint of $C = 0.75$ for the first primary path but fails for the second and third primary paths. Therefore, the leaky FxLMS with a fixed leakage factor cannot fully guarantee that the output signal can always satisfy the output constraint for a time-varying system or for different primary paths. In contrast, the rescaling and 2GD-FxLMS algorithm successfully restricted the output signal within the constraint during the whole adaptation process, as shown in Fig. 3.15 (b) and (c). These three algorithms successfully tracked the variation of the primary path, as illustrated in Fig. 3.16. Fig. 3.16 (d) depicts the distribution

FIGURE 3.16: The control filter weights of the (a) leaky FxLMS, (b) rescaling, and (c) 2GD-FxLMS algorithm, as a function of the number of iterations; and (d) the distribution ratio of the two gradient directions in 2GD-FxLMS.

ratio of two gradient directions (GD1 and GD2) used in the 2GD-FxLMS algorithm during the adaptive control process. Moreover, 2GD-FxLMS has a smaller residual error than both the rescaling and leaky FxLMS algorithms, as shown in Fig. 3.17. Its average output power constraint performs better with broadband noise.

FIGURE 3.17: The error signal of the (a) leaky FxLMS, (b) rescaling, and (c) 2GD-FxLMS algorithms, as a function of the number of iterations.

3.5.1.3 Tonal control in a single-channel practical active control system

In the final simulation, we considered an actual single-channel feedforward ANC system. The impulse responses of the primary path and secondary path are illustrated in Fig. 3.18 (a) and (b), respectively. In this system, the TDA2003 audio amplifier was used, whose output amplitude limit was 0.75 and gain was close to unity. Its input-output transfer function is approximated by an error function

$$y(n) = \frac{5}{\sqrt{2\pi}} \int_0^{x(n)} e^{-\frac{\tau^2}{0.18}} d\tau \quad (3.83)$$

which is depicted in Fig. 3.18 (c). Due to $\lim_{x \rightarrow \pm\infty} y(n) = \pm 0.75$, this error function will truncate the signal whose amplitude is above 0.75. The sampling frequency f_s is 10 kHz, and the reference signal $x(n)$ is $0.32 \sin(0.1\pi n)$.

In the simulation, the FxLMS, rescaling, and the 2GD-FxLMS algorithms have been deployed to cancel the noise disturbance. Furthermore, the length of the control filter and secondary path estimate is 512 and 256 taps, respectively. As the

FIGURE 3.18: The impulse response of the (a) primary path, and (b) secondary path, of a real ANC system, and (c) output value vs. the input value of the amplifier TDA2003.

FxLMS algorithm did not restrict the output signal, the amplitude of the output signal quickly exceeded the constraint, as shown in Fig. 3.19 (a). The large output signal overdrives the amplifier, which causes the amplifier to distort the output signal and hence, generates significant distortions and harmonics in the residual error, as shown in Fig. 3.20 (b). In this case, the noise disturbance far exceeds the maximum output limit of the secondary source, which results in a high residual error that pushed the filter weights increasingly beyond the limit. The harmonics of the residual error also increased and finally exceeded the power of the disturbance, pushing the total harmonic distortion (THD) to -17.4 dB (13.5%). Even though the FxLMS algorithm reduced the disturbance by 23 dB, this came at the expense of increased harmonic distortions (e.g., 3rd harmonic distortion increased to 83 dB). Note that the power of the disturbance is just 83 dB as shown in Fig. 3.20 (a).

The leaky algorithm even with a large leaky factor $\gamma = 2$ failed to restrict the output signal within the constraint. Although the leaky FxLMS algorithm achieved an attenuation of 12 dB, as shown in Fig. 3.20 (c), its final output signal is slightly

FIGURE 3.19: The output signal of the (a) FxLMS algorithm, (b) leaky FxLMS algorithm, (c) rescaling algorithm, and (d) the 2GD-FxLMS algorithm, all as a function of the number of iterations.

above the constraint, which resulted in obvious distortion in the residual error with a THD of -18.6 dB (11.7%).

The rescaling algorithm successfully confines the output signal within the output constraints, as illustrated in Fig. 3.19 (c). Its spectrum of the residual error after convergence is shown in Fig. 3.20 (d). Despite the disturbance decreasing by 12 dB, some harmonics still exist in the residual error with a THD of -34.4 dB (1.9%).

Similar to the rescaling algorithm, the 2GD-FxLMS effectively restricts the output signal, as shown in Fig. 3.19 (d). Although its noise reduction is just 12 dB,

FIGURE 3.20: The spectrum of the error signal for a real ANC system, (a) without control, (b) with FxLMS algorithm, (c) with leaky FxLMS algorithm, (d) with the rescaling algorithm, and (e) with the 2GD-FxLMS algorithm, all as a function of normalized frequency. The distribution ratio of two gradient directions in 2GD-FxLMS algorithm is shown in (f).

only a minor third harmonic is observed in the residual error signal, as shown in Fig. 3.20 (e), and further exemplified by the low THD of -55 dB (0.2%). Its distribution ratio of two gradient directions shown in Fig. 3.20 (f) indicates that the algorithm spends 9.8% of the time on restricting the output signal by using the second gradient direction. Therefore, this simulation highlights the capabilities of the 2GD-FxLMS algorithm in constraining the output signal, as well as in prohibiting the nonlinearity due to saturation in the ANC system.

TABLE 3.1: Comparison on the computational complexity of different algorithms

Algorithm	Multiplication	Addition	Division	Switch	Comparison
FxLMS	$2N + L_s + 1$	$2N + L_s - 2$	0	0	0
LFxLMS	$3N + 1$	$2N + L_s - 1$	0	0	0
Rescaling	$3N + L_s + 2$	$2N + L_s - 2$	1	0	1
Proposed	$2N + L_s + 1$	$2N + L_s - 2$	0	$N + 1$	1

3.5.2 Implementation of 2GD-FxLMS algorithm

The section analyses the implementation of the 2GD-FxLMS algorithm and shows its experimental result in an air duct.

3.5.2.1 An analysis on the computational complexity of the algorithm

Computational burden is a critical issue that should be taken into account when referring to the practical implementation of an adaptive algorithm. For a real-time platform, such as the digital signal processor (DSP), the microcontroller unit (MCU), or the field programmable gate array (FPGA), the burden is mainly contributed by multiplication/division, addition/subtraction in the execution of the algorithm. Compared to the conventional FxLMS algorithm, the 2GD-FxLMS algorithm has the same computational complexity (multiplication and addition), as illustrated in Table 3.1.

However, the 2GD-FxLMS also requires several additional switches and one threshold, as shown in Fig. 3.21. When the output signal $y(n)$ exceeds the constraint, it will be clipped by the threshold. Meanwhile, the threshold triggers the switches to select the “Y” branch so that the weight update direction changes to reduce the output power until the output signal returns within the constraint. In contrast, if the output signal is within the constraint, the switches choose the “N” branch, which forces the algorithm to search for the optimal solution in the negative gradient direction of FxLMS algorithm.

To swap the two gradient directions of the 2GD-FxLMS algorithm in the processor, we can use some conditional statements, such as “if...else”, as the pseudocode shows in Table 3.2. The conditional branch sentences cost fewer computations compared to the addition, multiplication, and division. When we implement the

FIGURE 3.21: The detailed block diagram of the 2GD-FxLMS algorithm.

2GD-FxLMS on a digital circuit, such as VLSI or FPGA, the mechanism of the switch in Fig. 3.21 can be realized by the multiplexer. The multiplexer is a basic component in the digital circuit and comprises of a few gates. Compared to the adder, multiplier, and divider, the number of gates used in the multiplexer is negligible. Therefore, the additional complexity for the control part of the 2GD-FxLMS algorithm is trivial compared to the computational complexity of the conventional adaptive algorithm.

3.5.2.2 Experiment results of 2GD-FxLMS

To use the 2GD-FxLMS algorithm for solving amplifier saturation, the following procedure is implemented.

Stage 1: Obtain the threshold ρ^2 of the power amplifier by experimental measurement or referring to its manual book.

Stage 2: Implement 2GD-FxLMS as described in Table 3.2.

TABLE 3.2: Pseudocode of the 2GD-FxLMS algorithm

Two gradients FxLMS algorithm (2GD-FxLMS)
Input: Reference signal $x(n)$ and error signal $e(n)$.
Output: Clipped Output signal $y_{out}(n)$.
Step 1: $y(n) = \mathbf{w}^T(n)\mathbf{x}(n);$ $x'(n) = \hat{\mathbf{s}}^T(n)\mathbf{x}(n).$
Step 2: If $ y(n) \leq C$ $\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}'(n);$ else $\mathbf{w}(n+1) = \mathbf{w}(n) - \mu y(n)\mathbf{x}(n).$
Step 3: If $y(n) > C$ $y_{out} = C;$ else if $y(n) < -C$ $y_{out}(n) = -C;$ else $y_{out}(n) = y(n).$

Stage 3: Determine μ_1 in (3.78) as the conventional FxLMS. Facing severe saturation distortion, adopt a large μ_2 in (3.79) to exert a tight restraint on the output power. However, if higher noise reduction with slight distortion is acceptable, a small μ_2 is more suitable.

A real-time implementation based on the feedforward FxLMS and 2GD-FxLMS algorithms in a NI PXIe 8880 controller was carried out in a small noise duct. The audio amplifier TDA2030A had a threshold of 750 mV. The NI PXIe-6368 module accomplished analog input, signal conditioning, and output of the system, with a sampling rate of 16 kHz. The length of the control filter and the secondary path estimate were chosen as 512 taps and 256 taps, respectively. Two primary noise signals were tested: a 1 kHz tone, and a broadband signal from 500 Hz to 1.4 kHz. For the tonal noise, as shown in Fig. 3.22, FxLMS reduces the disturbance from 58 dB to 43 dB but cannot prevent the saturation distortion, resulting in a total harmonic distortion (THD) of 174%. In contrast, 2GD-FxLMS not only mitigates the disturbance to 53 dB but also prevents the audio amplifier from distorting with THD of 0.6%. For broadband noise, 2GD-FxLMS also suppresses the high-frequency distortion better than FxLMS, as illustrated in Fig. 3.23.

FIGURE 3.22: Error signals' power spectrum of FxLMS and 2GD-FxLMS for the tonal noise.

3.6 Optimal leak factor selection for Leaky FxLMS algorithm

As mentioned in previous sections, the saturation distortion from the overdriven amplifier and amplifier will deteriorate both the attenuation performance and residual sound quality. The severe distortions even result in the divergence of the adaptive algorithm. Hence, numerous adaptive algorithms have been developed to solve this problem, by minimizing the adaptive filter output power. Owing to its stability

FIGURE 3.23: Error signal power spectrum of FxLMS and 2GD-FxLMS as a function of frequency for the broadband noise.

and low complexity, the leaky filtered-x least mean square (LFXLMS) algorithm is the most widely implemented in ANC applications.

The LFXLMS algorithm introduces a leak factor in its cost function, which forces the algorithm to seek the optimal control filter with the minimal cost while confining the weight of the filter. Hence, the leak factor is vital to the performance of the algorithm. A significantly large leak factor will tightly restraint the weight of the filter, but often results in poor noise reduction due to a significant bias in the control filter. On the contrary, a small leak factor is usually insufficient in confining the power of the output signal within a safe range, resulting in saturation nonlinearities.

Due to ambiguity in the leak factor selection criterion, many trials and errors are required for different applications. Hence, a constant leak factor has been proposed based on a specific nonlinear saturation model. However, this deterministic leak factor requires the system saturation model and an impulse response estimate of the primary path – the acoustic path from the reference sensor to the error sensor where the noise is to be attenuated. An alternative based on the assumption that the reference signal is independent of the control filter is ineffective for narrow-band noise. There are also modified approaches, with a variable and appropriately-sized

leak factor, do not require an offline prediction of the leak factor. However, these modified approaches are computationally complex, hindering their practical implementation. Hence, this section presents an optimal leak factor selection algorithm, which guarantees that the LFXLMS algorithm reaches the optimal control filter within a specified output power constraint. The proposed leak factor selection algorithm does not require any system saturation model nor the primary path information and can be readily calculated for real applications.

3.6.1 Derivation of the optimal leaky factor

The optimal control filter with output constraint is given by (3.65), where the Lagrangian factor has been proven $\lambda \geq 0$. Hence, the resultant system operation can be described in two cases.

Case 1 : As indicated by (3.62), when $y'_o(n)$ equals $d(n)$, λ_o is 0. It indicates that the system possesses enough power to mitigate the disturbance, and the system operates linearly.

Case 2 : The final anti-noise $y'_o(n)$ is smaller than $d(n)$, which refers to the case when the system is incapable of canceling the disturbance entirely.

To investigate the relationship between $y'_o(n)$ and $d(n)$, we analyze the system in frequency domain [13]. As such, the error signal can be expressed as

$$E(\omega) = X(\omega)P(\omega) - X(\omega)W(\omega)S(\omega) \quad (3.84)$$

where $P(\omega)$, $W(\omega)$ and $S(\omega)$ denote the transfer functions of the primary path, control filter, and secondary path, respectively. $X(\omega)$ is the reference signal. Hence, the optimal control filter is similarly obtained by solving

$$\begin{aligned} \min_W J(W) &= \mathbb{E}[E(\omega)E^*(\omega)] \\ \text{s.t. } g(W) &= \mathbb{E}[W(\omega)X(\omega)X^*(\omega)W^*(\omega)] \leq \phi^2 \end{aligned} \quad (3.85)$$

where ϕ^2 represents the output constraint. To solve (3.85), we define a Lagrangian-like function given by

$$\mathbb{L}(W, \lambda) = J(W) + \lambda [g(W) - \phi^2] \quad (3.86)$$

where the Lagrangian factor λ is a real scalar. According to the Karush-Kuhn-Tucker condition [66],

$$\begin{cases} \nabla_W \mathbb{L}(W_o, \lambda_{\text{opt}}) = 0 \\ \nabla_\lambda \mathbb{L}(W_o, \lambda_{\text{opt}}) \leq 0 \\ \lambda_{\text{opt}} [g(W_o) - \phi^2] = 0 \\ \lambda_{\text{opt}} \geq 0 \end{cases} \quad (3.87)$$

where ∇ represents the gradient, we obtain

$$(1 + \lambda_{\text{opt}})\phi^2 = P(\omega)S_x(\omega)W_o^*(\omega)S^*(\omega) \quad (3.88)$$

and

$$\phi^2 = S(\omega)W_o(\omega)S_x(\omega)W_o^*(\omega)S^*(\omega) \quad (3.89)$$

where $W_o(\omega)$ and λ_{opt} represent the optimal control filter and the leak factor, respectively. Combining (3.88) and (3.89) yields

$$(1 + \lambda_{\text{opt}})S(\omega)W_o(\omega) = P(\omega). \quad (3.90)$$

So, the disturbance can be represented as

$$\begin{aligned} d(n) &= x(n) * \mathcal{F}^{-1}[P(\omega)] \\ &= x(n) * \mathcal{F}^{-1}[(1 + \lambda_{\text{opt}})W_o(\omega)S(\omega)] \\ &= (1 + \lambda_{\text{opt}}) \cdot y'_o(n) \end{aligned} \quad (3.91)$$

where $\mathcal{F}^{-1}[\cdot]$ and $*$ denote the inverse Fourier transform and convolution operation, respectively. Since the secondary path is the non-minimum phase in practice, it is only to obtain an approximation of the perfect control filter $W_o(\omega)$ in (3.91) [67, 68]. Therefore, the optimal anti-noise $y'_o(n)$ is approximately linearly correlated to the disturbance $d(n)$, and their correlation coefficient is given by

$$\frac{\mathbb{E}\{[y'_o(n) - \mathbb{E}[y'_o(n)]] [d(n) - \mathbb{E}[d(n)]]\}}{\sqrt{\sigma_{y'_o}^2} \sqrt{\sigma_d^2}} \approx 1 \quad (3.92)$$

where $\sigma_{y'_o}^2$ is the variance of the final anti-noise. As the expectation of the $y'(n)$ and $d(n)$ equates to 0 (ANC can only deal with the bipolar signal), we obtain $\mathbb{E}[y'_o(n)d(n)] = \sqrt{\sigma_{y'_o}^2} \sqrt{\sigma_d^2}$ from (3.92), and $\mathbb{E}[y'_o(n)^2] = \sigma_{y'_o}^2$. Hence, (3.62) can be

rewritten as

$$\lambda_o = \frac{\sigma_{y_o'}^2}{\rho^2} \left(\sqrt{\frac{\sigma_d^2}{\sigma_{y_o'}^2}} - 1 \right). \quad (3.93)$$

The variance of the final anti-noise is given by

$$\sigma_{y_o'}^2 = \mathbb{E} \left\{ \left[\sum_{l=0}^{L_s-1} s_l y_o(n-l) \right]^2 \right\} = \sum_{l=0}^{L_s-1} s_l^2 \sigma_{y_o}^2 \quad (3.94)$$

where the output signal $y(n)$ is assumed to be zero-mean white noise, and $\sigma_{y_o}^2$ presents the variance of the final output signal $y_o(n)$. Substituting (3.94) into (3.93) yields

$$\lambda_o = \frac{\sigma_{y_o}^2 \sum_{l=0}^{L_s-1} s_l^2}{\rho^2} \left(\sqrt{\frac{\sigma_d^2}{\sum_{l=0}^{L_s-1} s_l^2 \sigma_{y_o}^2}} - 1 \right). \quad (3.95)$$

It should be noted that when the amplifier is saturated, the final output signal has the maximum average power ($\sigma_{y_o}^2 = \rho^2$). By considering both the cases above, we define the degree of nonlinearity η^2 of the system as

$$\eta^2 = \max \left(\frac{\sigma_d^2}{\sum_{l=0}^{L_s-1} s_l^2 \rho^2}, 1 \right) \quad (3.96)$$

where $\max(\cdot)$ is the maximum value of arguments. Hence, the optimal Lagrangian factor (3.95) can be expressed as

$$\lambda_o = \sum_{l=0}^{L_s-1} s_l^2 (\eta - 1). \quad (3.97)$$

To simplify this analysis, we assume that the reference signal $x(n)$ is zero-mean white noise. So, the optimal control filter (3.65) with the output constraint can be rewritten as

$$\mathbf{w}_o = (\lambda_o \sigma_x^2 \mathbf{I} + \mathbf{R}_{x'})^{-1} \mathbf{P}_{dx'} \quad (3.98)$$

where σ_x^2 and $\mathbf{I}$ denote the variance of the reference signal and an identity matrix. Based on the derivation in [69, 70], the optimal solution of the LFXLMS algorithm (3.51) is given by

$$\mathbf{w}_o = (\gamma \mathbf{I} + \mathbf{R}_{x'})^{-1} \mathbf{P}_{dx'} \quad (3.99)$$

when γ is a positive scalar, or

$$\mathbf{w}_o = (\gamma + \mathbf{R}_{x'})^{-1} \mathbf{P}_{dx'} \quad (3.100)$$

when γ is a positive definite matrix [70] (proven in A.1). To ensure that the LFXLMS algorithm has the same optimal solution as (3.55), (3.99) should equate to (3.98). Hence, the sufficient condition is given by

$$\gamma = \lambda_o \sigma_x^2 \quad (3.101)$$

Substituting (3.97) into (3.101) yields the optimal leak factor as

$$\gamma_o = \sigma_x^2 \sum_{l=0}^{L_s-1} s_1^2 (\eta - 1). \quad (3.102)$$

In consideration of practical cases where the reference signal and the output signal are not white noise, their auto-correlation matrices will not be diagonal. To ensure that the LFXLMS algorithm has the same solution as (3.55), (3.100) should equate to (3.65). Hence, the sufficient condition is given by A.1

$$\gamma = \lambda_o \mathbf{R}_x. \quad (3.103)$$

Substituting (3.93) into (3.103) yields the optimal leak factor as

$$\gamma_o = G_s (\eta - 1) \mathbf{R}_x. \quad (3.104)$$

where $G_s = \sigma_{y_o}^2 / \rho^2$ is the power gain of the secondary path. If the spectrum of the control signal is flat or narrowband, such as a tonal noise, G_s can be written as

$$G_s = \frac{1}{2\pi} \int_{\omega_1}^{\omega_2} |S(e^{j\omega})|^2 d\omega \quad (3.105)$$

where the signal frequency ranges between $[\omega_1, \omega_2]$. The degree of system nonlinearity also changes to

$$\eta^2 = \max \left(\frac{\sigma_d^2}{G_s \rho^2}, 1 \right). \quad (3.106)$$

Furthermore, the parameters in either (3.102) or (3.104) can be readily measured in a real-world system. Subsequently, by following the detailed steps in Fig. 3.24,

FIGURE 3.24: The flow chart of the LFXLMS algorithm with the optimal leak factor.

we can obtain the optimal leak factor before the active control phase.

3.6.2 Simulation: LFXLMS algorithm with the optimal leak factor

In the first simulation, the reference signal is a sinusoid $x(n) = 2.15 \sin(0.16\pi n)$, and the power constraint of the system is $\rho^2 = 0.75^2/2$. The secondary path is set to $\mathbf{s} = [1, 0]^T$, and the step size of algorithms is $\mu = 0.005$. Fig. 3.25 depicts the convergence paths of the LFXLMS algorithm under four different primary paths. In Fig. 3.25 (a), the primary path is $\mathbf{p}_1 = [0.1, 0.2]^T$, and the optimal solution falls within the constraint, which is the case where ANC cancels the disturbance without violating the output constraint. Hence, the optimal leak factor γ_o is $\mathbf{0}$, and the LFXLMS algorithm reaches the optimal solution. When the primary path changes to $\mathbf{p}_2 = [0.1, 0.2834]^T$, the optimal solution of the control filter coincides with the output constraint, as illustrated in Fig. 3.25 (b). In this situation, the leak factor γ_o remains as $\mathbf{0}$, and the LFXLMS converges to the optimal control filter. When the primary path changes to $\mathbf{p}_3 = [0.35, 0.3]^T$ and $\mathbf{p}_4 = [0.15, 0.35]^T$, the optimal solution of the control filter without the output constraint are located outside the

FIGURE 3.25: The convergence paths of the leaky FxLMS algorithm with the optimal leak factor, for different primary paths with a sinusoid as the reference signal.

constraint, as shown in Fig. 3.25 (c) and (d). In these cases, the leak factor derived from (3.104) still guarantees LFxLMS algorithm to the optimal solution without violating the output constraint.

Figure 3.26 exhibits the LFxLMS algorithm convergence paths of the second simulation with different primary paths using White Gaussian noise $N(0, 2.5)$ as the reference signal. The secondary path and the step size are set to $\mathbf{s} = [0, 0.8]^T$ and $\mu = 0.001$, respectively, and the average-power constraint is set to $\rho^2 = 0.75^2$. The LFxLMS algorithm converges to the optimal solutions with the leak factor $\gamma_o = 0$ under the primary paths $\mathbf{p}_1 = [0, 0.8, 0.16]^T$ and $\mathbf{p}_2 = [0, 0.8, 0.2267]^T$, as shown in Fig. 3.26 (a) and (b), respectively. Their optimal solutions are located within the constraint and at the edge of the constraint, respectively. In contrast, when the primary path changes to $\mathbf{p}_3 = [0, 0.28, 0.24]^T$ and $\mathbf{p}_4 = [0, 0.13, 0.28]^T$, the optimal solutions without output constraint are outside of the feasible area. Meanwhile, the LFxLMS algorithms with the optimal leak factor $\gamma_o = 2.1$ and $\gamma_o = 1.1$ successfully approach their corresponding restrained optimal solutions, which are located

FIGURE 3.26: The convergence paths of the leaky FxLMS algorithm with the optimal leak factor, for different primary paths with white noise as the reference signal.

at the edge of the constraint, as illustrated in Fig. 3.26 (c) and (d). It is worth noting that all the optimal leak factors of this simulation are derived with (3.102).

In the third simulation, we used the real measured primary and secondary paths of an air duct. To simulate the saturation effect of the ANC system, we also introduced a clipping function cascaded with the control filter that truncates parts of the output signal (above 0.75 or below -0.75). The spectrum and time domain plots of the disturbance signal (a sinusoidal reference $x(n) = 0.32 \sin(0.1\pi n)$ modified by the primary path with the measurement noise $v(n) \sim \mathcal{N}(0, 0.001)$) are shown in Fig. 3.27 (a) and (b), respectively. If the FxLMS algorithm is adopted, the control signal will violate the constraint and result in severe distortion, as illustrated in Fig. 3.27 (c) and (d). In contrast, the LFxLMS algorithm with the optimal leak factor derived with (3.104) can successfully restrain the output signal within the constraint and mitigate the disturbance, as shown in Fig. 3.27 (e) and (f).

FIGURE 3.27: The spectrum of the error signal and the control signal of different algorithms.

3.7 Conclusions

This chapter focus on the effect of saturation on active noise control systems and the approaches that overcome this output nonlinearity. Distortion is a common issue in audio amplifiers, but its effect is seldom studied in ANC systems. Section 3.1 takes the single-frequency feedforward ANC controller as an example and elaborates the generation of additional high-frequency annoyance due to the moderate overdriving and the slightly distorted anti-noise signal. As long as the amplitude of the disturbance is less than the maximum amplitude of the fundamental harmonic in the distorted output of the secondary source, the FxLMS algorithm can still converge. Otherwise, the mild overdriving would lead to the divergence of the FxLMS algorithm.

Section 3.3 revisits some modified adaptive algorithms from the literature that alleviate the output saturation distortion. To further effectively solve this issue,

Section 3.3 and Section 3.5 propose a new adaptive algorithm with an output constraint for practical active noise control systems. The proposed algorithm, the two-gradient direction FxLMS (2GD-FxLMS) algorithm, uses two gradient directions to update the weights of the control filter. When the output signal violates the constraint, the adaptive algorithm switches the weight update direction to reduce the power of the output signal. Compared to other output constraint algorithms, the 2GD-FxLMS algorithm has the same order of computational complexity as the conventional FxLMS algorithm. In addition, it has a stricter output constraint, which can restrict the instantaneous amplitude and average power of the output signal at the same time. Therefore, it can be applied to mobile devices and portable equipment, where the output power of amplifier and computational power of the processor are limited. Moreover, Simulations and real-time implementation on a hardware platform demonstrated its effectiveness in confining the system output under a specific power constraint, over other output constraint algorithms.

Section 3.6 retrospects the optimal leak factor selection issue in the LFxLMS algorithm when it is adopted to avoid the output saturation distortion and to obtain the satisfactory noise reduction performance in active noise control (ANC) application. An optimal leak factor is proposed for the LFxLMS algorithm, which assists the ANC system in achieving the best noise reduction while restraining the average output power within a specified range to prevent saturation distortion. Furthermore, compared to the previous related works, this optimal leak factor does not require any system saturation model and impulse response of the primary path and can be readily calculated in practice. The simulation results based on the real primary and secondary paths verify its performance.

Chapter 4

Virtual sensing feedforward active noise control

Virtual sensing feedforward active noise control technique uses the physical microphone to project the quiet zone to the desired location where it is impractical to place the real error microphone. This chapter reviews the current virtual sensing techniques, extends the virtual sensing technique to multichannel ANC applications, and proves the effectiveness of the multichannel virtual sensing ANC algorithm.

As mentioned in previous sections, the active noise control (ANC) system precisely actuates ‘anti-noise’, which destructively attenuates the disturbance, based on a priori and a posteriori information from reference and error sensors, respectively. It is worth noting that, in most ANC applications, the ‘quiet zone’ is only localized around the monitoring ‘error sensor’ that provides a posteriori information to the adaptive algorithm. The quiet zone is defined as the area on one axial plane where more than 10 dB reduction is attained and its diameter is about a tenth of the acoustic wavelength of the highest frequency in a diffuse field [71]. Hence, the conventional ANC approach where the quiet zone is dependent on the placement of the physical error sensor seems impractical in scenarios where physical placement is inconvenient, such as at the eardrums [72–76] or in domestic living spaces [77].

FIGURE 4.1: A single-channel ANC duct with the virtual microphone.

4.1 Virtual sensing approaches

Many types of virtual sensing ANC (VS-ANC) techniques presently have been proposed to alleviate the restriction on error sensor placement [78]. These approaches effectively generate the quiet zone at desired ‘virtual’ locations, which are away from physical error sensors [79]. The VS-ANC techniques are generally categorized into two classes. Class I techniques do not require offline training and directly estimate the sound characteristics at the virtual location with the physical error microphone based on acoustical models [80, 81] or an extrapolating polynomial [82, 83].

4.1.1 Class I: Virtual signal prediction based on an acoustic model

Figure 4.1 illustrates the block diagram of a single-channel ANC duct. There are two error microphones (physical and virtual microphones) mounted inside the duct. The sound pressure p_a at the physical microphone is given by

$$p_a = p_{pa} + z_a q_s \quad (4.1)$$

where p_{pa} denotes the primary noise at the physical microphone, z_a represents the transfer impedance from the secondary source to the physical microphone position, and q_s denotes the strength of the secondary source. Similarly, the sound pressure p_v of the virtual microphone is expressed as

$$p_v = p_{pv} + z_v q_s \quad (4.2)$$

where p_{pv} denotes the primary noise of the virtual microphone, and z_v represents the transfer impedance from the secondary source to the virtual microphone location. At low frequencies where the wavelength is large compared to the distance, it is reasonable to assume $p_{pa} \approx p_{pv}$ [80]. By subtracting (4.1) by (4.2), we get

$$p_v = p_a + (z_a - z_v) q_s. \quad (4.3)$$

Since the transfer impedance z_a and z_v can be measured in the preliminary stage, and the physical microphone acquires p_a , we can predict the sound pressure p_v at the virtual location through the secondary source strength q_s . On the other hand, we also can derive q_s to minimize p_v so that the ANC system achieves noise control at the virtual location, based on the sound field information picked up by the physical microphone.

Other similar modified methods incorporate accurate acoustic models or the extrapolating polynomial to increase the precision of the prediction. However, the noise reduction performances of these approaches are highly sensitive to the accuracy of the models [80, 84]. Therefore, they are most suitable for low-frequency tonal sound fields [85].

4.1.2 Class II: Preliminary training method

Class II VS-ANC techniques require a preliminary training stage to obtain a filter of the system model from the physical to the virtual error sensor location [86–88] or the information of the optimum noise control filter [89–91]. Subsequently, the pre-trained filter assists the ANC system in achieving satisfactory control of the disturbance at the virtual error sensor location.

Currently, several practical algorithms have been developed to implement class II VS-ANC techniques [86, 92]. The remote microphone technique estimates an ‘observation filter’ from the physical sensor to the desired location by initially placing sensors in situ at both locations [71, 93–97].

Figure 4.2 shows the block diagram of a feedforward ANC algorithm. In the figure, the primary noise vector $\mathbf{v} = [v_1, v_2, \dots, v_{N_v}]^T$ goes through the transfer responses $\mathbf{p}_e$, $\mathbf{p}_m$, and $\mathbf{R}$ and generates the disturbance vector $\mathbf{d}_e = [d_{e1}, d_{e2}, \dots, d_{eN_e}]^T$ at the virtual microphones, the disturbance vector $\mathbf{d}_m = [d_{m1}, d_{m2}, \dots, d_{mN_m}]^T$ at

FIGURE 4.2: Block diagram of the feedforward active control algorithm with the remote microphone

the monitoring microphones, the reference signal vector $\mathbf{u} = [u_1, u_2, \dots, u_{N_u}]^T$, respectively. The error signal vector at the virtual microphones is given by

$$\mathbf{e} = \mathbf{d}_e + \mathbf{G}_e \mathbf{u} = \mathbf{d}_e + \mathbf{G}_e \mathbf{W} \mathbf{x} = \mathbf{P}_e \mathbf{v} + \mathbf{G}_e \mathbf{W} \mathbf{R} \mathbf{v} \quad (4.4)$$

where $\mathbf{W}$ and $\mathbf{x}$ denote the control filter matrix and the reference vector, respectively. $\mathbf{G}_e$ represents the transfer function matrix from the secondary sources to the virtual microphones. Meanwhile, the error signal vector at the monitoring microphones is expressed as

$$\mathbf{m} = \mathbf{d}_m + \mathbf{G}_m \mathbf{u} = \mathbf{d}_m + \mathbf{G}_m \mathbf{W} \mathbf{x} = \mathbf{P}_m \mathbf{v} + \mathbf{G}_m \mathbf{W} \mathbf{R} \mathbf{v} \quad (4.5)$$

where $\mathbf{G}_m$ is the transfer function matrix from the secondary sources to the monitoring microphones. As illustrated in Fig. 4.2, the estimated error signal vector at the virtual microphones is expressed as

$$\hat{\mathbf{e}} = \hat{\mathbf{d}}_e + \hat{\mathbf{G}}_e \mathbf{u} = \hat{\mathbf{O}} \hat{\mathbf{d}}_m + \hat{\mathbf{G}}_e \mathbf{u} = \hat{\mathbf{O}} (\mathbf{m} - \hat{\mathbf{G}}_m \mathbf{u}) + \hat{\mathbf{G}}_e \mathbf{u} \quad (4.6)$$

where $\hat{\mathbf{O}}$ denotes the linear observation filter used to estimate $\hat{\mathbf{d}}_e$ from $\hat{\mathbf{d}}_m$. The optimal linear observation filter is obtained as

$$\mathbf{O}_{\text{opt}} = \mathbf{S}_{\mathbf{d}_m \mathbf{d}_e} (\mathbf{S}_{\mathbf{d}_m \mathbf{d}_m} + \beta \mathbf{I})^{-1} = \mathbf{P}_e \mathbf{S}_{\mathbf{v} \mathbf{v}} \mathbf{P}_m^H (\mathbf{P}_m \mathbf{S}_{\mathbf{v} \mathbf{v}} \mathbf{P}_m^H + \beta \mathbf{I})^{-1} \quad (4.7)$$

where $\mathbf{S}_{\mathbf{d}_m \mathbf{d}_m}$ and $\mathbf{S}_{\mathbf{v}_v}$ denote the spectral density of $\mathbf{d}_m$ and $\mathbf{d}_e$, respectively, and $\mathbf{S}_{\mathbf{d}_m \mathbf{d}_e} = \mathbb{E}(\mathbf{d}_e \mathbf{d}_m)$. To minimize the cost function $J = \text{trace}\left\{\mathbb{E}[\hat{\mathbf{e}}\hat{\mathbf{e}}^H]\right\}$, the optimal control filter is derived as [93]

$$\mathbf{W}_{\text{opt}} = -(\mathbf{G}^H \mathbf{G})^{-1} \mathbf{G}^H \mathbf{O}_{\text{opt}} \mathbf{S}_{\mathbf{x} \mathbf{d}_m} \mathbf{S}_{\mathbf{x} \mathbf{x}}^{-1} \quad (4.8)$$

where $\mathbf{G} = \hat{\mathbf{G}}_e + \hat{\mathbf{O}}(\mathbf{G}_m - \hat{\mathbf{G}}_m)$. $\mathbf{S}_{\mathbf{x} \mathbf{d}_m} = \mathbb{E}(\mathbf{d}_m \mathbf{x}^H)$ denotes the cross spectral density matrix of $\mathbf{d}_m$ and $\mathbf{x}$, and $\mathbf{S}_{\mathbf{x} \mathbf{x}} = \mathbb{E}(\mathbf{x} \mathbf{x}^H)$ represents the power spectral density matrix of $\mathbf{x}$.

From (4.7), it becomes clear that the remote microphone technique requires a strong spatial correlation [98] and causality relation [99] between the physical or monitoring and virtual error sensor locations. Hence, the physical arrangement of both the virtual and physical error sensors are critical to the performance of the remote microphone technique. For instance, if the location of the virtual error sensor is closer to the secondary actuator than the physical error sensor position, the observation filter $\hat{\mathbf{O}}$ will be non-causal. The modified remote sensing technique slackens the causal issue of microphones by introducing several delays into the estimates of the secondary path and observation filter [100]. However, the delay selection process and the magnitude of the measurement error are vital to the estimation accuracy of the observation filter [101]. Moreover, the number of delays somewhat strains the step size bound of the FxLMS algorithm [34].

4.2 Single-channel virtual sensing ANC

There is another virtual sensing ANC (VS-ANC) algorithm that uses an auxiliary filter to achieve the optimal control on the virtual position and overcomes the spatial correlation and causality constraint between error microphones [102–106]. Similar to the remote sensing technique, the VS-ANC technique still consists of two phases: (1) the tuning stage, (2) the control stage.

In the tuning stage, as shown in Fig. 4.3, the control filter output signal is given by

$$y(n) = \mathbf{w}^T(n) \mathbf{x}(n) \quad (4.9)$$

FIGURE 4.3: Block diagram of the single-channel virtual sensing ANC algorithm in the tuning stage.

where $\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{N-1}(n)]^T \in \mathbb{R}^{N \times 1}$ is the weight vector of the control filter, and $\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-N+1)]^T \in \mathbb{R}^{N \times 1}$ denotes the reference signal vector; $\mathbb{R}$ indicates real-numbers. Hence, the error signal at the virtual microphone is given by

$$e_v(n) = d_v(n) - g_v(n) * y(n) \quad (4.10)$$

where $d_v(n)$ and $g_v(n)$ represent the disturbance on the virtual position and the impulse response of the virtual secondary path, respectively. To minimize the square of (4.10), we use the weight update formula of the FxLMS algorithm as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu_1 e_v(n) \mathbf{x}'_v(n) \quad (4.11)$$

FIGURE 4.4: Block diagram of the single-channel virtual sensing ANC algorithm in the control stage.

where μ_1 denotes the step size, and the filtered reference vector is given by

$$\mathbf{x}'_v(n) = \hat{g}_v(n) * \mathbf{x}(n) \quad (4.12)$$

Meanwhile, the output signal of the auxiliary filter is written as

$$y_h(n) = \mathbf{h}^T(n) \mathbf{x}(n) \quad (4.13)$$

where $\mathbf{h}(n) = [h_0(n), h_1(n), \dots, h_{N-1}(n)]^T$ is the impulse response of the auxiliary filter at time n . The error signal $e_h(n)$, as shown in Fig. 4.3, is expressed as

$$e_h = e_p(n) - y_h(n) \quad (4.14)$$

where $e_p(n)$ is the error signal picked up by the physical microphone. Based on LMS algorithm, the update equation for the auxiliary filter is given by

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu_2 e_h(n) \mathbf{x}(n) \quad (4.15)$$

where μ_2 denotes the step size of the LMS algorithm.

In the control stage, the virtual microphone is removed. In this case, the optimal auxiliary filter $H_o(z)$, whose impulse response $\mathbf{h}_o$ is obtained from (4.15), is used to assist the VS-ANC in achieving optimal control at the virtual position, as shown in Fig. 4.4. At present, the error signal $e_h(n)$ is derived as

$$e_h(n) = e_p(n) - \mathbf{h}_o^T \mathbf{x}(n). \quad (4.16)$$

According to the FxLMS algorithm, the control filter in the control stage is obtained by

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu_3 e_h(n) \mathbf{x}'(n) \quad (4.17)$$

where μ_3 denotes the step size of the FxLMS algorithm. In the control stage, the filtered signal is given by

$$\mathbf{x}'(n) = \hat{g}(n) * \mathbf{x}(n) \quad (4.18)$$

where $\hat{g}(n)$ represents the impulse response of the secondary path estimate, as shown in Fig. 4.4.

4.3 Multichannel virtual sensing ANC

To obtain the larger quiet zone at the desired location, we extend VS technique from single-channel to multichannel ANC application. In this section, a generic multichannel ANC system with J reference microphones, K secondary sources, and M physical error microphones, is adopted, as shown in Fig. 4.5. Additionally, there are N_p independent primary sources placed ahead of the reference microphones. The multichannel virtual sensing ANC (MVANC) technique is applied to this multichannel system to cancel the disturbances at N_v virtual microphone locations in the far field. For brevity and without loss of generality, the number of reference microphones is assumed to be greater than the primary sources ($N_p < J$).

In the tuning stage, as shown in Fig. 4.6, the output signal vector is given by

$$\mathbf{y}(n) = \mathbf{w}^T(n) \mathbf{x}(n) \in \mathbb{R}^{K \times 1} \quad (4.19)$$

FIGURE 4.5: A simplified schematic of the MVANC algorithm.

where $\mathbf{y}(n) = [y_1(n), y_2(n), \dots, y_k(n), \dots, y_K(n)]^T$, and $y_k(n)$ denotes the k th output signal at time sample n . The stacked reference vector $\mathbf{x}(n)$ is expressed as

$$\mathbf{x}(n) = [\mathbf{x}_1^T(n), \mathbf{x}_2^T(n), \dots, \mathbf{x}_J^T(n)]^T \in \mathbb{R}^{JN \times 1} \quad (4.20)$$

where $\mathbf{x}_j(n) \in \mathbb{R}^{N \times 1}$ is the j th reference vector with N samples and given by

$$\mathbf{x}_j(n) = [x_j(n), x_j(n-1), \dots, x_j(n-N+1)]^T. \quad (4.21)$$

The control filter matrix $\mathbf{w}(n) \in \mathbb{R}^{JN \times K}$ in (4.19) is given by

$$\mathbf{w}(n) = \begin{bmatrix} \mathbf{w}_{11}(n) & \mathbf{w}_{21}(n) & \cdots & \mathbf{w}_{K1}(n) \\ \mathbf{w}_{12}(n) & \mathbf{w}_{22}(n) & \cdots & \mathbf{w}_{K2}(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{w}_{1J}(n) & \mathbf{w}_{2J}(n) & \cdots & \mathbf{w}_{KJ}(n) \end{bmatrix} \quad (4.22)$$

where the $\mathbf{w}_{kj}(n) \in \mathbb{R}^{N \times 1}$ represents the coefficient vector of the control filter from the j th input to the k th output,

$$\mathbf{w}_{kj}(n) = [w_{jk,0}(n), w_{jk,1}(n), \dots, w_{jk,N-1}(n)]^T. \quad (4.23)$$

FIGURE 4.6: Training the control filter in the tuning stage of the MVANC technique.

Based on the conventional multichannel FxLMS (MCFxLMS) algorithm, the new control filter from the j th input to the k th output is defined as

$$\mathbf{w}_{kj}(n) = [w_{jk,0}(n), w_{jk,1}(n), \dots, w_{jk,N-1}(n)]^T. \quad (4.24)$$

Similarly, the new control filter from the j th input to the k th output is defined as

$$\mathbf{w}_{kj}(n+1) = \mathbf{w}_{kj}(n) - \mu_1 \sum_{i=1}^{N_v} \mathbf{x}'_{v,jki}(n) e_{v,i}(n) \quad (4.25)$$

where $e_{v,i}(n)$ represents the error signal at the i th virtual microphone, and μ_1 denotes the step size of FxLMS algorithm. The filtered reference $x'_{v,jki}(n)$ is a convolution between the j th reference signal $x_j(n)$ and the virtual secondary path estimate $\hat{g}_{v,ik}(n)$, which is from the k th secondary source to the i th virtual microphone. From (4.25), it can be seen that the final control filter in the tuning stage converges to the same optimal control filter obtained with the conventional MCFxLMS [9].

Once the control filter converges, the auxiliary filters are trained, as illustrated in Fig. 4.7. Utilizing the least mean square (LMS) algorithm [107], the m th new auxiliary filter vector becomes

$$\mathbf{h}_m(n+1) = \mathbf{h}_m(n) + \mu_2 e_{h,m}(n) \mathbf{x}(n) \in \mathbb{R}^{JN \times 1} \quad (4.26)$$

FIGURE 4.7: Training the auxiliary filter in the tuning stage of the MVANC technique.

where μ_2 denotes the step size of LMS algorithm and

$$\mathbf{h}_m(n) = [\mathbf{h}_{m1}^T(n), \mathbf{h}_{m2}^T(n), \dots, \mathbf{h}_{mj}^T(n), \dots, \mathbf{h}_{mJ}^T(n)]^T.$$

The expression $\mathbf{h}_{mj}(n) \in \mathbb{R}^{N \times 1}$ denotes the auxiliary filter from the j th reference to the m th physical microphone, with a length of N taps. Therefore,

$$\mathbf{h}_{mj}(n) = [h_{mj,0}(n), h_{mj,1}(n), \dots, h_{mj,N-1}(n)]^T \quad (4.27)$$

The expression $e_{h,m}(n)$ is stated as

$$e_{h,m}(n) = e_{p,m}(n) - \mathbf{h}_m^T(n) \mathbf{x}(n) \quad (4.28)$$

where $e_{p,m}(n)$ is the error signal at the m th physical microphone.

In the control stage, the microphones at the virtual locations are removed, while the physical error microphones are untouched. As depicted in Fig. 4.8, the new control filter is given by

$$\mathbf{w}_{kj}(n+1) = \mathbf{w}_{kj}(n) - \mu_3 \sum_{m=1}^M \mathbf{x}'_{p,jkm}(n) e_{h,m}(n) \quad (4.29)$$

FIGURE 4.8: Training the control filter in the control stage of the MVANC technique.

where $x'_{p,jkm}(n)$ is the product of the convolution between the secondary path estimate $\hat{g}_{mk}(n)$, which is from the k th secondary source to the m th physical microphone, and the j th reference signal $x_j(n)$. The symbol μ_3 denotes the step size of FxLMS algorithm, and the error signal $e_{h,m}(n)$ in (4.29) is given by

$$e_{h,m}(n) = e_{p,m}(n) - \mathbf{h}_{o,m}^T \mathbf{x}(n), \quad (4.30)$$

where $\mathbf{h}_{o,m}$ represents the m th optimal auxiliary filter obtained from (4.27).

4.4 Analysis on multichannel virtual sensing ANC

This section exhibits the analysis of the multichannel virtual sensing ANC in the time and frequency domains. These theoretical investigations demonstrate the effectiveness of the MVANC algorithm.

4.4.1 Frequency domain analysis of the MVANC technique

This analysis initially assumes that there are fewer secondary sources than both the physical and virtual microphones ($K \leq N_v, K \leq M$). The influence of alternate

FIGURE 4.9: Block diagram of the tuning stage in the MVANC technique, where Σ denotes acoustic summation and $+$ denotes digital summation.

configurations on the optimal control in both the tuning and control stages are analyzed in the following sub-sections.

4.4.1.1 Tuning stage

In the tuning stage, the primary noise is defined as a random noise $\mathbf{V}_1 \in \mathbb{C}^{N_p \times 1}$, where $\mathbb{C}^{N_p \times 1}$ denotes a $N_p \times 1$ vector of complex values, and comes from N_p individual primary sources. The reference signal $\mathbf{X}_1 \in \mathbb{C}^{J \times 1}$ is modified by the control filter $\mathbf{W} \in \mathbb{C}^{K \times J}$ and the virtual secondary path $\mathbf{G}_v \in \mathbb{C}^{N_v \times K}$ to cancel the disturbance $\mathbf{D}_{v1} \in \mathbb{C}^{N_v \times 1}$ at the virtual positions, as shown in Fig. 4.9. The virtual error signal vector is thus given by

$$\mathbf{E}_{v1} = \mathbf{D}_{v1} + \mathbf{G}_v \mathbf{W} \mathbf{X}_1 \in \mathbb{C}^{N_v \times 1}. \quad (4.31)$$

The virtual secondary path matrix is given by

$$\mathbf{G}_v = \begin{bmatrix} G_{v,11} & G_{v,12} & \cdots & G_{v,1K} \\ G_{v,21} & G_{v,22} & \cdots & G_{v,2K} \\ \vdots & \vdots & \ddots & \vdots \\ G_{v,N_v1} & G_{v,N_v2} & \cdots & G_{v,N_vK} \end{bmatrix}_{N_v \times K} \quad (4.32)$$

where $G_{v,nk}$ represents the transfer function of the secondary path from the k th secondary source to the n th virtual microphone. $\mathbf{G}_v$ is assumed to be a full-rank matrix, and hence, $\mathbf{G}_v^H \mathbf{G}_v \in \mathbb{C}^{K \times K}$ is invertible.

The power spectral density of the virtual error signal can be expressed as

$$J_{v1} = \mathbb{E}(\mathbf{E}_{v1}^H \mathbf{E}_{v1}) = \text{tr}[\mathbb{E}(\mathbf{E}_{v1} \mathbf{E}_{v1}^H)] \quad (4.33)$$

where $\text{tr}[\cdot]$ denotes the trace operator. Expanding (4.33) yields

$$J_v = \text{tr}[\mathbb{E}(\mathbf{D}_{v1} \mathbf{D}_{v1}^H + \mathbf{D}_{v1} \mathbf{X}_1^H \mathbf{W}^H \mathbf{G}_v^H + \mathbf{G}_v \mathbf{W} \mathbf{X}_1 \mathbf{D}_v^H + \mathbf{G}_v \mathbf{W} \mathbf{X}_1 \mathbf{X}_1^H \mathbf{W}^H \mathbf{G}_v^H)]. \quad (4.34)$$

By invoking the properties of the trace operator whereby $\text{tr}[\mathbf{A} + \mathbf{B}] = \text{tr}[\mathbf{A}] + \text{tr}[\mathbf{B}]$ and $\text{tr}[\mathbf{AB}] = \text{tr}[\mathbf{BA}]$, (4.34) is rewritten as

$$J_v = \text{tr}(\mathbf{S}_{d,v1} + \mathbf{W}^H \mathbf{G}_v^H \mathbf{S}_{xd,v1} + \mathbf{S}_{dx,v1} \mathbf{G}_v \mathbf{W} + \mathbf{W}^H \mathbf{G}_v^H \mathbf{G}_v \mathbf{W} \mathbf{S}_{x1}) \quad (4.35)$$

where the power spectral density matrix for the reference signals, the virtual disturbance signals and the cross-spectral density matrix between the reference signals and the virtual disturbance signals are defined as

$$\mathbf{S}_{x1} = \mathbb{E}(\mathbf{X}_1 \mathbf{X}_1^H) \in \mathbb{C}^{J \times J} \quad (4.36)$$

$$\mathbf{S}_{d,v1} = \mathbb{E}(\mathbf{D}_{v1} \mathbf{D}_{v1}^H) \in \mathbb{C}^{N_v \times N_v} \quad (4.37)$$

and

$$\mathbf{S}_{xd,v1} = \mathbb{E}(\mathbf{D}_{v1} \mathbf{X}_1^H) \in \mathbb{C}^{N_v \times J}. \quad (4.38)$$

To obtain the optimal solution of the control filter by minimizing the cost function (4.35), we set its gradient to zero as

$$\frac{\partial J_v}{\partial \mathbf{W}^*} = 2\mathbf{S}_{xd,v1}^H \mathbf{G}_v + 2\mathbf{S}_{x1}^H \mathbf{W}^H \mathbf{G}_v^H \mathbf{G}_v = 0. \quad (4.39)$$

By solving (4.39), the optimal control filter at the virtual positions is given by

$$\mathbf{W}_{o,t} = -(\mathbf{G}_v^H \mathbf{G}_v)^{-1} \mathbf{G}_v^H \mathbf{S}_{\text{xd},v1} \mathbf{S}_{x1}^{-1} \in \mathbb{C}^{K \times 1} \quad (4.40)$$

where $\mathbf{S}_{x1}$ is assumed to be invertible [71]. Due to the random primary disturbance $\mathbf{V}_1$ and the reference signal $\mathbf{X}_1$ are modified by $\mathbf{P}_v \in \mathbb{C}^{N_v \times N_p}$ and $\mathbf{R} \in \mathbb{C}^{J \times N_p}$ respectively, as shown in Fig. 4.9, the power spectral density matrix $\mathbf{S}_{x1}$ and the cross-spectral density matrix $\mathbf{S}_{\text{xd},v1}$ are expressed as

$$\mathbf{S}_{x1} = \mathbf{R} \mathbf{S}_{v1} \mathbf{R}^H \quad (4.41)$$

and

$$\mathbf{S}_{\text{xd},v1} = \mathbf{P}_v \mathbf{S}_{v1} \mathbf{R}^H \quad (4.42)$$

where $\mathbf{S}_{v1} = \mathbb{E}(\mathbf{V}_1 \mathbf{V}_1^H) \in \mathbb{C}^{N_p \times N_p}$ denotes the spectral density matrix of the random noise source and is invertible. $\mathbf{P}_v$ denotes the virtual primary path matrix and is given by

$$\mathbf{P}_v = \begin{bmatrix} P_{11,v} & P_{12,v} & \cdots & P_{1L,v} \\ P_{21,v} & P_{22,v} & \cdots & P_{2L,v} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N_v 1,v} & P_{N_v 2,v} & \cdots & P_{N_v L,v} \end{bmatrix}_{N_v \times N_p} \quad (4.43)$$

where $P_{nl,v}$ denotes the transfer function of the primary path from the l th primary source to the n th virtual microphone. $\mathbf{R}$ denotes the reference filter matrix and is given by

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1L} \\ R_{21} & R_{22} & \cdots & R_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ R_{J1} & R_{J2} & \cdots & R_{JL} \end{bmatrix}_{J \times N_p} \quad (4.44)$$

where R_{jl} stands for the transfer function from the l th primary source to the j th reference microphone. $\mathbf{R}$ is assumed to be a full-rank matrix, and hence, $[\mathbf{R}^H \mathbf{R}] \in \mathbb{C}^{N_p \times N_p}$ is invertible.

By substituting both (4.41) and (4.42) into (4.40), the optimal control filter in tuning stage is rewritten as

$$\begin{aligned}
\mathbf{W}_{o,t} &= -(\mathbf{G}_v^H \mathbf{G}_v)^{-1} \mathbf{G}_v^H \mathbf{P}_v \mathbf{S}_{v1} \mathbf{R}^H (\mathbf{R} \mathbf{S}_{v1} \mathbf{R}^H)^{-1} \\
&= -(\mathbf{G}_v^H \mathbf{G}_v)^{-1} \mathbf{G}_v^H \mathbf{P}_v \times \mathbf{I}_{N_p} \times \mathbf{S}_{v1} \mathbf{R}^H (\mathbf{R} \mathbf{S}_{v1} \mathbf{R}^H)^{-1} \\
&= -(\mathbf{G}_v^H \mathbf{G}_v)^{-1} \mathbf{G}_v^H \mathbf{P}_v \times (\mathbf{R}^H \mathbf{R})^{-1} \mathbf{R}^H \mathbf{R} \times \mathbf{S}_{v1} \mathbf{R}^H (\mathbf{R} \mathbf{S}_{v1} \mathbf{R}^H)^{-1} \\
&= -(\mathbf{G}_v^H \mathbf{G}_v)^{-1} \mathbf{G}_v^H \mathbf{P}_v (\mathbf{R}^H \mathbf{R})^{-1} \mathbf{R}^H.
\end{aligned} \tag{4.45}$$

From (4.45), it is evident that the optimal control filter in the MVANC technique only relies on the reference path $\mathbf{R}$, the primary path $\mathbf{P}_v$, and the secondary path $\mathbf{G}_v$. Furthermore, (4.45) is also the optimal solution of the conventional MCFxLMS algorithm, which achieves optimal control at the error microphones placed at the virtual positions.

The error signal that monitors the performance of the auxiliary filters in the tuning stage, as shown in Fig. 4.9, is given by

$$\mathbf{E}_{h1} = \mathbf{E}_{p1} - \mathbf{Y}_{h1} = \mathbf{D}_{p1} + \mathbf{G} \mathbf{W}_{o,t} \mathbf{X}_1 - \mathbf{H} \mathbf{X}_1 \tag{4.46}$$

where $\mathbf{E}_{p1} \in \mathbb{C}^{M \times 1}$ and $\mathbf{D}_{p1} \in \mathbb{C}^{M \times 1}$ represent the error signal and the disturbance on the physical microphones, respectively; $\mathbf{Y}_{h1} \in \mathbb{C}^{M \times 1}$ denotes the output signal of the auxiliary filters; $\mathbf{H} \in \mathbb{C}^{M \times J}$ denotes the auxiliary filter matrix, and $\mathbf{G}$ is a matrix of the physical secondary paths given by

$$\mathbf{G} = \begin{bmatrix} G_{11} & G_{12} & \cdots & G_{1k} \\ G_{21} & G_{22} & \cdots & G_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ G_{M1} & G_{M2} & \cdots & G_{MK} \end{bmatrix}_{M \times K} \tag{4.47}$$

where G_{mk} represents the transfer function of the secondary path from the k th secondary source to the m th physical microphone. $\mathbf{G}$ is assumed to be a full-rank matrix, and hence, $\mathbf{G}^H \mathbf{G} \in \mathbb{C}^{K \times K}$ is invertible. The power spectral density of (4.46) is expressed as

$$J_{h1} = \mathbb{E}(\mathbf{E}_{h1}^H \mathbf{E}_{h1}) = \text{tr}[\mathbb{E}(\mathbf{E}_{h1} \mathbf{E}_{h1}^H)]. \tag{4.48}$$

Expanding (4.48) yields

$$\begin{aligned} J_{h1} = \text{tr} \big[& \mathbb{E}(\mathbf{D}_{p1} \mathbf{D}_{p1}^H + \mathbf{D}_{p1} \mathbf{X}_1^H \mathbf{W}_{o,t}^H \mathbf{G}^H - \mathbf{D}_{p1} \mathbf{X}_1^H \mathbf{H}^H + \mathbf{G} \mathbf{W}_{o,t} \mathbf{X}_1 \mathbf{D}_{p1}^H + \\ & \mathbf{G} \mathbf{W}_{o,t} \mathbf{X}_1 \mathbf{X}_1^H \mathbf{W}_{o,t}^H \mathbf{G}^H - \mathbf{G} \mathbf{W}_{o,t} \mathbf{X}_1 \mathbf{X}_1^H \mathbf{H}^H - \mathbf{H} \mathbf{X}_1 \mathbf{D}_{p1}^H \\ & - \mathbf{H} \mathbf{X}_1 \mathbf{X}_1^H \mathbf{W}_{o,t}^H \mathbf{G}^H + \mathbf{H} \mathbf{X}_1 \mathbf{X}_1^H \mathbf{H}^H) \big]. \end{aligned} \quad (4.49)$$

By invoking the properties of the trace operator, (4.49) is rewritten as

$$\begin{aligned} J_{h1} = & \text{tr}(\mathbf{S}_{d,p1} + \mathbf{S}_{xd,p1} \mathbf{W}_{o,t}^H \mathbf{G}^H - \mathbf{H}^H \mathbf{S}_{xd,p1} + \mathbf{G} \mathbf{W}_{o,t} \mathbf{S}_{dx,p1} + \\ & \mathbf{G} \mathbf{W}_{o,t} \mathbf{S}_{x1} \mathbf{W}_{o,t}^H \mathbf{G}^H - \mathbf{H}^H \times \mathbf{G} \mathbf{W}_{o,t} \mathbf{S}_{x1} - \mathbf{H} \mathbf{S}_{dx,p1} - \mathbf{H} \mathbf{S}_{x1} \mathbf{W}_{o,t}^H \mathbf{G}^H + \\ & \mathbf{H}^H \mathbf{H} \mathbf{S}_{x1}) \end{aligned} \quad (4.50)$$

where the cross spectral density matrix $\mathbf{S}_{xd,p1}$ is given by

$$\mathbf{S}_{xd,p1} = \mathbb{E}(\mathbf{D}_{p1} \mathbf{X}_1^H) \in \mathbb{C}^{M \times J}. \quad (4.51)$$

To obtain the optimal auxiliary filter $\mathbf{H}_o \in \mathbb{C}^{M \times J}$, gradient of (4.50) is set to zero to give

$$\frac{\partial J_{h1}}{\partial \mathbf{H}^*} = -2\mathbf{S}_{xd,p1}^H - 2\mathbf{S}_{x1}^H \mathbf{W}_{o,t}^H \mathbf{G}^H + 2\mathbf{S}_{x1}^H \mathbf{H}_o^H = 0. \quad (4.52)$$

As the disturbance $\mathbf{D}_{p1}$ at the location of the physical microphone is the result of random noise $\mathbf{V}_1$ propagating through the physical primary paths $\mathbf{P} \in \mathbb{C}^{M \times J}$, as illustrated in Fig. 4.9, the cross spectral density matrix $\mathbf{S}_{xd,p1}$ can be rewritten as

$$\mathbf{S}_{xd,p1} = \mathbf{P} \mathbf{S}_{v1} \mathbf{R}^H. \quad (4.53)$$

The matrix of primary paths $\mathbf{P}$ is given by

$$\mathbf{P} = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1L} \\ P_{21} & P_{22} & \cdots & P_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ P_{M1} & P_{M2} & \cdots & P_{ML} \end{bmatrix}_{M \times N_p}, \quad (4.54)$$

where P_{ml} denotes the transfer function of the primary path from the l th primary source to the m th physical microphone. By substituting (4.45) and (4.53) into

FIGURE 4.10: Block diagram of the control stage in the MVANC technique.

(4.52), the optimal auxiliary filter is derived as

$$\begin{aligned}
 \mathbf{H}_o &= \mathbf{P} \mathbf{S}_{v1} \mathbf{R}^H (\mathbf{R} \mathbf{S}_{v1} \mathbf{R}^H)^{-1} + \mathbf{G} \mathbf{W}_{o,t} \\
 &= \mathbf{P} (\mathbf{R}^H \mathbf{R})^{-1} \mathbf{R}^H \mathbf{R} \mathbf{S}_{v1} \mathbf{R}^H (\mathbf{R} \mathbf{S}_{v1} \mathbf{R}^H)^{-1} - \mathbf{G} (\mathbf{G}_v^H \mathbf{G}_v)^{-1} \\
 &\quad \times \mathbf{G}_v^H \mathbf{P}_v (\mathbf{R}^H \mathbf{R})^{-1} \mathbf{R}^H \\
 &= \left[\mathbf{P} - \mathbf{G} (\mathbf{G}_v^H \mathbf{G}_v)^{-1} \mathbf{G}_v^H \mathbf{P}_v \right] (\mathbf{R}^H \mathbf{R})^{-1} \mathbf{R}^H.
 \end{aligned} \tag{4.55}$$

4.4.1.2 Control stage

In the control stage, a different random noise $\mathbf{V}_2 \in \mathbb{C}^{N_p \times 1}$ is assigned as the primary noise, which also comes from N_p individual noise sources. Since the optimal auxiliary filters assist to update control filters, as shown in Fig. 4.10, the error signal vector $\mathbf{E}_{h2} \in \mathbb{C}^{M \times 1}$ is expressed as

$$\mathbf{E}_{h2} = \mathbf{E}_{p2} - \mathbf{Y}_{h2} = \mathbf{D}_{p2} + \mathbf{G} \mathbf{W} \mathbf{X}_2 - \mathbf{H}_o \mathbf{X}_2 \tag{4.56}$$

where $\mathbf{E}_{p2} \in \mathbb{C}^{M \times 1}$ and $\mathbf{D}_{p2} \in \mathbb{C}^{M \times 1}$ denote the error signal and disturbance at the physical microphones, respectively. The power spectral density of (4.56) is defined

as

$$J_{h2} = \text{tr}(\mathbf{S}_{d,p2} + \mathbf{W}^H \mathbf{G}^H \mathbf{S}_{xd,p2} - \mathbf{S}_{xd,p2} \mathbf{H}_o^H + \mathbf{G} \mathbf{W} \mathbf{S}_{dx,p2} + \mathbf{W}^H \mathbf{G}^H \mathbf{G} \mathbf{W} \mathbf{S}_{x2} - \mathbf{G} \mathbf{W} \mathbf{S}_{x2} \mathbf{H}_o^H - \mathbf{H}_o \mathbf{S}_{dx,p2} - \mathbf{W}^H \mathbf{G}^H \mathbf{H}_o \mathbf{S}_{x2} + \mathbf{H} \mathbf{S}_{x2} \mathbf{H}_o^H). \quad (4.57)$$

To obtain the optimal control filter, the gradient of (4.57) is set to zero to give

$$\frac{\partial J_{h2}}{\partial \mathbf{W}^*} = 2\mathbf{S}_{xd,p2}^H \mathbf{G} + 2\mathbf{S}_{x2}^H \mathbf{W}^H \mathbf{G}^H \mathbf{G} - 2\mathbf{S}_{x2}^H \mathbf{H}_o^H \mathbf{G} = 0 \quad (4.58)$$

where the power spectral density matrix $\mathbf{S}_{x2}$ and the cross spectral density matrix $\mathbf{S}_{xd,p2}$ are given by

$$\mathbf{S}_{x2} = \mathbf{R} \mathbf{S}_{v2} \mathbf{R}^H \in \mathbb{C}^{J \times J} \quad (4.59)$$

and

$$\mathbf{S}_{xd,p2} = \mathbf{P} \mathbf{S}_{v2} \mathbf{R}^H \in \mathbb{C}^{M \times J}. \quad (4.60)$$

Where $\mathbf{S}_{v2} = \mathbb{E}(\mathbf{V}_2 \mathbf{V}_2^H) \in \mathbb{C}^{N_p \times N_p}$ denotes the cross spectral density matrix of primary noise $\mathbf{V}_2$. By solving (4.58), the optimal solution $\mathbf{W}_{o,c} \in \mathbb{C}^{K \times J}$ of the control filter for the control stage is given by

$$\mathbf{W}_{o,c} = (\mathbf{G}^H \mathbf{G})^{-1} (\mathbf{G}^H \mathbf{H}_o \mathbf{S}_{x2} - \mathbf{G}^H \mathbf{S}_{xd,p2}) \mathbf{S}_{x2}^{-1} \quad (4.61)$$

where $\mathbf{S}_{x2}$ is assumed to be invertible [71]. By substituting (4.55), (4.59), and (4.60) into (4.61) yields

$$\mathbf{W}_{o,c} = \mathbf{W}_{o,t} = -(\mathbf{G}_v^H \mathbf{G}_v)^{-1} \mathbf{G}_v^H \mathbf{P}_v (\mathbf{R}^H \mathbf{R})^{-1} \mathbf{R}^H. \quad (4.62)$$

Therefore, in both the tuning and control stages, the control filter converges to the same optimal solution in (4.45) and (4.62), respectively.

It is worth noting that the auxiliary filter is essential to obtain the optimal control filter in the control stage of the MVANC technique. Furthermore, the final auxiliary filter, as defined in (4.55), is composed of the primary path and the virtual primary path transfer functions, which are independently defined by the positions of the physical microphone and the virtual microphone, respectively. Furthermore, the optimal control filter of MVANC, as defined in (4.61), does not contain any transfer function about the physical microphone location. Therefore, compared to

the conventional virtual sensing active control methods [86, 98], the MVANC technique does not require the causal transfer function from the physical microphone and the virtual microphone and has more flexibility in the selection of the locations of virtual and physical microphones.

4.4.2 Time domain analysis on the MVANC technique

The above analysis based on the frequency domain proves that VS-ANC achieves the optimal noise control at the virtual positions. However, it does not enforce the causal requirement on the filter. Hence, this section carries out the time domain analysis on MCANC to verify its effectiveness.

The primary noise is set to be $\mathbf{V}$ in this section. The final virtual error signal in the tuning stage is given by

$$\mathbf{e}_v(n) = \mathbf{d}_v(n) + \mathbf{X}'_v(n)\mathbf{w}_{o,t} \quad (4.63)$$

where $\mathbf{w}_{o,t}$ denotes the optimal control filter vector in the tuning stage. $\mathbf{X}'_v(n)$ represents the filtered reference signal matrix as

$$\mathbf{X}'_v(n) = \begin{bmatrix} \mathbf{r}_{v,11}^T(n) & \mathbf{r}_{v,21}^T(n) & \cdots & \mathbf{r}_{v,K1}^T(n) \\ \mathbf{r}_{v,12}^T(n) & \mathbf{r}_{v,22}^T(n) & \cdots & \mathbf{r}_{v,K2}^T(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{r}_{v,1N_v}^T(n) & \mathbf{r}_{v,2N_v}^T(n) & \cdots & \mathbf{r}_{v,KN_v}^T(n) \end{bmatrix} \quad (4.64)$$

where

$$\mathbf{r}_{v,ki}^T = [\mathbf{x}'_{v,1k}(n)^T, \mathbf{x}'_{v,2k}(n)^T, \cdots, \mathbf{x}'_{v,Jki}(n)^T]. \quad (4.65)$$

$\mathbf{x}'_{v,jki}(n) = [x_{v,jki}(n), x_{v,jki}(n-1), \cdots, x_{v,jki}(n-N+1)]^T$ denotes the j th filtered reference vector modelled by the secondary path estimate from the k th secondary source to the i th virtual microphone.

The cost function in the tuning stage is defined as

$$J_v = \mathbb{E} [\mathbf{e}_v^T(n)\mathbf{e}_v(n)]. \quad (4.66)$$

The optimal control filter that minimizes the cost function is derived as

$$\mathbf{w}_{o,t} = - \left\{ \mathbb{E} [\mathbf{X}'_v(n)^T \mathbf{X}'_v(n)] \right\}^{-1} \mathbb{E} [\mathbf{X}'_v(n)^T \mathbf{d}_v(n)] . \quad (4.67)$$

Meanwhile, the signal vector $\mathbf{e}_h(n)$ is given by

$$\mathbf{e}_h(n) = \mathbf{d}_p(n) + \mathbf{X}'_p(n) \mathbf{w}_{o,t} - \mathbf{X}(n) \mathbf{h}(n) \quad (4.68)$$

where $\mathbf{X}'_p(n)$ represents the filtered reference signal matrix modelled by the physical secondary path estimates and has the similar format as (4.64). $\mathbf{X}(n)$ denotes the reference matrix, and $\mathbf{h}(n)$ stands for the stacked vector of the auxiliary filters given by

$$\mathbf{h}(n) = [\mathbf{h}_{11}^T(n), \dots, \mathbf{h}_{m1}^T(n), \mathbf{h}_{12}^T(n), \dots, \mathbf{h}_{m2}^T(n), \dots, \mathbf{h}_{MJ}^T(n)]^T \quad (4.69)$$

where $\mathbf{h}_{mj}^T(n)$ denotes the impulse response of the auxiliary filter from the j th reference to the m th physical microphone. By minimizing the mean square sum of (4.69), the optimal auxiliary filters is derived as

$$\mathbf{h}_o = \left\{ \mathbb{E} [\mathbf{X}^T(n) \mathbf{X}(n)] \right\}^{-1} \mathbb{E} [\mathbf{X}^T(n) \mathbf{d}_p(n) + \mathbf{X}^T(n) \mathbf{X}'_p(n) \mathbf{w}_{o,t}] . \quad (4.70)$$

Substituting (4.70) into (4.68) yields

$$\mathbf{e}_{o,h}(n) = \mathbf{d}_p(n) + \mathbf{X}'_p(n) \mathbf{w}_{o,t} - \mathbf{X}(n) \mathbf{h}_o. \quad (4.71)$$

In the control stage, the physical error signal is given by

$$\mathbf{e}_p(n) = \mathbf{d}_p(n) + \mathbf{X}'_p(n) \mathbf{w}(n). \quad (4.72)$$

Hence, as shown in Fig. 4.7,

$$\mathbf{e}_h(n) = \mathbf{e}_p(n) - \mathbf{X}(n) \mathbf{h}_o. \quad (4.73)$$

By minimizing the mean square sum of (4.73), the optimal control filter of the control stage is derived as

$$\mathbf{w}_{o,c} = - \left\{ \mathbb{E} [\mathbf{X}'_p(n)^T \mathbf{X}'_p(n)] \right\}^{-1} \mathbb{E} [\mathbf{X}'_p(n)^T \mathbf{d}_p(n) - \mathbf{X}'_p(n)^T \mathbf{X}(n) \mathbf{h}_o] . \quad (4.74)$$

TABLE 4.1: System characteristics of alternative sensor-actuator configurations in the tuning and control stages. Here, K , N_v , and M denote the number of secondary sources, virtual microphones, and physical microphones, respectively.

No.	Tuning stage	Control stage
Case I	$K \leq N_v$ Over- or fully-determined	$K \leq M$ Over- or fully-determined
Case II	$K > N_v$ Under-determined	$K \leq M$ Over- or fully-determined
Case III	$K \leq N_v$ Over- or fully-determined	$K > M$ Under-determined
Case IV	$K > N_v$ Under-determined	$K > M$ Under-determined

Substituting (4.71) into (4.74) yields

$$\mathbf{w}_{o,c} = - \left\{ \mathbb{E} [\mathbf{X}'_p(n)^T \mathbf{X}'_p(n)] \right\}^{-1} \mathbb{E} [\mathbf{X}'_p(n)^T \mathbf{e}_{o,h}(n) - \mathbf{X}'_p(n)^T \mathbf{X}'_p(n) \mathbf{w}_{o,t}] . \quad (4.75)$$

Since the steady-state error $\mathbf{e}_{o,h}(n)$ is orthogonal to the reference signal $\mathbf{X}(n)$, and the filtered reference matrix $\mathbf{x}'_p(n)$ is the linear combination of the reference signal $\mathbf{X}(n)$,

$$\mathbb{E} [\mathbf{X}'_p(n) \mathbf{e}_{o,h}(n)] = 0 \quad (4.76)$$

By combining (4.75) and (4.76), the optimal control filter in the control stage is rewritten as

$$\mathbf{w}_{o,c} = \mathbf{w}_{o,t}. \quad (4.77)$$

Therefore, MVANC achieves the same optimal control filter as the conventional MCANC, which utilizes the real error microphones at the virtual position.

4.5 Effect of alternative sensor-actuator configurations

The optimal control filter in the previous section has been derived on the pretext of an over- or fully-determined system setup, where there are more or equal number of sensors than actuators. Although the effect of under-determined formulations on the optimal solution have been defined for multichannel ANC systems, it is not directly applicable to the MVANC technique due to the presence of the

tuning stage. In the interest of minimizing the computational complexity, it is, therefore, worthwhile to investigate the influence of a partially under-determined implementation within the MVANC technique. Hence, four alternative configurations considering various combinations of virtual and physical sensors are analyzed, as listed in Table 4.1.

In this section, the primary noise in both the tuning stage and control stage is denoted by random noise vector $\mathbf{V}$.

To achieve optimal attenuation of the disturbance at the virtual locations, $\mathbf{D}_v$ is given by [13]

$$\mathbf{D}_v = -\mathbf{G}_v \mathbf{Y}_o \in \mathbb{C}^{N_v \times 1} \quad (4.78)$$

where $\mathbf{Y}_o \in \mathbb{C}^{K \times 1}$ stands for the optimal control signal.

In the tuning stage, the error signal from the physical error microphones is given by

$$\mathbf{E}_p = \mathbf{G} \mathbf{Y}_o + \mathbf{D}_p \in \mathbb{C}^{M \times 1} \quad (4.79)$$

where $\mathbf{D}_p \in \mathbb{C}^{M \times 1}$ is the disturbance at the physical microphones.

In the control stage, with the aid of the auxiliary filter, the physical error signal remains unchanged and is given by

$$\mathbf{E}_p = \mathbf{G} \mathbf{Y}_{o,c} + \mathbf{D}_p \in \mathbb{C}^{M \times 1} \quad (4.80)$$

where $\mathbf{Y}_{o,c} \in \mathbb{C}^{K \times 1}$ denotes the steady-state control signal in the control stage.

4.5.1 Case i: Over- or fully-determined tuning and control stages.

The first case shows the optimal output control signal of the over- or fully-determined formulation in the tuning and control stages. The transfer function matrix of virtual secondary paths $\mathbf{G}_v \in \mathbb{C}^{N_v \times K}$ is assumed to be a full-rank matrix, and hence, $\mathbf{G}_v^H \mathbf{G}_v$ is invertible. In the tuning stage, the optimal control signal in (4.78) is derived through the least-squares (LS) approach as

$$\mathbf{Y}_o = -(\mathbf{G}_v^H \mathbf{G}_v)^{-1} \mathbf{G}_v^H \mathbf{D}_v. \quad (4.81)$$

The steady-state control signal of control stage is the LS solution of (4.80) and is obtained as

$$\mathbf{Y}_{o,c} = (\mathbf{G}^H \mathbf{G})^{-1} \mathbf{G}^H (\mathbf{E}_p - \mathbf{D}_p) \quad (4.82)$$

where we also assumed the transfer function of physical secondary path $\mathbf{G} \in \mathbb{C}^{M \times K}$ to be a full-rank matrix, and hence, $\mathbf{G}^H \mathbf{G}$ is invertible. By substituting (4.79) into (4.82) the steady-state control signal becomes

$$\mathbf{Y}_{o,c} = \mathbf{Y}_o. \quad (4.83)$$

Therefore, when there are fewer secondary sources than the physical and virtual microphones, the MVANC technique is able to achieve optimal control at the virtual locations.

4.5.2 Case ii: Under-determined tuning stage and an over-determined control stage.

The second case illustrates an under-determined tuning stage, where there are less sensors at the virtual locations than control sources. The optimal control signal in the tuning stage is derived from the LS solution of (4.78) as [108]

$$\mathbf{Y}_o = -\mathbf{G}_v^H (\mathbf{G}_v \mathbf{G}_v^H)^{-1} \mathbf{D}_v + [\mathbf{I}_K + \mathbf{G}_v^H (\mathbf{G}_v \mathbf{G}_v^H)^{-1} \mathbf{G}_v] \mathbf{z}, \forall \mathbf{z} \in \mathbb{C}^{K \times 1} \quad (4.84)$$

where $\mathbf{z}$ is an arbitrary vector and $\mathbf{I}_K$ denotes an $K \times K$ identity matrix. Since $\mathbf{G}_v$ is a full-rank matrix, $\mathbf{G}_v \mathbf{G}_v^H$ is invertible. In the control stage, the steady-state control signal is derived from the LS solution of (4.80) given by

$$\mathbf{Y}_{o,c} = (\mathbf{G}^H \mathbf{G})^{-1} \mathbf{G}^H (\mathbf{E}_p - \mathbf{D}_p). \quad (4.85)$$

By substituting (4.79) and (4.84) into (4.85) the steady-state control signal also reduces to

$$\mathbf{Y}_{o,c} = \mathbf{Y}_o. \quad (4.86)$$

Therefore, when the number of secondary sources are greater than the virtual microphones but fewer than physical microphones, the MVANC technique is still able to achieve optimal control at the virtual locations.

4.5.3 Case iii: Over-determined tuning stage and an under-determined control stage.

In the third case, the tuning stage is over- or fully-determined, where there are more control sources than sensors at the virtual locations. Hence, the optimal control signal in the tuning stage is given by (4.81), as defined above in Case i. Since $\mathbf{G}$ is assumed to be a full-rank matrix, $\mathbf{G}\mathbf{G}^H$ is invertible. In the control stage, the steady-state control signal in the control stage is derived from the LS solution of (4.80) as

$$\mathbf{Y}_{o,c} = \mathbf{G}^H (\mathbf{G}\mathbf{G}^H)^{-1} (\mathbf{E}_p - \mathbf{D}_p) + \left[\mathbf{I}_K - \mathbf{G}^H (\mathbf{G}\mathbf{G}^H)^{-1} \mathbf{G} \right] \mathbf{z}, \quad \forall \mathbf{z} \in \mathbb{C}^{K \times 1} \quad (4.87)$$

Substituting (4.79) into (4.87) yields

$$\mathbf{Y}_{o,c} = \mathbf{G}^H (\mathbf{G}\mathbf{G}^H)^{-1} \mathbf{G}\mathbf{Y}_o + \left[\mathbf{I}_K - \mathbf{G}^H (\mathbf{G}\mathbf{G}^H)^{-1} \mathbf{G} \right] \mathbf{z}, \quad \forall \mathbf{z} \in \mathbb{C}^{K \times 1} \quad (4.88)$$

Hence,

$$\mathbf{Y}_{o,c} \neq \mathbf{Y}_o. \quad (4.89)$$

Therefore, when the number of the secondary sources is greater than the physical microphones but is fewer than the virtual microphones, it is unable to achieve optimal control at the virtual locations.

4.5.4 Case iv: Under-determined tuning and control stage.

In the final case, both the tuning and control stages are under-determined. The steady-state control signal in the control stage is given by (4.88) as derived in Case iii, which does not equate to the under-determined control signal in the tuning stage as shown in (4.89).

Therefore, it can be deducted that once the number of secondary sources exceeds that of physical microphones, the MVANC technique fails to achieve optimal control at the desired virtual positions.

FIGURE 4.11: Schematic diagram of the four channel MVANC system with different configurations: (a) $K = 4$ (secondary sources), $M = 4$ (Physical MICs), $N_v = 4$ (Virtual MICs); (b) $K = 4$, $N_v = 4$, $M = 4$; (c) $K = 3$, $M = 4$, $N_v = 4$; (d) $K = 3$, $M = 3$, $N_v = 4$; (e) $K = 3$, $M = 2$, $N_v = 4$; (f) $K = 3$, $M = 4$, $N_v = 2$.

4.6 Simulation: MVANC with measured transfer functions

To verify the causal independence, as well as the effect of the alternative configurations, a series of numerical simulations are conducted with measured transfer functions from a real system. All the impulse responses of the secondary paths and primary paths used are measured from a four-channel ANC system, as shown in Fig. 4.11. In the MVANC system, the length of the control filters and secondary path models are set to 512 and 256 taps, respectively. The system operated with

FIGURE 4.12: Coefficients of the four steady-state control filters in simulation (a).

a sampling rate of 16 KHz. The primary noise sources in the tuning and control stages are Gaussian white noise and uniformly distributed random noise, respectively. These noises are filtered by a band-pass filter with a frequency range of 500 Hz to 1500 Hz.

FIGURE 4.13: Noise reduction in dB at the virtual microphone positions of the six simulation conditions.

In simulation (a), a fully-determined system is investigated, where there are four secondary sources ($K = 4$), four virtual microphones ($N_v = 4$), and four physical microphones ($M = 4$), as shown in Fig. 4.11 (a). The coefficients of the control filters in the tuning and control stages of simulation (a) in the steady-state are displayed in Fig. 4.12. The control filters in both stages, as well as their noise reduction levels, are visibly similar, as illustrated in Fig. 4.12 and Fig. 4.13 (a), respectively. This result verifies the above-mentioned conclusion that the MVANC

FIGURE 4.14: Coefficients of the four steady-state control filters in simulation (b).

technique can achieve the optimal noise control on virtual positions in a causal and fully-determined setup. It is worth noting that the control filter in the tuning stage is the same as the optimal control of the conventional MCANC technique.

To investigate the causal relation between the physical microphones and virtual microphones, the positions of the physical microphones and virtual microphones are swapped in Simulation (b), as illustrated Fig. 4.11 (b). In this situation, the virtual

FIGURE 4.15: Steady-state control filters in the tuning (blue line) and control (yellow line) stages in simulation (e).

microphones are closer to the secondary sources, and the disturbance and anti-noise wave propagates through the virtual microphone to physical microphone positions. Hence, a causal transfer function that describes the path from physical microphone to virtual microphone positions is non-existent. Hence, the signals from the physical microphones are not causally related with those from the virtual microphones. By analyzing the coefficients of both the tuning and control stages, however, the MVANC technique still obtains visibly similar control filters, as shown in Fig. 4.14. The noise reduction levels achieved at the virtual positions are arguably similar, as shown in Fig. 4.13 (b). Therefore, the MVANC technique overcomes the causal dependency between physical microphones and virtual microphones.

In simulation (c) and (d), there are fewer secondary sources than virtual microphones and physical microphones, as depicted in Fig. 4.11 (c) to (d) respectively, which satisfy the conditions in Case i. Hence, similar noise reduction levels are achieved at the virtual positions in both the tuning and control stages, as shown in Fig. 4.11 (c) and (d). This result verifies the analysis in Case i.

In simulation (e), there are three secondary sources ($K = 3$), two physical microphones ($M = 2$), and four virtual microphones ($N_v = 4$). In this situation, the control filters in the tuning stage and the control stage are different, as shown in Fig. 4.15. This results in the poor noise reduction in the control stage at the virtual locations, as illustrated in Fig. 4.11 (e). The simulation outcome validates the conclusion of Case iii and iv that once there are more secondary sources than physical microphones, the MVANC technique fails to achieve optimal control at the desired virtual locations.

In simulation (f), there are three secondary sources ($K = 3$), four physical microphones ($M = 4$), and two virtual microphones ($N_v = 2$), which satisfies the condition in Case ii. The noise reduction performances of the control and tuning stage at the virtual positions are similar, as shown in Fig. 4.11 (f). This result supports the conclusion in Case ii.

Through these simulation results, we found that once the number of secondary sources exceeds that of the physical microphones, the MVANC technique will fail to obtain satisfactory noise cancellation at the virtual positions.

4.7 Experiments of the 4-channel MVANC

To validate the theoretical analysis and simulations presented in the previous sections, we built a small noise chamber, as shown in Fig. 4.16. The noise chamber possesses one primary noise source inside the chamber and a full $4 \times 4 \times 4$ MCANC system, which consists with four reference microphones, four secondary sources, and four error microphones. The MCANC system is equipped with a high-speed controller (NI PXIe 8880) and executes at a sampling rate of 16 kHz. The control filters and the secondary path estimate in the ANC system have 512 and 256 taps, respectively. In experiments, the primary noise is a broadband noise with the frequency range from 500 – 1400 Hz.

FIGURE 4.16: Physical configuration of the $(4 \times 4 \times 4)$ MCANC with 4 virtual microphones in the far field in Experiment 1.

4.7.1 Experiment 1: four secondary sources, four physical microphones, and four virtual microphones.

In Experiment 1, the virtual microphones are located at a position further away from the aperture than the physical microphones, as illustrated in Fig. 4.16. Fig. 4.17 and Fig. 4.18 exhibit the power spectrum of the error signal at the physical microphones and virtual microphones. It can be seen that the physical microphones and virtual microphones have apparent noise reduction when ANC is on. Furthermore, the power spectrum of each microphone at the tuning stage and control stage almost overlap each other, which implies that the tuning and control stages have similar control filters.

FIGURE 4.17: The power spectrum of physical error signals in Experiment 1.

FIGURE 4.18: The power spectrum of virtual error signals in Experiment 1.

FIGURE 4.19: Noise reduction levels of microphones in Experiment 1.

Figure 4.19 displays the noise reduction levels of MVANC in the tuning stage, the control stage, and without virtual sensing technique. It can be seen that the sound pressure levels (SPL) of each microphone in the tuning stage and the control stage are similar. Furthermore, the virtual microphone achieves better noise reduction performance when the ANC is running with the virtual sensing technique than that without virtual-sensing technique.

4.7.2 Experiment 2: Swapping physical microphones and virtual microphones.

In Experiment 2, the locations of the physical microphones and virtual microphones are swapped, as shown in Fig. 4.20. Fig. 4.21 and Fig. 4.22 exhibit the power spectrum of the error signal at the physical and virtual microphones. The error signals of MVANC in the tuning stage and the control stage have similar power spectrum patterns.

Figure 4.23 illustrates the noise reduction levels of MVANC in the tuning stage, the control stage, and without the virtual sensing technique. The results prove that, at least for the control of noise through an aperture, MVANC can project the quiet zone in the near field while the physical microphones are placed in the far field.

FIGURE 4.20: Physical configuration of the $(4 \times 4 \times 4)$ MCANC with 4 virtual microphones in the near field in Experiment 2.

FIGURE 4.21: The power spectrum of physical error signals in Experiment 2.

FIGURE 4.22: Power spectrum of virtual error signals in Experiment 2.

FIGURE 4.23: Noise reduction levels of microphones in Experiment 2.

FIGURE 4.24: Physical configuration of the $(4 \times 3 \times 4)$ MCANC with 4 virtual microphones In Experiment 3.

Moreover, MVANC achieves better noise reduction at the virtual microphones than MCANC, which does not adopt the virtual sensing technique.

4.7.3 Experiment 3: Three secondary sources, four physical microphones, and four virtual microphones.

In Experiment 3, the number of secondary sources is reduced to three, as shown in Fig. 4.24. Fig. 4.25 and Fig. 4.26 displays the power spectrum of the error signals in the physical and virtual microphones. The power spectrum of error signals in the tuning and the control stage is similar. Hence, MVANC still works well even when the number of secondary sources is less than the number of physical microphones. Furthermore, the noise reduction levels at the microphones, as shown in Fig. 4.27, also verify the fact that MVANC performs better in noise cancellation at the virtual locations than the MCANC without virtual sensing technique.

4.7.4 Experiment 4; three secondary sources, three physical microphones, and four virtual microphones.

In Experiment 4, the numbers of both secondary sources and physical microphones are reduced to three, as illustrated in Fig. 4.28. The power spectrum of the error signal at the physical microphones and virtual microphones are shown in Fig. 4.29

FIGURE 4.25: The power spectrum of physical error signals in Experiment 3.

FIGURE 4.26: The power spectrum of virtual error signals in Experiment 3.

FIGURE 4.27: Noise reduction levels of microphones in Experiment 3.

FIGURE 4.28: Physical configuration of the $(4 \times 3 \times 3)$ MCANC with 4 virtual microphones in Experiment 4.

FIGURE 4.29: The power spectrum of physical error signals in Experiment 4.

and Fig. 4.30, respectively. Noise reduction levels at the microphones are depicted in Fig. 4.31. All these results prove that MVANC functions well when the number of physical microphones equals that of the secondary sources. Moreover, MVANC achieves better noise reduction levels at the virtual microphones than conventional MCANC.

4.7.5 Experiment 5: three secondary sources, two physical microphones, and four virtual microphones.

In Experiment 5, MVANC possesses three secondary sources and two physical microphones, as shown in Fig. 4.32. The power spectrum of the error signals at virtual microphones during the tuning stage and the control stage is different, as shown in Fig. 4.34. The noise reduction levels at the virtual microphones in the control stage are worse than in the tuning stage, as illustrated in Fig. 4.35. It confirms the previous theoretical analysis that when there are less physical microphones than secondary sources, MVANC will fail to achieve the optimal control on the virtual disturbance.

FIGURE 4.30: The power spectrum of virtual error signals in Experiment 4.

FIGURE 4.31: Noise reduction levels of microphones in Experiment 4.

FIGURE 4.32: Physic configuration of the $(4 \times 3 \times 2)$ MCANC with 4 virtual microphones in Experiment 5.

FIGURE 4.33: The power spectrum of physical error signals in Experiment 5.

4.7.6 Experiment 6: three secondary sources, four physical microphones, and two virtual microphones.

In Experiment 6, the numbers of the secondary sources and the virtual microphones are set to three and two, respectively, as shown in Fig. 4.36. Fig. 4.37 and Fig. 4.38 shows that the power spectrum of the error signal at the physical and virtual microphones is almost the same. The noise reduction levels at virtual microphones in the tuning stage and the control stage are similar, as depicted in Fig. 4.39. Therefore, when the number of physical microphones is more than that of the secondary sources, no matter how many virtual microphones there are, MVANC is capable of attenuating the noise at the virtual position.

FIGURE 4.34: The power spectrum of virtual error signals in Experiment 5.

FIGURE 4.35: Noise reduction levels of microphones in Experiment 5.

FIGURE 4.36: Physical configuration of the $(4 \times 3 \times 2)$ MCANC with 2 virtual microphones in Experiment 6.

FIGURE 4.37: The power spectrum of physical error signals in Experiment 6.

FIGURE 4.38: The power spectrum of virtual error signals in Experiment 6.

FIGURE 4.39: Noise reduction levels of microphones in Experiment 6.

4.8 Conclusions

The single-channel and multichannel virtual-sensing active noise control (MVANC) method examined in this section has been previously demonstrated as an effective approach to project the quiet zone from the error microphones to desired virtual positions where the physical placement of microphones is impractical. The frequency- and time-domain analysis based on random primary noise show that the MVANC technique converges to the same optimal control filter as with conventional MCANC, which utilizes the real error microphones at the virtual locations. Further analysis from an over- or fully-determined system revealed that the MVANC technique is unaffected by causal restrictions between microphones.

Analysis of under-determined formulations in either or both the tuning and control stages exposes the limitations of the MVANC technique. The MVANC technique failed to achieve the optimal control on the disturbance at the desired virtual positions when there are fewer physical microphones than secondary sources. A series of simulations and experiments conducted under a number of conditions verify the conclusions obtained in the frequency domain analysis.

Chapter 5

Selective fixed-filter feedforward active noise control technique

Active control of broadband noise in large spaces requires multiple control sources that operate at higher sampling frequencies with long control filters. To realize large multichannel ANC systems, high-performance processors, such as multi-core DSP processors, FPGA, and GPU [109] are required, which increases cost and complicates the programming effort. These processing limitations undermine its potential applications (e.g., in building windows, large dimension ducts or closure rooms, active noise barriers) and deteriorate the system stability.

There are many different approaches to reduce the computational burden for large structure ANC systems. One such solution is the use of a fixed filter to control different types of noise sources and orientations [110]. The coefficients of the control filter are first pre-trained for different ANC setups, and then these trained (fixed) coefficients of the control filter can be deployed in the actual operation using low-cost analog and digital processing units [111]. A point in case is from ANC headphones, which generally use analog filters by pre-tuning the gain and phase response to handle different types of noise environments [112, 113]. However, this fixed-coefficient filter ANC usually cannot change the control filter's coefficients for different primary noises on the fly, which may result in poor noise reduction performance.

Selective active noise control (SFANC) is a method to select a pre-trained control filter for different primary noise, instead of using conventional real-time computation of control filter coefficients. SFANC has the advantage of improving the robustness of the control filter while reducing the computational complexity. This chapter presents a practical, computationally-efficient strategy to adapt to different primary noise types. This method is known as the selective fixed-filter active noise control (SFANC) and uses the frequency-band-match approach to match the incoming noise frequency patterns with the spectrum of the control filter. The pre-trained control filter that matches the primary noise signature will be selected as the control filter for the ANC system. In this way, the control system can achieve a satisfactory noise reduction performance while maintaining a low computational complexity level. Furthermore, due to the fact that it does not involve a feedback control mechanism, SFANC obtains considerable high robustness even for the non-stationary disturbance. Simulations are carried out to validate the noise reduction performance of SFANC compared to the conventional FxLMS algorithm.

5.1 Selective active noise control approach

Since the primary and secondary paths vary slowly in many realistic operating environments, such as in a car cabin or an air duct [97], it is reasonable to assume that the impulse responses of these acoustic paths are time-invariant. Hence, we pre-train the control filter coefficients for a specific acoustic path in a preliminary stage to attenuate a specific primary noise. This describes the aforementioned fixed-filter method, which avoids the adaptive updating during the control stage. However, each fixed filter only works well for a specific primary noise, and thus, variation in the primary noise will unquestionably affect the noise cancellation performance. To improve the noise reduction performance of the fixed filter to accommodate different primary noise types, the selective fixed-filter active noise control (SFANC) approach is proposed [91, 110]. The SFANC approach operates in two stages: (1) an offline pre-training stage where a database of control filters are trained before implementation, and (2) an online control stage where the disturbance is mitigated by the most appropriate control filter, as shown in Fig. 5.1.

FIGURE 5.1: The processing block of the SFANC method, which features the first stage of deriving a set of control filters based on commonly encountered noise types and noise paths. Subsequently, selective fixed-filter algorithm extracts the closest control filter to filter out the actual noise in real-time.

5.1.1 Offline pre-training stage of control filters

Compared to the conventional ANC method, SFANC involves a preliminary training stage, as shown in Fig. 5.1. In this stage, a series of measured or synthesized noises sources are utilized for training. The training noises are firstly modified by real or measured primary paths, which creates corresponding disturbances. Subsequently, the adaptive algorithm utilizes both the training noises and disturbances to obtain corresponding sets of optimal filters.

5.1.2 Online control: Feature extraction and control filter selection

In the online control stage, features of the detected noise are firstly extracted, as shown in Fig. 5.1. These extracted features, such as its amplitude, frequency, or phase, should be unique characteristics that can distinguish noise types up to the required accuracy. Subsequently, the SFANC system selects the control filter from the database based on the extracted features.

Hence, it becomes apparent that an inappropriately chosen control filter produces inferior noise reduction performance compared to the conventional FxLMS algorithm. Popular machine learning or deep learning methods, such as the convolutional neural network (CNN) [114], can classify the incoming primary noise to match the most appropriate control filter. Nevertheless, since these computationally-intensive classification methods increase the overall latency of the SFANC online control stage, its practicality is undermined. To overcome this problem, the following sections use the frequency-band-match method for filter selection and propose an efficient SFANC algorithm based on this method.

Once the control filter is decided, the SFANC system functions as a fixed filter. It processes the primary noise acquired by the reference microphone and generates an anti-noise wave to cancel the disturbance, as illustrated in Fig. 5.1. In contrast to the conventional adaptive ANC method, the absence of feedback signal [115] from the error microphone in SFANC minimizes the risk of algorithm divergence, enhancing its stability.

However, it is worth noting that the selective fixed-filter active noise control approach is only suitable for the time-invariant primary path since the pre-trained control filters are obtained from a specific path. Otherwise, the variation of the primary path would deteriorate the noise reduction performance of the active control system.

5.2 Frequency-Band-Match algorithm for SFANC

As mentioned in the previous section, the challenge of the SFANC system is the selection of a suitable control filter for the specific type of primary noise. This section introduces the frequency-band-match approach, which chooses the pre-trained filter based on the similarity of frequency band matching of the training signal with the actual primary noise to be controlled. Hence, the frequency band of the primary noise becomes the key feature for filter selection.

Figure 5.2 shows a single channel feedforward ANC, where $P(\omega)$ and $S(\omega)$ represent the transfer function of the primary path and the secondary path, respectively, and $W(\omega)$ denotes the transfer function of the control filter. For simplicity, the reference signal is assumed to be directly acquired from the noise source.

FIGURE 5.2: Block diagram of single channel feedforward ANC. The primary noise disturbance is denoted as $d(n)$ and the residual error signal $e(n)$ is acoustically measured from microphone at the control region.

In the training phase, a broadband training signal $X_0(\omega)$ is input as a reference signal

$$X_0(\omega) = T_0(\omega) \cdot \text{rect} \left(\frac{\omega - \omega_0}{2B_0} \right) \quad (5.1)$$

where $T_0(\omega)$ is a conjugate symmetry function with respect to frequency ω [116]. ω_0 and B_0 are the central frequency and bandwidth of the training signal, respectively. The rectangular function in (5.1) is given by

$$\text{rect} \left(\frac{\omega - \omega_0}{2B_0} \right) = \begin{cases} 1 & |\omega| \in [\omega_0 - B_0, \omega_0 + B_0] \\ 0 & \text{otherwise.} \end{cases} \quad (5.2)$$

The filtered reference signal is stated as

$$r_0(n) = x_0(n) * s(n) \quad (5.3)$$

where $*$ stands for the linear convolution operation; $x_0(n)$ and $s(n)$ represent the training signal and the impulse response of the secondary path, respectively. Hence, the power spectral density (PSD) of the filtered reference signal is written as

$$S_{r_0}(\omega) = S_{x_0}(\omega) |S(\omega)|^2 \cdot \text{rect} \left(\frac{\omega - \omega_0}{2B_0} \right) \quad (5.4)$$

where $S(\omega)$ denotes the transfer function of the secondary path, and the PSD of the training signal is stated to

$$S_{x_0}(\omega) = \mathbb{E} [X_0(\omega) X_0^*(\omega)]. \quad (5.5)$$

$\mathbb{E}[\cdot]$ stands for the expectation operation. The cross-spectral density of the filtered reference $r_0(n)$ and the disturbance $d(n)$ is expressed as

$$S_{d_0 r_0}(\omega) = S_{x_0}(\omega)P(\omega)S^*(\omega) \cdot \text{rect}\left(\frac{\omega - \omega_0}{2B_0}\right). \quad (5.6)$$

According to [9], we can obtain the optimal control filter of the training signal at $\omega \in [\omega_0 - B_0, \omega_0 + B_0]$ as

$$W_{\text{opt}}^0(\omega) = \frac{S_{d_0 r_0}(\omega)}{S_{r_0}(\omega)} = \frac{P(\omega)}{S(\omega)}. \quad (5.7)$$

The recursive algorithm, such as the FxLMS algorithm, can be used to compute $W_{\text{opt}}^0(\omega)$ outside the frequency range of interest (i.e., $|\omega| \in [0, \omega_0 - B_0) \cup (\omega_0 + B_0, \pi]$). Since the training signal $X_0(\omega) = 0$ ($|\omega| \in [0, \omega_0 - B_0) \cup (\omega_0 + B_0, \pi]$), the optimal control filter is derived in Appendix B.1, and stated as

$$W_{\text{opt}}^0(\omega) = 0, \quad |\omega| \in [0, \omega_0 - B_0) \cup (\omega_0 + B_0, \pi]. \quad (5.8)$$

By combining (5.8) and (5.7), the optimal control filter for the training signal can be rewritten as

$$W_{\text{opt}}^0(\omega) = \frac{S_{d_0 r_0}(\omega)}{S_{r_0}(\omega)} \cdot \text{rect}\left(\frac{\omega - \omega_0}{2B_0}\right) = \frac{P(\omega)}{S(\omega)} \cdot \text{rect}\left(\frac{\omega - \omega_0}{2B_0}\right). \quad (5.9)$$

In the control stage, the primary noise is assumed to be a broadband noise with the central frequency ω_1 and bandwidth B_1

$$X_1(\omega) = T_1(\omega) \cdot \text{rect}\left(\frac{\omega - \omega_1}{2B_1}\right) \quad (5.10)$$

where $T_1(\omega)$ is a conjugate symmetric function with respect to frequency ω . The rectangular function in (5.10) is given by

$$\text{rect}\left(\frac{\omega - \omega_1}{2B_1}\right) = \begin{cases} 1 & |\omega| \in [\omega_1 - B_1, \omega_1 + B_1] \\ 0 & \text{otherwise.} \end{cases} \quad (5.11)$$

The optimal control filter of the primary noise $X_1(\omega)$ is derived as

$$W_{\text{opt}}^1(\omega) = \frac{S_{d_1 r_1}(\omega)}{S_{r_1}(\omega)} \cdot \text{rect} \left(\frac{\omega - \omega_1}{2B_1} \right) = \frac{P(\omega)}{S(\omega)} \cdot \text{rect} \left(\frac{\omega - \omega_1}{2B_1} \right). \quad (5.12)$$

As Fig. 5.2 illustrates, the reference signal $u(n)$ is given by

$$u(n) = x_1(n) + v(n) \quad (5.13)$$

where $v(n)$ denotes a white Gaussian noise $v(n) \sim \mathcal{N}(0, N_0/2)$, which represents the quantization noise, measurement noise on the reference sensor, or electronic components interference. If we apply $W_{\text{opt}}^0(\omega)$ as the control filter in Fig. 5.2 to cancel the primary noise, the error signal is expressed as

$$e(n) = d_1(n) - [x_1(n) + v(n)] * w_{\text{opt}}^0(n) * s(n) \quad (5.14)$$

where $d_1(n)$ denotes the disturbance, and $w_{\text{opt}}^0(n)$ represents the impulse response of the optimal control filter in pre-training stage. The filtered reference signal is $r_1(n) = x_1(n) * s(n)$.

To simplify the analysis, we assume that $\omega_0 = \omega_1$ and $B_1 \leq B_0$. According to Appendix B.2, the PSD of the error signal in (5.14) is derived as

$$\begin{aligned} S_e(\omega) = & S_e(\omega)_{\min} \cdot \text{rect} \left(\frac{\omega - \omega_0}{2B_1} \right) \\ & + \frac{N_0}{2} |P(\omega)|^2 \left[\text{rect} \left(\frac{\omega - \omega_0}{2B_0} \right) - \text{rect} \left(\frac{\omega - \omega_0}{2B_1} \right) \right] \end{aligned} \quad (5.15)$$

where

$$S_e(\omega)_{\min} = S_{d_1}(\omega) - \frac{|S_{d_1 r_1}(\omega)|^2}{S_{r_1}(\omega)} + \frac{N_0}{2} |P(\omega)|^2 \quad (5.16)$$

and $S_{d_1}(\omega) = \mathbb{E}[D_1^*(\omega)D_1(\omega)]$. Here, $S_e(\omega)_{\min}$ is the PSD of the error signal when the control filter arrives the optimal solution $W_{\text{opt}}^1(\omega)$ [9]. If the primary path is assumed to be approximated as a pure delay with unit gain, the mean square error (MSE) is derived by integrating (5.15) as

$$\mathbb{E}[e^2(n)] = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_e(\omega) d\omega = J_{\min} + \frac{N_0}{2\pi} (B_0 - B_1) \quad (5.17)$$

where

$$J_{\min} = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_e(\omega)_{\min} d\omega \quad (5.18)$$

which is the minimal mean square error (MMSE) of the Wiener-Hopf solution [9, 117]. In (5.17), if the bandwidth B_0 equals B_1 , the MMSE of SFANC is the same as that of the FxLMS algorithm. It is worth noting that the same result can be observed even when the primary path is not a unit gain. Therefore, if the training signal of the control filter in the SFANC algorithm has the same frequency band as the primary noise, the SFANC algorithm can achieve the same noise reduction level as the FxLMS algorithm.

5.3 Proposed selective fixed-filter active noise control algorithm

In this section, we describe the proposed selective fixed-filter noise control (SFANC) algorithm, which uses the frequency-band-match method to perform the control filter selection for different primary noises. SFANC adopts the comb-partitioning filter structure, and the coefficients of control filters are obtained through a modified comb-partitioning frequency-domain adaptive filter (comb-PFDAF) [118, 119] algorithm. In comparison to other frequency-domain adaptive filters (PFDAF) [120–123], comb-PFDAF significantly reduces the computational complexity of the Fourier transformation in the algorithm [118, 124].

It is worth noting that in many ANC applications, the reference sensor is used to acquire as much information about the primary source as possible. Hence, it is reasonable to assume that the reference signal has the same frequency band as the primary noise.

5.3.1 Pre-training stage of the control filters

In the pre-training stage, a series of synthesized broadband signals are utilized as the primary noise to train a database of control filters. These broadband signals are created by passing white Gaussian noise through band-pass filters with different frequency bands. The comb-PFDAF algorithm is applied to train the control filters

FIGURE 5.3: Partitioned structure of the filter in K combs. The input signal is down-sampled at K times and then finishes the FFT operation to generate M frequency bins. The frequency bins are input to the M filter branches, and the summation of these branches is the output of the partitioned filter.

due to its reduced computational complexity in extracting the frequency spectrum of the reference signal in an adaptive manner.

In the section, we consider a control filter with a length of N . According to the comb-partitioning method, we split the impulse response of the control filter into K combs. The length of each comb is equal to $M = N/K$, where $K = 1, 2, \dots, N$. A DFT of length M is needed in the partitioned structure, as shown in Fig. 5.3.

In the partitioned filter, the input signal $x(k)$, at the k th iteration, is decimated by a factor of K to form the comb-partitioned reference signal given by

$$\mathbf{x}^c(k) = \begin{bmatrix} x(k) & x(k-K) & \dots & x(k-(M-1)K) \end{bmatrix}^T \quad (5.19)$$

and its Fourier transform is

$$\mathbf{X}(k) = \mathbf{F}_M \mathbf{x}^c(k) = \begin{bmatrix} X_0^c(k) & X_1^c & \cdots & X_{M-1}^c(k) \end{bmatrix}^T \quad (5.20)$$

where $\mathbf{F}_M \in \mathbb{C}^{M \times M}$ denotes the Fourier transform matrix given by

$$\mathbf{F}_M = \left(e^{-\frac{2\pi \cdot \mathbf{j} \cdot \mathbf{i}}{M}} / \sqrt{M} \right)_{i,j=0,\dots,M-1} \quad (5.21)$$

The output of the m th branch in the control filter, as shown in Fig. 5.3, is expressed as

$$y_m^c(k) = [\mathbf{W}_m^c(k)]^T \mathbf{X}_m^c(k) = \sum_{i=0}^{K-1} W_{m,i}^c(k) X_m^c(k-i) \quad (5.22)$$

where

$$\begin{aligned} \mathbf{W}_m^c(k) &= \begin{bmatrix} W_{m,0}^c(k) & W_{m,1}^c(k) & \cdots & W_{m,K-1}^c(k) \end{bmatrix}^T \\ \mathbf{X}_m^c(k) &= \begin{bmatrix} X_m^c(k) & X_m^c(k-1) & \cdots & X_m^c(k-K+1) \end{bmatrix}^T \\ m &= 0, 1, \dots, M-1. \end{aligned} \quad (5.23)$$

The output signal of the control filter is obtained from the summation of all branches as

$$y(k) = \sum_{m=0}^{M-1} y_m(k). \quad (5.24)$$

To recursively update the coefficient of the control filter, we propose the filtered reference comb-partitioning frequency domain adaptive filter (FxCFDAF), as illustrated in Fig. 5.4 and Fig. 5.5. For each weight vector $\mathbf{W}_m^c(k)$, as proven in Appendix B.3, its update equation is given by

$$\mathbf{W}_m^c(k+1) = \mathbf{W}_m^c(k) + \mu e(k) \mathbf{X}_m'^*(k) \quad (5.25)$$

where μ denotes the step size, and

$$\mathbf{X}_m'(k) = \begin{bmatrix} X_m'(k) & X_m'(k-1) & \cdots & X_m'(k-K+1) \end{bmatrix}^T. \quad (5.26)$$

$X_m'(k)$ represents the m th frequency bin in the Fourier transform of the filtered reference given by

$$x'(k) = \hat{s}(k) * x(k) \quad (5.27)$$

FIGURE 5.4: Block diagram of the filtered reference comb-partitioning frequency domain adaptive filter (FxCFDAF), which adopts the filtered reference signal based on the secondary path estimate $\hat{S}(z)$.

where $\hat{s}(k)$ denotes the impulse response of the secondary path estimate.

To obtain (5.26), as shown in Fig. 5.4, the filtered reference signal is first partitioned to K combs as

$$\mathbf{x}_f^c(k) = \begin{bmatrix} x'(k) & x'(k-K) & \cdots & x'(k-(M-1)K) \end{bmatrix}^T \quad (5.28)$$

FIGURE 5.5: Block diagram of the m th branch in FxCFDAF algorithm.

and its Fourier transform is calculated by

$$\mathbf{X}'_f(k) = \mathbf{F}_M \mathbf{x}'_f(k) = \begin{bmatrix} X'_0(k) & X'_1(k) & \cdots & X'_{M-1}(k) \end{bmatrix}^T. \quad (5.29)$$

By staking the m th element (frequency bin) of (5.29) from time k to $k - K + 1$, we can get the vector (5.26).

After the coefficients of control filters for different primary noises are obtained, they are stored into a control filter library (or database), as shown in Fig. 5.6. In order to generate a unique index or an address for different control filters, we first estimate the power spectral density of the corresponding reference signal $X(k)$ by accumulating its fast Fourier transform (FFT) result of each L iterations [125]

$$\mathbf{S}_x = \left[\sum_{l=0}^{L-1} \|X_1^c(k-l)\|^2, \sum_{l=0}^{L-1} \|X_2^c(k-l)\|^2, \dots, \sum_{l=0}^{L-1} \|X_{M/2}^c(k-l)\|^2 \right]^T. \quad (5.30)$$

FIGURE 5.6: The construction of coefficient library, which is used to store the pre-trained filters in training stage. The approximate power spectrum $\mathbf{S}_x$ is used to create a bit array A as the address in the library for storing the corresponding control filter coefficients.

Then, a threshold is derived by averaging of the minimum and maximum elements ($S_{\min}$ and $S_{\max}$) of the vector $\mathbf{S}_x$ in (5.30)

$$V_{\text{thr}} = \frac{S_{\min} + S_{\max}}{2} \quad (5.31)$$

It is compared with the elements of the vector $\mathbf{S}_x$, which results in a bit vector as

$$A = [a_1, \dots, a_i, \dots, a_{M/2}]^T \in \mathbb{R}^{M/2 \times 1} \quad (5.32)$$

where

$$a_i = \begin{cases} 0, & S_{x,i} < V_{\text{thr}} \\ 1, & S_{x,i} \geq V_{\text{thr}} \end{cases} \quad (5.33)$$

$S_{x,i}$ represents the i th element of $\mathbf{S}_x$. Finally, A is utilized as the index or address of the corresponding control filter, as illustrated in Fig. 5.6.

FIGURE 5.7: Overall block diagram of the SFANC algorithm. It composed of three parts: (1). the FFT transform part of the partitioned reference signal, (2). $M/2$ frequency filter branches, and (3). the noise classification and filter weight update part.

5.3.2 Selective active noise control algorithm

The online control stage of the selective fixed-filter active noise control (SFANC) algorithm replaces the adaptive update process of the FxCFDAF algorithm with classification and weight update based on the frequency-band-match method. In the algorithm, the partitioned reference signal is still decimated from the reference signal $X(k)$ by a factor of K , as shown in Fig. 5.7. In this way, the computations in the FFT of SFANC reduces by K times, compared with performing the Fourier transform with the reference signal.

Furthermore, since the output signal of the control filter is a real signal, its frequency spectrum is conjugate symmetric. Moreover, ANC does not handle the DC component (or zero-frequency bin) of the reference signal, and thus, $y_0^c(k) = [\mathbf{W}_0^c(k)]^T \mathbf{X}_0^c(k) = 0$. Therefore, the output signal is expressed as

$$y(k) = 2 \sum_{m=1}^{M/2} \text{real}\{y_m^c(k)\} \quad (5.34)$$

FIGURE 5.8: The classification and weight update part in SFANC. The bit vector A created from the approximate power spectrum $\mathbf{S}_x$ that is utilized as the index or address to fetch the corresponding control filter for the specific primary noise.

where M is assumed to be an even number, and $\text{real}\{\cdot\}$ denotes the real part of a complex number.

The online part of SFANC adopts the frequency-band-match for classification, and continuously updates the coefficients of the control filter, every L iterations. The block diagram of the classification and weight update is illustrated in Fig. 5.8. This unit estimates the power spectral density (PSD) of the reference in (5.30) and obtains the index or address A in (5.32). Subsequently, it fetches the pre-trained filter in the library corresponding to the index A and updates the control filter in the online control stage.

Hence, for online control, SFANC replaces the real-time adaptive update progress with a classification and weight update process. The latter accomplishes one calculation in L iterations and thus, reduces the computational complexity per iteration compared to the conventional FDAF algorithm, especially for multichannel ANC applications. However, a bit vector A with a long dimension requires that the system to store a considerable number of control filters. For instance, a A vector with 16 bit elements can represent 65,536 different control filters, which are impractical to be obtained by experiments. In contrast, if we shorten the dimension of the

A vector, the frequency resolution of the SFANC will decrease, which worsens the noise reduction performance of the algorithm.

Therefore, in the spirit of practicality, the classifier should select the most appropriate control filter based on similarity to input A , as shown in Fig. 5.8 and further detailed in Fig. 5.9. In its first layer of the classifier, the AND-distance network will calculate the similarity value between the input bit vector A and the indices of all pre-trained filters. d_i represents the similarity value that equates to the number of positive comparisons (logic one) in A and A^i , where A^i indicates the address of the i th pre-trained filter. Hence, a large similarity value indicates a larger bit-for-bit similarity. Subsequently, the maximum priority queue outputs the index i with largest similarity index d_i . The corresponding i th control filter, which is obtained in the preliminary stage by (5.25), will be chosen as the current control filter. It is worth noting that the addresses or indices in the first layer are arranged in ascending bandwidth of the corresponding training noise. This similarity value is thus equivalent to the binary distance [126, 127] defined in Appendix B.4, which indirectly represents the similarity (or closeness) of the input noise frequency bands to the training noise.

From Fig. 5.9, it is observed that the classifier only contains several AND gates and accumulators, which consumes minimal resources in the processor.

5.4 Computational complexity analysis

The most time-consuming operations in a digital signal processing (DSP) algorithm are its multiplication and addition operations. Hence, in this section, we analyze the computational complexity, in terms of the multiplication and addition operations of the SFANC algorithm in comparison with the other FxLMS-based algorithms.

We consider a SFANC algorithm with N coefficients and K combs. First, it consumes $\frac{3N}{2K} \log_2 \frac{N}{K}$ real multiplications and $\frac{2N}{K} \log_2 \frac{N}{K}$ real additions to complete the Fourier transform in (5.20). Subsequently, the control filter unit in (5.34) operates with N multiplications and $N - 1$ additions. To compute the power spectrum, the algorithm uses $\frac{2N}{K}$ multiplications and $\frac{2N}{K}$ additions. Moreover, the classifier runs in parallel with the power spectral density estimation and takes I (the number of

FIGURE 5.9: Implementation architecture of the modified classifier. The index A created from the reference signal is input into the classifier. The first layer computes the binary distance or similarity between input A and the indexes of all the stored pre-trained filters (I pre-trained filters stored in the coefficient library), where ‘&’ stands for the logic AND operation. The index, which owns the maximum similarity ratio with the input A , is used to fetch the corresponding filter in the second layer.

the pre-trained filter in the library) iterations to complete the classification, only consuming one addition in each iteration. Therefore, the computational complexity of SFANC is obtained as

$$\begin{cases} \Theta_{\text{Multiplication}} &= \frac{3N}{2K} \log_2 \frac{N}{K} + N + \frac{2N}{K} \\ \Theta_{\text{Addition}} &= \frac{2N}{K} \log_2 \frac{N}{K} + N + \frac{2N}{K} \end{cases} \quad (5.35)$$

In contrast, the FDAF algorithm uses $\frac{2N}{K} \log_2 \frac{N}{K} + 6N$ multiplications and $\frac{4N}{K} \log_2 \frac{N}{K} + 5N - 3$ additions in each iteration, where the length of the secondary path estimate is assumed to be N .

To exemplify, we assume that the length of the control filter in FxLMS, FDAF, and SFANC is $N = 1024$, and the secondary path estimate also has $N = 1024$ taps. The number of combs in FDAF and SFANC is equal to $K = 16$. The addition and multiplication usages per iteration in these three algorithms are summarized in Table 5.1. The comparison result suggests that the SFANC algorithm has a

TABLE 5.1: The number of additions and multiplications used in FxLMS, FDAF, SFANC algorithms, when the control filter has 1024 taps.

Algorithm	Real multiplications	Real addtions
FxLMS	3073	2037
FDAF	6624	5540
SFANC	1392	1312

FIGURE 5.10: Frequency and phase responses of the primary (upper row) and secondary (lower row) paths.

distinct advantage in alleviating the computation at load in practical real-time active control.

5.5 Numerical simulation results

To evaluate the SFANC performance and its comparative performance with the conventional FxLMS algorithm, simulations are conducted at the sampling rate of 10 kHz. The primary and secondary paths are chosen as a band-pass filter and a low-pass filter, respectively, and their magnitude and phase responses are shown in Fig. 5.10.

FIGURE 5.11: Noise reduction as a function of similarity ratios and the bandwidth of the training noise B_{train} in simulation A, where the bandwidth of the primary noise B_{primary} is set to 500 Hz. The noise reduction peaks as the similarity ratio approaches 1.

5.5.1 Simulation A: Noise reduction as a function of similarity

In Simulation A, the primary noise is a broadband noise (750 to 1250 Hz). A series of pre-trained filters are chosen to control the noise. The training signals of these filters have different frequency bandwidths (from 10 to 1000 Hz) but the same central frequency at 1000 Hz. To describes the frequency-band correlation of the primary noise and the training signal, a frequency-band similarity ratio is defined as

$$SR = \frac{B_{\text{overlap}}}{B_{\text{primary}} + B_{\text{train}} - B_{\text{overlap}}} \quad (5.36)$$

where B_{primary} and B_{train} denote the bandwidths of the primary noise and training signal and B_{overlap} is the overlapped bandwidth in both the primary noise and training noise. The similarity ratio represents the similarity between the primary noise and training signal. To derive the similarity ratio between the primary noise and training noise, we first obtain their index with (5.33), and then compute their binary distance, as described in Appendix B.4.

FIGURE 5.12: The power spectrum of the primary noise, and the frequency band of the training signal of the control filter chosen by SFANC algorithm.

Figure 5.11 shows the noise reduction levels of the different pre-trained filters. The similarity ratios on the x -axis show their degrees of similarity with the controlled primary noise. The highest noise reduction is achieved as the similarity ratio approaches 1. In this case, the training signal of the control filter occupies the same frequency band as the primary noise. As the similarity ratio starts to decrease below one, the noise reduction level of the corresponding control filter also gradually reduces. For example, when $B_{\text{train}} < B_{\text{primary}}$, there is uncanceled disturbance remaining in the error signal, which largely reduces the noise reduction performance. Therefore, a higher similarity ratio is desired.

5.5.2 Simulation B: Active control of stationary broadband noise

To train the control filters in the preliminary stage, we synthesize 88 band-limited white Gaussian noise tracks. The lower-bound of the band-pass filter increases from 500 Hz to 1000 Hz at 50 Hz interval for each bandwidth of 50 Hz to 800 Hz at 100 Hz interval. Moreover, the partitioned structure in the SFANC algorithm contains 512 filter coefficients and 4 combs. The accumulation iteration L in (5.30) is set to 2048, which implies that the update rate of the control filter is 2048 iterations.

FIGURE 5.13: The mean square error (MSE) of FxLMS and SFANC as a function of time in Simulation B. The SFANC algorithm achieves faster convergence speed and similar steady-state noise reduction performance as the FxLMS algorithm whose step size is less than 0.001. When the FxLMS algorithm chooses the step size of 0.002, it obtains a faster convergence but worse steady-state error than the SFANC algorithm.

In Simulation B, the primary noise is set as a band-limited noise (600 Hz to 1000 Hz) with an amplitude of 1.2. White noise is added to the reference signal as the measurement noise, and the signal-to-noise ratio (SNR) of the reference signal is set to 15.6 dB. In this case, both the SFANC and FxLMS algorithms are used to attenuate the primary noise. In the simulation, the training noise of the pre-trained filter chosen by the SFANC algorithm has a high similarity ratio of 0.913, which implies that the frequency bands of training signal and primary noise are mostly overlapped, as illustrated in Fig. 5.12.

Figure 5.13 illustrates the mean square error (MSE) of the FxLMS and SFANC algorithms as a function of time. The FxLMS algorithm is executed with a 512-tap FIR adaptive filter and three different step sizes. In the simulation, the SFANC algorithm has a faster convergence speed and a similar steady-state performance as the FxLMS when the step size is less than 0.001. In contrast, the FxLMS algorithm converges faster but exhibits poorer steady-state performance as the step

TABLE 5.2: Frequency band and amplitude of the time-varying broadband noise

Time (Second)	Frequency band (Hz)	Amplitude
0 to 3.5	500 to 850	1.5
3.5 to 7	720 to 1170	1.2
7 to 10.5	620 to 870	0.8

size increases. The steady-state noise reduction levels of three FxLMS algorithms are 9.5 dB, 13.5 dB, and 14.5 dB, respectively, whereas the SFANC algorithm achieves a noise reduction level of 14.3 dB. Therefore, as long as a high similarity ratio is obtained, the SFANC algorithm can attenuate stationary broadband noise satisfactorily with a high convergence rate.

5.5.3 Simulation C: Active control of time-varying broadband noise

In Simulation C, a change in the primary noise is invoked every 3.5 seconds, as shown in Table 5.2.

Taking reference from Simulation B, two step sizes, 0.001 and 0.0005, are adopted for the FxLMS algorithm. Upon detecting a change in the primary noise, the SFANC algorithm switches to the control filter with the highest similarity ratio while the FxLMS adapts and converges to a new steady-state error, as shown in Fig. 5.14. The similarity ratios between the training noises of these filters and the primary noise are 0.78, 0.89, and 0.83, respectively. The time history in Fig. 5.14 reveals that the SFANC algorithm reacts in a similar fashion to the conventional FxLMS algorithm.

5.5.4 Simulation D: Active control of recorded real-world noise

To further illustrate the behavior of SFANC under a realistic scenario, a band-limited train-whistle noise (500 to 1200 Hz) is selected as the primary noise, and the SNR of the reference is 20 dB. The time history of the disturbance signal without control, after control with FxLMS and after control with SFANC is illustrated in Fig. 5.15. The FxLMS algorithm is executed with a step size of 0.001. It is

FIGURE 5.14: The time history of the FxLMS and SFANC algorithm when dealing with the varying primary noise in Simulation C. The primary noise changes every 3.5 seconds. An updated control filter is invoked by the SFANC algorithm upon the onset of a different disturbance.

immediately apparent that the SFANC algorithm maintains a lower steady-state error than FxLMS, even with a time-varying disturbance. The noise reduction level of the SFANC and FxLMS algorithms are 25 dB and 19 dB, respectively.

Furthermore, since the convergence duration of the FxLMS algorithm is related to the variation in the power of the primary noise, the high power of the train-whistle noise results in longer convergence time. In contrast, the convergence time of the SFANC algorithm is independent of the primary noise. Without using the step size to control the learning rate under different input powers, the SFANC algorithm only takes L iterations to select the suitable filter and updates to the new control filter. Subsequently, the coefficients of the control filter remain unchanged during the L iterations, which avoids the instability of adaptive updating. Therefore, the presented simulations support the feasibility of the SFANC algorithm in real-world applications.

FIGURE 5.15: From top to bottom, the disturbance: without control, after control with FxLMS, and after control with SFANC. The SFANC algorithm shows a faster convergence speed when dealing with the whistle sound than the FxLMS algorithm, without a specific step size selection strategy.

5.5.5 Simulation E: The effect of FFT accumulation over L iterations on the noise reduction performance

The accumulation over L iterations in (5.30) determines the rate at which the SFANC algorithm updates the coefficients of the control filter. To determine L in Simulation D, we compute the noise reduction level as a function of L , as illustrated in Fig. 5.16. As L increases, the noise reduction level approaches its peak.

Instinctively, a small L ($L < 900$ in this case) leads to a fast response time of the SFANC algorithm to the detected primary noise. This comes at the expense of increased variance in the periodogram of the detected primary noise, which deteriorates the noise reduction. In contrast, a large L ($L \geq 900$ in this case) increases the accuracy of the periodogram at the expense of the response time. Therefore, in practice, a relatively large L should be chosen to ensure sufficient noise reduction while maintaining a satisfactory response time. At the moment, the optimal L has to be determined by trial and error.

FIGURE 5.16: Noise reduction level as a function of FFT accumulation iterations L of the SFANC algorithm in Simulation E. A small value of L ($L < 900$) results in the degradation of the noise reduction performance.

5.6 Conclusions

Selective active noise control (SFANC) is an efficient approach to alleviate the computational complexity of the adaptive algorithms used in active noise control applications. The distinct feature of this approach lies in its preliminary training stage in which a series of control filters are pre-trained for different types of noise.

SFANC selects the most suitable pre-trained filter as the control filter. However, the challenge of this approach is the computational complexity of the control filter selection process. A classification scheme for the SFANC algorithm, which adopts the frequency-band-match method is proposed to choose a control filter with similar frequency bands to the detected primary noise. The frequency-domain analysis revealed that the SFANC algorithm could achieve a comparable noise reduction level to the conventional adaptive filter.

For online control, the comb-partitioning filter technique is integrated into SFANC and hence, further reduces the computational complexity in the Fourier transformation. The proposed classifier of the algorithm significantly reduces the storage

requirements of the control filters and alleviates the computational load in the filter selection. Therefore, compared to the conventional adaptive algorithm used in ANC, the SFANC algorithm reduces computational complexity with a satisfactory noise reduction performance. Since the SFANC algorithm does not require adaptive feedback, unlike the conventional FxLMS algorithm, its response time, stability, and steady-state noise reduction performance is unaffected by the step size.

Chapter 6

The efficient implementation architectures of the feedforward active noise control on FPGA

A high sampling rate substantially improves the noise reduction performance of the digital feedforward ANC system. This is because the high sampling rate does not only broaden the frequency bandwidth of noise but also alleviates the causality constraint [128, 129]. However, it is impractical to implement a high-speed feedforward ANC system on a general-purpose processor, such as CPU or DSP, as they are designed for serial operation. Even for a GPU [109, 130], it is still challenging to deal with the high data throughput based on a sample-by-sample rather than batch-by-batch processing. In this situation, the FPGA appears to be the right choice due to its excellent parallel processing ability.

However, there are few dedicated algorithms or efficient architectures for the implementation of the feedforward ANC on FPGA. Hence, this chapter examines optimization methods to realize a high-speed digital processing algorithm. Based on these methods, the systolic FxLMS algorithm and multiple-parallel-branch and folding structure are proposed to implement the single-channel and multiple-channel ANC on FPGA.

6.1 Background: Programmable logic device

The programmable logic device (PLD) is a type of logic device that is programmed by the user to implement a specific logic function. It is usually categorized into the complex programmable logic device (CPLD) and the field-programmable gate array (FPGA). Aided by the rapid development of electronic design automation (EDA) tools, FPGA is becoming increasingly important in electronic design fields.

The FPGA consists of three parts: configurable logic unit, programmable input and output ports, and the interconnect unit. The high density of integrated circuits, processing speed, and robustness are the most apparent features of FPGA with low latency that can be within several nanoseconds. Combined with excellent parallel processing capability, FPGA is widely applied in the high-speed signal processing systems and real-time control systems. In contrast to general-purpose processors or digital signal processors (DSP), FPGA also exhibits higher stability during extended operations.

In comparison to the application-specific integrated circuit (ASIC), FPGA has the apparent advantages of a shorter development cycle, lower monetary investment, faster product realization, stronger market adaptability, and larger room for hardware upgrade. Furthermore, once the product is fully verified, the hardware description language (HDL) code of FPGA can be readily implemented on the ASIC, which will significantly reduce the cost of a complex digital system, such as the multichannel active noise control system.

6.2 Optimization of digital signal processing structure on programmable logic devices

A highly-efficient structure or architecture design is critical for the implementation of the digital signal processing system on the FPGA or ASIC. Hence, this section briefly introduces the terminologies and techniques for the structural optimization of the digital signal processing system. These techniques are also adopted in the realization of an active noise control system.

FIGURE 6.1: The data flow graph (DFG) of a 3-order FIR filter.

6.2.1 Latency and Throughput

Latency and throughput are two essential parameters that describe how fast a digital system operates.

Latency is the amount of time the system takes to complete one operation. It is usually expressed by the delay time from the data input to the data output and often described in the unit of seconds or system clocks.

Throughput is the rate at which the digital system can process the inputs. It represents how much input data the system deals within a given time, and hence, it is commonly expressed in samples per seconds (S/s).

6.2.2 Critical path

The critical path describes the data-flow path, which accounts for the longest computation time among all paths that contain zero delays in a data flow graph (DFG). This longest computation time dictates the maximum clock frequency of the digital system. For instance, Fig. 6.1 depicts the DFG of a 3-order FIR filter. We assume that the computation times of addition and multiplication are T_{Adder} and $T_{\text{Multiplier}}$, respectively. The critical path of the DFG is found to contain 1 multiplier and 3 adders, and hence, its longest computation time is given by

$$T_{\text{Critical path}} = T_{\text{Multiplier}} + 3T_{\text{Adder}} \quad (6.1)$$

FIGURE 6.2: DFG of a 3-order pipelining FIR filter.

as shown in Fig. 6.1. The maximum running frequency of the 3-order FIR is obtained from

$$f_{\max} = \frac{1}{T_{\text{Critical path}}}. \quad (6.2)$$

Therefore, the critical path is vital to the running speed of a digital system. The most effective approach to increase the maximum running frequency of a digital system is to shorten its critical path, which means reducing the longest computation time. Furthermore, the output of the 3-order FIR filter is given by

$$y(n) = \sum_{i=0}^3 w_i x(n-i). \quad (6.3)$$

6.2.3 Pipelining technique

The term of pipelining comes from an idea of a water pipe that continues pumping in the water without waiting for the water in the pipe to come out. Inspired by this method, we insert the latch in between the critical path of the digital system, which reduces the length of the path and therefore increases the maximum running or clock frequency.

For instance, to increase the maximum running frequency of the 3-order FIR filter in Fig. 6.1, we insert four latches into its original critical path, as illustrated in Fig. 6.2. Its new critical path only contains one multiplier, where its computation time $T_{\text{Multiplier}}$ is assumed to be greater than $3T_{\text{Adder}}$. Hence, the new maximum

FIGURE 6.3: Transposed structure of a 3-order peeling FIR filter.

running frequency of the FIR filter becomes

$$f_{\max} = \frac{1}{T_{\text{Multiplier}}}. \quad (6.4)$$

However, the output of the 3-order FIR filter also changes to

$$y(n) = \sum_{i=0}^3 w_i x(n-1-i) \quad (6.5)$$

which adds one more delay in the input signal.

6.2.4 Graph transposition approach

For the FIR filter design, another approach that increases the processing speed is graph transposition. According to this method, we reverse all arcs, input, and output directions in the original DFG to achieve a transposed structure. The transposed structure of the 3-order FIR filter is shown in Fig. 6.3.

The critical path of the transposed 3-order FIR filter contains one adder and one multiplier, as shown in Fig. 6.3. Therefore, its maximum running frequency is obtained as

$$f_{\max} = \frac{1}{T_{\text{Multiplier}} + T_{\text{Adder}}}, \quad (6.6)$$

which is obviously faster than the conventional transversal 3-order FIR filter in Fig. 6.1. Moreover, its output signal remains the same as (6.3).

6.2.5 Parallel technique

To improve the throughput of the 3-order FIR filter, we can rewrite (6.1) in the terms of its odd and even parts as

$$y(2n) = \sum_{i=0}^3 w_i x(2n - i) \quad (6.7)$$

$$y(2n + 1) = \sum_{i=0}^3 w_i x(2n + 1 - i) \quad (6.8)$$

We can use two identical FIR filters to implement (6.7) and (6.8). Therefore, the new FIR filter can process two inputs at one time, and its throughput increases by a factor of two times. However, the system consumes more additions and multiplications than the original FIR filter. This parallel technique effectively increases the system throughput at the expense of an increase in hardware resources consumption.

6.2.6 Folding technique

In many complex digital signal processing applications, the amount of system computations largely exceeds the limitations of the hardware resources in the FPGA or ASIC. In this situation, we can reuse some identical computing units in the algorithm to save hardware resources at the expense of increasing the processing time. This method is known as the folding technique. To implement this method, we can time-multiplex many operations into single-function units.

To illustrate, a DSP program with two addition operations is considered and given by

$$y(n) = a(n) + b(n) + c(n) \quad (6.9)$$

which is shown in Fig. 6.4 (a). To reduce the hardware resource doing addition, one hardware adder is reused to complete two additions in the program, as illustrated in Fig. 6.4 (b), which can also be described as that two addition operations are folded to a single hardware adder [131]. Hence, the new structure of two addition operations is a folded structure. With the assistance of a switch and latch, the folded structure completes $a(n) + b(n)$ in the first cycle, and subsequently completes

FIGURE 6.4: (a) A DSP program with two addition operations; (b) a folded structure of the two addition operations, where $l \in \mathbb{Z}^+$, and $2l + i$ denotes the $(2l + i)$ th cycle.

$a(n) + b(n) + c(n)$ in the second cycle. In this way, the folded structure uses one adder to finish one iteration in two cycles.

To derive a folded structure for a digital signal processing algorithm, the folding equation is usually applied to calculate new delay times at the edges (data path) of the DFG. Consider a data path connecting the nodes U and V , which can be the adder or multiplier, as shown in Fig. 6.5 (a), with $w(e)$ delays. In the folded

FIGURE 6.5: (a) The data path $U \rightarrow V$ with $w(e)$ delays; (b) The corresponding new path in the folded structure

structure, we utilize N cycles to complete one iteration of the algorithm, and the N is defined as the folding factor. The execution of the nodes U and V are scheduled at time $Nl + u$ and $Nl + v$, respectively, where l denotes the l th iteration. The operation of nodes U and V are executed by the hardware unit H_U and H_V in the corresponding folded structure, and the new delay time between these two nodes is given by

$$D_F(U \rightarrow V) = Nw(e) - P_U + v - u \quad (6.10)$$

where P_U denotes the number of the pipelining stages used for executing the operation of node U .

The following example shows how to construct a folded structure for a 3-order FIR filter. There are eight nodes from NO.1 to NO.8 in the DFG of the 3-order FIR

FIGURE 6.6: Folding a 3-order FIR filter.

filter, as shown as Fig. 6.6. The computations of the FIR filter in one iteration can be executed by one hardware adder S_1 and one hardware multiplier S_0 in 4 cycles. Hence, the folding factor N is equal to 4. In the figure, $(S_i|n)$ indicates that the operation of the node will be executed by S_i in the n th cycle.

Based on (6.10), the new delay times at the edges of the FIR filter is derived as

$$D_F(1 \rightarrow 5) = 4(0) - 0 + 0 - 0 = 0$$

$$D_F(2 \rightarrow 6) = 4(0) - 0 + 1 - 1 = 0$$

$$D_F(3 \rightarrow 7) = 4(0) - 0 + 2 - 2 = 0$$

$$D_F(4 \rightarrow 8) = 4(0) - 0 + 3 - 3 = 0$$

$$D_F(5 \rightarrow 6) = 4(0) - 0 + 6 - 5 = 1$$

$$D_F(6 \rightarrow 7) = 4(0) - 0 + 7 - 6 = 1$$

$$D_F(7 \rightarrow 8) = 4(0) - 0 + 8 - 7 = 1$$

where the pipelining stage P_U equates to 0. According to the new delay on each data path, the folded structure of Fig. 6.6 can be depicted as Fig. 6.7. In the figure, the label $[4l+i]$ implies that the labeled port should be connected to the switch in the $(4l+i)$ th cycle, where $i = 0, 1, 2, 3$ and l stands for the l th iteration. Although this folded structure reduces the hardware resources usages of the adder and multiplier, it also increases the usage of the registers. The variable life analysis and forward-backward register allocation [131] approaches can effectively solve this issue, as shown in Fig. 6.8, but these approaches increase the circuit design complexity. The next section will introduce an alternative method using multiple clock domains and block RAMs to reduce the number of registers in the folded structure.

FIGURE 6.7: Folded structure of the 3-order FIR filter.

FIGURE 6.8: Modified folded structure of the 3-order FIR filter with forward-back register allocation.

6.2.7 Systolic structure

The systolic structure that consists of many identical processing elements (PE) is one type of optimized architecture, which carries the features of synchronization, modularity, regularity, and extendibility. Since these PEs are highly parallelized and pipelined, the systolic architecture usually has a considerable high throughput rate.

To construct the systolic architecture of a DSP algorithm, we first need to draw the dependency graph (DG) of the algorithm. The DG describes the dependence of the computations in an algorithm. The node in the DG denotes the computation,

FIGURE 6.9: The dependency graph (DG) of a 3-order FIR filter.

and the edge represents the connection among nodes. Moreover, the DG has the computations of all iterations but does not contain delay elements.

For instance, the DG of a 3-order FIR filter, as shown in Fig 6.9. In the graph, each node stands for one computation (one addition and multiplication). It possesses three input edges, which can be expressed in vector form as, $[1, 0]^T$, $[0, 1]^T$, and $[1, -1]^T$, and one output edge, $[1, -1]^T$.

By projecting all the nodes and edges in the DG into processor axis P and schedule or time axis S, we can create its corresponding systolic structure. The computations of FIR algorithm can be completed by four PEs (PE0, PE1, PE2, and PE4). For the input edge $[1, 0]^T$, its projections on P and S axis are $[0, 0]^T$ and $[1, 0]^T$, respectively, which means the weight stays in the same PE in the next iteration and has one delay between iterations. For the input x edge $[0, 1]^T$, its projections on the P and

FIGURE 6.10: The systolic structure of a 3-order FIR filter.

S axis are $[0, 1]^T$ and $[0, 0]^T$, which means the x is passed to the next PE in the same iteration. For the input edge $[1, -1]^T$, its projections on P and S axis are $[0, -1]^T$ and $[1, 0]^T$, which means the data is passed to the forward PE in the next iteration. Repeating this procedure, we can derive the projections of all nodes and edges, and obtain its corresponding systolic structure. The systolic structure of the 3-order FIR filter is thus obtained as Fig. 6.10.

From Fig. 6.10, it can be seen that the critical path of the systolic FIR filter only contains one multiplier, where it is assumed that the multiplier consumes longer time than the adder. Therefore, the systolic FIR can achieve a higher operating frequency or throughput rate than other architectures.

6.3 Systolic FxLMS algorithm for feedforward ANC implementation

The section introduces an FxLMS algorithm with a systolic structure, which utilizes the techniques of graphic transposition and pipelining to achieve a high sampling rate and throughput with minimum influence on its convergence performance.

The feedforward control structure of ANC is widely applied, when dealing with broadband noise. The feedforward control structure requires a reference sensor to be placed near the noise source to provide time-advanced information of the noise wave to be canceled. The distance from the reference sensor to the secondary source generating the anti-noise wave is usually made long enough so that real-time control is feasible. Hence, when the distance is short, a fast controller is essential to the performance of the ANC system.

TABLE 6.1: Comparison of implementation platforms of noise canceling headset.

Chip	L_F	L_S	Structure	Precision	F_s
VLSI [132]	24	24	Feedforward	16 bit	96
TMS320C25 [133]	64	64	Feedback	16 bit	8
TMS320C50 [134]	16	256	Feedback	32 bit	19.2
TMS320C6711 [135]	-	-	Feedback	32 bit	8
STM32F407 [136]	32	32	Feedback	32 bit	4
PIC24H [137]	20	20	Feedback	16 bit	10

L_F : Tap length of the adaptive filter

L_S : Tap length of the secondary path model

F_s : Sampling rate (kHz)

In some applications, due to the compact size of the device, such as the noise canceling headset, it is not feasible to put a reference sensor far from the secondary source. The current practice in many noise canceling headsets is to apply the feedback control structure that is only efficient to reduce narrowband noise. As summarized in Table 6.1, the general purpose digital signal processor (DSP) and micro-controller are adequate to implement the feedback control structure.

To achieve good noise reduction performance, the adaptive filter must be sufficiently long. In this case, the general-purpose DSP and micro-controller may be incapable of high-sampling-rate processing. Hence, the field-programmable gate array (FPGA) is considered to be a better prototyping platform, due to its parallel processing capability [138]. When the adaptive filter is implemented in parallel, the sampling rate can be increased significantly.

On the other hand, adaptive algorithms, such as least squares (LS), least mean squares (LMS), recursive least squares (RLS), and frequency-domain LMS algorithms have been widely studied in linear estimators [139, 140]. However, these algorithms are not suitable for ANC because of the existence of the secondary path. The filtered-x LMS (FxLMS) algorithm has since been proposed [101]. The FxLMS algorithm is believed to be one of the most robust and widely adopted algorithms in ANC. Subsequently, a family of algorithms based on the basic FxLMS algorithm has been developed. The normalized FxLMS and variable step size FxLMS algorithms converge faster than the basic FxLMS algorithm by manipulating the step size [141, 142]. The moment FxLMS, filtered-x RLS, and sub-band FxLMS algorithms are alternative approaches to accelerate the convergence [143, 144]. The

FIGURE 6.11: Block diagram of the conventional FxLMS structure.

sign FxLMS algorithm can reduce the computational complexity greatly. Some of the above-mentioned algorithms have been described through this thesis in detail.

In order to investigate the feasibility of implementing the feedforward FxLMS algorithm in a noise-canceling headphone, a systolic FxLMS structure is proposed for implementation on the FPGA. As its critical path is short, the systolic FxLMS structure allows high sampling rate processing. With pipelining, the systolic FxLMS structure possesses a high system throughput and good scalability. If there are sufficient resources on the FPGA, increasing the tap length of the adaptive filter does not affect the processing time, which is significant merit of the systolic FxLMS structure.

6.3.1 Derivation of systolic FxLMS Structure

Fig. 6.11 illustrates the block diagram of the FxLMS algorithm composing of a N -tap adaptive filter, a secondary path model $\hat{S}(z)$, and a LMS coefficient update mechanism, where z^{-1} denotes the unit delay block; w_i denotes the i -th coefficient of the adaptive filter; $x(n)$ and $y(n)$ represent the input and output of the adaptive filter, respectively; and $e(n)$ denotes the error signal.

The critical path in Fig 6.11 extends from the leftmost multiplier to the rightmost adder. Hence, the maximum sampling rate is calculated by

$$f_{\max} = \frac{1}{T_{\text{Multiplier}} + (N - 1) T_{\text{Adder}}}. \quad (6.11)$$

It can be deduced that, as the tap length increases, the maximum sampling rate decreases.

The most direct method that improves the maximum operating frequency is to insert N registers into the critical path of the adaptive filter, as mentioned in the pipelining technique in Section 6.2.3. The LMS algorithm implemented in such a way is known as the delay LMS algorithm when an N -unit delay is used [145]. It is trivial to extend the delay LMS algorithm to include a secondary path model and become the delay FxLMS algorithm. The critical path of the delay FxLMS algorithm is twice that of the delay LMS algorithm because the secondary path model introduces one multiplier and one adder. However, the delay LMS and delay FxLMS algorithms have small step size bounds that lead to slow convergence.

To speed up the convergence of the delay FxLMS algorithm, we propose a systolic FxLMS structure based on the graphic transposition theorem [146]. As shown in Fig. 6.12, both the adaptive filter and the secondary path model are transposed, where $\hat{s}_i$ denotes the i th coefficient of the secondary path model; μ denotes the step size; and $x'(n)$ denotes the filtered input signal.

The critical path of the systolic FxLMS structure consists of two multipliers and two adders, as illustrated in Fig. 6.12. Hence, the maximum sampling rate is given by

$$f_{max} = \frac{1}{2T_{\text{Multiplier}} + 2T_{\text{Adder}}} \quad (6.12)$$

which is independent of the tap length of the adaptive filter.

Under the assumptions of slow adaptation and perfect secondary path model, the error signal of the systolic FxLMS structure is formulated as

$$\begin{aligned} e(n) &= d(n) - y'(n) \\ &= d(n) - s(n) * [\hat{\mathbf{w}}^T(n) \mathbf{x}(n)] \\ &= d(n) - \hat{\mathbf{w}}^T(n) \mathbf{x}'(n). \end{aligned} \quad (6.13)$$

where $d(n)$ is the primary noise signal; $s(n)$ is the secondary path that is assumed to be perfectly modeled by $\hat{s}(n)$; $*$ denotes the linear convolution; $y'(n) = s(n) * y(n)$ is the filtered output signal; $\mathbf{x}(n)$ is the input signal vector; $\mathbf{x}'(n) = s(n) * \mathbf{x}(n)$ is the filtered input signal vector; and $\hat{\mathbf{w}}(n)$ is the coefficient vector of

FIGURE 6.12: Block diagram of the FxLMS algorithm with a N taps control filter.

the adaptive filter, i.e.

$$\hat{\mathbf{w}}(n) = [w_{N-1}(n - N + 1), w_{N-2}(n - N + 2), \dots, w_0(n)]^T. \quad (6.14)$$

Therefore, (6.13) is rewritten as

$$e(n) = d(n) - \sum_{i=0}^{N-1} x'(n-i) w_i(n-i). \quad (6.15)$$

The gradient of the mean squared error with respect to the i -th coefficient of the adaptive filter is given by

$$\frac{\partial \mathbb{E}[e^2(n)]}{\partial w_i(n-i)} = 2\mathbb{E}\left[e(n) \frac{\partial e(n)}{\partial w_i(n-i)}\right] = -2\mathbb{E}[e(n)x'(n-i)]. \quad (6.16)$$

It is noteworthy that the gradient of the mean squared error is similarly calculated with respect to $w_i(n)$ in the derivation of the FxLMS algorithm, and with respect to $w_i(n-N)$ in the derivation of the delay FxLMS algorithm when an N -unit delay is used. However, the definition of the coefficient vector is different in the FxLMS and delay FxLMS algorithms, as compared to the proposed systolic FxLMS structure.

Based on (6.16), the i th coefficient update equation is given by

$$w_i(n-i+1) = w_i(n-i) + \mu \mathbb{E}[e(n)x'(n-i)]. \quad (6.17)$$

If it is further assumed that $e(n)$ and $x(n)$ are wide-sense stationary processes, the coefficient update equation is rewritten as

$$\begin{aligned} w_i(n+1) &= w_i(n) + \mu \mathbb{E}[e(n+i)x'(n)] \\ &= w_i(n) + \mu \mathbb{E}[e(n)x'(n-i)]. \end{aligned} \quad (6.18)$$

Replacing the expected value with the instant value yields the coefficient update equation of the systolic FxLMS structure as

$$w_i(n+1) = w_i(n) + \mu e(n)x'(n-i). \quad (6.19)$$

Equation (6.19) appears to be similar to the coefficient update equation of the FxLMS and delay FxLMS algorithms. However, the derivation is slightly different, as highlighted through (6.14) to (6.16).

6.3.2 Theoretical analysis on the systolic FxLMS algorithm

In this subsection, the theoretical analysis of the systolic FxLMS structure is presented to show the step size bound.

To simplify the derivation, the time-aligned coefficient vector of the adaptive filter is defined as

$$\mathbf{w}(n) = [w_{N-1}(n), w_{N-2}(n), \dots, w_0(n)]^T \quad (6.20)$$

which is equal to the coefficient vector of the FxLMS algorithm. Therefore, (6.19) can be rewritten in the same vector form as the FxLMS algorithm, i.e.

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n) \mathbf{x}'(n). \quad (6.21)$$

Taking the expected value of (6.21) yields

$$\bar{\mathbf{w}}(n+1) = \bar{\mathbf{w}}(n) + \mu E[e(n) \mathbf{x}'(n)]. \quad (6.22)$$

Then substituting (6.13) into (6.22) yields

$$\bar{\mathbf{w}}(n+1) = \bar{\mathbf{w}}(n) + \mu [\mathbf{p} - \mathbf{R}_{\mathbf{x}'} \bar{\mathbf{w}}(n)] \quad (6.23)$$

where

$$\mathbf{R}_{\mathbf{x}'} = \mathbb{E} [\mathbf{x}'(n) \mathbf{x}'^T(n)] \quad (6.24)$$

is the filtered input auto-correlation matrix, and

$$\mathbf{p}_{\mathbf{d}\mathbf{x}'} = \mathbb{E} [d(n) \mathbf{x}'^T(n)]. \quad (6.25)$$

Since $\mathbf{R}_{\mathbf{x}'}$ is a real symmetric matrix, it can be decomposed as

$$\mathbf{R}_{\mathbf{x}'} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T \quad (6.26)$$

where $\mathbf{\Lambda}$ is a diagonal matrix composed by the eigenvalues of $\mathbf{R}_{\mathbf{x}'}$; $\mathbf{Q}$ is the modal matrix formed by the eigenvectors of $\mathbf{R}_{\mathbf{x}'}$; and $\mathbf{Q}^T = \mathbf{Q}^{-1}$ holds.

Meanwhile, it is known that the Wiener-Hopf [31] method defines

$$\mathbf{p}_{\mathbf{d}\mathbf{x}'} = \mathbf{R}_{\mathbf{x}'} \mathbf{w}^o \quad (6.27)$$

where $\mathbf{w}^o$ is the optimal solution of the FxLMS algorithm.

Let

$$\mathbf{v}'(n) = \mathbf{Q}^T [\bar{\mathbf{w}}(n) - \mathbf{w}^o] \quad (6.28)$$

and substituting (6.28) into (6.23) yields

$$\mathbf{v}'(n+1) = \mathbf{v}'(n) + \mu \mathbf{\Lambda} \mathbf{Q}^T [\mathbf{w}^o - \hat{\mathbf{w}}(n)]. \quad (6.29)$$

The i -th element of $\mathbf{v}'$ is thus written as

$$v'_i(n+1) = v'_i(n) - \mu \lambda_k v'_i(n-i). \quad (6.30)$$

where λ_k is an eigenvalue of $\mathbf{R}_{x'}$. When (6.30) is treated as an impulse response function, the single-side z domain transfer function is formulated as

$$V'_i(z) = \frac{z^{i+1} v'_i(0)}{z^{i+1} - z^i + \mu \lambda_k}. \quad (6.31)$$

Since a stable system has all the poles of its transfer function inside the unit circle, the step size bound for the i -th coefficient in the systolic FxLMS structure is found as (proven in Appendix. C.1)

$$0 < \mu < \frac{2}{\lambda_k} \sin \frac{\pi}{2(2i+1)}. \quad (6.32)$$

When the tap length of the adaptive filter is sufficiently long, the step size bound of the systolic FxLMS is approximated by

$$0 < \mu < \frac{\pi}{\lambda_{\max}(2N+1)} \quad (6.33)$$

where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{R}_{x'}$.

6.3.3 Implementation of systolic FxLMS algorithm

The systolic FxLMS structure, as shown in Fig. 6.12, can be divided into N identical processing modules (PMs). Pipelining is applicable when PMs are arranged as Fig. 6.13. The systolic FxLMS structure allows an arbitrary number of PMs to be inserted into the pipeline. Hence, the implementation of the systolic FxLMS structure can be easily scaled up as long as there are sufficient hardware resources.

The block diagram of the PM is elaborated in Fig. 6.14. The PM consists of three adders, three multipliers, four registers, one gate, and a few switches. All of those

FIGURE 6.13: Block diagram of the systolic FxLMS structure with pipelining of processing modules.

components are basic elements of the FPGA. It should be noted in Figs. 6.12 and 6.13 that the rightmost PM, i.e. PM0, is different from the rest of PMs. Therefore, there is an endpoint defined in the PM, namely the PM0 endpoint. When this endpoint is activated, two unit delays blocks are bypassed.

6.3.4 Simulation: systolic FxLMS algorithm

The simulation is conducted using the MATLAB Simulink environment. The primary noise is Gaussian noise with zero mean and unit variance. The primary path is a bandpass finite impulse response (FIR) filter. When the tap length is longer, the attenuation of stopband in the primary path is higher, as shown in Figs. 6.15 (a) and (b). The secondary path is a pure delay. When the tap length is 40, the delay is 5 taps. When the tap length is 200, the delay is 15 taps. They are shown in Figs. 6.15 (c) and (d).

Figure 6.16 displays the convergence curves of the FxLMS algorithm, delay FxLMS algorithm, and systolic FxLMS structure when the length of the adaptive filter is 40 taps. The step size is increased from 0.0025 to 0.03 to demonstrate four different cases. In Fig. 6.16 (a), when the step size is sufficiently small, the FxLMS algorithm, delay FxLMS algorithm, and systolic FxLMS structure converge at almost the same speed. In Fig. 6.16 (b), the FxLMS algorithm is the fastest, and the delay FxLMS algorithm is slower than the systolic FxLMS structure. When the step size continues to increase, the delay FxLMS algorithm is the first to diverge, as

FIGURE 6.14: Block diagram of the processing module.

shown in Fig. 6.16 (c). Furthermore, it is implied by Fig. 6.16(d) that the FxLMS algorithm has the broadest step-size bound.

Figure 6.17 shows the results when the simulation is conducted using a 200-tap adaptive filter and 15-tap secondary path. The general trends in Fig. 6.17 are similar to those in Fig. 6.16. The convergence behavior of the systolic FxLMS structure lines between the FxLMS and delay FxLMS algorithms. Hence, the systolic FxLMS structure is demonstrated to converge faster and possess a more relaxed step-size bound as compared to the delay FxLMS algorithm.

FIGURE 6.15: Primary and secondary paths used in the simulation.

6.3.5 Implementation result of systolic FxLMS algorithm

When compared to the FxLMS algorithm, the advantage of the systolic FxLMS structure is the shorter critical path and higher maximum sampling rate. To confirm the difference between the performance of the systolic FxLMS and FxLMS algorithm, we use the 2015 version of the Vivado Design Suite software to carry out the FPGA implementation on an educational Xilinx XC7Z010 development platform. The implementation steps and results are posted on the website of Online Tutor for Engineering Students (ONTEN: <http://onten.ddns.net/>), which is a web-service learning platform containing student-friendly, easy-to-navigate review modules for students in engineering [147].

Figure 6.18 shows the processing time achieved by the FxLMS algorithm and systolic FxLMS structure. The secondary path has the same tap length as that of the adaptive filter. It is clearly seen that the processing time of the FxLMS algorithm increases with the tap length, while the systolic FxLMS structure has a constantly low processing time of 24.65 ns.

FIGURE 6.16: Convergence curves of the FxLMS algorithm, delay FxLMS algorithm, and systolic FxLMS structure, when the tap length is 40, using the step size of (a) 0.0025; (b) 0.02; (c) 0.023; (d) 0.03.

Therefore, the systolic FxLMS structure with its pipelining compatibility is highly efficient to be implemented on the FPGA. Hence, the parallel processing capability, system throughput, and scalability are significantly improved. This will benefit feedforward control in a noise canceling headphone and eventually a multichannel ANC system. Furthermore, the step-size bound of the systolic FxLMS structure has been found to be narrower than the FxLMS algorithm but more relaxed than the delay FxLMS algorithm in theory.

FIGURE 6.17: Convergence curves of the FxLMS algorithm, delay FxLMS algorithm, and systolic FxLMS structure, when the tap length is 200, using the step size of (a) 0.00155; (b) 0.0025; (c) 0.0045; (d) 0.006.

6.4 Practical implementation of feedforward multichannel active noise control

Multichannel active noise control (MCANC) is widely recognized as an effective and efficient solution for acoustic noise and vibration cancellation, such as in large-dimension ventilation ducts, open windows, and mechanical structures [16]. The feedforward multichannel filtered-x least mean square (FFMCFxLMS) algorithm is commonly used to dynamically adjust the transfer function of the multichannel controllers for different noise environments. The computational load incurred by the FFMCFxLMS algorithm, however, increases exponentially with increasing channel count and thus requiring high-end field-programmable gate array (FPGA) processors. Nevertheless, such processors still need specific configurations to cope with soaring computing loads as the channel count increases. To achieve a high-efficiency implementation of the FFMCFxLMS algorithm with floating-point arithmetic, a

FIGURE 6.18: Processing time of the FxLMS algorithm and systolic FxLMS structure on an Xilinx XC7Z010 FPGA.

novel architecture, the multiple-parallel-branch with folding (MPBF), is proposed. This architecture parallelizes the branches and reuses the multiplier and adder in each folded branch so that the trade-off between throughput and the usage of the hardware resources is balanced.

Figure 6.19 shows the block diagram of the MCANC, which consists of J reference sensors, J secondary sources, and M error sensors. By using the stacked coefficient vector of the J control filters $\mathbf{w}(n) = [\mathbf{w}_1^T(n), \mathbf{w}_2^T(n), \dots, \mathbf{w}_J^T(n)]^T$, where $\mathbf{w}_j(n) = [w_{j,1}(n), w_{j,2}(n), \dots, w_{j,N}(n)]^T$ is the coefficient vector of the j th control filter, the MCANC creates J control signals $\mathbf{y}(n) = [y_1(n), y_2(n), \dots, y_J(n)]^T$. These control signals propagate through acoustic secondary paths to cancel the noise disturbance $\mathbf{d}(n) = [d_1(n), d_2(n), \dots, d_M(n)]^T$ at the error sensor locations. The transfer function $\mathbf{P}$ denotes the primary paths from the primary source to the error sensors, and the transfer function $\mathbf{S}$ represents the physical secondary paths from the secondary sources to the error sensors, and its estimate is denoted as $\hat{\mathbf{S}}$. The

FIGURE 6.19: A $J \times J \times M$ MCANC using the FxLMS algorithm

matrix of reference signals picked up by J reference sensors is written as

$$\mathbf{X}(n) = \text{diag} \{ \mathbf{x}_1(n), \mathbf{x}_2(n), \dots, \mathbf{x}_J(n) \} \quad (6.34)$$

where $\text{diag}\{\cdot\}$ denotes the diagonal elements of a matrix, and the j th reference signal vector is stated as $\mathbf{x}_j(n) = [x_j(n), x_j(n-1), \dots, x_j(n-N+1)]^T$. Hence, the output signal vector of J secondary sources is given by

$$\mathbf{y}(n) = \mathbf{X}(n)\mathbf{w}(n). \quad (6.35)$$

In the FFMCFxLMS algorithm, the weight updating equation of the j th control filter is given by [9]

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \mu \sum_{m=1}^M \mathbf{x}'_{jm}(n) e_m(n) \quad (6.36)$$

where μ denotes the step size, and $e_m(n)$ represents the error signal picked up by the m th error sensor. The filtered reference signal is expressed by $x'_{jm}(n) = \hat{s}_{mj}(n) * x_j(n)$, where $\hat{s}_{mj}(n)$ is the estimate of the secondary path from the j th secondary source to the m th error sensor, and $x_j(n)$ is the reference signal acquired by the j th reference sensor.

By observing (6.35) and (6.36), a $J \times J \times M$ FFMCFxLMS algorithm requires $J \times N \times 2(M+1)$ additions and multiplications in each iteration. For instance,

a $24 \times 24 \times 24$ MCANC based on the FFMCFxLMS algorithm with a sampling rate of 25 kHz and filter lengths of $N = 200$ (the control filter and secondary path estimate) amounts to 6 billion MAC operations per second. In comparison, the high-end Texas Instruments TMS320C674X DSP can only handle up to 2.7 billion MAC operations per second [148]. Hence, the FPGA is better equipped to realize MCANC with its parallel operations that potentially allows all the arithmetic computations to be completed at the same time. However, without appropriate optimization, FPGA would consume a tremendous amount of hardware resources to implement the FFMCFxLMS algorithm, and it is thus impractical to scale.

6.4.1 Multiple-Parallel-Branch with folding Architecture

The challenge of implementing the FFMCFxLMS algorithm on FPGA is to achieve both high throughput rate and low hardware resource consumption. To address this issue, we propose a new architecture for the FFMCFxLMS algorithm, which integrates both the parallel and folded structures that improve throughput and reduce hardware resources, respectively.

6.4.1.1 Parallel Branch

To parallelize the FFMCFxLMS architecture, we propose to split the $J \times J \times M$ multichannel FxLMS algorithm into J identical branches. Each branch includes a control filter, a weight updating block, and $J \times M$ secondary path model blocks. These branches run in parallel simultaneously, which yields J times faster processing speed and higher throughput than sequential execution by one single processor.

6.4.1.2 Recursive formula derivation

In this architecture, we let the branch execute N (the length of control filter is N) iterations to complete one computing cycle¹, which reduces the hardware resource usage of the algorithm on FPGA. The recursive formula of the FFMCFxLMS algorithm was derived in the z -transform domain.

¹One computing cycle: the duration that the filter processes one input data. Here, 1 computing cycle = N iterations, and 1 iteration = $M + 1$ clock periods.

FIGURE 6.20: Block diagram of the j th control filter.

The block diagram of the control filter in the j th branch is illustrated in Fig. 6.20. $X_j(z)$ denotes the z -transform of the j th reference signal, and $W_{j,l}(z)$ represents the l th weight in the j th control filter. Hence, the recursive formula of the j th control filter's output can be expressed as

$$Y_j(z) = Y'_{j,N}(z) = Y'_{j,N-1}(z) + W_{j,N}(z)X_j(z)z^{-N+1} \quad (6.37)$$

where $Y'_{j,l}(z)$ is the output of the l th iteration and given by

$$Y'_{j,l}(z) = Y'_{j,l-1}(z) + W_{j,l}(z)X_j(z)z^{-l+1} \quad (6.38)$$

and the first iteration output is stated as

$$Y'_{j,1}(z) = W_{j,1}(z)X_j(z). \quad (6.39)$$

Here, $j \in [1, J]$ and $l \in [1, N]$. By using (6.38) and (6.39), we can transform one computing cycle of the j th control filter into N iterations, which is indicated by dashed lines in Fig. 6.20.

The weight updating equation of the j th control filter is given by (6.36), the l th element of which can be rewritten as its z -transform

$$W_{j,l}(z) = \left(W_{j,l}(z) + \mu \sum_{m=1}^M X'_{jm}(z)z^{-l}e_m \right) z^{-1} \quad (6.40)$$

where e_m and $X'_{jm}(z)$ denote the m th error signal and the filtered reference signal, respectively.

FIGURE 6.21: Block diagram of the reference signal filtering.

The transposed filter is utilized to realize the filtering of the reference signal, which increases the throughput compared to the conventional transversal filter[131, 149]. Each branch possesses M secondary path models. The block diagram of the m th secondary path mode in the j th branch is illustrated in Fig. 6.21, in which $\hat{s}_{mj}(i)$ represents the i th coefficients of the secondary path estimate from the j th secondary source to the m th error microphone, and $i \in [0, N-1]$. Hence, the recursive formula of the filtered reference signal is given by

$$X'_{jm}(z) = X''_{jm}(0) = \hat{s}_{mj}(0)X_j(z) + X''_{jm}(1)z^{-1} \quad (6.41)$$

where

$$X''_{jm}(1) = \hat{s}_{mj}(1)X_j(z) + X''_{jm}(2)z^{-1} \quad (6.42)$$

analogously, the output of the l th iteration can be expressed as

$$X''_{jm}(i) = \hat{s}_{mj}(i)X_j(z) + X''_{jm}(i+1)z^{-1} \quad (6.43)$$

and the final iteration result is stated as

$$X''_{jm}(N-1) = \hat{s}_{mj}(N-1)X_j(z). \quad (6.44)$$

By combining (6.38), (6.40) and (6.43), we outline the recursive formula for one computing cycle of the $J \times J \times M$ FFMCFxLMS algorithm, whose block diagram is shown in Fig. 6.22. In the figure, the multichannel algorithm is divided into J parallel branches, and each branch consists of $2(M+1)$ multipliers and adders.

In summary, the J branches require N iterations to complete one computing cycle of the algorithm. Hence, compared to the original implementation structure of the

FIGURE 6.22: Block diagram of the FFMCFxLMS based on J branches.

FFMCFxLMS algorithm, the proposed architecture utilizes N times less hardware resource at this stage.

6.4.1.3 Folded structure derivation

In Fig. 6.22, each branch of the FFMCFxLMS algorithm consists of many identical operators (adders and multipliers). Hence, we can reuse some of them to complete all the computations in each branch, which further reduces the hardware resource usage of the FFMCFxLMS algorithm on FPGA. To derive the folded structure of

each branch, we adopted a modified folding method as described in the following procedure:

1. Assigning the reused unit. The redrawn data-flow graph (DFG) of the j th branch is shown in Fig. 6.23. In the DFG, there are $2(M + 1)$ additions and multiplications, which can be executed by two MAC units, S_0 and S_1 , according to a particular running sequence or folding order. S_0 is assigned to complete $M + 1$ additions and multiplications of the weight updating in the branch; S_0 is used to complete the other $M + 1$ additions and multiplications coming from the control filter and reference signal filtering tasks, as shown in Fig. 6.23.
2. Scheduling folding order. The folding order refers to the executing sequence of the MAC. In the paper, the folding order corresponds to the data flow direction in the DFG. The operation label $(S_0|m)$ in Fig. 6.23 denotes that this operation will be executed by S_0 in the m th period.
3. Labeling the ports of nodes. We notated all the adding and multiplying operations as $2(M + 1)$ nodes in arbitrary sequence, as shown in Fig. 6.23. Next, the ports of the node are labeled as A, B, C, D, E, and H, which indicate the connections between different nodes.
4. Calculating the new delay time. To derive the new delay time in the folded structure of Fig. 6.23, we utilized the folding equation (6.10). The derived delays are listed in Table 6.2. It is worth noting that the delay of all the data paths equate to one sampling period, as a result of the folding order of the DFG being set along the direction of data flow, as illustrated in Fig. 6.23.
5. Constructing the folded structure. According to the new delay time table, the folded structure of the j th branch can be constructed, as illustrated in Fig. 6.24. The label $\{m\}$ on the data route indicates that the switch connects to this route in the m th clock period, and l denotes the l th iteration. These labels construct a specific execution schedule for the switches. Accordingly, the switches in the folded structure select different routes based on its schedule so that the MAC can sequentially process the data to complete one iteration. These switches can be readily implemented by the multiplexers and de-multiplexers in the digital circuit. Between the adjacent iterations,

FIGURE 6.23: Block diagram of the FFMCFxLMS based on J branches.

TABLE 6.2: The new delay on each data path.

Path $D_F(U \rightarrow V)$	Folding equation	Delay time
$M + 1 \rightarrow M + 2$	$(M + 1) \times 0 + M + 2 - M - 1$	1
$M + 2 \rightarrow M + 3$	$(M + 1) \times 0 + M + 3 - M - 2$	1
$M + 3 \rightarrow M + 4$	$(M + 1) \times 0 + M + 4 - M - 3$	1
$\dots$	$\dots$	1
$2M \rightarrow 2M + 1$	$(M + 1) \times 0 + 2M + 1 - 2M$	1

*Note: One delay time is equivalent to one sampling period.

the data communication is realized by the dual-port block random-access memories (BRAM), which will be explained in detail in the next section.

FIGURE 6.24: The folded structure of the j th branch.

6.4.2 Implementation of MPBF architecture on FPGA.

By integrating J parallel branches with the folded structures as described above, a new architecture for the $J \times J \times M$ FFMCFxLMS algorithm is obtained, as illustrated in Fig. 6.25. In implementing the proposed architecture on FPGA, two

FIGURE 6.25: Proposed architecture of the FFMCFxLMS algorithm.

critical issues have to be first addressed: realizing the control of the switches in the folded structure, and implementing the data communication and storage between iterations.

6.4.2.1 Finite-state machine to control the switches

With the assistance of the switches in the branch, two MACs are reused $M + 1$ times to complete one iteration according to the folding order in Fig. 6.24. In the proposed architecture, we used a Mealy finite-state machine (FSM) to control the switches in charge of the data communication between MAC and ports during every clock period. The FSM comprises of $M+1$ states and outputs the control signals through the seven control lines (ctr-a to ctr-h), as shown in Fig. 6.25. The control signals at different states are listed in a statechart diagram, as shown in Fig. 6.26. These control signals decide the inter-connections of the multiplexers and

FIGURE 6.26: Statechart diagram of the finite state machine.

de-multiplexers to realize the function of the switches, as illustrated in Fig. 6.24. To avoid the risk of competition [150] and to obtain enough setup time and hold on time for the input data, the MAC in architecture uses CLK/S as the clock signal, which is generated by shifting the system clock CLK_S by 180°, as illustrated in Fig. 6.27. In the FSM, the ‘iteration’ signal indicates the index of the iteration. Since the branch requires N iterations to complete one computational cycle, an N mode counter is used to create the ‘iteration’ signal. The output of the multiplexers and de-multiplexers during each clock period is exhibited in Fig. 6.27.

Therefore, during one iteration, the j th branch picks up the j th reference signal $x_j(n)$ and all the error signals from $e_1(n)$ to $e_M(n)$, as shown in Fig. 6.25. Subsequently, it feeds these data into two MACs in the specific sequence through the ports (A, B, C, and E) based on the described mechanisms. After $M + 1$ clock periods, it obtains the iteration results from the ports D and H. The read-write timing of the ports are elaborated in Fig. 6.28. Meanwhile, other parallel branches execute the same procedure simultaneously.

FIGURE 6.27: Read and write timing of the ports.

6.4.2.2 Data communication between iterations

After every iteration, the branch needs to store the computed result and fetch new data for the next iteration. Again, we can utilize the folding technique to schedule the data communication sequence between iterations. The folded structure used between iterations, however, requires a significant amount of registers whose quantity is limited by the FPGA [151] used. The workaround still requires a complicated control scheduler and a large number of registers, even when combining with retiming and lifetime analysis techniques [131, 152]. Therefore, in this implementation we replaced the registers with BRAMs, which are abundant and under-utilized in the FPGA. Although BRAM runs slower than the register, the data communication between iterations happens at the interval of $M + 1$ system clocks, which coincides with the reading and writing frequencies of BRAM.

FIGURE 6.28: Read and write timing of BRAM.

In the architecture, each branch utilizes $3M + 2$ dual-port BRAMs to realize data communication and temporary storage between iterations. Among them, the BRAM $\mathbf{x}_j$, $\mathbf{x}'_{jm}$, and $\mathbf{x}''_{jm}$ implement the delay lines in the DFG of the j th branch. Meanwhile, the BRAM $\mathbf{w}_j$ and $\hat{\mathbf{s}}_{mj}$ store the coefficients of the j th control filter and the secondary path model from the j th secondary source to the m th error sensor. The address signals (A1, B1, and C1) of these BRAMs are generated by three N mode counters, and A1 is always greater than B1 by 1, as illustrated in Fig. 6.28. Furthermore, there are two clocks CLK_R and CLK_W driving these BRAMs, in the architecture. The CLK_R, having $M + 1$ periods of the system clock CLK_S, drives both reading and writing operations of the BRAM $\mathbf{w}_j$, $\hat{\mathbf{s}}_{mj}$, $\mathbf{x}''_{jm}$, and the reading operation of the BRAM $\mathbf{x}_j$ and $\mathbf{x}'_{jm}$, as shown in Fig. 6.27. The clock CLK_W, whose period is N times that of CLK_R, is used to trigger the writing operation of the BRAMs $\mathbf{x}_j$ and $\mathbf{x}'_{jm}$. With the help of the address signal C1 and the reading and writing clock, as shown in Fig. 6.28, the BRAMs $\mathbf{x}_j$ and $\mathbf{x}'_{jm}$ realize the delay lines by the circular shifting operation, which avoids the shifting operation of every input data and thus reduces the power consumption of the FPGA.

6.4.2.3 Clock divider

As mentioned in the previous subsection, there are three derived clock signals, CLK/S, CLK_R, and CLK_W, of system clock CLK_S in the architecture. They need to be synchronized but running at different frequencies. To synthesize these derived clocks on FPGA, we implemented a novel divider, as shown in Fig. 6.29.

FIGURE 6.29: The schematic diagram of the divider.

It comprises of a 32-bit accumulator and two control ports: an input clock port and a frequency control port. During operation, the input clock signal drives the accumulator to increase gradually at the step size of a frequency tuning word (FTW), which is loaded from the frequency control port. Once the value of the accumulator exceeds 2^{31} , the most significant bit (MSB) of the accumulator will toggle its original value to create the desired output clock with a 50% duty cycle.

This divider works like the direct digital synthesizer (DDS) [153], in which the frequency f_{out} of the output clock is decided by the FTW. Hence, the relationship between f_{out} , FTW, and the frequency of the system clock f_{CLK_S} is given by

$$f_{\text{out}} = \left\lceil \frac{\text{FTW}}{2^{32}} \right\rceil \times f_{\text{CLK}_S} \quad (6.45)$$

where FTW is a 32-bit integer from 0 to $2^{31} - 1$. Therefore, the FTW for CLK_R and CLK_W can be derived to $\text{round}\left[\frac{2^{32}}{2M+2}\right]$ and $\text{round}\left[\frac{2^{32}}{N(2M+2)}\right]$ from (6.45), where the $\text{round}[\cdot]$ function rounds the argument to the nearest integer.

Altogether, the folded structure, BRAMs, and the clock divider of the proposed architecture implemented on FPGA requires $N(M+1)$ clock periods to accomplish one computing cycle of the $J \times J \times M$ FFMCFxLMS algorithm. However, the reduction of MACs by $N(M+1)$ times and owing to the J parallel branches, a faster processing speed is still achieved over the sequential structure.

TABLE 6.3: Device utilization of the multichannel FxLMS algorithm on Kintex-7 7K235T.

Item	Ratio of the device utilization
Total slices	97.1% (49494 out of 50950)
Slice registers	44.8% (182665 out of 407600)
Slice LUTs	63.9% (130214 out of 203800)
Block RAMs	36.2% (161 out of 445)
DSP48s	57.1% (480 out of 840)

6.4.3 Implementation results

To verify the proposed architecture, we implemented a $24 \times 24 \times 24$ floating-point FFMCFxLMS algorithm which comprised of 24 control filters and 576 secondary path estimates. The algorithm was deployed on the National Instruments (NI) 9039 Compact RIO (with Xilinx Kintex-7 7K235T FPGA as the main processor) equipped with NI 9220 ADC and NI 9264 DAC modules. In this architecture, we used 24 parallel branches and set the length N of both the control filters and the secondary path estimates to 200 taps.

A summary of the device utilization after compilation on Vivado v2015.4 is shown in Table 6.3. It is evident that the proposed architecture balances the usage of MAC, registers, and BRAMs. In comparison, the conventional FFMCFxLMS requires $2JN(M + 1) = 28800$ MACs, whereas the proposed architecture demonstrated a significant savings of $2JN(M + 1) - 2J = 28752$ MACs.

Since the architecture requires $N(M + 1) = 5000$ clock periods to complete one computing cycle, the maximum achievable sample rate f_s was 25 kHz when the FPGA runs with a system clock of 125 MHz. The maximum floating-point operations per second (FLOPS) of the architecture can be computed as

$$\text{FLOPS} = 2 \times J \times N \times 2(M + 1) \times f_s = 12 \times 10^9. \quad (6.46)$$

As each branch receives 1 reference (16 bits), 24 error (16 bits), and 1 step size (32 bits) inputs, the throughput rate of the architecture is given by

$$R_{\text{th}} = 24 \times (25 \times 16 + 32) \times f_s = 260 \times 10^6. \quad (6.47)$$

6.4.4 Real-time active control of noise through the open window

FIGURE 6.30: The front view of the 24-channel ANC window.

To evaluate the performance of the $24 \times 24 \times 24$ MCANC system, active control units were distributed across the open aperture of a $1 \text{ m} \times 0.5 \text{ m}$ sliding window mounted on the wall of a $2.1 \text{ m} \times 2.1 \text{ m} \times 2.1 \text{ m}$ cuboid room, as shown in Fig. 6.30 and Fig. 6.31. The 24 identical ANC units were attached to a security grille. Each unit consists of a 4.5 cm diameter loudspeaker facing into the room interior at one end and a reference electric microphone facing the noise source at the other end, 12 cm apart, as illustrated in Fig. 6.32. The interior of the ANC unit is filled with dampening material to reduce acoustic feedback from the loudspeaker to the reference microphone. The error microphone array is placed in the room, 20 cm away from the ANC units. The primary noise to be attenuated is generated by a loudspeaker (GENELEC 8341a) mounted 2 m away from the window.

To evaluate the noise reduction performance, the energy-average sound pressure level (SPL) is calculated from five observation microphones (G.R.A.S. 40PH) distributed in the room interior according to ISO 16283 [154], as depicted in Fig. 6.33.

FIGURE 6.31: The schematic diagram of the 24-channel ANC window.

FIGURE 6.32: The cross-sectional view of the ANC unit.

FIGURE 6.33: The positions of the observation microphones.

The connections between the 24-channel ANC units and the FPGA system is described in Fig. 6.34. The reference and error signals picked up by the microphones are conditioned by their preamplifiers and anti-aliasing filters, and are subsequently converted to digital signals with a data-width of 16 bit by the NI-9220 ADC modules. The ADC modules transmit the data to the FPGA (Kintex-7 7K325T) via the PXIe bus. After computation, the FPGA transmits the output data to the NI-9264 DAC modules that have a maximum data update rate of 25 kHz. Finally, with the aid of the DACs, reconstruction filters, and audio amplifiers, the output data are converted back to analog signals that drive the secondary sources to generate the anti-noise waves.

6.4.5 Real-time experimental result

In the first experiment, tonal noise from 100 Hz to 2500 Hz at intervals of 100 Hz were the primary disturbances to be controlled. The time-averaged SPL of the monitoring microphone A, located 20 cm away from the window, with the MCANC was turned on and off, is illustrated in Fig. 6.35. The maximum noise reduction of the 24-channel ANC was 35.1 dB at 1800 Hz; the minimum noise reduction was 3.6 dB at 2400 Hz.

The SPL at monitoring microphone B, positioned in the center of the room, is shown in Fig. 6.36 (a) for the same range of tonal signals. A maximum noise

FIGURE 6.34: The implementation diagram of the 24-channel ANC system.

FIGURE 6.35: The time-averaged SPL of the monitoring microphone A as a function of frequency in the active control of tonal disturbances.

reduction was recorded at 32.2 dB for 1600 Hz, and the minimum noise reduction was 0.7 dB at 2300 Hz. Fig. 6.36 (b) depicts the SPL of monitoring phone C, where its maximum noise reduction was 19.9 dB at 2100 Hz. However, its SPL increased by 12.6 dB at 2300 Hz when MCANC was turn on. The SPL of the monitoring microphone D is shown in Fig. 6.36 (c). Its maximum noise reduction was 12.9 dB at 1900 Hz. Its SPL increased at higher frequencies. Finally, the SPL of monitoring microphone E, sheltered by the glass panel of the sliding window, is shown in Fig. 6.36 (d). When the MCANC system was turned on, the noise level reduced slightly and even increased at higher frequencies. The poor performance at higher frequencies can be attributed to the fundamental limits of control of the MCANC system in an open aperture [155, 156].

Figure 6.37 shows the energy-average noise reduction level (NR) of microphone A, B, C, D, and E for tonal noises at different frequencies, which is calculated by [154]

$$\text{NR}_{\text{average}} = 10 \log \left[\frac{\sum_{i=A}^{A,B,C,D,E} p_{\text{OFF},i}^2}{\sum_{i=A}^{A,B,C,D,E} p_{\text{ON},i}^2} \right] \quad (6.48)$$

where $p_{\text{OFF},i}$ and $p_{\text{ON},i}$ denote the SPL at the location of the microphone i when

FIGURE 6.36: The SPL of monitoring microphone B, C, D, E for sine tones.

ANC is turned off and on, respectively. The MCANC system has the maximum energy-average noise reduction level of 17.5 dB at 600 Hz but negative noise reduction level at 1300 Hz. As these results indicate, the ANC window can achieve better global noise reduction of the room at lower frequencies.

In the second experiment, the primary disturbance was a bandlimited broadband noise from 500 Hz to 2000 Hz. Fig. 6.38 shows the power spectrum of the broadband noise at the monitoring microphone A when MCANC was turned off and on. It is evident that the noise in the band was reduced, and the noise reduction level was 10.1 dB. Furthermore, the noise at other locations was also mitigated by the ANC window, as listed in Table 6.4. Therefore, the 24-channel ANC window can yield global reduction for the broadband noise from 500 Hz to 2000 Hz.

The results from both experiments validate that the FPGA implementation of

FIGURE 6.37: The energy-average noise reduction level of microphone A, B, C, D, and E.

FIGURE 6.38: The power spectrum of the broadband noise at the monitoring microphone A.

TABLE 6.4: Broadband noise reduction level at the locations of the monitoring microphones.

Position	A	B	C
Reduction	10.1 dB	9.5 dB	7.2 dB
Position	D	E	Energy-average
Reduction	3.6 dB	3.5 dB	8.4 dB

the 24-channel FFMCFxLMS with the proposed architecture was implemented successfully in a complex acoustic scenario of the open window. Global control of broadband noise was achieved in a wide frequency range.

6.5 Conclusions

This chapter introduces the programmable logic device (PLD), which is widely utilized in digital signal processing field, in section 6.1 and further revisits several optimization structures, which effectively increased the maximum operating frequency or saved the hardware resource usages in computations, in section 6.2.

Section 6.3 proposes a systolic FxLMS structure that was compatible with pipelining, which is highly efficient to be implemented on the FPGA. Due to the parallel processing capability, the system throughput and scalability are greatly improved. This will benefit the feedforward control in the noise canceling headset and eventually the multiple channel ANC. The step size bound of the systolic FxLMS structure has been found to be narrower than the FxLMS algorithm but more relaxed than the delay FxLMS algorithm in theory. Simulation results have validated the trend of the theoretical bound.

Section 6.4 suggests a multiple-parallel-branch with folding floating-point architecture for the implementation of the FFMCFxLMS algorithm on FPGA. In realizing a $J \times J \times M$ FFMCFxLMS algorithm with the filter length of N taps, the proposed architecture requires only $2J$ MACs, which equates to a hardware resource reduction by $N(M+1)$ times. The parallel branches in the architecture enable the FPGA to execute J adaptive filters simultaneously, which increases the processing speed or the throughput rate by J times. So, with the aid of the folded and parallel structures, the proposed architecture achieves a reasonable trade-off between

the lower hardware resource utilization and higher throughput of the system. Furthermore, the proposed architecture does not alter, nor reduce the operations of the FFMCFxLMS algorithm, and thus, will not compromise on its performance. Moreover, a $24 \times 24 \times 24$ FFMCFxLMS algorithm based on the proposed architecture on an FPGA has been implemented for real-time control of noise through a $1 \text{ m} \times 0.5 \text{ m}$ window opening into the room interior. The experimental results show that a maximum tonal noise reduction of 35.1 dB and broadband noise reduction of 10.1 dB from 500 Hz to 2000 Hz was achieved directly in front of the window. The energy-average noise reduction levels in the room for the tonal and broadband noise were 17.6 dB and 8.4 dB, respectively. Furthermore, the proposed architecture achieved a computation performance of 12 billion floating-point operations per second (FLOPS) on the FPGA of Xilinx Kintex-7.

Chapter 7

Conclusions and future work

7.1 Conclusions

This thesis reexamined and proposed adaptive algorithms and implementation architectures to overcome the practical issues, such as the computational complexity, robustness, and processing speed, in active noise control applications. Corresponding numerical simulations and real-time experiments were carried out to verify the effectiveness of these proposed algorithms and structures.

In Chapter 1, the effect of the urban noise issue is discussed. The common techniques to deal with this issue are passive and active noise cancellation. Compared to the passive approach, the active noise control technique shows the apparent advantages of low cost, relatively compact size, and high efficiency in low-frequency noise reduction. Active noise control contains different structures, such as feed-forward, feedback, and multichannel structures, to cope with different application requirements.

In Chapter 2, the adaptive filtering theory has been briefly examined for the active noise control application. Among these adaptive algorithms, the filtered-x least square (FxLMS) algorithm originated from the least mean square (LMS) algorithm successfully compensates the effect of the secondary path and is widely used in ANC. Other modified algorithms are also mentioned to improve the performance in terms of computational complexity, convergence speed, and steady-state noise reduction level.

In Chapter 3, the saturation effect of the active noise control system was discussed. The investigation shows that the amplitude and phase distortion coming from the saturated state of the audio amplifier not only affects the noise reduction performance but also deteriorates the stability of the adaptive algorithm. To solve the output saturation problem, this chapter proposes the two-gradient FxLMS (2GD-FxLMS) algorithm, which automatically switches the updating direction of the control filter coefficients to search for the optimal solution while restraining the output power so that the system operates linearly. Compared to other nonlinear adaptive algorithms, the 2GD-FxLMS algorithm only consumes the same amount of computations as the FxLMS algorithm. The alternative approach is to use the optimal leaky factor FxLMS algorithm, which confines the amplitude of the control filter in a specific range while achieving the maximum noise reduction level by using a pre-estimate optimal leak factor. Moreover, the calculation of this optimal leak factor does not involve any amplifier model and thus, is more feasible in practice.

In Chapter 4, the virtual sensing technique used in active noise control was retrospectively. The single-channel virtual seeing active noise control (VS-ANC) has been extended into the multichannel virtual sensing ANC (MVANC) implementation, although this has been done previously. The frequency- and time-domain analysis in this chapter proves that MVANC is capable of achieving the optimal control at desired positions where it does not place any real error microphone. Furthermore, the over- and fully determined analysis of the system reveals the limitation of the MVANC method where the secondary sources cannot be more than physical microphones.

In Chapter 5, a novel method, selective fixed-filter active noise control (SFANC), was proposed. The proposed SANC algorithm efficiently alleviates the computational complexity of the adaptive process by using the pre-trained control filter, which is trained in a preliminary stage and chosen based on the frequency-band-match approach. The analysis shows that the control filter chosen by the frequency-band-match method has comparable noise reduction performance as the FxLMS algorithm. Since the SANC algorithm does not contain adaptive updating process as the adaptive algorithm, it maintains a good performance in term of the response time, stability, and steady-state noise reduction.

In Chapter 6, the programmable logic device (PLD) and some optimization strategies of digital processing algorithm have been revisited. Based on systolic technique, this chapter proposes the systolic FxLMS algorithm, which successfully transforms the conventional FxLMS algorithm to a fully pipelined structure. Hence, the systolic FxLMS algorithm achieves higher processing speed and sampling rate by slightly sacrificing its convergence speed. By integration with parallel and folded structure, the multiple-branch parallel and folding architecture has been developed for the MCANC implementation on an FPGA. The architecture using parallel and folding techniques seeks a balance between the extremely large hardware resource usage and the throughput rate of the multichannel FxLMS algorithm. A $24 \times 24 \times 24$ channel MCANC based on this architecture has been realized on an FPGA and is tested in an open window. The noise reduction performance of the noise propagating through the window validates the effectiveness of the proposed architecture.

7.2 Future works

For the 2GD-FxLMS algorithm, the theoretical analysis and real-time experiments described in this thesis have proven its effectiveness and advantages. However, there remain questions about the step-sizes of the 2GD-FxLMS algorithm that need to be answered. Its requirement of two step-sizes can potentially affect the convergence rate and steady-state performance of the algorithm if not initiated appropriately. Hence, further investigation into the step-size selection is necessary.

Moreover, state of the art nonlinear adaptive algorithms mostly focuses on the single-channel active noise control application, due to their vast computational load. Since the 2GD-FxLMS algorithm has the same computational complexity as the conventional FxLMS algorithm, there is a possibility for extension to multichannel ANC (MCANC).

The derivation of the optimal leak selection strategy is under some assumption that the reference signal is white noise or the narrow-band noise. However, in practice, we usually need to deal with broadband noise. Hence, it should be more interesting to extend this method for broadband noise cancellation.

The selective fixed-filter active noise control (SFANC) technique proposed in the thesis is only suitable for the single-channel ANC application. However, the computational complexity is the real urgent issue for the implementation of the multichannel ANC system. Hence, the SFANC algorithm is necessary to be modified for the MCANC so as to alleviate the computational burden the system.

In the thesis, the effectiveness and limitation of the multichannel virtual-sensing ANC technique have been discussed. However, there are still some interesting issues remaining to be investigated, such as spatial independence between the positions of the physical and virtual microphones. In future work, we plan to prove this argument based on the Green function and acoustic spherical harmonic theory. Furthermore, It is more interesting to integrate the SFANC and MVANC technique in future.

For the new architecture of the MCANC implementation, we will use it to implement the larger scale ANC system to verify its effectiveness. Furthermore, we will try other methods to optimize this architecture to improve its sampling rate or reduce hardware source usage. It would also be interesting to extend this technique to arbitrary numbers of reference sensors, error sensors, and secondary sources in the future.

Appendix A

Proofs of part I

A.1 Optimal solution of extended leaky FxLMS algorithm

The residual error signal of active noise control is given by

$$e(n) = d(n) - \sum_{l=0}^{L-1} s_l \mathbf{w}^T(n-l) \mathbf{x}(n-l). \quad (\text{A.1})$$

If we extend the leaky FxLMS algorithm as

$$\begin{aligned} J(n) &= e^2(n) + [\mathbf{v}\mathbf{w}(n)]^T [\mathbf{v}\mathbf{w}(n)] \\ &= e^2(n) + \mathbf{w}^T(n) \cdot \gamma \cdot \mathbf{w}(n) \end{aligned} \quad (\text{A.2})$$

where $\mathbf{v}$ is a vector used to constrain the control filter $\mathbf{w}(n)$, and $\gamma = \mathbf{v}^T \mathbf{v}$. By setting the gradient of the mean of (A.2) to zero

$$\frac{\partial E[J(n)]}{\partial \mathbf{w}(n)} = -2E[d(n)\mathbf{x}'(n)] + 2\mathbf{w}(n) \cdot E[\mathbf{x}'(n)\mathbf{x}'(n)^T] + 2\gamma \cdot \mathbf{w}(n) = 0 \quad (\text{A.3})$$

we can derive the optimal solution of the extended leaky FxLMS (A.2) to

$$\mathbf{w}_o = (\gamma + \mathbf{R}_{\mathbf{x}'})^{-1} \mathbf{P}_{\mathbf{d}\mathbf{x}'} \quad (\text{A.4})$$

Furthermore, the coefficient updating equation can be obtained by using the gradient descent method as

$$\mathbf{w}(n+1) = (\mathbf{I} - \mu\gamma) \mathbf{w}(n) + \mu e(n) \mathbf{x}'(n). \quad (\text{A.5})$$

To let the solution (A.4) equals the optimal control filter of (3.65), we can get

$$\gamma = \lambda_o \mathbf{R}_x. \quad (\text{A.6})$$

Appendix B

Proofs for Part II

B.1 Optimal control filter out the frequency band

In the FxLMS algorithm, the k th recursive update equation is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k) \sum_{i=0}^{L_s-1} s_i \mathbf{x}(k-i) \in \mathbb{R}^{N \times 1} \quad (\text{B.1})$$

where μ represents the step size, and s_i denotes the i th coefficient of the secondary path model with L_s taps; $\mathbb{R}$ and N stand for the real number and the length of the control filter, respectively. If we assume $\mathbf{w}(0) = \mathbf{0}$, repeating (B.1) obtains the optimal solution as

$$\mathbf{w}_o = \mu \sum_{k=0}^{\infty} e(k) \sum_{i=0}^{L_s-1} s_i \mathbf{x}(k-i) \in \mathbb{R}^{N \times 1}. \quad (\text{B.2})$$

Multiplying (B.2) by Fourier matrix

$$\mathbf{F}_N = \left(e^{-\frac{2\pi \cdot \mathbf{j} \cdot i}{N}} / \sqrt{N} \right)_{i,j=0,\dots,N-1} \quad (\text{B.3})$$

yields

$$W_{\text{opt}}(\omega) = \mu \sum_{k=0}^{\infty} e(k) \sum_{i=0}^{L_s-1} s_i X(\omega) \in \mathbb{C}^{N \times 1} \quad (\text{B.4})$$

where $\mathbb{C}$ denotes the complex number. Therefore, if

$$X(\omega) = 0, \quad |\omega| \in [0, \omega_0 - B_0) \cup (\omega_0 + B_0, \pi] \quad (\text{B.5})$$

substituting (B.5) into (B.4) yields

$$W_{\text{opt}}(\omega) = 0, \quad |\omega| \in [0, \omega_0 - B_0) \cup (\omega_0 + B_0, \pi]. \quad (\text{B.6})$$

B.2 Power Spectral Density of the Residual Error

This section aims to derive the power spectral density of the residual error in (5.14). Since the disturbance and reference are energy signal, the auto-correlation function of (5.14) is derived as

$$\begin{aligned} r_e(n) &= \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{k=-N}^N e(k)e(k-n) \\ &= r_{d_1}(n) - r_{r_1 d_1}(n) * w_{\text{opt}}^0(n) - r_{d_1 r_1}(n) * w_{\text{opt}}^0(-n) + r_{r_1}(n) \\ &\quad * w_{\text{opt}}^0(-n) * w_{\text{opt}}^0(n) + r_v(n) * w_{\text{opt}}^0(-n) * s(-n) * w_{\text{opt}}^0(n) * s(n) \end{aligned} \quad (\text{B.7})$$

where

$$r_{d_1}(n) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{k=-N}^N d_1(k)d_1(k-n) \quad (\text{B.8})$$

$$r_{r_1}(n) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{k=-N}^N r_1(k)r_1(k-n) \quad (\text{B.9})$$

$$r_{d_1 r_1}(n) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{k=-N}^N d_1(k)r_1(k-n) \quad (\text{B.10})$$

$$r_{r_1 d_1}(n) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{k=-N}^N r_1(k)d_1(k-n) \quad (\text{B.11})$$

and

$$r_v(n) = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{k=-N}^N v(k)v(k-n). \quad (\text{B.12})$$

Hence, the PSD of (B.7) is calculated as

$$\begin{aligned}
 S_e(\omega) &= \sum_{n=-\infty}^{\infty} r_e(n) e^{-j\omega n} \\
 &= S_{d_1}(\omega) - S_{r_1 d_1}(\omega) W_{\text{opt}}^0(\omega) - S_{d_1 r_1}(\omega) W_{\text{opt}}^0(-\omega) \\
 &\quad + S_{r_1}(\omega) |W_{\text{opt}}^0(\omega)|^2 + S_v(\omega) |W_{\text{opt}}^0(\omega) S(\omega)|^2
 \end{aligned} \tag{B.13}$$

Substituting (5.9) and (5.12) into (B.13) yields

$$\begin{aligned}
 S_e(\omega) &= S_{d_1}(\omega) - S_{r_1 d_1}(\omega) \frac{P(\omega)}{S(\omega)} \cdot \text{rect}\left(\frac{\omega - \omega_0}{2B_0}\right) \\
 &\quad + S_v(\omega) |P(\omega)|^2 \cdot \text{rect}\left(\frac{\omega - \omega_0}{2B_0}\right)
 \end{aligned} \tag{B.14}$$

Since the PSD of the disturbance $d_1(n)$ can be expressed as

$$S_{d_1}(\omega) = S_{x_1} |P(\omega)|^2 \cdot \text{rect}\left(\frac{\omega - \omega_0}{2B_1}\right) = S_{d_1}(\omega) \cdot \text{rect}\left(\frac{\omega - \omega_0}{2B_1}\right) \tag{B.15}$$

the cross-spectral density of the disturbance and filtered reference signals is stated as

$$S_{r_1 d_1}(\omega) = S_{x_1}(\omega) P^*(\omega) S(\omega) \cdot \text{rect}\left(\frac{\omega - \omega_0}{2B_1}\right) = S_{d_1 r_1}^*(\omega) \cdot \text{rect}\left(\frac{\omega - \omega_0}{2B_1}\right) \tag{B.16}$$

and the PSD of the noise $v(n)$ is given by

$$S_v(\omega) = \frac{N_0}{2} \tag{B.17}$$

substituting (5.12), (B.15), (B.16), and (B.17) into (B.14) yields

$$\begin{aligned}
 S_e(\omega) &= \left[S_{d_1}(\omega) - \frac{|S_{d_1 r_1}(\omega)|^2}{S_{r_1}(\omega)} \right] \cdot \text{rect}\left(\frac{\omega - \omega_0}{2B_1}\right) \\
 &\quad + \frac{N_0}{2} |P(\omega)|^2 \cdot \text{rect}\left(\frac{\omega - \omega_0}{2B_0}\right).
 \end{aligned} \tag{B.18}$$

By split the second term of the right side of (B.18), it can be rewritten as

$$\begin{aligned}
 S_e(\omega) &= \left[S_{d_1}(\omega) - \frac{|S_{d_1 r_1}(\omega)|^2}{S_{r_1}(\omega)} + \frac{N_0}{2} |P(\omega)|^2 \right] \cdot \text{rect}\left(\frac{\omega - \omega_0}{2B_1}\right) \\
 &\quad + \frac{N_0}{2} |P(\omega)|^2 \cdot \left[\text{rect}\left(\frac{\omega - \omega_0}{2B_0}\right) - \text{rect}\left(\frac{\omega - \omega_0}{2B_1}\right) \right].
 \end{aligned} \tag{B.19}$$

B.3 Filtered reference com-partitioning frequency domain adaptive filter

This section introduces how to derive the update equation of the FxCFDAF algorithm. The residual error in Fig. 5.4 is expressed as

$$e(k) = d(k) - \sum_{i=0}^{L_s} s_i \mathbf{w}^T(k-i) \mathbf{x}(k-i) \quad (\text{B.20})$$

where s_i denotes the i th weight of the secondary path, which has L_s taps, and the reference signal vector is given by

$$\mathbf{x}(k) = [x(k), x(k-1), \dots, x(k-N+1)]^T. \quad (\text{B.21})$$

The control filter is stated to

$$\mathbf{w}(k) = [w_0(k), w_1(k), \dots, w_{N-1}(k)]^T. \quad (\text{B.22})$$

We separate the inner product between the control filter vector and the reference signal vector in (B.20) to K parts as

$$\begin{aligned} e(k) &= d(k) - \sum_{i=0}^{L_s-1} s_i \sum_{j=0}^{K-1} \mathbf{w}_j^T(k-i) \mathbf{x}^c(k-j-i) \\ &= d(k) - \left[\sum_{i=0}^{L_s-1} s_i \sum_{j=0}^{K-1} \mathbf{w}_j^T(k-i) \mathbf{x}^c(k-j-i) \right]^* \end{aligned} \quad (\text{B.23})$$

where $*$ denotes the conjugation, and

$$\mathbf{w}_j(k) = [w_j(k), w_{j+K}(k), \dots, w_{j+(M-1)K}(k)]^T. \quad (\text{B.24})$$

Inserting $\mathbf{F}_M^{-1}\mathbf{F}_M = \mathbf{I}_M$ into (B.23) yields

$$\begin{aligned}
 e(k) &= d(k) - \left[\sum_{i=0}^{L_s-1} s_i \sum_{j=0}^{K-1} \mathbf{w}_j^T(k-i) \mathbf{F}_M^{-1} \mathbf{F}_M \mathbf{x}^c(k-j-i) \right]^* \\
 &= d(k) - \left[\sum_{i=0}^{L_s-1} s_i \sum_{j=0}^{K-1} [\mathbf{F}_M^{-1} \mathbf{w}_j(k-i)]^T \mathbf{F}_M \mathbf{x}^c(k-j-i) \right]^* \\
 &= d(k) - \sum_{i=0}^{L_s-1} s_i \sum_{j=0}^{K-1} [\mathbf{W}_j(k-i)]^H [\mathbf{F}_M \mathbf{x}^c(k-j-i)]^*
 \end{aligned} \tag{B.25}$$

We define the cost function as the square of (B.25):

$$J(k) = e^2(k). \tag{B.26}$$

The gradient of (B.26) with respect to $\mathbf{W}_j(k)$ is derived to

$$\begin{aligned}
 \nabla J(k) &= -2e(k) \sum_{i=0}^{L_s-1} s_i [\mathbf{F}_M \mathbf{x}^c(k-j-i)]^* \\
 &= -2e(k) [\mathbf{F}_M \mathbf{x}_f^c(k-j)]^* \\
 &= -2e(k) \mathbf{X}_f'^*(k-j).
 \end{aligned} \tag{B.27}$$

where we assume $\mathbf{W}_j(k) = \mathbf{W}_j(k-j)$ in the slow adaptive progress. According to least mean square (LMS) method, the new control filter weight vector is given by

$$\mathbf{W}_j(k+1) = \mathbf{W}_j(k) + \mu e(k) \mathbf{X}_f'^*(k-j) \tag{B.28}$$

where μ stands for the step size. Hence, the m th element of (B.28) is stated to

$$W_{m,j}^c(k+1) = W_{m,j}^c(k) + \mu e(k) X_m'^*(k-j). \tag{B.29}$$

Combing (B.29) from $j = 0$ to $j = K-1$ yields

$$\mathbf{W}_m^c(k+1) = \mathbf{W}_m^c(k) + \mu e(k) \mathbf{X}_m'^*(k). \tag{B.30}$$

B.4 Binary distance or similarity ratio

There are two bit vectors x_i and x_j with the dimension of N . We define that $\delta_i(n)$ denotes the value of n th element in x_i , and $\delta_j(n)$ denotes the n th element of x_j , where $n = 0, 1, \dots, N - 1$.

If $\delta_i(n) = \delta_j(n) = 1$, we describe that x_i and x_j has a $1 - 1$ pair at the n th element; If $\delta_i(n) = \delta_j(n) = 0$, we describe that x_i and x_j has a $0 - 0$ pair at the n th element; If $\delta_i(n) \neq \delta_j(n) = 1$, we describe that x_i and x_j has a inequality pair at the n th element.

Furthermore, m_0 represents the number of $0 - 0$ pairs in x_i and x_j ; m_1 and m_2 denote the numbers of $1 - 1$ and inequality pairs. The binary distance or similarity ratio in the paper is defined as

$$d_{ij} = \frac{m_1}{m_1 + m_2} \quad (\text{B.31})$$

which is also called as Jaccard's coefficient [126] usually used in similarity measurement. Moreover, it is easy to know that $d_{ij} \in (0, 1]$.

For example, there are two 4-bit vectors, which are $x_1 = 0100$ and $x_2 = 1111$, respectively, as show in Table B.1. m_1 and m_2 of these two binary vectors are found to be 1 and 3.

TABLE B.1: The binary vector expressions of two 4-bit binary vectors

4-bits	$\delta_{1 \text{ or } 2}(0)$	$\delta_{1 \text{ or } 2}(1)$	$\delta_{1 \text{ or } 2}(2)$	$\delta_{1 \text{ or } 2}(3)$
x_1	0	1	0	0
x_2	1	1	1	1

Therefore, the binary distance between x_1 and x_2 is calculated as

$$d_{12} = \frac{1}{3 + 1} = 0.25. \quad (\text{B.32})$$

Appendix C

Proofs of part III

C.1 Root locus analysis of systolic FxLMS algorithm

The single-side z -domain transformation of (6.30) is given by

$$zV'_i(z) - zv'_i(0) = V'_i(z) - \mu\lambda_k [z^{-i}V'_i(z) - z^{-i}v'_i(-i)] \quad (\text{C.1})$$

where $v'(n)$ is assumed to be a causal signal, and hence, $v'(n) = 0$, $n < 0$. By rearranging (C.1), we obtain

$$V'_i(z) = \frac{z^{i+1}v'_i(0)}{z^{i+1} - z^i + \mu\lambda_k}. \quad (\text{C.2})$$

The characteristic equation of (C.2) is written as

$$z^{i+1} - z^i + \mu\lambda_k = 0. \quad (\text{C.3})$$

When μ equates to 0, the solutions of (C.3) is derived as

$$z = 0 \text{ or } z^i = 1 \quad (\text{C.4})$$

which are the poles of (C.2). Moreover, due to (C.3) is monotonic function with respect to μ , the $|z|$ will increase as μ gradually varies from 0 to $+\infty$.

FIGURE C.1: The root locus curve when $\mu \rightarrow \infty$, where $i = 3$ and $\lambda_k = 1$.

When $|z|$ is equal to 1, it can be rewritten as

$$z = e^{jx} \quad (\text{C.5})$$

and substituting (C.5) into (C.3) yields

$$e^{j \cdot (i+1)x} - e^{j \cdot ix} + \mu\lambda_k = 0. \quad (\text{C.6})$$

According to Euler's formula, (C.5) is rewritten as

$$z = \cos(x) + j \sin(x). \quad (\text{C.7})$$

Substituting (C.7) into (C.6) yields

$$\begin{cases} \cos(i+1)x - \cos ix + \mu\lambda_k = 0 \\ j \sin(i+1)x - j \sin ix = j \cdot 0. \end{cases} \quad (\text{C.8})$$

Solving (C.8), we can get the minimum solution as

$$\mu^* = \frac{2}{\lambda_k} \sin \frac{\pi}{2(2i+1)}. \quad (\text{C.9})$$

Therefore, to guarantee the convergence of (C.2), the step size bound should be

$$0 < \mu < \frac{2}{\lambda_k} \sin \frac{\pi}{2(2i+1)}. \quad (\text{C.10})$$

Example 1: If we assume that $i = 3$, and $\lambda_k = 1$ in (C.2). When μ is equal to 0, its poles is derived as

$$z_1 = 1; \quad z_2 = z_3 = 0. \quad (\text{C.11})$$

When $\mu \rightarrow \infty$, its root locus gradually travel from the poles of (C.11) across the unit cycle, as depicted in Fig. C.1. To ensure these poles not to move out of the unit cycle, we can compute the feasible area of step size as

$$\mu \in \left(0, 2 \sin \frac{\pi}{14}\right) \quad (\text{C.12})$$

which accords with the step size bound of (C.10).

List of Author's Awards, Patents, and Publications¹

Awards

- **National Instruments (NI) 2018 Academic Research Grant**, “Analysis of Multichannel Virtual Sensing Active Noise Control to Overcome Spatial Correlation and Causality Constraints.” *National Instruments*.

Patents

- **Shi, Dongyuan**, Dongmei He, and Meng Cai. “Adaptive filtering method and system based on error sub-band.” U.S. Patent 9,419,826, issued August 16, 2016.
- **Shi, Dongyuan**, Dongmei He, and Meng Cai. “Adaptive filter method and system based on error subband.” Publication of CN103004086B, issued Jun 8, 2016.
- **Shi, Dongyuan**, Dongmei He, and Meng Cai. “Adaptive filtering method and system based on error sub-band.” Publication of WO2014047854A1, issued April 3, 2014.
- **Shi, Dongyuan**, Dongmei He, and Meng Cai. “Adaptive filtering method and system based on error sub-band.” EP2897334A4, granted Dec 30, 2017.
- **Shi, Dongyuan**, Dongmei He, and Meng Cai. “Adaptive filtering method and system based on an error subband.” ES2625275T3, granted July 30, 2017.

¹The superscript * indicates joint first authors

- GUO JINHONG, **SHI DONGYUAN**, KANG YUEJUN. “Lock-in amplification technology based cell microflow image acquisition system”, CN104212706A, issued Dec 17, 2014.
- GUO JINHONG, **SHI DONGYUAN**, KANG YUEJUN. “Pulse detection technology based dual-passage cell microflow image acquisition system”, CN104212705A, issued Sep 19, 2014.
- GUO JINHONG, **SHI DONGYUAN**, YANG ZHAOHENG. “Cellphone-based portable measurement module for blood sugar, blood fat, uric acid and cholesterol”, CN204394521U, issued Jun 17, 2015.
- GUO JINHONG, **SHI DONGYUAN**, YANG ZHAOHENG. “Cell phone embedded type blood glucose, blood fat, uric acid and cholesterol comprehensive tester”, CN204330658U, issued May 13, 2015.
- GUO JINHONG, **SHI DONGYUAN**, YANG ZHAOHENG. “Blood testing appearance”, CN204882443U, issued Dec 16, 2015.
- CHEN LEI, ZHOU DONG, GUO JINHONG, **SHI DONGYUAN**, CHEN FEI, FENG PEIMIN. “Wearable monitoring instrument for monitoring partial psychomotor seizure of epileptic”, CN105266818A, issued Jan 27, 2016.
- ZHENG DESHENG, **SHI DONGYUAN**, MU LIANG, HAN YUANHAO. “Two -dimensional code moisture recorder”, CN205808586U, issued Dec 14, 2016.
- ZHENG DESHENG, **SHI DONGYUAN**, MU LIANG, HAN YUANHAO. “Two-dimensional code temperature recorder and application methods thereof”, CN106017716A, issued Oct 12, 2016.
- **SHI DONGYUAN**, ZHENG DESHENG, MU LIANG, HAN YUANHAO. “High accuracy USB real -time temperature takes notes appearance based on linear electric bridge of noble gas”, CN206038161U, issued Jul 2, 2017.
- **SHI DONGYUAN**, ZHENG DESHENG, MU LIANG, HAN YUANHAO. “High accuracy low -power consumption real -time temperature takes notes appearance based on near field communication”, CN206038179U, issued Mar 3, 2017.

Journal Articles

- **Shi, DongYuan**, Woon-Seng Gan, Bhan Lam, and Chuang Shi. "Two-gradient direction FXLMS: An adaptive active noise control algorithm with output constraint." *Mechanical Systems and Signal Processing* 116 (2019): 651-667.
- **Shi, Dongyuan**, Bhan Lam, Woon-Seng Gan, and Shulin Wen. "Optimal Leak Factor Selection for the Output-Constrained Leaky Filtered-Input Least Mean Square Algorithm." *IEEE Signal Processing Letters* 26, no. 5 (2019): 670-674.
- **Shi, Dongyuan**, Woon-Seng Gan, Jianjun He, and Bhan Lam. "Practical Implementation of Multichannel Filtered-x Least Mean Square Algorithm based on the Multiple-Parallel-Branch with Folding Architecture for Large-Scale Active Noise Control." *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*.
- **Shi Dongyuan**, Woon-seng Gan, and Bhan Lam. "Feedforward Multichannel Virtual-sensing Active Noise Control Technique: Analysis on causal and sensor-actuator constraints." *The Journal of the Acoustical Society of America*.
- **Shi, Dongyuan**, Jinhong Guo, Liang Chen, Chuncheng Xia, Zhefeng Yu, Ye Ai, Chang Ming Li, Yuejun Kang, and Zhiming Wang. "Differential microfluidic sensor on printed circuit board for biological cells analysis." *Electrophoresis* 36, no. 16 (2015): 1854-1858.
- He, Jianjun, Bhan Lam, **Dongyuan Shi**, and Woon Seng Gan. "Exploiting the underdetermined system in multichannel active noise control for open windows." *Applied Sciences* 9, no. 3 (2019): 390.
- Lam, Bhan, Chuang Shi, **Dongyuan Shi**, and Woon-Seng Gan. "Active control of sound through full-sized open windows." *Building and Environment* 141 (2018): 16-27.
- Guo, Jinhong, Xiwei Huang, **Dongyuan Shi**, Hao Yu, Ye Ai, Chang Ming Li, and Yuejun Kang. "Portable resistive pulse-activated lens-free cell imaging system." *RSC Advances* 4, no. 99 (2014): 56342-56345.

- **Shi Dongyuan**, Woon-seng Gan, Bhan Lam, and Wen Shulin. “Feedforward Selective Active Noise Control: Algorithm and Implementation.” *IEEE Transactions on audio, speech, and language processing*, (Under review).

Conference Proceedings

- **Shi, Dong Yuan**, and Woon-Seng Gan. “Comparison of different development kits and its suitability in signal processing education.” In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6280-6284. IEEE, 2016.
- **Shi, Dongyuan**, Chuang Shi, and Woon-Seng Gan. “A systolic FxLMS structure for implementation of feedforward active noise control on FPGA.” In 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), pp. 1-6. IEEE, 2016.
- **Shi, Dongyuan**, Jianjun He, Chuang Shi, Tatsuya Murao, and Woon-Seng Gan. “Multiple parallel branch with folding architecture for multichannel filtered-x least mean square algorithm.” In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1188-1192. IEEE, 2017.
- **Shi, Dongyuan**, Chuang Shi, and Woon-Seng Gan. “Effect of the audio amplifier’s distortion on feedforward active noise control.” In 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 469-473. IEEE, 2017.
- **Shi, Dong Yuan**, Bhan Lam, and Woon-Seng Gan. “A novel selective active noise control algorithm to overcome practical implementation issue.” In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1130-1134. IEEE, 2018.
- **Shi, Dongyuan**, Bhan Lam, and Woon-seng Gan. “Analysis of Multichannel Virtual Sensing Active Noise Control to Overcome Spatial Correlation and Causality Constraints.” In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8499-8503. IEEE, 2019.

- **Shi, Dongyuan**, Bhan Lam, Woon-seng Gan and Shulin Wen. "Practical Consideration and Implementation for Avoiding Saturation of Large Amplitude Active Noise Control." In 23th Intl. Congress on Acoustics, Aachen, Germany, 2019.
- **Shi, Dongyuan**, and Zishu He. "A new frequency source based on Sigma Delta and CORDIC." In 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), pp. 2288-2291. IEEE, 2012.
- Shi, Chuang, Tatsuya Murao, **Dongyuan Shi**, Bhan Lam, and Woon-Seng Gan. "Open loop active control of noise through open windows." In Proceedings of Meetings on Acoustics 172ASA, vol. 29, no. 1, p. 030007. ASA, 2016.
- Shi, Chuang, Nan Jiang, Huiyong Li, **Dongyuan Shi**, and Woon-Seng Gan. "On algorithms and implementations of a 4-channel active noise canceling window." In 2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), pp. 217-221. IEEE, 2017.
- Shi, Chuang, Huiyong Li, **Dongyuan Shi**, Bhan Lam, and Woon-Seng Gan. "Understanding multiple-input multiple-output active noise control from a perspective of sampling and reconstruction." In 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (AP-SIPA ASC), pp. 124-129. IEEE, 2017.
- Lam, Bhan, W. Gan, **Dongyuan Shi**, and Stephen Elliott. "Hybrid source arrangement for active control of noise through apertures at oblique incidences: A preliminary investigation." Proc. 25th Int. Congr. Sound Vib. ICSV25, Hiroshima, Japan (2018): 1-8.
- Hasegawa, Rina, **Dongyuan Shi**, Yoshinobu Kajikawa, and Woon-Seng Gan. "Window Active Noise Control System with Virtual Sensing Technique." In INTER-NOISE and NOISE-CON Congress and Conference Proceedings, vol. 258, no. 1, pp. 6004-6012. Institute of Noise Control Engineering, 2018.

Bibliography

- [1] WHO Europe. Night noise guidelines for europe. *World Health Organization-Regional Office for Europe*, 2009. [xxxi](#), [2](#)
- [2] C. M. Hogan and G. L. Latshaw. The relationship between highway planning and urban noise. In *Proceedings of the ASCE Urban Transportation Division Environment Impact Specialty Conference, May 21-23, 1973, Chicago, Illinois.*, 1973. [1](#)
- [3] L. Goines and L. Hagler. Noise pollution: a modern plague. *Southern medical journal*, 100(3):287–295, 2007. [1](#)
- [4] K. Rokho et al. Burden of disease from environmental noise. In *WHO International Workshop on “Combined Environmental Exposure: Noise, Air Pollutants and Chemicals” Ispra*, 2007. [1](#)
- [5] T. Munzel, T. Gori, W. Babisch, and M. Basner. Cardiovascular effects of environmental noise exposure. *European heart journal*, 35(13):829–836, 2014. [1](#)
- [6] E. Murphy and E. King. *Environmental noise pollution: Noise mapping, public health, and policy*. Newnes, 2014. [2](#)
- [7] World Health Organization (WHO) et al. Night noise guidelines for europe; regional office for europe: Copenhagen, denmark, 2009. *Google Scholar*, 2009. [3](#)
- [8] C. N. Hansen. *Understanding active noise cancellation*. CRC Press, 1999. [3](#), [4](#), [6](#), [7](#), [13](#)
- [9] S. M. Kuo and D. Morgan. *Active noise control systems: algorithms and DSP implementations*. John Wiley & Sons, Inc., 1995. [7](#), [13](#), [88](#), [130](#), [131](#), [132](#), [175](#)

- [10] S. J. Elliott and P. A. Nelson. The active control of sound. *Electronics and communication engineering journal*, 2(4):127–136, 1990. 3, 4, 13
- [11] L. Paul. Process of silencing sound oscillations, June 9 1936. US Patent 2,043,416. 4
- [12] M. J. Crocker. *Encyclopedia of acoustics*. John Wiley, 1997. 4
- [13] S. J. Elliott. *Signal processing for active control*. Elsevier, 2000. 4, 8, 39, 69, 101
- [14] Y. Kajikawa, W. S. Gan, and S. M. Kuo. Recent advances on active noise control: open issues and innovative applications. *APSIPA Transactions on Signal and Information Processing*, 1, 2012. 6
- [15] Kihiro Tanaka, Chuang Shi, and Yoshinobu Kajikawa. Binaural active noise control using parametric array loudspeakers. *Applied Acoustics*, 116:170 – 176, 2017. ISSN 0003-682X. doi: <https://doi.org/10.1016/j.apacoust.2016.09.021>. URL <http://www.sciencedirect.com/science/article/pii/S0003682X16302961>. 6
- [16] P. A. Nelson and S. J. Elliott. *Active control of sound*. Academic press, 1991. 7, 13, 173
- [17] W. S. Gan, S. Mitra, and S. M. Kuo. Adaptive feedback active noise control headset: implementation, evaluation and its extensions. *IEEE Transactions on Consumer Electronics*, 51(3):975–982, Aug 2005. ISSN 0098-3063. doi: 10.1109/TCE.2005.1510511. 7
- [18] W. S. Gan and S. M. Kuo. An integrated audio and active noise control headset. *IEEE Transactions on Consumer Electronics*, 48(2):242–247, May 2002. ISSN 0098-3063. doi: 10.1109/TCE.2002.1010128. 7
- [19] S. Elliott, I. Stothers, and P. Nelson. A multiple error lms algorithm and its application to the active control of sound and vibration. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(10):1423–1434, October 1987. ISSN 0096-3518. doi: 10.1109/TASSP.1987.1165044. 7, 19
- [20] H. Sano, T. Inoue, A. Takahashi, K. Terai, and Y. Nakamura. Active control system for low-frequency road noise combined with an audio system. *IEEE*

- Transactions on Speech and Audio Processing*, 9(7):755–763, Oct 2001. ISSN 1558-2353. doi: 10.1109/89.952494. 8
- [21] K. Tanaka, C. Shi, and Y. Kajikawa. Multi-channel active noise control using parametric array loudspeakers. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, pages 1–6, Dec 2014. doi: 10.1109/APSIPA.2014.7041721. 8
- [22] Chuang Shi and Woon-Seng Gan. Using length-limited parametric source in active noise control applications. *Proc. 20th Int. Congr. Sound Vib.*, 2013.
- [23] Chuang Shi, Tatsuya Murao, Dongyuan Shi, Bhan Lam, and Woon-Seng Gan. Open loop active control of noise through open windows. In *Proceedings of Meetings on Acoustics 172ASA*, volume 29, page 030007. Acoustical Society of America, 2016.
- [24] C. Shi and Y. Kajikawa. A partial-update minimax algorithm for practical implementation of multi-channel feedforward active noise control. In *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 1–15, Sep. 2018. doi: 10.1109/IWAENC.2018.8521376.
- [25] C. Shi, N. Jiang, H. Li, D. Shi, and W. Gan. On algorithms and implementations of a 4-channel active noise canceling window. In *2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pages 217–221, Nov 2017. doi: 10.1109/ISPACS.2017.8266476.
- [26] C. Shi, H. Li, D. Shi, B. Lam, and W. Gan. Understanding multiple-input multiple-output active noise control from a perspective of sampling and reconstruction. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 124–129, Dec 2017. doi: 10.1109/APSIPA.2017.8282013.
- [27] Stephen Elliott, Jordan Cheer, Bhan Lam, Chuang Shi, and Woon-Seng Gan. Controlling incident sound fields with source arrays in free space and through apertures. 2017.
- [28] Bhan Lam, Chuang Shi, and Woon Seng Gan. Active noise control systems for open windows: Current updates and future perspectives. In *Proceedings of the 24th International Congress on Sound and Vibration*, pages 1–7, 2017.

- [29] Jihui Zhang, Thushara D Abhayapala, Wen Zhang, Prasanga N Samarasinghe, and Shouda Jiang. Active noise control over space: A wave domain approach. *IEEE/ACM Transactions on audio, speech, and language processing*, 26(4):774–786, 2018.
- [30] Tatsuya Murao, Chuang Shi, Woon-Seng Gan, and Masaharu Nishimura. Mixed-error approach for multi-channel active noise control of open windows. *Applied Acoustics*, 127:305–315, 2017. [8](#)
- [31] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, J. Eugene Dong, and R. C. Goodlin. Adaptive noise cancelling: Principles and applications. *Proceedings of the IEEE*, 63(12):1692–1716, Dec 1975. ISSN 0018-9219. doi: 10.1109/PROC.1975.10036. [13](#), [14](#), [17](#), [32](#), [33](#), [35](#), [42](#), [167](#)
- [32] N. V. George and G. Panda. Advances in active noise control: A survey, with emphasis on recent nonlinear techniques. *Signal processing*, 93(2):363–377, 2013. [13](#)
- [33] S. M. Kuo and D. R. Morgan. Active noise control: a tutorial review. *Proceedings of the IEEE*, 87(6):943–973, June 1999. ISSN 0018-9219. doi: 10.1109/5.763310. [14](#), [17](#), [18](#), [19](#), [42](#)
- [34] M. Rupp and R. Frenzel. Analysis of lms and nlms algorithms with delayed coefficient update under the presence of spherically invariant processes. *IEEE Transactions on Signal Processing*, 42(3):668–672, March 1994. ISSN 1053-587X. doi: 10.1109/78.277860. [17](#), [83](#)
- [35] D. T. M. Slock. On the convergence behavior of the lms and the normalized lms algorithms. *IEEE Transactions on Signal Processing*, 41(9):2811–2825, Sep. 1993. ISSN 1053-587X. doi: 10.1109/78.236504. [21](#)
- [36] C. Shi and Y. Kajikawa. Identification of the parametric array loudspeaker with a volterra filter using the sparse nlms algorithm. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3372–3376, April 2015. doi: 10.1109/ICASSP.2015.7178596. [21](#)
- [37] J. R. Treichler, C. R. Johnson Jr, and M. J. Larimore. *Theory and design of adaptive filters*. Wiley-Interscience, 1987. [23](#)

- [38] Feiran Yang, Yin Cao, Ming Wu, Felix Albu, and Jun Yang. Frequency-domain filtered-x lms algorithms for active noise control: a review and new insights. *Applied Sciences*, 8(11):2313, 2018. [24](#)
- [39] F. Yang, M. Wu, and J. Yang. A computationally efficient delayless frequency-domain adaptive filter algorithm. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 60(4):222–226, April 2013. ISSN 1558-3791. doi: 10.1109/TCSII.2013.2240877.
- [40] F. Yang, G. Enzner, and J. Yang. Statistical convergence analysis for optimal control of dft-domain adaptive echo canceler. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(5):1095–1106, May 2017. ISSN 2329-9304. doi: 10.1109/TASLP.2017.2671422.
- [41] F. Yang, G. Enzner, and J. Yang. Transient analysis of partitioned-block frequency-domain adaptive filters. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5, Sep. 2019. doi: 10.23919/EUSIPCO.2019.8902963. [24](#)
- [42] R. Zhu, F. Yang, and J. Yang. An rls-based lattice-form complex adaptive notch filter. *IEEE Signal Processing Letters*, 23(2):217–221, Feb 2016. ISSN 1558-2361. doi: 10.1109/LSP.2015.2509007. [25](#)
- [43] M. H. Costa, J. C. M. Bermudez, and N. J. Bershad. Stochastic analysis of the lms algorithm with a saturation nonlinearity following the adaptive filter output. *IEEE Transactions on Signal Processing*, 49(7):1370–1387, July 2001. ISSN 1053-587X. doi: 10.1109/78.928691. [42](#)
- [44] V. G. Nithin V. and P. Ganapati. Advances in active noise control: A survey, with emphasis on recent nonlinear techniques. *Signal Processing*, 93(2):363 – 377, 2013. ISSN 0165-1684. doi: <https://doi.org/10.1016/j.sigpro.2012.08.013>. [42](#)
- [45] Y. Kajikawa. The adaptive volterra filter: its present and future. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 83(12):51–61, 2000. doi: 10.1002/1520-6440(200012)83:12<51::AID-ECJC6>3.0.CO;2-K. [42](#)

- [46] L. Z. Tan and J. Jiang. Filtered-X second-order Volterra adaptive algorithms. *Electronics Letters*, 33(8):671–672, April 1997. ISSN 0013-5194. doi: 10.1049/el:19970477. [42](#)
- [47] J. C. Patra, R. N. Pal, B. N. Chatterji, and G. Panda. Identification of nonlinear dynamic systems using functional link artificial neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(2):254–262, April 1999. ISSN 1083-4419. doi: 10.1109/3477.752797. [42](#)
- [48] C. Y. Chang and K. K. Shyu. Active noise cancellation with a fuzzy adaptive filtered-x algorithm. *IEE Proceedings - Circuits, Devices and Systems*, 150(5):416–, Oct 2003. ISSN 1350-2409. doi: 10.1049/ip-cds:20030406. [42](#)
- [49] X. Qiu and C. H. Hansen. A study of time-domain fxlms algorithms with control output constraint. *The Journal of the Acoustical Society of America*, 109(6):2815–2823, 2001. doi: 10.1121/1.1367247. [42](#)
- [50] O. J. Tobias and R. Seara. Performance comparison of the FXLMS, nonlinear FXLMS and leaky FXLMS algorithms in nonlinear active control applications. In *2002 11th European Signal Processing Conference*, pages 1–4, Sep. 2002. [43](#), [44](#)
- [51] R. Srazhidinov and R. Kamil. Performance comparison of THF-NLFXLMS and VFXLMS algorithms for Hammerstein NANC. In *2016 IEEE Conference on Systems, Process and Control (ICSPC)*, pages 76–79, Dec 2016. doi: 10.1109/SPC.2016.7920707. [44](#)
- [52] National-Instrument. Least Mean Squares (LMS) Algorithms (Adaptive Filter Toolkit). http://zone.ni.com/reference/en-XX/help/372357A-01/lvaftconcepts/aft_lms_algorithms/, June 2008. [45](#)
- [53] W. J. Kozacky and T. Ogunfunmi. An active noise control algorithm with gain and power constraints on the adaptive filter. *EURASIP Journal on Advances in Signal Processing*, 2013(1):17, Feb 2013. ISSN 1687-6180. doi: 10.1186/1687-6180-2013-17. URL <https://doi.org/10.1186/1687-6180-2013-17>. [45](#), [46](#)
- [54] F. Taringoo, J. Poshtan, and M. H. Kahaei. Analysis of effort constraint algorithm in active noise control systems. *EURASIP Journal on Advances in*

- Signal Processing*, 2006(1):054649, Dec 2006. ISSN 1687-6180. doi: 10.1155/ASP/2006/54649. URL <https://doi.org/10.1155/ASP/2006/54649>. 45
- [55] S. J. Elliott and K. H. Back. Effort constraints in adaptive feedforward control. *IEEE Signal Processing Letters*, 3(1):7–9, Jan 1996. ISSN 1070-9908. doi: 10.1109/97.475821. 46
- [56] W. J. Kozacky and T. Ogunfunmi. An adaptive IIR filter with constraints on the output power level. In *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*, pages 984–987, Nov 2010. doi: 10.1109/ACSSC.2010.5757547. 46
- [57] O. J. Tobias and R. Seara. On the lms algorithm with constant and variable leakage factor in a nonlinear environment. *IEEE Transactions on Signal Processing*, 54(9):3448–3458, Sep. 2006. ISSN 1053-587X. doi: 10.1109/TSP.2006.879274.
- [58] B. Rafaely and S. J. Elliot. A computationally efficient frequency-domain lms algorithm with constraints on the adaptive filter. *IEEE Transactions on Signal Processing*, 48(6):1649–1655, June 2000. ISSN 1053-587X. doi: 10.1109/78.845922.
- [59] W. J. Kozacky and T. Ogunfunmi. A cascaded iir–fir adaptive anc system with output power constraints. *Signal Processing*, 94:456 – 464, 2014. ISSN 0165-1684. doi: <https://doi.org/10.1016/j.sigpro.2013.06.036>. URL <http://www.sciencedirect.com/science/article/pii/S0165168413002661>. 46
- [60] X. Qiu and C. H. Hansen. Applying effort constraints on adaptive feedforward control using the active set method. *Journal of Sound and Vibration*, 260(4): 757–762, 2003. 46
- [61] H. Lan, M. Zhang, and W. Ser. A weight-constrained FxLMS algorithm for feedforward active noise control systems. *IEEE Signal Processing Letters*, 9(1):1–4, Jan 2002. ISSN 1070-9908. doi: 10.1109/97.988714.
- [62] W. H. Swann. A survey of non-linear optimization techniques. *FEBS Letters*, 2(S1):S39–S55, 1969. doi: 10.1016/0014-5793(69)80075-X.
- [63] S. M. Roberts and H. I. Lyvers. The gradient method in process control. *Industrial & Engineering Chemistry*, 53(11):877–882, 1961. doi: 10.1021/ie50623a020. URL <https://doi.org/10.1021/ie50623a020>. 46, 52

- [64] T. Hsia. Convergence analysis of lms and nlms adaptive algorithms. In *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 8, pages 667–670, April 1983. doi: 10.1109/ICASSP.1983.1172047. [53](#)
- [65] S. C. Douglas. A family of normalized lms algorithms. *IEEE Signal Processing Letters*, 1(3):49–51, March 1994. ISSN 1070-9908. doi: 10.1109/97.295321. [53](#)
- [66] M. A. Hanson. On sufficiency of the Kuhn-Tucker conditions. *Journal of Mathematical Analysis and Applications*, 80(2):545–550, 1981. [70](#)
- [67] S. D. Snyder and C. H. Hansen. The effect of transfer function estimation errors on the filtered-x LMS algorithm. *IEEE Transactions on Signal Processing*, 42(4):950–953, April 1994. ISSN 1053-587X. doi: 10.1109/78.285659. [70](#)
- [68] A. A. Milani, I. M. S. Panahi, and R. Briggs. Performance analysis of sub-band nLMS, APA and RLS in fMRI ANC with a non-minimum phase secondary path. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 2, pages II–353–II–356, April 2007. doi: 10.1109/ICASSP.2007.366245. [70](#)
- [69] O. J. Tobias and R. Seara. Leaky-FXLMS algorithm: stochastic analysis for gaussian data and secondary path modeling error. *IEEE Transactions on Speech and Audio Processing*, 13(6):1217–1230, Nov 2005. ISSN 1063-6676. doi: 10.1109/TSA.2005.852018. [71](#)
- [70] L. Wu, X. Qiu, and Y. Guo. A generalized leaky FxLMS algorithm for tuning the waterbed effect of feedback active noise control systems. *Mechanical Systems and Signal Processing*, 106:13–23, 2018. [71](#), [72](#)
- [71] S. J. Elliott and J. Cheer. Modeling local active sound control with remote sensors in spatially random pressure fields. *The Journal of the acoustical Society of america*, 137(4):1936–1946, 2015. [79](#), [81](#), [93](#), [97](#)
- [72] M. Pawelczyk. Analog active control of acoustic noise at a virtual location. *IEEE Transactions on Control Systems Technology*, 17(2):465–472, March 2009. ISSN 1063-6536. doi: 10.1109/TCST.2008.2000988. [79](#)

- [73] M. Pawelczyk. Noise control in the active headrest based on estimated residual signals at virtual microphones. In *Proceedings of the 10th International Congress on Sound and Vibration*, pages 251–258, 2003.
- [74] B. Rafaely, S. J. Elliott, and J. Garcia-Bonito. Broadband performance of an active headrest. *The journal of the Acoustical Society of America*, 106: 787–793, 1999.
- [75] Nobuhiro Miyazaki and Yoshinobu Kajikawa. Head-mounted active noise control system with virtual sensing technique. *Journal of Sound and Vibration*, 339:65 – 83, 2015. ISSN 0022-460X. doi: <https://doi.org/10.1016/j.jsv.2014.11.023>. URL <https://www.sciencedirect.com/science/article/pii/S0022460X14009201>.
- [76] H. Deng, C. Chang, and S. M. Kuo. Active noise control with virtual sensing technology for headrest. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1272–1275, Nov 2018. doi: 10.23919/APSIPA.2018.8659516. 79
- [77] R. Hasegawa, D. Shi, Y. Kajikawa, and W. S. Gan. Window active noise control system with virtual sensing technique. In *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, volume 258, pages 6004–6012. Institute of Noise Control Engineering, 2018. 79
- [78] M. R. F. Kidner, C. Petersen, A. C. Zander, and C. H. Hansen. Feasibility study of localised active noise control using an audio spotlight and virtual sensors. In *Proceedings of ACOUSTICS*, pages 55–61, 2006. 80
- [79] X. Qiu. A review of near field acoustic error sensing strategies for active sound radiation control. In *25th International Congress on Sound and Vibration 2018, ICSV 2018: Hiroshima Calling*, 2018. 80
- [80] J. Garcia-Bonito, S. J. Elliott, and C. C. Boucher. Generation of zones of quiet using a virtual microphone arrangement. *The journal of the Acoustical Society of America*, 101(6):3498–3516, 1997. 80, 81
- [81] J. M. Munn, B. S. Cazzolato, C. D. Kestell, and C. H. Hansen. Virtual error sensing for active noise control in a one-dimensional waveguide: Performance prediction versus measurement (I). *The Journal of the Acoustical Society of America*, 113(1):35–38, 2003. 80

- [82] C. D. Kestell, B. S. Cazzolato, and C. H. Hansen. Active noise control in a free field with virtual sensors. *The Journal of the Acoustical Society of America*, 109(1):232–243, 2001. [80](#)
- [83] L. Liu, S. M. Kuo, and M. Zhou. Virtual sensing techniques and their applications. In *2009 International Conference on Networking, Sensing and Control*, pages 31–36, March 2009. doi: 10.1109/ICNSC.2009.4919241. [80](#)
- [84] J. Yuan. Virtual sensing for broadband noise control in a lightly damped enclosure. *The Journal of the Acoustical Society of America*, 116(2):934–941, 2004. [81](#)
- [85] D. J. Moreau, J. Ghan, B. S. Cazzolato, and A. C. Zander. Active noise control in a pure tone diffuse sound field using virtual sensing. *The Journal of the acoustical Society of america*, 125(6):3742–3755, 2009. [81](#)
- [86] D. Moreau, B. Cazzolato, A. Zander, and C. Petersen. A review of virtual sensing algorithms for active noise control. *Algorithms*, 1(2):69–99, 2008. [81](#), [98](#)
- [87] W. Jung, S. J. Elliott, and J. Cheer. The effect of remote microphone technique and head-tracking on local active sound control. In *Proceedings of of 23rd International Congress on Sound and Vibration*, 2016.
- [88] Dunant Halim, Li Cheng, and Zhongqing Su. Virtual sensors for active noise control in acoustic–structural coupled enclosures using structural sensing: Robust virtual sensor design. *The Journal of the Acoustical Society of America*, 129(3):1390–1399, 2011. doi: 10.1121/1.3531941. URL <https://doi.org/10.1121/1.3531941>. [81](#)
- [89] M. Pawelczyk. Feedback control of acoustic noise at desired locations. *Silesian University of Technology, Gliwice*, 2005. [81](#)
- [90] S. Edamoto, C. Shi, and Y. Kajikawa. Virtual sensing technique for feedforward active noise control. In *Proceedings of Meetings on Acoustics 172ASA*, volume 29, page 030001. ASA, 2016.
- [91] C. Shi, R. Xie, N. Jiang, H. Li, and Y. Kajikawa. Selective virtual sensing technique for multi-channel feedforward active noise control systems. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics*,

- Speech and Signal Processing (ICASSP)*, pages 8489–8493, May 2019. doi: 10.1109/ICASSP.2019.8682705. [81](#), [126](#)
- [92] D. P. Das, D. J. Moreau, and B. S. Cazzolato. A computationally efficient frequency-domain filtered-x lms algorithm for virtual microphone. *Mechanical Systems and Signal Processing*, 37(1-2):440–454, 2013. [81](#)
- [93] W. Jung, S. J. Elliott, and J. Cheer. Local active sound control using the remote microphone technique and head-tracking for tonal and broadband noise sources. In *24th International Congress on Sound and Vibration (27/07/17)*, July 2017. URL <https://eprints.soton.ac.uk/417552/>. [81](#), [83](#)
- [94] D. P. Das, D. J. Moreau, and B. S. Cazzolato. A nonlinear active noise control algorithm for virtual microphones controlling chaotic noise. *The Journal of the Acoustical Society of America*, 132(2):779–788, 2012.
- [95] S. J. Elliott, W. Jung, and J. Cheer. Head tracking extends local active control of broadband sound to higher frequencies. *Scientific reports*, 8(1):5403, 2018.
- [96] D. P. Das, D. J. Moreau, and B. S. Cazzolato. Performance evaluation of an active headrest using the remote microphone technique. *Proceedings of Acoustics 2011*, pages 2–4, 2011.
- [97] W. Jung, S. J. Elliott, and J. Cheer. Local active control of road noise inside a vehicle. *Mechanical Systems and Signal Processing*, 121:144–157, 2019. [81](#), [126](#)
- [98] W. Jung, S. J. Elliott, and J. Cheer. Combining the remote microphone technique with head-tracking for local active sound control. *The Journal of the Acoustical Society of America*, 142(1):298–307, 2017. [83](#), [98](#)
- [99] C. D. Petersen, R. Fraanje, B. S. Cazzolato, A.C. Zander, and C. H. Hansen. A kalman filter approach to virtual sensing for active noise control. *Mechanical Systems and Signal Processing*, 22(2):490–508, 2008. [83](#)
- [100] S. J. Elliott, W. Jung, and J. Cheer. Causality and robustness in the remote sensing of acoustic pressure, with application to local active sound control. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8484–8488. IEEE, 2019. [83](#)

- [101] B. Widrow and M. Bilello. Adaptive inverse control. In *Proceedings of the 1993 IEEE International Symposium on Intelligent Control, Chicago*, Aug 1993. [83](#), [162](#)
- [102] M. Pawelczyk. Adaptive noise control algorithms for active headrest system. *Control Engineering Practice*, 12(9):1101–1112, 2004. [83](#)
- [103] H. Miyazaki and Y. Kajikawa. Head-mounted active noise control system with virtual sensing technique. *Journal of Sound and Vibration*, 339:65–83, 2015.
- [104] R. Maeda, Y. Kajikawa, C. C. Chang, and S. M. Kuo. Helmet anc system with virtual sensing technique. In *Proceedings of of 25rd International Congress on Sound and Vibration*, 2018.
- [105] N. Miyazaki and Y. Kajikawa. Adaptive feedback ANC system using virtual microphones. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 383–387, May 2013. doi: 10.1109/ICASSP.2013.6637674.
- [106] S Edamoto, C Shi, and Y Kajikawa. Directional feedforward anc system with virtual sensing technique. In *Proc. Int. Workshop Smart Info-Media Syst. Asia*, 2016. [83](#)
- [107] B. Widrow, J. McCool, M. G. Larimore, and C. R. Johnson. Stationary and nonstationary learning characteristics of the lms adaptive filter. In *Aspects of Signal Processing*, pages 355–393. Springer, 1977. [88](#)
- [108] M. James. The generalised inverse. *The Mathematical Gazette*, 62(420): 109–114, 1978. [102](#)
- [109] J. Lorente, M. Ferrer, M. de Diego, and A. González. GPU implementation of multichannel adaptive algorithms for local active noise control. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(11):1624–1635, Nov 2014. ISSN 2329-9290. doi: 10.1109/TASLP.2014.2344852. [125](#), [151](#)
- [110] R. Ranjan, T. Murao, B. Lam, and W. S. Gan. Selective active noise control system for open windows using sound classification. In *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, volume 253, pages 1921–1931. Institute of Noise Control Engineering, 2016. [125](#), [126](#)

- [111] D. Y. Shi, B. Lam, and W. Gan. A novel selective active noise control algorithm to overcome practical implementation issue. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1130–1134, April 2018. doi: 10.1109/ICASSP.2018.8461458. [125](#)
- [112] B. Rafaely. Active noise reducing headset-an overview. In *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, volume 2001, pages 2144–2153. Institute of Noise Control Engineering, 2001. [125](#)
- [113] M. Pawełczyk. Analogue active noise control. *Applied Acoustics*, 63(11): 1193–1213, 2002. [125](#)
- [114] Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10): 1995, 1995. [128](#)
- [115] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, J. Eugene Dong, and R. C. Goodlin. Adaptive noise cancelling: Principles and applications. *Proceedings of the IEEE*, 63(12): 1692–1716, Dec 1975. ISSN 0018-9219. doi: 10.1109/PROC.1975.10036. [128](#)
- [116] A. V. Oppenheim and A. S. Willsky and S. Nawab. Signals and systems (prentice-hall signal processing series). (*Prentice-Hall signal processing series*, 1996. [129](#)
- [117] S. S. Haykin. *Adaptive filter theory*. Pearson Education India, 2008. [132](#)
- [118] P. C. W. Sommen. Partitioned frequency domain adaptive filters. In *Twenty-Third Asilomar Conference on Signals, Systems and Computers, 1989.*, volume 2, pages 677–681, Oct 1989. doi: 10.1109/ACSSC.1989.1200983. [132](#)
- [119] P. C. W. Sommen and E. D. Wilde. Equal convergence conditions for normal- and partitioned-frequency domain adaptive filters. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 69–72 vol.4, Mar 1992. doi: 10.1109/ICASSP.1992.226409. [132](#)
- [120] J. J. Shynk. Frequency-domain and multirate adaptive filtering. *IEEE Signal Processing Magazine*, 9(1):14–37, Jan 1992. ISSN 1053-5888. doi: 10.1109/79.109205. [132](#)

- [121] S. Liu, X. Li, and J. Tian. Transform domain adaptive filter in active noise control. In *6th International Conference on Signal Processing, 2002.*, volume 1, pages 272–275 vol.1, Aug 2002. doi: 10.1109/ICOSP.2002.1181043.
- [122] F. Yang and J. Yang. Optimal step-size control of the partitioned block frequency-domain adaptive filter. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(6):814–818, June 2018. ISSN 1549-7747. doi: 10.1109/TCSII.2017.2780880.
- [123] F. Yang, G. Enzner, and J. Yang. Statistical convergence analysis for optimal control of dft-domain adaptive echo canceler. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(5):1095–1106, May 2017. ISSN 2329-9290. doi: 10.1109/TASLP.2017.2671422. [132](#)
- [124] P. G. Estermann, A. Kaelin, and A. G. Lindgren. Analysis of partitioned frequency-domain lms adaptive algorithm with application to a hands-free telephone system echo canceller. *International Journal of Adaptive Control and Signal Processing*, 14(6):587–608, 2000. [132](#)
- [125] P. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, June 1967. ISSN 0018-9278. doi: 10.1109/TAU.1967.1161901. [136](#)
- [126] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu. Using of jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, pages 380–384, 2013. [140](#), [210](#)
- [127] S. S. Choi, S. H. Cha, and C. C. Tappert. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1): 43–48, 2010. [140](#)
- [128] K. Xuan and S. M. Kuo. Study of causality constraint on feedforward active noise control systems. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 46(2):183–186, Feb 1999. ISSN 1057-7130. doi: 10.1109/82.752950. [151](#)

- [129] L. Zhang and X. Qiu. Causality study on a feedforward active noise control headset with different noise coming directions in free field. *Applied Acoustics*, 80:36–44, 2014. [151](#)
- [130] Y. Kim and Y. Park. CPU-GPU architecture for active noise control. *Applied Acoustics*, 153:1–13, 2019. [151](#)
- [131] K. K. Parhi. *VLSI digital signal processing systems: design and implementation*. John Wiley and Sons, 2007. [156](#), [158](#), [178](#), [185](#)
- [132] H. Vu and K. Chen. A Low-Power Broad-Bandwidth Noise Cancellation VLSI Circuit Design for In-Ear Headphones. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 24(6):2013–2025, June 2016. ISSN 1063-8210. doi: 10.1109/TVLSI.2015.2480425. [162](#)
- [133] S. M. Kuo, I. Panahi, K. M. Chung, T. Horner, M. Nadeski, and J. Chyan. Design of active noise control systems with the TMS320 family. *Texas Instruments Application Report*, Jun 1996. doi: SPRA042. [162](#)
- [134] S. Boucher, M. Bouchard, A. Lesperance, and B. Paillard. Implementing a single channel active adaptive noise canceller with the TMS320C50 DSP starter kit. *Texas Instruments Application Report*, Nov 1997. doi: SPRA285. [162](#)
- [135] J. Lin, S. Tang, W. Han, C. Chuang, P. Liu, and S. Young. Evaluation of speech intelligibility for feedback adaptive active noise cancellation headset. In *2006 International Conference on Biomedical and Pharmaceutical Engineering*, pages 24–29, Dec 2006. [162](#)
- [136] K. Shyu, C. Ho, and C. Chang. A study on using microcontroller to design active noise control systems. In *2014 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pages 443–446, Nov 2014. doi: 10.1109/APCCAS.2014.7032814. [162](#)
- [137] C. Chang and S. Li. Active noise control in headsets by using a low-cost microcontroller. *IEEE Transactions on Industrial Electronics*, 58(5):1936–1942, May 2011. ISSN 0278-0046. doi: 10.1109/TIE.2010.2058071. [162](#)
- [138] L. K. Ting, R. Woods, and C. F. N. Cowan. Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic support measures receivers.

- IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(1): 86–95, Jan 2005. ISSN 1063-8210. doi: 10.1109/TVLSI.2004.840403. [162](#)
- [139] S. S. Haykin. *Adaptive filter theory*. Pearson Education: Inida, 2008. [162](#)
- [140] C. Liu, Z. He, and W. Xia. A Frequency Domain LMS Algorithm with Dynamic Selection of Frequency Bins. *Circuits, Systems, and Signal Processing*, 31(6):2103–2118, Dec 2012. ISSN 1531-5878. doi: 10.1007/s00034-012-9422-1. [162](#)
- [141] D. T. M. Slock. On the convergence behavior of the LMS and the normalized LMS algorithms. *IEEE Transactions on Signal Processing*, 41(9):2811–2825, Sep. 1993. ISSN 1053-587X. doi: 10.1109/78.236504. [162](#)
- [142] B. Huang, Y. Xiao, J. Sun, and G. Wei. A Variable Step-Size FXLMS Algorithm for Narrowband Active Noise Control. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(2):301–312, Feb 2013. ISSN 1558-7916. doi: 10.1109/TASL.2012.2223673. [162](#)
- [143] R. M. Reddy, I. M. S. Panahi, and R. Briggs. Hybrid FxRLS-FxNLMS Adaptive Algorithm for Active Noise Control in fMRI Application. *IEEE Transactions on Control Systems Technology*, 19(2):474–480, March 2011. ISSN 1063-6536. doi: 10.1109/TCST.2010.2042599. [162](#)
- [144] L. Wang, M. N. S. Swamy, and M. O. Ahmad. An efficient implementation of the delay compensation for sub-band filtered-x least-mean-square algorithm. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 53(8):748–752, Aug 2006. ISSN 1549-7747. doi: 10.1109/TCSII.2006.875371. [162](#)
- [145] M. D. Meyer and D. P. Agrawal. A high sampling rate delayed lms filter architecture. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 40(11):727–729, Nov 1993. ISSN 1057-7130. doi: 10.1109/82.251841. [164](#)
- [146] D. L. Jones. Learning characteristics of transpose-form lms adaptive filters. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 39(10):745–749, Oct 1992. ISSN 1057-7130. doi: 10.1109/82.199901. [164](#)

- [147] D. Y. Shi and W. Gan. Comparison of different development kits and its suitability in signal processing education. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6280–6284, March 2016. doi: 10.1109/ICASSP.2016.7472885. 171
- [148] Texas Instruments. C6000 power-optimized DSP : Energy efficient single and multicore dsps for embedded processing applications. <http://www.ti.com/processors/dsp/c6000-dsp/overview.html/>, 2018. Copyright 1995-2018 Texas Instruments Incorporated. 176
- [149] S. M. Kuo and W. S. Gan. *Digital signal processors: architectures, implementations, and applications*. Prentice Hall, 2005. 178
- [150] J. F. Wakerly. *Digital design*. Pearson Australia Pty Limited, 2016. 184
- [151] C. Maxfield. *FPGAs: world class designs*. Newnes, 2009. 185
- [152] Y. Yi, R. Woods, L. K. Ting, and C. F. N. Cowan. High speed FPGA-Based Implementations of Delayed-LMS filters. *Journal of VLSI signal processing systems for signal, image and video technology*, 39(1):113–131, Jan 2005. ISSN 0922-5773. doi: 10.1023/B:VLSI.0000047275.54691.be. URL <https://doi.org/10.1023/B:VLSI.0000047275.54691.be>. 185
- [153] ADI. 3.5 GSPS direct digital synthesizer with 12-bit DAC: AD9914. *Analog Devices*, 2018. 187
- [154] BSEN ISO. 16283-3: 2016 acoustics– field measurement of sound insulation in buildings and of building elements. *Part 3. Facade sound insulation*, 2018. 189, 193
- [155] B. Lam, S. J. Elliott, J. Cheer, and W. S. Gan. Physical limits on the performance of active noise control through open windows. *Applied Acoustics*, 137:9–17, 2018. 193
- [156] S. J. Elliott, J. Cheer, B. Lam, C. Shi, and W. S. Gan. A wavenumber approach to analysing the active control of plane waves with arrays of secondary sources. *Journal of Sound and Vibration*, 419:405–419, 2018. 193