SYSTEM IDENTIFICATION WITH APPLICATIONS IN SPEECH ENHANCEMENT

Xiang Lin
MSc. BEng.

A Thesis submitted in fulfilment of requirements for the degree of Doctor of Philosophy of Imperial College London

Communications and Signal Processing Research Group
Department of Electrical and Electronic Engineering
Imperial College London
July 2009
To my parents, grandparents,

my wife Ling, and

To Patrick
Abstract

As the increasing popularity of integrating hands-free telephony on mobile portable devices and the rapid development of voice over internet protocol, identification of acoustic systems has become desirable for compensating distortions introduced to speech signals during transmission, and hence enhancing the speech quality. The objective of this research is to develop system identification algorithms for speech enhancement applications including network echo cancellation and speech dereverberation.

A supervised adaptive algorithm for sparse system identification is developed for network echo cancellation. Based on the framework of selective-tap updating scheme on the normalized least mean squares algorithm, the MMax and sparse partial update tap-selection strategies are exploited in the frequency domain to achieve fast convergence performance with low computational complexity. Through demonstrating how the sparseness of the network impulse response varies in the transformed domain, the multidelay filtering structure is incorporated to reduce the algorithmic delay.

Blind identification of SIMO acoustic systems for speech dereverberation in the presence of common zeros is then investigated. First, the problem of common zeros is defined and extended to include the presence of near-common zeros. Two clustering algorithms are developed to quantify the number of these zeros so as to facilitate the study of their effect on blind system identification and speech dereverberation. To mitigate such effect, two algorithms are developed where the two-stage algorithm based on channel decomposition identifies common and non-common zeros sequentially; and the forced spectral diversity approach combines spectral shaping filters and channel undermodelling for deriving a modified system that leads to an improved dereverberation performance. Additionally, a solution to the scale factor ambiguity problem in subband-based
blind system identification is developed, which motivates further research on subband-based dereverberation techniques. Comprehensive simulations and discussions demonstrate the effectiveness of the aforementioned algorithms. A discussion on possible directions of prospective research on system identification techniques concludes this thesis.
Acknowledgement

I would like to express my sincere gratitude to my supervisor Dr. Patrick Naylor who has introduced me to the exciting research of speech and audio signal processing and offered a great guidance throughout my research work. This thesis would not have been completed without his insightful suggestions, constant support and encouragement.

I would also like to thank friends and colleagues in the Speech and Audio Processing Group who have expressed their support in one way or another. It has really been a wonderful time of being a member of the group and an invaluable experience of working with group members. Special thanks are given to Dr. Andy W. H. Khong, Dr. Nikolay Gaubitch and Jimi Wen for their comments and suggestions in many aspects of my research. I am also very grateful to Prof. Mark Plumbley from Queen Mary, University of London and Dr. Danilo Mandic from Imperial College London, who examined and reviewed this thesis.

This thesis is dedicated to my family. I am indebted to my parents and grandparents for their endless care, love and support during my study and living overseas. I would like to express my gratitude to my wife, Ling, for her continual love, understanding and confidence in me that has been with me through every stage of this research.

XIAOCHAO CHEN
July 2009
## Contents

**Abstract** 3

**Acknowledgement** 5

**Contents** 6

**List of Figures** 9

**List of Tables** 13

**Acronyms** 14

**Mathematical Notations** 16

**Chapter 1. General Introduction** 20

1.1 Context of Work 20

1.2 Research Objective and Thesis Structure 23

1.3 Statement of Originality and Contributions 26

1.4 List of Publications Related to This Thesis 28

1.5 Other Publication 29

**Chapter 2. Literature Overview of System Identification for Speech Enhancement** 30

2.1 Introduction 31

2.1.1 The Fundamentals of System Identification 31

2.1.2 Current Development of Speech Enhancement 33

2.2 Supervised System Identification for Network Echo Cancellation 34

2.2.1 Problem Formulation and Assumptions 36

2.2.2 Performance Evaluation 38

2.2.3 Adaptive Algorithms for Network Echo Cancellation 39

2.2.4 Simulation Examples 42

2.3 Speech Dereverberation 44

2.4 Blind Multichannel Systems Identification 46

2.4.1 Problem Formulation and Assumptions 49

6.1 Introduction to the NMCFLMS Algorithm
   6.1.1 Algorithm Derivation
   6.1.2 The Direct-Path Constraint

6.2 Multichannel Diversity and The Near-Common Zeros

6.3 The Concept of Forced Spectral Diversity (FSD)
   6.3.1 Channel undermodelling
   6.3.2 Illustrative Examples
   6.3.3 Numerical Results

6.4 FSD Processing for SIMO acoustic systems

6.5 Design of Spectral Diversifying Filters

6.6 Simulations
   6.6.1 Setup
   6.6.2 Blind System Identification with FSD Processing
   6.6.3 Application to Speech Dereverberation

6.7 Summary

Chapter 7. Scale Ambiguity Correction for Subband-based Blind System Identification

7.1 Introduction to Subband-based Blind System Identification

7.2 The Oversampled GDFT Filter Bank

7.3 Complex Subband Decomposition

7.4 Correction for the Scale Factor Ambiguity

7.5 Simulations Results
   7.5.1 Common Zeros in Decimated Subbands
   7.5.2 Performance of the Scale Factor Ambiguity Corrector
   7.5.3 Application to Speech Dereverberation

7.6 Summary

Chapter 8. Conclusions and Future Work

8.1 Summary
8.2 Conclusions
8.3 Future work

Bibliography
# List of Figures

1.1 A structural overview of the thesis. ........................................... 26

2.1 Schematic of system identification. ............................................. 32
2.2 Schematic of adaptive network echo cancellation (NEC) ..................... 36
2.3 Acoustic impulse response simulated using the method of images. .......... 42
2.4 Convergence performance of NLMS and MMax-NLMS in normalized mis-
alignment using WGN input. ................................................... 44
2.5 Convergence performance of NLMS and MMax-NLMS in normalized mis-
alignment using speech input. .................................................. 44
2.6 Reverberation of speech due to multiple reflections off surrounding objects
in an enclosed environment. ...................................................... 45
2.7 Simulated room impulse response using the method of images with $f_s =$
8 kHz and $T_{60} = 300$ ms. ..................................................... 45
2.8 Diagram of (a) an $M$-channel SIMO acoustic system and (b) the problem
of BSI. .................................................................................. 50
2.9 The example five-channel SIMO system of length $L = 32$. .................. 59
2.10 Performance variation of MCLMS algorithm in NPM using WGN input
and random channels. .......................................................... 60
2.11 Performance variation of the subspace algorithm in NPM using WGN in-
put and random channels. ....................................................... 61
2.12 Speech dereverberation based on BSI and channel equalization. ............ 63
2.13 Dereverberation performance using the MINT algorithm and exact system
estimates. .............................................................................. 66

3.1 Schematic of adaptive network echo cancellation (NEC) (reproduced from
Fig. 2.2). ............................................................................... 68
3.2 A sparse network impulse response sampled at 8 kHz ......................... 69
List of Figures

3.3 The multidelay filtering (MDF) structure .................................................. 73
3.4 Variation of $M_i(m)$ and $M_f(m)$ against $M_{\text{mmax}}$ .......................... 79
3.5 Magnitude variation of $|h_{nw}|$ of length $2L_{nw}$ ....................................... 81
3.6 Sparseness variation of $|h_{nw}|$ against $K$ ................................................. 82
3.7 Performance variation of MMax-MDF$_i$ and MMax-MDF$_f$ with $M_{\text{mmax}}$ using WGN input .............................................................. 87
3.8 Performance variation of MMax-MDF$_N$ and MMax-MDF with $M_{\text{mmax}}$ using CGN input ................................................................. 87
3.9 Performance variation of SPMMax-MDF with $M_{\text{mmax}}$ using WGN input .. 87
3.10 Performance variation of SPMMax-MDF using CGN input for $T = 8$, $M_{\text{mmax}} = 0.5 \times 2L_{nw}$, and $K = 1$ .............................................. 88
3.11 Performance variation of SPMMax-MDF using CGN input for $T = 8$, $M_{\text{mmax}} = 0.5 \times 2L_{nw}$, and $K > 1$ .............................................. 88
3.12 Performance of SPMMax-MDF using speech input for $T = 8$, $M_{\text{mmax}} = 0.5 \times 2L_{nw}$, $K = 64$ and corresponding computational complexity .... 89
3.13 Development of the SPMMax-MDF algorithm .......................................... 90
4.1 Positions of zeros for the two-channel illustrative example SIMO system ....... 97
4.2 Number of iterations required for the cost function of MCLMS to reach $-60$ dB against $\Delta z$ ................................................................. 97
4.3 An example of two clusters of NCZs in a three-channel system in the $z$-plane, 98
4.4 Examples of the zeros of various channels showing (a) that they cluster around the unit circle and (b) with uniformly distributed phases ............ 99
4.5 The GMC-DC algorithm ........................................................................... 104
4.6 Variation of $N_c$ found using GMC algorithms against $\delta_c$ for simulated and recorded impulse responses ........................................ 106
4.7 Search time of GMC algorithms against $\delta_c$ using simulated impulse responses ................................................................. 106
4.8 Variation of $N_c$ found using GMC algorithms against $\delta_c$ with different $M$ ................................................................. 108
4.9 Variation of $N_c$ found using the GMC-DC algorithm with $\delta_c = 1 \times 10^{-3}$ against channel length $L$ ................................................................. 108
4.10 Performance variation of NMCFLMS algorithm with different $M$ .............. 108
4.11 Performance variation of the NMCFLMS algorithm against $N_c$ ................. 108
4.12 Performance variation of the MINT algorithm against $M$ ............................ 109
4.13 Variation of BSD score against $N_c$ .......................................................... 109
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Diagram of an $M$-channel SIMO system (reproduced from Fig. 2.8(b)).</td>
<td>113</td>
</tr>
<tr>
<td>5.2</td>
<td>Channel decomposition for SIMO system with exactly-common zeros.</td>
<td>114</td>
</tr>
<tr>
<td>5.3</td>
<td>Zeros of the example two-channel SIMO system.</td>
<td>116</td>
</tr>
<tr>
<td>5.4</td>
<td>BSI performance on estimating $\mathcal{H}_m(z)$ against various $N_c$.</td>
<td>116</td>
</tr>
<tr>
<td>5.5</td>
<td>Eigenvector distribution of $\mathbb{R}_x$ for the output signals of the example two-channel SIMO system of length $L = 32$.</td>
<td>117</td>
</tr>
<tr>
<td>5.6</td>
<td>An example of quantized grid in the $z$-plane for blind identification of $H_C(z)$.</td>
<td>119</td>
</tr>
<tr>
<td>5.7</td>
<td>Nonlinear function $\Lambda_q$.</td>
<td>119</td>
</tr>
<tr>
<td>5.8</td>
<td>Two-stage speech dereverberation.</td>
<td>122</td>
</tr>
<tr>
<td>5.9</td>
<td>Recorded room impulse responses from the MARDY database with superimposed exactly-common zeros.</td>
<td>123</td>
</tr>
<tr>
<td>5.10</td>
<td>Performance of the two-stage BSI approach in the $z$-domain.</td>
<td>123</td>
</tr>
<tr>
<td>5.11</td>
<td>Performance of the two-stage BSI approach in the time domain.</td>
<td>124</td>
</tr>
<tr>
<td>5.12</td>
<td>Comparison of BSI performance in NPM against SNR.</td>
<td>125</td>
</tr>
<tr>
<td>5.13</td>
<td>Comparison of BSI performance in NPM against channel length $L$.</td>
<td>125</td>
</tr>
<tr>
<td>5.14</td>
<td>Time-domain samples of the (a) original speech, (b) reverberant speech, and recovered speech using (c) the subspace algorithm, and (d) the two-stage approach.</td>
<td>126</td>
</tr>
<tr>
<td>5.15</td>
<td>Spectrograms of the (a) original speech, (b) reverberant speech, dereverberated speech using (c) the subspace method, and (d) the two-stage method.</td>
<td>127</td>
</tr>
<tr>
<td>5.16</td>
<td>Variation of BSD score against $N_c$ using recorded impulse responses.</td>
<td>127</td>
</tr>
<tr>
<td>6.1</td>
<td>Diagram of (a) an $M$-channel SIMO acoustic system and (b) the problem of BSI (reproduced from Fig. 2.8).</td>
<td>131</td>
</tr>
<tr>
<td>6.2</td>
<td>Positions of zeros for the two-channel illustrative example SIMO system (reproduced from Fig. 4.1).</td>
<td>135</td>
</tr>
<tr>
<td>6.3</td>
<td>Variation of $D(\mathbf{H})$ against (a) zero separation $\Delta z$ defined in (4.5) for the example two-channel system shown in Fig. 6.2 and (b) the number of channels $M$ for simulated acoustic SIMO systems.</td>
<td>135</td>
</tr>
<tr>
<td>6.4</td>
<td>Zeros of the illustrative example of (a) the original system $\mathbf{h}$, and that from the (b) convolution with spectral diversifying filters $\mathbf{z}_{f,m}$ with various $\rho_1$, and (c) the zeros of the resultant modified system $\mathbf{h}'$.</td>
<td>142</td>
</tr>
<tr>
<td>6.5</td>
<td>Various results for the effect of FSD processing on the modified system against various $\rho_1$: a) Mean zero separation; b) Channel diversity; c) BSI performance in NPM.</td>
<td>143</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6.6</td>
<td>Schematic for an $M$-channel SIMO system with FSD processing for speech dereverberation</td>
<td>144</td>
</tr>
<tr>
<td>6.7</td>
<td>Examples of spectral diversifying filters.</td>
<td>147</td>
</tr>
<tr>
<td>6.8</td>
<td>Floor plan of the room for acoustic impulse responses simulation.</td>
<td>148</td>
</tr>
<tr>
<td>6.9</td>
<td>Comparison of $N_c$ in the original system $h_m$ and that in the modified system $h_m'$ due to FSD processing against various tolerance $\delta_c$.</td>
<td>148</td>
</tr>
<tr>
<td>6.10</td>
<td>Variation of BSI performance for SIMO acoustic system with and without FSD processing for recorded impulse responses from the MARDY database using (a) WGN input, and (b) speech input.</td>
<td>149</td>
</tr>
<tr>
<td>6.11</td>
<td>Speech samples: (a) time sequence, and (b) spectrogram.</td>
<td>150</td>
</tr>
<tr>
<td>6.12</td>
<td>Dereverberation results: (a) reverberant speech, (b) dereverberated speech using NMCFLMS without FSD, and (c) dereverberated speech using NM-CFLMS with FSD.</td>
<td>151</td>
</tr>
<tr>
<td>7.1</td>
<td>Magnitude response of the example GDFT analysis filters.</td>
<td>157</td>
</tr>
<tr>
<td>7.2</td>
<td>Relationship between full-band channel and equivalent subband filters.</td>
<td>157</td>
</tr>
<tr>
<td>7.3</td>
<td>Comparison of $N_c$ between full-band channel and subband channel for $k = 1.163$</td>
<td>163</td>
</tr>
<tr>
<td>7.4</td>
<td>Variation of the full-band BSI performance in $NPM_f$ against scale factor variance before and after correction. The subband $NPM_s$ is assumed to be $NPM_s = -\infty$ dB.</td>
<td>163</td>
</tr>
<tr>
<td>7.5</td>
<td>Variation of $NPM_s$ against (a) $NPM_s$ before correction, after correction, and after correction with the ideal coefficients and (b) scale factor error $\xi$.</td>
<td>164</td>
</tr>
<tr>
<td>7.6</td>
<td>Spectrograms of (a) the reverberant speech, (b) dereverberated speech for the case of $NPM_s = -30$ dB, and (c) for the cases of $NPM_s = -80$ dB employing the developed scale factor ambiguity corrector.</td>
<td>167</td>
</tr>
<tr>
<td>8.1</td>
<td>Relationship between the common zeros and other key parameters in the context of BSI</td>
<td>171</td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The MMax-NLMS Algorithm</td>
<td>42</td>
</tr>
<tr>
<td>2.2</td>
<td>The MCLMS Algorithm</td>
<td>57</td>
</tr>
<tr>
<td>3.1</td>
<td>The MDF Algorithm</td>
<td>76</td>
</tr>
<tr>
<td>3.2</td>
<td>The SPMMax-MDF Algorithm</td>
<td>84</td>
</tr>
<tr>
<td>3.3</td>
<td>Complexity of various NEC algorithms</td>
<td>85</td>
</tr>
<tr>
<td>3.4</td>
<td>Complexity of various NEC algorithms for the example case</td>
<td>86</td>
</tr>
<tr>
<td>4.1</td>
<td>The GMC-ST algorithm</td>
<td>105</td>
</tr>
<tr>
<td>6.1</td>
<td>The direct-path constrained NMCFLMS Algorithm</td>
<td>136</td>
</tr>
<tr>
<td>7.1</td>
<td>Effect of scale factor ambiguity correction on BSI error</td>
<td>166</td>
</tr>
</tbody>
</table>
### Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEC</td>
<td>Acoustic echo cancellation</td>
</tr>
<tr>
<td>AP</td>
<td>Affine projection algorithm</td>
</tr>
<tr>
<td>ASR</td>
<td>Automated speech recognition</td>
</tr>
<tr>
<td>BSD</td>
<td>Bark spectral distortion</td>
</tr>
<tr>
<td>BSI</td>
<td>Blind system identification</td>
</tr>
<tr>
<td>CGN</td>
<td>Colored Gaussian noise</td>
</tr>
<tr>
<td>CR</td>
<td>Cross relation</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier transform</td>
</tr>
<tr>
<td>DTD</td>
<td>Double-talk detector</td>
</tr>
<tr>
<td>EVD</td>
<td>Eigenvalue decomposition</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier transform</td>
</tr>
<tr>
<td>FLMS</td>
<td>Fast-LMS algorithm</td>
</tr>
<tr>
<td>FRLS</td>
<td>Frequency-domain RLS algorithm</td>
</tr>
<tr>
<td>GCC</td>
<td>Generalized cross-correlation</td>
</tr>
<tr>
<td>GDF</td>
<td>Generalized DFT</td>
</tr>
<tr>
<td>GMC</td>
<td>Generalized multichannel clustering algorithm</td>
</tr>
<tr>
<td>GSVD</td>
<td>Generalized singular value decomposition</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov model</td>
</tr>
<tr>
<td>HOS</td>
<td>Higher-order statistics</td>
</tr>
<tr>
<td>IPMDF</td>
<td>Improved proportionate MDF algorithm</td>
</tr>
<tr>
<td>IPNLMS</td>
<td>Improved proportionate NLMS algorithm</td>
</tr>
<tr>
<td>LMS</td>
<td>Least mean squares algorithm</td>
</tr>
<tr>
<td>LS</td>
<td>Least squares</td>
</tr>
<tr>
<td>MCLMS</td>
<td>Multichannel LMS algorithm</td>
</tr>
<tr>
<td>MDF</td>
<td>Multidelay filtering</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multiple-input multiple-output</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>MINT</td>
<td>Multichannel inverse theorem</td>
</tr>
<tr>
<td>MOS</td>
<td>Mean opinion score</td>
</tr>
<tr>
<td>MSD</td>
<td>Mean square deviation</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean square error</td>
</tr>
<tr>
<td>NCZs</td>
<td>Near-common zeros</td>
</tr>
<tr>
<td>NEC</td>
<td>Network echo cancellation</td>
</tr>
<tr>
<td>NLMS</td>
<td>Normalized LMS algorithm</td>
</tr>
<tr>
<td>NMCFLMS</td>
<td>Normalized frequency-domain MCLMS algorithm</td>
</tr>
<tr>
<td>NPM</td>
<td>Normalized projection misalignment</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal digital assistant</td>
</tr>
<tr>
<td>PNLMS</td>
<td>Proportionate NLMS algorithm</td>
</tr>
<tr>
<td>PSTN</td>
<td>Public switched telephone network</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of service</td>
</tr>
<tr>
<td>SCLS</td>
<td>Single-channel LS equalization</td>
</tr>
<tr>
<td>SDR</td>
<td>Signal-to-distortion ratio</td>
</tr>
<tr>
<td>SIMO</td>
<td>Single-input multiple-output</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-noise ratio</td>
</tr>
<tr>
<td>SOS</td>
<td>Second-order statistics</td>
</tr>
<tr>
<td>SP</td>
<td>Sparse partial update tap selection</td>
</tr>
<tr>
<td>SPNLMS</td>
<td>Sparse partial update NLMS algorithm</td>
</tr>
<tr>
<td>SPMMax-MDF</td>
<td>Sparse partial update MMax-MDF algorithm</td>
</tr>
<tr>
<td>STFT</td>
<td>Short-time Fourier transform</td>
</tr>
<tr>
<td>SVD</td>
<td>Singular value decomposition</td>
</tr>
<tr>
<td>TDOA</td>
<td>Time delay of arrival</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over internet protocol</td>
</tr>
<tr>
<td>VSS</td>
<td>Variable step-size</td>
</tr>
<tr>
<td>WGN</td>
<td>White Gaussian noise</td>
</tr>
</tbody>
</table>
Mathematical Notations

Operators

| \cdot | Elemental absolute operator  
⌈ · ⌉ Ceiling operator  
⌈ · ⌉ T Matrix/vector transpose operator  
⌈ · ⌉ + Moore-Penrose pseudo-inverse operator  
⌈ · ⌉ H Hermitian transpose operator  
⌈ · ⌉ 1/2 Element-by-element (Hadamard) square root  
∥ · ∥ p l_p-norm  
* Complex conjugate operator  
⊗ Linear convolution operator  
⊙ Element-by-element (Hadamard) product  
E{·} Mathematical expectation operator  
var{·} Variance operator  
O(·) Order complexity in terms of number of computation  
(·)_N Down-sampling by a factor of N  
(·)_{1N} Up-sampling by a factor of N  

General Notations

x Scalar quantity  
x Vector quantity  
X Matrix quantity  
x(n) Function of a discrete variable x  
X(z) z-transform of a discrete function x(n)  
\hat{x} Fourier transform of a discrete variable x  
\hat{x} Estimate of a discrete variable x
\text{diag}\{\mathbf{x}\} \quad \text{Diagonal matrix with its diagonal elements being formed by } \mathbf{x} \\
\text{Rank}(\mathbf{X}) \quad \text{Column rank of matrix } \mathbf{X} \\
\lambda_m(\mathbf{X}) \quad \text{The } m\text{th eigenvalue (singular value) of a matrix } \mathbf{X} \\
\Re\{x\} \quad \text{Real part of a discrete complex variable } x \\

\textbf{Variables and Symbols}

\begin{align*}
0_{M \times N} & \quad \text{Null matrix of dimension } M \text{ rows } \times N \text{ columns} \\
1_{M \times N} & \quad \text{Matrix of dimension } M \text{ rows } \times N \text{ columns with all elements being 1} \\
b_s(n) & \quad \text{Uncorrelated single-channel additive noise for NEC} \\
b_{m}(n) & \quad \text{The } m\text{th channel uncorrelated additive noise for multichannel BSI} \\
B_{m}(z) & \quad \text{Transfer function of } b_{m}(n) \\
\mathcal{C}_{\{m,n\}} & \quad \text{The subcluster group matrix for channel } m \text{ and } n \text{ defined in (4.16)} \\
D_{\{m,n\}}(p,q) & \quad \text{Euclidean distance between the } p\text{th and } p\text{th zero for channel } m \text{ and } n \\
\mathcal{D}(\mathbf{H}) & \quad \text{Quantified multichannel diversity given a system matrix } \mathbf{H} \\
e(n) & \quad \text{a priori error} \\
e_{ij}(n) & \quad \text{a priori cross-relation (CR) error between channel } i \text{ and } j \\
e_{ij}(m) & \quad \text{Frequency-domain a priori CR error between channel } i \text{ and } j \\
f_s & \quad \text{Sampling frequency} \\
f_{m} & \quad \text{Vector of channel impulse response for the } m\text{th spectral shaping filter} \\
\mathbf{F}_M & \quad \text{DFT matrix of dimension } M \text{ rows } \times M \text{ columns} \\
\mathcal{F}_m & \quad \text{Convolutional matrix of } f_{m} \\
\mathbf{g}_m & \quad \text{Vector of inverse filter for the } m\text{th channel} \\
\mathbf{G}_{\text{eq},m} & \quad \text{Convolutional matrix of } \mathbf{g}_m \\
\mathbf{G}_{\text{eq},m}(z) & \quad \text{Transfer function of } \mathbf{g}_m \\
\mathbf{h}_m & \quad \text{Vector of the } m\text{th channel impulse response} \\
\mathbf{h}_{mk} & \quad \text{Vector of the } k\text{th subband filter for channel } m \\
\mathbf{h}_{nw} & \quad \text{Vector of single-channel network impulse response} \\
\mathbf{H}_m & \quad \text{Convolutional matrix of } \mathbf{h}_m \\
\mathbf{H}_{\text{eq},m} & \quad \text{Convolutional matrix of } \mathbf{h}_m \text{ for equalization} \\
H_{C}(z), \mathcal{H}_{m}(z) & \quad \text{Transfer functions of the channel components associated with exactly-common zeros and characteristic zeros respectively}
\end{align*}
$H_m(z)$ Transfer function of $h_m$

$H_{eq,m}(z)$ Transfer function of $H_{eq,m}$

$H_{mk}(z)$ Transfer function of $h_{mk}$

$\tilde{H}_m(z)$ Transfer function of the $m$th reconstructed full-band channel

$I_{M \times M}$ Identity matrix of dimension $M$ rows $\times$ $M$ columns

$J(n)$ Cost function for adaptive algorithm

$k$ Block index for the multidelay filtering (MDF) structure

$K$ Total number of subbands for GDFT filter bank

$\mathcal{K}$ Total number of blocks for the MDF structure

$L$ Length of the multichannel impulse responses

$L_g$ Length of inverse filters

$L_M$ Block length for the MDF structure

$L_{nw}$ Length of adaptive filter for NEC

$L_{pr}$ Length of prototype filter for GDFT filter bank

$L_{sub}$ Length of the subband system given by (7.16)

$m$ Frequency-domain frame index

$M$ Number of channels for multichannel systems

$M_{m_{max}}$ Number of selected taps based on MMax criterion

$M_{sp}$ Number of selected taps based on SP criterion

$\mathcal{M}(n)$ Normalized energy of the subselected tap-input vector

$n$ Time-domain sample index

$N_c$ Number of common zeros

$p_i$ The $i$th coefficient of the prototype filter in GDFT filter bank

$\mathcal{P}(z)$ Transfer function for the prototype filter in GDFT filter bank

$Q(n)$ Diagonal tap selection control matrix

$R$ Correlation-like matrix of the received signals defined in (2.37)

$R_s$ Autocorrelation matrix of the tap-input vector given by $E\{s(n)s^T(n)\}$

$R_{xs}$ Cross-correlation matrix between the tap-input vector and the desired output given by $E\{s(n)x_s(n)\}$

$R_{x_i x_j}$ Correlation matrix given by $E\{x_i(n)x_j^T(n)\}$
\( \tilde{R}(n) \) Instantaneous estimate of \( R \)
\( s(n) \) Vector of input signal \( s(n) \)
\( S(n) \) Hankel matrix of the input signal \( s(n) \) defined in (2.24)
\( T \) Parameter determining the relative significance of tap-selection strategies for the SPNLMS algorithm
\( T_{60} \) Reverberation time
\( u_{k,i} \) The \( i \)th coefficient of the analysis filter for the \( k \)th subband
\( U_k(z) \) Transfer function of the analysis filter for the \( k \)th subband
\( v_{k,i} \) The \( i \)th coefficient of the synthesis filter for the \( k \)th subband
\( V_k(z) \) Transfer function of the synthesis filter for the \( k \)th subband
\( x_m(n) \) Microphone signal for the \( m \)th channel
\( x_s(n) \) Desired single-channel received signal for NEC
\( \tilde{z}_m(p) \) The \( p \)th characteristic zero for the \( m \)th channel
\( z_{f,m}(p) \) The \( p \)th zero of the \( m \)th spectral shaping filter
\( z_m(p) \) The \( p \)th zero of the \( m \)th channel
\( z_C(p) \) The \( p \)th exactly-common zero in a multichannel system
\( a_k \) Scale factor associated with the \( k \)th subband
\( \beta_k \) Scale factor corrector for the \( k \)th subband subject to \( \| \beta_k \|_2^2 > 0, \forall k \)
\( \delta \) Regularization parameter for NLMS-based algorithms
\( \delta_c \) Tolerance parameter for the clustering algorithms
\( \delta_D \) Vector of delayed delta function defined in (2.52)
\( \eta(n) \) Normalized misalignment defined in (2.52)
\( \eta'(n) \) Normalized projection misalignment (NPM) defined in (2.27)
\( \theta_m(p) \) Phase of the \( p \)th zero of the \( m \)th channel transfer function
\( \lambda \) Forgetting factor with \( 0 \ll \lambda < 1 \)
\( \mu \) Step-size parameter for LMS-based algorithms
\( \sigma_b^2 \) Variance of the uncorrelated additive noise
\( \sigma_s^2 \) Variance of input signal
\( \varsigma(h_{n,w}) \) Sparseness measure of the network impulse response \( h_{n,w} \)
\( \tau \) Modelling delay for multichannel equalization
\( \omega \) Angular frequency
\( \emptyset \) Empty sets
Chapter 1

General Introduction

Alexander Bell, who invented the telephone in 1876, might have never imagined how fast and tremendously the technologies in speech telecommunications have developed. In the past, it was widely acceptable to use a close-talking microphone for speech input over traditional telephone systems. With the rapid development of broadband internet-based telephony technologies such as voice over internet protocol (VoIP) and long distance teleconferencing, the market is seeing a paradigm switch from traditional pure telephony to more general and flexible person-to-person communications using hands-free portable devices such as smart phones, telephony-enabled personal digital assistant (PDA)s, and laptop computing devices that incorporates phone and other multimedia functions. Recent technological evolution has further enabled such devices to support person-to-machine hands-free voice controllable functionalities for applications such as multimedia entertainment, online banking, text transcription and in-car communications. As a result, there is a strong desire to enhance signal quality in speech communications to realize aforementioned functionalities successfully on hands-free portable devices. Research on signal processing for speech enhancement has therefore intensified, of which innovative ideas are being sought [1].

1.1 Context of Work

Modern speech communication systems can no longer be assumed to operate in favorable environments [2]. In hands-free speech acquisition the talker would typically be lo-
cated at a distance of 0.3 – 4 m from the microphone(s). The captured speech signals can hence be distorted by the speaker’s surrounding environment while propagating through acoustic channels. Such distortions include the following two major components:

- **Additive noise** is introduced during the acquisition of speech signals. It can be of comparable level to speech signals, hence can largely degrade the quality of the captured speech.

In general, noise can be considered to contain any unwanted signal that interferes with the communication, measurement, perception or other processing of an information-bearing signal. Unfortunately, noise-free signals can rarely be recorded in reality since the natural environment contains inevitable and ubiquitous noise [3]. It is therefore desirable to develop signal processing techniques to “clean” the noisy speech signals before they are stored, transmitted, or played back so as to produce natural and comfortable voice communication regardless of the noise level. The problem of noise reduction has been at the heart of speech enhancement techniques for decades, and many great progresses have been made through intensive research [4, 1, 5].

- **Echoes and multipath effects** are due to acoustic reflections of speech signals from reflective points of the enclosure which introduce convolutional distortions at the microphone(s). The low speed of speech signals can even cause multipath propagation to stretch over time delay intervals that span several signal frames.

When the reflected sound waves arrive a few tens of milliseconds after the unreflected direct sound, it is heard by the talker as a distinct echo. Echoes are invariably annoying, and can completely disrupt a conversation under extreme conditions. Among various types of echoes, the network echo [6] is of particular interest in this work. Unlike acoustic echo which is due to the acoustic coupling between the loudspeaker and the microphone, the network echo is generated electrically where the mismatch between the two-wire subscriber line and four-wire trunk line in long-distance telephone systems causes the reflection of the transmitted signals. Although the problem of network echo cancellation (NEC) has been addressed and investigated for over four decades [7,8], most existing algorithms are still unsatisfactory for many applications, especially for VoIP
systems with analog phones being involved in PC-to-phone or phone-to-phone connections [9], where “PC” denotes all-digital terminals. The popularity of such emerging applications has thus called for more efficient NEC algorithms with fast rate of convergence, low computational complexity and low processing delay for real-time implementation so as to improve the quality of service (QoS).

If the reflected signals arrive a very short time after the direct sound, it would not be perceived as an echo but a spectral distortion known as reverberation, in which case multiple delayed and attenuated versions of the source signal are captured. Although most people prefer some amount of reverberation to a completely anechoic environment in, for example, a concert hall, reverberant speech sounds distant and “echoy”. Reverberation can degrade speech intelligibility when of high energy.

Reverberation often occurs while hands-free telecommunication devices are being used in an enclosed environment such as conference rooms, military vehicles and command centres. It is perceptually negligible in traditional telephony applications since the distance between human mouth and microphone is very short. However, in hands-free systems such as desktop teleconferencing terminals, reverberation affects the speech quality and can degrade the performance of subsequent speech processing algorithms including acoustic source localization [1], speaker verification [10] and speech recognition [11, 12]. These algorithms are crucial to the success of a variety of multimedia applications such as automatic camera steering for multiparty video conferencing, telephone banking and text transcription. Although the problem of reverberation can be avoided by placing microphones close to the mouth using, for example, bluetooth headsets, such solution restricts users’ flexibility which is the most desired feature of the hands-free devices. Research on speech dereverberation techniques that are independent of speaker-microphone configuration is therefore of great importance.

As both aforementioned distortions occur during the propagation of speech signals through the acoustic environment, it has been realized that the deficient knowledge about such environment is the core obstacle that makes acoustic signal processing problems difficult to solve [13]. Techniques derived from classic system identification algorithms have therefore been called for with the aim to identify acoustic systems for speech quality enhancement [14].
1.2 Research Objective and Thesis Structure

It is without doubt that any high-quality speech communication system requires effective noise control techniques. The main focus of this thesis, however, is to develop acoustic system identification algorithms so as to address and solve the problems brought about by echoes and reverberation which jointly and actively contribute to the performance degradation of various acoustic signal processing algorithms. The forthcoming chapters of this thesis will concentrate on:

- NEC techniques based on supervised (non-blind) single-channel adaptive system identification employing partial update tap-selection scheme;
- Speech dereverberation techniques based on multichannel blind system identification (BSI) and equalization with robustness to common zeros.

This thesis is organized as follows: In Chapter 2, the literature and current development of system identification algorithms with applications in speech enhancement are reviewed. Then, the problems of NEC and speech dereverberation using BSI and channel equalization are mathematically formulated, where general assumptions and performance measurement metrics are introduced. For NEC, classic adaptive algorithms including least-mean-square (LMS), normalized-LMS (NLMS) [15] and MMax-NLMS employing partial update tap-selection [16] are reviewed and their performances demonstrated through simulation examples. For speech dereverberation, two recently proposed BSI algorithms based on the cross-relations (CR) [17], namely, the multichannel LMS (MCLMS) [18] and the subspace algorithm [19] are reviewed. The objective of these BSI algorithms is to produce estimates of acoustic systems so that they can be inverted by channel equalization algorithms (which will also be briefly described in this chapter) to obtain an estimate of the source signal for speech dereverberation. Simulation examples demonstrate the performance of each reviewed algorithm as well as that of speech dereverberation.

In Chapter 3, a supervised single-channel frequency-domain adaptive system identification algorithm is developed for NEC. Based on the MMax tap selection [16] and the sparse partial (SP) update scheme [20], this algorithm achieves a fast convergence performance with reduced complexity. In addition, the multdelay filtering (MDF) struc-
ture [21] is incorporated into the frequency-domain implementation with the fast Fourier transform (FFT) technique [22] so as to mitigate the problem of processing delay inherent in the frequency-domain algorithms. As will be described, integrating MMax and SP partial update scheme in the MDF structure is not straightforward since the sparse nature of the network echo path is not necessarily preserved in MDF structure. Two approaches for achieving such integration are developed and the resultant tradeoff between convergence performance and computational complexity is discussed. Comparative simulation results are presented to demonstrate the performance improvement of the developed algorithm over existing ones.

The focus of this thesis then moves to the problem of blind identification of multichannel acoustic systems for speech dereverberation in the presence of common zeros. Chapter 4 begins with an introduction to the common zeros problem which originates from the classic channel identifiability conditions for BSI algorithms derived from second-order-statistics (SOS) of the system outputs. In the context of acoustic signal processing where acoustic systems are often of much larger size than communication systems in terms of channel length, the near-common zeros (NCZs) are likely to present. These zeros do not violate the channel identifiability conditions but are so close to each other in the $z$-plane that the performance of most classic BSI algorithms is significantly degraded. By demonstrating the presence of the NCZs in multichannel acoustic systems and their effect on the BSI algorithms, the conventional common zeros problem is extended. Two efficient clustering algorithms are then developed to quantify NCZs in multichannel acoustic systems with high order. Simulation results are presented to show the effectiveness of these algorithms and how they can facilitate the investigation of the common zeros problem in the context of BSI and channel equalization for speech dereverberation.

In Chapter 5, a two-stage approach for BSI and speech dereverberation is developed. This approach is based on the channel decomposition concept which separates the common zeros in the system from the remaining non-common ones resulting in two channel components. The two-stage algorithm then identifies these two channel components sequentially utilizing an eigenanalysis-based technique for blind order estimation of the decomposed channel components. Consequently, the characteristic channel components with non-common zeros are identified using classic BSI algorithm and the
common zeros component is identified by exploiting the stationarity of channel zeros in comparison with the zeros from source signal. Such two-stage implementation in turn leads to a speech dereverberation approach robust to the common zeros since the estimated channel components can also be equalized in a two-stage manner. Simulation results demonstrate the improvement of the two-stage algorithm in BSI and speech dereverberation performance over existing methods.

In Chapter 6, a novel concept for mitigating the problem of NCZs is developed, which is referred to as forced spectral diversity (FSD). This chapter starts with a review of the normalized frequency-domain multichannel LMS (NMCFLMS) algorithm with direct-path substitution. Benefiting from the efficient implementation using FFT, NMCFLMS provides faster convergence performance than MCLMS, and is more computationally attractive than the subspace algorithm reviewed in Chapter 2. Then, multichannel diversity is introduced and quantified using singular value decomposition (SVD). It is shown that such channel diversity can be linked to the presence of NCZs and the corresponding BSI performance. Inspired by this, the FSD concept is introduced through illustrative examples and various numerical experiments showing how the two key components, spectral diversifying filters and channel undermodelling, are combined and implemented to derive a modified system with additional channel diversity. The FSD processing is then applied to multichannel acoustic systems for speech dereverberation, where channel estimates can be obtained with more accuracy so as to allow an improved equalization performance for dereverberation. The effectiveness of the FSD concept is verified by simulation results in terms of the performance improvement for BSI and speech dereverberation.

Multirate signal processing in subbands has been successfully applied to adaptive system identification algorithms for acoustic echo cancellation (AEC) [7], where acoustic systems with a large number of coefficients are decimated to form corresponding subband systems with reduced size. As a result, classic adaptive filtering algorithms become computationally much cheaper hence can be implemented more efficiently [24]. As will be described, blind identification of decimated subband systems can also be more robust to the common zeros problem [25]. However, the main obstacle that limits the development of subband-based BSI is the scale factor ambiguity problem [19, 25]. In Chapter 7, a scale factor ambiguity correction algorithm is developed to facilitate successful
deployment of BSI algorithms in subband systems. To achieve this, complex subband decomposition \cite{26} using oversampled generalized discrete Fourier transform (GDFT) filter bank \cite{24} is reviewed. Such decomposition establishes a relationship between the full-band and subband CR error, which can be utilized to correct the scale factor ambiguity. The developed algorithm is then implemented using an iterative optimization manner. Simulation results demonstrate the performance of the developed algorithm and its application in speech dereverberation.

A structural overview of each chapter of this thesis is depicted in Fig. 1.1.

1.3 Statement of Originality and Contributions

To the best knowledge of the author, the following aspects of this thesis are believed to be original contributions:

1. Integration of MMax and sparse partial (SP) update tap selection to the frequency-domain adaptive filtering algorithm implemented under the MDF structure as depicted in Chapter 3. Publications related to this contribution are \cite{27, 28}. 

**Figure 1.1:** A structural overview of the thesis.
2. Investigation of how sparseness of the network echo impulse response varies in the MDF structure and the validation of incorporating SP tap selection in the MDF structure as also provided in Chapter 3. This has been presented in publication [28].

3. Extension of the conventional common zeros problem by introducing NCZs and demonstration of their presence in multichannel acoustic systems as developed in Chapter 4. The publications related to this contribution are [29, 30, 31].

4. Development of two clustering algorithms to quantify the number of common zeros as presented in Chapter 4. This has been presented in [30].

5. Verification of how common zeros affect the performance of BSI and speech dereverberation as discussed in Chapter 4. Publications related to this contribution include [29, 30, 31].

6. Use of channel decomposition and the development of the two-stage algorithm for blind identification of multichannel system with common zeros for speech dereverberation as described in Chapter 5. This contribution has resulted in publications [29, 32].

7. Work that links multichannel diversity, number of NCZs and the BSI performance to facilitate the development of robust BSI algorithms in the presence of NCZs as described in Chapter 6. This work has been presented in publications [31, 33].

8. Development of the forced spectral diversity (FSD) concept to mitigate the effect of NCZs and improve the performance of BSI and subsequent speech dereverberation as described in Chapter 6. The publications related to this contribution are [31, 33].

9. Derivation of the relationship between full-band and subband CR error which leads to the development of the scale factor ambiguity correction algorithm for subband-based BSI algorithms as described in Chapter 7. This contribution has been presented in publication [34].

As this thesis is not regarding the application of adaptive filtering algorithms for AEC applications, the author has decided not to include the following contribution in this thesis:
1. Development of improved-proportionate (IP) MDF algorithm with sparseness control for AEC. This contribution has resulted in publication [35].

1.4 List of Publications Related to This Thesis

- **Journal paper**


- **Conference proceedings**


1.5 Other Publication

Chapter 2

Literature Overview of System Identification for Speech Enhancement

This chapter serves as the technical foundation of this thesis by starting with an introductory overview of system identification and speech enhancement methodologies. A comprehensive literature survey of system identification algorithms for NEC and speech dereverberation are then presented. A number of classic algorithms are reviewed and their performance demonstrated by simulation results.

This chapter is organized as follows: In Section 2.1, fundamentals of system identification and state-of-the-art speech enhancement techniques are reviewed. In accordance with specific speech enhancement applications, Section 2.2 first formulates the problem of supervised adaptive system identification for NEC, where two classic adaptive algorithms, the NLMS and MMax-NLMS algorithms are derived and their performance demonstrated. Section 2.3 then addresses the problem of reverberation and classic dereverberation techniques. As a typical approach for speech dereverberation, blind multichannel system identification and channel equalization is mathematically formulated in Section 2.4. In addition, channel identifiability conditions for SOS-based BSI algorithms are described. These conditions ensure BSI algorithms to work successfully, but are likely to be violated in the context of blind identification of multichannel acoustic systems. This
consequently motivates the major contributions of this research as will be presented in forthcoming chapters. As two examples of BSI algorithms recently developed acoustic systems, the MCLMS and subspace algorithm are also reviewed in this section and their performance demonstrated through simulation examples. Last but not least, classic channel equalization algorithms are briefly reviewed in Section 2.5 since they are essential for achieving speech dereverberation. Utilizing these algorithms, simulation examples of speech dereverberation are presented. Section 2.6 summarizes this chapter.

2.1 Introduction

Inferring models from observations and studying their properties is what science is really about. System identification is such a technique that builds and estimates mathematical models of any system of interest [36], through which a better understanding of how signals are transmitted, processed, or distorted by the unknown system can be obtained so as to enable practical attempts to compensate for adverse effects introduced during signal transmission. It has therefore been applied to a variety of communications and signal processing applications [37,38] since the early 1980’s including inter-symbol interference cancellation, image deblurring, seismic signal processing [39], and underwater communications [40]. Recently, it has also found applications in the literature of acoustic signal processing for speech enhancement such as NEC/AEC [8], speech dereverberation [1, 41, 42, 25], and speech source separation [43].

2.1.1 The Fundamentals of System Identification

System identification is a diverse field that can be presented in various ways. Consider an unknown system as shown in Fig. 2.1, the output signal \( x(n) \) can be expressed as

\[
x(n) = s(n) \ast h(n) + b(n),
\]

where \( s(n) \) is the input signal, \( h(n) \) is the system impulse response, \( b(n) \) is the additive noise, “\( \ast \)” denotes linear convolution and \( n \) is the time-domain sample index. The objective of system identification is to identify the unknown system impulse response \( h(n) \). Depending on whether \( s(n) \) is accessible, system identification techniques can be
generally grouped into two different categories:

- **Supervised (non-blind) methods**: In this category, input signals are available to excite the unknown system and generate reference output signals. The solution to system estimates can thus be found by minimizing the error between the estimated and the reference output signals in, for example, a least squares (LS) manner. Historically, the Wiener theory \[15\] plays a fundamental role in supervised system identification problem and has motivated the development of adaptive filtering algorithms for efficiently solving the Wiener-Hopf equation. Such adaptive system identification approach for speech enhancement including NEC and AEC has been well-established for over four decades \[44\]. However, the performance of most existing algorithms are satisfactory only under certain conditions, and further development is highly desirable especially in reducing computational complexity and improving convergence speed \[45\].

- **Unsupervised (blind) methods**: “Blind” indicates that the input signal cannot be accessible to the processing algorithms, which leaves the output signals the only data that bear the information of the system. As a consequence, methods based on minimizing the error between estimated and the reference output signals are not applicable. However, if algorithms allowing blind identification are successfully developed, no training data or pilot signals would be required, hence allowing an extra transmission bandwidth and a relaxed relationship between the transmitter and receiver. Motivated by such advantageous potential, research in this area has become very active for a wide range of applications with systems of various characteristics \[37, 46\].

![Figure 2.1: Schematic of system identification.](image-url)
2.1.2 Current Development of Speech Enhancement

In general, the objective of speech enhancement is to improve the intelligibility or quality of speech signals. As described in Section 1.1, the problem of echo and noise control has traditionally been one of the most important and widely researched topics in the literature [8].

On one hand, noise reduction aims to restore the clean speech from noisy microphone signals. The research in this field started more than 40 years ago with the pioneering work by Schroeder who proposed the early version of spectral subtraction method [47]. Since then, many algorithms have been developed and the spectral subtraction methods have become the most popular and most used methods in real-world applications [48]. Echo cancellation, on the other hand, originates from the use of echo-suppression devices in traditional long-distance telephone communications in late 1950s [49]. Later, the theory of echo cancellation was formally developed by AT&T Bell Labs (and elsewhere) based on adaptive filtering theory where an echo replica is synthesized and subtracted from the reference microphone signal [44]. Such methods remain the most efficient for echo cancellation in hands-free telephony [50].

However, the development towards more advanced speech enhancement techniques has never stopped. Speech signal separation and dereverberation, for example, are two typical topics that have recently been addressed. With the aim of reinforcing speech signals from either the interference from competing speech or the degradation from filtered versions of themselves, these problems have been proved to be more challenging than traditional ones [1, 5].

In general, speech enhancement techniques can be categorized into two groups:

- **Single-channel approaches**: The majority of classic speech enhancement techniques are single-channel based since traditional telephony systems employ only one microphone. They can therefore be found in a wide variety of applications ranging from noise reduction, AEC/NEC to the more recently developed automatic speech recognition (ASR) [51]. In speech dereverberation, features of the speech signals such as quasi-periodicity, formant structures and harmonicity can also be exploited to derive an inverse filter with respect to the reverberation process giving
• **Multichannel approaches**: The fact that human perception of speech is based on binaural hearing suggests intuitively the use of two or more microphones. Multiple microphones allow to develop more sophisticated techniques for speech enhancement such as speech source separation and dereverberation that would be very difficult, if not impossible, to accomplish using only one microphone at a distance [53]. Speech enhancement algorithms seeking improved performance from spatial information brought about by microphone arrays are therefore becoming a popular research topic, and the development of telephony-enabled portable devices will shift from using a single microphone to multiple microphones in the near future. So far, several acoustic signal processing applications using multichannel BSI have been successfully developed [42] which include stereophonic AEC system for teleconferencing, synthesized stereo audio bridge system for multi-party conferencing and passive acoustic speaker tracking system for automatic camera steering in video conferencing. Microphone arrays have also been supported by Microsoft Vista Operating System for noise suppression and AEC [54].

Motivated by all these exciting developments, the objective of this research is to develop supervised (non-blind) and unsupervised (blind) system identification algorithms over single- and multichannel systems for speech enhancement applications such as NEC and speech dereverberation. In the rest of this chapter, these problems will be mathematically formulated and classic algorithms will be derived. Comparative simulation results will also be presented to demonstrate their performance.

### 2.2 Supervised System Identification for Network Echo Cancellation

It is well known that two-wire lines (also known as customer loop) are used in conventional analog telephones connected to a central office where a local call can be set up by simply connecting two such lines. When the distance between two telephones exceeds about 35 miles, four-wire lines are required to amplify the telephone signals [7]. The network hybrid devices are thus introduced to connect the four-wire part of the circuit to
the two-wire part at each end. Due to the wide variety of customer characteristics including distances, types of wires and telephones, number of extensions, and delays, the impedance between these two parts of the network hybrid devices can mismatch causing network echo [7].

Historically, the first echo canceller allowing full-duplex communication was invented at Bell Labs in the 1960s [44, 55, 56] and the use of adaptive filters for identifying the echo path represented the successful application of system identification in NEC. Recently, as network systems migrate from voice telephony over traditional public-switched telephone network (PSTN) to packet-switched network in VoIP, improving the QoS for VoIP has become a new topics of interest [9, 57]. In VoIP systems, network echoes occur when analog phones are involved in PC-to-phone or phone-to-phone connections. It can be propagated back to the originator with a delay due to transmission and algorithmic processing, hence impedes effective communication and degrades the QoS for VoIP [58]. In addition, the increase in VoIP traffic in recent years has resulted a high demand for running several hundred echo cancellers in one processor core. As a result, efficient network echo cancellers with reduced complexity and low delay for IP networks has received much attention.

Adaptive filtering algorithms with finite impulse response (FIR) have been applied to a wide range of signal processing applications. They were first introduced around 1960 for adaptive switching [59] and was later used for echo cancellation in [44]. They are now still the most commonly used in practice because of their straightforward implementation and relatively low complexity compared to the better performing but substantially more complex recursive LS (RLS) algorithms [15]. By far the most popular adaptive algorithms include the LMS and normalized LMS (NLMS) algorithms, which are based on the stochastic gradient algorithm. Other classic algorithms include fast RLS (FRLS) [60] and affine projection (AP) algorithms [61], which were developed to bridge the gap between RLS- and the LMS-based algorithms in terms of computational complexity. However, since further reduction in computational complexity has been of particular interest for applications with high density of echo cancellers such as VoIP, partial update adaptive algorithms have been developed, where only part of adaptive filter coefficients are selected and updated at each iteration. In return, more iterations are required to fully update all filter coefficients resulting in a slower rate of convergence. Therefore, the goal
of designing partial update adaptive algorithms is to seek a balance between complexity reduction and convergence performance with respect to specific applications of interest. In Chapter 3 of this thesis, a low complexity, low delay and fast converging frequency-domain adaptive algorithm for NEC is developed where the sparse nature of the network echo path is exploited.

### 2.2.1 Problem Formulation and Assumptions

Figure 2.2 depicts the schematic of a typical adaptive NEC system. As can be seen, the purpose of the network hybrid is to allow the far-end signal, which forms the tap-input vector 

\[ s(n) = [s(n) \ s(n-1) \ \ldots \ s(n-L_{nw}+1)]^T, \tag{2.3} \]

\[ h_{nw} = [h_{nw,0} \ h_{nw,1} \ \ldots \ h_{nw,L_{nw}-1}]^T, \tag{2.2} \]

where \([:]^T\) is the vector/matrix transpose operator. If no echo canceller is present, the output signal \(x_s(n)\) given by

\[ x_s(n) = h_{nw}^T s(n) + b_s(n) \tag{2.3} \]
with

\[ E\{b_s(n)s(n)\} = 0 \] (2.4)

and \( E\{\cdot\} \) being the mathematical expectation operator. To prevent this, supervised adaptive system identification algorithms can be employed, using an adaptive filter

\[ \hat{h}_{nw}(n) = [\hat{h}_{nw,0}(n) \hat{h}_{nw,1}(n) \ldots \hat{h}_{nw,L_{nw}-1}(n)]^T \] (2.5)

that functions as an echo canceller, to identify the unknown echo path \( h_{nw} \) so that a replica of the echo \( \hat{x}_s(n) \) can be obtained and subtracted from the transmitted signal \( x_s(n) \). The objective of supervised system identification in this context is to find the optimal \( \hat{h}_{nw}(n) \) that minimizes the resultant a priori error given by

\[ e(n) = x_s(n) - \hat{x}_s(n), \]
\[ = x_s(n) - \hat{h}_{nw}^T(n-1)s(n). \] (2.6)

For simplicity and mathematical tractability, unless otherwise stated, the following are assumed for the NEC problem in this thesis:

(A2.1) The network impulse responses \( h_{nw} \) is linear, shift-invariant and quasi-stationary;

(A2.2) A FIR filter configuration is used to model the echo path;

(A2.3) The far-end signal \( s(n) \) is not simultaneously present along with the near-end signal from the local two-wire line, i.e., no double-talk.

Linearity and shift-invariance are two important properties for simplifying the analysis of discrete-time systems. Linear systems satisfy the rules of homogeneity and additivity, which are combinatorially known as the principle of superposition. In addition, the shift-invariant property indicates that a time shift at the system input leads to the same shift at its output. This property allows a system to be characterized by its impulse response, and through identifying which, how the system would respond to any possible stimuli can be predicted [14]. Although the dynamic nature of network echo paths often requires a time-varying model, they can nevertheless be considered quasi-stationary within the duration of a time frame [7]. Since the main objective of this thesis
regarding supervised system identification algorithm for NEC is to develop fast converging algorithms with low complexity and low delay, the assumption (A2.1) is imposed.

The length of a typical network echo path ranges 64-128 milliseconds resulting in 512-1024 coefficients given an 8 kHz sampling frequency. Infinite impulse response (IIR) models are thus expected to possess better modelling capabilities than their FIR counterpart. However, it has been found that IIR filter configuration is not preferable over FIR configuration for NEC [7]. This is because that the FIR structure has several advantages including stability, good numerical properties, and linear phase behavior, which lead to an efficient hardware implementation. In addition, FIR filters can model network impulse responses accurately enough to satisfy most design criteria and there is a large number of adaptive algorithms developed for FIR filters with their performance being well investigated in terms of convergence behaviour and stability. Assumption (A2.2) is thus validated.

Double-talk situations arise when speech signals from far-end and near-end talkers are present simultaneously at the echo canceller. Under such situations, near-end speech is perceived as a high level noise source which causes the adaptive algorithms to misconverge [7]. This problem can be alleviated by employing a double-talk detector (DTD) such that adaptive filter stops adapting once double-talk is detected. One of the earliest forms of DTD algorithms for NEC is the Geigel algorithm [7]. The performance recommendation for hands-free terminals in the presence of double-talk is described in [62]. In this thesis, Assumption (A2.3) is made and as such, the adaptive algorithm in study converges to its steady-state in the absence of double-talk.

### 2.2.2 Performance Evaluation

The performance of adaptive algorithms is often evaluated by the mean-square deviation (MSD) [63] defined as \( E \{ \| h_{nw} - \hat{h}_{nw}(n) \|_2^2 \} \) with \( \| \cdot \|_2 \) denoting the \( l_2 \)-norm operator such that

\[
\| h_{nw} \|_2 = \left( h_{nw, i}^T h_{nw, i} \right)^{1/2} = \left( \sum_{i=0}^{l_{nw} - 1} h_{nw, i}^2 \right)^{1/2}.
\]
Its instantaneous measure $\|h_{nw} - \hat{h}_{nw}(n)\|_2^2$ can be normalized by the energy of the true impulse response $h_{nw}$ to derive what it is known as the normalized misalignment,

$$\eta(n) = 10 \log_{10} \left( \frac{\|h_{nw} - \hat{h}_{nw}(n)\|_2^2}{\|h_{nw}\|_2^2} \right) \text{dB},$$

(2.8)

where $\eta(n)$ measures the closeness of the estimated impulse response $\hat{h}_{nw}(n)$ to the true one $h_{nw}$. This is particularly useful to demonstrate the convergence behaviour of adaptive algorithms, although it should be noted that this measure is applicable only for synthetic (intrusive) simulations since $h_{nw}$ is unknown in practice.

### 2.2.3 Adaptive Algorithms for Network Echo Cancellation

The LMS and NLMS algorithms are now reviewed. They form the basis of most existing adaptive filtering algorithms for AEC and NEC. As a typical example of classic selective-tap adaptive algorithms, the MMax-NLMS algorithm \cite{16} is also reviewed. In Chapter 3 of this thesis, a frequency-domain adaptive system identification algorithm that utilizes multiple partial update techniques to achieve a fast converging performance with a low cost will be developed.

#### The LMS and NLMS Algorithms

The LMS algorithm is an iterative formulation which solves the Wiener-Hopf equation by employing the method of steepest descent \cite{63,15}. The basic concept of steepest descent is that from an arbitrary starting point, a small step is taken in the direction where the corresponding cost function decreases fastest as the adaptive filter coefficients are updated, iteration by iteration, towards the vicinity where the cost function is minimized \cite{15}. Letting $\mu$ be the adaptation step-size, the recursive coefficient updating equation is described by \cite{63,15}

$$\hat{h}_{nw}(n) = \hat{h}_{nw}(n - 1) - \mu \nabla J(n),$$

(2.9)

where $J(n)$ is the cost function given by

$$J(n) = E\{e^2(n)\}$$

(2.10)
with $e(n)$ being defined in (2.6). The gradient $\nabla J(n)$ can then be simplified for coefficients indices $i = 0, 1, \ldots, L_{nw} - 1$ such that

$$\nabla J(n) = \frac{\partial J(n)}{\partial \hat{h}_{nw,i}(n)} = -2R_{ss} + 2R_s\hat{h}_{nw}(n-1), \quad (2.11)$$

where, by assuming $s(n)$ and $x_s(n)$ to be statistically independent, $R_{ss} = E\{s(n)s(n)\}$ is the cross-correlation between the tap-input vector and the desired output, and $R_s = E\{s(n)s^T(n)\}$ denotes the autocorrelation of the tap-input vector. Substituting (2.11) into (2.9) and using (2.6), the LMS coefficients updating equation can be derived as

$$\hat{h}_{nw}(n) = \hat{h}_{nw}(n-1) + 2\mu s(n)e(n)s^T(n)\hat{h}_{nw}(n-1), \quad (2.12)$$

where $E\{s(n)s(n)\}$ and $E\{s(n)s^T(n)\}$ are approximated by their instantaneous estimates [15]. Defining $\lambda_{\text{max}}(R_s)$ as the maximal eigenvalue of $R_s$, the step-size $0 < \mu < 1/\lambda_{\text{max}}(R_s)$ serves as a control for adaptation speed [15]. A high value of $\mu$ can increase the rate of convergence but at the expenses of steady-state misalignment [45].

Based on the LMS algorithm, the NLMS algorithm minimizes the squared $l_2$-norm of the change in adaptive filter coefficients from one iteration to the next, that is,

$$\|\hat{h}_{nw}(n) - \hat{h}_{nw}(n-1)\|_2^2, \quad (2.13)$$

subject to the constraint of $\hat{h}_{nw}(n)s(n) = \hat{x}_s(n)$. Applying the Lagrange multipliers and following similar approach in [15], the NLMS update equation can be expressed as

$$\hat{h}_{nw}(n) = \hat{h}_{nw}(n-1) + 2\mu \frac{s(n)e(n)}{s^T(n)s(n) + \delta}, \quad (2.14)$$

where $\delta$ is the regularization parameter that ensures the stability during initialization when $s(0) = 0_{L_{nw} \times 1}$ is a null vector of dimension $L_{nw} \times 1$. 
The MMax-NLMS Partial Update Adaptive Algorithm

The fundamental basis of MMax tap selection \cite{16} is that the sensitivity of the performance error to adaptive filter coefficients at each iteration depends on two factors: (i) the shape of the mean-square error (MSE) surface and (ii) the location of that coefficient at each time instance relative to the minimal point of the MSE surface \cite{16}. Such sensitivity is reflected by the steepness of the gradient vector components as described by \eqref{eq:2.11}. Using \eqref{eq:2.12}, the instantaneous gradient estimate in the direction of the $i$th coefficient is $2s(n - i)e(n)$ for $i = 0, 1, \ldots, L_{nw} - 1$. Since the quantity $2e(n)$ is always involved, the MMax tap selection selects coefficients associated with the $M_{\text{mmax}}$ largest values of $|s(n - i)|$ for updating, where the selected coefficients contribute most to the trajectory of the adaptive algorithm towards the minimal point of the MSE surface.

The MMax-NLMS algorithm can thus be described by defining the $L_{nw} \times L_{nw}$ diagonal tap-selection control matrix

$$Q(n) = \text{diag}\{q_0(n) q_1(n) \ldots q_{L_{nw}-1}(n)\}, \quad \text{(2.15)}$$

where for coefficients indices $i = 0, 1, \ldots, L_{nw} - 1$,

$$q_i(n) = \begin{cases} 1, & i \in \{\text{indices of the } M_{\text{mmax}} \text{ maxima of } |s(n - i)|\}, \\ 0, & \text{otherwise}. \end{cases} \quad \text{(2.16)}$$

The MMax tap-selection scheme given by \eqref{eq:2.16} in the time domain can be implemented by sorting $s(n)$ using, for example, the SORTLINE \cite{64} or Short-sort \cite{65} routines. Consequently, the MMax-NLMS update equation can be written

$$\hat{h}_{nw}(n) = \hat{h}_{nw}(n - 1) + 2\mu \frac{Q(n - 1)s(n)e(n)}{s^T(n)s(n)} + \delta, \quad \text{(2.17)}$$

where as before, $\delta$ and $\mu$ are the regularization parameter and step-size respectively. The MMax-NLMS algorithm is summarized in Table 2.1.
\[ \hat{x}_s(n) = \hat{h}_{nw}^T(n-1)s(n) \]
\[ e(n) = x_s(n) - \hat{x}_s(n) \]
\[ q_i(n) = \begin{cases} 1, & i \in \{\text{indices of the } M_{\text{Mmax}} \text{ maxima of } |s(n-i)|\}, \\ 0, & \text{otherwise}, \end{cases} \]
\[ Q(n) = \text{diag}\{q_0(n) \ q_1(n) \ldots q_{L_{\text{nw}}-1}(n)\} \]
\[ \hat{h}_{nw}(n) = \hat{h}_{nw}(n-1) + 2\mu \frac{Q(n-1)s(n)e(n)}{s^T(n)s(n) + \delta} \]

Figure 2.3: Acoustic impulse response simulated using the method of images with \( f_s = 8 \) kHz and \( L_R = 1024 \).

### 2.2.4 Simulation Examples

In this section, simulation examples of NLMS and MMax-NLMS algorithms are presented. It should be noted that the simulations in this section are carried out in the context of single-channel AEC since algorithms for NEC will be introduced in Chapter 3. As will be shown in Chapter 3 for NEC applications where the echo path is sparse, sparse partial update techniques can be employed to achieve a fast rate of convergence.

For the simulation setup, an acoustic impulse response is generated using the method of images in a room of dimension \( 3 \times 4 \times 5 \) m with the source being placed 1 m and 1.1 m in front of the microphone in the transmission and receiving room, respectively. Figure 2.3 shows the generated acoustic impulse response with \( f_s = 8 \) kHz.
sampling frequency and \( L_T = L_R = 1024 \), where \( L_T \) and \( L_R \) denote the length of the impulse responses of the transmission and receiving rooms, respectively. Since the computational complexity of adaptive algorithms increases monotonically with the length of adaptive filter, \( L_{nw} = 512 < L_R \) is chosen for all simulations in this section \[45\]. Such implementation will not affect the steady-state solution to \( \hat{h}_{nw}(n) \) if the source signal is white, but a bias can be introduced for highly correlated signals like speech \[7, 67\]. The received signal \( x_s(n) \) defined in (2.3) is obtained with an uncorrelated zero mean white Gaussian noise (WGN) \( b_s(n) \) added such that an SNR as depicted in each experiment is achieved. The normalized misalignment \( \eta(n) \) defined in (2.8) is used to evaluate the convergence performance, where the estimated impulse response is compared to the first \( L_{nw} = 512 \) taps of the true one \[45\].

Figure 2.4 first compares the convergence performance of the NLMS algorithm with that of the MMax-NLMS algorithm, where the averaged \( \eta(n) \) over 10 independent trials using a WGN source sequence with zero mean is plotted. The MMax-NLMS algorithm is tested with \( M_{mmax} = L_{nw}/2 \), \( M_{mmax} = L_{nw}/4 \) and \( M_{mmax} = L_{nw}/8 \). For full adaptation \( M_{mmax} = L_{nw} \), MMax-NLMS is equivalent to NLMS. The step-size parameter for each algorithm is chosen \( \mu_{MMax} = 0.7 \) for MMax-NLMS and \( \mu_{NLMS} = 0.7 \) for NLMS and an uncorrelated zero mean WGN sequence is added to achieve an SNR of 30 dB. As can be seen from the figure, the NLMS algorithm achieves the highest rate of convergence since all taps are updated at each iteration. For the case of MMax-NLMS with \( M_{mmax} = L_{nw}/2 \), the convergence is close to that of NLMS suffering less than 1 dB degradation in \( \eta(n) \) during convergence. Results for other cases show that the rate of convergence reduces gracefully with a decreasing \( M_{mmax} \) while approximately the same steady-state \( \eta(n) \) is reached for each case of \( M_{mmax} \).

Figure 2.5 shows an additional result for NLMS and MMax-NLMS using a male speech extracted from the APLAWD database \[68\] and down-sampled at 8 kHz. The same experimental setup as in the last simulation is used. Similar to Fig. 2.4 a graceful degradation in convergence performance with a reducing \( M_{mmax} \) can be observed. It can be expected that the degradation in performance caused by the introduction of MMax-based processing is less significant on speech input signals than on WGN input signals because of the sparseness of speech.
2.3 Speech Dereverberation

Reverberation is caused by multiple reflections of the sound off surrounding objects and walls in an enclosed environment during propagation as depicted in Fig. 2.6. Although it can add warmth to music signals to help people better orient themselves in the listening environment such as a concert hall, reverberation introduces temporal and spectral distortions to the envelope and structure of the speech signal [69]. As a result, the intelligibility of speech can be significantly degraded which in turn affects the performance of various subsequent speech processing algorithms such as speech source localization and ASR [70]. Therefore, dereverberation is important for speech enhancement, and this is a blind problem since neither the source signal nor the acoustic impulse responses are available.

A typical room impulse response contains three parts: direct-path response, early and late reverberations. As depicted in Fig. 2.6, the direct-path response represents the acoustic path between loudspeaker and microphone(s) without any reflection. Early and later reverberations, however, are due to multipath reflections which highly depend on, for example, reflectivity coefficients of the enclosure and the speaker-microphone configuration. Although early reverberations tends to perceptually reinforce the direct sound and can be considered harmless to speech intelligibility, its frequency response is rarely
2.3 Speech Dereverberation

Figure 2.6: Reverberation of speech due to multiple reflections off surrounding objects in an enclosed environment.

Figure 2.7: Simulated room impulse response using the method of images with $f_s = 8$ kHz and $T_{60} = 300$ ms.

flat which distorts speech spectrum. Such effect is known as the coloration [69]. The late reverberations then adds energy in the valleys between peaks of the speech signal in the time domain degrading its intelligibility. Depending on how long reverberant sounds persist and how strong they are compared to the direct sounds, later reverberations with long delays can stretch out to the following syllables or even to the next word, hence affecting most frame-based speech processing algorithms. Figure 2.7 shows a simulated room impulse response using the method of images [66] in a room of dimension $10 \times 10 \times 3$ m, where the source is placed 3 m in front of the microphone. The early and later reverberations of the room impulse response are illustrated in the figure.

The amount of reverberation can be measured by the reverberation time which is defined as the amount of time taken for a broadband noise signal to decay by 60 dB after it has been abruptly switched off, hence it is also known as $T_{60}$ [71]. A guideline for measuring, estimating and simulating $T_{60}$ can be found in [72]. For a typical room such as a living room or office, $T_{60}$ can range from 50 ms to 600 ms. The room impulse response shown in Fig. 2.7 has a reverberation time of $T_{60} = 300$ ms, which gives rise to 2400 FIR coefficients with a 8 kHz sampling frequency.

Through intensive research, various dereverberation algorithms employing either single microphone or multiple microphones have been developed [41] [73]. According to the underlying principles, these algorithms can be classified into three categories [69]:

- **Single Microphone Dereverberation Algorithms**
  - These algorithms rely on the analysis of the signal captured by a single microphone. They take advantage of the temporal and spectral characteristics of reverberant sounds to estimate and remove them from the input signal.
  - Common techniques include cepstral subtraction, frequency warping, and spectral subtraction.

- **Multiple Microphone Dereverberation Algorithms**
  - These algorithms employ a network of microphones to capture the signal from different angles. By analyzing the phase and amplitude differences across microphones, they can effectively estimate and reduce the reverberant components.
  - Techniques such as beamforming, spatial filtering, and time delay and diversity (TDD) are used to enhance the signal-to-reverberation ratio.

- **Hybrid Dereverberation Algorithms**
  - Hybrid algorithms combine elements of single and multiple microphone approaches to leverage the strengths of both. They may use a single microphone as a reference and supplement it with additional microphones to improve performance in complex acoustic environments.
  - These algorithms often employ advanced signal processing techniques to adapt to varying conditions and reduce the effects of reverberation across different frequencies and signal-to-noise ratios.
Source model-based methods: Employing statistical speech models, source model-based dereverberation methods estimate the clean speech with the help of a priori knowledge of how they are distorted by reverberation. Typical algorithms include linear-prediction (LP)-based methods [74, 75, 76, 77] and harmonic filtering [78].

Reverberation separation via homomorphic transformation: Since the reverberation process in the time domain can be mathematically formulated as a linear convolution between the clean speech signal and the unknown room impulse response, homomorphic transformation such as cepstral analysis [79] can be used to convert time-domain convolution to summation in the complex cepstra domain, and consequently allows feasible separation of reverberation from speech signal.

Deconvolution: As described in Section [1.1], the acoustic environment where speech signals propagate is the key source to most of the distortions. The deconvolution approach to speech dereverberation therefore aims at equalizing acoustic impulse responses such that estimates of the clean speech signals can be obtained. To achieve this, BSI algorithms play a crucial role in identifying acoustic systems so as to enable channel equalization to remove reverberation from microphone signals [43].

Since this thesis focuses on the use of system identification techniques for speech enhancement, the third type of aforementioned dereverberation approaches is considered. For the rest of this chapter, a comprehensive review of BSI problems will be presented. Classic BSI algorithms based on cross-relation (CR) will be derived. Simulation results will demonstrate their performance.

2.4 Blind Multichannel Systems Identification

The idea of BSI was first introduced by Sato to the communications community with the intention of designing efficient communication systems without a training phase [80]. Since then BSI has become an extremely active topic of research, and has been widely applied to various areas such as communications [38], geophysical, multimedia signal processing [37, 39] and underwater communications [40]. In contrast to supervised system identification as described in Section [2.2] where the system input signals are available,
BSI algorithms only rely on the system output signals, and Higher-order-statistics (HOS)-based and second-order-statistics (SOS)-based methods constitute two major types of classic BSI algorithms [46].

The HOS of a stochastic process can be described by the $k$th-order cumulant or its Fourier transform known as polyspectrum for $k > 2$ [81, 82]. This statistical information of the system output signals contains the phase information of a non-Gaussian process and can be exploited by the HOS-based algorithms either directly using the polyspectra method [83] or indirectly using the Bussgang algorithm such as proposed in [84, 85]. Among these two methods, indirect methods are relatively simpler to implement, but they rely on complex minimization processes and are sensitive to noise which can cause the cost function to converge to a local minimum [42]. Direct HOS-based methods avoid the need of minimizing cost functions, but are much more complex. More importantly, both of them suffer slow rate of convergence since the time-average estimation of HOS requires a large sample size and may result in high estimation variance and approximation errors. This leads to a poor performance in terms of tracking the statistical variations of the system impulse responses such as occur in wireless communications [15].

Fortunately, these problems can be overcome in multichannel systems such as single-input multiple-output (SIMO) and multiple-input multiple-output (MIMO) systems where multiple sensors are employed [46]. As a result, SOS of the system outputs in the form of cyclostationarity can be exploited for blind identification, where cyclostationarity indicates the periodicity of the mean and autocorrelation function of a stochastic process [15]. By exploiting such periodicity, the preservation of phase information is now possible for cyclostationary process in wide sense. The use of cyclostationarity for recovering the amplitude and phase of the channel response was first recognized in [86], and then formally developed in the context of BSI in [87] based on the spatial diversity obtained from multiple channels. Since then, a large number of SOS-based multichannel BSI algorithms have been developed [46, 88], among which celebrated work include the cross-relation (CR) method [89, 90, 91, 92, 17], the subspace method [93], the LP-based subspace algorithm [94], and the two-step maximum likelihood algorithm [95]. These methods utilize either the cyclostationarity of the received signals or spatio-temporal diversity of the multichannel systems to achieve successful BSI up to a non-zero arbitrary scale factor using only the SOS of the multiple outputs [37, 38]. In this thesis, only SOS-
2.4 Blind Multichannel Systems Identification

Based BSI algorithms are considered so that the notation of SOS can be safely dropped without causing confusion unless otherwise stated.

Recent advances and innovations in speech communications have led to an increasing popularity for acoustic signal processing aiming at extracting and interpreting information from acoustic signals in, for example, the cocktail party situation [43, 42]. As a result, accurately identifying acoustic systems has become the main issue in the literature for the study of how acoustic signals are transmitted and distorted. This has therefore motivated the adoption of BSI techniques into the community of acoustic signal processing, where speech source separation and dereverberation are two typical topics. However, such adoption is not straightforward even though there is a rich literature for BSI in the communications community. This is due to the substantial differences between communication systems and acoustic systems [5]. Traditionally, antenna arrays for wireless communications work in a fairly open space. When multipath exists, delays between reflections and direct-path signals are long which results in channel impulse responses with only tens of FIR coefficients\(^1\). In acoustic signal processing, however, the microphone arrays are used in an enclosed space most of the time and are exposed to a large number of multipath reflections with short delays. In addition, human hearing has an extremely wide dynamic range and is very sensitive to even weak impulse response tails. As a result, acoustic impulse responses contain much more FIR coefficients than communication channel responses, so that channels with thousands of coefficients are typical. As described in Section 2.3 a typical office or living room exhibits reverberation time \( T_{60} \) in the order of 50 to 600 ms, which translates to acoustic impulse responses with 400 to 4800 FIR coefficients at 8 kHz sampling frequency. Undoubtedly, these facts have created great challenges to those well-established classic BSI algorithms in terms of computational complexity and identification accuracy [7, 42, 14].

Nevertheless, promising progress has been made in the acoustic signal processing community as several closed-form and adaptive algorithms have been developed for blind identification of SIMO acoustic systems [18, 23, 19, 96, 97]. Among these algorithms, closed-form ones converge quickly, yet they are difficult to implement in an adaptive mode and are computationally expensive given the high order nature of acoustic impulse responses [14]. In addition, their performance relies on the existence of numerically well-

\(^1\)It is assumed that those long delays are not modelled by FIR coefficients with small magnitudes.
defined dimensions of the signal or noise subspace. Adaptive algorithms, in contrast, are easier to implement and suitable for real-time applications. They are also capable of tracking the dynamic nature of acoustic impulse responses. However, the performance for these algorithms are still unsatisfactory due to various issues, which can be summarized as follows:

- The order of the unknown system is often invalidly assumed to be available [19];
- Most algorithms cannot operate successfully with even small amount of noise;
- The computational burden due to long acoustic impulse responses is still significant even for adaptive algorithms;
- The presence of common zeros limits the accuracy and convergence speed of blind identification.

These issues have become the subjects of current research in the community [17, 98, 95, 37, 46, 42, 25], and significant contributions have been made to overcome most of them. For example, algorithms have been developed in [99, 100, 101, 102, 103] to improve the noise robustness of MCLMS and NMCFLMS algorithms [18, 23]. Additionally, system order estimation was investigated in [92, 104]. However, the issue regarding common zeros are important yet rarely addressed. The main contributions of this research are the development of robust algorithms for blind identification of multichannel acoustic systems in the presence of common zeros.

### 2.4.1 Problem Formulation and Assumptions

The aim of BSI is to identify unknown system impulse responses excited by unknown source signals. As shown in Fig. 2.8 an acoustic environment containing one talker and $M$ microphones can be modelled as a linear SIMO system. Each microphone signal corresponds to a unique acoustic channel resulted from multipath reflections of the transmitted speech signals. In this thesis, only SIMO systems are considered, which is not difficult to be extended to the more general MIMO cases [42, 14]. It should also be noted that although acoustic channels are inherently time-varying due to, for example, changes of speaker-microphone configurations, it does not usually invalidate the use of quasi-stationary FIR models within a time frame [14]. Therefore, assumptions (A2.1) and (A2.2)
described in Section 2.2.1 are also applicable for the case of BSI since the focus of this thesis is on the common zeros problem. The order of the unknown system is also assumed to be known. Although algorithms such as minimum description length (MDL) [105] or information theoretic criteria (AIC) [106] can in principle be employed for channel order estimation, they are sensitive to variations in SNR and data sample size [104]. Similar to (2.4), the additive noise is assumed to be zero-mean and uncorrelated with the source signal.

For an $M$-channel SIMO system as shown in Fig. 2.8 denote the $m$th impulse response with $L$ coefficients as

$$h_m = [h_{m,0}, h_{m,1}, \ldots, h_{m,L-1}]^T,$$

(2.18)

for $m = 1, 2, \ldots, M$, the $m$th microphone signal can be expressed as

$$x_m(n) = h_m \ast s(n) + b_m(n),$$

(2.19)

where $s(n)$ is the source signal and $b_m(n)$ is the additive noise. In vector form, (2.19) can be written

$$x_m(n) = H_m s(n) + b_m(n),$$

(2.20)

where $s(n) = [s(n) \ s(n-1) \ \ldots \ s(n-2L+1)]^T$, $x_m(n) = [x_m(n) \ x_m(n-1) \ \ldots \ x_m(n-L+1)]^T$, $b_m(n) = [b_m(n) \ b_m(n-1) \ \ldots \ b_m(n-L+1)]^T$, and $H_m$ is the $L \times (2L - 1)$

---

**Figure 2.8**: Diagram of (a) an $M$-channel SIMO acoustic system and (b) the problem of BSI.
convolutional matrix for the $m^{th}$ channel such that

$$
H_m = \begin{bmatrix}
 h_{m,0} & \cdots & h_{m,L-1} & \cdots & 0 \\
 0 & h_{m,0} & \cdots & h_{m,L-1} & \cdots & 0 \\
 \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
 0 & \cdots & 0 & h_{m,0} & \cdots & h_{m,L-1}
\end{bmatrix}.
$$

(2.21)

Since the impulse responses are assumed to be quasi-stationary such as indicated by (2.18) and (2.19), $H_m$ is independent of $n$. By concatenating all $M$ outputs of (2.20), a system of equations

$$
x(n) = Hs(n) + b(n)
$$

(2.22)

can be obtained using the following quantities

$$
x(n) = [x_1^T(n) x_2^T(n) \ldots x_M^T(n)]^T,
$$

(2.23a)

$$
H = [H_1^T H_2^T \ldots H_M^T]^T,
$$

(2.23b)

$$
b(n) = [b_1^T(n) b_2^T(n) \ldots b_M^T(n)]^T.
$$

(2.23c)

The problem of BSI is to find $h_m$ using only $x(n)$. This means that, with reference to Fig. 2.8(b), for a given output $x(n)$ a unique solution to $h_{m}, \forall m$ should be obtained up to a non-zero scale factor across all channels. These scale factors are irrelevant in most of acoustic signal processing applications. As will be described in Chapter 7, however, employing BSI algorithms in multirate systems can result in scale factors ambiguities between different subbands, which can significantly degrade the accuracy of the full-band channel estimate.

### 2.4.2 Channel Identifiability Conditions

Channel identifiability is concerned with the existence of a unique solution to the unknown system impulse responses with respect to a particular type of system identification algorithms [37]. According to [17], two inductive conditions are necessary and sufficient for blind identifiability using BSI algorithms over SIMO systems, which can be summarized as follows:
(C2.1) Channel diversity: The use of multisensor techniques introduces channel diversity and enables the exploration of SOS of the system outputs for blind identification of SIMO systems [90, 88]. Channel diversity in this context refers to channels being coprime, that is, multichannel FIR transfer functions do not share any common zeros. If one or more common zeros exist across all channels then these channels are not coprime. As an extreme example, a SIMO system with $M$ identical channels is of no difference to a single-channel system which exhibits no channel diversity and is thus unidentifiable using BSI algorithms. Most existing methods [17, 18, 23, 19] fail to produce a unique solution of channel estimates when common zeros exist since they cannot distinguish the common zeros due to the unknown system from ones due to the source signal.

(C2.2) Condition for the input signals: Although BSI algorithms only rely on output signals of the system, the characteristics of input signals are not negligible. An obvious requirement is that the input data should be non-zero valued. According to [17], the $L \times L$ Hankel matrix of the source signal given by

$$S(n) = \begin{bmatrix}
s(n) & s(n-1) & \cdots & s(n-L+1) \\
s(n-1) & s(n-2) & \cdots & s(n-L) \\
\vdots & \vdots & \ddots & \vdots \\
s(n-L+1) & s(n-L) & \cdots & s(n-2L+2)
\end{bmatrix}$$

must be of full-rank. This can be understood by expressing,

$$S(n)h_m = x_m(n), \quad m = 1, 2, \ldots, M$$

(2.25)

for the noiseless case, from which it can be found that if $S(n)$ is rank deficient, (2.25) will not lead to a unique solution even if the source signal $s(n)$ is known since there are not sufficient number of linear equations to solve all unknown coefficients of $h_m$. In other words, a rank-deficient $S(n)$ can not fully excite any SIMO system.

The channel identifiability conditions have been studied in [107, 108, 109, 110, 111] and extended to MIMO case in [112]. Since this research focuses on the common zeros problem stated in the condition (C2.1), the condition (C2.2) will be assumed for the remainder of
this thesis. In Chapter 4 and Chapter 6 of this thesis, detailed investigation of channel identifiability in the presence of common zeros and channel diversity will be presented.

### 2.4.3 Performance Evaluation

The performance of BSI is measured by the similarity between true channel impulse responses and estimated ones. Although not available in practice, the true channel impulse response is available for reference in synthetic simulations as described in Section 2.2.2. The evaluation method should be based on an error measure that is appropriate for the specific problem and independent of how the estimates are derived. Accordingly, define the estimated impulse response for channel $m$ as

$$\hat{h}_m(n) = [\hat{h}_{m,0}(n) \hat{h}_{m,1}(n) \ldots \hat{h}_{m,L-1}(n)]^T,$$  

so that the normalized projection misalignment (NPM) \cite{113} given by

$$\eta'(n) = 20 \log_{10} \left( \frac{1}{\|h\|_2} \frac{\|h - \kappa(n)\hat{h}(n)\|_2}{\|\hat{h}(n)\|_2} \right) \text{dB}$$  

(2.27)

can be used as the performance measurement for BSI algorithms, where

$$h = [h_1^T, h_2^T, \ldots, h_M^T]^T,$$  

$$\hat{h}(n) = [\hat{h}_1^T(n), \hat{h}_2^T(n), \ldots, \hat{h}_M^T(n)]^T,$$  

are concatenated vectors of true and estimated channel responses respectively, and

$$\kappa(n) = \frac{\hat{h}^T\hat{h}(n)}{h^T(n)h(n)}.$$  

(2.30)

Different from $\eta(n)$ defined in (2.8), $\eta'(n)$ can be interpreted as the normalized minimum squared distance from the true channels to the linear manifold of the estimated channels, which is obtained by projecting the former onto the latter \cite{113}. Since classic BSI algorithms estimate the channel responses up to an (unknown) scale factor, the projection error defined in (2.30) ensures that only the intrinsic misalignment of the channel estimate is taken into account so that the scale factor would not affect the evaluation.
2.4 Blind Multichannel Systems Identification

2.4.4 Blind System Identification Algorithms employing Cross-Relation

As one of the first proposed SOS-based multichannel BSI algorithms, the CR method [17] has served as the foundation for many subsequent algorithms. Its principle is firstly derived for the noiseless case, i.e., \( b_m(n) = 0, \forall m \). Two classic CR-based algorithms for multichannel acoustic systems, the MCLMS [18] and the subspace algorithm [19] are then reviewed for the noisy case.

With reference to Fig. 2.8(b), the CR between two channels is expressed as

\[
x_m(n) \ast h_l = s(n) \ast h_m \ast h_l = x_l(n) \ast h_m, \quad m = 1, 2, \ldots, M - 1, \quad l = m + 1, \quad (2.31)
\]

for the noiseless case. In vector form, (2.31) can be rewritten

\[
x_m^T(n) h_l = x_l^T(n) h_m. \quad (2.32)
\]

Multiplying (2.32) by \( x_m(n) \) and taking expectation on both sides leads to

\[
R_{x_mx_m} h_l = R_{x_mx_l} h_m, \quad (2.33)
\]

where \( R_{x_mx_l} \) is the cross-correlation matrix between \( x_m(n) \) and \( x_l(n) \) given by

\[
R_{x_mx_l} = E\{x_m(n)x_l^T(n)\}. \quad (2.34)
\]

Summing (2.33) over \( M - 1 \) equations associated with \( h_m \) results in

\[
\sum_{m=1, m \neq l}^{M-1} R_{x_mx_m} h_l = \sum_{m=1, m \neq l}^{M-1} R_{x_mx_l} h_m. \quad (2.35)
\]

By combining the remaining \( M \) equations for \( h_l \), the following equation is obtained

\[
Rh = 0_{ML \times 1}, \quad (2.36)
\]
where \( h \) has been defined in (2.28), \( \mathbf{0}_{ML \times 1} \) is a \( ML \times 1 \) null vector, and

\[
\mathbf{R} = \begin{bmatrix}
\sum_{m \neq 1} \mathbf{R}_{x_m x_m} - \mathbf{R}_{x_1 x_1} & \cdots & -\mathbf{R}_{x_M x_1} \\
-\mathbf{R}_{x_1 x_2} & \sum_{m \neq 2} \mathbf{R}_{x_m x_m} & \cdots & -\mathbf{R}_{x_M x_2} \\
\vdots & \ddots & \ddots & \vdots \\
-\mathbf{R}_{x_1 x_M} & -\mathbf{R}_{x_2 x_M} & \cdots & \sum_{m \neq M} \mathbf{R}_{x_m x_m}
\end{bmatrix}_{ML \times ML}.
\] (2.37)

According to [17], if both conditions (C2.1) and (C2.2) described in Section 2.4.2 are satisfied, \( \mathbf{R} \) would be of full-rank, i.e., \( \text{Rank}(\mathbf{R}) = ML - 1 \), and the solution to \( h \) lies within the null space of \( \mathbf{R} \) [114].

For the noisy cases, (2.36) must be rewritten [18]

\[
\mathbf{R}h = \mathbf{e},
\] (2.38)

which can be used to define a cost function

\[
\mathcal{J} = \| \mathbf{e} \|_2^2 = \mathbf{e}^T \mathbf{e}.
\] (2.39)

Correspondingly, \( \hat{h} \) can be obtained by minimizing (2.39) in the LS sense, that is,

\[
\hat{h} = \arg \min_h h^T \mathbf{R}^T \mathbf{R} h,
\] (2.40)

where \( \hat{h} \) can be found from eigenvectors of \( \mathbf{R} \) associated with the smallest eigenvalue. Since (2.40) denotes a closed-form solution to channel estimate, the time-domain sample index \( n \) can be removed for simplicity of presentation. However, it is worthwhile noting that (2.39) does not necessarily lead to a noise-robust solution to \( h \) as indicated in [100].

The MCLMS Algorithm

The MCLMS algorithm [18] is one of the first adaptive algorithms proposed for BSI in the context of acoustic signal processing. Similar to (2.38) and (2.39), the \textit{a priori} error
between channels \(m\) and \(l\) for the adaptive filter is derived as

\[
e_{ml}(n) = x_m^T(n)\hat{h}_l(n-1) - x_l^T(n)\hat{h}_m(n-1), \quad m = 1, 2, \ldots, M - 1, \quad l = m + 1.
\]

(2.41)

Based on the LMS algorithm \([15]\) as described in Section 2.2.3, the MCLMS algorithm minimizes the normalized error

\[
e_{ml}(n) = \frac{e_{ml}(n)}{\|\hat{h}(n)\|_2}
\]

(2.42)

iteratively \([115]\) such that \(\hat{h}(n)\) asymptotically approaches the true channel impulse response \(h\), where the imposed normalization on the error \(e_{ml}(n)\) by \(\|\hat{h}(n)\|_2\) in (2.42) arises from the unit-norm constraint for avoiding the trivial solution of \(\hat{h}(n) = 0_{M \times 1}\).

Using (2.42), a cost function with respect to the estimated impulse responses \(\hat{h}_m(n)\)

\[
J_{MCLMS}(n) = \sum_{m=1}^{M-1} \sum_{l=m+1}^{M} e_{ml}^2(n),
\]

(2.43)
can be obtained, from which the MCLMS algorithm can be described as \([18]\)

\[
\tilde{e}(n) = \sum_{m=1}^{M-1} \sum_{l=m+1}^{M} e_{ml}^2(n) = J_{MCLMS}(n)\|\hat{h}(n)\|_2^2,
\]

(2.44)

\[
\hat{h}(n) = \hat{h}(n-1) - \mu \nabla J_{MCLMS}(n)
= \frac{\hat{h}(n-1) - 2\mu [\tilde{R}(n)\hat{h}(n-1) - \tilde{e}(n)\hat{h}(n-1)]}{\|\hat{h}(n-1) - 2\mu [\tilde{R}(n)\hat{h}(n-1) - \tilde{e}(n)\hat{h}(n-1)]\|_2},
\]

(2.45)

where \(\tilde{R}(n)\) denotes the instantaneous estimate of \(R\) and is formed by \(\tilde{R}_{x_mx_l}(n) = x_m(n)x_l^T(n)\), \(m, l = 1, 2, \ldots, M\). A summary of MCLMS algorithm is shown in Table 2.2.

The mean convergence of MCLMS algorithm on SIMO acoustic systems to the true channel impulse responses was theoretically derived and empirically justified in \([18]\). However, it has limited robustness to the observation noise and common zeros. As a result, many follow-up algorithms have been developed to enhance the performance of MCLMS algorithm. In \([23]\), the frequency-domain version of MCLMS algorithm, namely the NMCFLMS algorithm, is proposed for efficient implementation using FFT \([116]\). The variable step-size (VSS) technique was also introduced in \([97]\) to improve the convergence performance of MCLMS algorithm. The issue of misconvergence due to the observation
### Table 2.2: The MCLMS Algorithm \[18\]

<table>
<thead>
<tr>
<th>Step</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initialization</strong></td>
<td></td>
</tr>
<tr>
<td>$\hat{h}_m(0)$</td>
<td>$[1 \ 0 \ \ldots \ 0]^T$, $m = 1, 2, \ldots, M$</td>
</tr>
<tr>
<td>$\hat{h}(0)$</td>
<td>$= \hat{h}(0)/\sqrt{M}$, (to satisfy the unit - norm constraint)</td>
</tr>
<tr>
<td><strong>Algorithm</strong></td>
<td></td>
</tr>
<tr>
<td>$\tilde{R}_{x_mx_l}(n)$</td>
<td>$= x_m(n)x_l^T(n)$</td>
</tr>
<tr>
<td>$\tilde{R}(n)$</td>
<td>$= \left[ \begin{array}{ccc} \sum_{m \neq 1} \tilde{R}<em>{x_mx_m}(n) &amp; -\tilde{R}</em>{x_2x_1}(n) &amp; \cdots &amp; -\tilde{R}<em>{x_Mx_1}(n) \ -\tilde{R}</em>{x_1x_2}(n) &amp; \sum_{m \neq 2} \tilde{R}<em>{x_mx_m}(n) &amp; \cdots &amp; -\tilde{R}</em>{x_Mx_2}(n) \ \vdots &amp; \vdots &amp; \ddots &amp; \vdots \ -\tilde{R}<em>{x_1x_M}(n) &amp; -\tilde{R}</em>{x_Mx_M}(n) &amp; \cdots &amp; \sum_{m \neq M} \tilde{R}_{x_mx_m}(n) \end{array} \right]$</td>
</tr>
<tr>
<td>$e_{ml}(n)$</td>
<td>$= x_m(n)\hat{h}_l(n-1) - x_l^T(n)\hat{h}_m(n-1)$, $m = 1, 2, \ldots, M - 1$, $l = m + 1$</td>
</tr>
<tr>
<td>$\epsilon_{ml}(n)$</td>
<td>$= e_{ml}(n)/|\hat{h}(n)|_2$</td>
</tr>
<tr>
<td>$J_{\text{MCLMS}}(n)$</td>
<td>$= \sum_{m=1}^{M-1} \sum_{l=m+1}^{M} \epsilon_{ml}^2(n)$</td>
</tr>
<tr>
<td>$\bar{e}(n)$</td>
<td>$= \sum_{m=1}^{M-1} \sum_{l=m+1}^{M} \epsilon_{ml}^2(n) = J_{\text{MCLMS}}(n)|\hat{h}(n)|_2^2$</td>
</tr>
<tr>
<td><strong>Filter update</strong></td>
<td></td>
</tr>
<tr>
<td>$\hat{h}(n)$</td>
<td>$= \hat{h}(n-1) - \mu \nabla J_{\text{MCLMS}}(n)$</td>
</tr>
<tr>
<td></td>
<td>$= \frac{\hat{h}(n-1) - 2\mu [\tilde{R}(n)\hat{h}(n-1) - \bar{e}(n)\hat{h}(n-1)]}{|\hat{h}(n-1) - 2\mu [\tilde{R}(n)\hat{h}(n-1) - \bar{e}(n)\hat{h}(n-1)]|_2}$</td>
</tr>
</tbody>
</table>

Noise was addressed in \[99, 100\], which motivated several subsequent algorithms \[101, 102, 103\] aiming to enhance the noise robustness.

### The CR-based Subspace Algorithm

The subspace algorithm presented here was developed for speech dereverberation \[19\], which finds the channel estimates from $x_m(n)$ using generalized singular value decomposition (GSVD) \[117\]. Consider first a two-channel noiseless case with reference to \[2.20\] where $b_m(n) = 0$, $\forall m$, for finite samples of microphone signals $x_m(n)$ such that $n = 0, 1, \ldots, N - 1$, the $ML \times ML$ correlation-like matrix defined in \[2.37\] can be approx-
imated as \( \hat{R}_\chi = XX^T/N \), where

\[
X = [X_2 - X_1]^T, \quad (2.46)
\]

\[
X_m = \begin{bmatrix}
  x_m(0) & \cdots & x_m(N-1) & \cdots & \cdots & 0 \\
  0 & x_m(0) & \cdots & x_m(N-1) & \cdots & 0 \\
  \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
  0 & \cdots & \cdots & x_m(0) & \cdots & x_m(N-1)
\end{bmatrix}. \quad (2.47)
\]

As before, the channel estimates can be found from the singular vector associated with the smallest singular value of \( \tilde{R}_\chi \).

To generalize this for the \( M \)-channel case where \( M > 2 \), \( X \) in (2.46) becomes \([19]\)

\[
X = \begin{bmatrix}
  X_2 & X_3 & \cdots & X_M & 0 & \cdots & 0 & \cdots & 0 \\
- X_1 & 0 & \cdots & 0 & X_3 & \cdots & X_M & 0 \\
  0 & - X_1 & \cdots & 0 & - X_2 & \cdots & 0 & \vdots \\
  \vdots & 0 & \ddots & \vdots & \ddots & \ddots & \ddots & \vdots \\
  \vdots & \vdots & \ddots & 0 & X_M \\
  0 & 0 & \cdots & - X_1 & \cdots & - X_2 & \cdots & - X_{M-1}
\end{bmatrix}, \quad (2.48)
\]

where \( 0 \) denotes a \( L \times (N + L - 1) \) null matrix and \( X \) is of dimension \( ML \times M(M - 1)(N + L - 1)/2 \). If additive noise is present, the following approximation for a sufficiently large \( N \) can be considered,

\[
\hat{R}_\chi = \tilde{R}_\chi + \tilde{R}_b, \quad (2.49)
\]

where \( \tilde{R}_b = \sigma_b^2 I \) is obtained, similarly to \( \tilde{R}_\chi \), from \( b(n) \) which has been assumed to be zero-mean and of variance \( \sigma_b^2 \). As a result, \( \hat{R}_\chi \) will not in general have a zero-valued singular value \([118]\). However, (2.49) indicates that, as long as \( b(n) \) is white, the minimal singular value of \( \hat{R}_\chi \) is equal to that of \( \tilde{R}_\chi \) with an additional value of \( \sigma_b^2 \) which would not perturb the order of all singular values. Therefore the solution to channel estimates still correspond to the singular vector associated with the smallest singular value of \( \hat{R}_\chi \).

The subspace algorithm thus exhibits a better noise robustness compared to the MCLMS algorithm, as will be demonstrated in Section 2.4.5.
2.4 Blind Multichannel Systems Identification

Finally, it is important to note that this CR-based subspace algorithm is different from the classic subspace algorithm developed in [93] in terms of the formulation of correlation matrix of the output signals as will be described in Section 5.2. However, they are in principle the same since both methods look for the solution to the channel estimates in the null space of the data correlation matrix [19].

2.4.5 Simulation Examples

In this section, simulation examples of blind identification of SIMO systems using MCLMS and subspace algorithm are presented. As described in Section 2.4.4, MCLMS algorithm converges slowly and suffers from misconvergence due to the sensitivity to observation noise [23, 99, 100]. The GSVD-based subspace algorithm, on the other hand, is computationally expensive for systems with a large number of FIR coefficients such as acoustic systems. Therefore, experiments in this section are carried out with the purpose of illustrating the performance of MCLMS and subspace algorithm, where a five-channel SIMO system with \( L = 32 \) FIR coefficients extracted from a standard Gaussian distribution is used, and the results are obtained through 10 independent trials. Figure 2.9 shows such five-channel system for a typical trial. A sequence of uncorrelated zero-mean WGN is used as the excitation signal and the received signal \( x(n) \) is obtained with additive WGN such that various SNR as depicted in each experiment is achieved. The NPM \( \eta'(n) \) defined in (2.27) is used to evaluate the BSI performance.

Figure 2.9: The example five-channel SIMO system of length \( L = 32 \).
2.5 Channel Equalization

Channel equalization aims at designing inverse filters to deconvolve, or compensate for in some manner, the effects of linear convolution between the system impulse response and the unknown input signal such as shown in (2.20). It is therefore essential for the estimated channel responses to be equalized so as to achieve speech dereverberation. In

Figure 2.10 first shows the BSI performance of MCLMS algorithm in NPM for various SNR values, where a step-size of $\mu = 0.8$ is used. As can be seen, the performance of MCLMS algorithm degrades with decreasing SNR values. It can also be seen that the steady-state performance matches the noise floor for each case. Then, the BSI performance of subspace algorithm against SNR is shown in Fig. 2.11(a), where a similar relationship between NPM value and SNR can be observed. The comparatively better noise robustness for the subspace algorithm over MCLMS algorithm, as described in Section 2.4.4, is clearly indicated. Further result in Fig. 2.11(b) demonstrates the effect of channel length on the performance of subspace algorithm in NPM for the case of SNR=50 dB. It can be seen that the subspace algorithm performs worse for longer channels. One of the reasons for this is because of the presence of near-common zeros (NCZs), as will be described in Chapter 4.

2.5 Channel Equalization
In this section, the problem of channel equalization is briefly reviewed. Similar to (2.21), define \( H_{eq,m} \) as the \( L_g \times (L + L_g - 1) \) convolutional matrix for channel \( m \), the corresponding FIR inverse filter \( g_m \) with \( L_g \) coefficients given by

\[
g_m = [g_{m,0}, g_{m,1}, \ldots, g_{m,L_g-1}]^T, \quad m = 1, 2, \ldots, M
\]  

needs to satisfy, in principle,

\[
H_{eq,m}^T g_m = \delta_D,
\]  

where \( \delta_D \) is the vector of delayed delta function of length \( \mathcal{L} = L + L_g - 1 \) given by

\[
\delta_D = [0_{1\times \tau}, \theta_{1\times (\mathcal{L}-\tau-1)}]^T
\]  

with \( \tau \) being the modelling delay and \( \theta \in \mathbb{Z}^+ \). Minimizing the error function

\[
\hat{g}_m = \min_{g_m} \| H_{eq,m}^T g_m - \delta_D \|_2^2
\]  

formed from (2.51) leads to the classic single-channel LS (SCLS) algorithm [119].

In the z-domain, (2.51) can be rewritten

\[
G_{eq,m}(z) = \frac{\theta z^{-\tau}}{H_{eq,m}(z)},
\]  

which indicates that a stable \( G_{eq,m}(z) \) can be obtained by replacing zeros of \( H_{eq,m}(z) \) with
poles provided that $H_{eq,m}(z)$ is minimum phase, i.e., its zeros should all be inside the unit circle. In practice, SCLS is not widely applicable because: (i) The room impulse responses are in general non-minimum phase [120] so (2.54) does not give a stable causal solution for $G_{eq,m}(z)$; (ii) Inverse filters designed from inaccurate estimates of $H_{eq,m}(z)$ will cause distortion in the equalized signal [121]; (iii) Room impulse responses often contain spectral nulls that, after equalization, give strong peaks in the spectrum causing noise amplification [71]; and (iv) The length of room impulse responses are usually with several thousands of FIR coefficients which is computationally expensive for LS-type algorithms.

An alternative single-channel equalization technique is the homomorphic equalization algorithm [122]. This technique decomposes the room impulse response into minimum phase and all-pass components where the former can be inverted using (2.53) and the latter can be equalized, for example, using a matched filter [122]. In a comparative study between these two methods [119], the authors concluded that SCLS, although sometimes less accurate than homomorphic inversion, is more efficient in practice. In addition, the approximate nature due to partial equalization of the deep spectral nulls allows these single-channel approaches to be less sensitive to noise and inaccurate channel estimates [25]. They are advantageous in mitigating problems regarding the aforementioned issues (ii) and (iii). However, single-channel methods normally result in large processing delay, which can be problematic for extremely long and non-causal filters [123].

Multichannel equalization techniques have therefore been developed, where instead of invoking (2.53) for individual channel, an exact system inversion can be obtained using the Bezout’s theorem [123] such that

$$
\sum_{m=1}^{M} H_{eq,m}^T g_m = \delta_D,
$$

(2.55)

which relaxes the minimum-phase constraint since (2.51) does not need to be satisfied. This has led to the classic multichannel inverse theorem (MINT) [123] which states that as long as multiple channels do not share common zeros and $L_g \geq \lceil (L - 1)/(M - 1) \rceil$ with $\lceil \cdot \rceil$ being the ceiling operator, a system of $M$ inverse filters can be obtained by solving (2.55) as

$$
\hat{g} = H_{eq}^+ \delta_D,
$$

(2.56)
where \( \hat{g} = [\hat{g}_1^T \hat{g}_2^T \ldots \hat{g}_M^T]^T \), \( \mathbf{H}_{\text{eq}} = [\mathbf{H}_{\text{eq},1} \mathbf{H}_{\text{eq},2} \ldots \mathbf{H}_{\text{eq},M}] \), and \([\cdot]^+\) denotes Moore-Penrose pseudo-inverse operation such that \( \mathbf{H}_{\text{eq}}^+ = (\mathbf{H}_{\text{eq}}^T \mathbf{H}_{\text{eq}})^{-1} \mathbf{H}_{\text{eq}}^T \).

Forming \( \hat{\mathbf{H}}_{\text{eq}} \) from estimated channel responses obtained by BSI algorithms, an estimate of the source signal \( \hat{s}(n) \) can be obtained as

\[
\hat{s}(n) = \sum_{m=1}^{M} \hat{\mathbf{G}}_{\text{eq},m}^T \mathbf{x}_m(n)
\]

(2.57)

to achieve dereverberation, where \( \hat{\mathbf{G}}_{\text{eq},m} \) is the convolutional matrix of \( \hat{g}_m \) formed in a similar manner as in (2.21). It is observed from (2.57) that, the BSI error inherited in \( \hat{\mathbf{H}}_{\text{eq}} \) and the additive noise contained in \( \mathbf{x}_m(n) \) can directly affect the accuracy of \( \hat{s}(n) \). To overcome this, regularization parameters were introduced to MINT in [124]. Efficient methods based on adaptive methods and oversampled filter banks have also been developed to reduce the computational complexity of inverting \( \mathbf{H}_{\text{eq}} \) in [125, 126, 127]. The effect of common zeros on multichannel equalization algorithms such as the MINT will be demonstrated in Chapter 4 and Chapter 5. Figure 2.12 depicts a schematic of speech dereverberation using BSI and channel equalization algorithms.

2.5.1 Performance Evaluation

In general, the performance of channel equalization algorithms can be measured by comparing the estimated source signal \( \hat{s}(n) \) as derived in (2.57) with the true input signal.
The signal-to-distortion ratio (SDR) \[ \text{SDR} = 10 \log_{10} \left( \frac{\sum_{n=0}^{N_s-1} s^2(n)}{\sum_{n=0}^{N_s-1} (s(n) - \hat{s}(n))^2} \right) \text{ dB} \] (2.58)
is often used for finite \((N_s < \infty)\) samples of signals \(s(n)\) which is not necessary speech.

For speech dereverberation, however, the quality of the recovered speech \(\hat{s}(n)\) can be measured in terms of subjective and objective parameters \[128\]. The subjective measurements such as the mean opinion score (MOS) \[128\] suggest direct link between human perception and speech quality. Objective measurements, however, are essential to provide a rapid and reliable evaluation of the performance of speech processing algorithms. In this thesis, the normalized Bark spectral distortion (BSD) defined by \[129, 74\]

\[ \text{BSD} = \frac{\sum_{k=0}^{K-1} \sum_{n=kN_B}^{kN_B + N_B - 1} (B_s(k,n) - \hat{B}_s(k,n))^2}{\sum_{k=0}^{K-1} \sum_{n=kN_B}^{kN_B + N_B - 1} (B_s(k,n))^2} \] (2.59)
is employed, where \(N_B\) is the frame length in samples, \(B_s(k,n)\) and \(\hat{B}_s(k,n)\) are the Bark spectra of the clean speech \(s(n)\) and the estimated speech \(\hat{s}(n)\), respectively. BSD is known to be highly correlated with MOS measures for speech coders with a factor (on a normalized scale between 0 and 1) of 0.9 \[129\]. It measures the average squared Euclidean distance between spectral vectors of the original and estimated speech utterances, and takes into account auditory frequency warping, critical band integration, amplitude sensitivity variations, and subjective loudness \[129\]. Results in \[130\] have shown that BSD score successfully captures the distortion in speech caused by the reverberation tail, although it is less sensitive to coloration. An evaluation approach for dereverberation algorithms which considers the effects of coloration has recently been developed in \[131\].

### 2.5.2 Simulation Examples

Simulation examples are now presented to demonstrate speech dereverberation using the MINT algorithm \[123\]. A recorded two-channel SIMO acoustic system obtained from the MARDY database \[130\] is used, where the obtained impulse responses are down-sampled to 8 kHz and truncated to 1600 FIR coefficients. A speech sample containing both male and female utterances is extracted from the APLAWD database \[68\] and
used with a sequence of additive WGN giving an SNR of 60 dB to avoid the noise effect [124]. For simplicity of presentation, exact room impulse responses are assumed to be available. Figure 2.13 shows the results for the first 6 second of the speech, where Fig. 2.13(a) and Fig. 2.13(b) show the time sequence and spectrogram of the clean speech signal. The spectrogram displays the magnitude response of the short-time Fourier transform (STFT) of the speech signal [132]. In order to enhance the spectral resolution, longer window are used resulting in a narrowband spectrogram [133]. As can be seen from Fig. 2.13(c), the spectral distortion caused by reverberation blurs the spectrogram across all frequency range as the boundaries between each frequency for the clean speech shown in Fig. 2.13(b) become unclear. Time decay of frequency components caused by reverberation tail is also observed in most of the frequency bands, especially at the low-frequency range. The recovered speech is shown in Fig. 2.13(d), from which it can be seen that a good dereverberation performance is achieved, although a small amount of spectral distortion occurs around 3 kHz. BSD scores and SDR values for reverberant and recovered speech are also shown in Fig. 2.13(c) and Fig. 2.13(d).

2.6 Summary

In this chapter, a literature overview of system identification techniques and their applications in speech enhancement has been presented. The problem of supervised (non-blind) single-channel system identification for NEC was first addressed and formulated. Classic adaptive algorithms including NLMS and MMax-NLMS algorithm have been derived and studied through simulation examples. These algorithms offer a framework for more sophisticated algorithms to be built on. In Chapter 3 a fast-converging frequency-domain adaptive algorithm with low complexity and low delay will be developed for identifying sparse network impulse responses for NEC.

A literature survey of BSI techniques was then presented. In particular, the problem of blind identification of SIMO acoustic systems for speech dereverberation have been formulated, where channel identifiability conditions for BSI algorithms were also addressed. These conditions will be investigated in more detail in the forthcoming chapters. Two classic CR-based BSI algorithms, the MCLMS and subspace algorithm, have been reviewed and their performances demonstrated by simulation examples.
2.6 Summary

Last but not least, channel equalization algorithms were briefly reviewed. These algorithms are essential in order to recover clean speech signals for speech dereverberation. An simulation example using real room impulse responses demonstrated the performance of the classic MINT algorithm.

Figure 2.13: Dereverberation performance using the MINT algorithm and exact system estimates with (a) the time sequence of clean speech, and spectrograms of (b) clean speech, (c) reverberant speech, (d) recovered speech.
Chapter 3

Adaptive Sparse System Identification for Network Echo Cancellation

The popularity of VoIP coupled with an increasing expectation for natural speech communication over packet-switched networks has called for an improved QoS in VoIP. This has resulted in an increasing demand for efficient network echo cancellers for IP networks as described in Section 2.2.1. In this chapter, a supervised frequency-domain adaptive sparse system identification algorithm for NEC is developed. Incorporating partial update tap-selection criteria with a multidelay filtering (MDF) structure, this algorithm exhibits a low delay, low complexity and fast converging performance.

This chapter is organized as follows: In Section 3.1 the literature of sparse system identification for NEC is reviewed, where the sparseness of impulse responses is defined and quantified. The sparse partial update LMS (SPNLMS) [20] and MDF [21] algorithms are then reviewed in Section 3.2 which form the basis of the algorithm to be developed. Section 3.3 describes of how MMax tap selection [16] can be integrated into the MDF structure for complexity reduction. This can be achieved either in time domain or frequency domain, leading to a tradeoff in terms of complexity and convergence performance for the resultant MMax-MDF algorithm. The incorporation of SP tap selection [20] into MMax-MDF algorithm for fast convergence is presented in Section 3.3.2. Such proce-
3.1 Introduction to Sparse System Identification

The problem of NEC has been formulated in Section 2.2. This formulation is briefly reproduced here for convenience of presentation. With reference to Fig. 3.1, the \( L_{nw} \)-tap unknown network impulse response \( \mathbf{h}_{nw} \) and the adaptive filter \( \mathbf{\hat{h}}_{nw}(n) \) are given by

\[
\mathbf{h}_{nw} = [h_{nw,0} \ h_{nw,1} \ \ldots \ h_{nw,L_{nw}-1}]^T, \quad (3.1)
\]

\[
\mathbf{\hat{h}}_{nw}(n) = [\hat{h}_{nw,0}(n) \ \hat{h}_{nw,1}(n) \ \ldots \ \hat{h}_{nw,L_{nw}-1}(n)]^T. \quad (3.2)
\]

The resultant near-end signal is given by

\[
x_s(n) = \mathbf{h}_{nw}^T \mathbf{s}(n) + b_s(n), \quad (3.3)
\]
where $s(n) = [s(n) \ s(n-1) \ \ldots \ s(n-L_{nw} + 1)]^T$ is the far-end tap-input vector and $b_i(n)$ is the additive noise assumed to be uncorrelated with $s(n)$ such as described in (2.4).

In VoIP systems where traditional telephony devices are connected to the packet-switched network, the network impulse response is typically of length 64-128 ms [7], and exhibits an “active” region in the range of only 8-12 ms duration located within the impulse response at an unknown delay. Consequently, the network impulse response is dominated by “inactive” regions where magnitudes are close to zero, hence making it sparse. This sparseness is principally due to the presence of the bulk delay caused by unknown network propagation, encoding and jitter buffer delays [9]. As a result, a large fraction of the energy of a sparse impulse response is concentrated on a small fraction of its duration. Figure 3.2 shows a benchmark sparse network impulse response [134]. To quantify such sparseness, the sparseness measure given by [135, 136]

$$\varsigma(h_{nw}) = \frac{L_{nw}}{L_{nw} - \sqrt{L_{nw}} \left( 1 - \frac{\|h_{nw}\|_1}{\sqrt{L_{nw}}\|h_{nw}\|_2} \right)}, \quad \varsigma(\cdot) \in [0, 1], \quad (3.4)$$

is commonly employed, where $\| \cdot \|_1$ denotes $l_1$-norm such that

$$\|h_{nw}\|_1 = \sum_{i=0}^{L_{nw}-1} |h_{nw,i}|. \quad (3.5)$$

It has been shown in [14] that $\varsigma(h_{nw})$ increases with the sparseness of $h_{nw}$, i.e.,

$$\varsigma(\delta_D) = 1, \quad \varsigma(1) = 0, \quad (3.6)$$
where $\delta_D$ is the delta function as defined by (2.52) and $\mathbf{1} = [1 \ 1 \ldots \ 1]^T$. Additionally, $\varsigma(\cdot)$ is independent of the sorting order of the impulse response coefficients since the sparseness is all about the dynamic range of these coefficients [14]. In fact, it has recently been recognized that most, if not all, acoustic impulse responses are to some extent sparse. In [137], it was demonstrated that the sparseness of a room impulse response can be related to the distance between the loudspeaker and the microphone.

The proportionate NLMS (PNLMS) algorithm [138] is one of the first algorithms to exploit the sparse nature of the network impulse responses to achieve fast convergence performance. By updating each filter coefficient with a step-size that is proportional to its magnitude, the PNLMS algorithm outperforms classic adaptive algorithms with a uniform step-size across all filter coefficients such as the NLMS algorithm [138]. However, fast initial convergence for the PNLMS algorithm is at the expense of a significantly reduced rate of convergence in later iterations, which is due to the less frequent updating of the filter coefficients with small magnitudes. To mitigate this problem, improved PNLMS (IPNLMS) [139] and the improved IPNLMS [140] algorithm have been developed. These algorithms share the same characteristic of introducing a controlled mixture of proportionate (PNLMS) and non-proportionate (NLMS) adaptations, hence performing better than PNLMS for impulse responses with varying degrees of sparseness.

As described in Section 2.2, computational efficiency has become a core issue in designing adaptive algorithms for NEC. The PNLMS and IPNLMS algorithms require approximately $O(3L)$ and $O(4L)$ number of multiplications per iteration respectively compared to $O(2L)$ for the NLMS algorithm. The SPNLMS algorithm [20] was then proposed to reduce the computational complexity of PNLMS and IPNLMS by combining two adaptation strategies: sparse adaptation for improving the rate of convergence and MMMax partial update [16] for complexity reduction. Under the sparse partial (SP) adaptation, only those taps corresponding to tap-inputs and filter coefficients both having large magnitudes are updated for the majority of iterations. Otherwise, the algorithm employs the MMMax tap selection [16] on the coefficients by only updating those filter taps corresponding to the $M_{\text{mmax}} < L_{\text{nw}}$ largest magnitude of tap-inputs.

Aiming at an efficient implementation, frequency-domain adaptive algorithms have been developed. First introduced in [116], frequency-domain adaptive filtering such
as the fast-LMS (FLMS) algorithm [141] has been well known to offer an attractive means for efficient implementation. In contrast to time-domain algorithms such as LMS and NLMS as discussed in Section 2.2.3, frequency-domain algorithms incorporate block updating strategies whereby the FFT technique [22] is used together with the overlap-save method [142,133]. A direct consequence of block processing is the reduction in computational complexity since the filter output and tap updates are computed only after a block of data has been accumulated. In addition, the use of the FFT for computing the discrete Fourier transform (DFT) so as to perform linear convolution and gradient estimation further increases the efficiency of such algorithms.

However, one of the main drawbacks of frequency-domain approaches is the delay introduced between the input and output, which is generally equal to the adaptive filter length $L_{nw}$. This problem is significant since reducing the algorithmic delay for VoIP applications is crucial in respect to the delay budget for network echo cancellers. Frequency-domain adaptive algorithms with low delay are thus desirable especially for the identification of long network impulse responses. The MDF algorithm [21] has thus been developed in the context of AEC. It partitions an adaptive filter of length $L_{nw}$ into $K$ blocks each of length $L_M$ so that the processing delay is reduced by a factor of $L_{nw}/L_M$ compared to FLMS. The benefit of low delay for MDF over FLMS in the context of NEC has been shown in [143].

### 3.2 Review of SPNLMS and MDF Algorithms

#### The SPNLMS Algorithm

The SPNLMS algorithm [20] utilizes the sparse nature of network impulse responses and incorporates two updating strategies: MMax tap selection [16] for complexity reduction and SP adaptation for fast convergence. Although it is often expected that adapting filter algorithms using partial update strategies suffer from degradation in convergence performance, it was shown in [20] that such degradation can be offset by the SP tap selection. The updating equation for SPNLMS is given by

$$
\hat{h}_{nw}(n) = \hat{h}_{nw}(n - 1) + \mu \frac{Q(n - 1)s(n)e(n)}{s^T(n)Q(n - 1)s(n) + \delta}, \quad (3.7)
$$
3.2 Review of SPNLMS and MDF Algorithms

where $\mu$ is the step-size, $\delta$ is the regularization parameter, and the a priori error $e(n)$ is defined by (2.6). For SPNLMS, the $L_{nw} \times L_{nw}$ tap-selection control matrix $Q(n)$ in (3.7) determines the step-size gain for each filter coefficient and is dependent on the MMax and SP updating strategies. The relative significance of these strategies is controlled by the variable $T \in \mathbb{Z}^+$ such that for $\text{mod}(n, T) = 0$, elements $q_i(n)$ of $Q(n)$ for $i = 0, 1, \ldots, L_{nw} - 1$ are given by (2.16), i.e.,

$$q_i(n) = \begin{cases} 1 & i \in \{ \text{indices of the } M_{m\text{max}} \text{ maxima of } |s(n - i)| \}, \\ 0 & \text{otherwise}, \end{cases} \quad (3.8)$$

and for $\text{mod}(n, T) \neq 0$,

$$q_i(n) = \begin{cases} 1 & i \in \{ \text{indices of the } M_{\text{sp}} \text{ maxima of } |s(n - i)\hat{h}_{nw,i}(n - 1)| \}, \\ 0 & \text{otherwise}. \end{cases} \quad (3.9)$$

The variables $M_{m\text{max}}$ and $M_{\text{sp}}$ define the number of selected taps for MMax and SP respectively. It has been shown in [20] that the SPNLMS algorithm achieves lower complexity than NLMS including a modest overhead by the sorting operations for MMax tap selection given by (3.8).

The MDF Algorithm

The MDF structure [21] shown in Fig. 3.3 was developed to mitigate the delay problem inherent in FLMS [141]. Defining $m$ as the frame index, the adaptive filter

$$\hat{h}_{nw}(m) = [\hat{h}_{nw,0}(m) \hat{h}_{nw,1}(m) \ldots \hat{h}_{nw,L_{nw} - 1}(m)]^T$$

(3.10)

is partitioned into $\mathcal{K}$ subfilters each of length $L_M$ for $L_{nw} = \mathcal{K} L_M$ and $\mathcal{K} \in \mathbb{Z}^+$ such that

$$\hat{h}_{nw}(m) = [\hat{h}_{nw,0}(m) \hat{h}_{nw,1}(m) \ldots \hat{h}_{nw,\mathcal{K}-1}(m)]^T,$$

$$= \begin{bmatrix} \hat{h}_{nw,0}(m) & \ldots & \hat{h}_{nw,L_M-1}(m) & \ldots & \hat{h}_{nw,L_{nw}-L_M}(m) & \ldots & \hat{h}_{nw,L_{nw}-1}(m) \\ \hat{h}_{nw,0}(m) & \ldots & \hat{h}_{nw,L_M-1}(m) & \ldots & \hat{h}_{nw,L_{nw}-L_M}(m) & \ldots & \hat{h}_{nw,L_{nw}-1}(m) \end{bmatrix}^T \quad (3.11)$$
Let $k = 0, 1, \ldots, K - 1$ be the block index, the $k$th subfilter in (3.11) can be expressed as

$$\tilde{h}_{nw,k}(m) = [\tilde{h}_{nw,kL_M}(m) \tilde{h}_{nw,kL_M+1}(m) \ldots \tilde{h}_{nw,kL_M+L_M-2}(m)]^T.$$

(3.12)

As a result of such partitioning, the delay for the MDF is reduced by a factor of $K$ compared to FLMS. In addition, the smaller block size for $L_M < L_{nw}$ allows filter coefficients to be updated more frequently (once every $L_M$ samples compared to $L_{nw}$ for FLMS, hence resulting in faster convergence. For $L_M = L_{nw}$ and $K = 1$, MDF is equivalent to FLMS [141].

To describe the MDF algorithm, define the tap-input block vector for the $m$th frame

$$s(m) = [s(mL_M) s(mL_M - 1) \ldots s(mL_M - L_{nw} + 1)]^T.$$

(3.13)

Concatenating $L_M$ offset versions of this tap-input sequence, a $L_{nw} \times L_M$ matrix

$$S(m) = [s(mL_M) s(mL_M + 1) \ldots s(mL_M + L_M - 1)],$$

$$= \begin{bmatrix}
    s(mL_M) & s(mL_M + 1) & \cdots & s(mL_M + L_M - 1) \\
    s(mL_M - 1) & s(mL_M) & \cdots & s(mL_M + L_M - 2) \\
    \vdots & \vdots & \vdots & \vdots \\
    s(mL_M - L_{nw} + 1) & s(mL_M - L_{nw} + 2) & \cdots & s(mL_M + L_M - L_{nw})
\end{bmatrix}$$

(3.14)
is obtained. The filter output can then be expressed as the convolution between the input sequence and the filter coefficients given by the \( L_M \times 1 \) vector

\[
\hat{x}_s(m) = S^T(m) \hat{h}_{nw}(m - 1)
\]

\[
= \begin{bmatrix}
  s(mL_M) & s(mL_M - 1) & \cdots & s(mL_M - L_{nw} + 1) \\
  s(mL_M + 1) & s(mL_M) & \cdots & s(mL_M - L_{nw} + 2) \\
  \vdots & \vdots & \ddots & \vdots \\
  s(mL_M + L_M - 1) & s(mL_M + L_M - 2) & \cdots & s(mL_M + L_M - L_{nw})
\end{bmatrix}
\times
\begin{bmatrix}
  \hat{h}_{nw,0}(m - 1) \\
  \hat{h}_{nw,1}(m - 1) \\
  \vdots \\
  \hat{h}_{nw,L_{nw} - 1}(m - 1)
\end{bmatrix}^T,
\]

\[
= [\hat{x}_s(mL_M) \ \hat{x}_s(mL_M + 1) \ \ldots \ \hat{x}_s(mL_M + L_M - 1)]^T,
\]

from which the \textit{a priori} block error can be obtained as

\[
e(m) = x_s(m) - \hat{x}_s(m),
\]

where \( x_s(m) = [x_s(mL_M) \ x_s(mL_M + 1) \ \ldots \ x_s(mL_M + L_M - 1)]^T \) is the \( L_M \times 1 \) received near-end signal.

As shown in Fig. 3.3, the overlap-save method [133] is employed in the MDF structure. It is typical to use a 50% overlap, as is used in this thesis, since it is considered to be the most efficient case and clear for presentation [15]. To implement this, define the \( 2L_M \times 1 \) tap-input vector

\[
s(m - k) = [s(mL_M - kL_M - L_M) \ \ldots \ s(mL_M - kL_M + L_M - 1)]^T
\]

as the kth block of the mth frame tap-input block sequence \( s(m) \) defined by (3.13), a \( 2L_M \times 2L_M \) matrix

\[
D(m - k) = \text{diag} \{ F_{2L_M} s(m - k) \} = \text{diag} \{ s(m - k) \}
\]

(3.18)

can be constructed with diagonal elements containing the Fourier transform of \( s(m - k) \),
where $F_{2L_M}$ denotes the $2L_M \times 2L_M$ DFT matrix \[133\]

$$F_{2L_M} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_{2L_M} & W_{2L_M}^2 & \cdots & W_{2L_M}^{2L_M-1} \\ 1 & W_{2L_M}^2 & W_{2L_M}^4 & \cdots & W_{2L_M}^{2(2L_M-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_{2L_M}^{2L_M-1} & W_{2L_M}^{2(2L_M-1)} & \cdots & W_{2L_M}^{(2L_M-1)^2} \end{bmatrix}$$ (3.19)

for $W_{2L_M} = e^{-j\pi/2L_M}$ and $j = \sqrt{-1}$ such that $F_{2L_M}^{-1} = (1/(2L_M)) F_{2L_M}^H$ with $\{ \cdot \}^H$ being the Hermitian operator. Employing the following frequency-domain quantities \[7\]

$$x_s(m) = F_{2L_M} \begin{bmatrix} 0_{L_M \times 1} \\ x_s(m) \end{bmatrix},$$ (3.20a)

$$\hat{h}_{nw,k}(m) = F_{2L_M} \begin{bmatrix} \hat{h}_{nw,k}(m) \\ 0_{L_M \times 1} \end{bmatrix},$$ (3.20b)

$$e(m) = F_{2L_M} \begin{bmatrix} 0_{L_M \times 1} \\ e(m) \end{bmatrix},$$ (3.20c)

$$W_{2L_M}^{01} = \begin{bmatrix} I_{L_M \times L_M} & 0_{L_M \times L_M} \\ 0_{L_M \times L_M} & I_{L_M \times L_M} \end{bmatrix},$$ (3.20d)

$$W_{2L_M}^{10} = \begin{bmatrix} I_{L_M \times L_M} & 0_{L_M \times L_M} \\ 0_{L_M \times L_M} & 0_{L_M \times L_M} \end{bmatrix},$$ (3.20e)

$$G^{01} = F_{2L_M} W_{2L_M}^{01} F_{2L_M}^{-1},$$ (3.20f)

$$G^{10} = F_{2L_M} W_{2L_M}^{10} F_{2L_M}^{-1},$$ (3.20g)

the MDF algorithm can be described as \[21\]

$$e(m) = x_s(m) - G^{01} \sum_{k=0}^{K-1} D(m-k) \hat{h}_{nw,k}(m-1),$$ (3.21)

$$S(m) = \lambda S(m-1) + (1-\lambda) D^*(m) D(m),$$ (3.22)

$$P(m) = S(m) + \delta I_{2L_M \times 2L_M} = \text{diag} \{ p_0(m) \ p_1(m) \ \ldots \ p_{2L_M-1}(m) \},$$ (3.23)

$$\hat{h}_{nw,k}(m) = \hat{h}_{nw,k}(m-1) + \mu G^{10} D^*(m-k) P^{-1}(m) e(m),$$ (3.24)

where $0 \ll \lambda < 1$ is the forgetting factor, $\mu = \beta (1-\lambda)$ is the step-size with $0 < \beta \leq 1$ \[21\]
3.2 Review of SPNLMS and MDF Algorithms

Table 3.1: The MDF Algorithm \[21\]

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta$</td>
<td>$20\sigma_s^2 L_M/L_{nw}$</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>$\left[1 - \frac{1}{3L_{nw}}\right]^{L_M}$</td>
</tr>
<tr>
<td>$\mu$</td>
<td>$\beta(1-\lambda), \ 0 &lt; \beta \leq 1$</td>
</tr>
<tr>
<td>$\mathbf{S}(0)$</td>
<td>$\sigma_s^2/100$</td>
</tr>
<tr>
<td>$\mathbf{h}_{nw,k}(m)$</td>
<td>$\begin{bmatrix} \mathbf{h}<em>{nw,kL_M}(m) &amp; \mathbf{h}</em>{nw,kL_M+1}(m) &amp; \ldots &amp; \mathbf{h}_{nw,kL_M+L_M-1}(m) \end{bmatrix}^T$</td>
</tr>
<tr>
<td>$\mathbf{h}_{nw,k}(m)$</td>
<td>$\mathbf{F}<em>{2L_M} \begin{bmatrix} \mathbf{h}</em>{nw,k}(m) \ 0_{L_M \times 1} \end{bmatrix}$</td>
</tr>
<tr>
<td>$k$</td>
<td>$0, 1, \ldots, K - 1$</td>
</tr>
</tbody>
</table>

Algorithm

\[
\mathbf{D}(m-k) = \text{diag}\{\mathbf{F}_{2L_M} \mathbf{s}(m-k)\} = \text{diag}\{\mathbf{g}(m-k)\}
\]
\[
\mathbf{e}(m) = \mathbf{x}(m) - \mathbf{G}^0 \sum_{k=0}^{K-1} \mathbf{D}(m-k) \mathbf{h}_{nw,k}(m-1)
\]
\[
\mathbf{S}(m) = \lambda \mathbf{S}(m-1) + (1 - \lambda) \mathbf{D}^*(m) \mathbf{D}(m)
\]
\[
\mathbf{P}(m) = \mathbf{S}(m) + \delta \mathbf{I}_{2L_M \times 2L_M} = \text{diag}\{p_0(m) \ p_1(m) \ldots p_{2L_{nw}-1}(m)\}
\]

Filter update

\[
\mathbf{h}_{nw,k}(m) = \mathbf{h}_{nw,k}(m-1) + \mu \mathbf{G}^0 \mathbf{D}^*(m-k) \mathbf{P}^{-1}(m) \mathbf{e}(m)
\]

and $| \cdot |^*$ denotes complex conjugation. Letting $\sigma_s^2$ be the input signal variance, the initial regularization parameters \[7\] are $\mathbf{S}(0) = \sigma_s^2/100$ and $\delta = 20\sigma_s^2 L_M/L_{nw}$. It is worthwhile noting that the regularization parameter $\delta$ can also be varying with the input signal as described in \[144\], especially when $\sigma_s^2$ is unknown or the input signal has a large dynamic range. For nonstationary signals in particular, $\sigma_s^2$ can be estimated at each iteration by $\mathbf{s}^T(m-k)\mathbf{s}(m-k)/(2L_M)$ where $\mathbf{s}(m-k)$ is defined by \[3.17\]. The MDF algorithm is summarized in Table \[3.1\] and its convergence analysis can be found in \[145\].
3.3 The Sparse Partial Update Multidelay Filtering Algorithm

The motivation of employing the MDF approach is to combine its inherent low-delay characteristics with the fast convergence and low complexity brought about by SP and MMax tap-selection criteria. This can be achieved by first integrating MMax tap selection given in (3.8) into MDF so as to form the MMax-MDF algorithm. Introducing SP tap selection given by (3.9) to MMax-MDF, however, is not straightforward. An illustrative example will be presented in this section to show how the sparse nature of the impulse response can be exploited in the frequency domain to achieve the development of the SPMMax-MDF algorithm. Based on the MDF framework, these algorithms can be described by (3.21), (3.22), (3.23) and

$$\hat{h}_{nw,k}(m) = \hat{h}_{nw,k}(m-1) + \mu G^{10} \tilde{D}^*(m-k)P^{-1}(m)e(m).$$  (3.25)

The difference between (3.24) and (3.25) is that the latter employs $\tilde{D}^*(m-k)$. The following will describe how this $2L_M \times 2L_M$ diagonal matrix can be derived for the cases of MMax and SP tap-selection criteria.

3.3.1 The MMax-MDF Algorithm

The diagonal matrix $\bar{D}(m-k)$ for MMax-MDF can be obtained from performing tap selection in either time domain or frequency domain, which is denoted as MMax-MDF$_t$ and MMax-MDF$_f$ respectively. For the time-domain selection approach, elements for the $2L_M \times 2L_M$ diagonal tap-selection control matrix $Q(m)$ can be expressed by subselecting from elements in $s(m-k)$ as defined in (3.17) such that, for $1 \leq M_{mmax} \leq 2L_M$,

$$q_i(m) = \begin{cases} 
1 & i \in \{\text{indices of the } M_{mmax} \text{ maxima of } |s(mL_M - kL_M - L_M + i)|\}, \\
0 & \text{otherwise}
\end{cases}$$  (3.26)

with $i = 0, 1, \ldots, 2L_M - 1$. Utilizing $Q(m)$, elements of $\bar{D}(m-k)$ for MMax-MDF$_f$ can be expressed as

$$\bar{D}(m-k) = \text{diag}\{F_{2L_M}Q(m-k)s(m-k)\}.$$  (3.27)

The MMax-MDF$_f$ algorithm is thus described by (3.21), (3.22), (3.27) and (3.25).
For frequency-domain selection, the chosen frequency bins correspond to the $M_{m\text{max}}$ largest magnitudes of the Fourier transformed tap-input vector across all blocks for $k = 0, 1, \ldots, K - 1$. This is achieved by concatenating the Fourier transform of the $m$th frame tap-input vector $\mathbf{s}(m - k)$ defined in (3.18) for all $K$ blocks, i.e.,

$$\tilde{\mathbf{S}}(m) = \left[ \tilde{s}^T(m) \tilde{s}^T(m - 1) \ldots \tilde{s}^T(m - K + 1) \right]^T = \left[ \tilde{s}_0(m) \tilde{s}_1(m) \ldots \tilde{s}_{2L_{nw} - 1}(m) \right]^T$$

(3.28)

so that elements in the tap-selection matrix $\mathbf{Q}(m)$ can be given, for $i = 0, 1, \ldots, 2L_{nw} - 1$,

$$q_i(m) = \begin{cases} 1 & i \in \{\text{indices of the } M_{m\text{max}} \text{ maxima of } |s_i(m)|\}, \\ 0 & \text{otherwise}, \end{cases}$$

(3.29)

where for this case of frequency-domain selection $1 \leq M_{m\text{max}} \leq 2L_{nw}$. Denote then a $2L_{nw} \times 1$ vector $\tilde{\mathbf{S}}(m)$ containing the subselected frequency-domain tap-input vector as

$$\tilde{\mathbf{S}}(m) = \mathbf{Q}(m)\mathbf{S}(m) = \left[ \tilde{s}_0(m) \tilde{s}_1(m) \ldots \tilde{s}_{2L_{nw} - 1}(m) \right]^T,$$

(3.30)

the $2L_M \times 2L_M$ diagonal matrix $\tilde{\mathbf{D}}(m - k)$ for this frequency-domain tap-selection approach can be obtained as, for the $k$th block where $k = 0, 1, \ldots, K - 1$,

$$\tilde{\mathbf{D}}(m - k) = \text{diag} \left\{ \tilde{s}_{2kL_M}(m) \tilde{s}_{2kL_M + 1}(m) \ldots \tilde{s}_{2kL_M + 2L_M - 1}(m) \right\}.$$

(3.31)

The MMax-MDF$_f$ algorithm is described by (3.21), (3.22), (3.31) and (3.25).

It has been shown in [146] that the convergence performance of MMax-NLMS degrades with a reducing normalized energy of the subselected tap-input vector by the MMax criterion. This energy is defined by $\mathcal{M}(n) = ||\mathbf{Q}(n)\mathbf{s}(n)||_2^2/||\mathbf{s}(n)||_2^2$ with $\mathbf{Q}(n)$ given in (3.8). In the same manner and for the case of $L_M = L_{nw}$, convergence performance of MMax-MDF$_f$ and MMax-MDF$_t$ can be studied by defining respectively, for the time- and frequency-domain tap-selection approaches,

$$\mathcal{M}_t(m) = ||\mathbf{Q}(m)\mathbf{s}(m)||_2^2/||\mathbf{s}(m)||_2^2,$$

(3.32)

$$\mathcal{M}_f(m) = \left\| \mathbf{F}_{2L_{nw}}^{-1} \mathbf{Q}(m)\mathbf{S}(m) \right\|_2^2/\left\| \mathbf{F}_{2L_{nw}}^{-1} \mathbf{S}(m) \right\|_2^2,$$

(3.33)

where elements of $\mathbf{Q}(m)$ in (3.32) and (3.33) are defined by (3.26) and (3.29) respectively.
Due to the orthogonality property of the Fourier transform, matrix $F^{-1}$ can be omitted from (3.33). Figure 3.4 shows the variation of $M_t(m)$ and $M_f(m)$ against $M_{\text{max}}$ for $K = 1$ and $L_{\text{nw}} = L = 128$. Since $M_t(m) > M_f(m)$ when $M_{\text{max}} < 2L_{\text{nw}}$, it is then expected that the degradation in performance due to tap selection is less for time-domain selection MMax-MDF$^t$ than frequency-domain MMax-MDF$^f$. This will be further illustrated by the simulation examples in Section 3.4.

Although selection in the time domain induces a less significant degradation in convergence performance than in the frequency domain, the computational cost for the latter is lower. This is because the diagonal elements in $\tilde{D}(m - k)$ given by (3.31) for MMax-MDF$^f$ consists of $2L_{\text{nw}} - M_{\text{max}}$ null elements across all $k$. However, due to matrix $F_{2L_{\text{nw}}}$, diagonal elements in $\tilde{D}(m - k)$ given by (3.27) for MMax-MDF$^t$ does not necessarily contain null elements. With the aim of reducing the complexity, frequency-domain tap selection, namely, the MMax-MDF$^f$ algorithm described by (3.21), (3.22), (3.31) and (3.25), is considered in this chapter. Accordingly, the subscript $[\cdot]^f$ can be safely dropped from now on for simplicity of presentation.

As described in Section 3.2, the MMax tap-selection given in (3.8) is achieved by sorting $s(n)$. As indicated in (3.24), however, elements in $\tilde{D}(m - k)$ are normalized by elements $p_i(m)$ of the vector $P(m)$ defined in (3.23) for the MDF implementation. Hence, taps corresponding to the $M_{\text{max}}$ maxima of the Fourier transformed tap-inputs normalized by $p_i(m)$ can be subselected alternatively so that elements of the resultant
2$L_{nw} \times 2$L_{nw} diagonal MMax tap-selection matrix denoted as $Q_N(m)$ are given by

$$q_{N,i}(m) = \begin{cases} 1 & i \in \{ \text{indices of the } M_{mmax} \text{ maxima of } s_i^*(m)s_i(m)/p_i(m) \}, \\ 0 & \text{otherwise}, \end{cases}$$ (3.34)

for $i = 0, 1, \ldots, 2L_{nw} - 1$ and $1 \leq M_{mmax} \leq 2L_{nw}$. The subscript $[\cdot]_N$ is introduced due to the normalization by $p_i(m)$ in (3.34) compared to (3.29), hence this algorithm is referred to as MMax-MDF$^N$. Define a $2L_{nw} \times 1$ vector $\tilde{S}_N(m)$ containing the subselected Fourier transformed tap-inputs as

$$\tilde{S}_N(m) = Q_N(m)\tilde{s}(m) = [\tilde{s}_{N,0}(m) \tilde{s}_{N,1}(m) \ldots \tilde{s}_{N,2L_{nw}-1}(m)]^T,$$ (3.35)

the $2L_M \times 2L_M$ diagonal matrix $\tilde{D}_N(m - k)$ for MMax-MDF$^N$ is then given by

$$\tilde{D}_N(m - k) = \text{diag}\{\tilde{s}_{N,2kL_M}(m) \tilde{s}_{N,2kL_M+1}(m) \ldots \tilde{s}_{N,2kL_M+2L_M-1}(m)\}$$ (3.36)

for $k = 0, 1, \ldots, K - 1$. It can thus be found that elements in the vector $\tilde{D}_N(m - k)$ are obtained from the $k$th block of the subselected Fourier transformed tap-inputs contained in $\tilde{S}_N(m)$ with indices from $2kL_M$ to $2kL_M + 2L_M - 1$. The adaptation of MMax-MDF$^N$ algorithm is described by (3.34) – (3.36) and (3.25), where $\tilde{D}(m - k)$ is replaced by $\tilde{D}_N(m - k)$.

As will be presented in Section 3.4, the degradation in convergence performance due to tap selection is less in MMax-MDF$^N$ than in MMax-MDF. However, it is noted that the MMax-MDF$^N$ algorithm requires $2L_{nw}$ additional divisions for tap selection due to the normalization by $p_i(m)$ in (3.34). Since reducing complexity is of main interest, MMax-MDF is chosen to be the basis of the algorithm to be developed in Section 3.3.2 which incorporates the SP tap selection to achieve a fast rate of convergence.

### 3.3.2 The SPMMAX-MDF Algorithm

The integration of SP tap selection into the frequency domain to form the SPMMAX-MDF algorithm is now presented. The SP tap selection defined by (3.9) was proposed to achieve fast convergence for identifying sparse impulse responses. However, it is noted that direct implementation of SP tap selection into the frequency-domain algorithm such as FLMS is inappropriate since the impulse response in the transformed domain is not
necessarily sparse. This can be verified by studying the effect of $K \geq 1$ on the concatenated impulse response of the MDF structure $h_{nw}$ defined as

$$h_{nw} = F_{2L_{nw}} \begin{bmatrix} h_{nw,0}^T \\ 0_{L_M \times 1} \\ h_{nw,1}^T \\ 0_{L_M \times 1} \\ \vdots \\ h_{nw,K-1}^T \\ 0_{L_M \times 1} \end{bmatrix}^T,$$

(3.37)

where $h_{nw,k}$ the $k$th subfilter to be identified, i.e.,

$$h_{nw,k} = [h_{nw,kL_M} \ h_{nw,kL_M+1} \ldots h_{nw,kL_M+L_M-1}]^T,$$

(3.38)

for $k = 0, 1, \ldots, K - 1$ and the $2L_{nw} \times 2L_{nw}$ matrix

$$F_{2L_{nw}} = \begin{bmatrix} F_{2L_M} \ldots 0 \\ \vdots & \ddots & \vdots \\ 0 \ldots F_{2L_M} \end{bmatrix}_{2L_{nw} \times 2L_{nw}}$$

(3.39)

is constructed by $K$ DFT matrices each of dimension $2L_M \times 2L_M$. As indicated in (3.38), the impulse response $h_{nw}$ is partitioned into smaller blocks in the time domain as $K$ increases. Figure 3.5 shows the magnitude variation of $|h_{nw}|$ for $K = 1$, $K = 16$ and $K = 64$ using the network impulse response shown in Fig. 3.1. As can be seen from the figure, $|h_{nw}|$ is not sparse for $K = 1$, in which case MDF is equivalent to FLMS.
3.3 The Sparse Partial Update Multidelay Filtering Algorithm

SP tap selection in the MDF structure will not improve the convergence performance for \( K = 1 \). For the cases where \( K > 1 \), the number of taps with small \(|h_{nw}|\) increases with \( K \), that is, the number of subfilters. Figure 3.6 further shows how the sparseness of \(|h_{nw}|\) varies with \( K \) using the sparseness measure \( \varsigma(|h_{nw}|) \) given by (3.4), from which it can be seen that \(|h_{nw}|\) becomes more sparse as \( K \) increases. Consequently, it is expected that SP tap selection would improve the convergence performance of MDF for sparse system identification.

Although integrating SP tap selection can be beneficial in the frequency domain, it requires careful consideration since as can be seen from (3.17), the length of the input frame \( s(m-k) \) is \( 2L_M \) compared to \( L_{nw} \) for the adaptive filter. This causes a length mismatch between \( s(m-k) \) and \( \hat{h}_{nw}(m) \) if the SP tap selection given by (3.9) is to be implemented in the frequency domain. Fortunately, this can be overcome by concatenating all \( K \) frequency-domain subfilters, \( \hat{h}_{nw,k}(m) \) for \( k = 0, 1, \ldots, K-1 \) to express \( \hat{h}_{nw}(m) \) of length \( 2L_{nw} \), that is,

\[
\hat{h}_{nw}(m) = \begin{bmatrix} \hat{h}_{nw,0}^T(m) & \hat{h}_{nw,1}^T(m) & \cdots & \hat{h}_{nw,K-1}^T(m) \end{bmatrix}^T = \begin{bmatrix} \hat{h}_{nw,0}(m) & \hat{h}_{nw,1}(m) & \cdots & \hat{h}_{nw,2L_{nw}-1}(m) \end{bmatrix}^T. \tag{3.40}
\]

SP tap selection can then be implemented by selecting \( 1 \leq M_{sp} \leq 2L_{nw} \) elements from \( \left| s_i(m) \hat{h}_{nw,i}(m) \right| \) for \( i = 0, 1, \ldots, 2L_{nw} - 1 \), where elements \( s_i(m) \) can be obtained from \( S(m) \) defined in (3.28). This is equivalent to subselecting \( M_{sp} \) elements of which the
magnitude responses of both \( s_i(m) \) and \( \hat{h}_{nw,i}(m) \) are significantly large. Elements of the \( 2L_{nw} \times 2L_{nw} \) diagonal tap-selection control matrix \( Q(m) \) are hence given by

\[
q_i(m) = \begin{cases} 
1 & i \in \{ \text{indices of the } M_{sp} \text{ maxima of } \left| s_i(m) \hat{h}_{nw,i}(m) \right| \}, \\
0 & \text{otherwise} 
\end{cases}
\] (3.41)

for \( i = 0, 1, \ldots, 2L_{nw} - 1 \). Employing (3.41), the diagonal matrix \( \tilde{D}(m - k) \) in (3.25) for the SP tap selection can be obtained by invoking (3.30) and (3.31).

Finally, it is worthwhile mentioning that additional simulations performed using tap-selection criterion using \( \left| s^*_i(m) s_i(m) \hat{h}_{nw,i}(m) / p_i(m) \right| \) showed no significant improvement in terms of convergence rate compared to (3.41). This is because that the sparseness of \( |\hat{h}_{nw,i}(m)| \) dominates the subselecting process compared to the term \( s^*_i(m) s_i(m) / p_i(m) \), which results in subselecting the same filter coefficients for adaptation as by using (3.41). In addition, normalization by \( p_i(m) \) incurs an extra \( 2L_{nw} \) divisions, hence is not preferable. Moreover, since the number of the “active” coefficients of \( h_{nw} \) reduces with increasing \( K \), \( M_{sp} \) is chosen to be inversely proportional to \( K \) such that

\[
M_{sp} = \frac{(2 - \gamma)L_{nw}}{K} + \gamma L_{nw} = (2 - \gamma)L_M + \gamma L_{nw},
\] (3.42)

where \( K, M_{sp} \in \mathbb{Z}^+ \) and \( 2/(1 - K) < \gamma \leq 2 \) since \( 0 < M_{sp} \leq 2L_{nw} \). This allows adaptation to be more concentrated on the “active” region. In this chapter, \( \gamma = 1 \) is used.

The SPMMax-MDF algorithm is summarized in Table 3.2.

### 3.3.3 Computational Complexity

Although \( L_M = L_{nw} \) is the optimal choice for the MDF algorithm from the computational complexity point of view, it nevertheless is more efficient than time-domain implementations even for \( L_M < L_{nw} \), not to mention the benefit of a reduced delay [7]. As shown in Table 3.2, the SPMMax-MDF computes \( \tilde{D}(m - k) \) using tap-selection control matrix \( Q(m) \) defined by (3.29) and (3.41) for \( \text{mod}(m, T) = 0 \) and \( \text{mod}(m, T) \neq 0 \) respectively. To demonstrate comparatively the complexity of the developed SPMMax-MDF algorithm, Table 3.3 shows the number of multiplications and divisions required for MDF, MMax-MDF, MMax-MDF\(_N\), SPMMax-MDF and the recently proposed IPMDF [143] to compute
3.3 The Sparse Partial Update Multidelay Filtering Algorithm

Table 3.2: The SPMMax-MDF Algorithm

| Parameters | \( \delta = 20\sigma_s^2L_M/L_{nw} \) |
| \( \lambda \) | \( = \left[ 1 - \frac{1}{3L_{nw}} \right] \) |
| \( \mu \) | \( = \beta(1 - \lambda), \quad 0 < \beta \leq 1 \) |
| \( S(0) \) | \( = \sigma_s^2/100 \) |
| \( \hat{h}_{nw,k}(m) \) | \( = \left[ \hat{h}_{nw,k1M}(m) \; \hat{h}_{nw,k1M+1}(m) \; \cdots \; \hat{h}_{nw,k1M+L_M-1}(m) \right]^T \) |
| \( \hat{h}_{nw,k}(m) \) | \( = \hat{h}_{nw,k}(m) \) |
| \( S(m) \) | \( = \left[ \tilde{s}_0(m) \; \tilde{s}_1(m) \; \cdots \; \tilde{s}_{2L_{nw}-1}(m) \right] \) |
| \( i \) | \( = 0, 1, \ldots, 2L_{nw} - 1 \) |

Algorithm

**MMax tap selection** for \( \text{mod}(m, T) = 0 \)

\[
q_i(m) = \begin{cases} 
1 & i \in \{ \text{indices of the } M_{m_{\text{max}}} \text{ maxima of } |\tilde{s}_i(m)| \} \\
0 & \text{otherwise}
\end{cases}
\]

**SP tap selection** for \( \text{mod}(m, T) \neq 0 \)

\[
M_{sp} = (2 - \gamma)L_M + \gamma L_{nw}
\]

\[
q_i(m) = \begin{cases} 
1 & i \in \{ \text{indices of the } M_{sp} \text{ maxima of } |\tilde{s}_i(m)|h_{nw,i}(m) \} \\
0 & \text{otherwise}
\end{cases}
\]

\[
\tilde{S}(m) = Q(m)\tilde{S}(m) = \left[ \tilde{s}_0(m) \; \tilde{s}_1(m) \; \cdots \; \tilde{s}_{2L_{nw}-1}(m) \right]^T
\]

\[
\tilde{D}(m-k) = \text{diag} \left\{ \tilde{s}_{2kL_M}(m) \; \tilde{s}_{2kL_M+1}(m) \; \cdots \; \tilde{s}_{2kL_M+2L_M-1}(m) \right\}
\]

\[
e(m) = \tilde{S}(m) - G^{01} \sum_{k=0}^{\infty} \tilde{D}(m-k)\hat{h}_{nw,k}(m-1)
\]

\[
S(m) = \lambda S(m-1) + (1 - \lambda)D^\dagger(m)D(m)
\]

\[
P(m) = S(m) + \delta I_{2L_M \times 2L_M} = \text{diag} \{ p_0(m) \; p_1(m) \; \cdots \; p_{2L_{nw}-1}(m) \}
\]

Filter update

\[
\hat{h}_{nw,k}(m) = \hat{h}_{nw,k}(m-1) + \mu \tilde{h}(m-k)P^{-1}(m)e(m)
\]
Table 3.3: Complexity of various NEC algorithms.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Multiplication</th>
<th>Division</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDF</td>
<td>$2L_{nw}$</td>
<td>$2L_{nw}$</td>
</tr>
<tr>
<td>IPMDF</td>
<td>$3L_{nw}$</td>
<td>$4L_{nw}$</td>
</tr>
<tr>
<td>MMax-MDF</td>
<td>$M_{mmax}$</td>
<td>$M_{mmax}$</td>
</tr>
<tr>
<td>MMax-MDF$_{N}$</td>
<td>$M_{mmax}$</td>
<td>$M_{mmax} + 2L_{nw}$</td>
</tr>
<tr>
<td>SPMMax-MDF</td>
<td>$[M_{mmax} + (T - 1)M_{sp}] / T$</td>
<td>$[M_{mmax} + (T - 1)M_{sp}] / T$</td>
</tr>
</tbody>
</table>

the term $\hat{D}^*(m - k)P^{-1}(m)g(m)$ in (3.25), which consumes the majority of the computational resources and differentiates these MDF-based algorithms. It is important to note that for MMax and SP tap selection in (3.29) and (3.41), no additional computational complexity is introduced since $|\hat{e}_i(m)|$ and $|\hat{e}_i(m)\hat{h}_{nw,i}(m)|$ can be obtained from (3.22) and (3.21) respectively. For MMax-MDF$_{N}$, however, computing the subselected filter coefficients for adaptation using (3.34) incurs $2L_{nw}$ additional divisions. The complexity for each algorithm for an example case of $L_{nw} = 512$, $T = 8$, $M_{mmax} = 0.5 \times 2L_{nw}$ and $K = 64$ is shown in Table 3.4 from which it can be seen that the complexity of the SPMMax-MDF is approximately 50% of that for the MDF. Compared to MMax-MDF, SPMMax-MDF requires only an additional 2% of multiplications and divisions. As will be shown in Section 3.4, however, the performance of SPMMax-MDF is better than MMax-MDF. In addition, the complexity of SPMMax-MDF is 33% and 25% of that for the IPMDF algorithm [143] in terms of multiplications and divisions respectively.

3.4 Simulation Results

In this section, simulation results are presented to study the performance of the SPMMax-MDF algorithm for NEC using a network impulse response $h_{nw}$ with 512 coefficients [134] as shown in Fig. 3.2. The convergence performance is measured using normalized misalignment $\eta(n)$ defined in (2.8) which is reproduced here for convenience

$$\eta(n) = 10 \log_{10} \left( \frac{\|h_{nw} - \hat{h}_{nw}(n)\|_2^2}{\|h_{nw}\|_2^2} \right) \text{dB.}$$

(3.43)
Table 3.4: Complexity of various NEC algorithms for the example case of $L_{nw} = 512$, $T = 8$, $M_{\text{mmax}} = 0.5 \times 2L_{nw}$, $\gamma = 1$, and $M_{\text{sp}} = 520$.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Multiplication</th>
<th>Division</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDF</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>IPMDF</td>
<td>1536</td>
<td>2048</td>
</tr>
<tr>
<td>MMax-MDF</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>MMax-MDF_N</td>
<td>512</td>
<td>1536</td>
</tr>
<tr>
<td>SPMMMax-MDF</td>
<td>519</td>
<td>519</td>
</tr>
</tbody>
</table>

The sampling frequency is 8 kHz and WGN is added to achieve a SNR of 20 dB. Unless otherwise specified, the following parameters for the algorithms are chosen for all simulations [143]: $T = 8$, $\lambda = [1 - 1/(3L_{nw})]^{1/2}$, $S(0) = \sigma_s^2 / 100$, $\delta = 20\sigma_s^2 L_M / L_{nw}$ and $K = 64$. Step-size control variable $\beta$ is adjusted for each algorithm so as to achieve the same steady-state performance.

3.4.1 Performance of MMax-MDF_t, MMax-MDF_f and MMax-MDF_N

Figure 3.7 illustrates how the convergence of MMax-MDF_t and MMax-MDF_f vary with $M_{\text{mmax}}$ using WGN input and a step-size control variable of $\beta = 0.6$. It can be seen that for each case of $M_{\text{mmax}}$, the degradation in convergence of MMax-MDF_t due to tap selection is lower than that for MMax-MDF_f. This is because, as explained in Section 3.3.1, $M_t(m) > M_f(m)$. Since both $M_t(m)$ and $M_f(m)$ reduces with $M_{\text{mmax}}$ as shown in Fig. 3.4, the rate of convergence reduces with $M_{\text{mmax}}$ as expected. When $M_{\text{mmax}} = 2L_M$ for MMax-MDF_t and $M_{\text{mmax}} = 2L_{nw}$ for MMax-MDF_f, both algorithms reduce to MDF since no tap-selection is performed and all taps are updated.

The variation in convergence of MMax-MDF_N and MMax-MDF against $M_{\text{mmax}}$ is then plotted in Fig. 3.8 where step-size control variables $\beta = 0.7$ and $\beta = 0.6$ are used for MMax-MDF_N and MMax-MDF respectively. As described in (3.34), (3.22) and (3.23), the normalization term $p_i(m)$ is determined by the variance of the input sequence $\sigma_s^2$. If the input sequence is not statistically white, $\sigma_s^2$ can vary on a frame-by-frame basis resulting a varying normalization term $p_i(m)$. A CGN input generated by filtering a sequence of
3.4 Simulation Results

Figure 3.7: Performance variation of MMMax-MDF$_t$ and MMMax-MDF$_f$ with $M_{m\text{max}}$ using WGN input.

Figure 3.8: Performance variation of MMMax-MDF$_N$ and MMMax-MDF with $M_{m\text{max}}$ using CGN input.

Figure 3.9: Performance variation of SPMMax-MDF with $M_{m\text{max}}$ using WGN input for $T = 8$.

zero-mean WGN through a low pass filter with a single pole $[20]$ is hence used so as to simulate the case where input sequence is not WGN. As can be seen, for each case of $M_{m\text{max}}$, the degradation in convergence performance due to tap selection is less for the MMMax-MDF$_N$ than the MMMax-MDF. However, as shown in Table 3.3 and Table 3.4, MMMax-MDF$_N$ incurs $2L_{nw}$ additional divisions compared to the MMMax-MDF algorithm. It can also be seen that the performance of MMMax-MDF$_N$ for $M_{m\text{max}} = 0.75 \times 2L_{nw}$ is not significantly better than the MDF algorithm.
3.4 Simulation Results

3.4.2 Comparative results for SPMMax-MDF

Performance variation of the SPMMax-MDF algorithm in comparison to other existing algorithms is now demonstrated. As explained in Section 3.3.2, the tap selection of the SPMMax-MDF algorithm is performed in the frequency domain for reducing computational complexity.

First of all, Fig. 3.9 compares the convergence performance of the SPMMax-MDF and that of the MDF algorithm for WGN input where \( T = 8, \beta = 1 \) and \( M_{sp} = L_{nw} \) is used for the former and \( \beta = 0.6 \) for the latter such that the same steady-state performance can be achieved. The convergence of the substantially more complex IPMDF algorithm [143] is also included for the purpose of comparison. It can be seen that the SPMMax-MDF algorithm achieves higher rate of convergence by approximately 6 dB in terms of \( \eta(n) \) compared to the more complex MDF during adaptation. According to Table 3.3, SPMMax-MDF only requires approximately 29.7% and 22.3% of the number of multiplications and divisions compared to IPMDF for the case of \( M_{mmax} = 0.0625 \times 2L_{nw} \) and \( M_{sp} = L_{nw} \).

The convergence performance of SPMMax-MDF is then compared against MDF and IPMDF using CGN input specifically for \( K = 1 \) in Fig. 3.10, where \( T = 8 \) and \( \beta = 0.6 \) is used for SPMMax-MDF. \( M_{mmax} \) is chosen such that \( M_{mmax} = 0.5 \times 2L_{nw} \) since it is shown in [146] that by such setting, a good balance between complexity reduction
and performance degradation due to MMax tap selection can be reached. As can be seen from the figure, the performance of SPMMax-MDF is close to that of the MDF since $\mathcal{K} = 1$ results in $M_{sp} = 2L_{nw}$ for any $\gamma$ according to (3.42). This is to say, under the condition of $\text{mod}(m, T) \neq 0$, all $2L_{nw}$ filter coefficients are updated while under the condition of $\text{mod}(m, T) = 0$, only $M_{mmax} = 0.5 \times 2L_{nw}$ coefficients are updated. As a result of this, the performance of SPMMax-MDF approaches that of the MDF. Compared to IPMDF, SPMMax-MDF only requires approximately 63% and 47% of the number of multiplications and divisions according to Table 3.3.

Figure 3.12 further compares the convergence performance of SPMMax-MDF, MDF and IPMDF for $\mathcal{K} > 1$ using CGN input. As before, same step-size control variable of $\beta = 0.6$ is used for all algorithms except for the cases of SPMMax-MDF where $\beta = 0.8$ is used to achieve the same steady-state performance. It can be seen that for $\mathcal{K} = 64$, the SPMMax-MDF algorithm achieves higher rate of convergence in terms of $\eta(n)$ compared to the more complex MDF. Since, as shown in Fig 3.6, $\zeta(|h_{nw}|)$ increases with $\mathcal{K}$, it can be expected that such improvement will be greater when larger $\mathcal{K}$ is chosen. In addition, as the delay for MDF is reduced by a factor of $\mathcal{K}$ compared to FLMS, SPMMax-MDF can achieve further delay reduction for larger $\mathcal{K}$ and thus is desirable for NEC. For the case of $M_{mmax} = 0.5 \times 2L_{nw}$ and $\mathcal{K} = 64$, the number of multiplications and divisions required for each algorithm has been shown in Table 3.4.

Finally, Fig. 3.12 shows the convergence performance of various algorithms ob-
Figure 3.13: Development of the SPMMax-MDF algorithm.

Obtained using a male speech input which is obtained from the APLAWD database and down-sampled to 8 kHz. Parameters used for each algorithm are the same as in previous simulations except for SPMMax-MDF where $\beta = 1$ is used. The computational complexity required for each algorithm is also shown in the figure between square brackets where the first and the second integer represents the number of multiplications and divisions respectively. It can be seen that SPMMax-MDF achieves approximately 5 dB improvement in terms of $\eta(n)$ with lower complexity in comparison to MDF. In addition, the performance of the low-cost SPMMax-MDF algorithm approaches that of the substantially more complex IPMDF.

3.5 Summary

In this chapter, a supervised single-channel adaptive sparse system identification algorithm for NEC has been developed. This algorithm achieves a fast rate of convergence, low complexity and low delay by novelly exploiting both the MMax and SP tap selection in the frequency domain using MDF implementation. Two approaches for incorporating MMax tap selection into MDF were discussed and the resultant tradeoff between rate of convergence and complexity was subsequently discussed. The integration of SP tap selection into the MDF structure was also presented. Simulation results using both Gaussian noise and speech inputs have shown that the developed SPMMax-MDF algorithm achieves up to 5 dB improvement in convergence performance with significantly
reduced complexity compared to the MDF algorithm. In addition, the performance of the SPMMax-MDF algorithm approaches that of the IPMDF algorithm. Figure 3.13 summarizes how the SPMMax-MDF algorithm has been developed.
Chapter 4

The Common Zeros Problem in Blind Identification of SIMO Systems

IN the development of system identification algorithms with application to speech enhancement, the research reported in this thesis now moves from the single-channel supervised case discussed in Chapter 3 to multichannel unsupervised (blind) scenarios.

Blind system identification (BSI) and equalization algorithms have been applied to multichannel acoustic signal processing for speech dereverberation. As described in Section 2.4.2, classic SOS-based BSI algorithms assume that the channel identifiability conditions are satisfied. One such condition requires that multiple channel transfer functions must be coprime and not share common zeros. Consequently, only zeros of the source signal \( s(n) \) are assumed by these BSI algorithms to be common among the observed microphone signals as indicated by (2.19). That is to say, that these BSI algorithms cannot distinguish the common zeros, if any exist, from zeros of the source signal \( s(n) \), and hence would fail to identify the channels correctly. It is important to note that the coprimeness condition is also a prerequisite for equalization of SIMO systems using the Bezout theorem [123]. As a result, the presence of common zeros has been known to invalidate most existing BSI algorithms and limit the performance of channel equalization algorithms for the subsequent speech dereverberation [43]. So far, this fundamental problem has not been fully addressed and remains unsolved.

The main contribution of this chapter is to address and analyze the common zeros
4.1 Introduction to the Common Zeros Problem

The research on channel identifiability conditions was motivated by the development of SOS-based BSI algorithms in the context of wireless communications \cite{17}. Since the channel identifiability condition (C2.2) regarding the input signal, as described in Section 2.4.2, is relatively easier to satisfy, classic work was focused on the comprehensive proof of the identifiability of various communication systems with respect to the condition (C2.1) \cite{90,91,107,108,109,110,111}. The common zeros problem is so called as the presence of common zeros violates this condition and invalidates most classic SOS-based BSI algorithms.

problem in the context of BSI, where the existence of a new category of zeros known as near-common zeros (NCZs) is identified. Then, two efficient clustering algorithms are developed to quantify the number of common zeros in SIMO systems with a large number of FIR coefficients. These algorithms allow the effect of common zeros on BSI and channel equalization algorithms to be shown, hence facilitating the potential for the development of robust speech dereverberation algorithms. The motivation of these contributions is to provide a general framework for the study of the effect of common zeros in the context of acoustic signal processing.

This chapter begins with an introduction to the common zeros problem which includes a mathematical definition of the common zeros and a brief survey of the current literature regarding this problem. In Section 4.2, NCZs are introduced, thereby extending the definition of the common zeros problem. General conditions for the existence of NCZs are also discussed where conventional common zeros can be considered a special case. It is then demonstrated that for multichannel acoustic systems with high order, common zeros are likely to occur. In Section 4.3, two clustering algorithms are developed in order to quantify the number of common zeros, which are based on the use of dissimilarity matrices containing the Euclidean distances between zeros from different channels. Simulation examples are presented in Section 4.4 to demonstrate the effectiveness of the developed algorithms in terms of accuracy and complexity. Using these clustering algorithms, the effects of common zeros on the performance of BSI and channel equalization algorithms are also presented. Section 4.5 summarizes this chapter.
Recall the impulse response for the $m$th channel of a SIMO system $h_m = [h_{m,0}, h_{m,1}, \ldots, h_{m,L-1}]^T$ defined by (2.18) with its transfer function expressed by

$$H_m(z) = \sum_{n=0}^{L-1} h_{m,n} z^{-n} = K_m \prod_{p=1}^{L-1} \left(1 - z_m^{-1}(p)\right), \quad (4.1)$$

where $z = e^{2\pi f}$, $f$ is the normalized frequency and $K_m \in \mathbb{Z}^+$. Denote the $p$th zero by $z_m(p) = x_m(p) + jy_m(p)$, its location in the $z$-plane can be determined by $x_m(p)$ and $y_m(p)$ along the real and imaginary axis respectively such that

$$H_m(z)|_{z=z_m(p)} = \sum_{n=0}^{L-1} h_{m,n} z^{-n}(p) = 0, \quad 1 \leq p \leq L - 1. \quad (4.2)$$

For an $M$-channel FIR system, zeros of multiple transfer functions can be categorized, in terms of the closeness (common-ness) between each other in the $z$-plane, as

- “common zeros” if they are found in each and every channel’s transfer function;
- “non-common zeros” if they are found in only one out of $M$ channels.

The $p$th common zero in the system can therefore be defined as $z_C(p)$ such that

$$H_C(z)|_{z=z_C(p)} = 0, \quad p = 1, 2, \ldots, N_c, \quad (4.3)$$

where $H_C(z)$ is the greatest common divisor of the $M$ transfer functions given by

$$H_C(z) = \gcd\{H_1(z), H_2(z), \ldots, H_M(z)\}, \quad (4.4)$$

and $N_c$ denotes the total number of common zeros in the system. From (4.3) and (4.4), it is not difficult to find that $N_c = 0$ if $H_C(z) = 1$; and $N_c > 0$ if $H_C(z) \neq 1$. It is also worthwhile noting that each index $p$ actually corresponds to a cluster of $M$ zeros, one from each channel, located at the identical position in the $z$-plane as indicated by (4.4). As a result, these zeros are also known as the “exactly-common zeros”.

In communication systems, common zeros can occur if multiple channels are with specific delays or frequency nulls \[107\][108]. However, little attention has been received in the literature on studying the adverse effect brought about by these common zeros. This is largely because the FIR models for communication channels are normally short
as described in Section 2.4 where impulse responses with less than 10 coefficients are normal. In addition, the flexility of communication systems allows various ways to avoid common zeros. For example, the use of fractional sampling and specifically designed receiver arrays can introduce spatial diversity so that common zeros are less likely to occur [110]. Other methods such as filter bank pre-coders [147] have also been developed to induce extra cyclostationarity at the transmitters for guaranteed blind identifiability. As a result, the assumption of having no common zeros is not difficult to justify for most communications systems.

Recently, the increasing popularity of adopting BSI algorithms over SIMO acoustic systems for applications such as speech source localization and dereverberation has made the research on the common zeros problem once again becoming active since the assumption of having no common zeros is no longer valid. This is because on one hand, acoustic systems involving, for example, human speakers and microphone arrays, cannot be purposefully designed as opposed to antenna arrays in wireless communications. On the other hand, arrays with a large number of microphones are unrealistic and computationally expensive for most of the applications involving portable hands-free devices due to constraints on size and power consumption. Moreover, acoustic impulse responses normally contain hundreds or thousands of FIR coefficients as described in Section 2.3. It will be demonstrated in Section 4.2 that such high-order multichannel systems can be exposed to not only exactly-common zeros, but also a new category of zeros known as the NCZs, which are not comparatively less important in terms of degrading BSI and channel equalization performance. Therefore, it is desirable to develop generalized algorithms capable of quantifying the number of both exactly- and near-common zeros in order to study their effects.

4.2 The Near-Common Zeros (NCZs)

In this section, near-common zeros (NCZs) are introduced. Historically, their presence has rarely been addressed explicitly, although it was mentioned in [148] that the performance of the CR-based BSI methods can be as good as the maximum likelihood method [95] unless zeros of different channels become close to each other. Before formally defining NCZs, an illustrative example can be used to demonstrate their presence.
and effects on the BSI performance. Consider an example two-channel SIMO system generated by placing its zeros on the $z$-plane such that the distance between the zeros of the two channels can be expressed by the zero separation

$$\Delta z = 2 \cos \theta, \quad 0 \leq \theta \leq \pi/2$$

as shown in Fig. 4.1, where symbols □ and ○ represent zeros from channel 1 and channel 2 respectively. As can be seen, the zero separation $\Delta z$ decreases as $\theta$ increases and these zeros become exactly common for $\theta = \pi/2$ and $\Delta z = 0$. Now, employing the MCLMS algorithm [18] reviewed in Section 2.4.4 on this system excited by a sequence of WGN with a SNR of 60 dB, the convergence performance of MCLMS against a varying $\Delta z$ would indicate the effect of NCZs. As described in Section 2.4.4 and according to Fig. 2.10, the best channel estimates for the MCLMS algorithm can be obtained while its cost function $J_{\text{MCLMS}}(n)$ defined by (2.43) is minimized resulting a minimal steady-state value of NPM $\eta'(n)$ defined by (2.27). Figure 4.2 shows the number of iterations required for $J_{\text{MCLMS}}(n)$ to reach $-60$ dB against $\Delta z$, where the experiment is run for 10 independent trials using the same step-size of 0.5 and the results are obtained by averaging on these trails. As can be seen, the convergence rate of MCLMS in terms of number of iterations increases exponentially as $\Delta z$ reduces. When $1 \leq \Delta z \leq 2$, such increment is negligible. When $\Delta z < 1$, however, it can be observed that the reduction of convergence speed is significant, and it can be expected that MCLMS will not converge if $\Delta z = 0$. This experiment indicates that zeros do not have to be exactly-common to affect the performance of BSI algorithm, even though they do not violate the channel identifiability condition (C.22). The terminology of NCZs is therefore introduced to denote those zeros which are close enough to each other that BSI performance can be degraded. For the illustrative example system shown in Fig. 4.1, the two pairs of zeros can be considered NCZs when $\Delta z < 1$ due to their effect on the convergence of the MCLMS algorithm as shown in Fig. 4.2. This also indicates that the zero separation $\Delta z$ for this example is closely related to the existence of NCZs.

In general, the closeness (common-ness) of an corresponding pair of zeros from two channels in the $z$-plane can be expressed by the pairwise Euclidean distance. With reference to (4.1), such distance for the $p$th and $q$th zero of channel $m$ and $n$ for $m \neq n$
can be expressed by
\[
D_{\{m,n\}}(p,q) = |z_m(p) - z_n(q)| = \sqrt{[x_m(p) - x_n(q)]^2 + [y_m(p) - y_n(q)]^2}.
\]  
(4.6)

Hence, these two channels become coprime with each other if \( z_m(p) = z_n(q) \), i.e., \( D_{\{m,n\}}(p,q) = 0 \). Utilizing this measure in terms of the pairwise Euclidean distance, a cluster of NCZs in an \( M \)-channel system can now be defined as a group of \( M \) zeros that satisfy the following three conditions:

\[(\text{C4.1})\] The number of zeros within each cluster must equate the number of channels \( M \) with each channel contributing only one zero;

\[(\text{C4.2})\] Any pairs of zeros in a cluster must lie within a vicinity \( \delta_c \) in terms of their Euclidean distances where \( \delta_c \geq 0 \) is defined as the tolerance;

\[(\text{C4.3})\] Any zero can be a member of more than one cluster.

The first condition results from the definition of zeros being near-common across all channels. Condition \((\text{C4.2})\) defines the common-ness between pairs of zeros inside a cluster such that exactly-common zeros become a special case of NCZs when \( \delta_c = 0 \). As will be described in Chapter 8, metrics other than the pairwise Euclidean distance are worthwhile investigating to measure the closeness of the zeros within a cluster of NCZ. An interesting example is the centroid point of the cluster.
4.2 The Near-Common Zeros (NCZs)

The Near-Common Zeros (NCZs) $\delta$ are defined such that $c \delta \leq c \delta \leq c \delta \leq c \delta \leq c \delta \leq (0,0)$.

Figure 4.3: An example of two clusters of NCZs in a three-channel system in the $z$-plane.

To illustrate mathematically how clusters of NCZs form under these conditions, consider an example of a three-channel system of length $L$, where three vectors $z_1 = [z_1(1) \ z_1(2) \ \ldots \ z_1(L-1)]^T$, $z_2 = [z_2(1) \ z_2(2) \ \ldots \ z_2(L-1)]^T$, and $z_3 = [z_3(1) \ z_3(2) \ \ldots \ z_3(L-1)]^T$ are defined to contain $L-1$ zeros for each channel respectively. The pairwise distance of the zeros can be described by a $(L-1) \times (L-1)$ dissimilarity matrix $D_{\{m,n\}}$ defined between channels $m$ and $n$ with its $p$th row and $q$th column element being given by $D_{\{m,n\}}(p,q)$ as defined in (4.6). For this example, clusters containing zeros from every possible pair of channels can be obtained as $C_{\{1,2\}} = \arg D_{\{1,2\}}(p,q) \leq \delta_c$, $C_{\{1,3\}} = \arg D_{\{1,3\}}(p,r) \leq \delta_c$, $C_{\{2,3\}} = \arg D_{\{2,3\}}(q,r) \leq \delta_c$ for $p,q,r = 1, 2, \ldots, L-1$. Denoting $\emptyset$ as the empty set and if $C_{\{1,2\}} \neq C_{\{1,3\}} \neq C_{\{2,3\}} \neq \emptyset$, clusters of NCZs for this three-channel system can be determined by elements in

$$C_{\{1,3\}} = C_{\{1,2\}} \cup C_{\{1,3\}} \cup C_{\{2,3\}}.$$  \hspace{1cm} (4.7)

If $C_{\{m,n\}} = \emptyset$, for any $m \neq n$, no clusters can be found in this three-channel system given $\delta_c$. The computation of $C_{\{1,2\}}, C_{\{1,3\}}$ and $C_{\{2,3\}}$ hence ensures that each cluster satisfies condition (C4.2) while (4.7) ensures that condition (C4.1) is also satisfied. Figure 4.3 shows two example NCZ clusters in the $z$-plane for this three-channel system, where Symbols $\triangle$, $\square$ and $\circ$ represent zeros from each channel and they lie within pairwise $\delta_c$ vicinity from each other. The zero shared by both clusters reflects the condition (C4.3). If $\delta_c = 0$, these zeros would super-impose each other in the $z$-plane.
4.2 The Near-Common Zeros (NCZs)

Figure 4.4: Examples of the zeros of various channels showing (a) that they cluster around the unit circle and (b) with uniformly distributed phases.

As mentioned in Section 2.4, the reverberation time for a typical room can be several hundred milliseconds long, resulting in acoustic impulse responses with a large number of FIR coefficients. Results in [149] showed that for a random polynomial with real-valued coefficients which are drawn from a standard Gaussian distribution, its zeros tend to cluster around the unit circle with uniformly distributed phases as the order of polynomial increases. This indicates that the density of zeros in the vicinity of the unit circle in z-plane can be high for such multichannel systems with high order. This is true also for SIMO acoustic systems with high order as will be demonstrated next.

Define the magnitude and the phase of \( z_m(p) \) by \( \rho_m(p) = |z_m(p)| = \sqrt{x_m^2(p) + y_m^2(p)} \) and \( \theta_m(p) = \arctan(y_m(p)/x_m(p)) \) for \(-\pi \leq \theta_m(p) \leq \pi\) respectively, the conclusion in [149] can be expressed as, for \( p = 1, 2, \ldots, L - 1 \),

\[
\lim_{L \to \infty} E\{\rho_m(p)\} = 1, \quad (4.8)
\]
\[
\lim_{L \to \infty} P(\theta_m) = \frac{1}{2\pi}, \quad (4.9)
\]

where \( \theta_m = [\theta_m(1) \ \theta_m(2) \ \ldots \ \theta_m(L - 1)]^T \) and \( P(\cdot) \) denotes the discrete probability density function. Employing the recently developed FFT-based fast factorization algorithm [150], Fig. 4.4(a) shows the magnitude deviation between the zeros and the unit circle in decibel against an increasing channel length \( L \) for two recorded acoustic impulse responses obtained from the MARDY database [130] and a random impulse response.
with standard-Gaussian distributed coefficients. The corresponding phase variances of the zeros $\sigma_\theta = \text{var}\{\theta_m\}$ for the same set of channels is shown in Fig. 4.4(b). As can be seen, for impulse responses with length $L > 200$ the zeros indeed tend to cluster around the unit circle and that their phases asymptotically approximate a uniform distribution for which the variance can be computed theoretically as

$$\lim_{L \to \infty} \sigma_\theta = \frac{(\pi + \pi)^2}{12} \approx 3.289.$$  \hspace{1cm} (4.10)

This experiment clearly indicates that with such high density of zeros around the unit circle, NCZs are likely to occur in high-order SIMO systems such as acoustic systems. Simulation results in Section 4.4 will further demonstrate this by using the clustering algorithms to be developed in Section 4.3.

To summarize, the definition of NCZs generalizes the existence of both exactly- and near-common zeros through the pairwise tolerance $\delta_c$. It is therefore logical to extend the conventional common zeros problem to include the existence of NCZs such that the effect of both types of zeros can be considered. As a result, exactly- and near-common zeros are now considered collectively and the terminology “common zeros” will be used throughout the rest of this thesis to denote both of them for simplicity of presentation, unless otherwise explicitly emphasized.

### 4.3 Extraction of Common Zeros in Multichannel Systems

The literature of research on quantifying common zeros is very limited in both communications and acoustic signal processing communities. This is largely due to the computational intensity of factorizing polynomials resulted from high-order FIR impulse responses. Consequently, the idea of detecting and estimating common zeros without factorization has been of interest for many years in mathematics and control system theory [151][152][153][154]. However, these methods are only able to either detect the coprimeness of two polynomials, or estimate exactly-common zeros from more than two channel transfer functions with limited accuracy.

In this section, two efficient clustering algorithms are developed for quantifying the number of common zeros from high-order multichannel systems using the factorized
channel transfer functions. Although the true system transfer functions in the context of BSI are unknown and factorization is computationally expensive for high-order systems, the motivation of these algorithms is to provide a generalized utility to study how common zeros affect the performance of BSI and channel equalization algorithms. However, such clustering is not straightforward as the conditions described in Section 4.2 for NCZs’ presence imply that classic clustering algorithms such as the $k$- and $c$-means algorithms \[155\] are not employable. On one hand, the $k$-means algorithm requires \textit{a priori} knowledge of the number of clusters and it assumes that each zero belongs to only one cluster. On the other hand, the $c$-means algorithm violates condition (C4.1) since it does not appropriately constrain the number of zeros within a cluster. It is also worthwhile noting that the objective of common zeros extraction from multichannel system is \textit{not} an optimization problem.

### 4.3.1 Euclidean Distances between Zeros for Two Channels

Extraction of NCZ clusters in a multichannel system involves the computation of the Euclidean distances between any pair of zeros from different channels. An efficient approach to compute these quantities is now presented. As will be presented in Section 4.3.2, such approach can be utilized to develop generalized multichannel clustering algorithms.

Consider the $m$th and $n$th channel of a SIMO system of length $L$. To quantify pairwise distances between zeros of these two channels, two \((L-1) \times 1\) vectors

\[
\mathbf{z}_m = \begin{bmatrix} z_m(1) & z_m(2) & \ldots & z_m(L-1) \end{bmatrix}^T, \quad (4.11)
\]

\[
\mathbf{z}_n = \begin{bmatrix} z_n(1) & z_n(2) & \ldots & z_n(L-1) \end{bmatrix}^T
\] \quad (4.12)

containing $L-1$ zeros can be defined respectively. Computation of \[4.6\] for all the zeros in $\mathbf{z}_m$ and $\mathbf{z}_n$ can be efficient using

\[
D_{\{m,n\}} = \left| \tilde{\mathbf{Z}}_m - 2\mathbf{z}_m\mathbf{z}_n^T + \tilde{\mathbf{Z}}_n \right|^{\frac{1}{2}}
\] \quad (4.13)

in order to avoid the time-consuming “for” loops, where $\tilde{\mathbf{Z}}_m = \mathbf{Z}_m \odot \mathbf{Z}_m \odot \mathbf{1}_{(L-1) \times 1} =$
\[ \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}^T, \]

\[
Z_m = z_m 1^T = \begin{bmatrix} z_m(1) & \ldots & z_m(1) \\ \vdots & \ddots & \vdots \\ z_m(L-1) & \ldots & z_m(L-1) \end{bmatrix}, \quad Z_n = 1 z_n^T = \begin{bmatrix} z_n(1) & \ldots & z_n(L-1) \\ \vdots & \ddots & \vdots \\ z_n(1) & \ldots & z_n(L-1) \end{bmatrix},
\]

\[
2 z_m z_n^T = 2 \begin{bmatrix} z_m(1)z_n(1) & \ldots & z_m(1)z_n(L-1) \\ \vdots & \ddots & \vdots \\ z_m(L-1)z_n(1) & \ldots & z_m(L-1)z_n(L-1) \end{bmatrix}
\]

with \(\odot, [\cdot]^\frac{1}{2}\) and \(|\cdot|\) denoting respectively the Hadamard product, square root and elemental absolute operators. Equation (4.13) can thus be rewritten

\[
Z_m - 2 z_m z_n^T + Z_n = \begin{bmatrix} (z_m(1) - z_n(1))^2 & \ldots & (z_m(1) - z_n(L-1))^2 \\ \vdots & \ddots & \vdots \\ (z_m(L-1) - z_n(1))^2 & \ldots & (z_m(L-1) - z_n(L-1))^2 \end{bmatrix}. (4.14)
\]

Let \(a = x_m(p) - x_n(q)\) and \(b = y_m(p) - y_n(q)\), and invoke Euler’s identity \(a + jb = \rho e^{i\theta}\) \(\rho = \sqrt{a^2 + b^2}\) and \(\theta = \arctan(b/a)\), it can be derived that \(|(a + jb)| = |\rho^2 e^{i2\theta}| = \rho^2\), from which an important result can be obtained

\[
D_{\{m,n\}} = \left[ Z_m - 2 z_m z_n^T + Z_n \right]^{\frac{1}{2}} \quad (4.15)
\]

\[
= \begin{bmatrix} \sqrt{|(z_m(1) - z_n(1))^2|} & \ldots & \sqrt{|(z_m(1) - z_n(L-1))^2|} \\ \vdots & \ddots & \vdots \\ \sqrt{|(z_m(L-1) - z_n(1))^2|} & \ldots & \sqrt{|(z_m(L-1) - z_n(L-1))^2|} \end{bmatrix},
\]

hence verifying the validity of (4.13). Since all pairwise distances are computed only once for \(m \neq n\), no computational redundancy occurs. It is also noted that unless \(z_m - z_n = 0_{(L-1) \times 1}\), the diagonal elements of \(D_{\{m,n\}}\) are non-zero and \(D_{\{m,n\}}\) is not symmetrical, i.e., \(D_{\{m,n\}}(p,q) \neq D_{\{m,n\}}(q,p)\). Equation (4.13) can now be used for generalized multichannel clustering.
4.3 Extraction of Common Zeros in Multichannel Systems

4.3.2 Generalized Multichannel Clustering

Two generalized algorithms for extracting common zeros from multichannel systems are developed for $\delta_c \geq 0$. Similar to the illustrative three-channel example system in Section 4.2, the dissimilarity matrix $D_{\{m,n\}}$ can be computed for $M$-channel systems from which elements in $\{z_1, z_2, \ldots, z_M\}$ satisfy conditions as described in Section 4.2.

Employing $D_{\{m,n\}}$ between any two out of $M$ channels, consider the case where there are $N_{c,mn}$ clusters of common zeros between channels $m$ and $n$. A $N_{c,mn} \times 2$ subcluster group matrix $C_{\{m,n\}}$ containing these $N_{c,mn}$ clusters can be constructed by searching within elements in $D_{\{m,n\}}$ for indices $p$ and $q$ such that

$$C_{\{m,n\}} = \text{arg} \min_{D_{\{m,n\}}(p,q)} \leq \delta_c \quad (4.16)$$

is satisfied. Next, defining the $N_c \times M$ cluster matrix $C_{\{1:M\}}$ for the whole $M$-channel system where $N_c$ denotes the total number of clusters, it can be found that each row of $C_{\{1:M\}}$ contains a cluster with $M$ elements each with indices corresponding to elements in $\{z_1, z_2, \ldots, z_M\}$ for the respective channels 1 to $M$. Equation (4.16) can then be employed on each pair of channels selected from $M$. The aim of the generalized multichannel clustering (GMC) algorithm is therefore to obtain $C_{\{1:M\}}$ using subcluster groups $C_{\{m,n\}}$, which can be achieved using two approaches as will be described next.

The Divide-and-Conquer Algorithm

The GMC divide-and-conquer (GMC-DC) algorithm compares zeros of two channels stage-by-stage in a binary tree manner as depicted in Fig. 4.5. In the first stage, subcluster groups $C_{\{1,2\}}, C_{\{3,4\}}, \ldots$ are computed using (4.13) and (4.16). The second stage then extracts subcluster groups $C_{\{1:4\}}, C_{\{5:8\}}, \ldots$ from stage 1 and such process is repeated until comparisons between all branches of the tree are completed. For example, a cluster of 4 zeros in $C_{\{1:4\}}$ is obtained by invoking (4.13) and (4.16) for two subclusters of zeros, one from each of $C_{\{1,2\}}$ and $C_{\{3,4\}}$. This is to say, that both zeros in the subcluster from $C_{\{1,2\}}$ must satisfy condition (4.2) with respect to each of the two zeros in the subcluster from $C_{\{3,4\}}$, and vice versa. This indicates that (4.16) is computed 4 times between each subcluster in $C_{\{1,2\}}$ and $C_{\{3,4\}}$. During each stage, any pairwise distance greater
than tolerance $\delta_c$ is excluded. Consequently, the number of stages required for this GMC-DC algorithms equates $\lceil \log_2(M) + 1 \rceil$ while the computation of $D_{\{m,n\}}$ using (4.13) is required a total of $M - 1$ times.

**The Search-and-Trim Algorithm**

The GMC search-and-trim (GMC-ST) algorithm first computes $D_{\{m,n\}}$ for all possible pairs of channels with $m = 1, 2, \ldots, M - 1$ and $n = m + 1$ by employing (4.13) a total of $M(M - 1)/2$ times. Invoking (4.16), this algorithm then aims to extract $C_{\{1:M\}}$ from the resultant $M(M - 1)/2$ subcluster group matrices $C_{\{m,n\}}$ using an efficient search technique. This is achieved by first finding the subcluster group

$$C_{\{m_s,n_s\}} = \min_{N_{c,ms}} \{ C_{\{1,2\}} \ C_{\{1,3\}} \ C_{\{M-1,M\}} \}$$

with the smallest $N_{c,ms}$ as a reference group, where the two channels $m_s$ and $n_s$ containing the smallest number of subclusters determine the lower bound of $N_c$ according to condition (C4.1). For each row of $C_{\{m_s,n_s\}}$, GMC-ST initializes a matrix $C$ containing a single row vector $c = [c(1) \ c(2) \ \ldots \ c(M)]$ with only two non-empty elements $c(m_s) = p$ and $c(n_s) = q$ where $p$ and $q$ are two elements obtained from corresponding row in $C_{\{m_s,n_s\}}$. The next stage is to search, for the row vector $c$ of $C$, the remaining $M - 2$ empty elements. This search space is confined within $\{ C_{\{1,2\}} \ C_{\{1,3\}} \ \ldots \ C_{\{M-1,M\}} \}$ excluding $C_{\{m_s,n_s\}}$ since, from (4.16), only zeros within these groups are all within tolerance $\delta_c$. If $k$ elements are found (meaning that the zeros belong to $k$ different clusters), $C$ is updated.
4.4 Simulation Results

Table 4.1: The GMC-ST algorithm

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Compute $C_{{m,n}}$ and obtain $C_{{m_i,n_i}}$ using (4.17)</td>
</tr>
<tr>
<td>20</td>
<td>for each row of $C_{{m_i,n_i}}$, initialize $c = 0_{1 \times M}$ to form $C$</td>
</tr>
<tr>
<td>30</td>
<td>set $c(m_i) = p, c(n_i) = q$</td>
</tr>
<tr>
<td>40</td>
<td>while $C$ contains no empty elements</td>
</tr>
<tr>
<td>50</td>
<td>for each row $c$ in $C$</td>
</tr>
<tr>
<td>60</td>
<td>find all elements of $c$ within $C_{{m,n}}$</td>
</tr>
<tr>
<td>70</td>
<td>if $k$ values found, invoke (4.18)</td>
</tr>
<tr>
<td>80</td>
<td>else delete row</td>
</tr>
<tr>
<td>90</td>
<td>if $C$ has $&lt; 1$ row GOTO 20</td>
</tr>
<tr>
<td>100</td>
<td>else</td>
</tr>
<tr>
<td>110</td>
<td>for each row of $C$</td>
</tr>
<tr>
<td>120</td>
<td>Check if all pairwise elements $\in C_{{m,n}}$</td>
</tr>
<tr>
<td>130</td>
<td>if yes GOTO 110, else delete row</td>
</tr>
<tr>
<td>140</td>
<td>end for</td>
</tr>
<tr>
<td>150</td>
<td>end for</td>
</tr>
<tr>
<td>160</td>
<td>end while</td>
</tr>
<tr>
<td>170</td>
<td>$C \in C_{{1:M}}$</td>
</tr>
<tr>
<td>180</td>
<td>end for</td>
</tr>
</tbody>
</table>

as

$$\tilde{C} = 1_{k \times 1}c, \quad C = [\tilde{C}^T \ C^T]^T.$$ (4.18)

The trimming process then ensures that all pairwise elements for each row of $C$ can be found within the search space $\{C_{\{1,2\}} \ C_{\{1,3\}} \ \ldots \ C_{\{M-1,M\}}\}$ in order to satisfy condition (C4.2). If this condition is violated, the entire row is deleted and the search-and-trim process is repeated until every element in each row of $C$ is found or all rows have been deleted. Table 4.1 summarizes the GMC-ST algorithm.

4.4 Simulation Results

In this section, performance variation of the developed GMC algorithms over simulated and recorded SIMO acoustic systems is demonstrated. First, simulated impulse
4.4 Simulation Results

responses are generated using the method of images [66] in a room of dimension \(10 \times 10 \times 3\) m, where \(T_{60} = 200\) ms and an array of microphones with 5 cm spacings is used. The source is positioned at \(\{8.4, 2, 1.6\}\) m while the first microphone is placed at \(\{8.33, 1, 1.6\}\) m. The sampling rate is 16 kHz giving 1023 zeros for each channel. Then, recorded acoustic impulse responses are obtained from the MARDY database [130] where five-channel impulse responses are down-sampled to 16 kHz giving 1364 zeros for each channel. The microphone spacing is also 5 cm while the source is located at 1 m in front of the microphone array. Results showing the effect of common zeros on BSI and channel equalization algorithms will also be presented through the use of GMC clustering algorithms.

4.4.1 Performance of the GMC Algorithms

The variation of the number of clusters \(N_c\) against the pairwise tolerance \(\delta_c\) for GMC-DC and GMC-ST is first plotted in Fig. 4.6 for both simulated and recorded systems with \(M = 2\) channels. Although plotted on the same axes, the intention is not to compare both systems. It can be seen that as \(\delta_c\) increases, \(N_c\) for both systems increases as expected. In addition, no exactly-common zeros are found from both systems for \(\delta_c = 0\). The number of clusters found using GMC-DC is the same as GMC-ST.

For an example case of \(M = 4\) using simulated impulse responses, the efficiency of
4.4 Simulation Results

GMC-ST and GMC-DC in terms of their simulation times in MATLAB on a 2 GHz processor with 2 GB of memory is shown in Fig. 4.7 where the time required to extract clusters $C_{\{1:M\}}$ is plotted against various $\delta_c$, and the impulse responses are truncated to 512 taps to keep the simulations computationally tractable. As can be seen, the simulation times for both algorithms increase with $\delta_c$ since $N_c$ increases with $\delta_c$. It is also interesting to observe that for $\delta_c \leq 4 \times 10^{-3}$, GMC-DC is faster than GMC-ST, but the former becomes much slower as $\delta_c$ increases. This is because when $\delta_c$ is small, the dimensions of $C_{\{m,n\}}$ denoted by $N_{c,mn}$ is small for the first stage in GMC-DC, and so is the processing required for the following stages. GMC-ST, however, computes $D_{\{m,n\}}$ a total of $M(M-1)/2$ times initially, hence requiring more time. As $\delta_c$ increases, $N_{c,mn}$ becomes larger for the first stage in GMC-DC. As a result, (4.13) and (4.16) need to be invoked more times subsequently, and the resultant matrices propagate along the later stages of GMC-DC causing a significant increment in computational times. For GMC-ST, the search starts with $C_{\{m,n\}}$ and is confined within the space $\{C_{\{1,2\}}, C_{\{1,3\}}, \ldots, C_{\{M-1,M\}}\}$. This reduces the computational times significantly compared to GMC-DC. Nevertheless, it should be noted that clusters of common zeros with large $\delta_c$ is expected to be less harmful in the context of BSI, as indicated in Fig. 4.2.

4.4.2 Applications to Blind System Identification and Channel Equalization

The developed GMC algorithms now allow further studies of the effect of common zeros on BSI and channel equalization. First of all, variations of $N_c$ against tolerance $\delta_c$ for different number of channels $M$ is plotted in Fig. 4.8. It can be seen that $N_c$ increases with $\delta_c$ as expected, and GMC-ST and GMC-DC produce same results. More importantly, for each $\delta_c$, the number of clusters reduces with increasing $M$. For all cases of $M$, no exactly-common zeros are found for $\delta_c = 0$. Figure 4.9 then shows the number of clusters $N_c$ found using GMC-DC from simulated and recorded two-channel acoustic systems that are truncated to various lengths for $\delta_c = 1 \times 10^{-3}$. As can be seen, $N_c$ increases with an increasing $L$. This result hence confirms the expectation drawn from Fig. 4.4 that SIMO acoustic systems with high order can contain zeros with a high density.

To show the effect of common zeros on BSI algorithms, the NMCFLMS algorithm \[23\], as will be reviewed in Chapter 6, is employed over simulated SIMO acoustic
systems. By extracting systems with different number of channels and using a sequence of WGN as the input signal, the NMCLFLMS algorithm is run with a step-size of 0.5. A high SNR of 60 dB is chosen to avoid the misconvergence problem [156]. The BSI performance is measured by the NPM $\eta'(n)$ [113] defined in (2.27). The number of clusters $N_c$ found using GMC-ST for $\delta_c = 6 \times 10^{-3}$ in each case of $M$ is plotted in Fig. 4.10. As can be seen, the performance of NMCLFLMS increases with $M$ since $N_c$ reduces with increasing $M$. Using the MARDY database, Fig. 4.11 further shows the BSI performance of NMCLFLMS in terms of NPM at 10 s against $N_c$ obtained using GMC-ST for various $\delta_c$. The link between $\delta_c$, $N_c$, $M$, and $\eta'(n)$ can be clearly seen.
Since channel equalization algorithms can be used to perform speech dereverberation, Fig. 4.12 plots the performance of the MINT algorithm [123] evaluated using SDR [124] as defined by (2.58) against different number of channels \( M \) over the same set of simulated SIMO acoustic systems as used in Fig. 4.10. A similar monotonic relationship between common zeros and algorithm performance can be seen, which suggests the performance in SDR increases with \( M \) since \( N_c \) reduces with increasing \( M \).

The results shown so far indicates that the dependency of the BSI and channel equalization algorithms on the common zeros can be studied using the proposed clustering algorithms. They also agree with the intuition that the common zeros problem can be mitigated by using arrays with a large number of microphones, since the performance of BSI and channel equalization algorithms improve with increasing \( M \) which results in a reducing \( N_c \). More importantly, it can be expected that they would also affect the overall performance of dereverberation. Using a speech sample obtained from the APLAWD database [68] which contains both male and female utterances and is down-sampled to 8 kHz, the NMCFLMS [23] and MINT [123] algorithms are employed using a step-size of 0.2 and an SNR of 60 dB, over two-channel simulated SIMO systems, to perform speech dereverberation, of which the performance is measured by BSD [74] as defined in (2.59). Figure 4.13 plots the resultant BSD score against various \( N_c \) found using the GMC-ST algorithm with \( \delta_c = 2 \times 10^{-3} \). As can be seen, BSD score increases with number of clusters \( N_c \) indicating the degradation of the dereverberation performance. This suggests that
the inaccurate channel estimates produced by BSI algorithm can propagate to the also ill-performing channel equalization algorithm due to the presence of common zeros, which causes the degradation of dereverberation performance.

4.5 Summary

In this chapter, the classic common zeros problem in blind identification of SIMO systems has been addressed and extended by introducing the concept and definition of near-common zeros (NCZs). Such extension was demonstrated by analytical results which showed that zeros of long channels tend to distribute around the unit circle with a high density. In order to quantify the effect of common zeros on BSI and channel equalization algorithm, two clustering algorithms for multichannel systems with high order have been developed. These algorithms employ efficient approaches to compute the pairwise Euclidean distances and are generalized for multichannel systems with arbitrary dimensions. The GMC-DC algorithm extracts clusters using a binary tree approach whereas the GMC-ST algorithm concentrates on searching for solutions within subcluster groups, hence they are computationally attractive. The GMC-ST algorithm, for example, has computational times of the order 5 s on a typical MATLAB implementation for a four-channel system with 512 FIR coefficients per channel.

The developed clustering algorithms can facilitate the study of common zeros problem in SIMO acoustic systems, where simulation results obtained by applying them to the NMCFLMS and MINT algorithms have numerically established the link between tolerance $\delta_c$, number of clusters $N_c$, number of channels $M$, the performance of BSI and channel equalization algorithms, and that of the subsequent speech dereverberation. In the following two chapters of this thesis, two algorithms will be developed to mitigate respectively the performance degradation due to the presence of exactly- and near-common zeros.
Chapter 5

Channel Decomposition for Blind System Identification and Speech Dereverberation

BLIND identification of SIMO acoustic systems has been shown problematic in the presence of common zeros in Chapter 4. When exactly-common zeros exist, certain frequencies of the channel transfer functions are unidentifiable due to insufficient modes to make them fully excited, hence these exactly-common zeros are erroneously considered to arise from the source signal. The presence of NCZs limits the performance of BSI and channel equalization. The resultant identification error in the channel estimates propagates to the subsequent channel equalization algorithms causing further performance degradation for speech dereverberation. However, research contributions in the literature aiming to improve the BSI performance in the presence of common zeros are limited. This is perhaps due to the lack of motivation in the communications community and the difficulties of factorizing long FIR impulse responses in the acoustic signal processing community as described in Section 4.1.

As a special case in the context of the common zeros problem, exactly-common zeros can be found in communication systems even though the channel length is usually short. An inexhaustive list of such communication systems includes [107], for example, (i) channels obtained with delays that are all multiples of the symbol period $T_s$;
and (ii) band-limited channels with frequency nulls in the range of \([-\pi(1 - \beta_r)/T_s, \pi(1 - \beta_r)/T_s]\), where \(\beta_r\) is the roll-off parameter. It is noted that no zeros were found to be exactly-common from either simulated or recorded SIMO acoustic systems as described in Section 4.4. They are nevertheless expected to exist in longer impulse responses due to their highly dense distribution around the unit circle as described in Chapter 4.

In this chapter, the concept of channel decomposition is investigated. Utilizing this concept, the exactly-common zeros in SIMO systems can be separated from the remaining characteristic (non-common) zeros. This gives rise to two subsystem components, which can be identified sequentially by employing classic BSI algorithms in combination with a single-channel identification approach. Integrating the channel equalization algorithms in a similar manner then leads to a robust two-stage approach for speech dereverberation in the presence of common zeros. This chapter is organized as follows: In Section 5.1, the concept of channel decomposition is introduced and utilized to demonstrate the effect of exactly-common zeros on CR-based BSI algorithms. In order to achieve accurate channel decomposition, Section 5.2 presents an effective method for estimating the number of exactly-common zeros \(N_c\) based on eigen-analysis of the system outputs, where it is demonstrated that correct estimation of \(N_c\) allows BSI algorithms to be “aware of” and take account of the presence of these zeros. Utilizing this, a two-stage BSI algorithm is developed in Section 5.3 to identify the decomposed channel components sequentially. This algorithm is then combined with channel equalization algorithms, in Section 5.4, to form a two-stage speech dereverberation technique robust to exactly-common zeros. Simulation results are presented in Section 5.5 to demonstrate the performance of the two-stage algorithm and its application to speech dereverberation over various SIMO systems. Section 5.6 summarizes this chapter.

5.1 Introduction to Channel Decomposition

The concept of channel decomposition is inspired by the idea of separating a linear multichannel system into several components either in time domain or frequency domain such that these components can be processed individually and then recombined to recover the full system. The well-known subband-based multirate processing using filter banks in the context of AEC is a good example [24].
For an $M$-channel SIMO system as shown in Fig. 5.1, it has been demonstrated in Chapter 4 that common zeros are likely to occur in such multichannel systems with a large number of FIR coefficients. Since this chapter focuses on exactly-common zeros, it is assumed that no NCZs presents in the SIMO system. With reference to the definition of exactly-common zeros in (4.3) and (4.4), it is noted, that by separating these zeros from the characteristic (non-common) ones in the $z$-domain, the $m$th channel transfer function of length $L$ can be expressed by

$$H_m(z) = H_C(z)H_m(z), \quad m = 1, 2, \ldots, M, \quad (5.1)$$

where $H_C(z)$ has been defined in (4.4) and can be rewritten

$$H_C(z) = K_C \prod_{p=1}^{N_c} \left( 1 - z_C^{-1}(p) \right) \quad (5.2)$$

to denote the channel component associated with $N_c$ exactly-common zeros for $1 \leq N_c \leq L - 1$, $K_C \in \mathbb{Z}^+$, and $H_m(z)$ denotes the channel component associated with the characteristic zeros in channel $m$ such that

$$\mathcal{H}_m(z) = \frac{H_m(z)}{H_C(z)} = \tilde{K}_m \prod_{p=1}^{L-N_c-1} \left( 1 - \tilde{z}_m^{-1}(p) \right), \quad m = 1, 2, \ldots, M \quad (5.3)$$

with $\tilde{K}_m = K_m / K_C$ according to (4.1) and $\tilde{z}_m(p)$ being the $p$th characteristic zero of the $m$th channel. As a result, the original SIMO system, $H_m(z)$, is now decomposed into a single channel, $H_C(z)$, containing all $N_c$ exactly-common zeros and a sub-SIMO system.
5.1 Introduction to Channel Decomposition

Figure 5.2: Channel decomposition for SIMO system with exactly-common zeros.

$\mathcal{H}_m(z)$ formed by the remaining $L - N_c - 1$ non-common zeros of each channel. Figure 5.2 depicts such decomposition, from which the $m$th output signal can be written

$$X_m(z) = S(z)H_C(z)H_m(z) + B_m(z), \quad m = 1, 2, \ldots, M,$$

where $S(z), X_m(z)$ and $B_m(z)$ are the $z$-transforms of the source signal $s(n)$, received signal $x_m(n)$ and the additive noise $b_m(n)$ respectively; and

$$X_C(z) = S(z)H_C(z)$$

can be considered either the output signal from $H_C(z)$, or the source signal for $H_m(z)$.

As discussed in Section 2.4.2, classic BSI algorithms fail to work successfully in the presence of exactly-common zeros since these zeros give rise to $H_C(z) \neq 1$, which can be mistakenly considered part of $S(z)$ by the BSI algorithms. Using the channel decomposition approach described in (5.1), this can now be mathematically explained. Consider the $M$-channel SIMO system shown in Fig. 5.1 the CR error defined in (2.4) can be rewritten in the $z$-domain

$$\mathcal{E}_{ml}(z) = X_m(z)\hat{H}_l(z) - X_l(z)\hat{H}_m(z), \quad m, l = 1, 2, \ldots, M, \ m \neq l,$$

where $\hat{H}_m(z)$ is the $z$-transform of the estimated impulse response for channel $m$. If $N_c \neq$
0, invoking (5.1) results in
\[
E_{ml}(z) = \hat{H}_C(z) \left[ X_m(z) \hat{H}_l(z) - X_l(z) \hat{H}_m(z) \right], \tag{5.7}
\]
where \( \hat{H}_m(z) = \hat{H}_C(z) \hat{H}_m(z) \). It can thus be seen that any transfer function can replace \( \hat{H}_C(z) \) without affecting the minimization of (5.7), hence the solution to \( H_m(z) \) produced by CR-based BSI algorithms can be expected to contain the arbitrary effect \( \hat{H}_C(z) \) when \( N_c > 0 \), which is non-unique.

### 5.2 Blind Order Estimation for Exactly-Common Zeros

As described in Section 2.4.1, the order of the unknown system \( L \) is assumed to be available throughout this thesis. Implementing channel decomposition, however, is not straightforward since the order of either \( H_C(z) \) or \( H_m(z) \) is required. Recall that the key to the failure of classic BSI algorithms is that they do not take account of the presence of exactly-common zeros, but instead treat them as non-common ones. If BSI algorithms realize the existence of these unidentifiable zeros and concentrate on the remaining sub-SIMO system \( H_m(z) \) containing only characteristic zeros, an accurate \( \hat{H}_m(z) \) can be obtained since the identifiability condition (C2.1) is satisfied for the sub-SIMO system containing only non-common zeros. Consequently, accurately estimating the number of exactly-common zeros \( N_c \) is important for developing a BSI algorithm based on channel decomposition.

To demonstrate this, a simulation example is presented, where the subspace algorithm [19] reviewed in Section 2.4.4 is employed over an example two-channel SIMO system with its coefficients extracted from a standard Gaussian distribution. This system is of length \( L = 32 \) and contains \( N_c = 8 \) exactly-common zeros as shown in Fig. 5.3. In Fig. 5.4, the BSI performance of the subspace algorithm for estimating \( H_m(z) \), in terms of NPM as defined in (2.27), is plotted against the variation of the estimated number of exactly-common zeros \( \hat{N}_c \), where a sequence of WGN with 60 dB SNR is used as input signal. To estimate \( H_m(z) \), the order of subspace algorithm is set \( L - N_c \), which is equivalent to constructing the matrix \( X_m \) defined by (2.47) to be of dimension \((L - N_c) \times (N + L - N_c - 1)\) so that the resultant correlation-like matrix \( \hat{R}_x \) is of di-
5.2 Blind Order Estimation for Exactly-Common Zeros

Figure 5.3: Zeros of the example two-channel SIMO system for channel 1 (circles), channel 2 (squares) and exactly-common zeros (triangles).

Figure 5.4: BSI performance on estimating $\mathcal{H}_m(z)$ against various $\hat{N}_c$.

As can be seen from the figure, when $\hat{N}_c = 8$ the subspace algorithm obtains the correct information that the characteristic channel component $\mathcal{H}_m(z)$ is of length $L - \hat{N}_c = 24$, thus managing to produce an accurate $\tilde{\mathcal{H}}_m(z)$. It is also interesting to note that for $\hat{N}_c < 8$, the length of characteristic channel component $L - N_c$ is over-estimated. Although it was stated in [19] that the subspace algorithm could work in such situation, its performance is severely degraded.

As described in Section 4.1, various algorithms have been developed to detect and estimate exactly-common zeros without factorizing channel transfer functions [152, 153, 154]. However, these methods either support only two channels or suffer from approximation errors. In [92], a closed-form approach was proposed to determine the overall system order based on eigen-analysis of the system outputs (A similar approach was proposed later in [104]). Although not explicitly stated in the paper, such method can be utilized to estimate $N_c$ and this extension is employed in this chapter. With the reference to the system equation defined by (2.22), the autocorrelation matrix of the received signal $x(n)$ is given by

$$R_x = E\{x(n)x^T(n)\} = HR_sH^T + \sigma_b^2 I_{ML \times ML},$$

(5.8)

where $R_s = E\{s(n)s^T(n)\}$ is the autocorrelation matrix of the input signal, $\sigma_b^2$ is the variance of the uncorrelated additive noise with zero mean. Consider the example two-channel SIMO system with its zeros shown in Fig. 5.3, an approximated $\tilde{R}_x$ can be ob-
5.3 The Two-stage Blind System Identification Algorithm

Utilizing the blind order estimation method which allows successful channel decomposition, a two-stage BSI algorithm is now developed in this section. This algorithm first identifies the characteristic channel components $\mathcal{H}_m(z)$ blindly using classic BSI algorithms to obtain an estimate of $X_C(z)$, denoted by $\hat{X}_C(z)$. Next, the exactly-common zeros component $H_C(z)$ is identified using a single-channel approach. Finally, the overall system estimate $\hat{H}_m(z)$ is obtained by convolving $\hat{H}_m(z)$ with $\hat{H}_C(z)$.

Figure 5.5: Eigenvalue distribution of $\hat{R}_x$ for the output signals of the example two-channel SIMO system of length $L = 32$.

The resultant $ML = 64$ eigenvalues of $\hat{R}_x$ for this illustrative example, excited by a sequence of WGN with 60 dB SNR and computed using eigenvalue decomposition (EVD), are sorted in ascending order and plotted in Fig. 5.5. As can be seen, the first 9 eigenvalues are much smaller than the remaining ones. Hence, $\hat{N}_c = 8$ according to [92] and the order of $\mathcal{H}_m(z)$ can be correctly obtained as $L - \hat{N}_c = 24$. It is worthwhile recalling that, as described in Section 2.4.4 regarding the CR-based subspace algorithm [19], the relative amplitude between eigenvalues of $\hat{R}_x$ is robust to WGN which introduces only a global additive value $\sigma_b^2$. However, colored noise can affect the performance of this order estimation method.

The two experiments in this section indicate that $H_C(z)$ is essentially the channel with over-estimated order when it comes to EVD; and obtaining the order of $H_C(z)$, namely $N_c$, enables the subspace algorithm to estimate $\mathcal{H}_m(z)$ accurately since the resultant $M(L - N_c) \times M(L - N_c)$ correlation-like matrix $\tilde{R}_x$ is of full-rank. This verifies the result shown in Fig. 5.4 and the failure of the subspace algorithm for $N_c \neq 0$ is expected.
Stage 1: Blind Identification of Characteristic Zeros

As indicated in (5.4), $\mathcal{H}_m(z)$, $\forall m$ does not contain exactly-common zeros, hence satisfying the channel identifiability condition (C2.1). By obtaining $\hat{N}_c$, $\mathcal{H}_m(z)$ can be blindly identified by, for example, the subspace method [19] with order $L - \hat{N}_c$ such as demonstrated by the illustrative example in Section 5.2. Since no NCZs is assumed, the solution to $\mathcal{H}_m(z)$ can be found from the eigenvector associated with the smallest eigenvalue of the $M(L - \hat{N}_c) \times M(L - \hat{N}_c)$ matrix $\hat{R}_x$ as derived in Section 2.4.4. Alternatively, adaptive algorithms such as [18, 23] can be employed. Then, with the aid of a multichannel equalization algorithm such as MINT [123], a multichannel inverse system of $\hat{\mathcal{H}}_m(z)$, denoted by $\hat{\mathcal{G}}_m(z)$, can be derived such that $\hat{X}_C(z)$ is given, as described in Section 2.5, by

$$\hat{X}_C(z) = \sum_{m=1}^{M} \hat{G}_m(z)X_m(z)$$

$$= \left[ \sum_{m=1}^{M} \hat{G}_m(z)\mathcal{H}_m(z) \right] X_C(z) + \sum_{m=1}^{M} \hat{G}_m(z)B_m(z).$$  (5.9)

Stage 2: Single-Channel Identification of Exactly-Common Zeros

In the second stage, the channel component $H_C(z)$ is identified using $\hat{X}_C(z)$ obtained from (5.9) in stage 1. Recall the assumption (A2.1) discussed in Section 2.2.1 and Section 2.4.1 where acoustic impulse responses are considered quasi-stationary. Such assumption is not invalid in practice, especially when the enclosure of the room or the loudspeaker-microphone configuration is fixed. This motivates the development of the single-channel identification approach.

As indicated by Fig. 5.2 and (5.5), zeros from $H_C(z)$ are contained in $X_C(z)$. Since $S(z)$ is time-varying, the stationarity of $H_C(z)$ compared to that of $S(z)$ can be exploited. Similar to (5.2), denote $S(z)$ and $X_C(z)$ in factorized forms as

$$S(z) = K_S \prod_{p=1}^{N_s} \left( 1 - z_s^{-1}(p) \right),$$  (5.10)

$$X_C(z) = K_{xc} \prod_{p=1}^{N_{xc}} \left( 1 - z_{xc}^{-1}(p) \right),$$  (5.11)

where $K_s, K_{xc} \in \mathbb{Z}^+$, $N_s, N_{xc}$ denote respectively the numbers of zeros for $S(z)$ and $X_c(z)$.
5.3 The Two-stage Blind System Identification Algorithm

within a frame of finite length for which the channel is considered stationary. During this period of time, there is a fixed pattern of \( z_C(p) \) compared to the time-varying \( z_s(p) \). Such pattern difference can be identified by observing \( z_{xc}(p) \) over several frames of \( X_C(z) \) so as to distinguish zeros that are stationary from dynamic ones. As before, the fast factorization algorithm [150] can be employed when a large frame size is chosen.

To implement this single-channel approach of identifying \( H_C(z) \), a quantized grid is designed in the \( z \)-plane with respect to the magnitude and phase in order to capture the positions of the zeros. A similar technique was employed in [157] for single-channel dereverberation. As discussed in Section 4.2, \( N_c \) can be large for long impulse response where zeros tend to cluster around the unit circle with uniformly distributed phases. A nonlinear magnitude quantization is thus desirable for sufficient accuracy. Denote \( B_m \) as the number of bits used for a linear quantization ranging from 0 to 1, a nonlinear function [157] is used to map this linear quantization to a nonlinear one such that

\[
\Lambda_q = \frac{1}{\tan(\pi/\varrho)} \left[ \tan\left(\frac{\pi}{\varrho}\right) + \tan\left(\frac{\pi}{\varrho}a_q - \frac{\pi}{\varrho}\right) \right],
\]

(5.12)

where \( 0 \leq \varrho \leq 1 \) is a control parameter and \( a_q \) is the linearly quantized value. A \( B_p \)-bit linear quantization is imposed with respect to the phase. An example of such quantized grid is shown in Fig. 5.6 and a nonlinear quantization using \( \Lambda_q \) for an example case of \( B_m = 10, B_p = 12, \varrho = 2.05 \) is shown in Fig. 5.7.

With reference to Fig. 5.6, an integer counter is defined and initialized to zero for each cell of the grid. For each frame of \( X_C(z) \), \( z_{xc}(p) \) is computed and mapped into one
of these cells, whose corresponding counter is then incremented by one. Non-minimum phase zeros that are outside the unit circle are converted to their reciprocals before being processed, and they will be converted back during reconstruction of the estimated impulse response. After evaluating $N_f$ frames and there are $N_c$ cells with counter values being equal to $N_f$, the identification process is completed.

It should be noted that due to the linear convolution between the $S(z)$ and $H_C(z)$ as shown in (5.5), $z_C(p)$ can spread among the zeros of $X_C(z)$, hence the frame length is required to be accordingly long so as to contain every $z_C(p)$. This, however, requires long factorization resulting in not only unneglectable algorithmic delay but also numerical approximation errors that degrade the identification performance. As a result, the frame length of $X_C(z)$ should be carefully chosen, from which $z_C(p)$ is then found sequentially as frames of $X_C(z)$ are processed and $z_{xc}(p)$ are computed. Although it may not be possible to identify all $z_C(p)$ within one frame, it has been found from experiments that less than 5 frames are usually sufficient, as will be used in simulations in Section 5.5.

In practice, the frame length can be determined based on $\hat{N}_c$. This implementation ensures the performance of zero identification is accurate incurring only a small processing delay. After $\hat{H}_C(z)$ and $\hat{H}_m(z)$ are obtained, the overall system estimate $\hat{H}_m(z)$ can be derived as

$$\hat{H}_m(z) = \hat{H}_C(z)\hat{H}_m(z), \quad m = 1, 2, \ldots, M.$$  

(5.13)

### 5.4 Speech Dereverberation based on Channel Decomposition

Based on the two-stage BSI algorithm presented in Section 5.3, a speech dereverberation approach robust to exactly-common zeros is developed. Similar to Section 5.1, a mathematical explanation can be derived, using the concept of channel decomposition, to show the effect of exactly-common zeros on multichannel equalization algorithms using Bezout theorem such as MINT [123]. As described in Section 2.5, the objective of channel equalization is to find an estimate of the source signal $\hat{S}(z)$ such that

$$\hat{S}(z) = \sum_{m=1}^{M} \hat{G}_m(z)X_m(z)$$

$$= \left[ \sum_{m=1}^{M} \hat{G}_m(z)H_m(z) \right] S(z) + \sum_{m=1}^{M} \hat{G}_m(z)B_m(z)$$  

(5.14)
for $m = 1, 2, \ldots, M$, where $\hat{G}_m(z)$ is the multichannel inverse of $\hat{H}_m(z)$ satisfying
\begin{equation}
\sum_{m=1}^{M} \hat{G}_m(z) \hat{H}_m(z) = \vartheta z^{-\tau},
\end{equation}
as long as $H_C(z) = \gcd\{H_1(z), H_2(z), \ldots, H_M(z)\} = 1$ according to the Bezout theorem [43] with $\vartheta$ and $\tau$ defined in (2.52). However, if $N_c > 0$ and $H_C(z) \neq 1$, the solution for $\hat{G}_m(z)$ can only be found such that
\begin{equation}
\sum_{m=1}^{M} \hat{G}_m(z) \hat{H}_m(z) = \vartheta z^{-\tau}.
\end{equation}
As a result of this, (5.14) becomes
\begin{equation}
\hat{S}(z) = \vartheta z^{-\tau} H_C(z) S(z) + \sum_{m=1}^{M} \hat{G}_m(z) B_m(z),
\end{equation}
and $\hat{S}(z)$ will be distorted by $H_C(z)$.

Now, utilizing the two-stage BSI approach based on channel decomposition, where $\hat{H}_m(z)$ and $\hat{H}_C(z)$ are obtained separately as described in Section 5.3, the Bezout theorem can first be applied to $\hat{H}_m(z)$ to obtain $\hat{X}_C(z)$ using (5.9). In the second stage, the SCLS algorithm [119] derived in (2.53) can be employed to compute $\hat{G}_C(z)$ such that
\begin{equation}
\hat{G}_C(z) \hat{H}_C(z) = \vartheta z^{-\tau}.
\end{equation}
The estimated source signal can then be obtained as
\begin{equation}
\hat{S}(z) = \hat{G}_C(z) \hat{X}_C(z)
\end{equation}
\begin{equation}
= \hat{G}_C(z) \left[ \vartheta z^{-\tau} X_C(z) + \sum_{m=1}^{M} \hat{G}_m(z) B_m(z) \right]
\end{equation}
\begin{equation}
= \vartheta^2 z^{-2\tau} S(z) + \hat{G}_C(z) \sum_{m=1}^{M} \hat{G}_m(z) B_m(z).
\end{equation}
Note that the second term on the right-hand side of (5.19) is due to the presence of additive noise, which affects the accuracy of $\hat{S}(z)$. A regularization method can be employed on MINT as proposed in [124] to mitigate such problem, which is out of the scope of this thesis.
In summary, a system diagram of the two-stage speech dereverberation approach is shown in Fig. 5.8, where it can be seen that \( \hat{N}_C \), obtained by using the blind order estimation described in Section 5.2, not only validates the subspace algorithm, but can also be used to determine the frame length for the single-channel identification of \( \hat{H}_C(z) \) such as described in Section 5.3. Finally, it should be noted that if \( H_C(z) \) is non-minimum phase with zeros outside the unit circle, \( \hat{G}_C(z) \) can be unstable.

5.5 Simulation Results

In this section, simulation results are presented to evaluate the performance of the developed approach. For the two-stage BSI algorithm described in Section 5.3, the two-channel SIMO system with its zeros shown in Fig. 5.3 is used. An example with low order is chosen in this case in order to present clearly the identification performance of individual zeros. For the two-stage speech dereverberation described subsequently in Section 5.4, another two-channel SIMO system is obtained by extracting two recorded acoustic impulse responses from the MARDY database [130] as shown in Fig. 5.9, where the distance between each microphone is 5 cm and the speaker is positioned 1 m away from the microphones. These impulse responses are down-sampled to 8 kHz and truncated to 512 FIR coefficients. To illustrate the effectiveness of the developed algorithms under condition containing exactly-common zeros, \( N_c = 9 \) randomly generated exactly-common zeros are super-imposed onto these recorded impulse responses.
5.5 Simulation Results

Figure 5.9: Recorded room impulse responses from the MARDY database \([130]\) with superimposed exactly-common zeros.

Figure 5.10: Performance of the two-stage BSI approach in \(z\)-domain for (a) Stage 1: True characteristic zeros (circles) and identified zeros (crosses), (b) Stage 2: True exactly-common zeros (triangles) and identified zeros (crosses).

5.5.1 The Two-stage Blind System Identification

The performance of the two-stage BSI algorithm is first presented. A sequence of WGN is used as the source signal and the subspace algorithm \([19]\) is employed.

Figure 5.10 shows the performance of the two-stage approach in \(z\)-plane stage-by-stage for a noiseless case, where the true channel zeros are denoted by circles/triangles and crosses represent the identified ones. As can be seen from Fig. 5.10(a), the subspace algorithm \([19]\) works successfully in estimating the sub-SIMO system \(H_m(z)\) containing only characteristic zeros. Thanks to the blind order estimation method discussed in Sec-
Figure 5.11: Performance of the two-stage BSI approach in the time domain for (a) the true channel impulse response $h_1$, (b) estimated channel impulse response $\hat{h}_1$ using the two-stage approach, and (c) the error defined by $|h_1 - \hat{h}_1|$.

For Figure 5.12, the BSI performance of the two-stage approach in NPM is compared with that of the standard subspace algorithm [19] under various levels of noise. As can be seen, due to the robustness to exactly-common zeros, the two-stage approach outperforms the subspace algorithm for SNR > 25 dB with a more than 10 dB improvement in NPM at SNR = 40 dB. However, since the subspace algorithm is nevertheless employed in stage 1, its inherent sensitiveness to additive noise limits the overall performance of the two-stage method. As a result, when SNR is low, the issue of noise robustness becomes overwhelming and both algorithms fail to work successfully. It is thus expected
5.5 Simulation Results

Figure 5.12: Comparison of BSI performance in NPM against SNR.

Figure 5.13: Comparison of BSI performance in NPM against channel length $L$.

that replacing the subspace algorithm in stage 1 by other BSI algorithms with noise robustness such as proposed in [100, 102, 101, 103] can effectively improve the performance of the two-stage algorithm under noisy conditions.

Further comparison of BSI performance in NPM is performed using the MCLMS [18] and the subspace algorithm [19] over a set of two-channel SIMO systems of various lengths where the coefficients are drawn from standard Gaussian distribution so that $N_c$ is not specified. The SNR is set 60 dB to effectively remove the influence of noise as indicated in Fig. 5.12. For the MCLMS algorithm, the steady-state NPM value is used. All algorithms are run for 100 independent trials and the results are obtained from the average of these trials. Figure 5.13 shows the results where the NPM value is plotted against channel length $L$ for the MCLMS algorithm (diamonds), subspace algorithm (circles) and the two-stage method (squares). As can be seen, the two-stage method provides an approximately 5 dB improvement in NPM over the subspace method, while the MCLMS algorithm does not work successfully.

5.5.2 Application to Speech Dereverberation

In this simulation, a speech sample comprising an utterance of a female talker is extracted from the APLAWD database [68] and used as the source signal with an SNR of 60 dB. The recorded impulse responses shown in Fig. 5.9 is used. The sampling frequency is 8 kHz. For channel equalization, the MINT algorithm [123] is employed and the dereverberation performance is measured using BSD score [74] as defined in (2.59) which is reproduced.
5.5 Simulation Results

Figure 5.14: Time-domain samples of the (a) original speech, (b) reverberant speech, and recovered speech using (c) the subspace algorithm, and (d) the two-stage approach.

Here for convenience

\[
\text{BSD} = \frac{\sum_{k=0}^{K-1} \sum_{n=kN_B}^{kN_B+N_B-1} (B_s(k,n) - B_\hat{s}(k,n))^2}{\sum_{k=0}^{K-1} \sum_{n=kN_B}^{kN_B+N_B-1} (B_s(k,n))^2},
\]

where \(N_B\) is the frame length in samples, \(B_s(k,n)\) and \(B_\hat{s}(k,n)\) are the Bark spectra of the clean speech \(s(n)\) and the estimated speech \(\hat{s}(n)\), respectively. Note that \(\hat{X}_C(z)\) is segmented into frames of length 256 samples in the second stage.

Figure 5.14 first shows the normalized time-domain samples of (a) the original speech signal, (b) the reverberant speech signal, recovered speech signals using (c) the standard subspace algorithm and (d) the two-stage approach. It can be seen that the two-stage approach produces a better estimate of the original speech signal compared to the subspace method in the presence of exactly-common zeros. The corresponding spectrogram is shown in Fig. 5.15, where the standard subspace method gave a BSD score of 0.4309, while the two-stage approach scores 0.0016.

A further simulation is performed to compare the dereverberation performance in terms of BSD score using the same parameters except that various \(N_c\) are super-imposed to the recorded impulse responses. The simulation is run for 100 independent trials and the result is shown in Fig. 5.16. As can be seen, the two-stage algorithm provides a consis-
Figure 5.15: Spectrograms of the (a) original speech, (b) reverberant speech, dereverberated speech using (c) the subspace method, and (d) the two-stage method.

Figure 5.16: Variation of BSD score against $N_c$ using recorded impulse responses.

tent improvement in terms of BSD over standard subspace algorithm, although it reduces to the subspace algorithm when $N_c = 0$ as expected.

5.6 Summary

In this chapter, the concept of channel decomposition has been introduced, which decomposes a SIMO system into its characteristic component and its common component each comprising non-common and exactly-common zeros respectively. Utilizing this concept, it was shown that the presence of exactly-common zeros results in non-unique solutions from CR-based BSI algorithms and channel equalization algorithms using Bezout theorem. A two-stage BSI algorithm was then developed to sequentially identify charac-
teristic zeros and exactly-common zeros. To achieve this, an order estimation method was presented for blind estimation of the number of exactly-common zeros, which was shown to be essential for BSI algorithm to produce accurate estimation. The remaining common zeros are identified by taking into account the quasi-stationarity of channel zeros compared to the zeros from the source signal on a frame-by-frame basis. A non-linear quantized grid was designed in the $z$-plane to capture channel zeros efficiently. The effectiveness of the two-stage BSI algorithm motivates a corresponding speech dereverberation approach, where the characteristic channel component is equalized using Bezout theorem and the SCLS method is employed for the exactly-common zeros. Simulation results from a number of experiments confirmed the efficiency of the developed method, over existing methods, on BSI and dereverberation performance in terms of NPM and BSD respectively.
Chapter 6

Blind System Identification using Forced Spectral Diversity for Speech Dereverberation

The significance of the common zeros problem on blindly identifying SIMO systems using SOS-based BSI algorithms has been demonstrated and mathematically described in Chapter 4 and Chapter 5. Indeed, the presence of exactly-common zeros cannot be detected by classic BSI algorithms, hence cannot be accurately identified. This has motivated the development of the two-stage BSI algorithm in Chapter 5 which has been shown to achieve an improved performance in BSI and the subsequent speech dereverberation. However, SIMO systems with high order such as acoustic impulse responses results in zeros clustering around the unit circle with high density giving rise to the near-common zeros (NCZs). These zeros do not violate the coprimeness identifiability condition (C2.1) as described in Section 2.4.2 but make the BSI algorithms numerically ill-conditioned, hence causing performance degradation.

In this chapter, the problem of NCZs is further investigated and is interpreted using quantified multichannel diversity which can also be linked to the BSI performance in terms of NPM. Motivated by the detrimental effect of NCZs on BSI and speech dereverberation as presented in Chapter 4 a novel concept is developed by collectively combining effective channel undermodelling and spectral diversifying filters so as to mitigate
the effect of NCZs by deriving a modified system with additional diversity. This modified system is shown to be more diversified than the original system, hence resulting in an improved BSI and equalization performance for speech dereverberation. This concept is referred to as forced spectral diversity (FSD).

This chapter is organized as follows: In Section 6.1, the normalized multichannel frequency-domain LMS (NMCFLMS) algorithm [23] is reviewed and employed as the baseline BSI algorithm in this chapter. The direct-path constraint for improving the noise robustness [99] is also introduced. The link between quantified multichannel diversity and NCZs is established analytically in Section 6.2, where the former is computed by performing singular value decomposition (SVD) of the channel matrix, and the latter can be extracted using the GMC algorithms developed in Chapter 4. The concept of FSD is then developed in Section 6.3. To present this concept clearly, channel undermodelling is first reviewed in Section 6.3.1. Illustrative and numerical examples are presented in Section 6.3.2 and Section 6.3.3 with the aim of describing how spectral diversifying filters can be combined with channel undermodelling to achieve the objective of introducing additional channel diversity. In Section 6.4, FSD processing is applied to SIMO acoustic systems for blind identification and speech dereverberation. Additional remarks regarding the design of spectral diversifying filters are presented in Section 6.5. Section 6.6 presents a number of simulation results of applying FSD processing for BSI and speech dereverberation over simulated and recorded SIMO acoustic systems with only two channels, from which a consistent and significant performance improvement in terms of NPM and BSD score can be observed. The ability of two-channel FSD processing to achieve better performance over classic methods with five channels indicates the computational attractiveness of the FSD concept. This chapter is summarized in Section 6.7.

6.1 Introduction to the NMCFLMS Algorithm

The problem of BSI has been formulated in Section 2.4.1. With reference to Fig. 2.8(b), which is reproduced here in Fig. 6.1 for convenience, the \( m \)th microphone signal for an \( M \)-channel SIMO system is given, in a vector form, by

\[
x_m(n) = H_m s(n) + b_m(n), \quad m = 1, 2, \ldots, M
\]  

(6.1)
6.1 Introduction to the NMCFLMS Algorithm

![Diagram of (a) an M-channel SIMO acoustic system and (b) the problem of BSI](reproduced from Fig. 2.8).

as defined by (2.20). Concatenating (6.1) across $M$ channels, a system of equations is obtained

$$x(n) = Hs(n) + b(n).$$  \hspace{2cm} (6.2)

The objective of BSI is therefore to estimate $H$ given only $x(n)$, and this can be achieved to some level of accuracy by employing, for example, frequency-domain BSI algorithms.

As described in Section 3.1, one of the first frequency-domain adaptive algorithms proposed was the FLMS algorithm [141], where the overlap-save method [133] of implementing linear convolution using FFT is employed. Frequency-domain algorithms are therefore more computationally efficient than their time-domain counterparts. Since then, a wide range of frequency-domain algorithms derived from FLMS have been developed for supervised system identification in AEC and NEC [8]. With the same objective, time-domain MCLMS algorithm [18] reviewed in Section 2.4.4 was extended to form the frequency-domain NMCFLMS algorithm [23] with the aims of overcoming the problem of slow convergence and being suitable for real-time implementation. In [156], the fast convergence of the IPNLMS [139] and low processing delay brought about by the MDF structure [21] have also been exploited and incorporated into the NMCFLMS algorithm.

### 6.1.1 Algorithm Derivation

The NMCFLMS algorithm is developed based on the derivation of cross-relation (CR) in the frequency domain. With reference to (2.32) and (2.41), the linear convolution between the $m$th channel output $x_m(n)$ and the $l$th channel estimate $\hat{h}_l(n)$ can be implemented
using a vector of length $2L$ from the circular convolution

$$\tilde{y}_{ml}(k) = C_{x_m}(k) \hat{h}_l^{10}(k),$$

(6.3)

where $k$ is the frame index, $\hat{h}_l^{10}(k) = [\hat{h}_l^T(k) \ 0_{L \times 1}]^T$ is the $l$th channel estimate with zero padding, and

$$C_{x_m}(k) = \begin{bmatrix}
x_m(kL - L) & x_m(kL + L - 1) & \cdots & x_m(kL + L - 1) \\
x_m(kL - L + 1) & x_m(kL - L) & \cdots & x_m(kL + L - 2) \\
\vdots & \vdots & \ddots & \vdots \\
x_m(kL) & x_m(kL - L - 1) & \cdots & x_m(kL + 1) \\
\vdots & \vdots & \ddots & \vdots \\
x_m(kL + L - 1) & x_m(kL + L - 2) & \cdots & x_m(kL - L)
\end{bmatrix}$$

(6.4)

is the $2L \times 2L$ circulant matrix constructed from

$$\chi_m(k) = [x_m(kL - L) \ x_m(kL - L + 1) \ \ldots \ x_m(kL + L - 1)]^T.$$ 

(6.5)

It should be noted that $C_{x_m}(k)$ is only determined by $\chi_m(k)$, e.g., the second column of $C_{x_m}(k)$ is the shifted version of $\chi_m(k)$. As described in Section 3.2 and indicated by (6.5), a 50% overlap-save is employed so that $L$ previous samples of $x_m(n)$ is included in $\chi_m(k)$. For each $\tilde{y}_{ml}(k)$ of length $2L$, the last $L$ samples are retained since they correspond to the linear convolution given by $x_m^T(n) \hat{h}_l(n)$. As a result, by defining two selecting matrices

$$W_{01}^{01} = [0_{L \times L} \ I_{L \times L}], \quad W_{2L \times L}^{10} = [I_{L \times L} \ 0_{L \times L}]^T$$

(6.6)

similarly to (3.20d) and (3.20e), the desired result $y_{ml}(k)$ can be obtained from (6.3) as

$$y_{ml}(k) = W_{L \times 2L}^{01} \tilde{y}_{ml}(k) = W_{L \times 2L}^{01} C_{x_m}(k) \hat{h}_l^{10}(k),$$

$$= W_{L \times 2L}^{01} C_{x_m}(k) W_{2L \times 1}^{10} \hat{h}_l(k).$$

(6.7)
To employ FFT techniques for efficient implementation of circular convolution, it is important to note that the circulant matrix $C_{x_m}(k)$ can be decomposed as

$$C_{x_m}(k) = F_{2L}^{-1}D_m(k)F_{2L},$$  

where $F_{2L}$ is a $2L \times 2L$ DFT matrix as defined in (3.19) with $F_{2L}^H = F_{2L}^H/(2L)$, and $D_m(k)$ is a diagonal matrix with diagonal elements being given by the DFT of $\chi_m(k)$. Now, the frequency-domain CR error function can be written

$$e_{ml}(k) = F_L [y_{ml}(k) - y_{lm}(k)],$$

$$= F_l W_{L \times 2L}^{01} \left[ C_{x_m}(k) W_{2L \times 1}^{10} \hat{h}_l(k) - C_{x_l}(k) W_{2L \times 1}^{10} \hat{h}_m(k) \right],$$

$$= \mathcal{W}_{L \times 2L}^{01} \left[ D_m(k) W_{2L \times 1}^{10} \hat{h}_l(k) - D_l(k) W_{2L \times 1}^{10} \hat{h}_m(k) \right],$$

for $m, l = 1, 2, \ldots, M$, $m \neq l$, where

$$\mathcal{W}_{L \times 2L}^{01} = F_L W_{L \times 2L}^{01} F_{2L}^{-1}, \quad \mathcal{W}_{2L \times L}^{10} = F_{2L} W_{2L \times 1}^{10} F_L^{-1},$$

$$\hat{h}_m(k) = F_l \hat{h}_m(k).$$

The NMCFLMS algorithm is therefore given by [23], for $m = 1, 2, \ldots, M$,

$$\hat{h}_m^{10}(k) = F_{2L} \hat{h}_m^{10}(k) = F_{2L} W_{2L \times 1}^{10} \hat{h}_m(k),$$

$$\mathcal{P}_m(k) = \lambda \mathcal{P}_m(k - 1) + (1 - \lambda) \sum_{l=1, l \neq m}^M \mathcal{P}_l^\ast(k) \mathcal{P}_l(k),$$

$$\mathcal{E}_{ml}(k) = \mathcal{W}_{2L \times L}^{01} \mathcal{E}_{ml}(k) = F_{2L} \begin{bmatrix} 0_{L \times 1} \\ F_L^{-1} e_{ml}(k) \end{bmatrix},$$

$$\hat{h}_m^{10}(k) = \hat{h}_m^{10}(k - 1) - \mu [\mathcal{P}_m(k) + \delta I_{2L \times 2L}]^{-1} \times \sum_{l=1}^M \mathcal{P}_l^\ast(k) \mathcal{E}_{ml}(k),$$

where $\lambda = [1 - 1/(3L)]^L$ is the forgetting factor, $\mu$ is the step size, $\delta$ is the regularization parameter, and

$$\mathcal{W}_{2L \times L}^{01} = F_{2L} W_{2L \times 1}^{01} F_L^{-1} = 2 \left( \mathcal{W}_{L \times 2L}^{01} \right)^H$$

with $W_{2L \times L}^{01} = (W_{L \times 2L}^{01})^T$. To satisfy the unit-norm constraint [23] described in Section 2.4.4, the frequency-domain coefficients of the adaptive filter are initialized as
The Direct-Path Constraint

As mentioned in Section 2.4 and explained in [99, 100], the NMCFLMS algorithm still suffers from the misconvergence due to observation noise. To mitigate this problem, the direct-path constrained NMCFLMS algorithm was proposed in [99], where the estimated direct-path coefficient of each channel is forced, at each updating iteration, to match the actual direct-path coefficient. Denoting $h_{Dp,m}$ as the true direct-path coefficient of the $m$th channel, the direct-path constraint can be described by redefining the $m$th estimated impulse response for the $k$th frame as

$$
\hat{h}_m(k) = \left[\hat{h}_{m,0}(k) \hat{h}_{m,1}(k) \ldots h_{Dp,m} \ldots \hat{h}_{m,L-1}(k)\right]^T,
$$

where $\Delta \hat{h}_m(k) = [0 \ldots 0 h_{Dp,m} - \hat{h}_{Dp,m}(k) 0 \ldots 0]^T$ with the leading zeros representing a common bulk delay which is determined by the time delay of arrival (TDOA) from the speaker to the nearest microphone. Replacing $\hat{h}_m(k)$ in (6.9) and (6.11) by $\Delta \hat{h}_m(k)$ to obtain the following quantities

$$
\Delta \hat{h}_m(k) = F_L \Delta \hat{h}_m(k),
$$

$$
\Delta e_{ml}(k) = W_{01}^{01} [D_m(k) W_{2L \times 1}^{10} \Delta \hat{h}_l(k) - D_l(k) W_{2L \times 1}^{10} \Delta \hat{h}_m(k)],
$$

$$
\Delta e_{01}^{01}(k) = W_{01}^{01} \Delta e_{ml}(k),
$$

$$
\hat{h}_m^\dagger(k) = F_{2L} W_{2L \times 1}^{10} \hat{h}_m(k),
$$

the direct-path constrained NMCFLMS algorithm is given by (6.12) ∼ (6.14), (6.18) ∼ (6.21), and [99]

$$
\hat{h}_m^{10}(k) = \hat{h}_m^{10}(k - 1) - \mu [P_m(k) + \delta I_{2L \times 2L}]^{-1} \times \sum_{l=1}^{M} D_l^* (k) \left[e_{ml}^{01}(k) + \Delta e_{ml}^{01}(k)\right].
$$

In practice, the TDOA and $h_{Dp,m}$ can be estimated by using the generalized cross-correlation (GCC) method [158] such as proposed in [103], since the TDOA estimation
6.2 Multichannel Diversity and The Near-Common Zeros

Channel diversity is what makes multichannel system identification different from single-channel scenarios. Channel diversity in the context of multichannel signal processing implies that multiple channels would have no modes in common [14].

Recall the illustrative two-channel system shown in Fig. 4.1 which is reproduced in Fig. 6.2 if the zero separation $\Delta z$ defined in (4.5) reduces to 0, this system is equivalent to a single-channel system due to the presence of exactly-common zeros. It is important to note that for a SIMO system of size $L$ with no exactly-common zeros, the global channel matrix defined in (2.23b) and (6.2) is irreducible or of full column rank [93], i.e., $\text{Rank}(H) = ML - 1$. This indicates that the smallest singular value of $H$, denoted as $\lambda_{\text{min}}(H)$, is non-zero. When exactly-common zeros occur, $H$ becomes rank deficient and results in $\lambda_{\text{min}}(H) = 0$. However, the presence of NCZs does not violates channel coprimeness, thus cannot result in a rank-deficient $H$. Instead, they cause the system to be
### Table 6.1: The direct path constrained NMCFLMS Algorithm [23, 99]

<table>
<thead>
<tr>
<th>Special matrices</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{W}<em>0^{10}</em>{L \times 2L}$</td>
<td>$\begin{bmatrix} 0_{L \times L} &amp; \mathbf{I}_{L \times L} \end{bmatrix}$</td>
</tr>
<tr>
<td>$\mathbf{W}<em>2^{10}</em>{2L \times L}$</td>
<td>$\begin{bmatrix} \mathbf{I}<em>{L \times L} &amp; 0</em>{L \times L} \end{bmatrix}^T$</td>
</tr>
<tr>
<td>$\mathbf{W}<em>0^{01}</em>{2L \times L}$</td>
<td>$(\mathbf{W}<em>0^{01}</em>{L \times 2L})^T$</td>
</tr>
<tr>
<td>$\mathbf{W}<em>L^{01}</em>{L \times 2L}$</td>
<td>$\mathbf{F}_L \mathbf{W}<em>0^{01}</em>{L \times 2L} \mathbf{F}_L^{-1}$</td>
</tr>
<tr>
<td>$\mathbf{W}<em>2^{10}</em>{2L \times L}$</td>
<td>$\mathbf{F}_2L \mathbf{W}<em>2^{10}</em>{2L \times L} \mathbf{F}_L^{-1}$</td>
</tr>
<tr>
<td>$\mathbf{W}<em>2^{01}</em>{2L \times L}$</td>
<td>$2\left(\mathbf{W}<em>L^{01}</em>{L \times 2L}\right)^H$</td>
</tr>
</tbody>
</table>

**Initialization**

$0 < \mu \leq 1$

$\lambda = \left[1 - 1/(3L)\right]^L$

$\hat{\mathbf{h}}_{m}^{10}(0) = \frac{1}{\sqrt{M}} \mathbf{1}_{2L \times 1}, \quad m = 1, 2, \ldots, M$

**Algorithm**

$\chi_{m}(k) = \begin{bmatrix} x_{m}(kL-L) & x_{m}(kL-L+1) & \ldots & x_{m}(kL+L-1) \end{bmatrix}^T$

$\mathcal{D}_{m}(k) = \text{diag}\{\mathbf{F}_L \chi_{m}(k)\}$

$\mathbf{e}_{ml}(k) = \mathbf{W}_{L \times 2L}^{01} \mathbf{D}_{m}(k) \mathbf{W}_{2L \times L}^{10} \hat{\mathbf{h}}_{l}(k) - \mathbf{D}_{l}(k) \mathbf{W}_{2L \times L}^{10} \hat{\mathbf{h}}_{m}(k)$

$\Delta \mathbf{e}_{ml}(k) = \mathbf{W}_{L \times 2L}^{01} \left[ \mathbf{D}_{m}(k) \mathbf{W}_{2L \times L}^{10} \Delta \mathbf{h}_{l}(k) - \mathbf{D}_{l}(k) \mathbf{W}_{2L \times L}^{10} \Delta \mathbf{h}_{m}(k) \right]$
in a condition where multiple channels are not distinct enough from each other to enable adaptive BSI algorithms to produce an accurate estimation with rapid convergence. In addition, such condition causes $H$ to contain multiple singular values of which the values are within a small vicinity. This hence brings about numerical problems to determining the null space of the system correlation matrix for those eigenanalysis-based closed-form BSI algorithms such as the CR-based subspace algorithm as described in Section 2.4.4. Consequently, a binary indicator using, for example whether $H$ is rank deficient or not, is certainly insufficient to quantify the diversity of the multichannel system, denoted as $D(H)$, when NCZs are present.

Nevertheless, the inherent connection between channel coprimeness and the eigenstructure of the system can be exploited. Such connection can be described by recalling the EVD-based approach for blindly estimating the number of exactly-common zeros $N_c$ as discussed in Section 5.2, where the eigenvalue distribution of the autocorrelation matrix of the system outputs $R_x$ defined in (5.8) was utilized. Since it has been assumed in Section 2.4.2 that the autocorrelation matrix of the source signal $R_s$ is of full-rank, the rank deficiency of $R_x$ must be caused by $H$.

The quantified multichannel diversity, denoted by $D(H)$, can be measured by the minimal singular value of $H$ as proposed in [159]. To verify this in the presence of NCZs, consider again the example two-channel system shown in Fig. 6.2 which was introduced in Section 4.2. The variation of $D(H)$ against the zero separation $\Delta z$ is plotted in Fig. 6.3(a), where singular values of $H$ are computed using SVD [117]. As can be seen, $D(H)$ increases monotonically with an increasing $\Delta z$. When $\Delta z = 0$, the illustrative two-channel system is equivalent to a single-channel system which results in $D(H) = 0$ as expected. Figure 6.3(b) further depicts the variation of $D(H)$ against different $M$ over a set of simulated acoustic SIMO systems which were used to produce the result shown in Fig. 4.10. It can be clearly seen from the figure that $D(H)$ is also monotonically linked to $N_c$, which reduces with an increasing $M$ as was found in Fig. 4.8. To summarize, the results shown in Fig. 6.3 indicate that the common zeros problem can be interpreted using channel diversity quantified by the minimal singular value of global channel matrix $H$. 
6.3 The Concept of Forced Spectral Diversity (FSD)

As explained in Section 6.2, the presence of NCZs results in a lack of diversity for the multichannel system. However, the use of larger arrays with more microphones is practically limited. With the motivation of introducing extra diversity so as to improve BSI performance for the subsequent speech dereverberation, the concept of Forced Spectral Diversity (FSD) is developed in this section. To introduce this concept, channel undermodelling is first reviewed in Section 6.3.1. Section 6.3.2 and Section 6.3.3 then follows a description of how spectral diversifying filters are combined with channel undermodelling, using illustrative and numerical examples, to achieve the objective of introducing additional diversity.

6.3.1 Channel undermodelling

As a crucial building block of the FSD concept, channel undermodelling is now described. The channel undermodelling in the context of BSI was first introduced in [159], where it was stated that for an impulse response of length \( L \), the “significant” part of this impulse response of length \( L_s \) can be expressed in terms of \( h_m \) as

\[
h^L_{m} = h_m - d^L_{m},
\]

(6.23)

where \( h^L_{m} = [h_{m,0}, \ldots, h_{m,L_s-1}, 0, \ldots, 0]^T \) and \( d^L_{m} = [0, \ldots, 0, h_{m,L_s}, \ldots, h_{m,L-1}]^T \) for \( L_s < L \). Accordingly, \( h^L_{m} \) can be blindly estimated using \( L_s \)th-order least-squares (LS) method [17], provided that it is sufficiently diversified. The \( L_s \)th-order LS method is essentially equivalent to modelling the convolutional matrix \( H_m \) defined in (2.21) to be of dimension \( L_s \times (2L_s - 1) \) such that the correlation-like matrix \( R \) in (2.37) is of dimension \( ML_s \times ML_s \). Consequently, the \( ML_s \times 1 \) eigenvector associated with the smallest eigenvalue is the concatenated \( M \) channel estimates each of length \( L_s \times 1 \). This is to say, by employing LS-based BSI algorithm with an order of \( L_s \), an estimate of \( h^L_{m} \) can be obtained to some level of accuracy [159]. It is also worthwhile noting that in the context of BSI for speech dereverberation, \( d^L_{m} \) corresponds to the tail of the room impulse responses, and its effect on adaptive filtering algorithms can be found in [67].
6.3 The Concept of Forced Spectral Diversity (FSD)

6.3.2 Illustrative Examples

In [159], channel undermodelling was motivated by practical considerations since impulse responses of the microwave radio channels contain small leading and trailing coefficients. However, the intention of using channel undermodelling in this research is not to truncate the reverberation tail. Instead, it is employed in the $z$-domain utilizing (4.1) in combination with the use of spectral diversifying filters.

Consider initially a simple illustrative example comprising of a single-channel transfer function derived from the linear convolution of an impulse response $h = [h_n h_{n-1}]^T$ and a filter $f = [f_n f_{n-1}]^T$ such that

$$\overline{h} = h \oplus f.$$  \hspace{1cm} (6.24)

In the $z$-domain, (6.24) can be expressed as

$$\overline{H}(z) = H(z)F(z) = \sum_{n=0}^{2} \bar{h}_nz^{-n} = z^2 - (z_h + z_f)z + z_hz_f,$$  \hspace{1cm} (6.25)

where $H(z) = \sum_{i=0}^{1} h_nz^{-n} = z - z_h$, $F(z) = \sum_{h=0}^{1} f_nz^{-n} = z - z_f$ and $z_h, z_f \neq 0$. The undermodelling of this system can be performed by truncating the last coefficient of $\overline{H}(z)$ such that

$$H'(z) = \sum_{n=0}^{1} \bar{h}_nz^{-n} = z - (z_h + z_f) = z - z_h',$$  \hspace{1cm} (6.26)

where $H'(z)$ is equivalent to the 1st-order part of $\overline{H}(z)$.

The concept of FSD can be introduced by extending this illustrative example to an example of SIMO systems with two identical channels $H_1(z) = H_2(z) = H(z)$. Using the source signal $S(z)$ for a noiseless case, the output signals can be obtained $X_m(z) = S(z)H_m(z)$, $m = 1, 2$. Apparently, BSI algorithms are not expected to work successfully as the system functions $H_m(z)$ are not coprime to each other due to the common zero $z_h$. However, if we convolve the output signal $X_1(z)$ with an extra filter $F(z)$ similarly to (6.25),

$$\overline{X}_1(z) = X_1(z)F(z) = S(z)H_1(z)F(z) = S(z)\overline{H}_1(z)$$  \hspace{1cm} (6.27)

is obtained, where $\overline{H}_1(z) = H_1(z)F(z) = H(z)F(z)$. It is important to note that since
6.3 The Concept of Forced Spectral Diversity (FSD)

$H(z)$ and $F(z)$ are both of order 2, the resultant length of $\overline{H}_1(z)$ is 3 as indicated by (6.25). Hence, an $L$th-order BSI for $L = 2$ is employed on $X_1(z)$ and $X_2(z)$ to implement undermodelling, which in turn allows us to obtain the estimate of a modified system $H'_m(z)$ of order $L = 2$ with $H'_1(z) = H'(z)$ and $H'_2(z) = H_2(z)$. This is because, by imposing the order of $L = 2$, a modified system of order $L = 2$ is assumed by the BSI algorithm to undermodel $H$ as described in Section 6.3.1. Such undermodelling procedure can hence be interpreted by the derivation of (6.26) from (6.25), which suggests the resultant modified system to be $H'_m(z)$.

It is important to note that, compared to $H_m(z)$ the modified system $H'_m(z)$ now contains no common zeros since the distance between zeros has been increased from $|z_h - z_h'| = 0$ to $|z_m - z_h| = |z_f| > 0$, hence allowing a successful identification of both $H'_1(z)$ and $H_2(z)$. This example indicates that additional diversity brought about by the filter $F(z)$ is introduced inherently to the original system $H_m(z)$ through linear convolution given by $X_1(z)F(z)$ and the channel undermodelling subsequently leads to the estimation of $H'_m(z)$. Since such additional diversity can be quantified by $|z_m - z_h| = |z_f|$ which is similar to the zero separation as shown in Fig. 4.1, it can be expected that as long as $|z_f|$ is sufficiently large to prevent $z_h'$ and $z_h$ from becoming a pair of NCZs, the modified system $H'_m(z)$ can be accurately identified.

It is worthwhile noting that in some cases such as described in [159], undermodelling can be sufficient for increasing channel diversity without introducing $F(z)$ as long as the resultant system is sufficiently diversified. Utilizing $H(z)$, $F(z)$ and $H'(z)$ defined in (6.25) and (6.26), this situation can be verified by considering another illustrative example for a SIMO system

\[
\begin{align*}
H_1(z) &= F(z)H(z), \\
H_2(z) &= F(z)H'(z),
\end{align*}
\]

where $z_f$ is the common zero. Implementing undermodelling in (6.26) for both channels results in a modified system with zeros of each channel being $z_1' = z_h + z_f$ and $z_2' = z_f + z_m'$ respectively. It can then be seen that the modified system $H'_m(z)$ does not contain common zeros as long as $|z'_2 - z'_1| = |z_m' - z_h| = |z_f|$ is sufficiently large. However, this is not necessarily the case in practice, especially for acoustic impulse responses, where zeros...
6.3 The Concept of Forced Spectral Diversity (FSD) of acoustic systems tend to cluster around the unit circle with high density\[149\], and zeros like \( z_h \) and \( z_{h'} \) for different channels can rarely exist. As a result, \( F(z) \) is important and not negligible. In addition, since the benefit of introducing \( F(z) \) in combination with undermodelling is that the zeros in the modified system are located differently to those in the original system, adding additional channel diversity, up to \( M \) such filters can be employed for an \( M \)-channel system. Referring to \( F(z) \) as spectral diversifying filter, the FSD processing is comprised in summary of the use of spectral diversifying filter and channel undermodelling.

6.3.3 Numerical Results

Numerical results are now presented using simple SIMO systems with common zeros to demonstrate further how additional diversity can be introduced by FSD processing, from which important characteristics of the FSD concept can be summarized. For clarity of presentation, impulse responses with limited number of coefficients are used.

Consider \( h = [h_1^T \ h_2^T]^T \) as an example two-channel SIMO system with real coefficients of length \( L = 5 \). As before, the case where \( h_1 = h_2 \) is assumed, which implies that common zeros occur between the channels. Let \( z_1(p) \) and \( z_2(q) \) be the zeros of channel 1 and 2 respectively with \( p, q = 1, \ldots, 4 \). Figure 6.4(a) shows these zeros where the circles and crosses denote the zeros of each channel respectively. Only the upper half of the \( z \)-plane is shown since zeros appear as complex conjugates for systems with real coefficients. A set of spectral diversifying filters \( f_m \) with two zeros defined as \( z_{f,m} = [z_{f,m}(1) \ z_{f,m}(2)] \) for \( m = 1, 2 \) and \( z_{f,m}(1) = z_{f,m}^*(2) \) are then obtained by placing \( z_{f,m} \) in the \( z \)-plane so as to clearly show how they affect the resultant modified channels, denoted by \( h' \), in terms of their zeros. This is achieved by varying the magnitude \( \rho_1 = |z_{f,1}(1)| = |z_{f,1}(2)| \) from 0.05 to 1 and keeping \( \rho_2 = |z_{f,2}(1)| = |z_{f,2}(2)| \) fixed such as shown in Fig. 6.4(b) for the cases of \( \rho_1 = 0.35, 0.5, 0.75, \) and 0.9. Denoting the zeros of the modified system \( h' \) as \( z'_1(p) \) for the first channel and \( z'_2(q) \) for the second channel \( p, q = 1, \ldots, 4 \), Fig. 6.4(c) shows the resultant \( z'_m(1) \) on the left hand side of the \( z \)-plane and \( z'_m(2) \) on the right hand side respectively for \( m = 1, 2 \). Comparing Fig. 6.4(c) with Fig. 6.4(a), it can be seen that by introducing the spectral diversifying filters \( f_1 \), the zero separation \( \Delta z'(1) = |z'_1(1) - z'_2(1)| \) for the two modified channels in-
6.3 The Concept of Forced Spectral Diversity (FSD)

Figure 6.4: Zeros of the illustrative example of (a) the original system $h$, and that from the (b) convolution with spectral diversifying filters $z_{f,m}(p)$ with various $\rho_1$, and (c) the zeros of the resultant modified system $h'$.

... increases significantly with an increasing $\rho_1$. This is because $z'_1(1)$ has been affected by $z_{f,1}(1)$ and locates further away from $z'_2(1)$, which is due to by channel undermodelling in FSD processing, hence indicating an increment in channel diversity of $h'$ compared to $h$. It can also be seen from Fig. 6.4(c) that as $\rho_1$ increases, the effect of $z_{f,1}(1)$ becomes more significant as $\Delta z'(1)$ becomes larger. In contrast, it can be seen that the position of $z'_2 = [z'_2(1) \ldots z'_2(4)],$ as denoted by squares in Fig. 6.4(c), is not significantly changed in comparison to $z_2 = [z_2(1) \ldots z_2(4)],$ which indicates a less significant effect by $z_{f,2}$. Moreover, it is seen that the change in $\Delta z'(2) = |z'_1(2) - z'_2(2)|$ is relatively smaller than $\Delta z'(1)$. Such difference is clearly reflected by the relative locations between $z_{f,1}(1)$ and $z_1(1)$, and between $z_{f,1}(1)$ and $z_1(2)$.

In Fig. 6.5 the effect of FSD processing is further demonstrated using various $\rho_1$. In Fig. 6.5(a), the mean zero separation $\Delta z' = (\|z'_1 - z'_2\|_1)/4$ is plotted against $\rho_1$. As can be seen, $\Delta z'$ increases with $\rho_1$ as $z_{f,1}$ approaches the unit circle. The corresponding variation of channel diversity $\mathcal{D}(H)$ defined in Section 6.2 is then plotted in Fig. 6.5(b), which agrees with the result shown in Fig. 6.3(a) implying a positive correlation between zero separation and channel diversity. Moreover, a sequence of WGN with 50 dB SNR to is used to excite the resultant modified system $h'$ due to various $\rho_1$ so as to simulate the...
BSI performance in terms of NPM \[ \eta' = 20 \log_{10} \left( \frac{1}{\|h\|_2} \| h - \kappa h \|_2 \right) \text{ dB}, \] (6.30)

where \( \kappa = (h^T \hat{h})/(\hat{h}^T \hat{h}) \) is the projection misalignment vector. The dependency of \( \hat{h} \) on the time-domain sample index \( n \) is omitted here for simplicity of presentation since the closed-form subspace algorithm \[19\] is employed. Figure 6.5(c) shows the variation of BSI performance in NPM against \( \rho_1 \), from which it can be seen that an increasing \( \rho_1 \) leads to an improving BSI performance for \( h' \) due to the increasing zero separation and channel diversity. As will be demonstrated in Section 6.6, this can be found beneficiary for speech dereverberation.

In summary, the FSD processing involves two important components: (i) spectral diversifying filters to provide extra zeros, and then (ii) channel undermodelling of the system with these extra zeros gives rise to additional diversity compared to the original system. It should be noted that the zero locations for the modified system are determined by that of both spectral diversifying filters and the original system. This is insightful regarding the design of the spectral diversifying filters as will be presented in Section 6.5.
6.4 FSD Processing for SIMO acoustic systems

The FSD concept is now applied to the problem of blind identification of SIMO acoustic systems for speech dereverberation, where Fig. 6.6 depicts the schematic of an $M$-channel SIMO system with FSD processing. As can be seen, the $m$th microphone signal $x_m(n)$ is filtered by the corresponding spectral diversifying filter giving rise to

$$\bar{x}_m(n) = \mathcal{F}_m^T x_m(n) = \mathcal{F}_m^T [H_m s(n) + b_m(n)], \quad (6.31)$$

where

$$\mathcal{F}_m = \begin{bmatrix} f_{m,0} & \cdots & f_{m,L_p-1} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & f_{m,0} & \cdots & f_{m,L_p-1} \end{bmatrix}, \quad (6.32)$$

is the $L \times (L + L_p - 1)$ convolutional matrix for the $m$th spectral diversifying filter with $L_p$ FIR coefficients such that $f = [f_{m,0}, f_{m,1}, \ldots, f_{m,L_p-1}]^T$. According to Fig. 6.6, $\bar{x}_m(n)$ can be considered as the linear convolution between $s(n)$ and an equivalent SIMO system of length $L + L_p - 1$ such that (6.31) can be rewritten

$$\bar{x}_m(n) = \bar{H}_m s(n) + \bar{b}_m(n), \quad (6.33)$$

where $\bar{H}_m$ is the convolutional matrix for $\bar{H}_m = \mathcal{F}_m^T h_m$ and $\bar{b}_m(n) = \mathcal{F}_m^T b_m(n)$. The effective undermodelling is then implemented by employing an $L$th-order BSI process-
ing over \( \bar{x}_m(n) \). This is reasonable since in practice the BSI algorithms are equivalently “blind” to the existence of spectral diversifying filters. Similar to (6.23), the modified system to be identified can be written

\[
h'_m = \bar{h}_m - \bar{d}_m^L, \quad m = 1, 2, \ldots, M, \quad (6.34)
\]

where \( h'_m = [\bar{h}_{m,0} \ldots \bar{h}_{m,L-1} 0 \ldots 0]^T \) and \( \bar{d}_m^L = [0 \ldots 0 \bar{h}_{m,L} \ldots \bar{h}_{m,L+L_p-1}]^T \). Since the BSI algorithm operates with an order of \( L \), the undermodelling factor is determined by the length of spectral diversifying filter \( L_p \).

Utilizing FSD, the modified system \( h'_m \) is expected to be more diversified with improved identifiability. As shown in Fig. 6.6, an estimate of \( s(n) \) can be obtained for speech dereverberation without the need to deconvolve the procedures in (6.31), but by inverting \( \hat{h}_m' \) using, for example, the MINT algorithm [124] such that

\[
\hat{s}'(n) = \sum_{m=1}^{M} \hat{G}_{eq,m}^T \hat{x}_m(n), \quad (6.35)
\]

where \( \hat{G}_{eq,m} \) is the convolutional matrix of \( \hat{g}_m' \) of length \( L_g \) given by

\[
\hat{g}' = \hat{H}'_{eq}^+[0^T_{\tau \times 1} \theta 0^T_{(L-\tau-1)\times 1}]^T, \quad (6.36)
\]

for \( \hat{g}' = [\hat{g}_1' \hat{g}_2' \ldots \hat{g}_M']^T, \hat{H}'_{eq} = [\hat{H}'_{eq,1} \hat{H}'_{eq,2} \ldots \hat{H}'_{eq,M}], \theta \in \mathbb{Z}^+, L = L + L_g - 1, \) where \( \tau \) is the modelling delay, \( \hat{H}'_{eq,m} \) is a \( L_g \times L \) convolutional matrix of \( \hat{h}_m' \), and \( L_g \geq \lceil (L - 1)/(M - 1) \rceil \). It is foreseeable that the accuracy of such inversion process can be limited due to noise amplification when the spectral diversifying filters have zeros close to the unit circle. In addition, approximation errors can be introduced to (6.35) since \( \bar{x}_m(n) \) is given by the convolution between \( s(n) \) and \( \bar{h}_m' \), rather than \( h'_m \). Such error is thus caused by \( \bar{d}_m^L \) in (6.34) not being estimated by BSI algorithms with an order of \( L \). However, it can be found from simulation results in Section 6.6 that such limitations are outweighed by the advantages obtained in the BSI due to the increased channel diversity. Also, it is reasonable to assume \( L_p \ll L \), so that the aforementioned approximation error can be negligible. Now, since \( \hat{H}'_{eq} \) is expected to contain fewer NCZ clusters, the accuracy of implementing (6.36) can be expected to increase which subsequently results in an overall improvement in the ability to equalize channels for speech dereverberation.
6.5 Design of Spectral Diversifying Filters

Based on the discussions in Section 6.3.2 and Section 6.3.3, the criteria for designing spectral diversifying filters $f_m$ can now be summarized:

- It is preferable to use $f_m$ containing zeros with large magnitude (close to unity) as indicated by Fig. 6.4, although this may limit the equalization performance as described in Section 6.4.

- Zeros of $f_m$ need be sufficiently diversified with respect to the zeros in the original system.

In addition, it would be reasonable to place zeros of $f_m$ in such a way that their effect to the original zeros is complementary in terms of increasing channel diversity so as to enhance the benefits brought about by the FSD processing. As a result, the design of spectral diversifying filters is not trivial and can vary with respect to specific systems.

It has been shown in Chapter 4 that the presence of NCZs in SIMO acoustic systems is due to the long impulse responses that give rise to a high density of zeros around the unit circle. Utilizing this property, a set of FIR causal filters with overlapping frequency responses are employed as the spectral diversifying filters for FSD processing in this chapter such that the $m$th filter has a passband given by

$$
\Omega_m^n = \left\{ \omega : \frac{\pi(m-1)}{M} - \epsilon \leq \omega < \frac{\pi m}{M} + \epsilon \right\}
$$

(6.37)

for $m = 1, 2, \ldots, M$ where $\epsilon > 0$ denotes the transition bandwidth.

The effectiveness of such design can be demonstrated by a two-channel case where a pair of FIR causal highpass and lowpass filters are designed with their magnitude responses and zeros, $z_{f,1}$ and $z_{f,2}$, shown in Fig. 6.7(a) and Fig. 6.7(b) respectively. As can be seen, the zeros in $z_{f,1}$ and $z_{f,2}$ within the stopband of the filters are close to the unit circle. They are therefore very effective in terms of relocating the zeros of the modified system within the passband of the filters to be further away from the unit circle. Such effect can also be strengthened by the remaining zeros in $z_{f,1}$ and $z_{f,2}$ locating on the passband of the filters. In addition, the complimentary location of $z_{f,1}$ and $z_{f,2}$ avoids the FSD effect from $f_1$ and $f_2$ being compensated by each other. Such design also allows a simple imple-
6.6 Simulations

Simulation results are now presented to evaluate the performance of the developed FSD processing for BSI and speech dereverberation, where simulated and recorded SIMO acoustic systems are used and excited by both WGN and speech input. Since the purpose here is to show how the developed algorithm improves the BSI performance in the presence of NCZs, most of the simulations are carried out for two-channel cases, where the NCZs are most likely to occur.

6.6.1 Setup

Figure 6.8 depicts the layout for a room of dimension $10 \times 10 \times 3$ m with a reflection coefficient of 0.6869 in which a system of simulated impulse responses are obtained using the method of images [66]. The configuration of an array of 16 microphones and a speaker is also shown in the figure. The sampling frequency and reverberation time are 8 kHz and $T_{60} = 200$ ms, and the generated impulse responses are truncated to 512 FIR coefficients. In addition, recorded acoustic impulse responses are extracted from the MARDY
database [130] with the speaker placed 3 m in front of an array of 8 microphones with 5 cm spacing. These impulse responses are also resampled at 8 kHz and truncated to 1024 coefficients. It is important to note that for both of the acoustic systems, an overall bulk delay is present and determined by the TDOA of the microphone being closest to the speaker. Since the filter coefficients corresponding to this delay are essentially zero-valued and has no influence on the BSI problem [23][99], the generated acoustic impulse responses are aligned such that the channel with the smallest TDOA has no leading zeros before the direct-path component. As described in Section 6.1.2, the TDOA with respected to the microphone array can be estimated using the GCC algorithm [158].

The direct-path constrained NMCFLMS algorithm [99] reviewed in Section 6.1 is employed as the baseline algorithm for all simulations, where the BSI performance is evaluated using NPM $\eta'(n)$ defined in (2.27), and BSD score defined in (2.59) is employed to measure the dereverberation performance.

### 6.6.2 Blind System Identification with FSD Processing

The improvement in BSI performance brought about by employing FSD on both simulated and recorded SIMO acoustic systems is first shown. Using simulated acoustic systems with two channels, Fig. 6.9 compares the number of NCZ clusters $N_c$ in the original system and that of the modified system derived by (6.34) against pairwise tolerance $\delta_c$. The result is obtained by averaging the computed $N_c$ over various systems simulated
at 20 different positions in the room with a fixed speaker-microphone configuration. As can be seen from the figure, FSD processing results in a modified system containing approximately half of the number of NCZs compared to the original system.

Figure 6.10 then shows the comparative results for the BSI performance in terms of NPM using recorded SIMO acoustic systems obtained from the MARDY database [130], where two SIMO systems are extracted such that the first one corresponds to the first and the eighth microphone for $M = 2$ and the second one corresponds to the second to sixth microphone for $M = 5$. As before, the NMCFLMS algorithm is employed as the baseline algorithm with a step-size of 0.1 and an SNR of 50 dB. Figure 6.10(a) shows the result obtained using a sequence of WGN as the excitation signal, where the corresponding NPM values are computed between the estimates of the modified system $\hat{h}_m'$ and the true one as derived in (6.34). Alternatively, $\tilde{d}_m^l$ can be approximated by a vector with small values (since the reverberation tail of the room impulse response is often of small magnitude as shown in Fig. 2.7) such that NPM values can be computed between $\tilde{h}_m$ and its estimates. However, it has been found from experiments that the difference in NPM between these two evaluation methods is small and can be negligible, which is expected since $L_p \ll L$. 

Figure 6.10: Variation of BSI performance for SIMO acoustic system with and without FSD processing for recorded impulse responses from the MARDY database using (a) WGN input, and (b) speech input.
6.6 Simulations

As can be seen from Fig. 6.10(a), the FSD processing results in an approximately 6 dB gain in NPM over that of original system for the case of $M = 2$. This also outperforms the case of $M = 5$ without FSD by approximately 5.5 dB in NPM. It should be noted that the intention of plotting these convergence curves in the same figure is not to compare the performance of NMCFLMS algorithm, but to demonstrate that the use of FSD processing gives rise to a modified system that can be identified more accurately by classic BSI algorithm such that the subsequent speech dereverberation can be achieved with an improved performance. Using the same simulation parameters, Fig. 6.10(b) then shows the result obtained by using a speech sample obtained by concatenating both a male and female utterance obtained from the APLAWD database [68] and down-sampled at 8 kHz. The time sequence and spectrogram for the first 6 seconds of this speech sample is shown in Fig. 6.11 for clarity of presentation. As can be seen from Fig. 6.10(b), a 6.5 dB gain in NPM is obtained by employing FSD processing for the case of $M = 2$; and this result is significantly better than the case of $M = 5$ without FSD by about 4.5 dB. This indicates that with the use of FSD, improved BSI performance in NPM can be achieved using only $M = 2$ microphones as opposed to using $M = 5$ microphones which is usually assumed. The FSD concept is hence attractive in terms of reducing computational complexity.
6.6.3 Application to Speech Dereverberation

Last but not least, Fig. 6.12 shows the results for speech dereverberation using the same two-channel recorded acoustic system, by which the reverberant speech is obtained as shown in Fig. 6.12(a). Compared with the original clean speech, the BSD score for the reverberant speech is found to be 0.0785. With the improved BSI accuracy due to FSD as shown in Fig. 6.10(b), the MINT [124] algorithm can now be employed to equalize the estimated impulse response [43]. Figure 6.12(b) shows results obtained using NMCFLMS and MINT for speech dereverberation. Employing these algorithms, the BSD score of the dereverberated speech, as shown in Fig. 6.12(b), is found to be higher than that for the reverberant speech by about 0.0237. This is due to spectral distortion at around 3 kHz that is caused by the limited BSI and equalization performances in the presence of NCZs. Figure 6.12(c) shows the spectrogram of dereverberated speech using the FSD processing. In this case, the MINT algorithm is employed to equalize the estimated impulse responses of the modified system. As can be seen, a better dereverberation can be achieved since the reverberation effect in the low-frequency range is removed more effectively than that as shown in Fig. 6.12(b) using the NMCFLMS algorithm without the FSD processing. In addition, the spectral distortion at around 3 kHz as seen in Fig. 6.12(b) is also removed.
as shown in Fig. 6.12(c). These improvements are reflected by a much smaller BSD score of 0.0219 computed from the spectrogram as shown in Fig. 6.12(c). Compared to that for the reverberant speech as well as the dereverberated speech using NMCFLMS, the BSD score for the dereverberated speech using the FSD processing is significantly lower.

6.7 Summary

In this chapter, the concept of forced spectral diversity (FSD) has been developed to mitigate the detrimental effect of NCZs on the performance of BSI and the subsequent speech dereverberation, which has been shown to correlate with multichannel diversity. With the aim to achieve better identifiability, the developed FSD processing collectively combines the use of spectral diversifying filters and effective channel undermodelling so as to derive a modified system from the original system with additional diversity. The inversion of the modified system is sufficient to recover the clean speech signal for speech dereverberation. Using an effective example of spectral diversifying filters, a number of simulations over recorded and simulated SIMO acoustic systems were carried out. For the two-channel cases, FSD processing results in a significant performance improvement of up to 6.5 dB in NPM, which in turn gives rise to an improved speech dereverberation performance. The ability of two-channel FSD processing of achieving a 4.5 dB performance gain in NPM over classic methods with five channels demonstrated its computational attractiveness.
Chapter 7

Scale Ambiguity Correction for Subband-based Blind System Identification

In blind identification of multichannel acoustic systems, the impulse responses involved generally contain a large number of FIR coefficients. As described in Section 2.4.5 and Section 4.2, long impulse responses are computationally demanding and are likely to give rise to common zeros. The accuracy of BSI for speech dereverberation is therefore limited. As a result, it is desirable to partition long impulse responses into parts with reduced length so as to allow successful channel identification. One promising approach is to perform multirate signal processing in decimated subbands employing filter banks [24].

In this chapter, a preliminary but important work on subband-based BSI is presented. It is motivated by the scale factor ambiguity problem that significantly limits the deployment of BSI in the context of multirate signal processing. The work in this chapter therefore aims to correct such ambiguity so as to resolve a key obstacle for subband-based BSI and speech dereverberation. This chapter is organized as follows: In Section 7.1, the motivation of developing subband-based system identification techniques is presented. Section 7.2 and Section 7.3 then review the generalized DFT (GDFT) filter bank and the complex subband decomposition which combinatorially establish the rela-
tionship between the full-band system and the subband system, hence allowing the deployment of blind multichannel identification in subbands. In Section 7.4, the problem of scale factor ambiguity is addressed. To resolve this problem, the full-band and subband cross-relation (CR) error functions are combined such that an overall cost function can be derived to estimate ambiguity correction parameters. Simulation results presented in Section 7.5 demonstrate the effectiveness of the developed approach and its application to speech dereverberation. This chapter is summarized in Section 7.6.

7.1 Introduction to Subband-based Blind System Identification

The techniques of multirate signal processing were derived from classic filter bank theory [160]. Although its applications were largely limited to areas such as data compression as early work did not support signal processing in the subbands [161], it has been recently extended through the use of subband adaptive filters on, for example, adaptive beamforming [162], supervised system identification for AEC [24, 26], channel equalization [126] and speech dereverberation [163]. These have been motivated by the fact that subband processing can render computationally intractable problems feasible by decomposing the full-band system into a number of subbands with much shorter length due to decimation in the filter bank structure. In addition, a properly designed filter bank can allow an overall reduction of computational complexity, even though extra computation is required for forming the subband signals. In [164], the optimization of subband design parameters was studied to quantify the advantages of using subband-based method in supervised adaptive system identification for AEC.

Consequently, it has been found promising for BSI to be deployed in the context of subband structure where each subband system with reduced length can be estimated more accurately such as highlighted in [19]. In [25], the CR equation was derived in subbands and it was shown that such relation holds in subbands irrespective of the reconstruction properties of the filter bank. In addition, with successful development of subband-based channel equalization algorithm [126], it is expected that a full dereverberation system can be implemented in subbands to achieve significant complexity reduction and performance improvement. However, multichannel BSI in subbands has received much less attention than the non-blind case such as AEC. One of the main rea-
7.2 The Oversampled GDFT Filter Bank

A well-designed filter bank is crucial for multirate systems since it performs signal decompositions and determines the accuracy of reconstructing full-band signals from their subband versions [24]. Both critically-sampled filter bank and oversampled filter bank have been applied to adaptive system identification in the context of AEC [24]. Denoting $K$ as the number of subband and $N$ as the decimation ratio, critically-sampled filter bank indicates that $N = K$ while oversampled filter bank corresponds to $N < K$. In this research, the latter case is considered due to its computational efficiency and straightforward implementation.

The GDFT filter bank is now reviewed. Consider a GDFT filter bank with $K$ analysis and synthesis filters. For the $k$th analysis filter with bandwidth $2\pi / K$, its $n$th coefficient $u_{k,n}$ can be derived from a prototype lowpass filter of length $L_{pr}$ such that

$$u_{k,n} = p_n e^{j2\pi (k+k_0)(n+n_0)}, \quad n = 0, 1, \ldots, L_{pr} - 1,$$

where $p_n$ is the $n$th coefficient of the prototype filter, and the time and frequency offset terms are set $n_0 = 0$ and $k_0 = 1/2$ as in [24, 26]. The $k$th synthesis filter satisfying near perfect reconstruction [161] can be obtained accordingly from the time-reversed and conjugated version of the analysis filter [24],

$$v_{k,n} = u_{k,L_{pr}-n-1}^*, \quad n = 0, 1, \ldots, L_{pr} - 1.$$
In $z$-domain, (7.1) and (7.2) can be respectively rewritten

$$U_k(z) = \mathcal{P} \left( z W_K^{-((k+1)/2)} \right),$$  \hfill (7.3)

$$V_k(z) = \mathcal{P} \left( z^{-1} W_K^{-(k+1/2)} \right)$$ \hfill (7.4)

for $W_K = e^{-j2\pi/K}$, where $U_k(z)$, $V_k(z)$ and $\mathcal{P}(z)$ are the $z$-transforms of the $k$th analysis, synthesis and prototype filter respectively. It is important to note that such design results in complex subband signals, and that only $K/2$ subbands need to be processed for real input signals since the rest are complex conjugates of them. According to [24,26,126], the following assumptions are considered valid throughout this chapter:

(A7.1) Aliasing can be sufficiently suppressed in the subbands such that

$$U_k(z W_N^i) V_k(z) \approx 0, \quad i = 1, 2, \ldots, N - 1, \quad k = 0, 1, \ldots, K/2 - 1; \quad (7.5)$$

(A7.2) Magnitude distortion of the filter bank is negligible

$$\sum_{k=0}^{K/2-1} U_k(z) V_k(z) \approx \gamma z^{-\varphi}, \quad (7.6)$$

where $\gamma \in \mathbb{Z}^+$, and $\varphi$ is a delay introduced for causality [26].

According to [26], (7.5) can be satisfied by making $N$ not too large with respect to $K$; while (7.6) can be validated by making $\mathcal{P}(z)$ a root-Nyquist filter. In addition, these two approximations can be achieved with sufficient accuracy using analysis and synthesis filters with modest length [26]. Figure 7.1 shows an illustrative example of GDF filter bank with $K = 16$ and $N = 12$, where a $L_{pr} = 256$ prototype filter is designed using the iterative LS method [24], giving an estimated aliasing suppression of 79 dB.

### 7.3 Complex Subband Decomposition

Based on the filter bank structure, the relationship between a full-band system and the corresponding subband systems can be established using the complex subband decomposition [26]. Given the $m$th full-band channel $H_m(z)$ of length $L$ in an $M$-channel SIMO
7.3 Complex Subband Decomposition

Figure 7.1: Magnitude response of the example GDFT analysis filters with \[ K = 16, N = 12, L_p = 256 \].

![Magnitude response of the example GDFT analysis filters](image)

\[ H_{mk}(z), \quad \forall m, k \text{ such that the overall transfer function of the filter bank } \hat{H}_m(z), \text{ is equivalent to } H_m(z) \text{ up to an arbitrary scale factor } \gamma, \text{ and an arbitrary delay } \varphi, \text{ i.e.,} \]

\[
\hat{H}_m(z) = \gamma z^{-\varphi} H_m(z), \quad m = 1, 2, \ldots, M. \tag{7.7}
\]

Figure 7.2 depicts such relationship between the full-band and subband channels for real-valued input signals, from which \( \hat{H}_m(z) \) can be expressed as

\[
\hat{H}_m(z) = \frac{1}{N} \sum_{k=0}^{K/2-1} \sum_{i=0}^{N-1} U_k(z) W_k^i H_{mk}(z^N) V_k(z). \tag{7.8}
\]
With reference to the assumption (A7.1) in (7.5), \( \hat{H}_m(z) \) reduces to

\[
\hat{H}_m(z) \approx \frac{1}{N} \sum_{k=0}^{K/2-1} U_k(z) H_{mk}(z^N) V_k(z),
\]

(7.9)

which indicates that by suppressing aliasing between adjacent subbands to a sufficiently low level, the full-band channel \( H_m(z) \) can be related to a single subband channel per subband. \( H_{mk}(z) \) thus form the \( k \)th \( M \)-channel subband system for \( m = 1, 2, \ldots, M \).

Now, if each subband channel \( H_{mk}(z) \) is chosen to satisfy the relation

\[
U_k(z) H_{mk}(z^N) = U_k(z) H_m(z), \quad \forall k,
\]

(7.10)

\( \hat{H}_m(z) \) in (7.9) can be derived as

\[
\hat{H}_m(z) \approx H_m(z) \frac{1}{N} \sum_{k=0}^{K/2-1} U_k(z) V_k(z).
\]

(7.11)

Invoking the assumption (A7.2) in (7.6), (7.11) becomes

\[
\hat{H}_m(z) \approx \frac{\gamma}{N} z^{-\phi} H_m(z),
\]

(7.12)

which is desired by (7.7). In order to find \( H_{mk}(z) \) that satisfies (7.12), the following approximation is used, where (7.10) is decimated by a factor of \( N \),

\[
\frac{1}{N} \sum_{i=0}^{N-1} U_k(z^{1/N} W_N^i) H_{mk}(z) \approx \frac{1}{N} \sum_{i=0}^{N-1} U_k(z^{1/N} W_N^i) H_m(z^{1/N} W_N^i).
\]

(7.13)

In vector form, (7.13) can be rewritten

\[
U_N h_{mk} \approx r_{mk},
\]

(7.14)

where \( h_{mk} = [h_{mk,0} \ h_{mk,1} \ \ldots \ h_{mk,L_{sub}-1}]^T \) denotes the impulse response of the \( k \)th subband channel of length \( L_{sub} \), \( r_{mk} = [r_{mk,0} \ r_{mk,N} \ \ldots \ r_{mk,N(L-1)}]^T \) is a \(([(L + L_{pr} - 1)/N]) \times 1\) vector with the \( n \)th element being given by \( r_{mk,n} = (h_{m,n} \circ u_k,n)|_N \), \((\cdot)|_N\) denoting downs-
amplifying by a factor of $N$, and

$$
U_{N,k} = \begin{bmatrix}
    u_{k,0} & u_{k,N} & \cdots & u_{k,L_{pr}-1} & 0 \\
    \vdots & \ddots & \ddots & \vdots & \vdots \\
    0 & \cdots & u_{k,0} & u_{k,N} & \cdots & u_{k,L_{pr}-1}
\end{bmatrix}^T
$$

(7.15)

is the $(L_{\text{sub}} + \lceil L_{pr} / N \rceil - 1) \times L_{\text{sub}}$ convolutional matrix for the $k$th down-sampled analysis filter. Since the left hand side of (7.14) results in a vector of length $L_{\text{sub}} + \lceil L_{pr} / N \rceil - 1$ and $r_{mk}$ is of length $\lceil (L + L_{pr} - 1) / N \rceil$, the length of the subband filter $L_{\text{sub}}$ can be determined by

$$
L_{\text{sub}} = \left\lceil \frac{L + L_{pr} - 1}{N} \right\rceil - \left\lceil \frac{L_{pr}}{N} \right\rceil + 1.
$$

(7.16)

As a result, the solution to $h_{mk}$ can be obtained by solving

$$
\hat{h}_{mk} = \arg \min_{h_{mk}} \| U_{N,k} h_{mk} - r_{mk} \|^2_2
$$

(7.17)

in, for example, an LS sense \[26\] such that

$$
\hat{h}_{mk} = \left( U_{N,k}^T U_{N,k} \right)^{-1} U_{N,k}^T r_{mk}.
$$

(7.18)

Inversely, the full-band estimate for channel $m$ can be obtained from a set of subband estimates using the relation

$$
\hat{h}_m = \Re \left\{ \sum_{k=0}^{K/2-1} \tilde{q}_{mk} \right\},
$$

(7.19)

where the $n$th element of $\tilde{q}_{mk} = [\tilde{q}_{mk,0}, \tilde{q}_{mk,1}, \ldots, \tilde{q}_{mk,L_{pr}-1}]^T$ is given by

$$
\tilde{q}_{mk,n} = \left( (u_{k,n})_{\lceil N \rceil} \centerdot \hat{h}_{mk,n} \right)_{\lceil N \rceil} \centerdot v_{k,n},
$$

(7.20)

with $\Re \{ \cdot \}$ and $(\cdot)_{\lceil N \rceil}$ denoting respectively the real part of a complex variable and up-sampling by a factor of $N$. In summary, the complex subband decomposition maps an $M$-channel full-band SIMO system to $M$ subband systems each with $K/2$ complex-valued subband channels of length $L_{\text{sub}}$. Since $L_{\text{sub}} < L$ as indicated by (7.16), it is expected that these subband SIMO systems can be blindly identified with improved accuracy compared to their full-band counterpart.
7.4 Correction for the Scale Factor Ambiguity

In this section, the scale factor ambiguity problem in subband BSI is addressed and then resolved for the situations studied. As mentioned in [25], the derivation of CR error in subbands indicates that there exists a set of subband channels, which, if accurately identified, also minimize the full-band CR error. This implies that classic BSI algorithms can be applied in each subband to identify $h_{mk}$ and a full-band channel estimate satisfying CR can then be reconstructed by using (7.19). This is beneficial since the order of equivalent subband filters is reduced approximately by a factor of $N$ as indicated by (7.16), which can give rise to fewer common zeros as described in Chapter 4. In addition, decimation can lead to flatter signal spectra in each subband, hence enabling an improved BSI performance across all frequency range [19].

The deployment of CR-based BSI algorithms in subbands can be implemented by first defining the concatenated channel impulse response for the $k$th subband as $h_k = [h_{1k}^T \ h_{2k}^T \ldots \ h_{MK}^T]^T$ and then invoking (2.38) in this subband, i.e.,

$$R_k h_k = e_k, \quad \forall k,$$  \hspace{1cm} (7.21)

where $R_k$, similar to that in (2.37), is the correlation-like matrix for the $k$th subband microphone signal. Now, if the channel identifiability conditions as described in Section 2.4.2 are satisfied, each subband estimate $\hat{h}_k$ can be found up to a complex scale factor $\alpha_k$ by employing classic BSI algorithms such as the subspace algorithm [19] and complex MCLMS algorithm [165] such that

$$\hat{h}_k = \alpha_k h_k, \quad \forall k.$$  \hspace{1cm} (7.22)

It is evident from (7.22) that $\alpha_k$ will be the same within a subband but will be different across the $K/2$ subbands\footnote{Only $K/2$ subbands are considered since the rest are complex conjugates of them as described in Section 7.2.}. If such subband estimates are used to reconstruct the full-band channel estimate by invoking (7.19) and (7.20), the scale discrepancy will result in a scaled $\hat{q}_{mk}$ and consequently an erroneous full-band estimate when summing all $K/2$ subbands. In addition, since $\alpha_k$ is complex, an arbitrary phase ambiguity is introduced to $\hat{h}_k$ making it difficult to, for example, be equalized as such phase error would propagate...
7.4 Correction for the Scale Factor Ambiguity

One way to solve such scale ambiguity is to assume that the true amplitude of the first tap of $h_{mk}$ is available \[25\], which is similar to the direct-path constraint described in Section 6.1.2. By enforcing this as a constraint in each updating iteration, correct value of $\alpha_k$ can be computed to alleviate the ambiguity problem. In this thesis, the relationship between subband and full-band CR error is exploited to allow a more efficient approach for correcting scale factor ambiguity. Recall the CR error function \[17\] described in (2.32) and (2.41), incorporating (7.19) with the reconstructed full-band estimate results in

$$e_{lm} = x_m^T \hat{h}_l - x_l^T \hat{h}_m$$

$$= x_m^T \Re \left\{ \frac{K/2-1}{2} \sum_{k=0}^{K/2-1} \hat{q}_{lk} \right\} - x_l^T \Re \left\{ \frac{K/2-1}{2} \sum_{k=0}^{K/2-1} \hat{q}_{mk} \right\},$$

(7.23)

for $m, l = 1, 2, \ldots, M, m \neq l$. The time-domain sample index $n$ is again dropped here since any $L$-sample frame from the observed signals satisfies (7.23). As indicated by (7.20), the error in (7.23) will be large when the channel estimates are incorrect due to different scale factors $\alpha_k$. Consequently, correcting for the scale factors ambiguity can be achieved by minimizing this error. There are generally two possible ways to incorporate this into the subband BSI framework:

- as a constraint on the adaptive subband blind system identification;
- as a post-processing step after identification.

The latter approach is considered in this chapter, which is equivalent to assuming the subband channel estimates with different scale factors as shown in (7.22) are obtained. The objective of scale factor ambiguity correction is thus to rectify the scale factor discrepancy in different subbands, this can be expressed by introducing a correction term $\beta_k$ for the $k$th subband to satisfy

$$\phi = \sum_{k=0}^{K/2-1} \beta_k \alpha_k$$

(7.24)

for $\phi \neq 0$ so that (7.23) becomes

$$\tilde{e}_{lm} = x_m^T \Re \left\{ \sum_{k=0}^{K/2-1} \beta_k \alpha_k \hat{q}_{lk} \right\} - x_l^T \Re \left\{ \sum_{k=0}^{K/2-1} \beta_k \alpha_k \hat{q}_{mk} \right\} = \phi e_{lm}.$$  

(7.25)
Since parameters $\alpha_k$ are unknown, $\beta_k$ can be found by minimizing (7.25) such that
\[
\hat{\beta}_k = \arg \min_{\beta_k} \sum_{m=1}^{M-1} \sum_{l=m+1}^{M} \tilde{e}_{m}^2 \text{ subject to } \|\beta_k\|^2 > 0, \quad \forall k. \tag{7.26}
\]

It is interesting to note that (7.25) can be rearranged by moving $x_m$ inside the summation which causes the scale factors to cancel. This is equivalent to introducing the signal into the full-band channel reconstruction and demonstrating that minimizing the CR error in the subbands also minimizes the full-band CR error. However, this does not result in accurate full-band channel estimates and, hence, the optimization problem in (7.26) needs to be solved by first reconstructing the full-band estimates using (7.19) and then calculating (7.25). A straightforward approach that has been found suitable for solving (7.26) is the Simplex method [166]. Since $\alpha_k$ are complex, they are treated by the Simplex algorithm as two parameters per subband. Therefore, there are $K$ free parameters to optimize and $\beta_k$ is initialized as $1_{(K/2) \times 1}$ so as to avoid trivial solutions.

### 7.5 Simulations Results

In this section, simulation results are presented to demonstrate the effectiveness of the developed scale factor ambiguity correction algorithm, where two main features are investigated:

- The effect of the scale factor variance across different subbands on the full-band channel estimate reconstruction;
- The effect of BSI errors in each subband system on the reconstruction of full-band channel estimate as well as on the scale factor error.

The NPM $\eta'(n)$ defined in (2.27) is employed to measure the BSI performance. The scale factor error is then quantified by the normalized variance of the corrected scale factors given by
\[
\xi = \frac{\text{var}\{A\beta\}}{\|A\beta\|_2^2}, \tag{7.27}
\]
where $A = \text{diag}\{\alpha_0 \alpha_1 \ldots \alpha_{K/2-1}\}$ is a diagonal matrix with true scale factors and $\beta = [\beta_0 \beta_1 \ldots \beta_{K/2-1}]^T$ are the correction terms. If the parameters $\beta_k$ correct the ambiguity
7.5 Simulations Results

Figure 7.3: Comparison of $N_c$ between full-band channel and subband channel for $k = 1$. 

Figure 7.4: Variation of the full-band BSI performance in $\text{NPM}_f$ against scale factor variance before and after correction. The subband $\text{NPM}_s$ is assumed to be $\text{NPM}_s = -\infty \text{ dB}$.

such that the scale factor is uniform over all subbands, then $\xi = 0$.

The filter bank used for all the simulations in this section is with $K = 8$ subbands and a decimation factor of $N = 4$. An prototype filter of length $L_{pr} = 64$ is designed using the iterative LS method [24], giving an estimated aliasing suppression of 92 dB.

7.5.1 Common Zeros in Decimated Subbands

As described in Section 7.4, it is expected that common zeros would be less likely to occur in subband systems with reduced length. To demonstrate this, simulated acoustic impulse responses with 512 taps generated in the room, of which the floor plan is depicted in Fig. 6.8, is used. This gives rise to subband channels of length $L_{\text{sub}} = 129$ and the GMC-ST algorithm developed in Chapter 4 is employed to compute the number of common zeros $N_c$. The simulation is carried out over various acoustic systems simulated at 20 different locations in the room with fixed speaker-microphone configuration. Without loss of generality, the subband system corresponding to the first microphone is chosen to be evaluated in terms of $N_c$, and Fig. 7.3 shows the result. As can be seen, subband decomposition does reduce $N_c$ significantly. It can therefore be expected that using filter bank with more subbands would be an effective way to mitigate the common zeros effect.
7.5 Simulations Results

Figure 7.5: Variation of NPMs against (a) NPMs before correction, after correction, and after correction with the ideal coefficients and (b) scale factor error ξ.

7.5.2 Performance of the Scale Factor Ambiguity Corrector

To evaluate the performance of the developed scale factor ambiguity corrector, a three-channel system of length \( L = 256 \) is generated with coefficients drawn from a standard Gaussian distribution. The equivalent subband systems are obtained using (7.18) and scale factors \( α_k \) are generated randomly with varying variances and applied to each subband channel response to simulate the subband channel estimates produced by BSI algorithms for the following synthetic experiments.

In the first experiment, it is assumed that perfect estimates of subband systems are obtained, that is, \( \text{NPM}_s = -\infty \text{ dB} \) such that BSI error would not distort the result. Scale factors with different variances are then introduced to each subband estimate. The estimation error for the reconstructed full-band channel responses, denoted as \( \text{NPM}_f \), is measured before and after employing the developed scale factor ambiguity correction algorithm. Figure 7.4 shows the outcome of this experiment, from which it can be observed that, as expected, these scale factors cause the reconstructed full-band channels to be inaccurate with \( \text{NPM}_f \) close to 0 dB. Applying the developed correction method resolves this issue, resulting in channels with \( \text{NPM}_f \) in the vicinity of \(-80 \text{ dB}\). Additionally, the variance of the scale factors does not correlate with their effect to the full-band channel reconstruction. This means that even little discrepancies in scale factors for different subbands can significantly degrade the full-band channel estimation.
In the second experiment, performance variation of the developed scale factor corrector against various levels of subband BSI error in terms of NPMs is demonstrated, where the estimate of the mth channel for the kth subband is simulated by

\[ \hat{h}_{mk} = \alpha_k(I_{L_{sub}} \times L_{sub} + \epsilon_{mk})h_{mk}, \quad \forall m, k \]  

(7.28)

for \( \epsilon_{mk} = \text{diag}\{\epsilon_{mk,0}, \epsilon_{mk,1}, \ldots, \epsilon_{mk,L_{sub} - 1}\} \), where the variance of \( \epsilon_{mk} = [\epsilon_{mk,0} \epsilon_{mk,1} \ldots \epsilon_{mk,L_{sub} - 1}] \) is set according to the desired NPMs. These subband channels are then used to investigate the results obtained using the Simplex algorithm in terms of full-band misalignment NPMf and the variance of the corrected scale factors calculated using (7.27). The results, averaged over 100 independent realizations for \( \alpha_k \), are shown in Fig. 7.5. Figure 7.5(a) first shows the NPMf against NPMs before and after scale factor correction and for the “ideal” case where \( \alpha_k \) is known. Figure 7.5(b) then shows the variation of \( \xi \) against NPMs. The following can then deduced from these results: (i) the performance of the scale factor correction algorithm degrades with a decreasing NPMs; (ii) scale factor correction has little effect when NPMs > −10 dB, even if the exact values of \( \alpha_k \) were known; and (iii) the developed method operates with a high accuracy at NPMs ≤ −50 dB.

### 7.5.3 Application to Speech Dereverberation

The developed scale factor ambiguity corrector is further applied to speech dereverberation over a two-channel recorded acoustic system extracted from the MARDY database [130], which are down-sampled at 8 kHz and truncated to 2400 FIR coefficients. A sample speech signal is obtained by concatenating both male and female utterance extracted from the APLAWD database [68] and resampled at 8 kHz. The time sequence and spectrogram of this speech sample has been shown in Fig. 6.11. Since the objective is to evaluate the effect of scale factor correction to speech dereverberation in the context of subband processing, a noiseless case is considered.

Similar to Section 7.5.2, various levels of subband BSI error NPMs are simulated. Table. 7.1 shows the corresponding full-band BSI error NPMf for cases with and without scale factor correction as well as for the “ideal” case where \( \alpha_k \) is known. Consistent with the result shown in Fig. 7.5(a), BSI errors on subband systems overwhelm the scale factor
### Table 7.1: Effect of scale factor ambiguity correction on BSI error

<table>
<thead>
<tr>
<th>NPM_s (dB)</th>
<th>NPM_f (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>w/o. correction</td>
</tr>
<tr>
<td>−80</td>
<td>−0.2</td>
</tr>
<tr>
<td>−30</td>
<td>−0.17</td>
</tr>
<tr>
<td>−20</td>
<td>−0.14</td>
</tr>
<tr>
<td>−10</td>
<td>−0.1</td>
</tr>
</tbody>
</table>

ambiguity correction. It can also be seen that for the case with known $\alpha_k$, there is still approximately 6 dB performance loss in terms of NPM_f during subband to full-band reconstruction.

Figure 7.6 finally shows the spectrograms of the dereverberated speech signal for cases of NPM_s = −30 dB and NPM_s = −80 dB shown in Table 7.1. The recovered speech signal is obtained by equalizing the reconstructed full-band channel estimates. Alternatively, subband multichannel equalization can be employed. The performance improvement brought about by scale factor ambiguity correction is clearly seen by comparing Fig. 7.6(b.1) and Fig. 7.6(c.1) with Fig. 7.6(b.2), Fig. 7.6(b.3) and Fig. 7.6(c.2), Fig. 7.6(c.3). For the case where NPM_s = −30 dB, however, there still exist spectral errors in Fig. 7.6(b.2) and Fig. 7.6(b.3). This is due to subband BSI inaccuracies which propagated into full-band channel estimate during reconstruction. This is not observed in cases where NPM_s = −80 dB. The BSD score computed for each subfigure agrees with such observation.

#### 7.6 Summary

In this chapter, the problem of scale factor ambiguity in subband-based BSI has been addressed. These scale factors are the same for all channels in each subband but differ between subbands, hence distorting the reconstruction of the full-band channels. Having reviewed the complex subband decomposition over oversampled GDFT filter bank, a novel approach was developed to correct such ambiguity by exploiting the relation-
ship between subband and full-band CR error. Utilizing the Simplex algorithm, a set of correction terms was found. Simulation results showed that, although the correction performance degrades with an increasing subband BSI error, it solves the scale factor ambiguity problem accurately when the channel identification is good (NPM_s ≤ −50 dB). Further results on speech dereverberation show that large subband BSI error can result in poor full-band BSI performance even though an accurate correction performance is achieved. Nevertheless, applying BSI in the context of subbands has been shown to be promising due to the significant reduction of common zeros.
Chapter 8

Conclusions and Future Work

This final chapter summarizes and concludes this research, where major problems addressed in this thesis are reviewed and key results are highlighted. Outlooks and ideas for future work are also suggested.

8.1 Summary

In this thesis, system identification algorithms for speech enhancement have been developed and analyzed. By identifying acoustic systems, distortions introduced to the transmitted speech signals can be compensated. Two speech enhancement problems, the network echo cancellation (NEC) and speech dereverberation, have been formulated and studied.

Frequency-Domain Adaptive Sparse System Identification for NEC

This thesis begins with the development of supervised single-channel adaptive system identification algorithms for NEC as presented in Chapter 3. Exploiting the sparseness of the network impulse response, the MMax [16] and SP [20] tap-selection strategies were employed to reduce the computational complexity and achieve a good convergence performance. These techniques were integrated into the MDF framework [21] aiming for an efficient implementation using FFT with reduced algorithmic delay. It was found that such integration was not straightforward. This is because, first of all, there exists a
8.1 Summary

The tradeoff between convergence performance and computational complexity regarding the incorporation of MMa tap selection into the MDF structure. Second, the sparse nature of network impulse response does not necessarily preserve in the MDF domain. Having addressed and overcome these problems, the SPMMax-MDF algorithm was developed. Simulation results using both Gaussian noise and speech samples have demonstrated up to 5 dB improvement in convergence performance and a more than 50% reduction of complexity for the SPMMax-MDF algorithm compared to other existing algorithms.

**Blind Multichannel Identification in the Presence of Common Zeros for Speech Dereverberation**

In the development of system identification algorithms with application to speech enhancement, the research reported in this thesis then moved to multichannel unsupervised (blind) scenarios, where the main focus was the investigation of the common zeros problem in blind identification of SIMO acoustic systems for speech dereverberation. Chapter 4 began by extending the conventional definition of common zeros with the presence of near-common zeros (NCZs) which are likely to occur in multichannel systems with high order such as SIMO acoustic systems. Together with exactly-common zeros, these zeros can significantly limit the performance of blind system identification (BSI) algorithms and the subsequent speech dereverberation. Two computationally efficient algorithms have therefore been developed to quantify the number of common zeros in multichannel acoustic systems with arbitrary sizes. These algorithms also facilitated the study of the common zeros problem and the design of robust BSI algorithms by linking the common zeros with the number of channels and corresponding BSI performance in terms of normalized projection misalignment (NPM) [113].

In Chapter 5, the concept of channel decomposition has been presented. Employing a blind order estimation approach based on eigenanalysis of the microphone signals, this concept separates the exactly-common zeros in a SIMO system from the remaining non-common ones. Such decomposition allows to mathematically demonstrate the effect of exactly-common zeros on CR-based BSI algorithms and multichannel equalization algorithms using Bezout theorem. In addition, it was noted that the failure of classic BSI algorithm was due to the lack of information about the potential presence of exactly-
common zeros. To overcome this, a two-stage BSI algorithm was developed to identify the decomposed channel components sequentially. This then led to a speech dereverberation approach robust to the exactly-common zeros. A gain of up to 5 dB in NPM for BSI performance over existing methods and a consistent improvement of dereverberation performance in terms of Bark spectral distortion (BSD) \[129,74\] were demonstrated through simulation results.

A novel concept for mitigating the effect of NCZs on BSI and speech dereverberation has been developed in Chapter 6, which was referred to as the forced spectral diversity (FSD). This concept was inspired by the quantified multichannel diversity, which correlates monotonically with the number of NCZs. Through the collective use of spectral diversifying filters and effective channel undermodelling, FSD processing gives rise to a modified system with additional channel diversity, hence allowing an improved BSI and equalization performance for speech dereverberation. Various illustrative and numerical examples have been presented to describe and verify this concept. With a typical example of spectral diversifying filters, simulation results demonstrated an improvement in BSI performance of up to 6.5 dB in terms of NPM over various acoustic systems excited by WGN and speech input signals compared to classic methods. Such improvement in turn resulted in an improved speech dereverberation in terms of BSD. It was also observed that two-channel FSD processing outperformed classic methods with five channels, which makes the FSD concept computationally attractive.

In the last technical chapter, the deployment of subband-based BSI algorithms has been explored by utilizing complex subband decomposition over oversampled GDFT filter bank, which is expected to be beneficiary in terms of BSI performance and complexity due to shorter channel length in subbands. One of the major obstacles that limits the study of subband-based BSI, however, is the scale factor ambiguity across different subbands, which results in erroneous full-band channel estimate reconstructed from corresponding subband estimates. Utilizing the relationship between the subband and full-band CR error, a novel approach of correcting such discrepancy was developed and implemented using an iterative optimization technique. From simulation results, it was observed that a good scale ambiguity correction performance can be achieved using subband system estimates with sufficient accuracy. It was also motivating to find that there are much fewer common zeros in subband systems than in the full-band system. These
8.2 Conclusions

Figure 8.1: Relationship between the common zeros and other key parameters in the context of BSI.

results facilitated further study of subband-based BSI techniques.

Based on the work presented in this thesis, the relationship between the common zeros and other key parameters in the context of BSI can now be summarized and depicted by Fig. 8.1.

8.2 Conclusions

System identification algorithms are of great importance to speech enhancement applications such as NEC and speech dereverberation. This is because: (i) the popularity of VoIP coupled with an increasing expectation for natural communication over packet-switched networks has called for system identification algorithms for NEC with less complexity and delay yet fast convergence performance; and (ii) the increasing demand for robust speech dereverberation has expressed the desire for BSI algorithms to be capable of identifying multichannel acoustic systems accurately and efficiently.

To meet these demands, a frequency-domain adaptive algorithm was first developed for NEC, where the incorporation of MMax and SP tap-selection schemes with the MDF structure allowed this algorithm to achieve a fast convergence performance with reduced complexity and a low delay. Analysis and simulation results presented have shown that the developed SPMMax-MDF algorithm outperforms most existing algorithms in terms of converging speed with significantly reduced computational complexity and delay. This algorithm can therefore be found useful in VoIP where a large
density of network echo cancellers are employed.

More importantly, blind identification of SIMO acoustic systems for speech dereverberation has been studied. As one of the classic channel identifiability conditions, the common zeros problem significantly limits the development of robust SOS-based BSI algorithms for acoustic signal processing since the FIR models of acoustic systems contain a large number of FIR coefficients, hence resulting in a high density of zeros around the unit circle giving rise to not only exactly-common zeros but also NCZs. To address this problem fully, two computationally efficient clustering algorithms have been developed to extract common zeros from multichannel systems with arbitrary sizes. These algorithms are important since they allowed quantification of the effect of common zeros and facilitated the design of robust BSI algorithms for speech dereverberation.

To improve BSI performance in the presence of common zeros, three algorithms have been developed and analyzed in this thesis, which include: (i) the two-stage BSI and dereverberation algorithm; (ii) the FSD concept; and (iii) the scale factor ambiguity correction algorithm for subband-based BSI. The two-stage algorithm, first of all, is based on channel decomposition which allows a sequential identification of exactly-common zeros and the remaining non-common ones. Although exactly-common zeros have been known to be the greatest common divisor of the multiple channels, no exiting algorithms are explicitly based on such property. The novel FSD concept then aims at mitigating the effect of NCZs by combining spectral diversifying filters and effective channel under-modelling to derive a modified system with additional channel diversity, hence leading to an improved BSI and speech dereverberation performance. It is expected that employing BSI algorithms over decimated subbands would be beneficiary since subband systems have reduced length such that common zeros are less likely to occur. To motivate further research on subband-based BSI techniques, a scale factor ambiguity correction algorithm was developed lastly in this thesis to correct for different scale factors associated with each estimated subband system for a successful reconstruction of the full-band system estimate. These three algorithms have been demonstrated, by simulations, to improve significantly the performance of BSI and the subsequent speech dereverberation in the presence of common zeros. As a consequence, the common zeros problem can be mitigated.
8.3 Future work

A part from what have been addressed in this thesis, many unsolved issues associated with system identification techniques for speech enhancement have been continuously motivating further research in this field. In this very last section, some prospective ideas of future work are presented followed by an outlook.

As presented in Chapter 3, tap-selection criteria can be designed to take into account characteristics of acoustic impulse responses such as the sparseness for improving convergence performance. The same idea could be applied in the context of BSI. First, since acoustic impulse responses are always to some extent sparse and the MDF framework has been applied in blind multichannel identification [156], the SPMMax-MDF algorithm developed in Chapter 3 can potentially be exploited in the context of BSI where the SP tap-selection strategy could achieve a fast convergence performance with reduced computational complexity and algorithmic delay. Second, specific tap-selection scheme that correlates with the common zeros could also be developed to improve the robustness of BSI algorithms. A study of the relationship between zeros and adaptive filter coefficients would be an interesting start.

Clusters of NCZs have been defined based on the pairwise Euclidean distance between each member of the cluster. However, metrics other than the pairwise Euclidean distance are worthwhile investigating to measure the closeness of the zeros within a cluster of NCZ. An interesting example is the centroid point of the cluster. In addition, although the number of common zeros can be found using GMC algorithms given any tolerance $\delta_c$, there has not been a mechanism to determine what the value of $\delta_c$ should be to accurately reflect the effect of common zeros on the BSI algorithms. It is also expected that each cluster of common zeros could have different effect contributing to the performance degradation in BSI. The relationship between $\delta_c$ and the BSI performance would thus be worthwhile investigating.

The development of the FSD concept in this thesis has also opened several interesting directions of future research. Since one major component of the FSD concept is the spectral diversifying filters, an effective design of such filters is logically desirable for the maximization of the channel diversity introduced to the modified system. Although this objective is system-dependent as described in Section 6.5, design procedures with re-
spect to a general category of systems are nevertheless useful. It is also interesting to note that the FSD processing results in a successful identification of the modified system that is derived from the original system, where the former is equivalent to the “significant” part of the latter as described in Section 6.3.2. Although the motivation was to achieve robust speech dereverberation in the presence of NCZs, it would be worthwhile investigating, from a BSI point of view, how reverse processing can be achieved to recover the original system. The fact that those truncated filter coefficients may contain little energy for acoustic systems since they correspond to the reverberation tail suggests that such recovery, if successful, would be of limited approximation error.

The scale factor ambiguity correction developed in Chapter 7 has motivated further research on subband-based BSI techniques. It is noted, however, that frequency-domain BSI algorithm such as the NMCFLMS algorithm [23] is essentially a special case of the subband approach where each frequency bin can be blindly identified rather than each subband. It is therefore expected that these estimated frequency bins would also contain distinct complex scale factors. It would be insightful to investigate why, the NMCFLMS algorithm, for example, does not suffer from such ambiguity problem.

Over the last few years, there has been a substantial increase in the interest of applying BSI techniques on applications for both civil and military purposes. For example, integration of microphone arrays and the hands-free functionalities in portable devices has become more and more popular, which has made robust multichannel BSI algorithms extremely desirable for speech dereverberation and subsequent speech enhancement applications. The limited physical size and power supply for these portable devices will not allow many microphones to be integrated, hence resolving the common zeros problem is expected to be crucial. Additionally, BSI algorithms can be employed to identify the multipath effect in underwater acoustic communications, urban environments where strong signal reflections from buildings or other objects occur, and in other enclosed spaces such as in-vehicle military command centres for improving the overall communications quality. The tracking capability of adaptive algorithms can also be explored to deal with the time-varying nature of the impulse responses so as to provide practical solutions to the applications of interest. Therefore, many exciting developments involving system identification techniques can be expected to appear in the not so distant future.
Bibliography


