Hidden Markov Model-Based Speech Enhancement

Akihiro Kato

A thesis submitted for the Degree of Doctor of Philosophy

University of East Anglia

School of Computing Sciences

24, May, 2017

Abstract

This work proposes a method of model-based speech enhancement that uses a network of HMMs to first decode noisy speech and to then synthesise a set of features that enables a speech production model to reconstruct clean speech. The motivation is to remove the distortion and residual and musical noises that are associated with conventional filtering-based methods of speech enhancement.

STRAIGHT forms the speech production model for speech reconstruction and requires a time-frequency spectral surface, aperiodicity and a fundamental frequency contour. The technique of HMM-based synthesis is used to create the estimate of the time-frequency surface, and aperiodicity after the model and state sequence is obtained from HMM decoding of the input noisy speech. Fundamental frequency were found to be best estimated using the PEFAC method rather than synthesis from the HMMs.

For the robust HMM decoding in noisy conditions it is necessary for the HMMs to model noisy speech and consequently noise adaptation is investigated to achieve this and its resulting effect on the reconstructed speech measured. Even with such noise adaptation to match the HMMs to the noisy conditions, decoding errors arise, both in terms of incorrect decoding and time alignment errors. Confidence measures are developed to identify such errors and then compensation methods developed to conceal these errors in the enhanced speech signal.

Speech quality and intelligibility analysis is first applied in terms of PESQ and NCM showing the superiority of the proposed method against conventional methods at low SNRs. Three way subjective MOS listening test then discovers the performance of the proposed method overwhelmingly surpass the conventional methods over all noise conditions and then a subjective word recognition test shows an advantage of the proposed method over speech intelligibility to the conventional methods at low SNRs.

Acknowledgements

First, thanks go to Dr Ben Milner for his excellent supervision throughout my PhD at the University of East Anglia. This work would not have been possible without his support and advice. I would also like to thank Prof. Richard Harvey and Prof. Stephen Cox for their support during my doctoral research as well as their leadership of the Speech, Language and Audio Processing group in the University of East Anglia.

Thanks also go to the members in the speech lab for their kind and heart-warming support in my research and school life.

I am deeply grateful to my family in Japan for all their love. I was always able to feel their moral support during my study abroad and I would not be able to achieve this work without it.

Finally, I would also like to thank my examiners, Dr Mark Fisher of the University of East Anglia and Dr Jon Barker of the University of Sheffield, for their comments and suggestions which improved the quality of this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter first introduces the area of speech enhancement and clarifies problems that need to be addressed. The proposed method, which is a new approach to speech enhancement, is then introduced followed by its target applications. The objective of the research, main problems to achieve it and contributions to the research area are then clarified, and finally, the organisation of the thesis is explained.

1.1 Speech Enhancement

Speech enhancement is concerned with improving some perceptual aspects of speech that had been degraded by noise or other factors, e.g. channel distortion, packet loss and echo [1]. The focus on this work is noise in speech, which causes two main effects on the perception of speech. Firstly, the auditory perception about the quality of the speech signal is deteriorated and secondly, intelligibility of speech is affected. Such degradation of speech quality and intelligibility brings the potential of increasing listener fatigue and misunderstanding during communication and thus, techniques for speech enhancement are highly desirable.

Degradation of speech by noise occurs when the source of a speech signal is affected by noise or when noise exists on communication channels. Such a situation is very common in voice communication systems, and this phenomenon is mathematically modelled as

$$y(n) = x(n) + d(n) \tag{1.1}$$

where $x(n)$, $d(n)$ and $y(n)$ represent a discrete-time domain signal of speech, noise and degraded noisy speech respectively, and n denotes a discrete-time index. Therefore, the most intuitive approach to speech enhancement is to identify unknown $d(n)$ from $y(n)$ and then to remove it from $y(n)$. However, it is not possible to identify the exact sequence of $d(n)$ as long as the only accessible information is $y(n)$, and thus, a variety of methods to obtain an estimate of noise, $\hat{d}(n)$, instead of $d(n)$ have been proposed [1]. These often assume noise stationarity and exploit periods of nonspeech activity in $y(n)$. This enables subtraction of $\hat{d}(n)$ from $y(n)$ and derives an estimate of clean speech, $\hat{x}(n)$, as

$$\hat{x}(n) = y(n) - \hat{d}(n) \tag{1.2}$$

Details are discussed in Chapter 2 but this works as a noise filter of speech as shown in Figure 1.1, and it is explicit that residual noise is left in $\hat{x}(n)$ when $\hat{d}(n)$ is underestimated. Conversely, when $\hat{d}(n)$ is overestimated, the speech signal is distorted and it may further reduce speech intelligibility [2]. There are many alternative methods based on

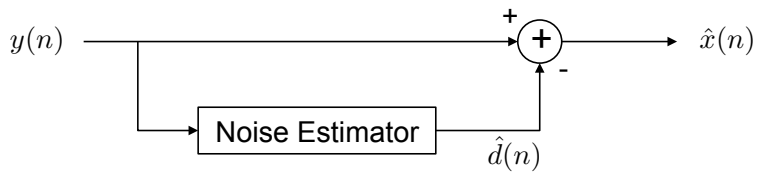

Figure 1.1: Noise filtering approach to speech enhancement.

this filtering approach to speech enhancement, e.g. spectral subtraction, Wiener filtering, statistical model-based methods and subspace algorithms [1]. As evaluated in Chapter 2, although these methods have shown effectiveness to suppress noise in conditions with relatively high signal to noise ratio (SNR), performance falls at low SNRs such as 0 dB and below. Therefore this work proposes a novel approach that moves away from the filtering methods to achieve significant improvement to performance at low SNRs in stationary and non-stationary noise.

Additionally, to acquire additional information to estimate $\hat{d}(n)$ from $y(n)$, various approaches have been proposed, for example, multi-channel speech enhancement uses multiple microphones to enhance $y(n)$ into a multiple dimensional signal in order to

extract positional relationship between speech source and noise source and then it is exploited to enable better source separation [3]. Alternatively, audio-visual speech enhancement uses a camera to capture visual articulators, e.g. the position of speaker's lips, as auxiliary speech information which is independent from the SNR [4]. This thesis, however, focuses on single-channel speech enhancement in which the only accessible information about speech is monaural noisy speech, $y(n)$. This represents a challenging problem but is easier from a practical implementation point of view.

1.2 Proposed Method

The method of speech enhancement proposed in this thesis is based on a model-based approach which uses statistical parametric models of speech and a speech production model. Specifically, the statistical parametric models are realised by hidden Markov models (HMMs), which are discussed in Chapter 4, and the STRAIGHT vocoder, which is explored in Chapter 3, is adopted for the speech production model. Figure 1.2 illustrates the basic architecture of the proposed method. In this method a set of speech features are

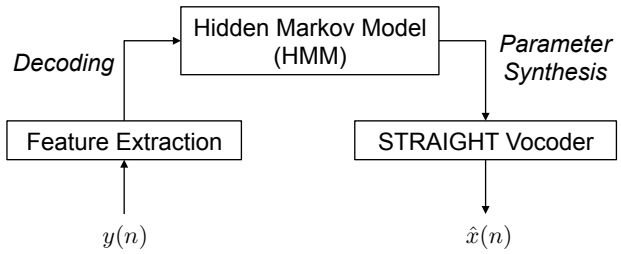

Figure 1.2: The basic architecture of the proposed method.

first extracted from noisy speech and then they are decoded into a sequence of statistical models of speech parameterised as HMMs. Since the HMMs have been trained with clean natural speech, they can synthesise a set of features of noise-free speech corresponding to the decoding result. Finally, the STRAIGHT vocoder reconstructs time-domain clean speech from the synthesised parameters. The output is isolated from the noise component of the input since the speech features of the output are determined only by the statistical parameters. Therefore, the output is free from residual noise and musical noise unlike the filtering-based method shown in Figure 1.1. The statistical processes are, however,

expected to bring other types of artefacts which are attributed to, for example, decoding errors and over-smoothing, to the output speech. Furthermore, model-based approaches require an off-line process to train HMMs of speech that is not needed for filtering-based methods. Thus, the size and complexity of the system tend to increase. The detail of the method and such problems are discussed in Chapter 4 and later.

1.3 Application

The proposed method of speech enhancement is assumed to have various uses with the most representative application being mobile communication. For example, talking on a mobile phone outdoors and automatic speech recognition (ASR) in an automobile. Therefore, the proposed method needs to deal with a wide range of noise types including both stationary and non-stationary noise. This thesis evaluates the performance of speech enhancement with white Gaussian noise that represents stationary noise and babble noise (NOISEX-92) that represents non-stationary noise at SNRs from -5 dB to 10 dB by both objective and subjective tests to match the test conditions to practical applications.

1.4 Objective and Problems

The objective of this research is set as follows.

- To develop a new method of speech enhancement based on a model-based approach in order to achieve better speech quality and intelligibility than conventional filtering-based approaches at low SNRs with more compact system resource than existing model-based speech enhancement.

In order to achieve the preceding purpose of the research with the proposed method, the main problems addressed in this thesis are as follows.

- To employ an speech production model and speech features for the proposed method of the model-based approach to speech enhancement

- To implement the framework which includes the processes of HMM decoding, HMM synthesis and speech reconstruction to realise the proposed method

- To develop methods to obtain better HMM decoding accuracy in the proposed method
- To develop methods to detect decoding errors and methods to compensate for these erroneous frames
- To obtain better quality and intelligibility in the HMM-based speech synthesis process

1.5 Contributions

This thesis contributes to the research area of speech processing by achieving the preceding objective. Simultaneously, a variety of experiments in this thesis show interesting findings in the related technologies. These also contribute to the research area in terms of both theoretical and practical development. Moreover, two conference papers have been published as interim reports during this research [5,6] and have given contributions to the research field.

1.6 Organisation of the Thesis

The remainder of this thesis is organised into seven further chapters as follows:

2. **Conventional Methods for Speech Enhancement:** This chapter first discusses a variety of conventional methods for speech enhancement based on the filtering approach and then evaluates performance with objective tests. The latter part of the chapter explores examples of reconstruction-based approaches to speech enhancement which have recently been proposed.

3. **Speech Production Models:** The proposed method in this thesis takes a model-based speech reconstruction approach to speech enhancement. This chapter, therefore, discusses speech production models for the process of speech reconstruction. The human physical speech production process is first described and it is then extended to engineering models for speech production such as the source-filter models, the STRAIGHT vocoder and the sinusoidal model. The fundamental frequency is

a critical speech feature for the speech production model, and thus, methods to extract the fundamental frequency are then explored.

4. **Hidden Markov Model-Based Speech Enhancement:** The details of the proposed method of speech enhancement are presented in this chapter. The concept of HMMs and algorithms to apply HMMs are first discussed and then techniques for HMM decoding and HMM-based speech synthesis are explored with their application examples, including automatic speech recognition and text-to-speech. Finally, the proposed method of HMM-based speech enhancement is presented by combining the techniques of HMM-decoding, HMM-based speech synthesis and the STRAIGHT voocder.

5. **Adaptation of Hidden Markov Models to Noisy Speech:** Decoding accuracy in noisy speech is poor when HMMs trained with clean speech are used in the HMM decoding process. Therefore, this chapter discusses methods to adapt HMMs trained with clean speech to noisy speech in order to improve HMM decoding accuracy practically.

6. **Improvement to Hidden Markov Model-Based Speech Enhancement:** This chapter discusses methods to improve performance of the proposed HMM-based speech enhancement. A method to compensate for decoding errors which reduce quality and intelligibility of the output speech is first presented. Then HMM-based speech enhancement using the global variance model is studied to compensate for over-smoothing in the synthesised speech parameters.

7. **Evaluation of the Proposed HMM-Based Speech Enhancement:** This chapter reports the evaluation results of the proposed method comparing with conventional filtering methods after carrying out objective and subjective tests.

8. **Conclusions and Further work:** The final chapter first draws conclusions about the proposed method of HMM-based speech enhancement and then describes how the system may be extended.

Chapter 2

Conventional Methods for Speech Enhancement

This chapter first shows overviews of the conventional methods for speech enhancement which use filtering-based approaches and then conducts practical experiments to show their performance on speech enhancement. Alternative methods to the conventional filtering-based approaches are then discussed as reconstruction-based approaches including the corpus and inventory-based method and the model-based method.

2.1 Introduction

Conventional methods for speech enhancement are normally formed as a two stage process. The contaminating noise in the speech or signal to noise ratio (SNR) of the noisy speech is estimated in the first stage and then the estimate of the noise is removed from the noisy speech by various types of filters in the second stage. Most speech enhancement methods consisting of these processes are largely categorised into spectral subtraction, Wiener filtering, statistical and subspace methods, and it is known that although these filtering-based approaches are effective to improve speech quality, those performance depends on the accuracy of noise and SNR estimation and, consequently, residual noise, musical noise and distortion are introduced to the enhanced speech by the estimation errors [1].

As an alternative to the filtering approaches, reconstruction-based approaches have

recently been proposed to reduce the artefacts produced by filtering-based methods [7]. Methods using these approaches reconstruct clean speech by estimating the acoustic features of the clean speech rather than filter the noisy speech. These methods are generally divided into two types in terms of approaches to reconstruct speech. The first uses a notion of unit selection synthesis [8], which have successfully been applied to text-to-speech (TTS) applications [9], for the speech reconstruction process in which segments of speech, e.g. phonemes, are first selected from a corpus or inventory of natural speech segments and then concatenated to synthesise clean speech while the other type of the methods utilises a speech production model, e.g. vocoders, to reconstruct clean speech. The work proposed in this thesis belongs to the latter category of the reconstruction-based approaches using the STRAIGHT vocoder for the speech production model.

The following sections first present overview of different methods for the noise estimation. After that, methods of speech enhancement which represent filtering-based enhancement are discussed and then examined by objective tests in terms of quality and intelligibility of the enhanced speech. The topic is then moved to the reconstruction-based approaches including the corpus and inventory-based method which represents the methods which use a notion of unit selection synthesis for the reconstruction process, and model-based speech enhancement, which represents the methods to utilise a speech production model, that have attracted a lot of research attention recently [5, 7, 10–14].

2.2 Noise Estimation

Noise estimation is the first process of filtering-based speech enhancement. The simplest method for this process is to use voice activity detection (VAD), whose overview is presented in the first part of the section. However, VAD-based estimation cannot achieve enough accuracy in low SNR conditions [1, 15], therefore, the latter part of the section introduces a method of minimum statistics representing minimal-tracking algorithms.

2.2.1 VAD-Based Noise Estimation

VAD is a simple method to classify frames of the speech as speech-active or inactive frames. Various algorithms for VAD have been proposed and applied successfully to

commercial applications [1, 16–18].

The simplest way of VAD for a discrete-time speech signal, $s(n)$, is to calculate the energy of the mixed signal at each frame, and to classify frames whose energy is more than certain threshold, λ, as speech-active frames, otherwise the frames are categorised as speech-inactive frames. Namely, when a frame in $s(n)$ is represented as a vector as

$$\mathbf{s}_i = [s(n+iL), s(n+iL+1), \ldots, s(n+(i+1)L-1)]^T \quad (i = 0, 1, \ldots) \tag{2.1}$$

where i and L denote a frame index and a frame length, the VAD scenario gives the following classification.

$$\left.\begin{array}{ll} \mathbf{s}_i \in \mathbb{C}_{sa} & \text{as} \quad \mathbf{s}_i^T \mathbf{s}_i > \lambda \\ \mathbf{s}_i \in \mathbb{C}_{si} & \text{otherwise} \end{array}\right\} \quad \text{for } \forall i \tag{2.2}$$

where $\mathbb{C}_{sa}$ and $\mathbb{C}_{si}$ represent the cluster of speech-active frames and the cluster of speech-inactive frames respectively.

To attain more robust performance [19] proposes another threshold σ by which all the frames in $\mathbb{C}_{si}$ are reclassified as follows.

$$\left.\begin{array}{ll} \mathbf{s}_j^{si} \in \mathbb{C}'_{si} & \text{as} \quad \|\mathbf{s}_j^{si} - \bar{\mathbf{c}}_{si}\| < \sigma \\ \mathbf{s}_j^{si} \in \mathbb{C}'_{sa} & \text{otherwise} \end{array}\right\} \quad \text{for } \forall \mathbf{s}_j^{si} \in \mathbb{C}_{si} \tag{2.3}$$

$$\bar{\mathbf{c}}_{si} = \frac{1}{N} \sum_{\forall \mathbf{s}_j^{si} \in \mathbb{C}_{si}} \mathbf{s}_j^{si} \tag{2.4}$$

where $\mathbf{s}_j^{si}$ denotes the j-th element in $\mathbb{C}_{si}$ and N is the number of elements in $\mathbb{C}_{si}$. A cepstral analysis has also been proposed to achieve more robustness, in which the frames are classified by cepstral distances [20]. After frames in the speech are categorised as speech-active or inactive, the centroid of the spectra in the speech-inactive cluster is calculated as the estimate of the noise spectrum.

2.2.2 Minimum Statistics

The notion of noise estimation with VAD is very simple and easy for implementation but not enough accurate at low SNRs [1, 15, 16]. Moreover, it cannot track changes of

statistical features in non-stationary noise during speech-active periods. To tackle this problem, a noise estimation method using minimum statistics [21] is discussed in this section.

In the discrete-time domain, a noisy speech, $y(n)$, can be described as the sum of the speech, $x(n)$, and noise, $d(n)$, as

$$y(n) = x(n) + d(n) \tag{2.5}$$

Each signal is divided into frames with an L-length window, $w(m)$, for analysis as follows.

$$\left.\begin{aligned} y_i(m) &= y(p) \cdot w(m) \\ x_i(m) &= x(p) \cdot w(m) \\ d_i(m) &= d(p) \cdot w(m) \end{aligned}\right\} \quad \begin{aligned} &p = iL,\ iL+1, \ldots,\ (i+1)L-1 \\ &m = 0, 1, \ldots, L-1 \end{aligned} \tag{2.6}$$

where i denotes a frame index ($i = 0, 1, \ldots$). These frames are then transformed into the frequency domain applying N-point short-time Fourier transform (STFT) analysis.

$$X_i(f) = \mathscr{F}\left[x_i(m)\right] \tag{2.7}$$

$$D_i(f) = \mathscr{F}\left[d_i(m)\right] \tag{2.8}$$

$$Y_i(f) = \mathscr{F}\left[y_i(m)\right] \tag{2.9}$$

where $\mathscr{F}$ is the notation of the discrete-time Fourier transform (DFT), and f represents a frequency bin index ($f = 0, 1, \ldots, F-1$). The power spectral density (PSD) of $Y_i(f)$ is approximated to the sum of the PSD of $X_i(f)$ and the PSD of $D_i(f)$ because the cross term of $X_i(f)$ and $D_i(f)$ can be ignored as long as the speech and noise are independent each other.

$$\begin{aligned} \mathcal{E}\left[|Y_i(f)|^2\right] &= \mathcal{E}\left[|X_i(f)|^2\right] + \mathcal{E}\left[|D_i(f)|^2\right] + 2\mathcal{E}\left[|X_i(f)|\right]\mathcal{E}\left[|D_i(f)|\right] \\ &\approx \mathcal{E}\left[|X_i(f)|^2\right] + \mathcal{E}\left[|D_i(f)|^2\right] \end{aligned} \tag{2.10}$$

where the notation $\mathcal{E}\left[\cdot\right]$ denotes the statistical expectation operator.

Minimum statistics is based on a notion where short term PSD in individual frequency bands often decays to the noise floor even during speech active periods [21].

Therefore, the short term PSD of noise during a fixed observation length, K, is estimated by tracking the minimum of the periodogram $|Y_i(f)|^2$ during K. However, $|Y_i(f)|^2$ fluctuates rapidly, therefore, the estimate of PSD of noise, $\hat{P}_i(f)$, is tracked after applying a weighted moving average.

$$\bar{P}_i(f) = \begin{cases} |Y_0(f)|^2 & i = 0 \\ |Y_i(f)|^2 & i = lK \quad (l = 1, 2, \dots) \\ \alpha \bar{P}_{i-1}(f) + (1-\alpha)|Y_i(f)|^2 & \text{otherwise} \end{cases} \tag{2.11}$$

$$\hat{P}_i(f) = \begin{cases} \bar{P}_0(f) & i = 0 \\ \bar{P}_i(f) & i = lK \quad (l = 1, 2, \dots) \\ \min\left\{\bar{P}_{i-1}(f), \bar{P}_i(f)\right\} & \text{otherwise} \end{cases} \tag{2.12}$$

where α denotes a weight constant.

Several algorithms to optimise and compensate the preceding algorithm have also been proposed [1, 21–23].

2.3 Filtering-Based Speech Enhancement

Filtering-based algorithms for speech enhancement is a two stage process of first estimating the noise, and then filtering the speech using the estimated noise. Various approaches to the filtering process have been proposed and they are categorised as mentioned in Section 2.1. Each of those filtering methods are discussed in this section.

2.3.1 Spectral Subtraction

Given noisy speech as Equations (2.5)-(2.9), a frame of the complex spectrum of the clean speech is derived in polar form.

$$X_i(f) = Y_i(f) - D_i(f) \tag{2.13}$$

$$= |Y_i(f)|e^{j\Phi_y^i(f)} - |D_i(f)|e^{j\Phi_d^i(f)} \tag{2.14}$$

where $\Phi_y(f)$ and $\Phi_d(f)$ are the phase spectra of the noisy speech and noise respectively. As the noise spectrum is not known precisely, the noise magnitude is replaced with the

magnitude of the estimated noise spectrum at the preceding process in order to derive the estimate of the spectral magnitude of the clean speech. The phase of the clean speech is not known so it is then replaced with the phase of the noisy speech. This is motivated by the fact that phase spectra do not contribute to intelligibility as much as magnitude spectra in the condition of short time window length [24], and derives the estimate of the spectrum of the clean speech, $\hat{X}_i(f)$.

$$\hat{X}_i(f) = \left(|Y_i(f)| - |\hat{D}_i(f)|\right) e^{j\Phi_y^i(f)} \tag{2.15}$$

where $|\hat{D}_i(f)|$ represents the estimated spectral magnitude of the noise. The time-domain enhanced speech can be obtain from Equation (2.15) by simply applying inverse Fourier transform.

Equation (2.15) is the underlying principle of the spectral subtraction and several derivative algorithms are proposed [1, 25–28]. for instance, the following applies the subtraction in the spectral power domain and simultaneously compensates overestimation or underestimation of $|\hat{D}_i(f)|^2$.

$$\begin{aligned} |\hat{X}_i(f)|^2 &= |Y_i(f)|^2 - \alpha|\hat{D}_i(f)|^2 && (2.16)\\ &= H_i(f)|Y_i(f)|^2 && (2.17)\\ H_i(f) &= 1 - \alpha\frac{|\hat{D}_i(f)|^2}{|Y_i(f)|^2} && (2.18) \end{aligned}$$

where α denotes an optimised constant value to adjust the estimation. The power of the resulting spectrum can be negative value in Equation (2.16) due to overestimation of the noise. Therefore, several methods for rectification are proposed [27], for example,

$$\begin{cases} |\hat{X}_i(f)|^2 = |Y_i(f)|^2 - \alpha|\hat{D}_i(f)|^2 & \text{as} \quad |Y_i(f)|^2 - \alpha|\hat{D}_i(f)|^2 \geqslant 0 \\ |\hat{X}_i(f)|^2 = |Y_i(f)|^2 & \text{otherwise} \end{cases} \tag{2.19}$$

The preceding examples of the spectral subtraction are linear process but several methods having non-linear processing are also proposed [27]. A method, for example,

applies weighted moving average to $|\hat{D}_i(f)|^2$ and $|Y_i(f)|^2$ before the subtraction.

$$|\hat{X}_i(f)|^2 = |\bar{Y}_i(f)|^2 - \alpha|\bar{D}_i(f)|^2 \quad (2.20)$$

$$|\bar{D}_i(f)|^2 = \lambda_d|\hat{D}_{i-1}(f)|^2 + (1-\lambda_d)|\hat{D}_i(f)|^2 \quad (2.21)$$

$$|\bar{Y}_i(f)|^2 = \lambda_y|Y_{i-1}(f)|^2 + (1-\lambda_y)|Y_i(f)|^2 \quad (2.22)$$

where λ_d and λ_y are weight constants. Another example is to divide the frequency domain of the speech and noise into K sub-bands, and then replace the constant, α, in Equation (2.20) with a variable $\alpha_k(i)$ associated with sub-band k ($k = 0, 1, \ldots, K-1$). $\alpha_k(i)$ varies according to the *a posteriori* SNR in the corresponding sub-band of the frame.

$$\hat{\mathbf{X}}_i^k = \bar{\mathbf{Y}}_i^k - \alpha_k(i)\bar{\mathbf{D}}_i^k \quad (2.23)$$

$$\alpha_k(i) = \beta \cdot 20\log_{10}\left(\frac{\bar{\mathbf{Y}}_i^k}{\bar{\mathbf{D}}_i^k}\right) \quad (2.24)$$

where $\hat{\mathbf{X}}_i^k$, $\bar{\mathbf{Y}}_i^k$ and $\bar{\mathbf{D}}_i^k$ represent vectors consisting of the power spectrum in the k-th sub-band of $|\hat{X}_i(f)|^2$, $|\bar{Y}_i(f)|^2$ and $|\bar{D}_i(f)|^2$ respectively, and β denotes a constant determined empirically.

The spectral subtraction algorithm is based on the assumption that phase spectra do not contribute to intelligibility as much as magnitude spectra in short time frame analysis as mentioned above. Recent research, however, has discovered that phase spectra can contribute to intelligibility as much as magnitude spectra even for short time duration when analysis-modification-synthesis parameters are properly selected [29]. This inconsistency has affected the performance of the spectral subtraction methods.

2.3.2 Wiener Filter

The spectral subtraction such as Equations (2.17) and (2.18) straightforwardly derive the spectral power or magnitude of the clean speech only from the noisy speech and the estimate of the noise. Therefore, the transfer function of the filter is not optimised by the estimation errors. The Wiener filtering approach discussed in this section optimises the transfer function of the filtering process by minimising the estimation errors in terms of mean-square error.

2.3.2.1 Theory of Wiener Filters

A Wiener filter is a linear and time-invariant filter to approximate an input signal, $s(n)$, to a desired signal, $\delta(n)$. Figure 2.1 shows the structure of a Wiener filter. The resultant

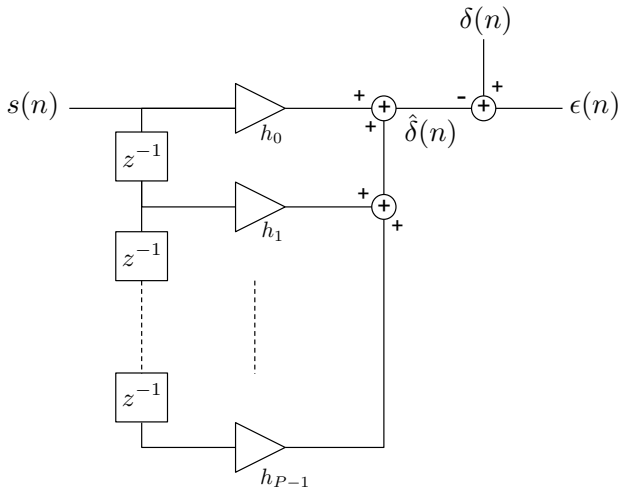

Figure 2.1: Block diagram of Wiener filters.

output of the filter, $\hat{\delta}(n)$ is given as

$$\hat{\delta}(n) = \sum_{k=0}^{P-1} h_k s(n-k) \tag{2.25}$$

where $h_0, h_1, \cdots, h_{P-1}$ are the filter coefficients (impulse response) of Pth-order Wiener filters, and the error between the filter output and desired signal is derived as

$$\begin{aligned} \epsilon(n) &= \delta(n) - \hat{\delta}(n) && (2.26) \\ &= \delta(n) - \sum_{k=0}^{P-1} h_k s(n-k) && (2.27) \end{aligned}$$

In the frequency domain, Equation (2.27) derives

$$\varepsilon(f) = \Delta(f) - H(f)S(f) \tag{2.28}$$

where $\varepsilon(f)$, $\Delta(f)$, $S(f)$ and $H(f)$ are the Fourier transform of $\epsilon(n)$, $\delta(n)$, $s(n)$ and $h(n)$ respectively. The frequency response of Wiener filters, $H(f)$, is optimised by minimising

the mean-square error, J.

$$\begin{aligned} J = \mathcal{E}\left[|\varepsilon(f)|^2\right] &= \mathcal{E}\left[\Delta(f)\Delta^*(f)\right] + H(f)H^*(f)\mathcal{E}\left[S(f)S^*(f)\right] \\ &\quad -H(f)\mathcal{E}\left[S(f)\Delta^*(f)\right] - H^*(f)\mathcal{E}\left[\Delta(f)S^*(f)\right] \end{aligned} \tag{2.29}$$

The derivative of J with respect to $H(f)$ is set equal to zero in order to minimise the mean-square error.

$$\begin{aligned} \frac{\partial J}{\partial H(f)} &= H^*(f)\mathcal{E}\left[S(f)S^*(f)\right] - \mathcal{E}\left[S(f)\Delta^*(f)\right] \\ &= \left[H(f)\mathcal{E}\left[S^*(f)S(f)\right] - \mathcal{E}\left[S^*(f)\Delta(f)\right]\right]^* \\ &= 0 \end{aligned} \tag{2.30}$$

Solving Equation (2.30) for $H(f)$, the general form of Wiener filters is derived as

$$H(f) = \frac{\mathcal{E}\left[\Delta(f)S^*(f)\right]}{\mathcal{E}\left[|S(f)|^2\right]} = \frac{P_{\delta s}(f)}{P_{ss}(f)} \tag{2.31}$$

where $P_{ss}(f)$ and $P_{\delta s}(f)$ represent the power spectrum of $s(n)$ and the cross-power spectrum of $\delta(n)$ and $s(n)$ respectively.

2.3.2.2 Wiener Filtering for Speech Enhancement

In an application of speech enhancement, Equations (2.26) and (2.27) are described as

$$\epsilon = x(n) - \hat{x}(n) \tag{2.32}$$

$$= x(n) - \sum_{k=0}^{P-1} h_k y(n-k) \tag{2.33}$$

where $y(n)$, $x(n)$ and $\hat{x}(n)$ correspond to the noisy speech, underlying clean speech and the estimate of the clean speech respectively. Applying Equations (2.5) - (2.9), the frequency response of the wiener filter at i-th frame, $H_i(f)$, is derived by referring to

Equation (2.31) as

$$
\begin{aligned}
H_i(f) &= \frac{\mathcal{E}\left[X_i(f)Y_i^*(f)\right]}{\mathcal{E}\left[|Y_i(f)|^2\right]} && (2.34)\\
&= \frac{\mathcal{E}\left[(X_i(f)(X_i(f)+D_i(f))^*\right]}{\mathcal{E}\left[(X_i(f)+D_i(f))(X_i(f)+D_i(f))^*\right]} && (2.35)\\
&= \frac{\mathcal{E}\left[|X_i(f)|^2\right]+\mathcal{E}\left[X_i(f)D_i^*(f)\right]}{\mathcal{E}\left[|X_i(f)|^2\right]+\mathcal{E}\left[|D_i(f)|^2\right]+\mathcal{E}\left[X_i(f)D_i^*(f)\right]+\mathcal{E}\left[D_i(f)X_i^*(f)\right]} && (2.36)\\
&= \frac{\mathcal{E}\left[|X_i(f)|^2\right]}{\mathcal{E}\left[|X_i(f)|^2\right]+\mathcal{E}\left[|D_i(f)|^2\right]} && (2.37)
\end{aligned}
$$

where the cross-power spectra of the clean speech and noise are equal to zero because they are assumed to be independent each other. $H_i(f)$ can be also expressed as a function of the *a priori* SNR, $\xi_i(f)$.

$$
\begin{aligned}
H_i(f) &= \frac{\xi_i(f)}{\xi_i(f)+1} && (2.38)\\
\xi_i(f) &= \frac{\mathcal{E}\left[|X_i(f)|^2\right]}{\mathcal{E}\left[|D_i(f)|^2\right]} && (2.39)
\end{aligned}
$$

In practice, the value of $\xi_i(f)$ is unknown and thus, [30] proposes the following decision-directed method to estimate the *a priori* SNR, $\hat{\xi}_i(f)$.

$$
\hat{\xi}_i(f) = \alpha\frac{|\hat{X}_{i-1}(f)|^2}{|\hat{D}_{i-1}(f)|^2} + (1-\alpha)\max\left(\frac{|Y_i(f)|^2}{|\hat{D}_i(f)|^2} - 1, 0\right) \qquad (2.40)
$$

where $|\hat{D}_i(f)|^2$, $|\hat{X}_i(f)|^2$ and α represent the estimate of the noise power spectrum obtained with the methods introduced in Section 2.2, the enhanced speech at frame i and a weight constant respectively. Equation (2.40) derives the estimate of the *a priori* SNR as a weighted moving average of the past *a priori* SNR and the present *a posteriori* SNR with a compensation for the case of the estimated power being negative.

In general, $|\ \hat{X}_{i-1}(f)\ |^2$ in Equation (2.40) is derived as $(\mathcal{E}\left[X_{i-1}(f)\right])^2$ rather than $\mathcal{E}\left[|\ X_{i-1}(f)\ |^2\right]$ by a speech enhancement algorithm. This causes a bias in the estimate of *a priori* SNR. Therefore, the following modification to the decision-directed approach has been recommended in order to reduce the influence of this bias [31].

$$
\hat{\xi}_i(f) = \max\left[\alpha\frac{|\hat{X}_{i-1}(f)|^2}{|\hat{D}_{i-1}(f)|^2} + (1-\alpha)\left(\frac{|Y_i(f)|^2}{|\hat{D}_i(f)|^2} - 1\right), \xi_{\min}\right] \qquad (2.41)
$$

where $\xi_{\min}$ denotes the minimum value allowed for $\xi_i(f)$. Different approaches to estimate low-variance SNR are proposed in addition to the preceding methods [1, 32].

Equations (2.37) and (2.38) are the underlying principles to optimise Wiener filters, and several derivative algorithms have been proposed, for example, [33] generalises the Wiener filtering as the parametric Wiener filters

$$H_i(f) = \left(\frac{P^i_{xx}(f)}{P^i_{xx}(f) + \alpha P^i_{dd}(f)} \right)^{\beta} \tag{2.42}$$

where $P^i_{xx}(f)$ and $P^i_{dd}(f)$ represent the power spectrum of $x(n)$ and $d(n)$ at i-th frame respectively, and the algorithm is parameterised by α and β.

The spectrum of the enhanced speech, $\hat{X}_i(f)$, is derived as

$$\hat{X}_i(f) = H_i(f) Y_i(f) \tag{2.43}$$

Moreover, an iterative wiener filtering algorithm in which $H_i(f)$ is renewed by the derived enhanced speech, $\hat{X}_i(f)$, recursively has also been proposed for speech enhancement [1].

2.3.3 Statistical-Model-Based Method

The Wiener filters in the previous section formed an optimised linear model between the complex spectra of the noisy and clean speech in terms of mean-square error. This section has a discussion about filtering algorithms which construct nonlinear statistical models between the magnitude of the clean and noisy speech.

Various techniques to build nonlinear statistical estimators have been proposed [1], and they are largely categorised into the methods based on the maximum-likelihood (ML) approach or the Bayesian approach. The first part of this section describes the overview of the ML estimator while the latter part shows the overview of the log-MMSE estimator as a representative of the Bayesian estimators.

2.3.3.1 Maximum-Likelihood Estimator

Supposing the speech signals are under the conditions of Equations (2.5)-(2.9) and (2.14), an ML estimator is derived with the hypothesis where the probability density function (pdf) of the noisy speech spectrum, $Y_i(f)$, is parametrised by the clean speech spectrum,

$X_i(f)$, and thus, the clean speech spectrum is estimated as follows [34].

$$\hat{X}_i(f) = \arg\max_{X_i(f)} p(Y_i(f); X_i(f)) \tag{2.44}$$

where $\hat{X}_i(f)$ and $p(Y_i(f); X_i(f))$ denote the estimate of the clean speech spectrum and the pdf of the noisy speech spectrum parameterised by the clean speech spectrum.

In the ML approach, $X_i(f)$ is assumed to be deterministic and the noise spectrum $D_i(f)$ is assumed to be zero-mean, complex Gaussian whose real and imaginary parts have variances of $\lambda_d^i(f)/2$. These assumptions give the pdf of $Y_i(f)$ as

$$p(Y_i(f); |X_i(f)|, \Phi_x^i(f)) = \frac{1}{\pi\lambda_d^i(f)} \exp\left[-\frac{|Y_i(f) - |X_i(f)|e^{j\Phi_x^i(f)}|^2}{\lambda_d^i(f)} \right] \tag{2.45}$$

The phase parameter is integrated to be eliminated from the parameters.

$$p_L(Y_i(f); |X_i(f)|) = \int_0^{2\pi} p(Y_i(f); |X_i(f)|, \Phi_x^i(f)) p(\Phi_x^i(f)) d\Phi_x^i(f) \tag{2.46}$$

Assuming the phase $\Phi_x^i(f)$ has a uniform distribution between $[0, 2\pi]$, the likelihood function is derived as

$$\begin{aligned} p_L(Y_i(f); |X_i(f)|) &= \frac{1}{\pi\lambda_d^i(f)} \exp\left[-\frac{|Y_i(f)|^2 + |X_i(f)|^2}{\lambda_d^i(f)} \right] \\ &\quad \cdot \frac{1}{2\pi} \int_0^{2\pi} \exp\left[\frac{2|X_i(f)| \Re\left(e^{-j\Phi_x^i} Y_i(f) \right)}{\lambda_d^i(f)} \right] d\Phi_x^i(f) \end{aligned} \tag{2.47}$$

Exploiting the modified Bessel function of the first kind [34], the preceding equation is simplified as

$$\begin{aligned} p_L(Y_i(f); |X_i(f)|) &= \frac{1}{\pi\lambda_d^i(f) \sqrt{2\pi \frac{2|X_i(f)||Y_i(f)|}{\lambda_d^i(f)}}} \\ &\quad \cdot \exp\left[\frac{-|Y_i(f)|^2 + |X_i(f)|^2 - 2|X_i(f)||Y_i(f)|}{\lambda_d^i(f)} \right] \end{aligned} \tag{2.48}$$

The derivative of the log-likelihood function, $\log p_L(Y_i(f); |X_i(f)|)$, with respect to $|X_i(f)|$ is set equal to zero in order to maximise the log-likelihood, and then, solving

for $|X_i(f)|$, the ML estimate of the clean spectral magnitude is derived as

$$|\hat{X}_i(f)| = \frac{1}{2}\left[|Y_i(f)| + \sqrt{|Y_i(f)|^2 - \lambda_d^i(f)}\right] \tag{2.49}$$

As the phase spectrum of the clean speech is unknown, the phase spectrum of the noisy speech is combined with the estimate of the clean magnitude spectrum in order to obtain the complex spectrum of the enhanced speech as well as the process in the spectral subtraction.

$$\begin{aligned}
\hat{X}_i(f) &= |\hat{X}_i(f)|e^{j\Phi_y^i(f)} = |\hat{X}_i(f)|\frac{Y_i(f)}{|Y_i(f)|} && (2.50)\\
&= \left[\frac{1}{2} + \frac{1}{2}\sqrt{\frac{|Y_i(f)|^2 - \lambda_d^i(f)}{|Y_i(f)|^2}}\right] Y_i(f) && (2.51)\\
&= \left[\frac{1}{2} + \frac{1}{2}\sqrt{\frac{\gamma_i(f) - 1}{\gamma_i(f)}}\right] Y_i(f) && (2.52)\\
\gamma_i(f) &= \frac{|Y_i(f)|^2}{\lambda_d^i(f)} && (2.53)
\end{aligned}$$

where $\gamma_i(f)$ represents the *a posteriori* SNR

2.3.3.2 Log-MMSE estimator

In the maximum-likelihood approach, the clean speech spectrum is assumed to be deterministic but unknown. This section discusses an estimator using the Bayesian approach in which the spectrum of the clean speech is assumed to be a random variable, and the *a priori* knowledge about the magnitude spectrum of the clean speech $p(|X_i(f)|)$ is given to the estimator. Several methods using the Bayesian approach have been proposed such as the MMSE magnitude estimator, log-MMSE estimator and maximum *a posteriori* (MAP) estimator [1, 35–38]. This section specifically explores the log-MMSE estimator as a representative of the Bayesian estimators which gives the best performance both objectively and subjectively in the statistical-model-based methods [1, 39].

The log-MMSE method forms a statistical model to minimises the mean-square error between the estimate and the true value of the magnitude spectrum of the clean speech

in the log-magnitude domain.

$$\hat{X}_i(f) = \arg\min_{\tilde{X}_i(f)} \left(\mathcal{E}\left[(\log|X_i(f)| - \log|\tilde{X}_i(f)|)^2 \right] \right) \tag{2.54}$$

Thus, given the complex spectrum of the noisy speech, $Y_i(f)$, the log-MMSE estimator is derived as

$$\log|\hat{X}_i(f)| = \mathcal{E}\left[\log|X_i(f)| \mid Y_i(f)\right] \tag{2.55}$$

$$|\hat{X}_i(f)| = \exp\left(\mathcal{E}\left[\log|X_i(f)| \mid Y_i(f)\right]\right) \tag{2.56}$$

Let $Z_f = \log X_i(f)$, and the moment-generating function of Z_f is given in order to evaluate the conditional expectation in the preceding equation.

$$\Phi_{Z_f|Y_i(f)}(\mu) = \mathcal{E}\left[\exp(\mu Z_f)\,|Y_i(f)\right] \tag{2.57}$$

$$= \mathcal{E}\left[|X_i^{\mu}(f)| \mid Y_i(f)\right]] \tag{2.58}$$

$\mathcal{E}\left[\log|X_i(f)| \mid Y_i(f)\right]$ can be obtained by the derivative of the moment-generating function at $\mu = 0$.

$$\mathcal{E}\left[\log|X_i(f)| \mid Y_i(f)\right] = \frac{d}{d\mu}\Phi_{Z_f|Y_i(f)}(\mu)\,\Big|_{\mu=0} \tag{2.59}$$

$$= \frac{1}{2}\log\frac{\mathcal{E}\left[|X_i(f)|^2\right]}{1+\xi_i(f)} + \frac{1}{2}\log\nu_i(f) + \frac{1}{2}\int_{\nu_i(f)}^{\infty}\frac{e^{-t}}{t}dt \tag{2.60}$$

where $\nu_i(f)$, $\xi_i(f)$ and $\gamma_i(f)$ are defined by

$$\nu_i(f) = \frac{\xi_i(f)}{1+\xi_i(f)}\gamma_i(f) \tag{2.61}$$

$$\xi_i(f) = \frac{\mathcal{E}\left[|X_i(f)|^2\right]}{\mathcal{E}\left[|D_i(f)|^2\right]} \tag{2.62}$$

$$\gamma_i(f) = \frac{|Y_i(f)|^2}{\mathcal{E}\left[|D_i(f)|^2\right]} \tag{2.63}$$

The log-MMSE estimator is obtained by substituting Equation (2.60) into (2.56).

$$|\hat{X}_i(f)| = \frac{\xi_i(f)}{1+\xi_i(f)}\exp\left\{\frac{1}{2}\int_{\nu_i(f)}^{\infty}\frac{e^{-t}}{t}dt\right\}|Y_i(f)| \tag{2.64}$$

The preceding equation shows the log-MMSE is parameterised by both *a priori* SNR, $\xi_i(f)$ and *a posteriori* SNR, $\gamma_i(f)$. Thus, the log-MMSE estimator can be expressed as

$$|\hat{X}_i(f)| = G\left(\xi_i(f), \gamma_i(f)\right)|Y_i(f)| \tag{2.65}$$

where $G(\xi_i(f), \gamma_i(f))$ represents a gain function of the log-MMSE estimator. In practice, the value of the *a priori* SNR is unknown and thus, it is estimated with, for example, the decision-directed method given by Equation (2.40).

2.3.4 Subspace Algorithm

The subspace algorithms transform the noisy speech into a new space that comprises speech and noise subspaces [40, 41]. Elimination of the noise subspace can retain speech components and remove noise components. Thus, the subspace algorithms do not require the noise estimation process unlike the other filtering algorithms mentioned above. However, retaining too few speech components oversmooths the speech while retaining too many components leaves residual noise.

Considering vectors representing the clean and noisy speech and noise, the noisy speech is determined with the vectors as

$$\mathbf{y}_i = \mathbf{x}_i + \mathbf{d}_i \tag{2.66}$$

$$\mathbf{x}_i = [x(n+(i-1)L), x(n+(i-1)L+1, \ldots, x(n+iL-1)]^T \tag{2.67}$$

$$\mathbf{y}_i = [y(n+(i-1)L), y(n+(i-1)L+1, \ldots, y(n+iL-1)]^T \tag{2.68}$$

$$\mathbf{d}_i = [d(n+(i-1)L), d(n+(i-1)L+1, \ldots, d(n+iL-1)]^T \tag{2.69}$$

where $x(n)$, $y(n)$, $d(n)$, i, and L denote the clean and noisy speech, noise, frame index and frame length respectively. The clean speech vectors constitute a speech subspace $\mathbf{X}$ as an $L \times M$ matrix.

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M] \tag{2.70}$$

Assuming the speech and noise are independent of each other, $\mathbf{x}_i$ and $\mathbf{d}_i$ are assumed to be orthogonal. Thus, they can be decoupled from $\mathbf{y}_i$ by projecting $\mathbf{y}_i$ into the subspace $\mathbf{X}$ and the orthogonal subspace to $\mathbf{X}$, namely noise subspace. This projection is given

by a projection matrix, $\mathbf{P}$, determined by

$$\mathbf{P} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \tag{2.71}$$

For simplification, $\mathbf{X}$ can be decomposed by singular value decomposition (SVD) as

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H \tag{2.72}$$

where $\mathbf{X}$ is assumed to be full column rank such that $\text{rank}(\mathbf{X}) = M$, $\mathbf{U}$ is an $L \times L$ unitary matrix consisting of eigenvectors of $\mathbf{X}\mathbf{X}^T$, $\mathbf{V}$ is an $M \times M$ unitary matrix comprising eigenvectors of $\mathbf{X}^T\mathbf{X}$ and $\mathbf{\Sigma}$ is an $L \times M$ diagonal matrix comprising the singular values of $\mathbf{X}$. Equations (2.71) and (2.72) leads to

$$\mathbf{P} = \mathbf{U}\mathbf{U}^H \tag{2.73}$$

This projection matrix divides $\mathbf{x}_i$ and $\mathbf{d}_i$ from $\mathbf{y}_i$ as

$$\mathbf{x}_i = \mathbf{U}\mathbf{U}^H\mathbf{y}_i \tag{2.74}$$

$$\mathbf{d}_i = (\mathbf{I} - \mathbf{U}\mathbf{U}^H)\mathbf{y}_i \tag{2.75}$$

If $\mathbf{X}$ is assumed not to be full rank, i.e. $\text{rank}(\mathbf{X}) = r < M$, Equation (2.72) is expressed as

$$\mathbf{X} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{bmatrix} = \mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^H \tag{2.76}$$

where $\mathbf{U}_1$, $\mathbf{U}_2$, $\mathbf{V}_1$ and $\mathbf{V}_2$ are $N \times r$, $N \times (N-r)$, $r \times M$ and $(M-r) \times M$ matrices respectively extracted from $\mathbf{U}$ and $\mathbf{V}$. $\mathbf{\Sigma}_1$ is a $r \times r$ diagonal matrix comprising the singular values of $\mathbf{X}$. Equations (2.71) and (2.76) lead to

$$\mathbf{P} = \mathbf{U}_1\mathbf{U}_1^H \tag{2.77}$$

Alternatively, the following is derived exploiting the unitarity of $\mathbf{U}$.

$$\mathbf{U}\mathbf{U}^H = \begin{bmatrix}\mathbf{U}_1 & \mathbf{U}_2\end{bmatrix}\begin{bmatrix}\mathbf{U}_1^H \\ \mathbf{U}_2^H\end{bmatrix} = \mathbf{U}_1\mathbf{U}_1^H + \mathbf{U}_2\mathbf{U}_2^H = \mathbf{I} \tag{2.78}$$

Therefore, another projection matrix, $\mathbf{Q}$, projecting $\mathbf{y}$ into the noise subspace is given by

$$\mathbf{Q} = \mathbf{I} - \mathbf{U}_1\mathbf{U}_1^H = \mathbf{U}_2\mathbf{U}_2^H \tag{2.79}$$

The above derives the underlying principle of the subspace algorithms for speech enhancement as

$$\mathbf{x}_i = \mathbf{U}_1\mathbf{U}_1^H\mathbf{y}_i \tag{2.80}$$

$$\mathbf{d}_i = \mathbf{U}_2\mathbf{U}_2^H\mathbf{y}_i \tag{2.81}$$

In empirical conditions, however, the speech and noise spaces are not entirely separable particularly with the coloured noise, therefore, it is necessary to embed a further filtering algorithm to remove the residual noise [42].

2.3.5 Experimental Results and Analysis

Various types of the filtering-based methods are discussed above and those different approaches to noise filtering bring different properties to the result of speech enhancement. It is important to understand the performance and limitations of each method prior to concluding the discussion of filtering-based speech enhancement. Therefore, this section examines the performance of filtering-based speech enhancement in noisy conditions and then evaluates the results in terms of speech quality and intelligibility objectively.

Experiments use speech from four speakers in the GRID database [43] (two males and two females) which is down-sampled to 8 kHz assuming telephony applications. From the 1000 utterances from each speaker, 200 are used for the tests. The test speech is contaminated with each of white noise and babble noise at SNRs from -5 dB to 10 dB. Then the noisy speech is first divided into 25 ms-frames with 50 % overlap by a Hamming window and then, the noise power spectrum, $|\hat{D}_i(f)|^2$, at the i-th frame is estimated by

using VAD-based estimation after 1024 point DFT as

$$| \hat{D}_i(f) |^2 = \begin{cases} | Y_i(f) |^2 & i = 0 \\ \alpha | \hat{D}_{i-1}(f) |^2 + (1-\alpha) | Y_i(f) |^2 & i > 0, \quad \hat{\gamma}_i < 3 \\ | \hat{D}_{i-1}(f) |^2 & i > 0, \quad \hat{\gamma}_i \geqslant 3 \end{cases} \tag{2.82}$$

$$\hat{\gamma}_i = 10 \log_{10} \frac{\| Y_i(f) \|^2}{\| \hat{D}_{i-1}(f) \|^2} \tag{2.83}$$

where $|Y_i(f)|^2$ represents the power spectrum of the observed noisy speech at the i-th frame, and α is set equal to 0.9. After the noise estimation the noisy speech is enhanced by four types of the filtering methods in Table 2.1. The log-MMSE is based on Equation

LOG:	Log MMSE
WIN:	Wienner Filter
SS:	Spectral Subtraction
SUB:	Subspace Algorithm

Table 2.1: Filtering-based methods for the tests

(2.64) while the Wiener filter is based on Equation (2.38), and the *a priori* SNR is estimated with Equation (2.40) in both methods ($\alpha = 0.98$). The spectral subtraction is based on Equation (2.16) where

$$\alpha = \begin{cases} 5 & \hat{\gamma}_i < -5 \\ 1 & \hat{\gamma}_i > 20 \\ 4 & \text{otherwise} \end{cases} \tag{2.84}$$

The subspace algorithm is based on Equation (2.80) with built-in pre-whitening [42].

2.3.5.1 Speech Quality

As mentioned in Section 1.1, speech enhancement is concerned with improving some perceptual aspect of speech degraded by noise, and noise in speech brings two main effects on perception of speech. The first is to degrade quality of speech and the second is to reduce intelligibility of speech. Therefore, speech quality and intelligibility are regarded as the most important attributes to gauge the performance of speech enhancement and have widely been used to evaluate speech signals. Speech quality generally gauges how

a speaker produces an utterance while speech intelligibility measures what the speaker said. These measures are attributed to many factors and the connection to the acoustic features of the speech has not been fully discovered yet [1]. Therefore, subjective listening tests, such as the mean opinion score (MOS) test to gauge speech quality and speech identifying test to measure speech intelligibility, are more reliable than objective tests to evaluate speech enhancement [44]. However, a number of objective measures have been proposed to predict the subjective measures and some of them have good correlation with subjective measures of speech quality or intelligibility. Evaluation across a range of objective measures shows that the perceptual evaluation of speech quality (PESQ) and the frequency-weighted segmental SNR (fwSNRseg) achieve the highest correlation with speech quality while the coherence speech intelligibility index (CSII) and the normalised-covariance measure (NCM) performs the best for speech intelligibility [45]. The results of the experiments in this section are objectively scored with PESQ and NCM to evaluate speech quality and intelligibility respectively.

Figure 2.2 shows the performance of the four filtering-based algorithms comparing with baseline performance given by no noise compensation (NNC) in terms of PESQ at different SNRs in white noise and babble noise. In white noise, SUB and LOG are

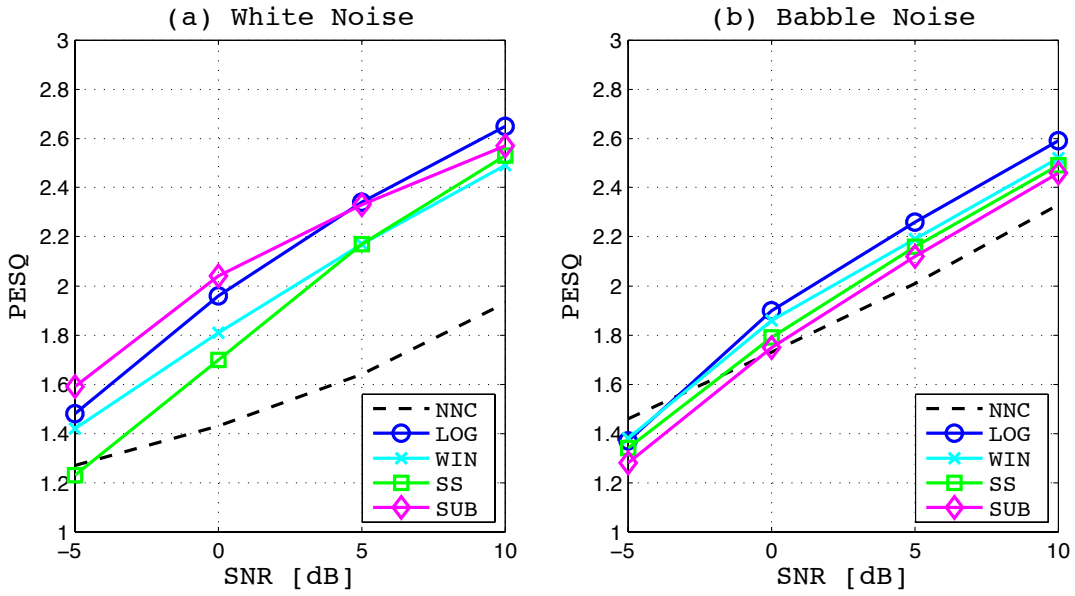

Figure 2.2: PESQ scores of different filtering-based methods at different SNRs in a) white noise, b) babble noise.

superior to the other methods over the SNR range. Specifically, SUB shows the higherst scores at SNRs of 0 dB and -5 dB while LOG shows the best performance at SNRs of 5 dB and 10 dB. WIN performs with higher scores than SS between -5 dB and 5 dB but SS becomes higher than WIN at 10 dB.

In babble noise, however, SUB shows always the lowest of the four methods over the SNR range as opposed to LOG showing always the highest scores followed by WIN an SS respectively. This is attributed to the fact that noise and speech are not sufficiently orthogonal in babble noise, and thus, noise and speech are not transformed to the proper subspace in this condition.

As the overall evaluation in terms of PESQ, LOG shows the best performance of the four methods while the worst is SS. Superiority between WIN and SUB depends on attributes of noise. Incidentally, even the best method reduces the score below 1.6 at -5 dB in white noise and below 1.4 at -5 dB in babble noise. This implies the filtering-based methods do not show their effectiveness at low SNRs such as below 0 dB.

Comparing with NNC, the effectiveness of each method for speech enhancement in babble noise is less than the case of white noise.

2.3.5.2 Speech Intelligibility

Figure 2.3 shows the performance of the four filtering-based methods comparing with baseline performance given by NNC in terms of NCM at different SNRs in white noise and babble noise. The performance of SUB looks superior to the others as the overall evaluation, but all of the four methods reduce the score at lower SNRs and cannot retain sufficient intelligibility. For example, NCM score of SUB falls below 0.6 at SNR of -5 dB and the other methods become below 0.5 in white noise. Moreover, scores of all the methods fall far below 0.5 at -5 dB in babble noise. The general tendency of speech intelligibility of each method at different SNRs does not show significant difference from NNC.

2.3.5.3 Spectral Analysis

To give further insight into filtering-based speech enhancement, Figures 2.4 and 2.5 depict narrowband spectrograms of an utterance, *"Bin Blue At E Seven Now"*, spoken by a

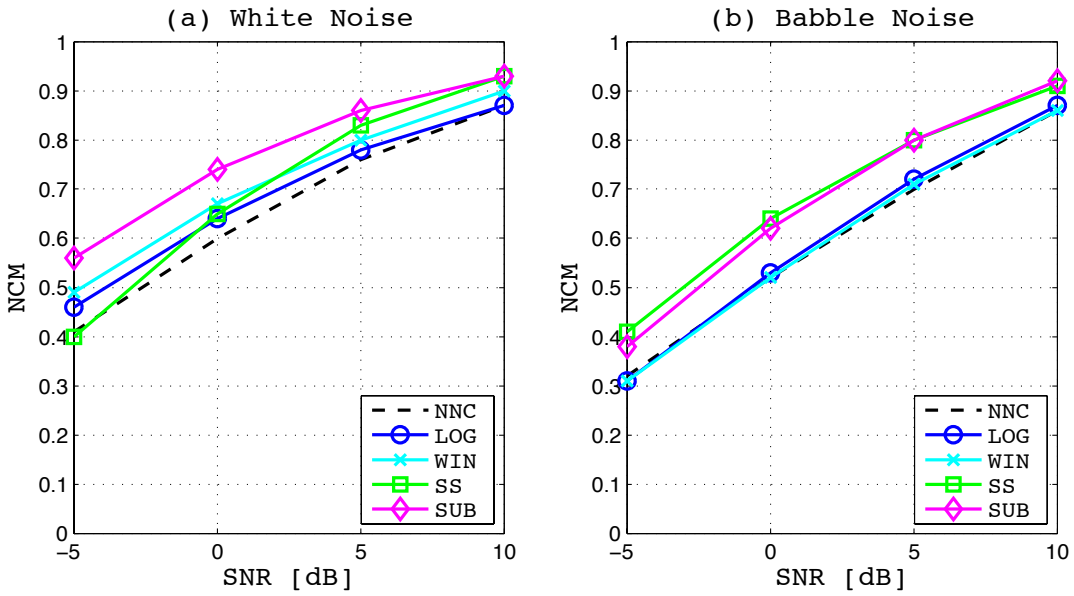

Figure 2.3: NCM scores of different filtering-based methods at different SNRs in a) white noise, b) babble noise.

male speaker in white noise and babble noise at SNRs of 10 dB and -5 dB. The figures show that large parts of spectral envelopes and harmonic information still remain among residual and musical noise after the process of each methods at SNR of 10 dB. However, at SNR of -5 dB, those are masked under residual noise or eliminated leaving musical noise especially in the frequency band above 1.5 kHz. These degradation are brought by overestimation and underestimation of the noise. The subspace method estimates the noise space on the assumption that it is orthogonal to the speech space rather than using VAD. Therefore, spectral information remains with less estimation errors even at SNR of -5 dB as long as the noise is orthogonal to the speech (i.e., white noise). However, it loses most of the de-noising function when the noise does not have orthogonality to the speech such as the case of babble noise.

The experiments show that the filtering-based methods are effective to reduce the noise at relatively high SNRs but but those performance are insufficient at low SNRs such as 0 dB and below. This brings a motivation to discuss the reconstruction-based speech enhancement as an alternative approach to the filtering-based methods.

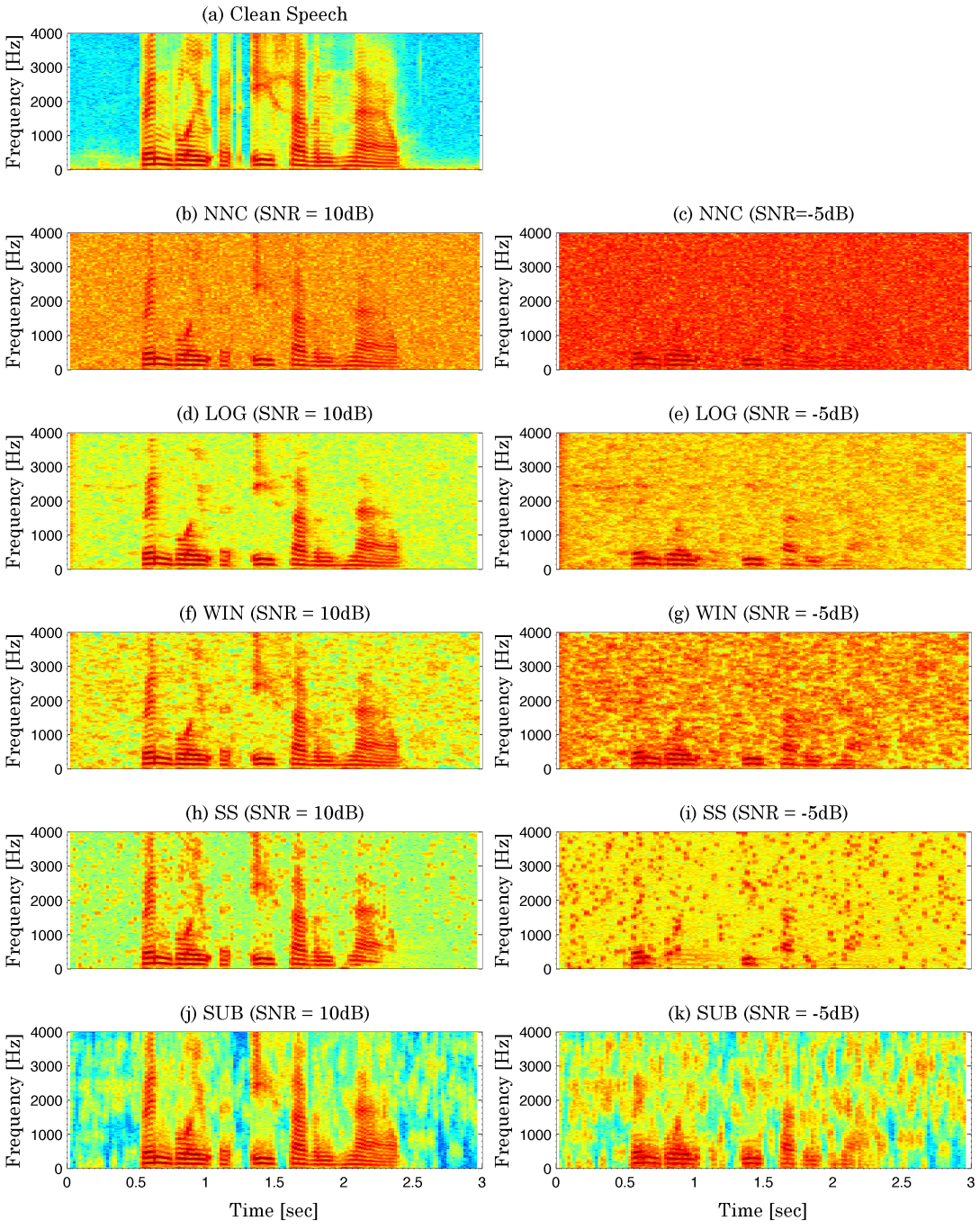

Figure 2.4: Narrowband spectrograms of an utterance, *"Bin Blue At E Seven Now"*, spoken by a male speaker in white noise. a) shows clean speech, b) and c) show noisy speech with no enhancement at SNR of 10dB and -5dB, and d), f), h), and j) show noisy speech at SNR of 10 dB enhanced by LOG, WIN, SS and SUB while c), e), g), i) and k) show noisy speech at SNR of -5 dB enhanced by LOG, WIN, SS and SUB.

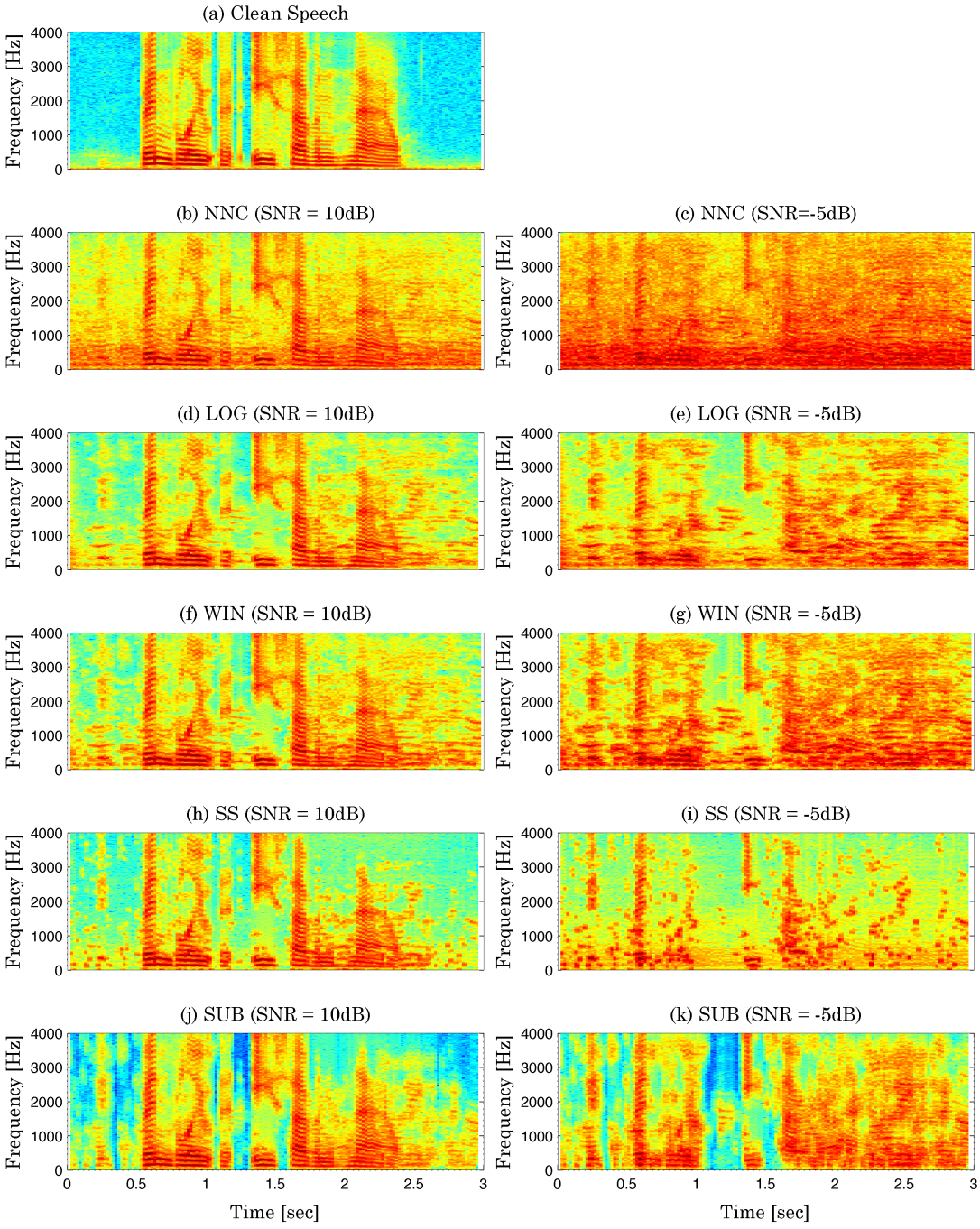

Figure 2.5: Narrowband spectrograms of an utterance, *"Bin Blue At E Seven Now"*, spoken by a male speaker in babble noise. a) shows clean speech, b) and c) show noisy speech with no enhancement at SNR of 10dB and -5dB, and d), f), h), and j) show noisy speech at SNR of 10 dB enhanced by LOG, WIN, SS and SUB while c), e), g), i) and k) show noisy speech at SNR of -5 dB enhanced by LOG, WIN, SS and SUB.

2.4 Reconstruction-Based Speech Enhancement

The spectral subtraction, Wiener filters and statistical-model-based methods described in the preceding sections are based on largely two processes of first estimating noise components or SNR from speech-inactive periods in the noisy speech, and then forming a linear or nonlinear filter to eliminate the noise. Therefore, these methods are classified as filtering methods. Subspace algorithms decompose the noisy speech into two orthogonal spaces, namely a signal space and a noise space by using SVD, and then the enhanced speech is obtained by employing the signal space. This can also be regarded as one of the filtering methods as well.

More recently, several alternative approaches, in which the enhanced speech is synthesised or reconstructed by exploiting features extracted from the noisy speech, have been developed. This section first discusses corpus and inventory-based speech enhancement as an example applying this new approach with speech reconstruction using an inventory or corpus of wide range of clean speech segments [10–12] while the latter part of the section explores another approach using a model-based speech reconstruction for the reconstruction process [5–7, 13]. These speech reconstruction-based methods require an offline training process in addition to the feature extraction and reconstruction processes for implementation while the filtering methods can be implemented as a complete real-time process.

2.4.1 Corpus and Inventory-based Speech Enhancement

Figure 2.6 shows a framework of corpus and inventory-based speech enhancement [12]. The input noisy speech, $y(n)$, is transformed to MFCC vectors after divided into short-duration frames. The input waveform and MFCC vector at i-th frame are denoted as

$$\mathbf{y}_i = [y(n+(i-1)L), y(n+(i-1)L+1), \ldots, y(n+iL-1)]^T \tag{2.85}$$

$$\mathbf{c}_i = \text{MFCC}\{\mathbf{y}_i\} \tag{2.86}$$

where L represents the frame length. $\mathbf{c}_i$ is referred to the noisy speech codebook associated with the clean speech codebook forming a network of Hidden Markov Models (HMMs) in order to estimate the state of HMMs, $s(i)$, which represents the most likely

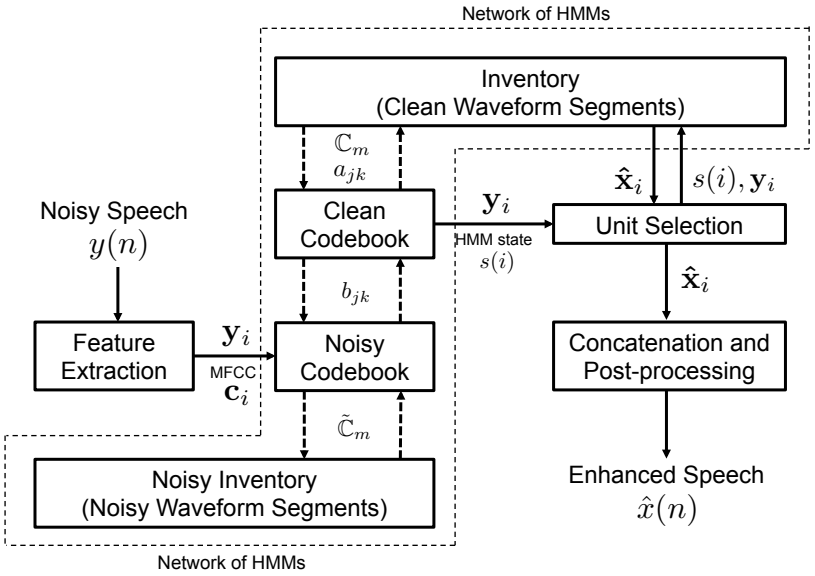

Figure 2.6: A framework of corpus and inventory-based speech enhancement.

waveform cluster in the clean inventory. A unit selection process selects $\hat{\mathbf{x}}_i$ as the closest inventory element to $\mathbf{y}_i$ from the cluster corresponding to $s(i)$, and then the estimates of all the frames, $\hat{\mathbf{x}}_i$ for $\forall i$, are concatenated and post-processed to reconstruct the enhanced speech, $\hat{x}(n)$.

2.4.1.1 System training

System training is an offline process to build the network of HMMs, which is the part enclosed by broken lines in Figure 2.6. A training data set consists of both clean speech, $x'(n)$, and noisy speech, $y'(n)$, contaminated with the target noise, $d'(n)$.

$$y'(n) = x'(n) + d'(n) \tag{2.87}$$

Each frame of the clean and noisy training data is converted to MFCC vectors in order to discriminate and classify the feature of the frames.

$$\mathbf{x'}_i = \left[x'(n+(i-1)L, x'(n+(i-1)L+1, \ldots, x'(n+iL-1)\right]^T \tag{2.88}$$

$$\mathbf{y'}_i = \left[y'(n+(i-1)L, y'(n+(i-1)L+1, \ldots, y'(n+iL-1)\right]^T \tag{2.89}$$

$$\mathbf{d'}_i = \left[d'(n+(i-1)L, d'(n+(i-1)L+1, \ldots, d'(n+iL-1)\right]^T \tag{2.90}$$

$$\mathbf{c'}_i = \text{MFCC}\left\{\mathbf{x'}_i\right\} \tag{2.91}$$

$$\tilde{\mathbf{c}}'_i = \text{MFCC}\left\{\mathbf{y'}_i\right\} \tag{2.92}$$

$\mathbf{c'}_i$ is first divided into speech-active/inactive group using a VAD and these two groups are completely separated in the inventory, then classified with intra-phonemic clustering to form the speech segments with the adjacent similar frames as follows.

$$\bar{\mathbf{c}'}_{\mathbb{I}_j} = \frac{1}{N_j} \sum_{i \in \mathbb{I}_j} \mathbf{c'}_i \tag{2.93}$$

where $\bar{\mathbf{c}'}_{\mathbb{I}_j}$ denotes the centroid of cluster, $\mathbb{I}_j$, j represents cluster index and N_j is the number of elements in $\mathbb{I}_j$. $\mathbf{c'}_i$ is included into $\mathbb{I}_j$ if the both Equations (2.94) and (2.95) are satisfied.

$$\|\bar{\mathbf{c}'}_{\mathbb{I}_j} - \mathbf{c'}_i\| < \lambda \tag{2.94}$$

$$\|\bar{\mathbf{c}'}_{\mathbb{I}_j \cup \mathbf{c'}_i} - \mathbf{c'}_k\| < \lambda \quad \text{for } \forall \mathbf{c'}_k \in \mathbb{I}_j \tag{2.95}$$

where λ represents a threshold value. If $\mathbf{c'}_i$ fails to satisfy the conditions, j should be incremented and $\mathbf{c'}_i$ becomes the first frame of the $\mathbb{I}_{j+1}$. The above procedure is iterated with the increment of i.

The inventory has to have a sufficient number of clusters, namely sufficient codebook size, in order to reconstruct the clean speech with good intelligibility. However, the number of the clusters produced by intra-phonemic clustering, nevertheless, needs to be quantised into an appropriate size so that each cluster, which corresponds to each HMM state, can occur with enough frequency during training to construct the network of HMMs with sufficient quality. Therefore, inter-phonemic clustering is operated to combine the

intra-phonemic clusters having similar characteristics each other. The followings are an example of vector quantisation using the k-means algorithm [12]. The first two seeds of the inter-phonemic clusters are chosen as follows.

$$(m_1, m_2) = \arg\max_{\forall j, \forall k} \|\bar{\mathbf{c}}_{\mathbb{I}_j} - \bar{\mathbf{c}}_{\mathbb{I}_k}\| \tag{2.96}$$

$\mathbb{I}_{m_1}$ and $\mathbb{I}_{m_2}$ become the first element of the seeds $\mathbb{C}_m$ $(m = 1, \ldots, M : M = 2)$ and then, all the intra-phonemic clusters, $\mathbb{I}_j$ for $\forall j$, are divided into $\mathbb{C}_m$ according to Equation (2.97).

$$\mathbb{I}_j \in \mathbb{C}_p \quad \text{as} \quad p = \arg\min_{m=1,\ldots,M} \|\bar{\mathbf{c}}_{\mathbb{I}_j} - \bar{\mathbf{c}}_{\mathbb{C}_m}\| \quad \text{for } \forall j \tag{2.97}$$

The next seed is created by the element of $z_{m'}$.

$$z_m = \arg\max_{\forall \mathbb{I}_j \in \mathbb{C}_m} \|\bar{\mathbf{c}}_{\mathbb{I}_j} - \bar{\mathbf{c}}_{\mathbb{C}_m}\| \qquad (m = 1, \ldots, M) \tag{2.98}$$

$$m' = \arg\max_{m=1,\ldots,M} \|z_m - \bar{\mathbf{c}}_{\mathbb{C}_m}\| \tag{2.99}$$

The operation from Equation (2.97) to (2.99) is iterated with increment of M until the number of the clusters amounts to the appropriate size.

The seeds of the inter-phonemic clusters, $\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_M$ are optimised using the k-means clustering algorithm [46]. The centroids of the optimised clusters, $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M\}$, forms the clean codebook. The noisy codebook, $\tilde{\mathcal{C}} = \left\{\tilde{\mathcal{C}}_1, \tilde{\mathcal{C}}_2, \ldots, \tilde{\mathcal{C}}_M\right\}$, is also constituted with noisy training data, $\tilde{\mathbf{c}}'_i$, in the same manner. In addition, the framed clean waveforms, $\mathbf{x}'_i$, are also classified into the clusters, $\mathbb{D} = \{\mathbb{D}_1, \mathbb{D}_2, \ldots, \mathbb{D}_M\}$, in the clean inventory. The elements in a waveform cluster $\mathbb{D}_m$ are associated with the elements in the corresponding optimised inter-phonemic cluster $\mathbb{C}_m$.

$$\mathbf{x}'_i \in \mathbb{D}_m \quad \text{as} \quad \mathbf{c}'_i \in \mathbb{C}_m \quad \text{for } \forall i \quad (m = 1, 2, \ldots, M) \tag{2.100}$$

After the clean and noisy codebooks are constituted, statistical parameters are estimated to construct the network of HMMs (The theory of HMMs is described in Chapter 4 in detail). Using the Viterbi algorithm, the HMM network converts a sequence of observed frames, $\mathbf{c}_i$, to a sequence of associated states, $s(i) \in \{1, 2, \ldots, M\}$. Therefore, the measure of the similarity between the clean observed frames and elements of the clean

codebook is firstly determined as following distortion measure [12, 47].

$$d(\mathcal{C}_m, \mathbf{c'}_i) = \|\mathbf{c'}_i\| \left(1 - \frac{\mathcal{C}_m^T \mathbf{c'}_i}{\|\mathcal{C}_m\| \|\mathbf{c'}_i\|}\right) \tag{2.101}$$

Applying this similarity measure, the state sequence of the observed clean training speech is estimated as

$$s'(i) = \underset{m=1,2,\ldots,M}{\arg\min}\ d(\mathcal{C}_m, \mathbf{c'}_i) \tag{2.102}$$

Next, the sequence of noisy observation codes, $\tilde{s}(i)$, corresponding to the observed noisy training speech is estimated as well.

$$\tilde{s'}(i) = \underset{m=1,2,\ldots,M}{\arg\min}\ d(\tilde{\mathcal{C}}_m, \tilde{\mathbf{c}}'_i) \tag{2.103}$$

These estimations of the state and observation codes enable the statistical calculation of the state transition probabilities required at HMM decoding.

$$a_{jk} = p\left(s'(i+1) = k | s'(i) = j\right) \tag{2.104}$$

$$b_{jk} = p\left(\tilde{s'}(i) = k | s'(i) = j\right) \tag{2.105}$$

2.4.1.2 Enhancement Process

The trained HMM network works as a function to decode the observed frames $\mathbf{c}_i$ into the most likely state sequence. This state sequence, $s(i)$, is derived by using Viterbi algorithm incorporating Equations (2.103), (2.104) and (2.105). (Viterbi algorithm is referred to Chapter 4.)

After the decoding $\mathbf{c}_i$ into $s(i)$, the waveform frames which are closest to the frames of the input noisy waveform are selected from the clean waveform cluster corresponding to the state sequence by exploiting the following similarity measure with power normalisation [9, 12].

$$\hat{\mathbf{x}}_i = \underset{\forall \mathbf{x'}_j \in \mathbb{D}_{s(i)}}{\arg\min} \frac{\mathbf{y}_i^T \mathbf{x'}_j}{\sqrt{|\,\|\mathbf{y}_i\|^2 - D^2|}\,\|\mathbf{x'}_j\|} \tag{2.106}$$

$$D^2 = \mathcal{E}\left[\mathbf{d'}_i^T \mathbf{d'}_i\right] \tag{2.107}$$

Finally, the selected clean waveform units are concatenated to reconstruct the enhanced

speech $\hat{x}(n)$.

2.4.1.3 Post-processing

The enhanced speech reconstructed with the preceding processes are concatenations of short-time waveform units and brings phase inconsistency at the frame boundaries. Consequently, post-processing is required to deal with this issue, for example, applying a Fourier analysis and synthesis model such as sinusoidal model to the selected frames and remove the phase discontinuity.

2.4.2 Model-Based Speech Enhancement

The model-based speech enhancement is another approach to the reconstruction-based speech enhancement. This method reconstructs speech with a speech production model such as a vocoder and the sinusoidal model, which are discussed in Chapter 3, instead of using a corpus or inventory of natural speech. A set of speech features of clean speech required by the speech production model to reconstruct speech is provided by statistical models of speech. Figure 2.7 illustrates a framework of model-based speech enhancement.

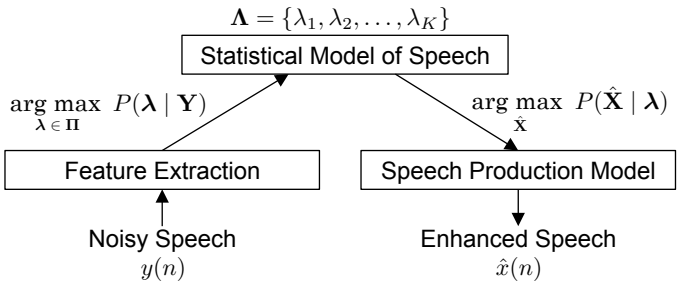

Figure 2.7: Framework of model-based speech enhancement.

At the training stage, a set of speech segments such as words and phonemes, $\mathbf{\Lambda} = \{\lambda_1, \lambda_2, \ldots, \lambda_K\}$, is statistically modelled by a set of parameters of speech features required by the speech production model. Then $\mathbf{\Lambda}$ is trained by clean speech.

At the test stage, the feature extraction process first extracts a set of the speech features from noisy speech, $y(n)$. Then, a sequence of the extracted features, $\mathbf{Y}$, is

decoded into a sequence of the statistical models, $\boldsymbol{\lambda}$, as

$$\boldsymbol{\lambda} = \arg\max_{\boldsymbol{\lambda}' \in \boldsymbol{\Pi}} P\left(\boldsymbol{\lambda}' \mid \mathbf{Y}\right) \tag{2.108}$$

where $\boldsymbol{\Pi}$ is a group which consists of all the possible sequence of the statistical models during the observation. In the next process, $\boldsymbol{\lambda}$ synthesises a sequence of statistical parameters of the speech features, $\hat{\mathbf{X}}$, as

$$\hat{\mathbf{X}} = \arg\max_{\hat{\mathbf{X}}'} P\left(\hat{\mathbf{X}}' \mid \boldsymbol{\lambda}\right) \tag{2.109}$$

$\hat{\mathbf{X}}$ is then passed to the speech production model and reconstructed to the time-domain speech, $\hat{x}(n)$.

The work proposed in this thesis falls into this category and uses STRAIGHT [48] as a speech production model while hidden Markov models (HMMs) are applied to the statistical model of speech segments. The largest advantage in the model-based speech enhancement over the corpus and inventory-based method is the cost required for a corpus or inventory because only statistical parameters rather than waveform data of natural speech are stored into the system [9]. Moreover, those statistical parameters can be adapted to different speakers or different types of noise without enlargement of the database. The model-based reconstruction, however, has challenging problems with its speech quality because it utilises only the statistical parameters to reconstruct the speech while the corpus and inventory-based method uses a range of natural speech. This causes artefacts to be produced in the resultant acoustic features. Moreover, quality of speech production model may also become a cause of degradation in model-based speech enhancement. Therefore, a wide range of speech speech production models are discussed in depth in the following chapter.

2.5 Conclusion of the Chapter

This chapter has presented conventional methods for the speech enhancement including the spectral subtraction, Wiener filters, statistical-model-based methods and subspace algorithms which are based on the filtering approaches. The topic was then extended

to the reconstruction-based approaches including corpus and inventory-based speech enhancement and model-based speech enhancement.

Experimental analysis has shown that the log MMSE method, which represents the statistical filtering methods, generally shows the best performance as the overall evaluation in terms of PESQ. The filtering-based methods including the log MMSE method, however, leave a lot of musical noise, residual noise and distortion at low SNRs such as 0 dB and below. This degradation is attributed to underestimation and overestimation at the noise estimation stage. Alternatively, the reconstruction-based methods are expected to obtain background noise-free speech because these methods reconstruct clean speech from an inventory which stores natural clean speech or statistical parameters of clean speech.

Chapter 3

Speech Production Models

Model-based speech enhancement utilises a speech production model to reconstruct clean speech. Therefore, the speech production model is one of the key processes to determine the baseline performance of model-based speech enhancement. This chapter first reviews physical speech production to give an insight into the characteristics of speech signals. Engineering models of speech production are then categorised into the source-filter models and the sinusoidal model and these are discussed in depth. Finally, different approaches to estimate the fundamental frequency, which gives harmonic information of speech to speech production models, are explored.

3.1 Introduction

Speech is airflow which is expelled from the lungs and then phonated through the vocal chords of the larynx and resonated in the vocal cavities before radiated through the oral articulators or the nose. It is known that the speech production process can be approximated as a digital filter comprising excitation signal inputs and vocal tract filters which models the spectral envelops of the vocal tract resonance, glottal flow and lip radiation based on the source-filter models. [49, 50]. Various types of vocoders have been developed that employ the source-filter model, for instance, linear predictive coding (LPC), residual-excited linear prediction (RELP), code-excited linear prediction (CELP), mixed excitation linear prediction (MELP) and STRAIGHT [48, 51–55].

Alternatively, a notion that speech signals can be synthesised as a sum of different

sinusoidal signals has derived another approach to the speech production models, namely, the sinusoidal model and HNM [56–58]. The process of speech production (i.e., reconstruction) is one of the key processes of model-based speech enhancement and a choice of the speech production model may determine the baseline performance of model-based speech enhancement. Therefore, it is important to discuss each of these models and to understand those properties.

The remainder of the chapter is organised as follows. The physical model of the human speech production process is first presented to understand the properties of the speech signals and to acquire insights into speech production models. The subsequent two sections explore different speech production models represented as source-filter models and the sinusoidal model. Finally, different methods to estimate the fundamental frequency (f_0) and voicing, which are referred to for the harmonic features of the speech by speech production models, are discussed prior to the conclusion of the chapter.

3.2 Physical Speech Production Process and Speech Signals

In the human speech production process, an excitation signal is created from air expelled by the lungs at the vocal chords. The signal then excites resonant cavities of the vocal tract which consist of the pharyngeal cavity, the oral cavity and nasal cavity. The resonant frequencies formed at this process are known as the formant frequencies. The signal is then radiated through the lips involving the tongue and teeth. When the velum is lowered, the nasal cavity is acoustically coupled to the pharyngeal and oral cavities to produce the nasal sounds of speech [1, 49]. Figure 3.1 illustrates the overview of this process.

Changes in the shape of the vocal tract causes the resonant frequencies of the vocal tract to change which in turn produces different speech sounds, therefore, the speech signals are nonstationary. However, it is known that the short-time segments of voiced speech signals (10-30 ms) can be fairly stationary because the muscles constituting the vocal tract do not change their shape as quickly as the short time duration for voicing. However, some changes in the vocal tract to produce unvoiced sounds, such as release

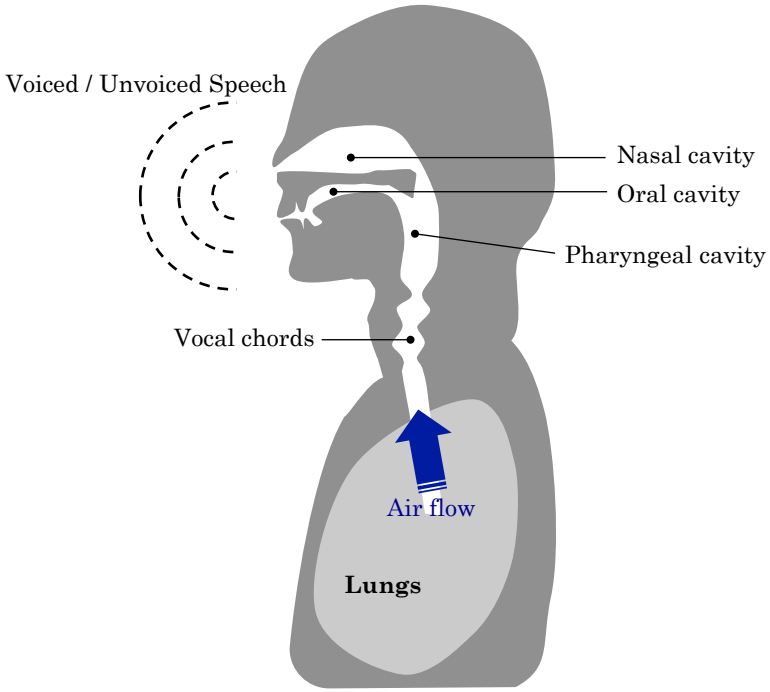

Figure 3.1: Overview of the human speech production model.

of a constriction which makes a plosive, are fast with regard to the analysis frame, and the excitation source for unvoiced sounds is turbulence which will only be filtered by the vocal tract after constriction. Therefore, formants of unvoiced sounds look different during frication.

Figure 3.2 illustrates the relation between the shape of the oral and pharyngeal cavities and the frequency response of those resonances. For voiced sounds, larger cavities give rise to lower formant frequency, and the tongue divides the vocal tract into two cavities. The rear cavity determines the first formant frequency, F_1, whereas the front cavity determines the second formant frequency, F_2.

Figure 3.3(a) shows an instance of the speech signal for the utterance "bin blue at L four again" of a male speaker sampled at 8 kHz. This shows that the speech signals can be divided into various segments corresponding to the different sounds and these segments can be largely categorised into voiced segments having periodic signal property or unvoiced segments having a property of the random noise. Figure 3.3(b) shows the zoomed-in plot of the voiced segment corresponding to the sound /ue/ in "blue" while Figure 3.3(c) depicts the zoomed-in plot of the unvoiced segment corresponding to /f/

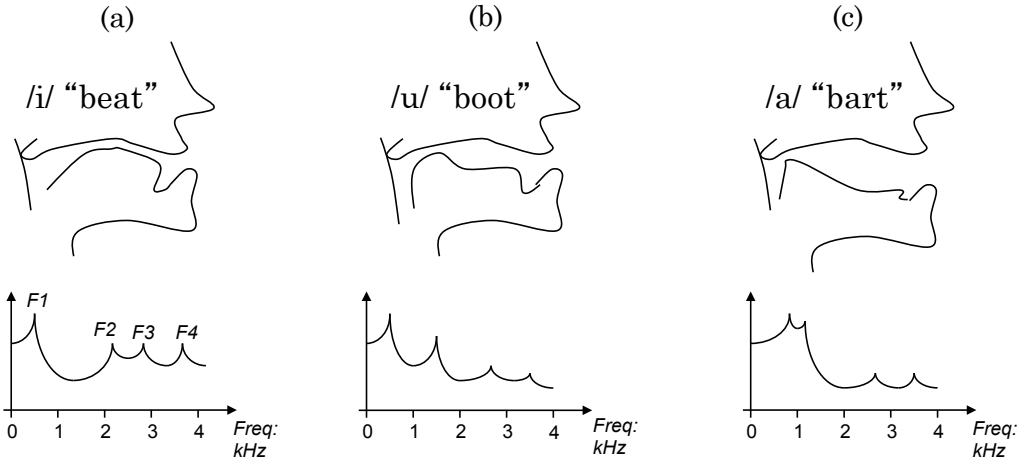

Figure 3.2: The relation between the shape of the oral and pharyngeal cavity and the frequency response of the resonance showing a) sound of /i/ in “beat”, b) sound /u/ in “boot” and c) sound /a/ in “bart”.

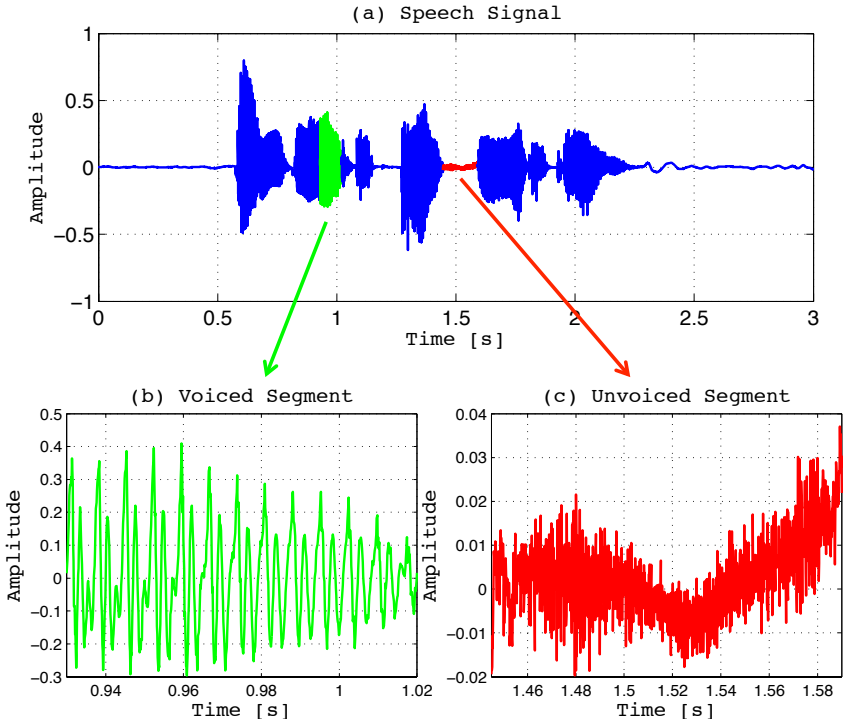

Figure 3.3: Time domain waveform of the utterance “bin blue at L four again” of a male speaker showing: a) the whole speech signal, b) the zoomed-in plot corresponding to the voiced segment “ue” in “blue”, c) the zoomed-in plot corresponding to the unvoiced segment “f” in “four”.

in “four”, and this derives the notion that the voiced signals can be modelled with the excitation source of the periodic pulse train whereas the unvoiced signals are modelled as the random white noise.

3.3 Source-Filter Models

The source-filter model is an engineering model of the speech production process exploiting the properties of the speech signals discussed above. This section first presents the underlying notion of the source-filter model and then specifically the LPC and STRAIGHT models are discussed as the representative examples of the source-filter model.

3.3.1 Overview

The source-filter model assumes that an excitation signal is generated by the lungs and vocal chords and this is then filtered by the vocal tract. The excitation signal of the voiced sound is a periodic signal and thus, the voiced speech has harmonic properties according to the period of the excitation signal, so-called pitch period, while the excitation signal of the unvoiced speech is a noise-like signal and the unvoiced speech signals have the characteristics of random noise. According to these properties of the source-filter model, the engineering model of the source-filter model is represented as Figure 3.4. For

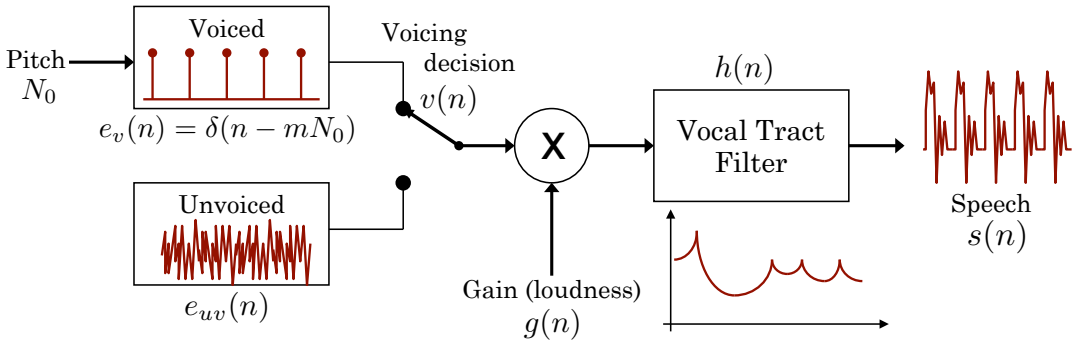

Figure 3.4: Overview of the source-filter model.

the discrete-time domain speech signal, $s(n)$, the model is parameterised by the pitch period, $N_0(n)$, the voiced/unvoiced decision, $v(n)$, the loudness gain, $g(n)$, the random noise for the unvoiced excitation, $e_{uv}(n)$, and the impulse response of the vocal tract filter, $h(n)$. The vocal tract filter is generally modelled as an all-pole filter so that the

poles of the transfer function form the formants in its frequency response and thus, the transfer function of the vocal tract filter is derived as

$$H(z) = \frac{1}{1 - \sum_{k=1}^{P} a_k z^{-k}} \tag{3.1}$$

where a_k are the filter coefficients to determine the frequency characteristics of the vocal tract filter and P is the order of the filter which determines the number of the formants. The number of the formants is specified P and frequency bandwidth.

3.3.2 Linear Predictive Coding

LPC represents the source-filter model with a simple structure and it models the speech signal as Equation (3.2) in z-domain.

$$S(z) = GH(z)E(z) \tag{3.2}$$

where $S(z)$, $H(z)$, $E(z)$ and G represent the speech signal, the transfer function of the vocal tract filter, the excitation signal in z-domain and a loudness gain factor respectively. $H(z)$ is determined by Equation (3.1) and therefore, the model derives

$$S(z) = \frac{G}{1 - \sum_{k=1}^{P} a_k z^{-k}} E(z) \tag{3.3}$$

This filter has a structure of the autoregressive (AR) filter as illustrated in Figure 3.5 and the excitation signal is derived as its inverse, namely a finite impulse response (FIR) filter

$$\begin{aligned} Ge(n) &= s(n) - (a_1 s(n-1) + a_2 s(n-2) + \cdots + a_p s(n-P)) && (3.4) \\ &= s(n) - \sum_{k=1}^{P} a_k s(n-k) && (3.5) \end{aligned}$$

where $e(n)$ and $s(n)$ are the time domain signals of the excitation and the speech signal respectively. Equation (3.5) signifies that the coefficients of the vocal tract filters, a_k $(k = 1, 2, \cdots, P)$ constitute a linear prediction filter which estimates $s(n)$ with the past samples, $s(n-1)$, $s(n-2)$, $\cdots$, $s(n-P)$, and the residual of the linear prediction corre-

sponds to $Ge(n)$ as illustrated in Figure 3.6. Thus, the coefficients, a_k, are determined

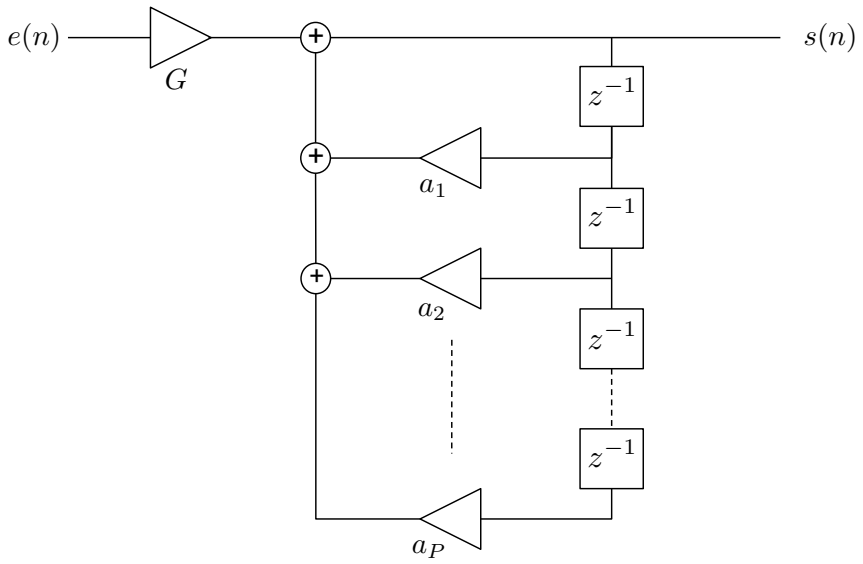

Figure 3.5: AR model forming a vocal tract filter of LPC vocoder.

by minimum mean square error (MMSE) such that the residual is minimised in terms of MMSE in order to approximate $e(n)$ as a pulse train as follows.

$$J = \mathcal{E}\left[(Ge(n))^2\right]] = \mathcal{E}\left[\left(s(n) - \sum_{k=1}^{P} a_k(n-k)\right)^2\right] \tag{3.6}$$

The derivative of J with respect to a_l ($l = 1, 2, \cdots, P$) is set equal to zero in order to minimise the mean square error.

$$\frac{\partial J}{\partial a_l} = -2\mathcal{E}\left[\left(s(n) - \sum_{k=1}^{P} a_k s(n-k)\right) s(n-l)\right] \tag{3.7}$$

$$= -2\left(\mathcal{E}\left[s(n)s(n-l)\right] - \sum_{k=1}^{P} a_k \mathcal{E}\left[s(n-k)s(n-l)\right]\right) \tag{3.8}$$

$$= -2\left(r_{ss}(l) - \sum_{k=1}^{P} a_k r_{ss}(|k-l|)\right) = 0 \tag{3.9}$$

where

$$r_{ss}(l) = \mathcal{E}\left[s(n)s(n-l)\right] \quad (\text{for } l = 1, 2, \cdots, P) \tag{3.10}$$

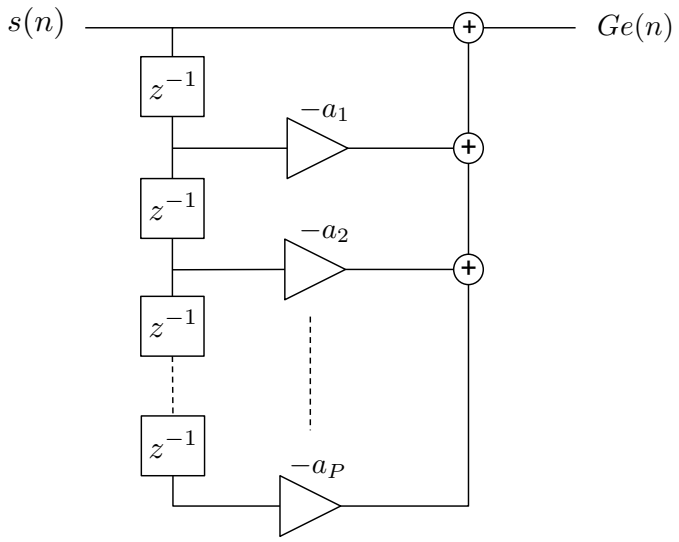

Figure 3.6: Linear prediction filter which is the inverse of the LPC model

The preceding equations derives the vocal tract filter coefficients as

$$\mathbf{r}_{ss} = \mathbf{R}_{ss}\mathbf{a} \tag{3.11}$$

$$\mathbf{a} = \mathbf{R}_{ss}^{-1}\mathbf{r}_{ss} \tag{3.12}$$

where $\mathbf{a}$ is the vector of the filter coefficients, $[a_1\ a_2\ \cdots\ a_P]^T$, $\mathbf{r}_{ss}$ is the autocorrelation vector, $[r_{ss}(0)\ r_{ss}(1)\ \cdots\ r_{ss}(P-1)]^T$, and $\mathbf{R}_{ss}$ is the autocorrelation matrix formed as

$$\mathbf{R}_{ss} = \begin{bmatrix} r_{ss}(0) & r_{ss}(1) & r_{ss}(2) & \cdots & r_{ss}(P-1) \\ r_{ss}(1) & r_{ss}(0) & r_{ss}(1) & \cdots & r_{ss}(P-2) \\ r_{ss}(2) & r_{ss}(1) & r_{ss}(0) & \cdots & r_{ss}(P-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{ss}(P-1) & r_{ss}(P-2) & r_{ss}(P-3) & \cdots & r_{ss}(0) \end{bmatrix} \tag{3.13}$$

Figure 3.7 shows an instance of the speech reconstruction with a 16th order LPC model. Subplot (a) depicts the natural speech of the sound /ue/ in "blue" of a male speaker sampled at 8 kHz. Subplot (b) and (c) show the residual of the linear prediction as the reference of the excitation signal, $e(n)$, and the frequency response of the vocal tract filter, $H(f)$, obtained by Equations (3.5) and (3.12). Subplot (d) represents the pulse train the period of which is equal to the pitch period, T_0, of subplot (b), and subplot

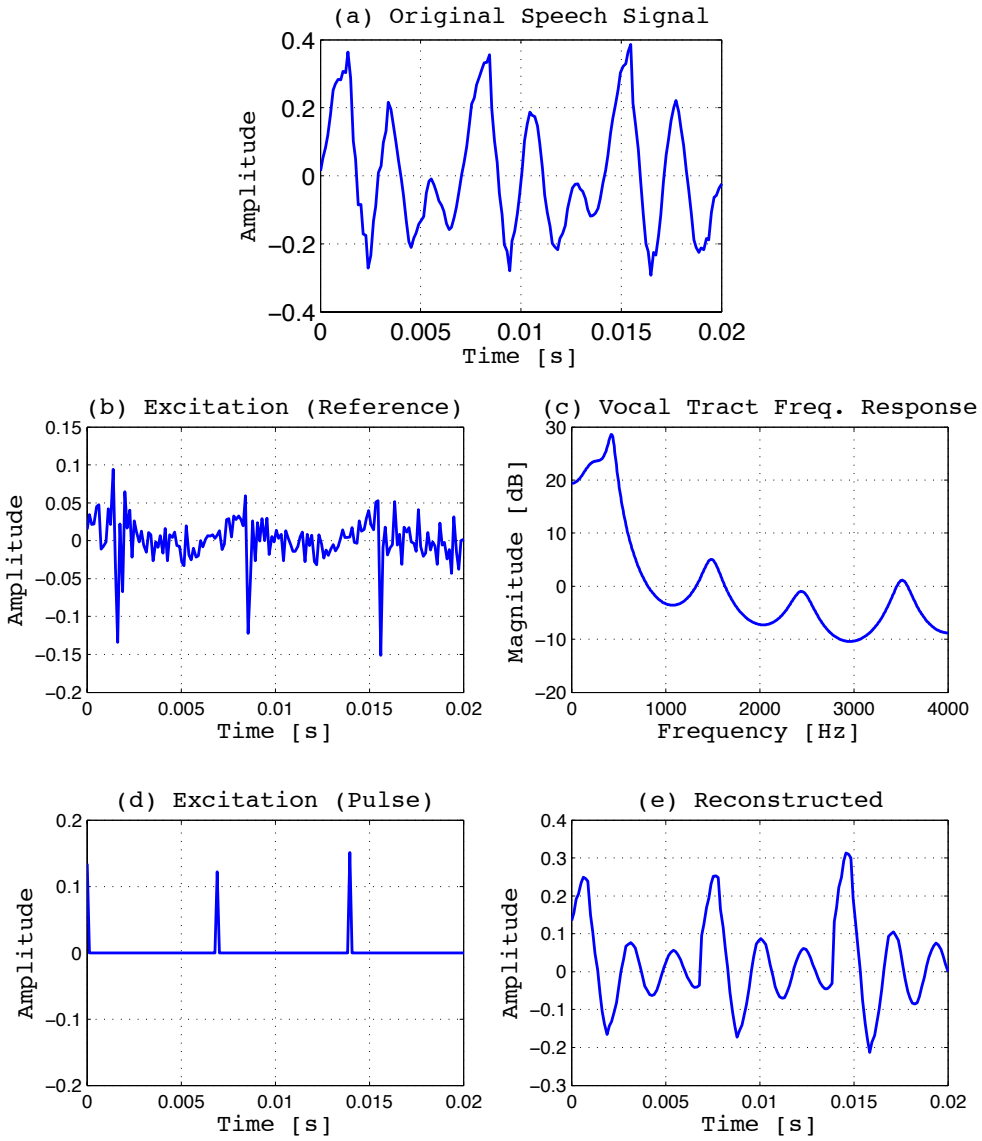

Figure 3.7: Example of the speech production with the LPC model (P = 16) showing: a) original natural speech of the sound /ue/ in "blue" uttered by a male speaker, b) the residual of the linear prediction as the reference of the excitation source, c) the frequency response of the vocal tract filter, d) pulse train used for the excitation and e) reconstructed speech with c) & d).

(e) shows the reconstructed speech signal from the vocal tract(c) and the excitation (d). These plots show that although the reconstructed signal is over-smoothed due to the excitation source which consists of only a simple pulse train, the characteristics of the speech is kept because it has the same formant frequencies and the pitch period as the original natural speech. Therefore, this notion to use a simple pulse train as the excitation of voiced speech is used as a basic technique of vocoders and also applied to low bit rate coding. Specifically, typical 10th order vocal tract filters use 40 bits/frame while the excitation source information requires typically 4 bits/frame. Therefore, if the speech is windowed into 20 ms frames, the required bit rate equals 2,200 bit/s. Subjective listening tests, however, discover explicit degradation with having a buzzy characteristic which is attributed to strong harmonics brought by using simple pulse excitation because the voiced speech in the natural speech signals are not exactly periodic [59, 60].

Different methods have been proposed to tackle the problem of this buzzy noise in the reconstructed speech and the mixed excitation models which utilise a mixture of the pulse and random noise rather than using binary pulse as the excitation source have been successfully adopted to reduce the buzzy sounds. These methods such as RELP, CELP, MELP and those variant algorithms [53–55, 59, 61, 62] require the mixture ratio of the periodic signal and aperiodic noise in the excitation source instead of the binary voicing information. The mixture ratio is estimated from, for example, the feature of the linear prediction residual [62], and the source signal is constituted with the mixture ratio, random white noise and the estimated fundamental frequency, f_0, which is the reciprocal of the pitch period, T_0.

$$f_0 = \frac{1}{T_0} \tag{3.14}$$

The methods of the fundamental frequency estimation are discussed in Section 3.5.

The LPC models determine the filter coefficients by using linear prediction analysis as discussed above. However, different approaches to estimate the filter coefficients for the source-filter model also have been proposed such as the methods using short-time Fourier analysis and Mel-cepstral analysis [63–65].

3.3.3 STRAIGHT

STRAIGHT is a sophisticated mixed excitation source-filter vocoder which has been successfully applied to HMM-based speech synthesis [9, 50, 66]. The filtering process of STRAIGHT is decomposed into minimum-phase and all-pass systems so that the group delay of the system can be adjusted to improve speech quality because it is known that the group delay in the speech signal is perceptually detectable [67].

STRAIGHT requires three inputs for speech reconstruction, namely i) the spectral surface, $S(f, i)$, of the speech, ii) the fundamental frequency contour, $f_0(i)$, and iii) the aperiodicity measure, $A(f, i)$ where f and i denotes the indices of the frequency bins and time frames respectively [68–70]. $S(f, i)$ is a time series of the spectral envelopes, $S_i(f)$, in which the harmonic information and temporal interference are eliminated [71]. $S_i(f)$ forms the vocal tract filter consisting of the minimum phase part, $H_i(f)$ and the all-pass part, $\Phi_i(f)$. When a filter has the minimum phase impulse response, the complex cepstrum of the filter coefficients is causal [72], and there is a relationship between complex cepstrum, $c_i(n)$, and real cepstrum, $\hat{c}_i(n)$, as

$$\hat{c}_i(n) = \frac{c_i(n) + c_i(-n)}{2} \tag{3.15}$$

therefore, $H_i(f)$ is derived from $S_i(f)$ as

$$H_i(f) = \exp\left(\mathscr{F}\left[c_i(n)\right]\right) \tag{3.16}$$

$$c_i(n) = \begin{cases} 2\hat{c}_i(n) & (n > 0) \\ \hat{c}_i(n) & (n = 0) \\ 0 & (n < 0) \end{cases} \tag{3.17}$$

$$\hat{c}_i(n) = \mathscr{F}^{-1}\left[\log\left(S_i(f)\right)\right] \tag{3.18}$$

where the notation $\mathscr{F}[\cdot]$ and $\mathscr{F}^{-1}[\cdot]$ denote Fourier and inverse Fourier transform respectively. The all-pass filter $\Phi_i(f)$ adjusts the group delay of the system and thus, the energy of the periodic pulse in the excitation source is spread to the adjacent time samples. This is also effective in reducing the buzzy noise from the reconstructed speech.

The impulse response of the vocal tract filter, $h_i(n)$, is then acquired as

$$h_i(n) = \mathscr{F}^{-1}\left[H_i(f)\Phi_i(f)\right] \tag{3.19}$$

The excitation source is synthesised from $f_0(i)$ and $A(f,i)$. The fundamental frequency at the i-th frame, f_{0i}, represents the periodic component in the excitation source while the aperiodicity at the i-th frame, $A_i(f)$, represents the indeterministic component and it is defined as the proportion of the lower spectral envelope to the upper spectral envelope to represent the relative energy distribution of the random noise components [60]. Therefore, the excitation source of the i-th frame, $e_i(n)$ is synthesised as

$$e_i(n) = \frac{1}{\sqrt{f_{0i}}}\delta\left(n - \frac{f_s}{f_{0i}}\right) + \mathscr{F}^{-1}\left[A_i(f)|N(f)|\right] \tag{3.20}$$

$$\delta(n) = \begin{cases} 1 & (n = 0) \\ 0 & (\text{otherwise}) \end{cases} \tag{3.21}$$

where $|N(f)|$ and f_s represents the magnitude spectrum of the random white noise and the sampling frequency respectively. The speech signal is finally reconstructed by the source-filter convolution.

$$\hat{s}_i(n) = h_i(n) * e_i(n) \tag{3.22}$$

where the symbol '$*$' denotes the operation of convolution.

Figure 3.8 shows an example of the speech reconstruction with STRAIGHT. The reconstruction process is performed with the following all-pass filtering settings.

i) Group delay $d(f) = 0$

ii) Group delay $d(f) = \begin{cases} 0 & (f \leqslant 2000 \text{ Hz}) \\ 0.5 \text{ ms} & (f > 2000 \text{ Hz}) \end{cases}$

iii) Group delay $d(f) = \begin{cases} 0 & (f \leqslant 2000 \text{ Hz}) \\ 2.0 \text{ ms} & (f > 2000 \text{ Hz}) \end{cases}$

Figure 3.8 (a) shows a segment of the natural speech of the sound /ue/ in "blue" uttered by a male speaker and the signal is sampled at 8 kHz but this signal is not identical to

Figure 3.7 (a). (b) depicts the magnitude spectrum of the vocal tract filter. (c) and (d) show the excitation and reconstructed speech respectively with the group delay setting i). (e) and (f) also represent the excitation and reconstructed speech respectively with the group delay setting ii) while (g) and (h) show the plots with the group delay setting iii). The blue lines in the excitation source plots represent the sum of the periodic and noise components in the signals whereas the red lines extract only the periodic pulse component in the signals. These plots illustrate the influence of the group delay settings and also show that the reconstructed speech has a better approximation than the over-smoothed signal reconstructed with the simple LPC model shown in Filgure 3.7.

3.4 Sinusoidal Model

The source-filter models are based on the notion that the excitation source represented by a mixture of a pulse train and white noise is resonated by the vocal tract filter. The sinusoidal model is an alternative to the preceding approach and it models speech as a summation of the sinusoids that have the harmonic frequencies of the speech [56, 57].

3.4.1 Basic Sinusoidal Model

The sinusoidal model leads a stationary short-time speech frame, $s_i(n)$, that is modelled as

$$\begin{aligned} \hat{s}_i(n) &= A_1 \cos\left(2\pi\frac{f_1}{f_s}n + \phi_1\right) + A_2 cos\left(2\pi\frac{f_2}{f_s}n + \phi_2\right) + \cdots \\ &\quad \cdots + A_L cos\left(2\pi\frac{f_L}{f_s}n + \phi_L\right) \qquad (3.23) \\ &= \sum_{l=1}^{L} A_l \cos\left(2\pi\frac{f_l}{f_s}n + \phi_l\right) \qquad (3.24) \end{aligned}$$

where L denotes the number of harmonics in the speech, f_s is the sampling frequency and A_l, f_l and ϕ_l are amplitude, frequency and phase of each sinusoid respectively. Regarding the harmonic frequencies are pure integral multiples of the fundamental frequency and the sinusoid amplitudes A_l correspond to the magnitude spectra at the harmonic frequencies while the sinusoid phases ϕ_l are represented by the phase spectra at the harmonic

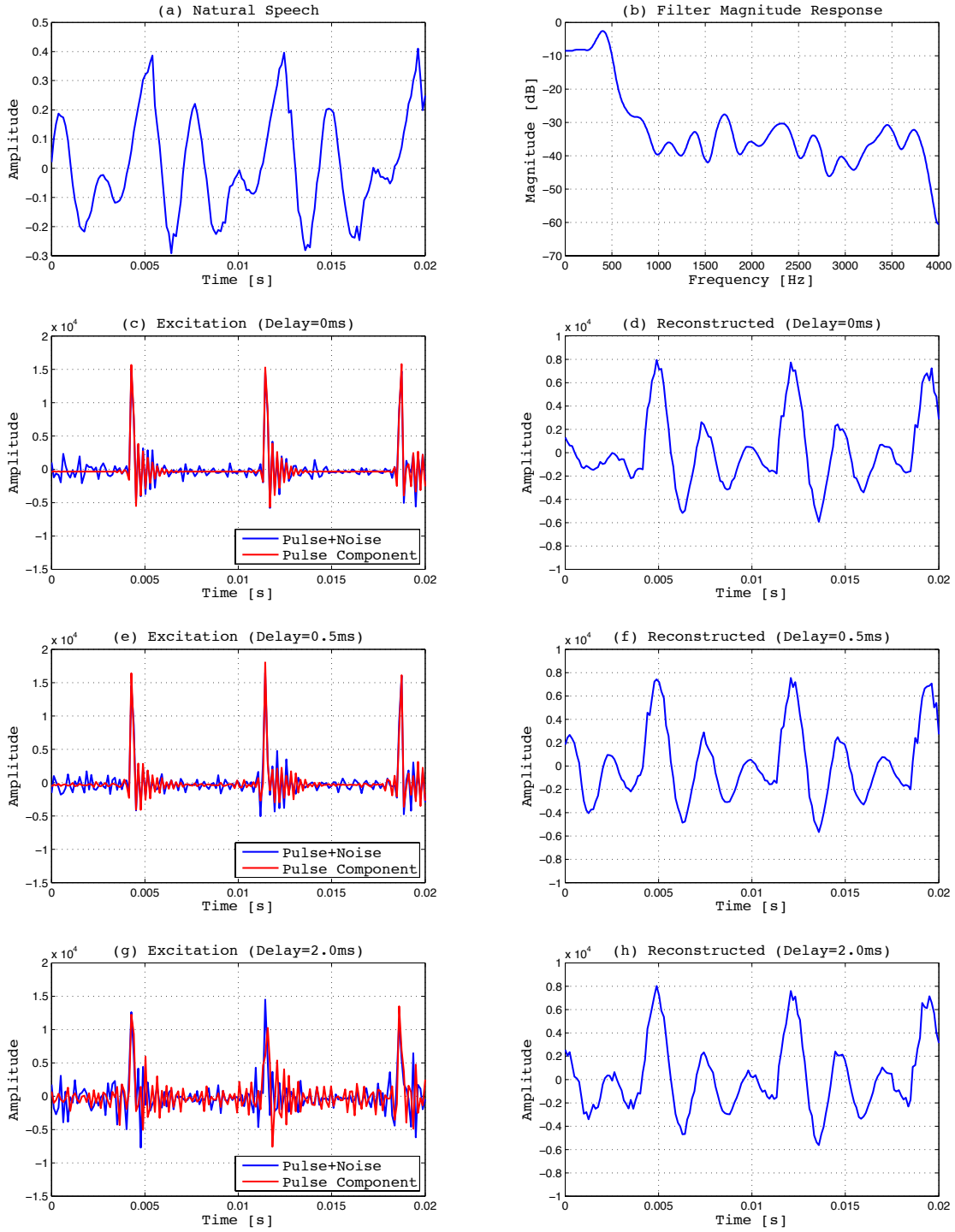

Figure 3.8: Example of the speech reconstruction with STRAIGHT showing: a) a segment of the natural speech, b) the magnitude spectrum of the vocal tract filter, c), e) and g) the excitation source where the blue line represents the sum of the periodic and noise components while the red line shows only the periodic pulse component at the group delay of 0, 0.5, and 2.0 ms respectively, d), f) and h) reconstructed speech at the group delay of 0, 0.5, and 2.0 ms respectively.

frequencies, the preceding equation is simplified as

$$\hat{s}_i(n) = \sum_{l=1}^{L} A_l \cos\left(2\pi l \frac{f_{0i}}{f_s} n + \phi_l\right) \tag{3.25}$$

where

$$A_l = |S_i(l f_{0i})| \tag{3.26}$$

$$\phi_l = \angle S_i(l f_{0i}) \tag{3.27}$$

where f_{0i} and $S_i(f)$ are the fundamental frequency at the i-th frame and the complex spectrum of $s_i(n)$ respectively.

Equation (3.25) models voiced speech signals. Frames of unvoiced speech, however, cannot be represented as the summation of the harmonic components as there is no pitch period in the signal. Therefore, $s_i(n)$ is modelled as the following binary state model by taking the notion of the source-filter model for the unvoiced speech [7].

$$\hat{s}_i(n) = \begin{cases} \sum_{l=1}^{L} A_l \cos\left(2\pi l \frac{f_{0i}}{f_s} n + \phi_l\right) & \text{(voiced)} \\ h_i(n) * w_i(n) & \text{(unvoiced)} \end{cases} \tag{3.28}$$

where $h_i(n)$ represents the impulse response of the filter, coefficients of which are determined from the spectral envelope of $s_i(n)$, and $w_i(n)$ is a sequence of white noise.

Figure 3.9 shows an example of the speech reconstruction with the sinusoidal model. Subplot (a) shows a short-time segment of the natural speech of the sound /ue/ in "blue" uttered by a male speaker and subplot (b) is the reconstructed speech. The spectral envelopes to estimate the amplitude and phase of the sinusoids are up-sampled to have the resolution of 1 Hz at the reconstruction process. The plots show that the sinusoidal model produces good quality speech and an evaluation across a range of speech production models has reported that variants of the sinusoidal models generally obtain better performance than variants of the source-filter models [66] but require more parameters for the input making it less suitable for coding applications.

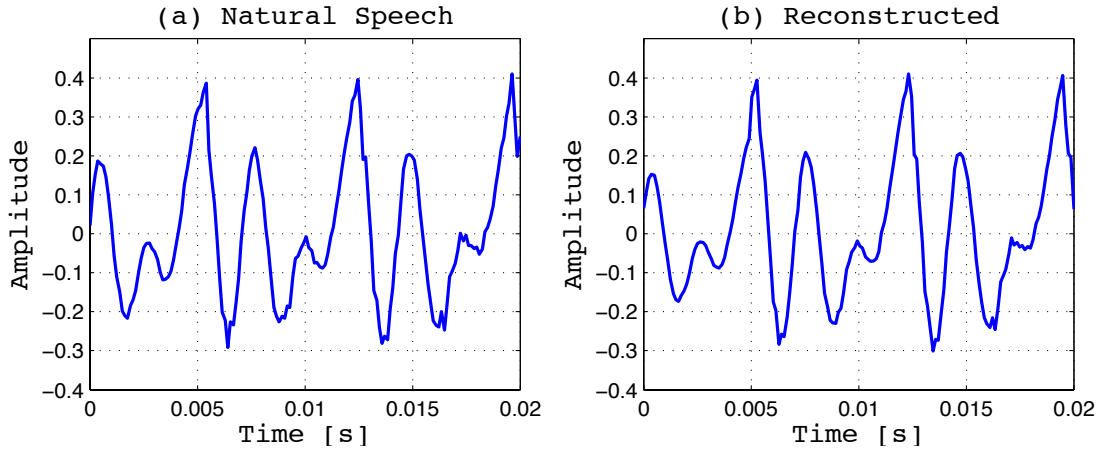

Figure 3.9: Example of the speech reconstruction with the sinusoidal model showing: a) a short-time segment of the natural speech of the sound "ue" in "blue" uttered by a male speaker and b) the reconstructed speech.

3.4.2 Harmonics Plus Noise Model

The harmonics plus noise model (HNM) is a variant of the sinusoidal model which divides voiced speech into a harmonic component and a stochastic component. The harmonic component is modelled as a summation of the harmonic sinusoids while the stochastic component is modelled as white noise [58, 73]. The reconstructed speech is derived by the summation of the harmonic and stochastic components as follows.

$$\hat{s}_i(n) = \sum_{l=1}^{L} A_l \cos\left(2\pi l \frac{f_{0i}}{f_s} n + \phi_l\right) + r_i(n) w_i(n) \tag{3.29}$$

where $r_i(n)$ is a time domain window which modulates the white noise in order to match the energy to the harmonic component in the original speech [58].

Several variants of the sinusoidal model and HNM have been proposed such as the adaptive harmonic model (aHM), and perceptual dynamic sinusoidal model (PDM) [66, 74, 75].

3.5 Estimation of the Fundamental Frequency

Both the source-filter and sinusoidal speech production models require the fundamental frequency of the speech in order to refer it for the harmonic information of speech. This

section discusses several methods to estimate the contour of the fundamental frequency of speech. The performance of each method is then evaluated in Section 3.5.3.

3.5.1 Time-Domain Analysis

Various methods to estimate the fundamental frequency of speech signals have been developed and they are largely categorised as methods to analyse a signal in the time-domain or methods to analyse a signal in the frequency or cepstrum-domain. The following sections discuss representative methods of time-domain approach.

3.5.1.1 Autocorrelation Method

Considering time-shift operation applied to a periodic signal, the signal matches the original when the amount of the time-shift is equal to the pitch period of the signal. In other words, the auto-correlation function of a periodical signal is maximised when the time-shift is equal to the pitch period of the signal. Therefore, the pitch period of the signal, T_{0i}, and the fundamental frequency, f_{0i}, at frame index, i, can be derived as

$$T_{0i} = \frac{\arg\max_m \{r_{ss}(m)\}}{f_s} \tag{3.30}$$

$$f_{0i} = \frac{1}{T_{0i}} \tag{3.31}$$

where m denotes the number of sample shift and $r_{ss}(m)$ is the autocorrelation function of a periodical discrete-time signal $s(n)$ sampled at f_s, therefore, $r_{ss}(m)$ is determined as

$$r_{ss}(m) = \sum_{n=0}^{M} s_i(n)s_i(n+m) \quad \text{for } M = L - m \tag{3.32}$$

where L is the number of the samples in a frame.

Periodic signals of speech often show high auto-correlation as the time-shift is equal to a half of the pitch period or integer multiples of the pitch period [76]. Therefore, the estimate of the pitch period could be detected as a half or integral multiple of the actual pitch period. This is called an octave error which causes necessity of using post processing that applies temporal continuity constraints. The estimate of voicing information is also obtained during that post processing.

Although it is known that the auto-correlation method has good performance and is robust to noise [77], this method has a major disadvantage in sensitivity to changes in signal amplitude. Therefore, if the signal has a rapid amplitude change in the frame, it can introduce incorrect lags that have greater autocorrelation values than the lag corresponding to the true pitch period and consequently estimation errors.

3.5.1.2 Normalised Autocorrelation

The normalised autocorrelation function has been applied to fundamental frequency estimation to tackle the problem on the auto-correlation method [78]. In this method, the autocorrelations between different lags are determined as

$$r_{ss}(m) = \frac{\sum_{n=0}^{M} s_i(n)s_i(n+m)}{\sqrt{\sum_{n=0}^{M} s_i(n) \sum_{n=0}^{M} s_i(n+m)}} \quad \text{for } M = L - m \tag{3.33}$$

In this case, autocorrelation is not affected by the changes in the signal amplitude because of the normalisation by the signal energy. A robust algorithm for pitch tracking (RAPT) which has been successfully applied to many applications uses this algorithm [77].

3.5.1.3 YIN Method

The YIN method uses a different approach to deal with the problem of amplitude changes in speech signals [79]. This method employs the squared difference function rather than the normalised autocorrelation function.

$$d(m) = \sum_{n=0}^{M} (s(n) - s(n+m))^2 \quad \text{for } M = L - m \tag{3.34}$$

$$= \sum_{n=0}^{M} s(n)s(n) + \sum_{n=0}^{M} s(n+m)s(n+m) - 2\sum_{n=0}^{M} s(n)s(n+m) \tag{3.35}$$

$$= r_{ss}(0) + r_{ss}^{(m)}(0) - 2r_{ss}(m) \tag{3.36}$$

$d(m)$ is minimised at m corresponding to the pitch period by the term of $-2r_{ss}(m)$ while the term of $r_{ss}^{(m)}(0)$ represents the energy of the lagged signal and compensates for changes in the signal amplitude. Furthermore, $d(m)$ is replaced with the following

function to keep the values high at low lag periods.

$$d'(m) = \begin{cases} 1 & (m = 0) \\ d(m) \ / \ [(1/m) \sum_{k=1}^{m} d(k)] & (\text{otherwise}) \end{cases} \tag{3.37}$$

This avoids producing dips at lags corresponding to the first formant and improves the accuracy of the pitch detection.

3.5.2 Cepstrum and Frequency-Domain Analysis

The preceding methods to estimate the fundamental frequency employ time-domain approaches. Alternatively, the following sections discuss cepstrum and time-domain approaches.

3.5.2.1 Cepstrum Method

Several methods to utilise the cepstrum of the speech have been proposed [80, 81]. The cepstrum of $s(n)$ is determined as

$$c_i(n) = \left| \mathscr{F}^{-1} \left[\log \left(\left| \mathscr{F} \left[s_i(n) \right] \right|^2 \right) \right] \right| \tag{3.38}$$

The log operation flattens the harmonic peaks in the spectral magnitude and thus, more distinct periodic peaks are given in the cepstrum. Consequently, the fundamental frequency can be estimated by detecting the peaks in the cepstrum.

Although the cepstrum method has good performance and is robust to noise [77], this method also has a problem of sensitivity to changes in signal amplitude as well as the auto-correlation method in the time-domain.

3.5.2.2 PEFAC

The correlation methods including the square difference function in the YIN method perform well in moderate noise levels. However, these methods do not give a distinct peak in the autocorrelation function in more severe noise conditions such as negative SNRs. The pitch estimation filter with amplitude compression (PEFAC) method is proposed as a frequency-domain approach to pitch estimation which is robust to high levels of noise

to resolve this problem [82].

PEFAC estimates the fundamental frequency from the autocorrelation in the log-frequency domain with a matched filter applying a novel spectral normalisation [83]. For a periodic source contaminated with stationary noise, the power spectral density at frame i in the log-frequency domain is determined as

$$S_i(q) = \sum_{k=1}^{K} a_k \delta \left(q - \log k - \log f_{0i}\right) + D_i(q) \tag{3.39}$$

where q denotes log-frequency, and K, a_k and f_{0i} are the number of harmonics, power at k-th harmonic and the fundamental frequency respectively, and $D_i(q)$ represents the power spectral density of the noise. In the log-spectral domain, the harmonic interval is determined by $\log k$ rather than f_{0i}, therefore, a matched filter is derived as

$$h_i(q) = \sum_{k=1}^{K} \delta \left(\log k - q\right) \tag{3.40}$$

and the filter output $S_i(q) * h_i(q)$ gives a peak at $q_0 = \log f_{0i}$.

$D_i(q)$, however, broadens the spectral peaks in $S_i(q)$ and the filter output is affected in severe noise conditions. Therefore, PEFAC applies a spectral normalisation to reduce the dominance of the noise component by using the smoothed periodogram and the universal long term average speech spectrum (LTASS) which is independent of language and speaker [84]. The normalised periodogram, $S'_i(q)$, is determined as

$$S'_i(q) = S_i(q) \frac{L(q)}{\bar{S}_i(q)} \tag{3.41}$$

where $L(q)$ represents the universal LTASS and $\bar{S}_i(q)$ is the smoothed periodgram filtered by moving average filters in the time and log-frequency domain. [82] reports that this normalisation can give heavy attenuation to any regions of the periodogram at which the SNR $\ll$ 0 dB. The autocorrelation function is finally obtained as

$$r_{ss}(q) = S'_i(q) * h_i(q) \tag{3.42}$$

and $\log f_{0i}$ is represented by a peak in $r_{ss}(q)$.

3.5.3 Experimental Results and Evaluation

This section examines the performance of methods to estimate the fundamental frequency discussed above. In applications using statistical parametric speech synthesis (SPSS) such as text-to speech, both spectral features and the fundamental frequency are synthesised by the statistical models of HMMs [50]. Model-based speech enhancement, however, can estimate the fundamental frequency of clean speech from the noisy speech rather than synthesise it from the statistical models, and this motivates each method of fundamental frequency estimation to be examined in noisy conditions. Therefore, RAPT, YIN and PEFAC are examined in noisy conditions with white noise and babble noise at SNRs from -5 dB to 10 dB. The GRID database (two male speakers and two female speakers) down sampled to 8 kHz is used for the test speech and each method estimates the fundamental frequency from 200 utterances for each speaker (800 utterances in total) in each noise condition. The ground truth is determined as follows. All three methods first estimate the voicing and fundamental frequency of the test utterances in noise-free condition and then the voicing decision of each frame is determined by majority vote. The fundamental frequency of the voiced frames are decided by taking the mean of the estimates of the algorithms constituting the voicing decision.

The performance of each method is examined in terms of gross error rate, p_g, and accuracy, p_a, which are marked by the following criteria.

$$p_g = \frac{n_{>20}}{N} \times 100 \tag{3.43}$$

$$p_a = \frac{n_{<}5}{N_v} \times 100 \tag{3.44}$$

where $n_{>20}$ represents the number of frames including voicing error frames the estimates of which are more than 20 % apart from the ground truth while $n_{<5}$ is the number of the frames whose distance to the ground truth is less than 5 %. N and N_v denote the number of the total frames and voiced frames respectively.

Each method takes the analysis window length as 90 ms duration with 5 ms window shift and the minimum and maximum of the fundamental frequency range is set to 50 Hz and 300 Hz respectively. RAPT and PEFAC employ dynamic programming (DP) [85] for post processing whereas YIN adopts an aperiodicity measure to obtain the voicing

decision. Other settings for each algorithm follow the empirical parameters in [77,79,82].

Figure 3.10 shows the test results. Subplots (a) and (b) illustrate the gross error rate of each method at the different SNRs with white and babble noise. Subplots (c) and (d) show the accuracy of each method in the same conditions. The test results show

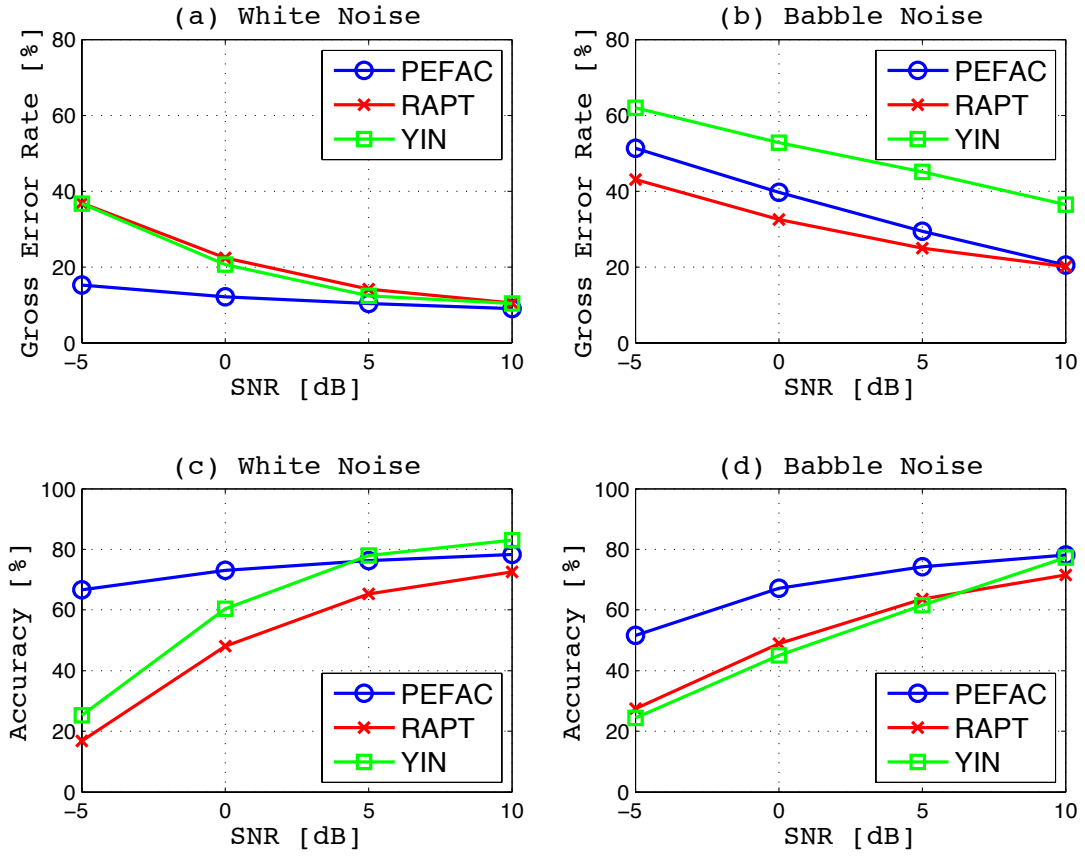

Figure 3.10: Fundamental frequency estimation performance with each methods showing: a) gross error rate in white noise, b) gross error rate in babble noise, c) estimation accuracy of the voiced speech in white noise and d) babble noise.

that the three methods have similar performance and significant differences are not found in high SNR conditions such as at SNR of 10 dB. In low SNR conditions, however, the performance of PEFAC is superior to the others. At SNR of -5 dB with white noise, PEFAC has a gross error rate of 18 % while the other two methods deteriorate to 40 %. Although gross error rates of PEFAC in babble noise are higher than RAPT, for example, PEFAC is scored 50 % at -5 dB while RAPT is 43 %, accuracy of PEFAC at the same condition is significantly higher than RAPT. The accuracy of PEFAC at -5 dB is 52 % and 63 % in babble noise and white noise respectively while the other two

methods fall to around 20 %. This test results give fair reason to adopt PEFAC for the fundamental frequency estimation in the subsequent experiments. YIN does not employ a post processing while PEFAC and RAPT use a dynamic programming post processing. This may cause a disadvantage to YIN in terms of gross error rate.

3.6 Conclusion of the Chapter

This chapter first discussed the human speech production process and properties of the speech signal attributed to the production process. For example, voiced speech is excited as a periodic signal at the vocal chords and then resonates at resonant cavities which consist of the pharyngeal cavity, the oral cavity and the nasal cavity prior to being radiated from the lips or the nose. Therefore, the characteristics of the signal are determined by the length of the vocal chords and motion of the resonant cavities involving the tongue and teeth. The former characterises the harmonic structure of speech and the latter determines the frequency response of the vocal tract. Alternatively, unvoiced speech is radiated with the property of random noise.

The source-filter model was then discussed with respect to the preceding human speech production process and the properties of speech signals. This model comprises the excitation source and the vocal tract filter. The excitation source is modelled as a periodic pulse train or random noise according to voicing information, and the vocal tract filter is generally designed as an all-pole filter to enable the frequency response to have the formants. LPC and STRAIGHT are representative of this model. Specifically, STRAIGHT showed remarkable performance in the experiment of speech reconstruction from information of the spectral envelopes, the aperiodicity and the fundamental frequency of speech

Alternatively, the sinusoidal model and the HNM, which is one of variants of the sinusoidal model, have also been studied as another approach to model voiced speech. These models are based on the notion where voiced speech is modelled as a summation of harmonic sinusoids, and an experiment showed the sinusoidal model reconstructing speech with high quality.

It has been reported that vocoders based on the source-filter model generally have

buzzy characteristic which is attributed to strong harmonics brought by using a pulse train for excitation, and the sinusoidal model and its variants tend to show better performance [9,50,66]. However, the number of parameters needed by the sinusoidal model and its variants is much more than the source-filter model, and such models do not suit applications which applies statistical models to the parameters, such as model-based speech enhancement because of too much variability. Alternatively, STRAIGHT has resolved the issue of buzzy noise by applying the mixed-excitation model and the parameters required for speech reconstruction, i.e. the spectral envelopes, the aperiodicity coefficients, and the fundamental frequency contour, are suitable to form statistical models because the number of parameters does not vary and the spectral envelopes and the aperiodicity coefficients are able to be transform to the Mel-filterbank domain. Therefore, the proposed method in this thesis adopts the STRAIGHT vocoder for the speech reconstruction process in HMM-based speech synthesis.

Finally, different methods to estimate the fundamental frequency of speech are explored and the experiments have shown that PEFAC has a distinct advantage over RAPT and YIN in the performance to estimate the fundamental frequency of speech in noisy condition.

Chapter 4

Hidden Markov Model-Based Speech Enhancement

This thesis proposes Hidden Markov Model (HMM)-based speech enhancement which is based on the reconstruction-based approach using HMMs for statistical models of speech segments and the STRAIGHT vocoder for the speech production model. HMMs are utilised to both decode the input speech into the state sequence of the models and synthesise speech features of the clean speech from the state sequence. This chapter first discusses an overview of HMMs and then the discussion is extended to the decoding stage and synthesis stage of the method, and finally they are combined to explore HMM-based speech enhancement.

4.1 Introduction

HMMs are statistical time series models which have been successfully applied to various applications to build statistical models of phenomena, e.g. speech, handwriting letters and facial image. One of the most important features of HMMs is that they have a structure that can be expressed as a mathematical description, and it enables a theoretical basis to be established for applications in many and various areas [86].

An HMM comprises a sequence of finite statistical states. Each state can have an independent statistical distribution and transitions among the states are also statistically determined. This structure enables HMMs to model statistical characterisation of

nonstationary signals such as speech and time-varying noise [28]. Specifically, automatic speech recognition (ASR) has successfully employed HMMs for decades [87] while various researches have shown the advantages of using HMMs for text to speech applications recently [9, 50].

HMM-based speech enhancement combines these techniques by first decoding noisy speech using a network of HMMs and then, using the same network of HMMs, synthesises a clean speech signal as illustrated in Figure 4.1. This is motivated by a desire to reduce

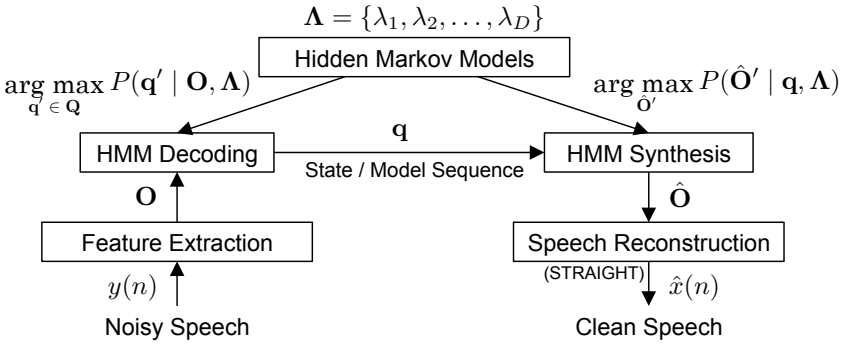

Figure 4.1: A combination of different HMM techniques to build HMM-based speech enhancement.

distortion and artefacts that conventional speech enhancement methods can introduce [1].

The first section of this chapter gives a basic overview of HMMs. Then HMM decoding is discussed in the context of Automatic Speech Recognition (ASR) as its representative application in the second section. The next section explores HMM-based speech synthesis with various experiments prior to the discussion and evaluation of the initial experiments of HMM-based speech enhancement.

4.2 Hidden Markov Models

Figure 4.2 illustrates an ergodic Markov chain with 4 states, in which each state in a model has a transition to every state with a particular probability. $S_1, S_2, \cdots, S_M$ are each state where M denotes the number of the states ($M = 4$ in Figure 4.2). HMMs determine the observation outputs probabilistically with the state transition probability, a_{ij}, probability density of the states, $b_j(\mathbf{o}_n)$, and initial state probability, π_j, where

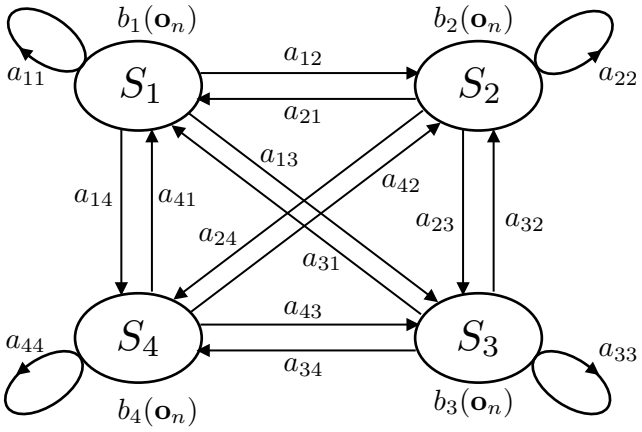

Figure 4.2: 4 state ergodic Markov chain.

$1 \leqslant i, j \leqslant M$. These probabilities are derived as follows.

$$a_{ij} = P\left[q_{n+1} = S_j \mid q_n = S_i\right] \tag{4.1}$$

$$b_j(\mathbf{o}_n) = P\left[\mathbf{o}_n \mid q_n = S_j\right] \tag{4.2}$$

$$\pi_j = P\left[q_0 = S_j\right] \tag{4.3}$$

where q_n represents the state of the model at discrete-time n and $\mathbf{o}_n$ is an observed vector at n. Using this notation, a parameter set of an HMM can be described as

$$\lambda = \{a_{ij}, b_j(\mathbf{o}_n), \pi_j\}, \qquad i, j = 1, 2, \cdots, M \tag{4.4}$$

Considering an ASR application using HMMs, the goal of the function is to calculate probabilities of the observation sequence, given models, and then the most likely sequence of the models is selected. Alternatively, HMM-based speech synthesis applications, such as text to speech (TTS), synthesise the speech features according to a state and model sequence corresponding to the target speech. Therefore, these applications require to reveal the hidden part of HMMs, i.e. state sequence. In addition, HMMs in any applications need to be optimised to observation training vectors in advance. Therefore, the preceding requirements to apply HMMs to speech applications are summarised as the following three problems [86], and the following subsections discuss these problems:

- Given observation sequence, $\mathbf{O} = \left[\mathbf{o}_0^T, \mathbf{o}_1^T, \cdots, \mathbf{o}_N^T\right]^T$, and model, λ, how can the

probability of the observation sequence, $P(\mathbf{O} \mid \lambda)$, be derived?, i.e. decoding.

- Given $\mathbf{O}$ and λ, how can the optimal corresponding state sequence, $\mathbf{q} = \{q_0, q_1, \cdots, q_N\}$, be found?, i.e. find the most likely state sequence.

- Given $\mathbf{O}$ and λ, how can λ be adjusted to maximise $P(\mathbf{O} \mid \lambda)$?, i.e. training.

4.2.1 Probability of the Observation Sequence

Given initial state probabilities, $\pi_j (j = 1, 2, \cdots, M)$, and state sequence, $\mathbf{q}$, for observation period, N, the probability of observation sequence, $\mathbf{O}$, and the probability of state sequence, $\mathbf{q}$, are derived as

$$P(\mathbf{O} \mid \mathbf{q}, \lambda) = b_{q_0}(\mathbf{o}_0) \cdot b_{q_1}(\mathbf{o}_1) \cdots b_{q_{N-1}}(\mathbf{o}_{N-1}) \tag{4.5}$$

$$P(\mathbf{q} \mid \lambda) = \pi_{q_0} \cdot a_{q_0 q_1} \cdot a_{q_1 q_2} \cdots a_{q_{N-2} q_{N-1}} \tag{4.6}$$

Then, the joint probability of $\mathbf{O}$ and $\mathbf{q}$ can be calculated as

$$P(\mathbf{O}, \mathbf{q} \mid \lambda) = P(\mathbf{O} \mid \mathbf{q}, \lambda) \cdot P(\mathbf{q} \mid \lambda) \tag{4.7}$$

The probability of $\mathbf{O}$ given λ is then calculated by summing over all possible state sequence $\mathbf{q} \in \mathbf{Q}$, therefore,

$$P(\mathbf{O} \mid \lambda) = \sum_{\mathbf{q} \in \mathbf{Q}} P(\mathbf{O}, \mathbf{q} \mid \lambda) \tag{4.8}$$

$$= \sum_{\mathbf{q} \in \mathbf{Q}} \pi_{q_0} b_{q_0}(\mathbf{o}_0) \cdot a_{q_0 q_1} b_{q_1}(\mathbf{o}_1) \cdots a_{q_{N-2} q_{N-1}} b_{q_{N-1}}(\mathbf{o}_{N-1}) \tag{4.9}$$

However, this algorithm requires $2NM^N$ calculations [86]. This is unfeasible even for small values of M and N, for example, 3 sec speech framed at 5 ms interval with 3 state-HMMs requires 1200×3^{600} calculations. Therefore, a more efficient procedure using forward and backward variables can be used instead.

A forward variable, $\alpha_n(i)$, at discrete-time n is first defined as

$$\alpha_n(i) = P(\mathbf{o}_0, \mathbf{o}_1, \cdots, \mathbf{o}_n, q_n = S_i \mid \lambda), \qquad i = 1, 2, \cdots, M \tag{4.10}$$

Then, $P(\mathbf{O} \mid \lambda)$ can be obtained inductively using this forward variable as follows

1) Initialisation:

$$\alpha_0(i) = \pi_i b_i(\mathbf{o}_0) \tag{4.11}$$

2) Induction:

$$\alpha_{n+1}(j) = \left[\sum_{i=1}^{M} \alpha_n(i) a_{ij}\right] b_j(\mathbf{o}_{n+1}) \qquad \begin{cases} 0 \leqslant n \leqslant N-2 \\ j = 1, 2, \cdots, M \end{cases} \tag{4.12}$$

3) Termination:

$$P(\mathbf{O} \mid \lambda) = \sum_{i=1}^{M} \alpha_{N-1}(i) \tag{4.13}$$

This algorithm requires M^2N calculations to obtain $\alpha_{N-1}(i)$ [86]. This can be a significantly smaller order than the preceding direct calculation ($2NM^N$), for instance, the computation for the preceding example of the speech can be attained with 5,400 calculations. Similarly, a backward variable, $\beta_n(i)$, can be defined as

$$\beta_n(i) = P(\mathbf{o}_{n+1}, \mathbf{o}_{n+2}, \cdots, \mathbf{o}_{N-1} \mid q_n = S_i, \lambda) \qquad i = 1, 2, \cdots, M \tag{4.14}$$

This backward variable also can be led to $P(\mathbf{O} \mid \lambda)$ inductively as follows:

1) Initialisation:

$$\beta_{N-1}(i) = 1 \tag{4.15}$$

2) Induction:

$$\beta_n(i) = \sum_{j=1}^{M} a_{ij} b_j(\mathbf{o}_{n+1}) \beta_{n+1}(j) \qquad 0 \leqslant n \leqslant N-2 \tag{4.16}$$

3) Termination:

$$P(\mathbf{O} \mid \lambda) = \sum_{i=1}^{M} \pi_i b_i(\mathbf{o}_0) \beta_0(i) \tag{4.17}$$

This backward algorithm also requires the calculation order of M^2N to obtain $\beta_0(i)$ as well as the forward algorithm.

4.2.2 Optimal State Sequence

There are several possible methods to find the optimal state sequence corresponding to the observation sequence from $\mathbf{O}$ and λ. One possible criterion is to select the states, q_n, which are individually most likely. A variable, $\gamma_n(i)$, is defined as follows to calculate the state sequence, $\mathbf{q}$, with this criterion.

$$\gamma_n(i) = P(q_n = S_i \mid \mathbf{O}, \lambda) \tag{4.18}$$

where

$$\sum_{i=1}^{M} \gamma_n(i) = 1 \tag{4.19}$$

Then, $\gamma_n(i)$ can be exploited to find the most likely state at time n as

$$q_n = \arg\max_{1 \leqslant i \leqslant M} [\gamma_n(i)], \qquad n = 0, 1, \cdots, N-1 \tag{4.20}$$

Equation (4.20) takes only the instantaneous most likely state into account. Therefore, it might be possible that the derived state sequence is invalid for the given models in some cases, for example, when the model has state transitions which are zero probability. To solve this problem, the probability of occurrence of the entire state sequence should be taken into account, and the Viterbi algorithm is widely used to find the single best state sequence by maximising $P(\mathbf{q} \mid \mathbf{O}, \lambda)$. A quantity, $\delta_n(i)$, is defined to derive Viterbi algorithm as

$$\delta_n(i) = \max_{q_0, q_1, \cdots, q_{n-1}} P\left[\{q_0, q_1, \cdots, q_{n-1}\}, q_n = S_i \mid \{\mathbf{o}_0, \mathbf{o}_1, \cdots, \mathbf{o}_n\}, \lambda\right] \tag{4.21}$$

where $i = 1, 2, \cdots, M$. The induction of this sequence is derived as

$$\delta_{n+1}(j) = \max_i [\delta_n(i) a_{ij}] \cdot b_j(\mathbf{o}_{n+1}), \qquad j = 1, 2, \cdots, M \tag{4.22}$$

Technically, the algorithm requires an array, $\psi_n(j)$, to keep track of each maximised argument in Equation (4.22) to allow the state sequence to be obtained. The complete procedure of the algorithm is given as follows [86].

1) Initialisation:

$$\delta_0(i) = \pi_i b_i(\mathbf{o}_0), \qquad i = 1, 2, \cdots, M \tag{4.23}$$

$$\psi_0(i) = 0 \tag{4.24}$$

2) Recursion:

$$\delta_n(j) = \max_{1 \leqslant i \leqslant M} \left[\delta_{n-1}(i) a_{ij}\right] b_j(\mathbf{o}_n) \tag{4.25}$$

$$\psi_n(j) = \arg\max_{1 \leqslant i \leqslant M} \left[\delta_{n-1}(i) a_{ij}\right] \tag{4.26}$$

where

$$1 \leqslant n \leqslant N - 1 \tag{4.27}$$

$$j = 1, 2, \cdots, M \tag{4.28}$$

3) Termination:

$$R = \max_{1 \leqslant i \leqslant M} \left[\delta_{N-1}(i)\right] \tag{4.29}$$

$$q_{N-1} = \arg\max_{1 \leqslant i \leqslant M} \left[\delta_{N-1}(i)\right] \tag{4.30}$$

4) Path backtracking:

$$q_n = \psi_{n+1}(q_{n+1}) \qquad n = N - 2, N - 3, \cdots, 0 \tag{4.31}$$

4.2.3 Training of the HMMs

There is no known analytical method to adjust the HMM parameters in order to maximise the probability of the observation sequence, $P(\mathbf{O} \mid \lambda)$ [86]. However, some iterative procedures such as the Baum-Welch method and Expectation-Modification (EM) method are known to optimise λ such that $P(\mathbf{O} \mid \lambda)$ can be locally maximised. This section discusses the Baum-Welch method for the HMM training process.

A quantity, $\xi_n(i, j)$, is defined in order to describe the procedure for training the HMM parameters, λ. This quantity denotes the probability such that the model is in

state S_i and S_j at the observation time n and $n+1$ respectively. Therefore, $\xi_n(i,j)$ can be derived as

$$\xi_n(i,j) = P(q_n = S_i, q_{n+1} = S_j \mid \mathbf{O}, \lambda) \tag{4.32}$$

$\xi_n(i,j)$ can also be given in terms of the forward-backward variables as

$$\begin{aligned} \xi_n(i,j) &= \frac{\alpha_n(i) a_{ij} b_j(\mathbf{o}_{n+1}) \beta_{n+1}(j)}{P(\mathbf{O} \mid \lambda)} && (4.33) \\ &= \frac{\alpha_n(i) a_{ij} b_j(\mathbf{o}_{n+1}) \beta_{n+1}(j)}{\sum_{i=1}^{M} \sum_{j=1}^{M} \alpha_n(i) a_{ij} b_j(\mathbf{o}_{n+1}) \beta_{n+1}(j)} && (4.34) \end{aligned}$$

Because of the definition of $\gamma_n(i)$ in Equation (4.18), it can be obtained by summing $\xi_n(i,j)$ over j

$$\gamma_n(i) = \sum_{j=1}^{M} \xi_n(i,j) \tag{4.35}$$

If $\gamma_n(i)$ is summed over n from 0 to $N-2$, it gives the expected number of transitions made from S_i, and if $\gamma_n(i)$ is summed over n from 0 to $N-1$, it gives the expected number of the times that state S_i is visited. Similarly, summing $\xi_n(i,j)$ over n from 0 to $N-2$ gives the expected number of transition from S_i to S_j. These expected numbers can be exploited to re-estimate the HMM parameters to maximise $P(\mathbf{O} \mid \lambda)$. Namely, a set of re-estimation formulas for the HMM parameters, π_j, a_{ij}, and $b_j(\mathbf{o}_n)$ are derived as

$$\begin{aligned} \bar{\pi}_j &= \gamma_0(j) && (4.36) \\ \bar{a}_{ij} &= \frac{\sum_{m=0}^{N-2} \xi_m(i,j)}{\sum_{m=0}^{N-2} \gamma_m(i)} && (4.37) \end{aligned}$$

$$\bar{b}_j(\mathbf{o}_n) = \frac{\sum_{\{m:\ \mathbf{o}_m = \mathbf{o}_n\}} \gamma_m(j)}{\sum_{m=0}^{N-1} \gamma_m(j)} \tag{4.38}$$

where $1 \leqslant i, j \leqslant M$ and $\{m:\ \mathbf{o}_m = \mathbf{o}_n\}$ represents time, m, such that $\mathbf{o}_n$ is observed then.

Alternatively, Equations (4.36) to (4.38) can be derived by maximising Baum's auxiliary function [86]

$$\mathbf{G}(\lambda, \bar{\lambda}) = \sum_{\mathbf{q} \in \mathbf{Q}} P(\mathbf{q} \mid \mathbf{O}, \lambda) \log \left[P(\mathbf{O}, \mathbf{q} \mid \bar{\lambda}) \right] \tag{4.39}$$

over $\bar{\lambda}$, and this maximisation brings a increase of the probability of $\mathbf{O}$ as follows.

$$\max_{\bar{\lambda}} \left[\mathbf{G}(\lambda, \bar{\lambda}) \right] \Rightarrow P(\mathbf{O} \mid \bar{\lambda}) \geqslant P(\mathbf{O} \mid \lambda) \tag{4.40}$$

In this way, iterative computation of $\bar{\lambda}$ in place of λ can improve the probability of the observation until some limiting point, and such a re-estimated $\bar{\lambda}$ is called a maximum likelihood estimate of the HMM.

At this point the HMM is now trained and can be applied to applications such as ASR and speech synthesis, which are discussed in the following sections.

4.3 HMM decoding and Automatic Speech Recognition

HMM decoding is one of the key processes in the proposed HMM-based speech enhancement shown in Figure 4.1 and this technique has successfully been applied to HMM applications such as ASR. This section briefly explores an application of ASR as a typical example of HMM decoding. A framework of ASR is illustrated in Figure 4.3. Input

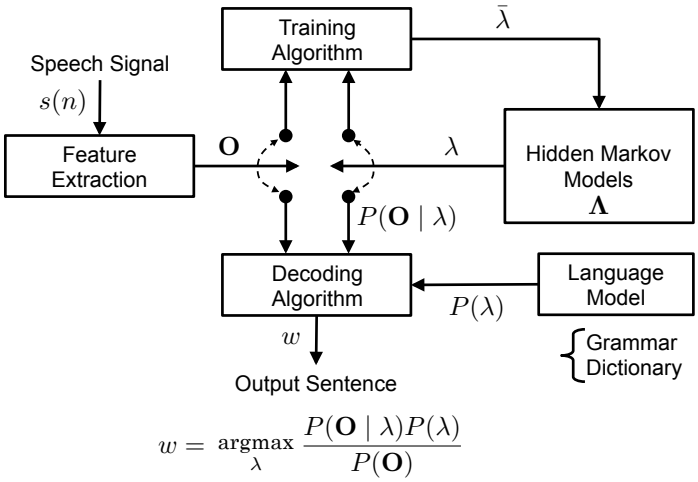

Figure 4.3: A framework of ASR.

speech, $s(n)$, is first framed and transformed to a sequence of feature vectors, $\mathbf{O}$ at the feature extraction process. The following process is divided into the training process and the decoding process. The training process is an offline process to optimise HMMs, λ, such that $P(\mathbf{O} \mid \lambda)$ is maximised by using the set of training speech and its word labels

while the decoding process is an online process to determine output words or sentence, w, corresponding to $\mathbf{O}$. In this process w is determined with the probabilities of acoustic model, $P(\mathbf{O} \mid \lambda)$, and language model, $P(\boldsymbol{\lambda})$, in the sense of maximum likelihood estimation. The acoustic models are trained at the training process while the language models are build as, for example, a network of linguistic grammar and a dictionary of vocabulary. The following subsections discuss each of these processes and experimental results are then evaluated at the end of this section.

4.3.1 Feature Extraction

Although different choices of feature vectors of speech exist based on, for instance, a Mel-frequency cepstrum, a linear-frequency cepstrum, a linear prediction cepstrum or a linear-prediction spectrum to represent the acoustic feature of speech [88], it has been reported that the feature vectors based on a Mel-frequency cepstrum represent the acoustic feature of speech very well for ASR applications [88] and specifically, Mel-Frequency Cepstral Coefficients (MFCCs) have successfully been applied to practical applications [89, 90]. This subsection discusses a process to extract sequence of MFCC vectors, $\mathbf{O}$, as the input of the ASR decoding process.

An overview of a feature extraction process to obtain MFCC vectors from input speech is shown in Figure 4.4. Discrete-time domain speech, $s(n)$ is first split into a

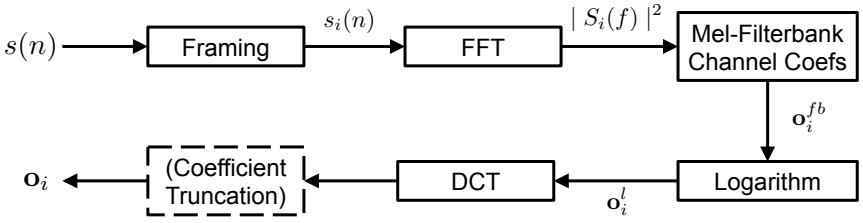

Figure 4.4: A block diagram to extract MFCC vectors.

series of frames with a Hamming window to obtain i-th frame of the speech, $s_i(n)$, where $i = 0, 1, 2, \ldots, I-1$ and I denotes the number of the frames. STFT analysis then derives

the power spectrum of $s_i(n)$ as

$$S_i(f) = \mathscr{F}[s_i(n)] \tag{4.41}$$

$$|\, S_i(f) \,|^2 = S_i(f) S_i^*(f) \tag{4.42}$$

where f denotes frequency bins of the Fourier transform, $\mathscr{F}$.

It is known empirically that human ears resolve frequencies non-linearly and that frequency resolution is lower at higher frequencies, and this characteristic is approximated by the Mel-scale [91] as Figure 4.5(a).

$$f_{mel} = 2595 \log_{10}\left(1 + \frac{f_{lin}}{700}\right) \tag{4.43}$$

where f_{lin} is a frequency in the linear space domain whereas f_{mel} represents the corresponding frequency in the Mel-scaled domain. MFCC vectors employ this characteristic to make the performance of ASR applications robust. To impose this feature in the observation vectors, a Mel-filterbank is first formed by setting equally spaced triangular filters in the Mel-frequency domain with 50 % overlapping, as shown in Figure 4.5(b), and then the Mel-filterbank is applied to $|\, S_i(f) \,|^2$ to obtain energy in j-th filterbank channel, $o_i^{fb}(j)$, which constitutes a vector of the Mel-filterbank coefficients, $\mathbf{o}_i^{fb} = [o_i^{fb}(0), o_i^{fb}(1), \ldots, o_i^{fb}(M-1)]^T$ where M is the number of filterbank channels. A

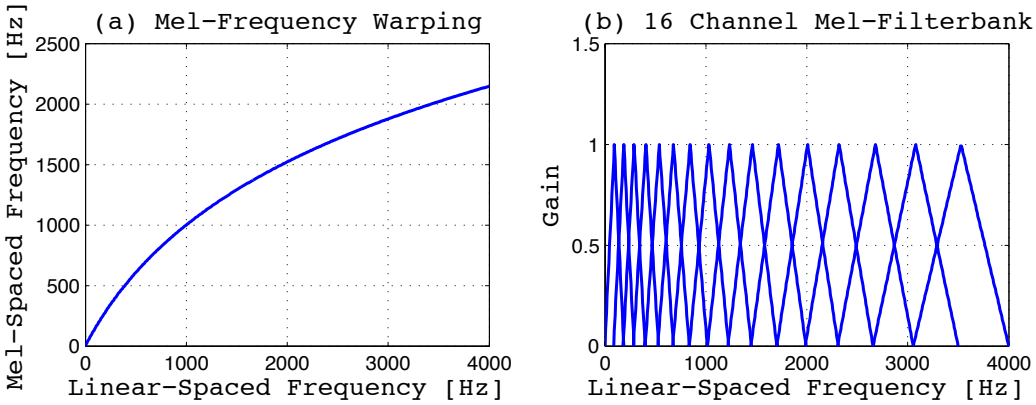

Figure 4.5: a) shows Mel-scale frequency warping while b) illustrates a 16 channel Mel-filterbank

logarithm is then taken to derive a log-Mel-filterbank coefficient vector, $\mathbf{o}_i^l$ as

$$\mathbf{o}_i^l = \left[\log o_i^{fb}(0), \log o_i^{fb}(1), \ldots, \log o_i^{fb}(M-1)\right]^T \tag{4.44}$$

$$= \left[o_i^l(0), o_i^l(1), \ldots, o_i^l(M-1)\right]^T \tag{4.45}$$

where $o_i^l(j)$ is a log-Mel-filterbank coefficient in j-th filterbank channel.

Speech signals can be modelled as a convolution of an excitation signal and vocal tract filter coefficients as shown in Figure 3.4, and this is represented as a multiplication in the frequency domain as

$$S_i(f) = H_i(f)E_i(f) \tag{4.46}$$

where $H_i(f)$ and $E_i(f)$ represents the frequency response of a vocal tract filter and the spectrum of an excitation signal of speech respectively. Taking logarithm of Equation (4.46), the components of excitation and vocal tract can be separable as a sum.

$$\log S_i(f) = \log H_i(f) + \log E_i(f) \tag{4.47}$$

Therefore, the log operation to derive $\mathbf{o}_i^l$ separates the components of excitation and vocal tract in $\mathbf{o}_i^{fb}$ as a sum.

Finally, MFCC vector, $\mathbf{o}_i$, is derived by applying a discrete cosine transform (DCT) to $\mathbf{o}_i^l$ as

$$\mathbf{o}_i = \left[o_0^i, o_1^i, \ldots, o_{M-1}^i\right]^T \tag{4.48}$$

where

$$o_j^i = \sqrt{\frac{2}{M}} \sum_{k=0}^{M-1} o_i^l(k) \cos\left(\frac{\pi(2k+1)j}{2M}\right) \tag{4.49}$$

By this DCT operation, i.e. transform to cepstrum, components of the spectral envelope in $\mid S_i(f) \mid^2$, which changes slowly along the frequency axis, are stored into low quefrency coefficients whereas harmonic components, which changes quickly along the frequency axis, are stored into high quefrency coefficients. For use in ASR, high quefrency elements in $\mathbf{o}_i$ are truncated since words or phonemes are discriminated by motion of vocal tract, i.e. spectral envelope, for example, [90] employs 23 channel Mel-filterbank in the process to calculate MFCC vectors, $\mathbf{o}_i$, and then coefficients from o_{13}^i to o_{22}^i are removed to

extract 13 coefficient MFCC vectors from 8 kHz-sampled speech. Figure 4.6 illustrates that the truncation of high quefrency coefficients extracts the energy of the spectral envelope. For better visualisation, the figure shows an example with a spectral feature in the linear-frequency domain without applying a Mel-filterbank rather than MFCCs, but the underlying notion is identical to MFCCs. Subplot (a) shows the spectral magnitude of a short-time (20 ms) segment of the natural speech of the sound /b/ in "blue" uttered by a male speaker. Subplot (b) is the log spectral magnitude of the same speech segment. Subplot (c) shows the cepstrum of the speech segment obtained by applying DCT to (b). Quefrency bins corresponding to more than 1 ms are then set to zero and this is shown in subplot (d). Subplot (e) and (f) are log spectral magnitude and linear spectral magnitude obtained with the inverse transform of (d). These match the envelope of the original spectra and the harmonic structure in the original spectrum, i.e pitch, which is not useful for identifying vocal tract motion has been removed.

Then, MFCC observation vector, $\mathbf{O}$, during N frames is formed as

$$\mathbf{O} = \left[\mathbf{o}_0^T, \mathbf{o}_1^T, \ldots, \mathbf{o}_{N-1}^T\right]^T \tag{4.50}$$

Additionally, previous researches have shown that adding dynamic features, such as velocity and acceleration derivatives, into feature vectors improves the robustness of ASR performance against noise [92]. In this case, $\mathbf{O}$ is formed as

$$\mathbf{O} = \left[\mathbf{o}_0^T, \Delta\mathbf{o}_0^T, \Delta^2\mathbf{o}_0^T, \mathbf{o}_1^T, \Delta\mathbf{o}_1^T, \Delta^2\mathbf{o}_1^T, \ldots, \mathbf{o}_{N-1}^T, \Delta\mathbf{o}_{N-1}^T, \Delta^2\mathbf{o}_{N-1}^T\right]^T \tag{4.51}$$

where $\Delta\mathbf{o}_i$ and $\Delta^2\mathbf{o}_i$ are a velocity derivative vector and an acceleration derivative vector of $\mathbf{o}_i$ respectively.

4.3.2 HMM Training

Phonemes constituting words are characterised by a series of motions of resonant cavities as mentioned in Section 3.2, therefore, in statistical models of words or sub-words, each state in the models represents a shape of the vocal tract cavities at a point and these states occurs in order along a time sequence. Thus, in the training process, the initial prototype model is designed as a left-right Markov chain, as shown in Figure 4.7, rather

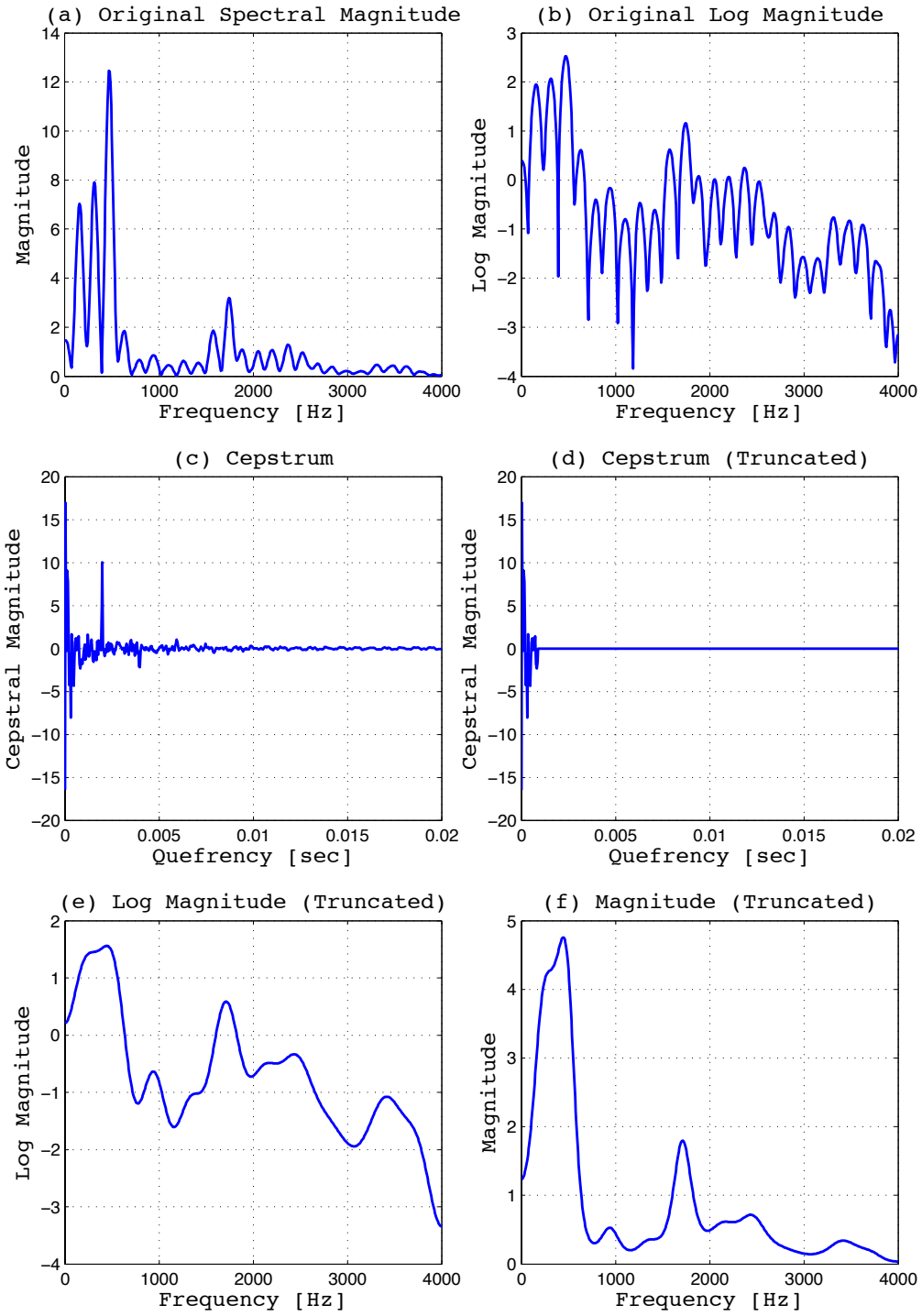

Figure 4.6: Extraction of a spectral envelope by truncating high quefrency bins of a cepstrum showing: a) spectral magnitude of speech, b) log spectral magnitude, c) cepstrum obtained with DCT, d) cepstrum in which quefrency bins corresponding to more than 1 ms are truncated and then padded with zeros, e) and f) log and linear spectral magnitude inverse-transformed from d).

than employing a ergodic HMM represented in Figure 4.2. The structure of left-right HMMs constrains the possible state sequences as follows.

$$\pi_1 = 1 \tag{4.52}$$

$$\pi_j = 0, \quad j \neq 1 \tag{4.53}$$

$$a_{jk} > 0, \quad k = j \text{ or } k = j+1 \tag{4.54}$$

$$a_{jk} = 0, \quad \text{otherwise} \tag{4.55}$$

$$q_{N-1} = S_M \tag{4.56}$$

where M is the number of the states while N denotes the number of the feature vectors (frames) constituting a word or a sub-word. Now, the prototype is determined by observation probabilities modelled by a Gaussian distributions at each state, $b_j(\mu_j, \mathbf{\Sigma}_j)$ where μ_j and $\mathbf{\Sigma}_j$ represent the mean value and the covariance matrix of the Gaussian distribution at state j, and state transition probabilities, a_{jj} and $a_{j(j+1)}$. Alternatively,

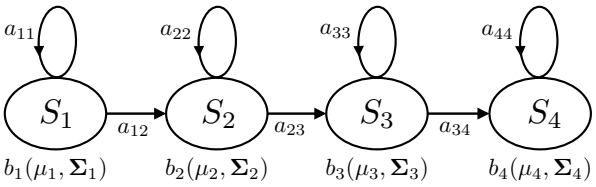

Figure 4.7: 4 state left-right HMM.

observation probabilities in each state can be modelled by multiple Gaussian distributions, i.e. Gaussian mixture models (GMMs), but HMMs discussed in this section employ observation probabilities, $b_j(\mu_j, \mathbf{\Sigma}_j)$, modelled by a single Gaussian distribution.

The prototype is then optimised to each word or sub-word model according to the corresponding parts in the training data set, which are labelled on transcripts in the data set. The trained HMMs $\bar{\lambda}$ are derived as

$$\bar{\lambda} = \arg\max_{\lambda} P(\mathbf{O} \mid \lambda) \tag{4.57}$$

where

$$\mathbf{O} = \left[\mathbf{o}_0^T, \mathbf{o}_1^T, \cdots, \mathbf{o}_{N-1}^T\right]^T \tag{4.58}$$

This optimisation is achieved by a two stage process. At the first stage $\mathbf{O}$ is first equally divided and assigned into each state in order to obtain the initial path of the state sequence. This initial path is then renewed by using the Viterbi algorithm mentioned in Section 4.2 as

$$\bar{\mathbf{q}} = \arg\max_{\mathbf{q}\in\mathbf{Q}} P\left(\mathbf{q} \mid \mathbf{O}, \boldsymbol{\lambda}\right) \tag{4.59}$$

where $\mathbf{q}$ is the initial state sequence while $\bar{\mathbf{q}}$ is renewed state sequence, and $\mathbf{Q}$ represents overall possible state sequences. According to $\bar{\mathbf{q}}$, $\mathbf{O}$ is reassigned into each state and it changes the parameters of the Gaussian distribution in each state, $b_j(\mu_j, \boldsymbol{\Sigma}_j)$, with Equation (4.57). The first stage of the model optimisation can be achieved by iterating the above procedure to renew the state sequence and the Gaussian distributions.

At the second stage of the process, the renewed HMMs at the first stage are further optimised iteratively by using the Baum-Welch algorithm mentioned in Section 4.2.

A model configuration, such as employing whole-word models or sub-word models and the number of the states in the models, needs to be decided at the beginning of the training process. These are important factors to determine the performance of decoding, therefore, different configurations of HMMs are examined and evaluated in Section 4.3.4.

4.3.3 HMM Decoding

The HMM decoding process finds the most likely model and state sequence corresponding to a sequence of feature vectors during an utterance. Therefore, in this process, λ denotes a possible sequence of the trained models and states, and $\mathbf{O}$ denotes a sequence of feature vectors during an utterance while they represented a single model and a feature vector sequence corresponding to the single model in an utterance in the training process.

This process calculates $P(\lambda \mid \mathbf{O})$ for each possible model and state sequence and selects the most likely model and state sequence, $\hat{\lambda}$, given the observed sequence of the feature vectors during an utterance, $\mathbf{O}$, as

$$\hat{\lambda} = \arg\max_{\lambda\in\Lambda} P\left(\lambda \mid \mathbf{O}\right) \tag{4.60}$$

where $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_D\}$ represents the group of all D possible combinations of models and states in the system, and if the HMMs consist of whole-word models, $\hat{\lambda}$ forms the spoken sentences corresponding to the input speech whereas $\hat{\lambda}$ forms words and then the sentences if the HMMs comprise sub-word models.

Applying the Bayes rule to Equation (4.60), the likelihood of each model and state sequence is derived as

$$P(\lambda \mid \mathbf{O}) = \frac{P(\mathbf{O} \mid \lambda) P(\lambda)}{P(\mathbf{O})} \tag{4.61}$$

where $P(\mathbf{O} \mid \lambda)$ corresponds to the acoustic model of the system which can be calculated by applying the algorithm, which uses forward and backward variables, mentioned in Section 4.2 while $P(\lambda)$ represents the language model of the system which linguistically constrains the network of the HMMs including their positions and combinations. The language model is, for instance, determined by a dictionary and grammar. The dictionary contains all the words covered in the application and defines each word as a combination of sub-words, e.g. phonemes. Therefore, it can constrain the selection of $\hat{\lambda}$, such that the resultant model and state sequence forms only the words defined in the dictionary and combinations which are not listed in the dictionary are eliminated from the decision. The grammar depends on characteristics of languages or data sets and is modelled based on statistical probabilities of word occurrences in order to enable $\hat{\lambda}$ to be selected with a linguistic perspective. The decoding accuracy can be further improved by extending the language model to bigrams or trigrams but it makes the footprint of the system larger [93].

$P(\mathbf{O})$ in Equation (4.61) is independent of λ, thus, $\hat{\lambda}$, is derived by comparing the product of the probabilities of the acoustic model and the language model for each λ as

$$\hat{\lambda} = \arg\max_{\lambda \in \Lambda} P(\mathbf{O} \mid \lambda) P(\lambda) \tag{4.62}$$

HMMs with different configurations of acoustic models and language models are examined and evaluated in the next section.

4.3.4 Experimental Evaluation on ASR

Understanding the performance of ASR and practical techniques to achieve accurate decoding is important to apply HMM decoding to HMM-based speech enhancement. For this purpose, various experiments are conducted in order to evaluate ASR performances with different feature vector settings and model configurations in this subsection.

Experiments use speech from four speakers in the GRID database [43], two males and two females, to form speaker dependent models and those speech data are downsampled to 8 kHz. Sentences in the data set conform to a particular grammatical structure (GRID grammar) of *command→colour→preposition→letter→digit→adverb*. From the 1000 utterances from each speaker, 800 are used for training and the remainder are for testing. Tests are carried out in white noise and babble noise at SNRs from -5 dB to 10 dB. In each experiment, the set of HMMs, Λ, is trained on feature vectors, $\mathbf{O}$, that are extracted from both clean and noisy speech which are in the same noise condition as the test so that the noisy speech input can be decoded by Λ including noise-matched HMMs.

4.3.4.1 Feature Vector settings

To evaluate HMM decoding performance with different feature vector settings, five configurations of MFCC vectors as shown in Table 4.1 are first examined with the trained set of 16 state whole-word single Gaussian HMMs. These configurations set different

MFCC Config.	Mel-FB Channels	MFCC Coeffs	Derivatives
MFCC16-16	16	16	Static
MFCC23-23	23	23	
MFCC40-40	40	40	
MFCC60-40	60	60	
MFCC128-128	128	128	

Table 4.1: Configurations of MFCC coefficients as the observation vectors without coefficient truncation.

number of Mel-filterbank channels without truncation of the MFCC coefficients. The frame length and frame interval of each vector are set equal to 25 ms and 5 ms respectively, The spectra of each frame are derived with 1024 point Fourier transform and the vectors comprise only static coefficients of MFCC. Λ consists of 52 whole-word models

and possible sequences of the models are constrained by the language model of GRID grammar.

The word accuracy, W_{acc}, is calculated as follows.

$$W_{acc} = \frac{W_N - (W_D + W_S + W_I)}{W_N} \times 100\ \% \tag{4.63}$$

where W_D, W_S and W_I are the total number of deletion errors, substitution errors and insertion errors respectively, and W_N denotes the total number of words in the reference transcripts. Figure 4.8 shows the ASR word accuracy in those different settings. Feature

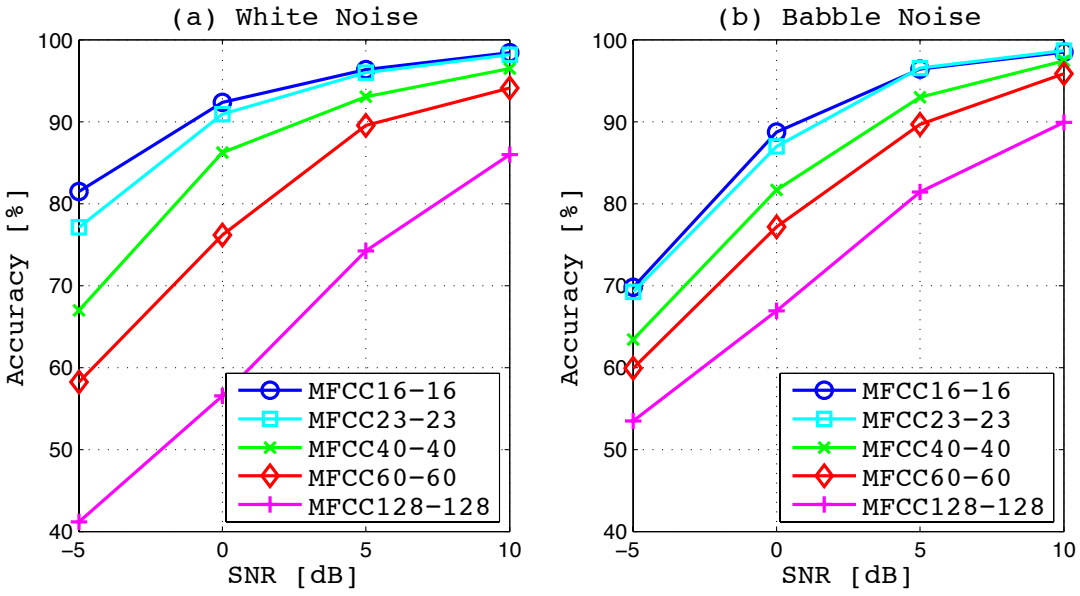

Figure 4.8: ASR accuracy with 16-state whole-word HMMs and different MFCC settings in A) white noise and b) babble noise. The frame interval is 5 ms.

vectors of 16, 23 and 40 coefficient MFCCs have the highest accuracy at high SNRs but 16 coefficient MFCC vectors are most robust to noise followed by 23 coefficient MFCC vectors. 60 coefficient MFCCs are inferior to the preceding three settings over the SNR range and 128 coefficient MFCCs show the worst performance of the five settings. This seems to be attributed to too much variability of the feature vectors.

The next experiment examine the ASR performance with different settings of MFCC observation vectors in which high order coefficients are truncated. Table 4.2 shows each setting for the test. 16, 23 and 40 coefficient MFCC vectors are first extracted (MFCC16-

MFCC Config.	Mel-FB Channels	MFCC Coeffs	Derivatives
MFCC16-16	16	16	Static
MFCC16-13		13	
MFCC16-8		8	
MFCC23-23	23	23	Static
MFCC23-13		13	
MFCC23-8		8	
MFCC40-40	40	40	Static
MFCC40-13		13	
MFCC40-8		8	

Table 4.2: Configurations of MFCC coefficients as the observation vectors with coefficient truncation.

16 / MFCC23-23 / MFCC40-40). The first 13 coefficients of them are then retained and other coefficients are eliminated (MFCC-16-13 / MFCC23-13 / MFCC40-13) in order to extract only the coefficients whose period of the DCT basis is more than $f_s/24$ where f_s denotes the sampling frequency. These vectors then have further truncation to extract only first 8 coefficients whose period of the DCT basis is more than $f_s/12$ (MFCC-16-8 / MFCC-23-8 / MFCC40-8).

The results of ASR with these MFCC settings are illustrated in Figure 4.9 showing that the truncation of the high order MFCC coefficients improves the robustness to the noise in each MFCC setting. The differences in improvement between the truncations of 8 coefficients and 13 coefficients are very little in white noise. In babble noise, however, the truncations of 8 coefficients give more robustness than the truncations of 13 coefficients in each settings, and MFCC16-8 shows the best ASR accuracy in total.

4.3.4.2 Acoustic Model Settings for Whole-Word HMMs

The following experiments examine different acoustic models by changing the number of states in whole-word single Gaussian HMMs, and employ MFCC16-8 as the observation vectors in which the frame length and interval are 25 ms and 5 ms respectively. The model configurations for the test are shown in Table 4.3.

Figure 4.10 shows the ASR results with these model configurations in white noise and babble noise. The influence of state numbers are little in the range of settings but a choice between 16 and 28 states seems to be better.

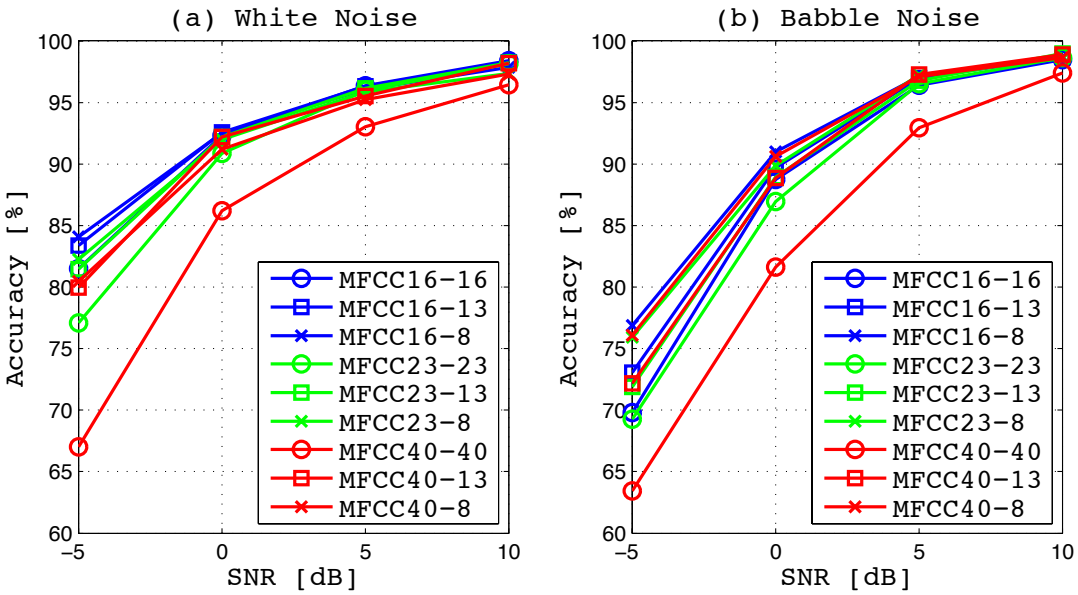

Figure 4.9: ASR accuracy with 16-state whole-word HMMs and different MFCC truncation settings in A) white noise and b) babble noise. The frame interval is 5 ms.

HMM Config.	Number of States	Feature Vector
WORD8	8	
WORD12	12	
WORD16	16	
WORD20	20	MFCC16-8
WORD24	24	
WORD28	28	
WORD32	32	

Table 4.3: Configurations for whole-word HMMs.

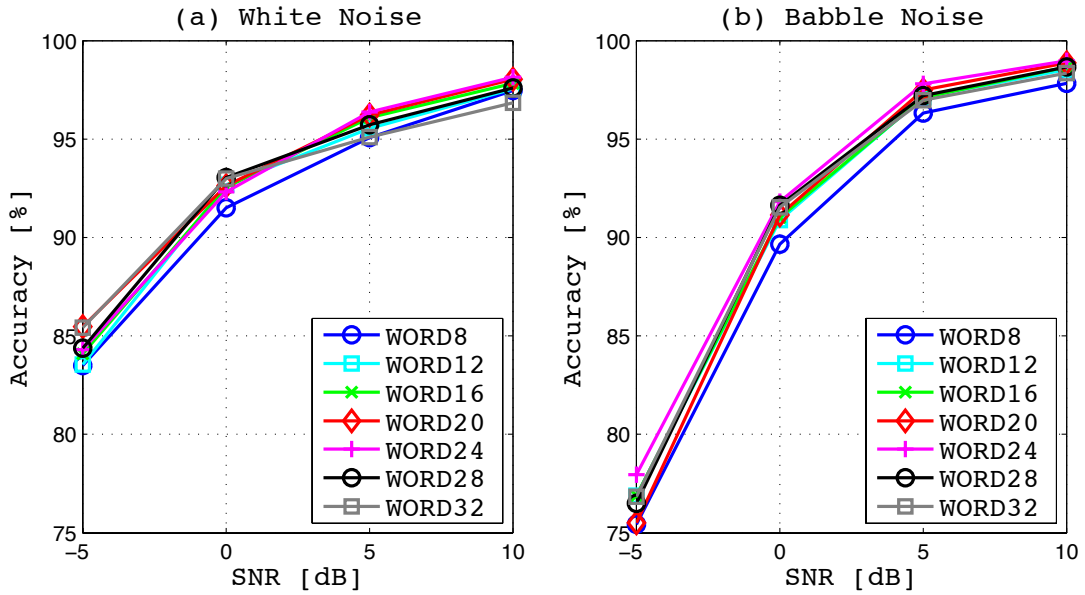

Figure 4.10: ASR accuracy with different whole-word HMM settings in a) white noise and b) babble noise. The configuration of feature vectors is MFCC16-8 the frame interval of which is equal to 5 ms.

A proper choice of the state numbers could depend on the frame interval, therefore, the frame interval of the feature vectors are changed to 10 ms and then the ASR performance is examined with each state settings. The result is shown in Figure 4.11. In this frame interval, the number of states should be selected between 12 and 16 for better ASR accuracy, and the results also show that the ASR accuracy with feature vectors framed at 10 ms interval is almost same as the case of 5 ms interval as long as the number of states in HMMs is properly modelled.

Alternatively, Figure 4.12 shows the results of ASR test in which feature vectors framed at 1 ms interval are employed and the state configurations in Table 4.4 are added. In the case of using the feature vectors framed at 1 ms interval, the influence of the

HMM Config.	Number of States	Feature Vector
WORD36	36	MFCC16-8
WORD40	40	

Table 4.4: Added configurations for the tests with 1 ms-framed feature vectors.

number of states in HMMs on the performance is very little as the number of states is between 24 and 40, but the robustness to the noise is slightly lower than the case of 5

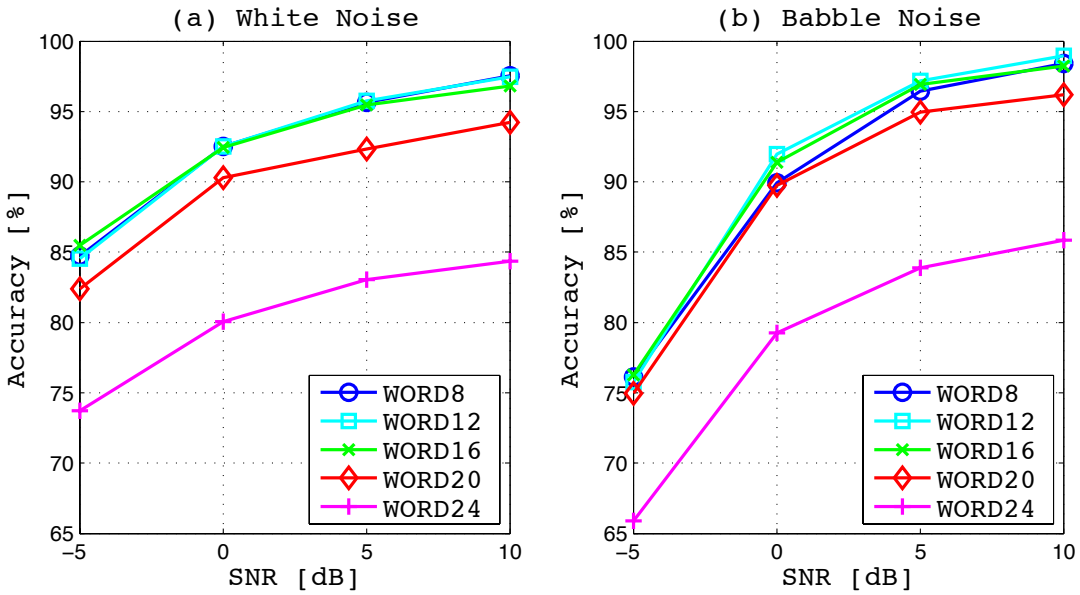

Figure 4.11: ASR accuracy with different whole-word HMM settings in a) white noise and b) babble noise. The configuration of feature vectors is MFCC16-8 the frame interval of which is equal to 10 ms.

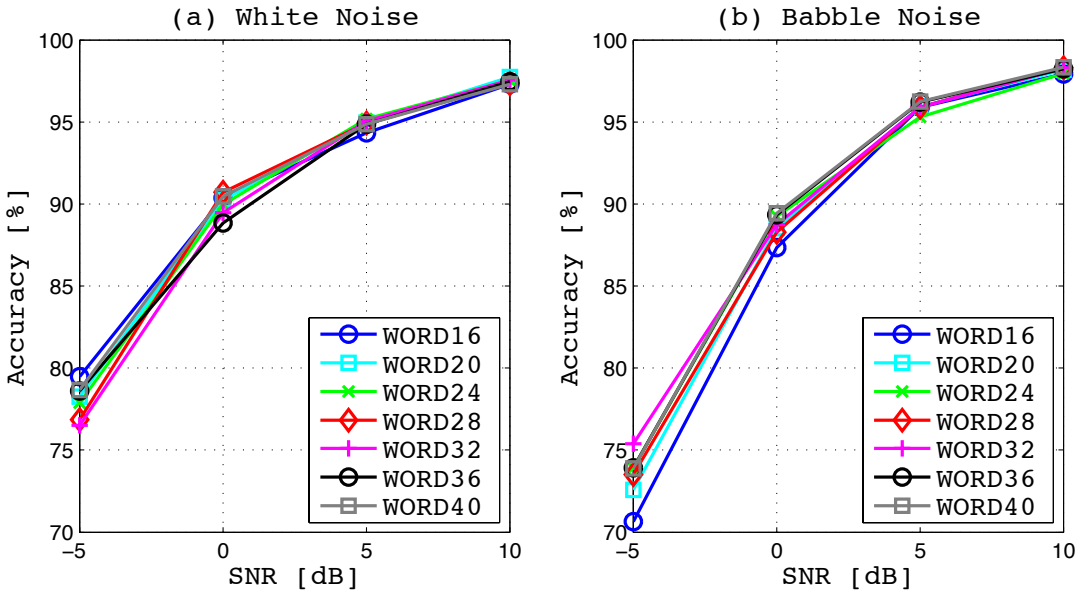

Figure 4.12: ASR accuracy with different whole-word HMM settings in a) white noise and b) babble noise. The configuration of feature vectors is MFCC16-8 the frame interval of which is equal to 1 ms.

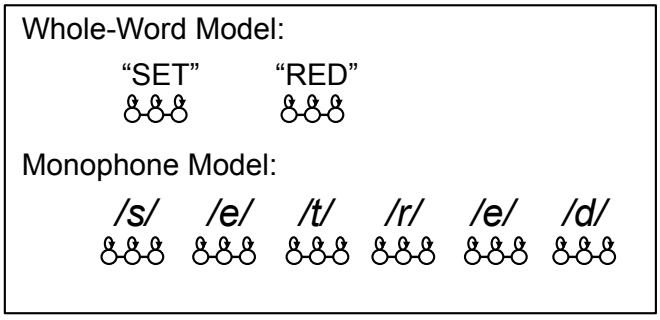

Figure 4.13: A structure of monophone models.

ms and 10 ms frame intervals. For example, the accuracy at SNR of 0 dB or less than 0 dB is 3 pt. to 5 pt. lower than the case of the preceding frame intervals.

4.3.4.3 Acoustic Model Settings for Monophone HMMs

The preceding experiments examined the ASR performance with whole-word HMMs. Applications using whole-word HMMs, however, have to define all the words with which the applications can deal in advance, and training data are required to include all of those words. This may be accepted for applications which work with limited vocabularies for a specific purpose but not practical for ASR of general spoken language. Additionally, if a decoding result has an error in whole-word HMM-based speech enhancement, the influence of the error in the enhanced speech spans the whole-word. To resolve the problems above, the acoustic model is extended to sub-word single Gaussian models at this point. All the words in the GRID vocabulary are first resolved into phonemes the number of which is 35 by referring to a dictionary, which lists all the words and corresponding phoneme sequences contained in GRID database. The segments in the training data set corresponding to each phoneme are then extracted to constitute a group of statistical models, i.e. monophone models, as shown in Figure 4.13. These monophone HMMs are trained with various settings configured by different state numbers in HMMs as shown in Table 4.5, and feature vectors of speech are represented by the setting of MFCC16-8 framed at 10 ms, 5 ms and 1 ms as well as the tests with the whole-word HMMs.

In the decoding process, the resultant model sequence is constrained by the dictio-

HMM Config.	Number of States	Feature Vector
MONO3	3	MFCC16-8
MONO5	5	
MONO7	7	
MONO9	9	
MONO12	12	
MONO16	16	
MONO20	20	
MONO24	24	

Table 4.5: Configurations for monophone HMMs.

nary and the GRID grammar, which form the language model of the system. Therefore, the choice of the most likely model, i.e. phoneme, is limited such that the resultant sequence of phonemes forms a word in the GRID vocabulary, meanwhile, the word matches the GRID grammar as well.

The ASR results with the test conditions above are shown in Figure 4.14 in which the accuracy of ASR is calculated by Equation (4.63). The subplots in the first column are the test results in white noise while the second column is for the test results in babble noise. Subplots (a) and (b) show the ASR accuracy with the feature vectors framed at 10 ms. In this condition, 7-state HMMs give the best performance whereas 12 states HMMs show the best scores in the case of 5 ms interval shown in subplot (c) and (d). Alternatively, the results with the observations framed at 1 ms in subplots (e) and (f) show the highest accuracy and noise robustness with 24-state monophone HMMs. The differences in the ASR accuracy among the different frame periods of the observation vectors at each SNR are very little as long as the best model configurations are employed in each case.

The results also show that the accuracy of ASR using monophone HMMs is lower than the case of whole-word HMMs shown in Figures 4.10 - 4.12 over the range of SNRs. For instance, the accuracy of the monophone HMMs is 12 pt. lower than the whole-word HMMs at SNR of 10 dB with white noise, and 8 pt. lower in babble noise. At SNR of -5 dB, the accuracy of the monophone HMMs is 15 pt. lower than the whole-word HMMs in white noise whereas 25 pt. lower in babble noise. This deterioration in ASR accuracy seems to be attributed to the fact that sub-word models have much more choices to select

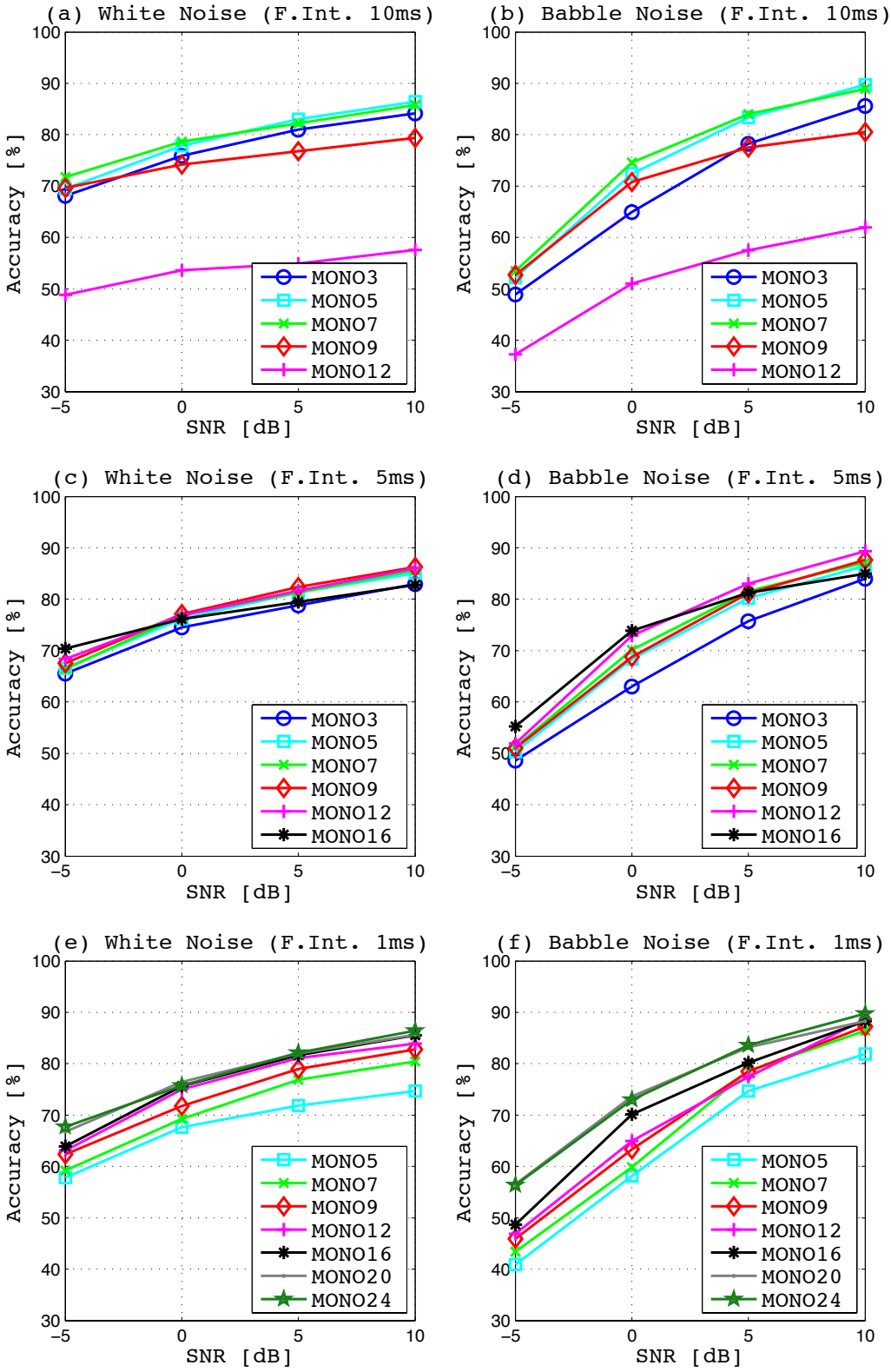

Figure 4.14: ASR accuracy with different monophone HMM configurations and frame intervals. a) & b) show the ASR accuracy in white noise and babble noise with the observation vectors framed at 10 ms interval while c) & d) are results with the frame interval at 5 ms, and e) & f) show the accuracy in white noise and babble noise with the observation vectors framed at 1 ms. The configuration of feature vectors is MFCC16-8.

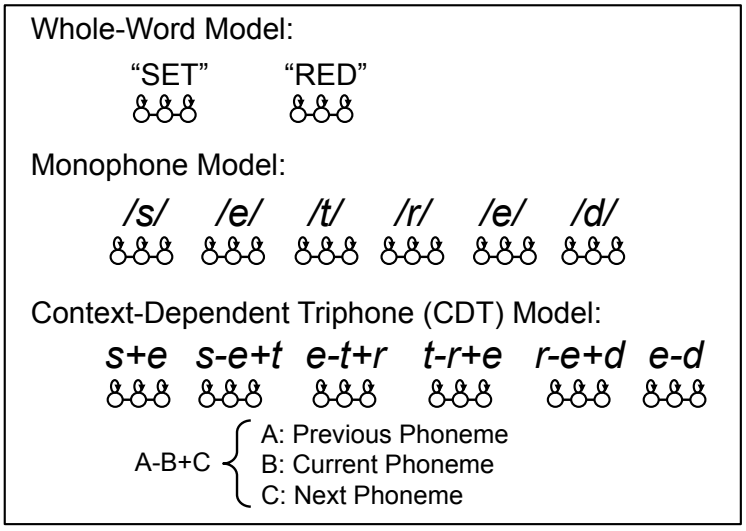

Figure 4.15: A structure of CD-triphone HMMs.

the most likely word than whole-word models. In whole-word model applications, each of the HMMs models a word itself, therefore, it can be more effective for word recognition than sub-word models. Alternatively, sub-word models can form any words by combining with other models and recognise even words which have not been trained in the training process as oppose to whole-word models which can deal with only the words trained in advance. This can be a good motivation to employ sub-word models in practical applications in spite of the lower decoding performance than whole-word models.

The monophone models comprising 35 phonemes may not have enough variation to represent speech having various prosodical characteristics especially for HMM-based speech synthesis discussed in Section 4.4. Therefore, context-dependent triphone HMMs (CD-triphone HMMs) are next discussed and examined as an alternative sub-word acoustic model to the monophone models.

4.3.4.4 Acoustic Model Settings for Context-Dependent Triphone HMMs

In the CD-triphone HMMs, a series of 3 phonemes corresponding to the previous, current and next phoneme forms a model regardless of word boundaries as illustrated in Figure 4.15. In this example, the CD-triphone model of /*e*/ in “set”, *s*-*e*+*t*, is different from /*e*/ in “red”, *r*-*e*+*d*, because of their different context, and this enables the models to represent speech with prosodical characteristics by including the contextual information

in each model.

In the case of GRID database used in the following tests, the transcripts of 800 utterances of each speaker for training make approximately 660 CD-triphone HMMs. These seem enough variation of the models as compared with 35 models in monophone HMMs. This fine division of the models, however, reduces the occurrences of each model during training, and consequently some of the models cannot be trained with enough samples and it causes overfitting of the models. In addition, cases where a CD-triphone in the test scripts is not included in the training set need to be considered. To tackle this problem, the following CD-triphone HMM ASR experiments employ tree-based model clustering [94]. This method constructs a decision tree, and different questions related with the characteristics of the CD-triphones are assigned at each node of the tree in order to cluster the models which have similar characteristics. Table 4.6 is an example of the questions and Figure 4.16 illustrates a decision tree structure. Each node of

C-Vowel	Is the current phoneme a vowel?
L-Vowel	Is the previous phoneme a vowel?
R-Vowel	Is the next phoneme a vowel?
C-Fricative	Is the current phoneme a fricative?
L-Fricative	Is the previous phoneme a fricative?
R-Fricative	Is the next phoneme a fricative?
C-/*a*/	Is the current phoneme /*a*/?
L-/*a*/	Is the previous phoneme /*a*/?
R-/*a*/	Is the next phoneme /*a*/?
⋮	⋮

Table 4.6: An Example of the questions at nodes of the decision tree.

the tree forms a cluster of the models and calculates the minimum description length (MDL) [94, 95] which decides whether the cluster is further divided by the next question or it stops the split. This method can reduce the number of models by clustering similar models and it enables each model to have enough training samples to form a better statistical model. Moreover, even if the test data includes an unknown CD-triphone, which did not appear in the training set, it can be led to the appropriate cluster through the tree and then HMMs are revised so that the unknown CD-triphone can be included in the HMMs.

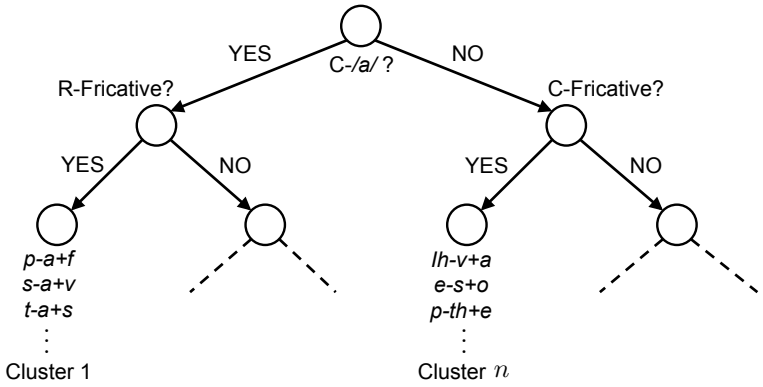

Figure 4.16: Tree-based model clustering

The ASR experiments with the CD-triphone HMMs in this section apply the clustering to the CD-triphone HMMs state-by-state and this reduces the number of the models to around 200.

The experiments examines the performances with various settings configured by different state numbers in HMMs as shown in Table 4.7, and feature vectors of speech are represented by the setting of MFCC16-8 framed at 10 ms, 5 ms and 1 ms as well as the tests with the monophone models and the whole-word models.

HMM Config.	Number of States	Feature Vector
TRI3	3	MFCC16-8
TRI5	5	
TRI7	7	
TRI9	9	
TRI12	12	
TRI16	16	
TRI20	20	
TRI24	24	

Table 4.7: Configurations for CD-triphone HMMs.

The test results with the CD-triphone HMMs are shown in Figure 4.17 in which the accuracy of ASR is calculated by Equation (4.63). The subplots in the first column are the test results in white noise while the second column is for the test results in babble noise. Subplots (a) and (b) show the ASR accuracy with the feature vectors framed at 10 ms. In this condition, HMMs comprising between 5 and 7 states give the best performance whereas HMMs with between 7 and 12 states show the best scores in the

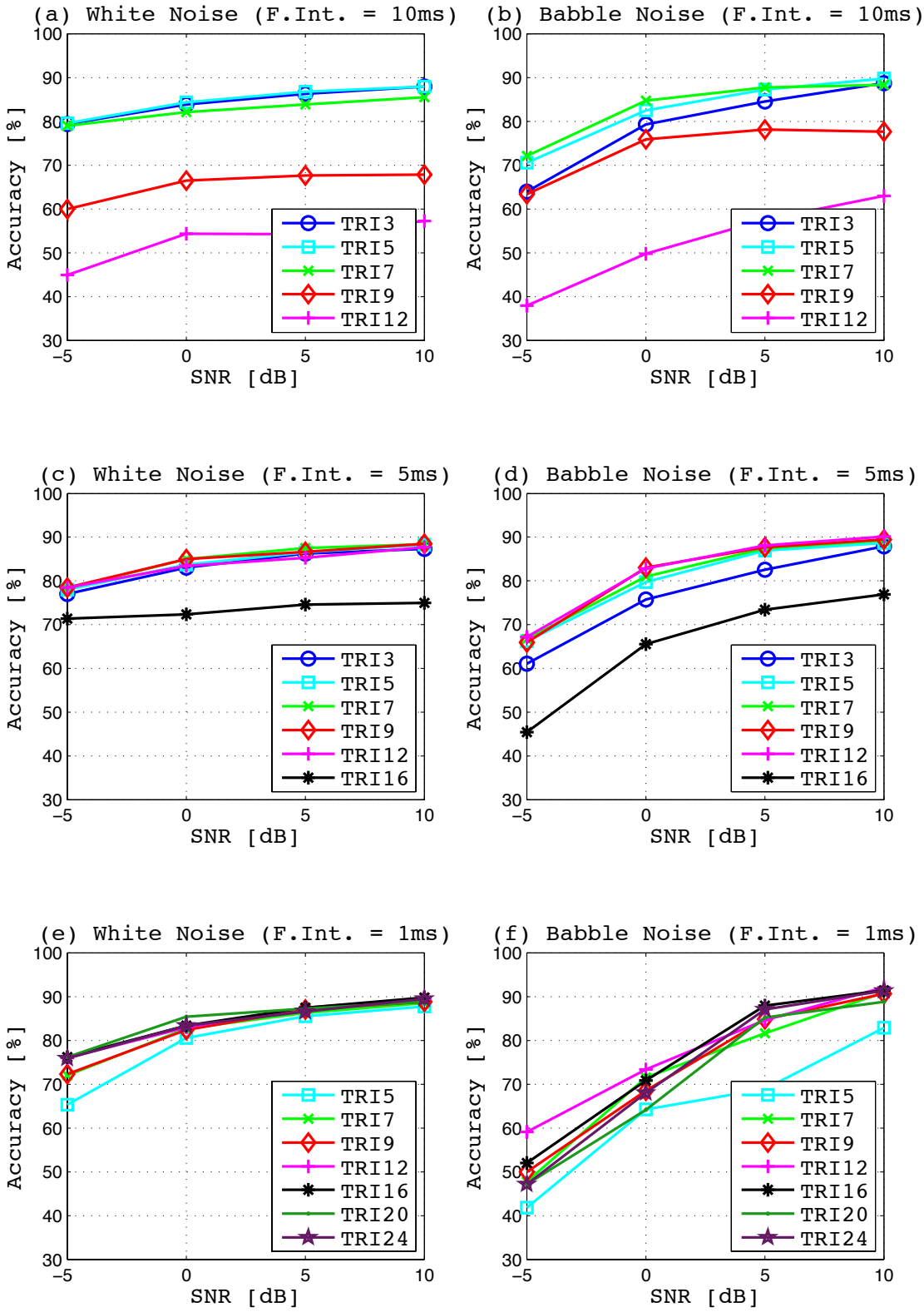

Figure 4.17: ASR accuracy with different CD-triphone HMM configurations and frame intervals. a) & b) show the ASR accuracy in white noise and babble noise with the observation vectors framed at 10 ms interval while c) & d) are results with the frame interval at 5 ms, and e) & f) show the accuracy in white noise and babble noise with the observation vectors framed at 1 ms. The configuration of feature vectors is MFCC16-8.

case of 5 ms interval shown in subplot (c) and (d). Alternatively, the results with the observations framed at 1 ms in subplots (e) and (f) show the HMMs with between 16 and 24 states generally perform with higher accuracy than other configurations but 12 state HMMs show the highest accuracy at SNRs of 0 dB and less than 0 dB in babble noise.

Taking the proper model configuration in each subplot, the differences in the ASR accuracy among the different frame periods of the observation vectors at each SNR are very little except for the 1 ms framed observation in babble noise, i.e. subplot (f), where the accuracy at SNRs of 0 dB and less than 0 dB is 12 to 15 pt. lower than other frame intervals.

The results also show that the CD-triphone HMMs raise the decoding accuracy as compared with monophone models in Figure 4.14. For example, the accuracy at SNR of -5 dB in babble noise improves by approximately 18 pts. in the case of 10 ms and 5 ms frame intervals while the improvement at 1 ms frame interval in babble noise is not significant. In the white noise, improvement of the accuracy is approximately 10 pts. over all noise conditions. The decoding accuracy of CD-triphone HMMs is, however, 5 pts. to 10 pts. lower than the whole-word HMMs with respect to the best configurations in each models.

Table 4.8 summarises the best configurations for each ASR conditions examined in the preceding experiments in terms of word recognition accuracy.

HMMs	Frame Interval	States	Observation
Whole-Word	10 ms	12 - 16	MFCC16-8
	5 ms	16 - 28	
	1 ms	24 - 40	
Monophone	10 ms	7	
	5 ms	12	
	1 ms	24	
CD-triphone HMM	10 ms	5 - 7	
	5 ms	7 - 12	
	1 ms	16 -24	

Table 4.8: A summary of the best configurations for each ASR experiments

4.3.4.5 Language Model

The previous experiments have employed the GRID grammar and dictionary as a language model in addition to the acoustic model brought by different configurations of HMMs. Applying a language model improves the ASR accuracy by constraining the choices of possible model sequences such that the resultant model sequence can match a local linguistic rules. The language model, however, also limits the input data because the input speech has to follow the local rules determined by the grammar and dictionary. The language model for the GRID corpus, for example, limits the number of recognisable words to only 52 including a model for silence by the dictionary and the grammar is far from practical English speech. Therefore, the use of this model cannot be applied to practical applications and loosening the constraint of the language model is essential to enable the system to be applied to practical use.

The following experiments eliminate the language model of the GRID grammar, i.e. the grammatical structure of *command* → *colour* → *preposition* → *letter* → *digit* → *adverb*, from the ASR system tested above in order to evaluate the influence of the language model by comparing the performances with the original results obtained with the language model. The settings of acoustic model and the feature vector for the tests are chosen as shown in Table 4.9 by referring to Table 4.8. The first column to the

Config	AM	LM	Observation	Interval	Practicality
WORD_G	WORD16	YES	MFCC16-8	5 ms	NO
WORD_N	WORD16	NO			NO
MONO_G	MONO12	YES			NO
MONO_N	MONO12	NO			YES
TRI_G	TRI12	YES			NO
TRI_N	TRI12	NO			YES

Table 4.9: Test configurations for the language model evaluation

last column of the table show names of configurations, settings of the acoustic model (AM), use of the language model (LM), observation vector settings, frame interval, and practicalities of the configurations. WORD_G, MONO_G and TRI_G exploit the GRID-specific language model as well as the preceding tests, therefore, these configurations cannot be applied to practical language applications. Although WORD_N gets rid of the

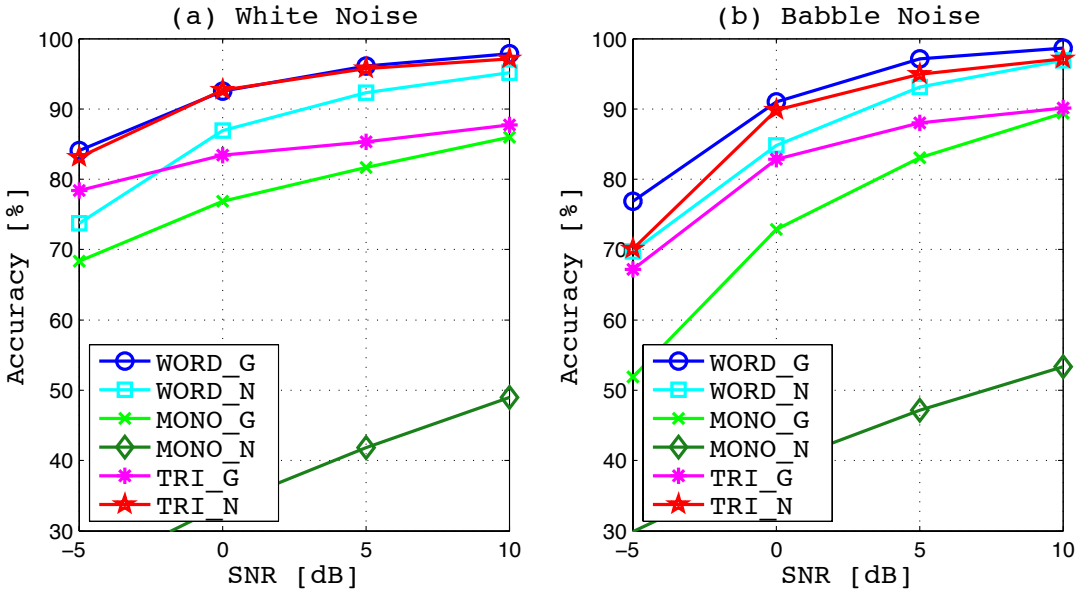

Figure 4.18: ASR accuracy with different model configurations with and without the language model. The feature vector is configured as MFCC16-8 framed at 5 ms interval.

language model, the whole-word HMMs for the acoustic model still limit word varieties to the vocabularies in GRID database. Alternatively, MONO_N is absolutely free from the constraint of the language model. TRI_N is constrained only by the previous and the next phoneme, therefore, this configuration is independent from the GRID-specific grammar and possible to be applied to practical applications. However, the small vocabulary of GRID database still constrains the triphone coverage.

The test results are shown in Figure 4.18 where the ASR accuracy, S_{acc}, is derived as

$$S_{acc} = \frac{S_N - (S_D + S_S + S_I)}{S_N} \times 100 \tag{4.64}$$

where S_D, S_S and S_I are the total number of word segments, i.e. phonemes for monophone and CD-triphone and words for whole-word models, making deletion errors, substitution errors and insertion errors respectively, and S_N denotes the total number of word segments in the reference transcripts. Triphone contexts are ignored at the error detection.

Subplot (a) shows the ASR accuracy in white noise whereas the test results in babble noise are shown in subplot (b). The accuracy of WORD_N is always lower than WORD_G.

For example, WORD_N is 10 pt. lower at SNR of -5 dB in white noise and 7 pt. lower at -5 dB in babble noise. This deterioration is attributed to the influence of the grammar. In the case of monophone HMMs, MONO_N always keeps low accuracy, which is lower than 55 % even at SNR of 10 dB, and does not reach the level of practical use over the range of SNRs. This seems to be due to too few variations of acoustic models to represent distinctive features of speech, and they cannot perform without referring to the language model. Interestingly, the performance of TRI_N is superior to TRI_G over the range of SNRs. It seems that the acoustic model of CD-triphone HMMs sufficiently represents the speech features and TRI_N can avoid extending a partial sub-word error to a whole-word error by not referring to the dictionary. This explains the reason why TRI_N obtains higher accuracy than TRI_G though it does not exploit the language model.

Alternatively, the smaller vocabulary of the GRID corpus could make the constraint of CD-triphone contexts stronger and it could give the good performance of TRI_N. However, the acoustic model constituted by CD-triphone HMMs seems to be effective solution to realise practical applications

4.3.4.6 Summary of the Experimental Results of ASR

The tests of ASR first examined ASR performance with different configurations of MFCC vectors using the 16 state noise-matched whole-word HMMs with the GRID grammar. 16-dimensional (16-D) MFCCs show the best word recognition accuracy, and higher dimensional MFCCs show the trend to have less accuracy. The results also show the effectiveness of the truncation of high order coefficients to improve the decoding accuracy. Therefore, selecting the observation features which do not have too much variability seems to be an important to achieve good performance in HMM decoding.

Next, different state configurations of the noise-matched whole-word HMMs have been examined with 8-D MFCCs (MFCC16-8) and the GRID grammar. The results show that the whole-word HMMs performs best with the number of the states being set between 12 and 16 when the frame interval of the observation is 5 ms and 10 ms. Alternatively, in the case of 1 ms-frame interval, the accuracy becomes relatively high when the number of the states is set between 24 and 40. However the noise robustness of the performance is reduced as compared with the frame intervals set 5 ms and 10 ms.

The ASR tests was then extended to the noise-matched monophone HMMs followed by the noise-matched CD-triphone HMMs with (MFCC16-8) and the GRID grammar. When the frame interval is set equal to 10 ms, 5 ms and 1 ms, Both of monophone HMMs and CD-triphone HMMs performs best with the number of the states set around 7, 12 and 24 respectively. Comparing the performance with the best configurations, the performance of the monophone HMMs is, however, always lower than whole-word HMMs, and it seems that the monophone HMMs which consit of 35 models of phonemes do not have enough variation to model natural speech while the whole-word HMMs consist of 52 models of words and the CD-triphone HMMs comprise around 200 models of phonemes. Alternatively, CD-triphone HMMs shows higher performance than monophone HMMs. Specifically, the noise robustness in babble noise is significantly improved as compared with monophone HMMs. Furthermore, the accuracy of CD-triphone HMMs at -5 dB in babble noise surpass the whole-word HMMs though the whole-word HMMs performs best of the three at the other noise conditions.

Finally, the effectiveness of the LM was examined by comparing the decoding accuracy of each HMM configuration with and without the GRID grammar. The performance of the whole-word HMMs and monophone HMMs falls when the GRID grammar is not applied, specifically, the monophone HMMs shows significant deterioration which is attributed to the fact that the monophone HMMs without the LM is completely free from the constraint and the performance depends on only the AM which does not have enough variation to represent natural speech. Conversely, CD-triphone HMMs performs better with no GRID grammar. It seems that the constraint from CD-triphone itself is relatively strong in this test condition because of the smaller vocabulary of the GRID corpus, and thus, the LM makes the constraint too strong and affects the decoding accuracy. For example, when a word “BIN” in an utterance failed to be recognised, TRI_N can possibly select “IN” instead of “BIN” but alternatives to “BIN” in TRI_G are limited within “LAY”, “PLACE” or “SET”.

As the overall evaluation, the 16 state noise-matched whole-word HMMs with the GRID grammar performs best with the observation configured as MFCC16-8 followed by the 12 state noise-matched CD-triphone HMMs with no language model. The difference of the performance between these two models is not significant, and thus, it is more

notable that CD-triphone HMMs performs without the LM, considering practical use.

4.4 HMM-Based Speech Synthesis

HMM synthesis is another key technique for HMM applications such as statistical parametric speech synthesis including text to speech (TTS) [50] and it also forms key process in the proposed HMM-based speech enhancement shown in Figure 2.7. This section explores the techniques related with the HMM-based speech synthesis with an example of a TTS application. A framework of the HMM-based speech synthesis for TTS is illustrated in Figure 4.19. The processes in the TTS application are divided into the offline

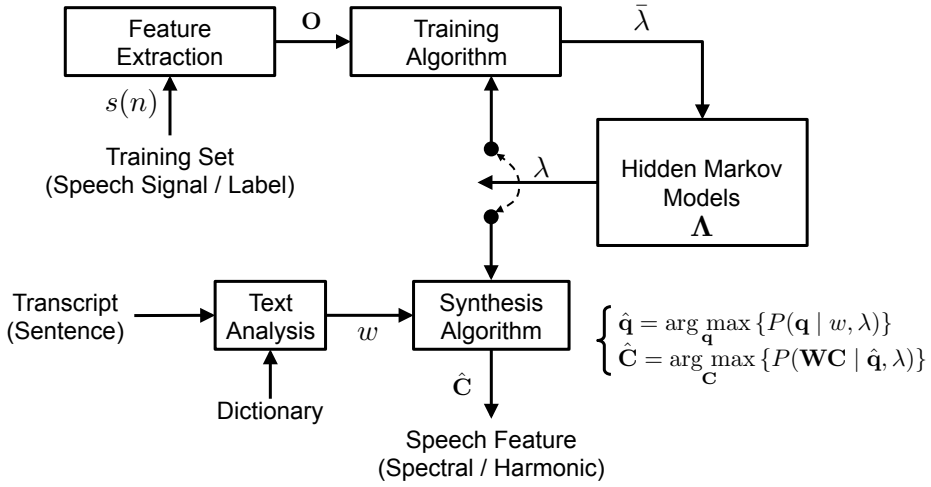

Figure 4.19: A framework of HMM-based speech synthesis for TTS.

training process and the online synthesis process as well as the ASR applications, which also consist of training process and decoding process.

4.4.1 HMM Training

HMM training in HMM-based speech synthesis is an offline process to optimise whole-word or sub-word model, λ, such that $P(\mathbf{O} \mid \lambda)$ is maximised by using the training data set of the speech and its transcript labels. The training procedure in this process is largely the same as the training process for ASR discussed in Section 4.3.2. However, the sequence of the feature vectors, $\mathbf{O}$, is different from ASR applications because the

observation sequence in HMM-based speech synthesis needs to contain all the speech features required by the speech production model, such as the fundamental frequency and voicing in addition to MFCCs, for the purpose of the application. Therefore, in the case where STRAIGHT is employed as the speech production model, a sequence of static observation vectors, $\mathbf{C}$, is composed from a sequence of MFCC vectors, $\mathbf{X}$, sequence of the aperiodicity vectors, $\mathbf{A}$, and the log of the fundamental frequency contour, $\mathbf{G}$ as

$$\mathbf{X} = \left[\mathbf{x}_0^T, \mathbf{x}_1^T, \ldots, \mathbf{x}_{N-1}^T\right]^T \tag{4.65}$$

$$\mathbf{A} = \left[\mathbf{a}_0^T, \mathbf{a}_1^T, \ldots, \mathbf{a}_{N-1}^T\right]^T \tag{4.66}$$

$$\mathbf{G} = \left[g_0, g_1, \ldots, g_{N-1}\right]^T \tag{4.67}$$

$$\mathbf{C} = \left[\mathbf{c}_0^T, \mathbf{c}_1^T, \ldots, \mathbf{c}_{N-1}^T\right]^T \tag{4.68}$$

$$\mathbf{c}_i = \left[\mathbf{x}_i^T, \mathbf{a}_i^T, g_i\right]^T \tag{4.69}$$

where $\mathbf{x}_i$, $\mathbf{a}_i$ and g_i represent the static MFCC vector, the static aperiodicity vector and log of fundamental frequency at frame i respectively. The aperiodicity vectors are formed from aperiodicity measure of speech, $A(f, i)$, mentioned in Section 3.3.3 by the same processes as MFCC where a Mel-filterbank is applied to $A(f, i)$, and then logarithm is taken prior to DCT but the coefficients should not be truncated. To obtain g_i, the fundamental frequency at frame i, f_{0i}, is first estimated by using the algorithms discussed in Section 3.5, such as PEFAC and then g_i is calculated as

$$g_i = \begin{cases} \log f_{0i} & \text{Voiced frames} \\ -10^{10} & \text{Unvoiced frames} \end{cases} \tag{4.70}$$

The log operation to f_{0i} reduces the dynamic range of f_{0i} at voiced frames while constant value, -10^{10}, is set instead of $\log f_{0i}$ at unvoiced frames so that unvoiced frames can be strongly isolated from voiced frames.

Moreover, a velocity derivative and an acceleration derivative of the feature vectors should be added into $\mathbf{C}$ in order to compose augmented observation sequence, $\mathbf{O}$, in order to avoid a piecewise state-dependent sequence of synthesised feature vectors that results

in poor speech quality [9] as follows.

$$\mathbf{O} = \left[\mathbf{o}_0^T, \mathbf{o}_1^T, \cdots, \mathbf{o}_{N-1}^T\right]^T \tag{4.71}$$

$$\mathbf{o}_i = \left[(\mathbf{o}_i^x)^T, (\mathbf{o}_i^a)^T, (\mathbf{o}_i^f)^T\right]^T \tag{4.72}$$

$$\mathbf{o}_i^x = \left[\mathbf{x}_i^T, \Delta\mathbf{x}_i^T, \Delta^2\mathbf{x}_i^T\right]^T \tag{4.73}$$

$$\mathbf{o}_i^a = \left[\mathbf{a}_i^T, \Delta\mathbf{a}_i^T, \Delta^2\mathbf{a}_i^T\right]^T \tag{4.74}$$

$$\mathbf{o}_i^f = \left[g_i, \Delta g_i, \Delta^2 g_i\right]^T \tag{4.75}$$

where Δ and Δ^2 denote the velocity derivative and the acceleration derivative respectively. These derivatives of the feature vectors are taken from augmented MFCC sequence, $\mathbf{O}_x$, augmented aperiodicity sequence, $\mathbf{O}_a$, and augmented log fundamental frequency contour, $\mathbf{O}_f$, which are derived as

$$\mathbf{O}_x = \mathbf{W}_x\mathbf{X} = \left[(\mathbf{o}_0^x)^T, (\mathbf{o}_1^x)^T, \ldots, (\mathbf{o}_{N-1}^x)^T\right]^T \tag{4.76}$$

$$\mathbf{O}_a = \mathbf{W}_a\mathbf{A} = \left[(\mathbf{o}_0^a)^T, (\mathbf{o}_1^a)^T, \ldots, (\mathbf{o}_{N-1}^a)^T\right]^T \tag{4.77}$$

$$\mathbf{O}_f = \mathbf{W}_f\mathbf{G} = \left[(\mathbf{o}_0^f)^T, (\mathbf{o}_1^f)^T, \ldots, (\mathbf{o}_{N-1}^f)^T\right]^T \tag{4.78}$$

where matrix, $\mathbf{W}_x$, $\mathbf{W}_a$ and $\mathbf{W}_f$ contain the regression coefficients to transform a sequence of the static vectors into the sequence of the augmented vectors. The following equation, for example, shows the transform from a sequence of static vectors, $\mathbf{U} = [\mathbf{u}_0^T, \mathbf{u}_1^T, \ldots, \mathbf{u}_{N-1}^T]^T$, to the sequence of the augmented vectors including the velocity derivative, $\mathbf{V} = [\mathbf{u}_0^T, \Delta\mathbf{u}_0^T, \mathbf{u}_1^T, \Delta\mathbf{u}_1^T, \ldots, \mathbf{u}_{N-1}^T, \Delta\mathbf{u}_{N-1}^T]^T$, by matrix, $\mathbf{W}_u$, but accel-

eration derivatives are omitted.

$$\underset{\mathbf{V}}{\begin{bmatrix} \vdots \\ \mathbf{u}_{i-1} \\ \Delta\mathbf{u}_{i-1} \\ \mathbf{u}_i \\ \Delta\mathbf{u}_i \\ \mathbf{u}_{i+1} \\ \Delta\mathbf{u}_{i+1} \\ \vdots \end{bmatrix}} = \underset{\mathbf{W}_u}{\begin{bmatrix} \cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ \cdots & 0 & I & 0 & 0 & 0 & \cdots \\ \cdots & -0.5I & 0 & 0.5I & 0 & 0 & \cdots \\ \cdots & 0 & 0 & I & 0 & 0 & \cdots \\ \cdots & 0 & -0.5I & 0 & 0.5I & 0 & \cdots \\ \cdots & 0 & 0 & 0 & I & 0 & \cdots \\ \cdots & 0 & 0 & -0.5I & 0 & 0.5I & \cdots \\ \cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \end{bmatrix}} \underset{\mathbf{U}}{\begin{bmatrix} \vdots \\ \mathbf{u}_{i-2} \\ \mathbf{u}_{i-1} \\ \mathbf{u}_i \\ \mathbf{u}_{i+1} \\ \vdots \end{bmatrix}} \tag{4.79}$$

Figure 4.20 illustrates the structure of augmented observation vector, $\mathbf{O}$.

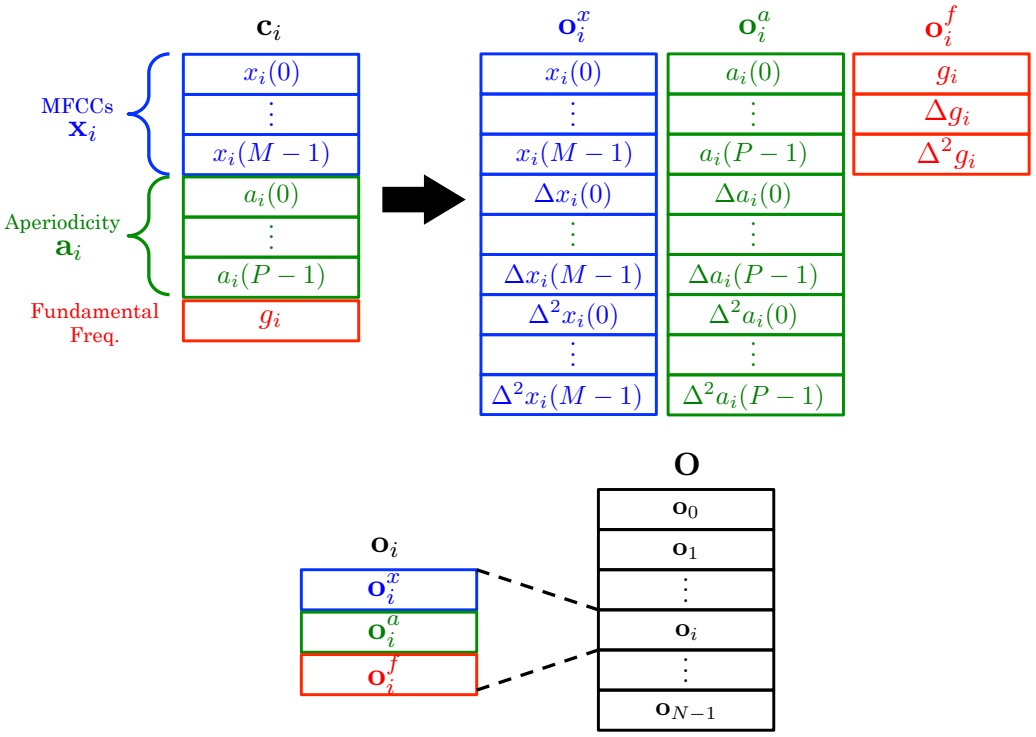

Figure 4.20: Structure of an augmented observation vector.

There exists another technique required in the training process. In TTS applications, the input of the system is a text-based transcript and thus, duration of the words, sub-words and states of the models for the output are unknown. To tackle this problem,

duration of each model and each state are also statistically modelled in the training process [96, 97] in which the duration models are formed as a Gaussian distribution of how many frames keep staying in each state per occurrence from the estimated state sequences of the training set obtained by the Viterbi algorithm which solves Equation (4.59).

4.4.2 Synthesis Process

In the synthesis process, the input transcript is first resolved into words and then sub-words by referring to the word dictionary in the system in order to convert the text input into sequence of HMMs, w. Then an estimate of static feature vector, $\hat{\mathbf{C}} = \left[\hat{\mathbf{c}}_0^T, \hat{\mathbf{c}}_1^T, \cdots, \hat{\mathbf{c}}_{N-1}^T\right]^T$, corresponding to w, is synthesised with the following steps.

$$\begin{aligned}
\hat{\mathbf{O}} &= \arg\max_{\mathbf{O}} \{P(\mathbf{O} \mid w, \lambda)\} && (4.80)\\
&= \arg\max_{\mathbf{O}} \left\{ \sum_{\text{all } \mathbf{q} \in \mathbf{Q}} P(\mathbf{O}, \mathbf{q} \mid w, \lambda) \right\} && (4.81)\\
&\approx \arg\max_{\mathbf{O}} \left[\max_{\mathbf{q} \in \mathbf{Q}} \{P(\mathbf{O}, \mathbf{q} \mid w, \lambda)\} \right] && (4.82)\\
&= \arg\max_{\mathbf{O}} \left[\max_{\mathbf{q} \in \mathbf{Q}} \{P(\mathbf{q} \mid w, \lambda) \cdot P(\mathbf{O} \mid \mathbf{q}, \lambda)\} \right] && (4.83)
\end{aligned}$$

where $\hat{\mathbf{O}}$ is an estimate of the augmented vector including static and dynamic features that corresponds to w while $\mathbf{q}$ and $\mathbf{Q}$ represent a state sequence, $[q_0, q_1, \cdots, q_{N-1}]$, and a group of the all possible state sequences respectively. Equation (4.83) is approximated as

$$\hat{\mathbf{O}} \approx \arg\max_{\mathbf{O}} \{P(\mathbf{O} \mid \hat{\mathbf{q}}, \lambda)\} \tag{4.84}$$

where $\hat{\mathbf{q}}$ is the most likely state sequence defined as

$$\hat{\mathbf{q}} = \arg\max_{\mathbf{q}} \{P(\mathbf{q} \mid w, \lambda)\} \tag{4.85}$$

In TTS applications, this most likely state sequence is derived by splitting each model in w into its states according to the trained duration models mentioned in Section 4.4.1.

Probability density of $\mathbf{O}$ in each state in λ are defined as Gaussian distributions,

thus, Equation (4.85) derives

$$\hat{\mathbf{O}} = \arg\max_{\mathbf{O}} \left\{ \mathcal{N}(\mathbf{O}; \boldsymbol{\mu}_{\hat{\mathbf{q}}}, \boldsymbol{\Sigma}_{\hat{\mathbf{q}}}) \right\} \tag{4.86}$$

where

$$\boldsymbol{\mu}_{\hat{\mathbf{q}}} = \left[\boldsymbol{\mu}_{\hat{q}_0}^T, \boldsymbol{\mu}_{\hat{q}_1}^T, \cdots, \boldsymbol{\mu}_{\hat{q}_{N-1}}^T \right]^T \tag{4.87}$$

$$\boldsymbol{\Sigma}_{\hat{\mathbf{q}}} = \mathrm{diag} \left[\boldsymbol{\sigma}_{\hat{q}_0}^T, \boldsymbol{\sigma}_{\hat{q}_1}^T, \cdots, \boldsymbol{\sigma}_{\hat{q}_{N-1}}^T \right]^T \tag{4.88}$$

where $\boldsymbol{\mu}_{\hat{q}_i}$ represents the mean vector of the Gaussian distribution in state $\hat{q}_i$ while $\boldsymbol{\sigma}_{\hat{q}_i}$ is the diagonal vector of the covariance matrix in state $\hat{q}_i$. Equations (4.76 - 4.78) shows that $\mathbf{O}$ is a linear transform of $\mathbf{C}$, therefore, the following equation is derived from Equation (4.86).

$$\hat{\mathbf{C}} = \left[\hat{\mathbf{c}}_0^T, \hat{\mathbf{c}}_1^T, \ldots, \hat{\mathbf{c}}_{N-1}^T \right]^T \tag{4.89}$$

$$= \arg\max_{\mathbf{C}} \left\{ \mathcal{N}(\mathbf{WC}; \boldsymbol{\mu}_{\hat{\mathbf{q}}}, \boldsymbol{\Sigma}_{\hat{\mathbf{q}}}) \right\} \tag{4.90}$$

where $\hat{\mathbf{c}}_i$ is the synthesised static feature vector at i-th frame. The derivative of log-normal distribution with respect to $\mathbf{C}$ is set equal to zero in order to synthesise $\hat{\mathbf{C}}$ as

$$\frac{\partial \log \mathcal{N}\left(\mathbf{WC}; \boldsymbol{\mu}_{\hat{\mathbf{q}}}, \boldsymbol{\Sigma}_{\hat{\mathbf{q}}}\right)}{\partial \mathbf{C}} = 0 \tag{4.91}$$

This derives the following relationship and $\hat{\mathbf{C}}$ is finally obtained.

$$\mathbf{W}^T \boldsymbol{\Sigma}_{\hat{\mathbf{q}}}^{-1} \mathbf{W} \hat{\mathbf{C}} = \mathbf{W}^T \boldsymbol{\Sigma}_{\hat{\mathbf{q}}}^{-1} \boldsymbol{\mu}_{\hat{\mathbf{q}}} \tag{4.92}$$

In the case of using the STRAIGHT vocoder as the speech production model, synthesised vector, $\hat{\mathbf{c}}_i$, comprises MFCC vector, $\hat{\mathbf{x}}_i$, aperiodicity vector $\hat{\mathbf{a}}_i$ and log fundamental frequency $\hat{g}_i$.

$$\hat{\mathbf{c}}_i = \left[\hat{\mathbf{x}}_i^T, \hat{\mathbf{a}}_i^T, \hat{g}_i \right]^T \tag{4.93}$$

$$= \left[\hat{x}_i(0), \hat{x}_i(1), \ldots, \hat{x}_i(M-1), \hat{a}_i(0), \hat{a}_i(1), \ldots, \hat{a}_i(P-1), \hat{g}_i \right]^T \tag{4.94}$$

where $\hat{x}_i(m)$ represents the synthesised m-th MFCC while $\hat{a}_i(p)$ is p-th mel-filterbank-cepstral coefficient of aperiodicity measure at i-th frame. In the case where the MFCCs are truncated to the first M' coefficients in original observation, $\mathbf{O}$, $(M - M')$ zeros are padded to the tail of $\hat{\mathbf{x}}_i$ as

$$\hat{\mathbf{x}}_i = \left[\hat{x}_i(0), \hat{x}_i(1), \ldots, \hat{x}_i(M'-1), 0, 0, \ldots\right]^T \tag{4.95}$$

The fundamental frequency contour is derived by taking exponent of $\hat{g}_i$ for $i = 0, 1, \ldots, N-1$.

$$\hat{f}_{0i} = \begin{cases} \exp\left(\hat{g}_i\right), & \text{as } \hat{g}_i \neq -10^{10} \\ 0, & \text{as } \hat{g}_i = -10^{10} \end{cases} \tag{4.96}$$

and inverse DCT is applied to $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{a}}_i$ to derive the log-Mel-filterbank coefficient vector of the spectral power, $\hat{\mathbf{x}}_i^l$, and the log-Mel-filterbank coefficient vector of the aperiodicity measure, $\hat{\mathbf{a}}_i^l$.

$$\hat{\mathbf{x}}_i^l = \left[\hat{x}_i^l(0), \hat{x}_i^l(1), \ldots, \hat{x}_i^l(M-1)\right]^T \tag{4.97}$$

$$\hat{\mathbf{a}}_i^l = \left[\hat{a}_i^l(0), \hat{a}_i^l(1), \ldots, \hat{a}_i^l(P-1)\right]^T \tag{4.98}$$

where

$$\hat{x}_i^l(j) = \sum_{k=1}^{M-1} \sqrt{\frac{2}{M}} \hat{x}_i(k) \cos\left(\frac{(2j+1)k\pi}{2M}\right) \tag{4.99}$$

$$\hat{a}_i^l(j) = \sum_{k=1}^{P-1} \sqrt{\frac{2}{P}} \hat{a}_i(k) \cos\left(\frac{(2j+1)k\pi}{2P}\right) \tag{4.100}$$

$\hat{\mathbf{x}}_i^l$ and $\hat{\mathbf{a}}_i^l$ are then transformed to the linear-Mel-filterbank domain as

$$\hat{\mathbf{x}}_i^{fb} = \left[\hat{x}_i^{fb}(0), \hat{x}_i^{fb}(1), \ldots, \hat{x}_i^{fb}(M-1)\right]^T \tag{4.101}$$

$$\hat{\mathbf{a}}_i^{fb} = \left[\hat{a}_i^{fb}(0), \hat{a}_i^{fb}(1), \ldots, \hat{a}_i^{fb}(P-1)\right]^T \tag{4.102}$$

where

$$\hat{x}_i^{fb}(j) = \exp\left(\hat{x}_i^{l}(j)\right) \quad (4.103)$$

$$\hat{a}_i^{fb}(j) = \exp\left(\hat{a}_i^{l}(j)\right) \quad (4.104)$$

The bandwidth of each Mel-filterbank channel is equalised in the Mel-frequency domain, therefore, $\hat{\mathbf{x}}_i^{fb}$ and $\hat{\mathbf{a}}_i^{fb}$ need to be normalised as follows.

$$\bar{\mathbf{x}}_i = [\bar{x}_i(0), \bar{x}_i(1), \ldots, \bar{x}_i(M-1)]^T \quad (4.105)$$

$$\bar{\mathbf{a}}_i = [\bar{a}_i(0), \bar{a}_i(1), \ldots, \bar{a}_i(P-1)]^T \quad (4.106)$$

where

$$\bar{x}_i(j) = \frac{2\hat{x}_i^{fb}(j)}{B(j)} \quad (4.107)$$

$$\bar{a}_i(j) = \frac{2\hat{a}_i^{fb}(j)}{B(j)} \quad (4.108)$$

where $B(j)$ is the bandwidth of the band-pass filter in j-th Mel-filterbank channel. Finally, cubic spline interpolation [98] is applied to $\bar{\mathbf{x}}_i$ and $\bar{\mathbf{a}}_i$ in order to obtain the spectral envelope, $\hat{X}_i(f)$, and the aperiodicity measure, $\hat{A}_i(f)$, where $f = 0, 1, \ldots, F-1$ and F denotes the number of the frequency bins. $\hat{X}_i(f)$ and $\hat{A}_i(f)$ at $i = 0, 1, \ldots, N-1$, are aligned to the time-frequency domain to derive $\hat{X}(f,i)$ and $\hat{A}(f,i)$, and all the parameters required by the STRAIGHT vocoder, i.e. $\hat{X}(f,i)$, $\hat{A}(f,i)$ and $\hat{f}_{0i}$, are now synthesised from the HMMs.

4.4.3 Experimental Evaluation on HMM-Based Speech Synthesis

Understanding the performance of HMM-based speech synthesis and proper settings to improve the quality of synthesised speech is important to apply its techniques to HMM-based speech enhancement. This motivates to evaluate the performances of HMM-based speech synthesis with different model and observation vector settings. For that purpose, this section conducts experiments in which model and state sequences of the synthesised speech are obtained from natural speech by HMM decoding with forced alignment rather

than having a text input as a TTS application.

The experiments use clean speech from four speakers in the GRID database [43], two males and two females, which is down sampled to 8 kHz as well as the ASR experiments in Section 4.3.4. From the 1000 utterances from each speaker, 800 are used for training and the remainder are for testing to derive the model and state sequences by forced alignment.

Trained HMMs synthesise the spectral envelope and aperiodicity in the time-frequency domain and the fundamental frequency contour as discussed above, and then the STRAIGHT vocoder converts them to the time-domain speech. Different configurations of HMMs including whole-word, monophone and CD-triphone HMMs (single Gaussian) are evaluated.

4.4.3.1 Feature Vectors

The feature vector is formed as a combination of the MFCC coefficients, the log-Mel-filterbank aperiodicity coefficients and the fundamental frequency with the velocity and acceleration derivatives as shown in Figure 4.20. Different configurations of the MFCC are examined while the number of the aperiodicity coefficients is fixed as 40, and Table 4.10 shows the feature vector configurations examined with the following experiment.

Config	Mel-FBank	MFCC Coeffs	Aperiodicity Coefs	Derivatives
MFCC16-8	16	8	40	Δ and Δ^2
MFCC16-16		16		
MFCC23-8	23	8		
MFCC23-23		23		

Table 4.10: Configurations of the feature vectors.

MFCC16-8 and MFCC16-16 are based on 16 coefficient MFCCs while MFCC23-8 and MFCC23-23 are based on 23 coefficient MFCCs, but MFCC16-8 and MFCC23-8 contain only the first 8th coefficients and other coefficients are truncated. These configurations are chosen because the ASR tests in Section 4.3.4.1 show these configurations represent speech features better than other settings.

The first experiment uses whole-word HMMs which consist of 16 states, and they are trained with each configuration of feature vectors which are framed at 5 ms interval.

The HMMs then decode clean test speech to obtain their word and state sequences by which the HMMs synthesise the HMM-based speech features, i.e. the spectral envelope, the aperiodicity and the fundamental frequency. The synthesised fundamental frequency contour is, however, not used for speech reconstruction. Instead, PEFAC estimates the fundamental frequency directly from the original speech. Reconstructed speech synthesised with different configurations of feature vectors is then evaluated in terms of PESQ, and the configuration which shows the best PESQ scores is used for the following experiments.

PESQ is an objective measure recommended by ITU-T (2000) for speech quality assessment of narrow-band handset telephony and narrow-band speech codecs [99]. This measure assesses distortions in speech including packet loss, signal delays and codec distortions by comparing a degraded signal with the original reference signal. The reference and degraded signals are first equalised to a standard listening level and then filtered by a filter having a response similar to a telephone handset. The filtered signals are then aligned in time to correct time delays followed by an auditory transform to obtain the loudness spectra. Finally, the difference of the loudness spectra between the reference and degraded signals is averaged over time and frequency to produce the objective score which correlates with subjective MOS listening tests.

The results of the first experiment are shown in Table 4.11. The feature vectors where

	MFCC16-8	MFCC16-16	MFCC23-8	MFCC23-23
PESQ	2.48	2.59	2.48	2.63

Table 4.11: Average PESQ scores of the synthesised speech with 16-state whole-word HMMs and different feature vector configurations framed at 5 ms interval.

the MFCC coefficients are not truncated show higher PESQ score while the truncation of MFCC coefficients are effective in ASR decoding. Since the proposed HMM-based speech enhancement utilises both techniques of HMM decoding and synthesis, the feature vectors need to be optimised so that the performance of HMMs can be balanced between decoding and synthesis. Considering this result with respect to Figure 4.9, MFCC23-23 is employed as the appropriate feature vector to the following experiments which evaluate the performance of speech synthesis in further detail.

4.4.3.2 Whole-Word Model

The next experiment examines whole-word HMMs with speech feature vectors framed at 10 ms, 5 ms and 1 ms intervals. The optimum HMM settings for each frame interval have been evaluated in the ASR experiment and summarised in Table 4.8, therefore, the test configurations here are set as Table 4.12.

Frame Interval	Number of States	Observation
10 ms	12	
5 ms	16	MFCC23-23
1 ms	40	

Table 4.12: Configurations of the whole-word HMMs for different frame interval.

Figure 4.21 shows narrowband spectrograms of synthesised speech. Subplot(a) is the spectrogram of natural speech of "Bin Blue At E Seven Now." spoken by a male speaker. Subplot (b), (c) and (d) show the spectrograms of the HMM-based speech with the feature vectors framed at 10 ms, 5 ms and 1 ms respectively. These results show that the whole-word HMMs synthesise speech in fair quality but the synthesised speech with 10 ms frame interval is over-smoothed as compared with others. Figure 4.22 illustrates the fundamental frequency contour of the same speech synthesised by the whole-word HMMs. HMM-based fundamental frequency contour cannot trace rapid changes in the natural speech and it causes voicing errors and different intonation. Therefore, the fundamental frequency contour required by the speech production model should be estimated from the original speech by using f_0 estimation techniques, discussed in Section 3.5, rather than using synthesised parameters in the case of HMM-based speech enhancement.

4.4.3.3 Monophone Model

HMM-based speech synthesis with monophone HMMs are next examined with different frame intervals. Monophone HMM state settings for each frame interval are configured as Table 4.13, according to Table 4.8. Figure 4.23 shows narrowband spectrograms of synthesised speech. Subplot(a) is the spectrogram of natural speech of "Bin Blue At E Seven Now." spoken by a male speaker. Subplot (b), (c) and (d) show the spectrograms of the HMM-based speech with the feature vectors framed at 10 ms, 5 ms and 1 ms respec-

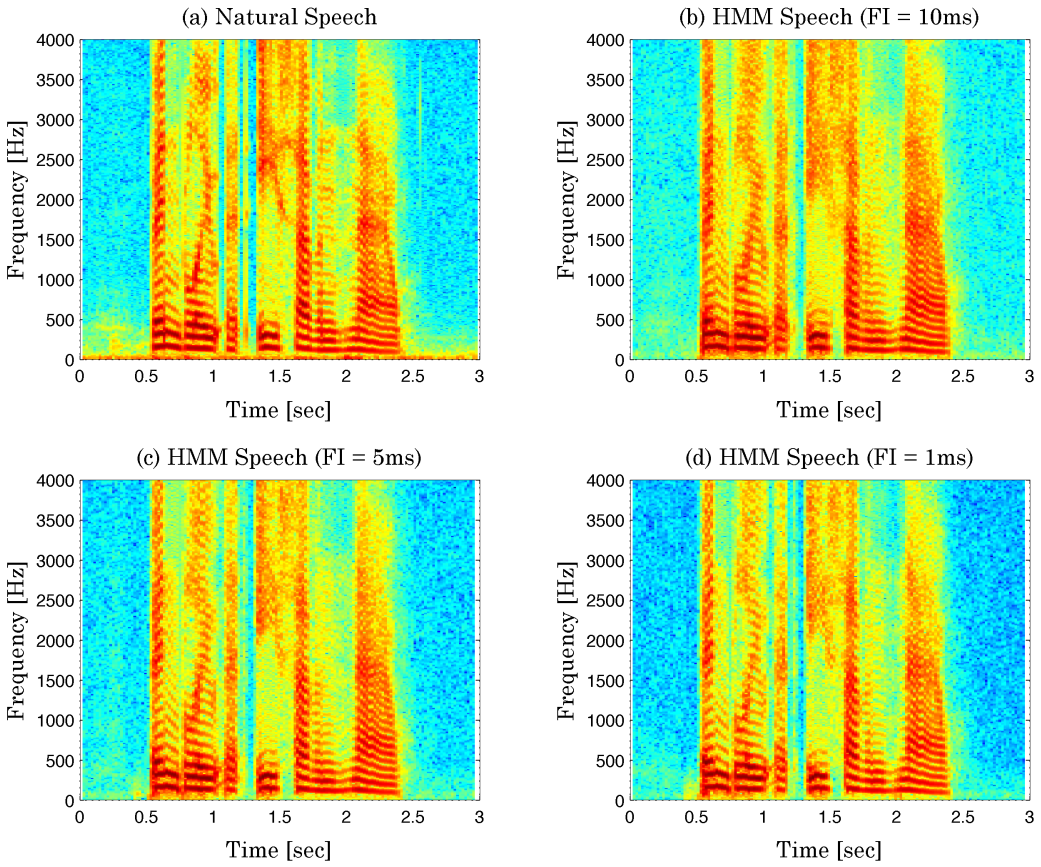

Figure 4.21: Narrowband spectrograms of a) the original natural speech of "Bin Blue At E Seven Now" spoken by a male speaker, b) HMM-based speech synthesised by 12-state whole-word HMMs with feature vector, MFCC23-23, framed at 10 ms interval, c) HMM-based speech synthesised by 16-state whole-word HMMs with MFCC23-23 framed at 5 ms interval and d) HMM-based speech synthesised by 40-state whole-word HMMs with MFCC23-23 framed at 1 ms interval.

Frame Interval	Number of States	Observation
10 ms	7	
5 ms	12	MFCC23-23
1 ms	24	

Table 4.13: Configurations of the monophone HMMs for different frame interval.

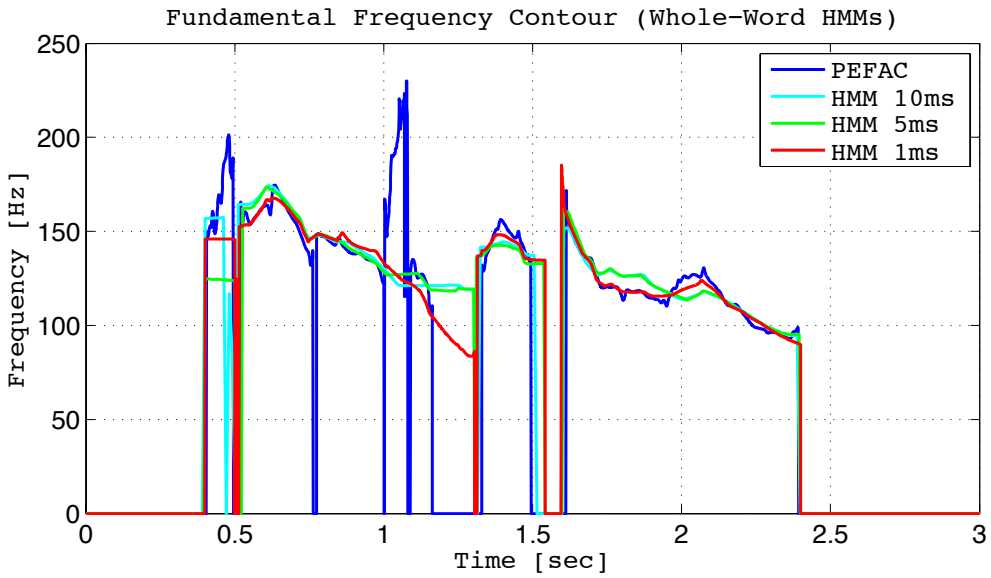

Figure 4.22: Fundamental frequency contours synthesised by different configurations of whole-word HMMs.

tively. These results show that the quality of speech synthesised with monophone HMMs is totally inferior to the whole-word HMM-based speech because of over-smoothing. This is attributed to the lack of model variation and the use of CD-triphone HMMs are motivated.

4.4.3.4 Context Dependent Triphone HMMs

HMM-based speech synthesis with CD-triphone HMMs are next examined with different frame intervals. In the training process, the tree-based clustering is also applied to the CD-triphone HMMs as well as the ASR experiments in Section 4.3.4.4. The state settings of CD-triphone HMMs for each frame interval are configured as Table 4.14, referring to Table 4.8.

Frame Interval	Number of States	Observation
10 ms	7	
5 ms	12	MFCC23-23
1 ms	24	

Table 4.14: Configurations of the CD-triphone HMMs for different frame interval.

Figure 4.24 shows narrowband spectrograms of synthesised speech. Subplot(a) is

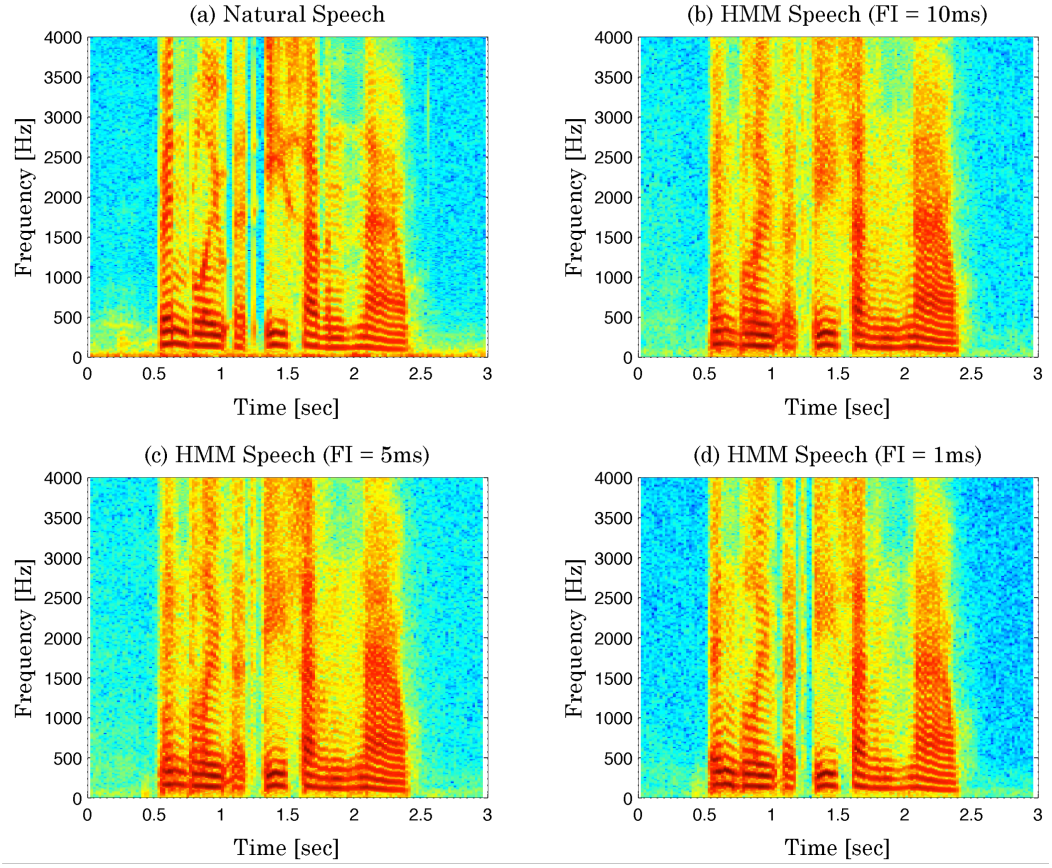

Figure 4.23: Narrowband spectrograms of a) the original natural speech of "Bin Blue At E Seven Now" spoken by a male speaker, b) HMM-based speech synthesised by 7-state monophone HMMs with feature vector, MFCC23-23, framed at 10 ms interval, c) HMM-based speech synthesised by 12-state monophone HMMs with MFCC23-23 framed at 5 ms interval and d) HMM-based speech synthesised by 24-state monophone HMMs with MFCC23-23 framed at 1 ms interval.

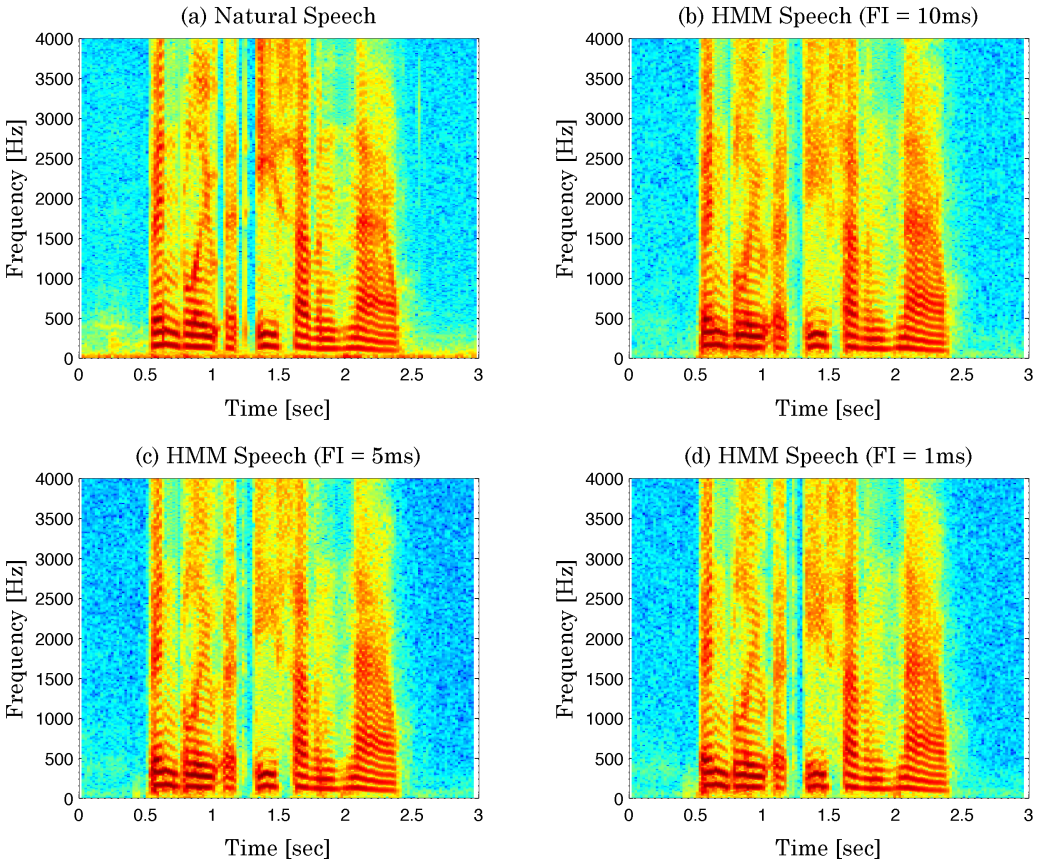

Figure 4.24: Narrowband spectrograms of a) the original natural speech of "Bin Blue At E Seven Now" spoken by a male speaker, b) HMM-based speech synthesised by 7-state-CD-triphone HMMs with feature vector, MFCC23-23 framed at 10 ms interval, c) HMM-based speech synthesised by 12-state-CD-triphone HMMs with MFCC23-23 framed at 5 ms interval and d) HMM-based speech synthesised by 24-state-CD-triphone HMMs with MFCC23-23 framed at 1 ms interval.

the spectrogram of natural speech of "Bin Blue At E Seven Now." spoken by a male speaker. Subplot (b), (c) and (d) show the spectrograms of the HMM-based speech with the feature vectors framed at 10 ms, 5 ms and 1 ms respectively. These results show that the CD-triphone HMMs synthesise the speech in as good quality as the whole-word HMM-based speech synthesis, and provides fairly natural speech especially in the case of short time frame shift.

Figure 4.25 illustrates the fundamental frequency contour of the same speech synthesised by the CD-triphone HMMs. This result shows that the CD-triphone HMMs

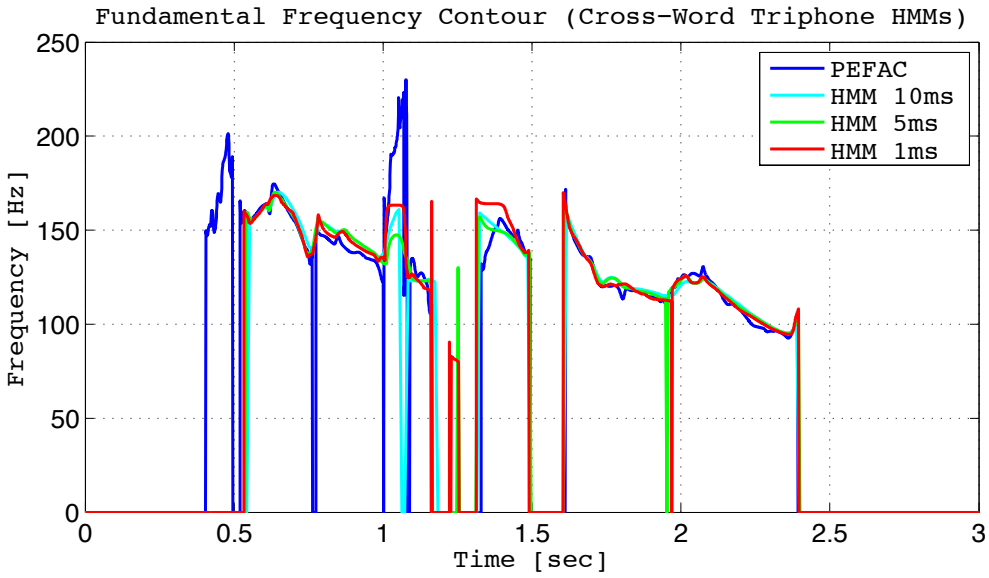

Figure 4.25: Fundamental frequency contours synthesised by different configurations of CD-triphone HMMs.

synthesise the fundamental frequency contour which can trace rapid change as compared with the whole-word HMMs because the duration of each model is shorter than the whole-word HMMs. However the accuracy of the contour is still not sufficient and it encourages estimating the fundamental frequency contour directly from the original speech rather than using synthesised f_0 contour.

The average PESQ scores of the synthesised speech in different configurations are summarised in Table 4.15 showing whole-word HMMs and CD-triphone HMMs synthesise speech with the same PESQ scores which are superior to monophone HMMs.

The resultant PESQ scores set the upper limit to the proposed method of HMM-

HMM	Frame Int. 10 ms	Frame Int. 5 ms	Frame Int. 1 ms
Whole-Word	2.55	2.63	2.73
Monophone	2.18	2.25	2.42
CD-triphone	2.55	2.63	2.73

Table 4.15: PESQ scores of synthesised speech in different model configurations. The feature vector configuration is MFCC23-23.

based speech enhancement because the model and state sequences in these experiments are obtained in error-free condition by forced alignment. The results shown in Table 4.15 use 23-D MFCC vector to give the best PESQ scores to synthesised speech whereas the results illustrated in Figure 4.18 use 8-D MFCC vector to give priority to the noise robustness. Both of these characteristics are important for HMM-based speech enhancement, therefore, the best balanced setting needs to be explored.

4.4.4 Summary of the Experimental Results of HMM-Based Speech Synthesis

The experiments synthesised speech parameters with the clean whole-word HMMs, monophone HMMs and CD-triphone HMMs and then STRAIGHT reconstructed speech from the synthesised parameters. The model and state sequence to synthesise speech was obtained by forced alignment with clean natural speech and its transcript.

The experimental results first found that an acoustic configuration containing higher dimension MFCCs performs better than the other, comparing the output speech synthesised from the whole-word HMMs with the acoustic configurations of MFCC16-16 and MFCC23-23 in terms of PESQ. Moreover, it was also found that the acoustic configurations containing MFCCs without truncation performs better than the configurations using the truncated MFCCs. This is opposite tendency to the decoding process, therefore, it is important to find a balance when the acoustic model is configured in HMM-based speech enhancement.

Then it was found that the whole-word HMMs and CD-triphone HMMs obtain the same PESQ score while the monophone HMMs performs lower than them. This result supports the notion that the monophone HMMs do not have enough variation to represent natural speech.

The experiments also showed that the fundamental frequency contour synthesised from the HMMs is not as accurate as the fundamental frequency contour estimated by PEFAC. Therefore, the proposed method of HMM-based speech enhancement should estimate the fundamental frequency contour of clean speech from noisy speech by PEFAC rather than synthesising from HMMs.

4.5 HMM-Based Speech Enhancement

In the preceding sections, the key techniques for HMM applications, i.e. HMM training, HMM decoding and HMM synthesis, are individually discussed and explored with the practical examples of ASR and TTS. This section combines these techniques in order to constitute HMM-based speech enhancement and evaluates its performance in terms of PESQ and NCM, which represent objective measures of speech quality and intelligibility.

Speech from two female speakers and two male speakers in the GRID database is used for the evaluation as well as other tests in the preceding sections. 1000 utterances from each speaker are down sampled to 8 kHz and 800 of them are used for training and the remainder are used for tests of speech enhancement in white noise and babble noise at SNRs from -5 dB to 10 dB.

The empirical tests in this section follow earlier works in [5,6] and the framework of speech enhancement is illustrated in Figure 4.26.

4.5.1 Feature Extraction

In the proposed method of HMM-based speech enhancement, a configuration of feature vectors cannot be changed between the decoding process and synthesis process because it is important to share the same HMMs between those processes. Considering the balance of the performance between decoding and synthesis, 8-D MFCC vectors produced by truncating 16-D log-Mel-filterbank coefficient vectors and 23-D MFCC vectors with no truncation are employed as the spectral component in the feature vectors by referring to the empirical knowledge of both Figure 4.9 and Table 4.11. The former configuration gives priority to the noise robustness at decoding while the latter sets priority to raising the upper limit of the quality of enhanced speech. In addition to the MFCC vectors,

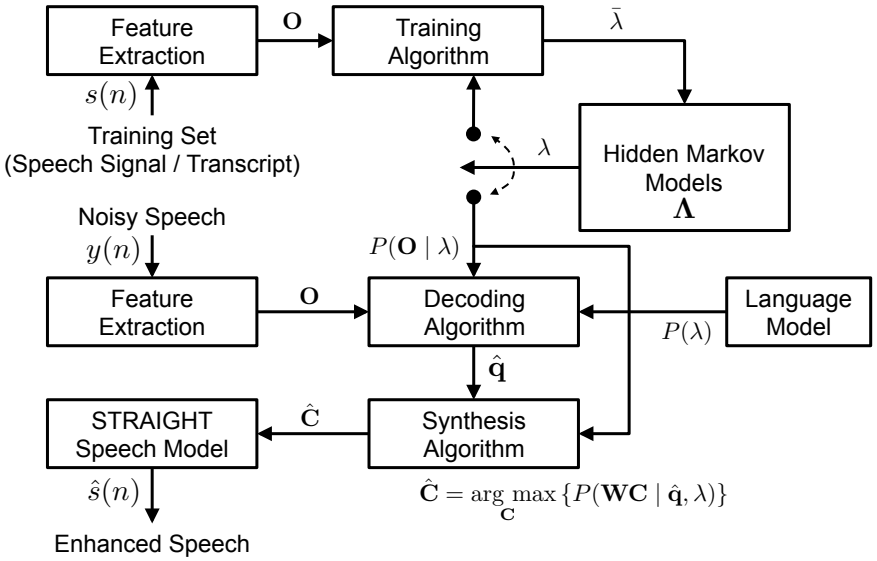

Figure 4.26: The framework of HMM-based speech enhancement.

the aperiodicity measure and fundamental frequency contour need to be included in the feature vectors in order to utilise the STRAIGHT speech model. Table 4.16 shows the specification of these components in the feature vectors to be examined.

Configuration	Component	DFT	Mel-FBK	MFCC Coefs	Derivatives
MFCC16-8	MFCC	1024	16	8	Δ, Δ^2
	Aperiodicity	1024	40	40	
	$\log f_0$	-	-	-	
MFCC23-23	MFCC	1024	23	23	Δ, Δ^2
	Aperiodicity	1024	40	40	
	$\log f_0$	-	-	-	

Table 4.16: The configuration of the feature vectors for the test.

Discrete-time speech is first divided into 25 ms-length frames by applying a Hamming window in which the frame shift is set equal to 1 ms and 5 ms. A 1024-point STFT is then applied to obtain the power spectrum and aperiodicity measure which are then input into a Mel-filterbank. The number of Mel-filterbank channels is set equal to 16 and 23 for the power spectrum and 40 for the aperiodicity measure. A logarithm is then applied to the filterbank energies followed by a discrete cosine transform to produce 8-D and 23-D MFCC vector, $\mathbf{x}_i$, and 40-D aperiodicity vector, $\mathbf{a}_i$. The fundamental frequency contour, f_{0i}, is estimated with PEFAC followed by logarithm applied to produce

$\log f_{0i}$. These three components are assigned into augmented feature vector $\mathbf{O}$ with their velocity derivatives, $\Delta\mathbf{x}_i$, $\Delta\mathbf{a}_i$, and $\Delta \log f_{0i}$, and acceleration derivatives, $\Delta^2\mathbf{x}_i$, $\Delta^2\mathbf{a}_i$, and $\Delta^2 \log f_{0i}$, as illustrated in Figure 4.20.

The velocity and acceleration derivatives of $\mathbf{x}_i$ are calculated as the second-order regression

$$\Delta\mathbf{x}_i = \begin{cases} (\mathbf{x}_{i+1} - \mathbf{x}_{i-1})/2 & i = 1, 2, \ldots, N-2 \\ \mathbf{x}_{i+1} - \mathbf{x}_i & i = 0 \\ \mathbf{x}_i - \mathbf{x}_{i-1} & i = N-1 \end{cases} \tag{4.109}$$

$$\Delta^2\mathbf{x}_i = \begin{cases} \mathbf{x}_{i-1} - 2\mathbf{x}_i + \mathbf{x}_{i+1} & i = 1, 2, \ldots, N-2 \\ 0 & i = 0 \\ 0 & i = N-1 \end{cases} \tag{4.110}$$

where N is the number of frames, and $\Delta\mathbf{a}_i$, $\Delta^2\mathbf{a}_i$, $\Delta \log f_{0i}$ and $\Delta^2 \log f_{0i}$ are also calculated in the same manner.

4.5.2 HMM Training

The tests of HMM-based speech enhancement in this section examine four configurations of HMMs. The first configuration, WORD_G/8, employs whole word HMMs to deal with the feature vectors constructed as MFCC16-8 while the second configuration, WORD_G/23, also uses whole-word HMMs but they are based on the feature vectors formed as MFCC23-23. Each of these two configurations models 52 whole-word HMMs for each speaker training set and uses the GRID grammar and the vocabulary list to constrain the choice of possible model sequences as the language model. The third configuration, TRI_N/8, employs CD-triphone HMMs with the feature vectors configured as MFCC16-8 whereas the other, TRI_N/23, uses CD-triphone HMMs which are based on the feature vectors formed as MFCC23-23. These CD-triphone HMMs constrain the model sequences only with the CD-triphone context and no language model is applied. The training process first models 658 CD-triphone HMMs for the female speaker training set and 663 CD-triphone HMMs for the male speaker training set. Tree-based clustering

is then applied state-by-state to reduce the number of models to the range between 150 and 250 for each speaker with the MDL criterion.

Each of the model configurations is examined with the feature vectors framed at 5 ms interval and 1 ms interval. In the case of whole-word HMMs, i.e. WORD_G/8 and WORD_G/23, the number of states is set equal to 16 for 5 ms-frame interval, and 40 for 1 ms-frame interval. Alternatively, the number of states of TRI_N/8 and TRI_N/23 are set equal to 12 for 5 ms-frame interval and 24 for 1 ms-frame interval. These settings are based on the empirical knowledge shown in Figure 4.10, 4.12 and 4.17.

Table 4.17 summarises the model configurations for the evaluation. The configurations using 5 ms-frame interval or MFCC16-8 give priority to the noise robustness in the decoding process whereas the configurations using 1 ms-frame interval or MFCC23-23 give priority to raising the upper limit of speech quality of enhanced speech.

Configuration	Frame Int.	Feature	# States	# HMMs	Lang. Model
WORD_G/8	5 ms	MFCC16-8	16	52	YES
	1 ms		40		
WORD_G/23	5 ms	MFCC23-23	16		
	1 ms		40		
TRI_N/8	5 ms	MFCC16-8	12	150-250	NO
	1 ms		24		
TRI_N/23	5 ms	MFCC23-23	12		
	1 ms		24		

Table 4.17: Model configurations.

The performance of the proposed method is examined in white noise and babble noise at SNRs from -5 dB to 10 dB. Therefore, HMMs are also trained in those noise conditions so that the noise-matched HMMs can be selected in the decoding process. This method using the noise-matched HMMs is impractical unless the noise type and its SNR is known *a priori*. This problem is discussed later in Chapter 5 and thus, the noise-matched HMMs are provisionally exploited regardless of those impracticality at this point.

4.5.3 HMM Decoding

After off-line training of HMMs, the test set of 200 utterances from each speaker is decoded by HMMs, which are trained in clean and noise-matched conditions, to produce the model and state sequences. The model configurations of WORD_G/8 and WORD_G/23 constrain the model sequences with the language model representing GRID-specific configurations while TRI_N/8 and TRI_N/23 constrain the model sequences with the context of the previous and next phonemes and no language model representing practical configurations though the small vocabulary of GRID database constrains the triphone coverage.

The observation vectors for the tests comprise components of the MFCCs, aperiodicity and fundamental frequency with their velocity and acceleration derivatives. However, the Viterbi algorithm and forward-backward algorithm to obtain the most likely model and state sequences should be applied only to the MFCC components in the vectors because the MFCC vectors represent the motion of vocal tract cavities as discussed in Section 3.2 and 4.3.2. Therefore, other components in the vectors are ignored during the decoding process, and the derivatives of MFCC vectors are also ignored from the computation because the preliminary tests have shown these components cause a decline in decoding accuracy in case of using noise-matched HMMs.

Figure 4.27 shows the resultant accuracies of model sequences in different model configurations and frame intervals. Subplots (a) and (b) show the accuracies of model sequences in white noise and babble noise respectively with the feature vectors, configured as MFCCC16-8 and MFCC23-23, framed at 5 ms interval. In these conditions, TRI_N/8 performs as well as the whole-word models in spite of no grammar applied. This reproduces the result shown in Figure 4.18, in which the advantages of using CD-triphone HMMs are discussed. Alternatively, the accuracy of TRI_N/23 is significantly lower than others. This is attributed to the effectiveness of the truncation of MFCC coefficients to represent motion of the vocal tract cavities as discussed in Section 4.3.1, and the disadvantage of the vector configuration is not compensated by the language model unlike WORD_G/23.

Subplots (c) and (d) show the results in white and babble noise with the feature vectors framed at 1 ms interval. In these conditions the noise robustness becomes lower

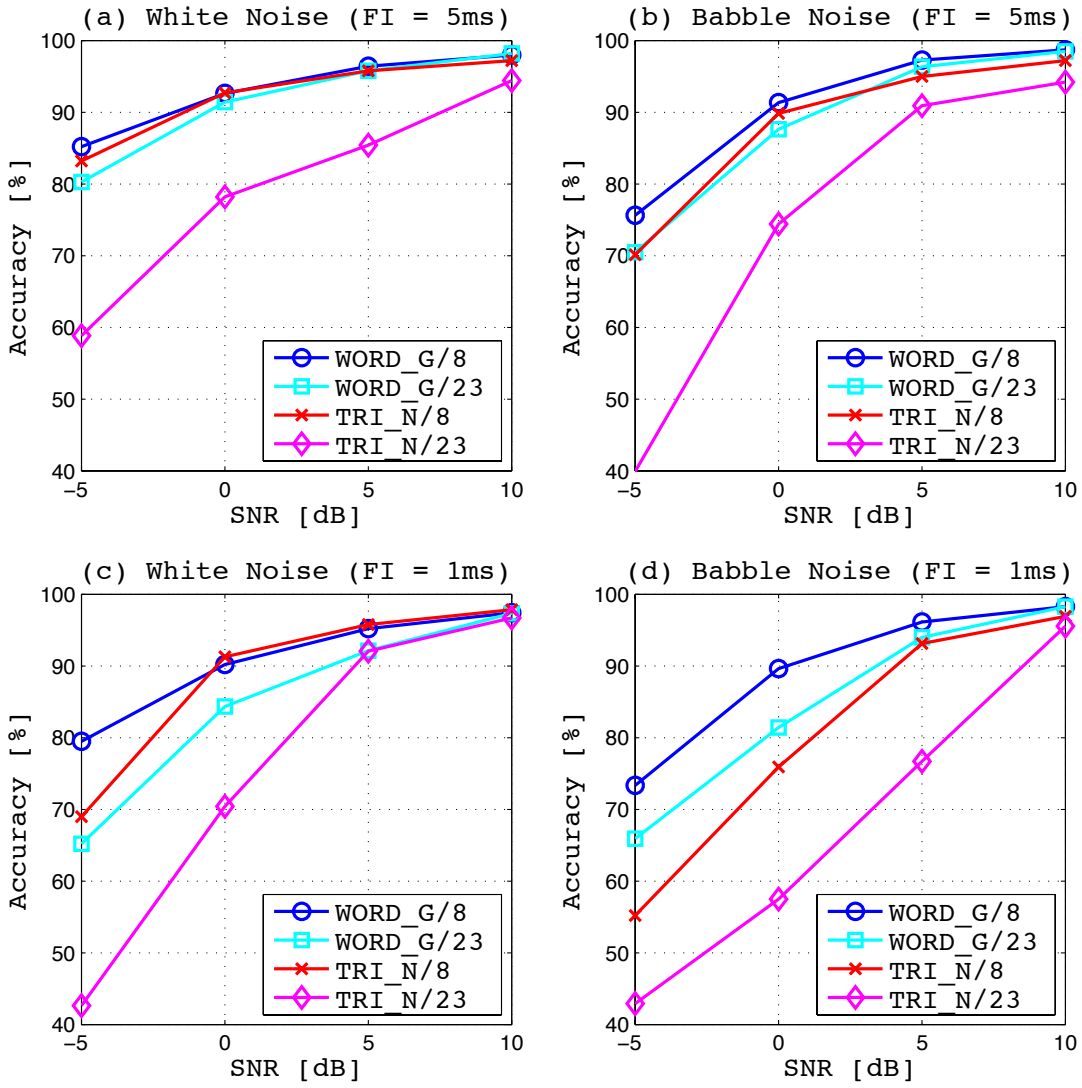

Figure 4.27: The accuracy of model sequences in different model configurations. a) and b) show accuracy in white noise and babble noise respectively, with the feature vectors framed at 5 ms interval while c) and d) shows the results with the feature vectors framed at 1 ms interval.

than the cases of 5 ms-frame interval over all the model configurations.

4.5.4 HMM-Based Parameter Synthesis

In the synthesis process, the HMMs trained in a clean condition are used to synthesise clean speech parameters according to the model and state sequences obtained in the decoding process though they are derived by the noise-matched HMMs. This brings inconsistency between the decoding result and synthesised parameters, and causes the resultant enhanced speech having distortions because the clean HMMs and the noise-matched HMMs are trained independently. This problem is discussed in Chapter 5 and thus, using different HMMs between decoding and synthesis is provisionally allowed regardless of the mismatch in HMMs at this point.

The synthesised speech parameters are converted to spectral envelopes and aperiodicity measures in the time-frequency domain as discussed in Section 4.4.2. However, the fundamental frequency contour is estimated from the original noisy speech directly with PEFAC instead of using the synthesised $\log f_0$ parameters, and the enhanced speech is finally reconstructed from these parameters through the STRAIGHT vocoder.

4.5.5 Speech Quality

Figure 4.28 shows PESQ scores for different HMM-based enhancement configurations and for comparison results are included for the log MMSE representing the filtering methods of speech enhancement and no noise compensation (NNC). Subplots (a) and (b) show the results in white noise and babble noise respectively with the feature vectors framed at 5 ms interval. In these conditions WORD_G/23 shows the best performance over the range of SNRs though its accuracy of model sequences is lower than WORD_G/8 because the PESQ score of speech synthesis with the feature vectors configured as MFCC23-23 is higher than the case of using the truncated vectors, MFCC16-8, as shown in Table 4.11. This represents the importance of balance between the performances in decoding and synthesis. TRI_N/8 keeps higher score than TRI_N23 at SNRs at 0 dB and below but the scores in high SNRs, such as at 5 dB and 10 dB, are inferior to TRI_N/23 whose noise robustness is the worst of the four HMM-based enhancement configurations. In white noise, the PESQ scores of all four configurations are superior to log MMSE at SNRs of

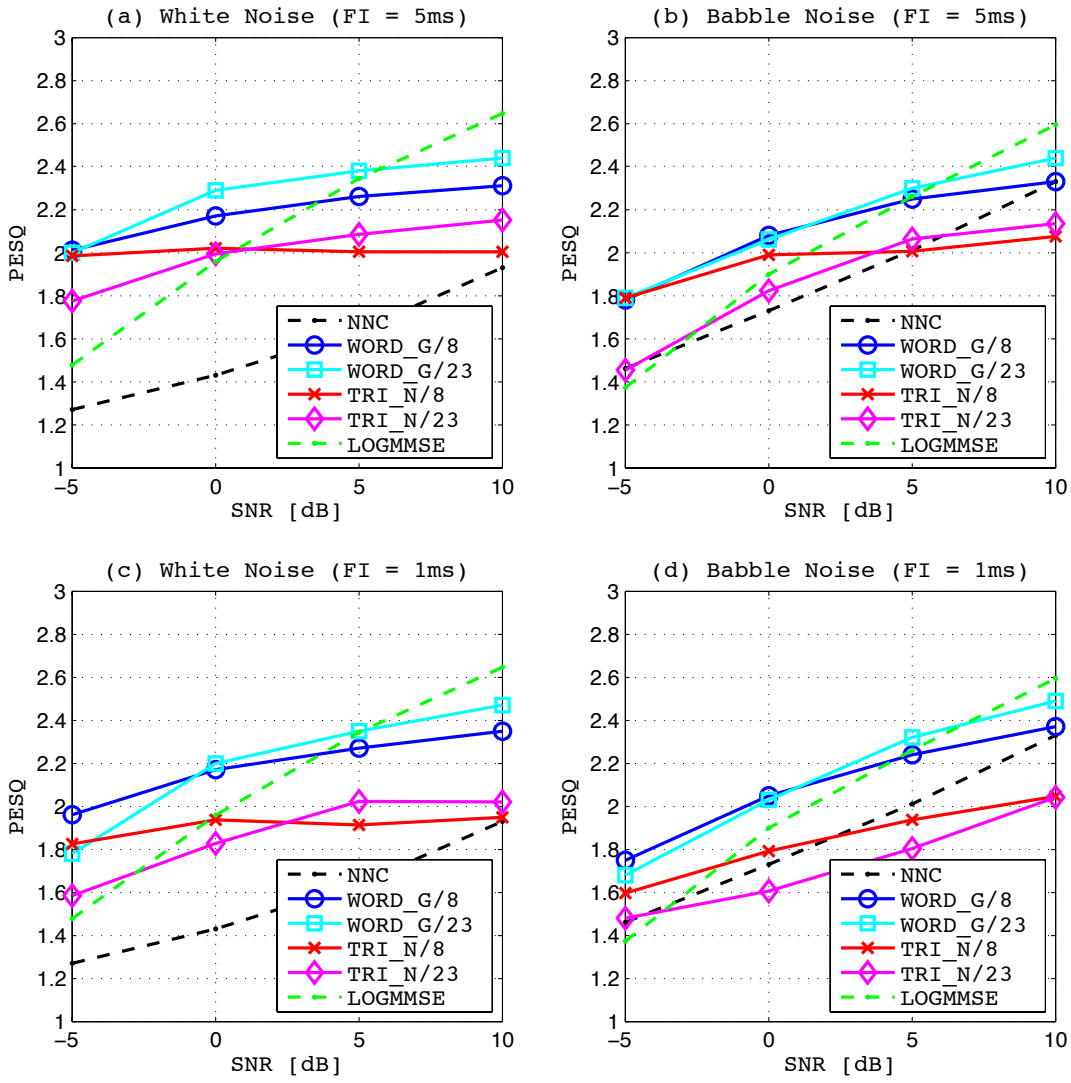

Figure 4.28: PESQ scores in different model configurations comparing with the log MMSE method and no noise compensation (NNC). a) and b) show the PESQ scores of enhanced speech in white noise and babble noise respectively, with the feature vectors framed at 5 ms interval while c) and d) show the results with the feature vectors framed at 1 ms interval.

0 dB and below while only TRI_N/23 is lower than the PESQ scores of log MMSE and does not show significant effect to NNC in babble noise.

Subplots (c) and (d) illustrate the PESQ scores in white noise and babble noise in the case of using the feature vectors framed at 1 ms interval showing the PESQ scores of each model configuration except WORD_G/8 become lower than the case of 5 ms-frame interval. This is attributed to the lower decoding accuracy than the case of 5 ms-frame interval, and the configurations using the triphone models do not show significant effect to NNC in babble noise in this condition.

The results of this experiment motivate to use the proposed method for speech enhancement at low SNRs such as 0 dB and below specifically with the configuration of TRI_N/8 with 5 ms-frame interval because it is not GRID-specific setting.

4.5.6 Speech Intelligibility

Figure 4.29 shows NCM representing objective measures for speech intelligibility. Subplots (a) and (b) show the results in white noise and babble noise respectively with the feature vectors framed at 5 ms interval. In these conditions WORD_G/8, WORD_G/23 and TRI_N/8 remain very stable even at low SNRs whereas TRI_N/23 drops the NCM score at low SNRs because of low decoding accuracy in those SNRs. The NCM scores of the log MMSE method falls more rapidly and when SNR falls to around 0 dB, log MMSE performs worse than the whole-word models in terms of NCM.

Subplots (c) and (d) illustrate the NCM scores in white noise and babble noise in the case of using the feature vectors framed at 1 ms interval showing the NCM scores of CD-triphone HMM configurations fall. This is also attributed to the results of decoding accuracy with each configurations.

In addition to the results based on PESQ and NCM scores, figures 4.30 and 4.31 compare the spectrogram of enhanced speech with the original clean and noisy speech. Subplot (a) of each figure shows the narrowband spectrogram of natural clean speech of an utterance, “Bin Blue At E Six Now”, spoken by a female speaker. Subplot (b) represents the noisy speech in white noise at SNR of -5 dB in Figure 4.30 and in babble noise in Figure 4.31. Subplot (c) in each figure shows the spectrogram of enhanced speech by HMM-based speech enhancement with the model configuration of TRI_N/8

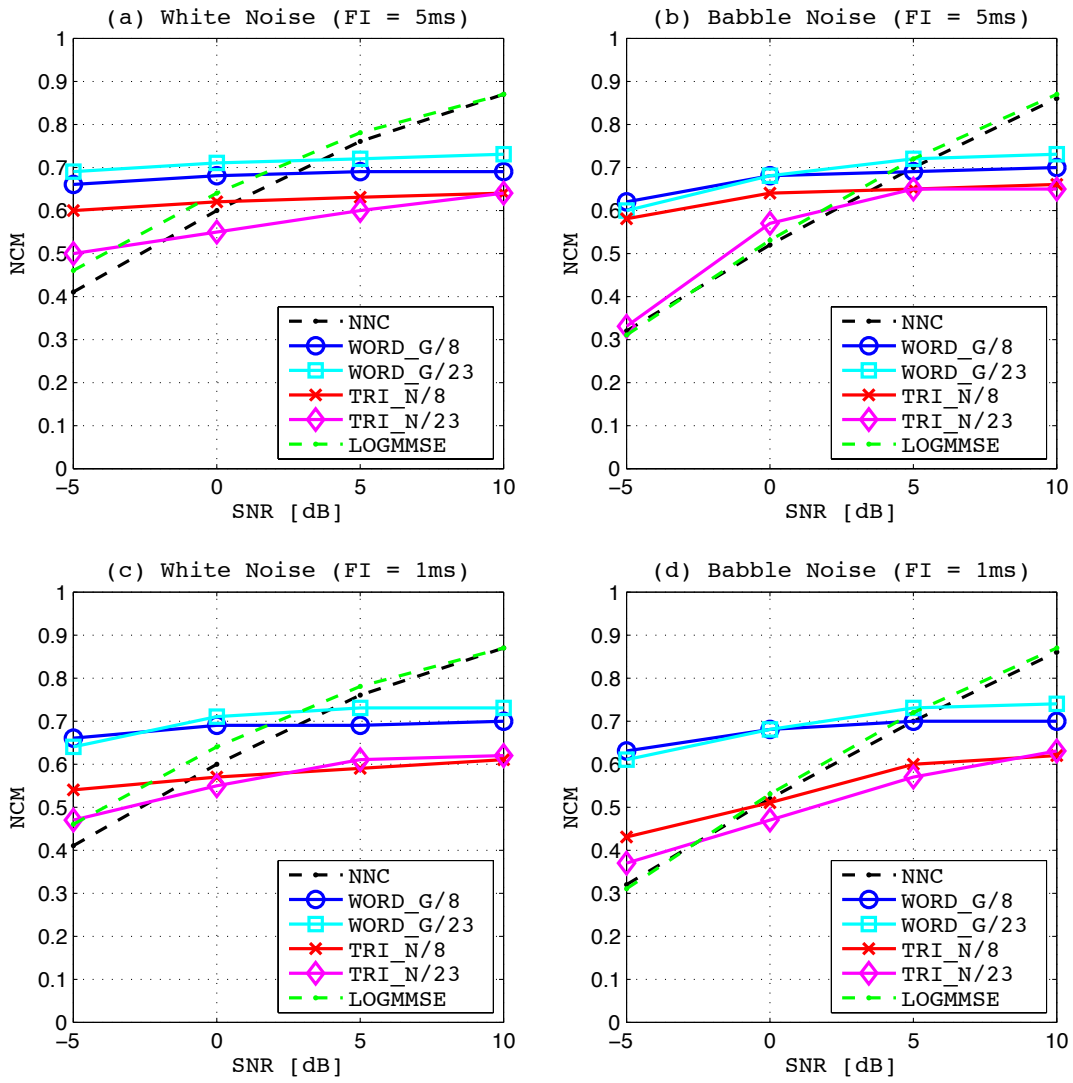

Figure 4.29: NCM scores in different model configurations comparing with the log MMSE method and no noise compensation (NNC). a) and b) show the NCM scores of enhanced speech in white noise and babble noise respectively, with the feature vectors framed at 5 ms interval while c) and d) show the results with the feature vectors framed at 1 ms interval.

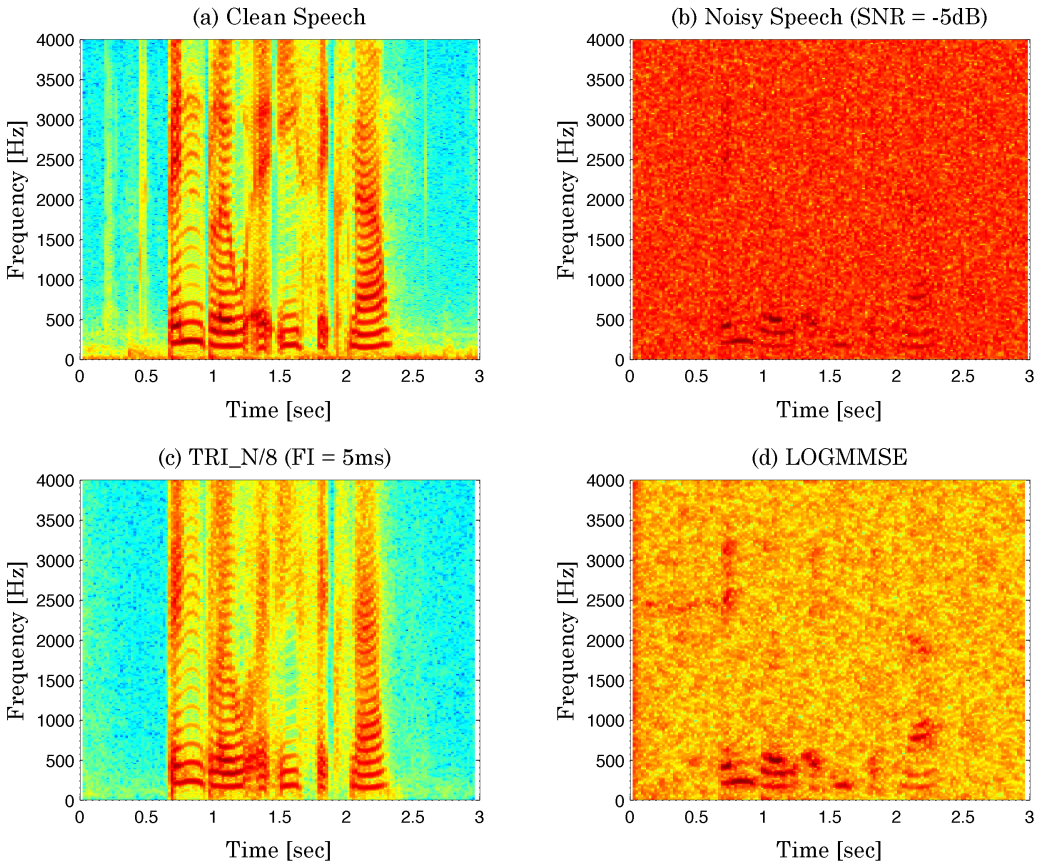

Figure 4.30: Narrowband spectrograms of speech, "Bin Blue At E Six Now", spoken by a female speaker. a) is natural clean speech. b) is contaminated with white noise at SNR of -5 dB. c) is enhanced speech with HMM-based speech enhancement using TRI_N/8 configuration while d) is enhanced by the log MMSE method.

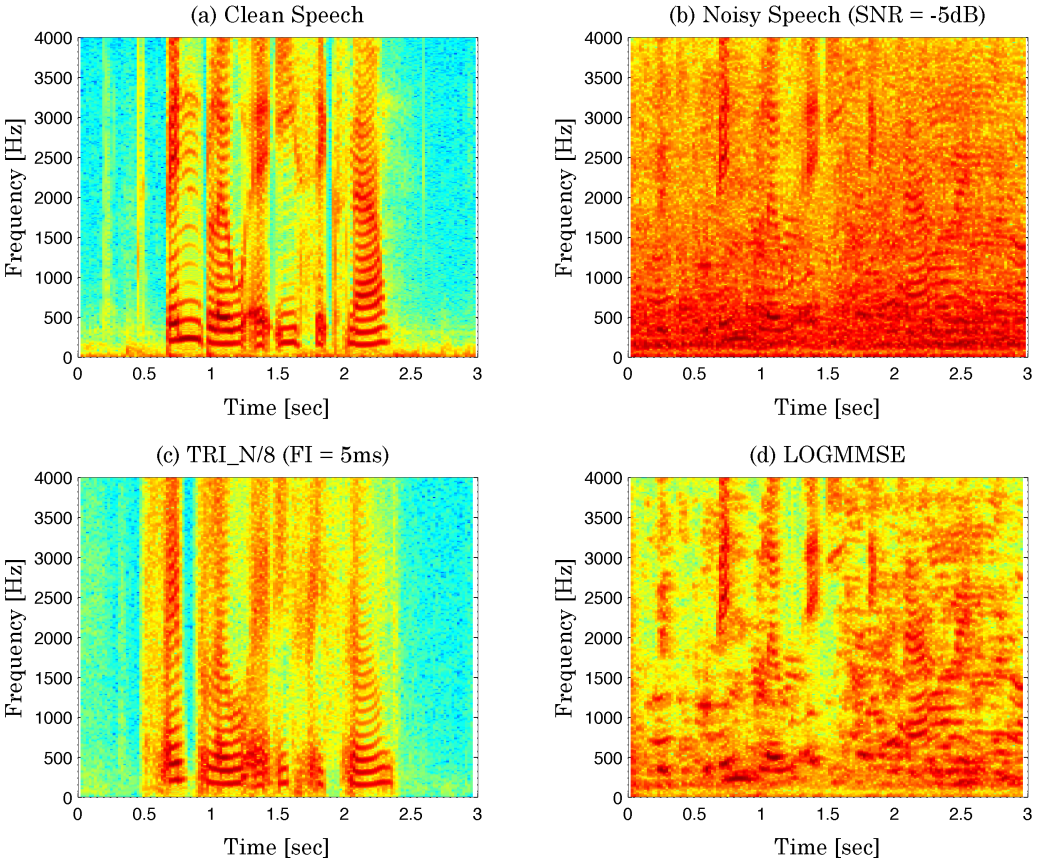

Figure 4.31: Narrowband spectrograms of speech, "Bin Blue At E Six Now", spoken by a female speaker. a) is natural clean speech. b) is contaminated with babble noise at SNR of -5 dB. c) is enhanced speech with HMM-based speech enhancement using TRI_N/8 configuration while d) is enhanced by the log MMSE method.

using the feature vectors framed at 5 ms interval while subplot (d) is enhanced by the log MMSE method. These figures show that the enhanced speech by HMM-based speech enhancement can reconstruct the original clean speech without residual noise even at SNR of -5 dB whereas the filtering method remains a lot of residual noise. Subplot (c) in Figure 4.31, however, shows the influences of decoding errors at the beginning and end of the utterance and this is the biggest issue of HMM-based speech enhancement.

4.6 Conclusion of the Chapter

This chapter first discussed the overview of HMMs and the theories were then extended to the practical applications such as ASR and HMM-based speech synthesis. The latter part of the chapter explored HMM-based speech enhancement achieved by combining the techniques of HMM training, HMM decoding and HMM synthesis with the STRAIGHT speech production model. Experiments evaluated the performance of the speech enhancement with different sets of configurations comparing with the log MMSE method which represents the conventional filtering methods. The experimental analysis has shown that using CD-triphone HMMs with no grammar constraints, e.g. TRI_N/8, with 5 ms-frame interval achieves the PESQ and NCM scores sufficiently close to that with grammar constrained whole-word models, but puts no restrictions on the input speech. Compared to conventional methods of speech enhancement, HMM-based speech enhancement has higher PESQ and NCM scores at lower SNRs. In fact, the scores of PESQ and NCM remain relatively stable as SNRs reduce. However, tests at higher SNRs show that those scores are limited to relatively low levels, compared to that of the original speech, which puts a low upper limit on performance.

Chapter 5

Adaptation of Hidden Markov Models to Noisy Speech

HMM-based speech enhancement in the previous chapter trained the HMMs with clean speech for the clean HMMs and with noisy speech for the noise-matched HMMs and then decoded the noisy speech with the noise-matched HMMs. This is, however, impractical because it is not possible to know in advance the noise type and SNR of the input speech and train the HMMs in that condition *a priori*. Moreover, the parameters of enhanced speech were synthesised by using the clean HMMs though the model and state sequences had been derived from the noise-matched HMMs, and thus, this may cause distortion of the synthesised speech. To tackle this problem, this chapter discusses a method to model the input noisy speech from the clean HMMs as an online process by using techniques of HMM adaptation in order to acquire accurate state and model sequences from the noisy speech at the decoding process of HMM-based speech enhancement.

5.1 Introduction

At the stage of HMM decoding in HMM-based speech enhancement, acquisition of the model and state sequence from noisy speech without decoding errors is the key problem. If a set of HMMs, $\mathbf{\Lambda} = \{\lambda_1, \lambda_2, \ldots, \lambda_D\}$, is trained with clean speech, the statistical parameters in λ_d $(d = 1, 2, \ldots, D)$ do not match the statistical distribution of the features in noisy speech especially at low SNRs. Therefore, decoding noisy speech with clean

HMMs is not a suitable solution for HMM-based speech enhancement. For this reason, the experiments in Section 4.5 employed the noise-matched HMMs, $\mathbf{\Lambda}' = \{\lambda'_1, \lambda'_2, \ldots, \lambda'_D\}$, which had been trained with noisy speech.

This method gives the statistical parameters which match the statistical distribution of the features in noisy speech to λ'_d and thus, the decoding accuracy is expected to be improved. This method, however, has two main drawbacks. Firstly, it is generally not possible to know in advance the noise type and SNR of the input speech at the training stage in practical use. Therefore, HMMs need to be trained with vast amount of speech contaminated with various types and levels of noise in order to deal with any unknown noisy speech. Furthermore, this raises the number of the models, i.e. the candidates of the choice are increased, and then it may affects the decoding accuracy.

Secondly, HMM-based speech enhancement uses $\mathbf{\Lambda}$ to synthesise the clean speech parameters, therefore, a state sequence, $\mathbf{q}$, for $\mathbf{\Lambda}$ is required at the synthesis stage. However, the method using the noise-matched HMMs obtains a state sequence, $\mathbf{q}'$, for $\mathbf{\Lambda}'$ instead of $\mathbf{q}$ in the decoding process. Consequently, $\mathbf{\Lambda}$ synthesises the parameters according to $\mathbf{q}'$, and this may cause the output to have distortion because the state allocation in λ_k and the state allocation in λ'_k are not identical. Figure 5.1 illustrates this problem. λ and λ' in the figure have different state allocation for the same waveform, and if the segments of the waveform corresponding to each state of λ (i.e., $S1, S2, S3, S4$) are allocated according to the allocation of the states in λ' (i.e., $S1', S2', S3', S4'$), the reconstructed wave form has distortion (red line). This may occur in HMM-based speech enhancement as long as the noise-matched HMMs are used in the decoding process.

To achieve robust performance against noise in HMM decoding process while avoiding the preceding problems, an approach to adapt $\mathbf{\Lambda}$ to characteristics of the noisy speech input without changing their state allocation is discussed in this chapter. Several methods for this approach have been proposed such as state based filtering [100], cepstral mean compensation [101, 102], HMM decomposition [103] and parallel model combination [104–106] and it is reported that parallel model combination can perform more effectively for HMMs modelled in the MFCC domain [105].

The remainder of this chapter first discusses the HMM adaptation with parallel model combination including the techniques to determine mismatch function and to

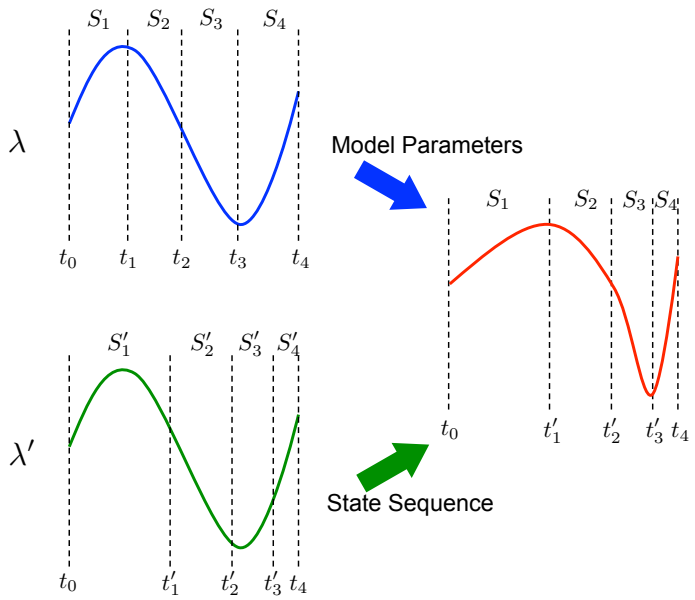

Figure 5.1: Distortion brought by temporal inconsistency of the states between clean and noise-matched HMMs.

deal with a non-linear transform of the statistical distribution needed by the logarithm operation in the MFCC extraction process. Experiments then evaluate the performance of the noise-adapted HMMs modified from the clean HMMs by parallel model combination comparing with noise-matched HMMs prior to the conclusion of the chapter.

5.2 Parallel Model Combination

HMM adaptation with parallel model combination in this section is discussed on the following assumptions [106] and an outline of parallel model combination is illustrated in Figure 5.2.

- Speech and noise are independent
- Speech and noise are additive in the time domain
- The clean HMMs have been modelled as static parameters of MFCCs, i.e. dynamic features are not included.
- Noise is modelled as a single-state HMM.

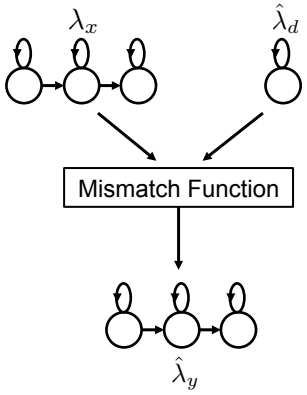

Figure 5.2: Outline of parallel model combination

The noise-adapted HMMs, $\hat{\lambda}_y$, are produced by combining the clean HMMs, λ_x, with a noise model, $\hat{\lambda}_d$, estimated from the noisy speech input according to a mismatch function which is determined as the effect of noise in speech.

5.2.1 Mismatch Function

Discrete-time speech, $x(n)$, and random noise, $d(n)$, are additive in the time domain, therefore, noisy speech, $y(n)$, is given as

$$y(n) = x(n) + d(n) \tag{5.1}$$

Using an F-point STFT, the power spectrum of the noisy speech at the i-th frame is derived as

$$\begin{aligned} \mid Y_i(f) \mid^2 &= (X_i(f) + D_i(f))\,(X_i(f) + D_i(f))^* && (5.2)\\ &= \mid X_i(f) \mid^2 + \mid D_i(f) \mid^2 + 2 \mid X_i(f) \mid\mid D_i(f) \mid \cos(\phi(f)) && (5.3) \end{aligned}$$

where $f = 0, 1, \ldots, F-1$ and $\phi(f)$ represents the phase difference between the clean speech and noise in frequency bin, f. Although [107, 108] have reported that this phase difference term should not be ignored for precise calculation, $\phi(f)$ can not be estimated from the noisy speech input. Therefore, [107] has proposed to utilise a lookup table of phase differences that is computed offline during a training process but this solution requires *a priori* knowledge about noise at the training stage. Therefore, this brings a

limitation into practical use. Alternatively, [108] has modelled the phase difference term at each frequency bin as independent zero-mean Gaussian distributions from empirical data. Thus, the phase difference term in Equation (5.3) should be set equal to zero in a maximum likelihood sense, which gives,

$$| Y_i(f) |^2 \approx | X_i(f) |^2 + | D_i(f) |^2 \tag{5.4}$$

$|D_i(f)|^2$ in Equation (5.4) is estimated from the noisy speech input, $y(n)$, by using the noise estimation algorithms discussed in Section 2.2 and it derives the following equation.

$$| Y_i(f) |^2 \approx | X_i(f) |^2 + | \hat{D}_i(f) |^2 \tag{5.5}$$

where $|\hat{D}_i(f)|^2$ denotes the estimated power spectrum of noise.

In the M-channel linear Mel-filterbank domain, the mismatch function is derived as

$$Y_i^{fb}(m) = X_i^{fb}(m) + \hat{D}_i^{fb}(m), \qquad m = 0, 1, \ldots, M-1 \tag{5.6}$$

where $Y_i^{fb}(m)$, $X_i^{fb}(m)$ and $\hat{D}_i^{fb}(m)$ are the m-th linear Mel-filterbank coefficient of noisy speech, clean speech and estimated noise respectively at the i-th frame. At this point, the noise in the linear Mel-filterbank domain is modelled as Gaussian distribution with mean vector, $\hat{\boldsymbol{\mu}}_d^{fb}$, and covariance matrix, $\hat{\boldsymbol{\Sigma}}_d^{fb}$ given as

$$\hat{\lambda}_d^{fb} = \mathcal{N}\left(\hat{\mathbf{D}}^{fb}; \hat{\boldsymbol{\mu}}_d^{fb}, \hat{\boldsymbol{\Sigma}}_d^{fb}\right) \tag{5.7}$$

$$\hat{\boldsymbol{\mu}}_d^{fb} = \frac{1}{I}\sum_{i=0}^{I-1} \hat{\mathbf{d}}_i^{fb} \tag{5.8}$$

$$= \left[\hat{\mu}_0^d, \hat{\mu}_1^d, \ldots, \hat{\mu}_{M-1}^d\right]^T \tag{5.9}$$

$$\hat{\mathbf{\Sigma}}_d^{fb} = \frac{1}{I-1}\sum_{i=0}^{I-1}\left(\hat{\mathbf{d}}_i^{fb}-\hat{\boldsymbol{\mu}}_d^{fb}\right)\left(\hat{\mathbf{d}}_i^{fb}-\hat{\boldsymbol{\mu}}_d^{fb}\right)^T \tag{5.10}$$

$$= \begin{bmatrix} \hat{\Sigma}_{00}^d & \hat{\Sigma}_{01}^d & \cdots & \hat{\Sigma}_{0(M-1)}^d \\ \hat{\Sigma}_{10}^d & \hat{\Sigma}_{11}^d & \cdots & \hat{\Sigma}_{1(M-1)}^d \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\Sigma}_{(M-1)0}^d & \hat{\Sigma}_{(M-1)1}^d & \cdots & \hat{\Sigma}_{(M-1)(M-1)}^d \end{bmatrix} \tag{5.11}$$

where

$$\hat{\mathbf{D}}^{fb} = \left[(\hat{\mathbf{d}}_0^{fb})^T, (\hat{\mathbf{d}}_1^{fb})^T, \ldots, (\hat{\mathbf{d}}_{I-1}^{fb})^T\right]^T \tag{5.12}$$

$$\hat{\mathbf{d}}_i^{fb} = \left[\hat{D}_i^{fb}(0), \hat{D}_i^{fb}(1), \ldots, \hat{D}_i^{fb}(M-1)\right]^T \tag{5.13}$$

where I denotes the number of frames in $y(n)$.

The clean HMMs, λ_x, are, however, in the non-linear MFCC domain. Therefore, they need to be transformed to the linear Mel-filterbank domain in order to be combined with the noise model, $\hat{\lambda}_d^{fb}$, in the linear Mel-filterbank domain. The mean vectors and covariance matrices of the clean HMMs, $\boldsymbol{\mu}_x$ and $\mathbf{\Sigma}_x$, are first transformed to the log Mel-filterbank domain as

$$\boldsymbol{\mu}_x^l = \mathbf{C}^{-1}\boldsymbol{\mu}_x \tag{5.14}$$

$$= \left[\mu_0^{lx}, \mu_1^{lx}, \ldots, \mu_{M-1}^{lx}\right]^T \tag{5.15}$$

$$\mathbf{\Sigma}_x^l = \mathbf{C}^{-1}\mathbf{\Sigma}_x(\mathbf{C}^{-1})^T \tag{5.16}$$

$$= \begin{bmatrix} \Sigma_{00}^{lx} & \Sigma_{01}^{lx} & \cdots & \Sigma_{0(M-1)}^{lx} \\ \Sigma_{10}^{lx} & \Sigma_{11}^{lx} & \cdots & \Sigma_{1(M-1)}^{lx} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{(M-1)0}^{lx} & \Sigma_{(M-1)1}^{lx} & \cdots & \Sigma_{(M-1)(M-1)}^{lx} \end{bmatrix} \tag{5.17}$$

where $\mathbf{C}^{-1}$ is the notation of the inverse of DCT matrix, $\mathbf{C}$.

The model parameters of the clean HMMs in the log Mel-filterbank domain, $\boldsymbol{\mu}_x^l$ and $\mathbf{\Sigma}_x^l$, are next transformed to the linear Mel-filterbank domain in which the clean HMMs and the noise model are combined in accordance with Equation (5.6). This transform is, however, non-linear and thus, a technique to deal with the non-linear mapping of the

statistical parameters is required. Specifically, previous research has shown that a mapping of Gaussian distribution in the log spectral domain into log-normal distribution in the linear spectral domain performs well [104] whereas [7,109] have successfully exploited unscented transform to obtain the statistical distribution of clean HMMs in the linear Mel-filterbank domain. Therefore, these two approaches to map the model parameters between the linear domain and log domain are explored below.

5.2.2 Distribution Mapping between Gaussian and Log-Normal

To transform the statistical distribution in the log Mel-filterbank domain to the linear Mel-filterbank domain, this section employs a mapping between a Gaussian distribution in the log Mel-filterbank domain and a log-normal distribution in the linear Mel-filterbank domain as follows.

The mean vectors and the covariance matrices of the clean HMMs in the log-Mel filterbank domain, which have been derived by Equations (5.14) - (5.17), are first transformed to the linear Mel-filterbank domain using the distribution mapping between Gaussian and log-normal [104] as

$$\boldsymbol{\mu}_x^{fb} = \left[\mu_0^x, \mu_1^x, \ldots, \mu_{M-1}^x\right]^T \tag{5.18}$$

$$\boldsymbol{\Sigma}_x^{fb} = \begin{bmatrix} \Sigma_{00}^x & \Sigma_{01}^x & \cdots & \Sigma_{0(M-1)}^x \\ \Sigma_{10}^x & \Sigma_{11}^x & \cdots & \Sigma_{1(M-1)}^x \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{(M-1)0}^x & \Sigma_{(M-1)1}^x & \cdots & \Sigma_{(M-1)(M-1)}^x \end{bmatrix} \tag{5.19}$$

where

$$\begin{aligned} \mu_m^x &= \exp\left(\mu_m^{lx} + \Sigma_{mm}^{lx}/2\right), && m = 0, 1, \ldots, M-1 \\ \Sigma_{jk}^x &= \mu_j^x \mu_k^x \left[\exp\left(\Sigma_{jk}^{lx}\right) - 1\right], && j, k = 0, 1, \ldots, M-1 \end{aligned} \tag{5.20}$$

At this point the models corresponding to $X_i^{fb}(m)$ and $\hat{D}_i^{fb}(m)$ in the mismatch function of Equation 5.6 are obtained as the parameter sets, ($\boldsymbol{\mu}_x^{fb}$, $\boldsymbol{\Sigma}_x^{fb}$) and ($\hat{\boldsymbol{\mu}}_d^{fb}$, $\hat{\boldsymbol{\Sigma}}_d^{fb}$), therefore, the noise-adapted HMMs, $\hat{\lambda}_y$, are constituted by the mean vectors, $\hat{\boldsymbol{\mu}}_y^{fb}$, and

the covariance matrices, $\hat{\mathbf{\Sigma}}_y^{fb}$, which are derived as

$$\hat{\lambda}_y = \mathcal{N}\left(\mathbf{Y}^{fb}; \hat{\boldsymbol{\mu}}_y^{fb}, \hat{\mathbf{\Sigma}}_y^{fb}\right) \tag{5.21}$$

$$\hat{\boldsymbol{\mu}}_y^{fb} = \boldsymbol{\mu}_x^{fb} + \hat{\boldsymbol{\mu}}_d^{fb} \tag{5.22}$$

$$= \left[\hat{\mu}_0^y, \hat{\mu}_1^y, \ldots, \hat{\mu}_{M-1}^y\right]^T \tag{5.23}$$

$$\hat{\mathbf{\Sigma}}_y^{fb} = \mathbf{\Sigma}_x^{fb} + \hat{\mathbf{\Sigma}}_d^{fb} \tag{5.24}$$

$$= \begin{bmatrix} \hat{\Sigma}_{00}^y & \hat{\Sigma}_{01}^y & \cdots & \hat{\Sigma}_{0(M-1)}^y \\ \hat{\Sigma}_{10}^y & \hat{\Sigma}_{11}^y & \cdots & \hat{\Sigma}_{1(M-1)}^y \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\Sigma}_{(M-1)0}^y & \hat{\Sigma}_{(M-1)1}^y & \cdots & \hat{\Sigma}_{(M-1)(M-1)}^y \end{bmatrix} \tag{5.25}$$

where

$$\mathbf{Y}^{fb} = \left[(\mathbf{y}_0^{fb})^T, (\mathbf{y}_1^{fb})^T, \ldots, (\mathbf{y}_{I-1}^{fb})^T\right]^T \tag{5.26}$$

$$\mathbf{y}_i^{fb} = \left[Y_i^{fb}(0), Y_i^{fb}(1), \ldots, Y_i^{fb}(M-1)\right]^T \tag{5.27}$$

Assuming the combined distribution remains a log-normal distribution [104], $\hat{\boldsymbol{\mu}}_y^{fb}$ and $\hat{\mathbf{\Sigma}}_y^{fb}$ are converted back to a Gaussian distribution in the log Mel-filterbank domain to derive the mean vectors and the covariance matrices in the log Mel-filterbank domain, $\hat{\boldsymbol{\mu}}_y^l$ and $\hat{\mathbf{\Sigma}}_y^l$, as

$$\hat{\boldsymbol{\mu}}_y^l = \left[\hat{\mu}_0^{ly}, \hat{\mu}_1^{ly}, \ldots, \hat{\mu}_{M-1}^{ly}\right]^T \tag{5.28}$$

$$\hat{\mathbf{\Sigma}}_y^l = \begin{bmatrix} \hat{\Sigma}_{00}^{ly} & \hat{\Sigma}_{01}^{ly} & \cdots & \hat{\Sigma}_{0(M-1)}^{ly} \\ \hat{\Sigma}_{10}^{ly} & \hat{\Sigma}_{11}^{ly} & \cdots & \hat{\Sigma}_{1(M-1)}^{ly} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\Sigma}_{(M-1)0}^{ly} & \hat{\Sigma}_{(M-1)1}^{ly} & \cdots & \hat{\Sigma}_{(M-1)(M-1)}^{ly} \end{bmatrix} \tag{5.29}$$

where

$$\hat{\mu}_m^{ly} = \log\left(\hat{\mu}_m^y\right) - \frac{1}{2}\log\left(\frac{\hat{\Sigma}_{mm}^y}{(\hat{\mu}_m^y)^2} + 1\right) \tag{5.30}$$

$$\hat{\Sigma}_{jk}^{ly} = \log\left(\frac{\hat{\Sigma}_{jk}^y}{\hat{\mu}_j^y \hat{\mu}_k^y} + 1\right) \tag{5.31}$$

These parameters are then transformed to the MFCC domain as

$$\hat{\boldsymbol{\mu}}_y = \mathbf{C}\hat{\boldsymbol{\mu}}_y^l \tag{5.32}$$

$$\hat{\boldsymbol{\Sigma}}_y = \mathbf{C}\hat{\boldsymbol{\Sigma}}_y^l\mathbf{C}^T \tag{5.33}$$

Now the noise-adapted HMMs, $\hat{\lambda}_y$, which comprise a set of mean vectors, $\hat{\boldsymbol{\mu}}_y$, and covariance matrices, $\hat{\boldsymbol{\Sigma}}_y$, are obtained.

5.2.3 Unscented Transform

Alternatively, this section discusses the unscented transform as an alternative to the distribution mapping between Gaussian and log-normal to convert a set of clean speech HMMs in the log Mel-filterbank domain, $\boldsymbol{\mu}_x^l$ and $\boldsymbol{\Sigma}_x^l$, to the distribution in the linear Mel-filterbank domain. The unscented transform first extracts sigma points from the distribution, using $\boldsymbol{\mu}_x^l$ and $\boldsymbol{\Sigma}_x^l$, and then sigma point matrices, $\mathbf{S}_x^{l+}$ and $\mathbf{S}_x^{l-}$, are formed as

$$\mathbf{S}_x^{l+} = \begin{bmatrix} s_{00}^{lx+} & s_{01}^{lx+} & \cdots & s_{0(M-1)}^{lx+} \\ s_{10}^{lx+} & s_{11}^{lx+} & \cdots & s_{1(M-1)}^{lx+} \\ \vdots & \vdots & \ddots & \vdots \\ s_{(M-1)0}^{lx+} & s_{(M-1)1}^{lx+} & \cdots & s_{(M-1)(M-1)}^{lx+} \end{bmatrix} \tag{5.34}$$

$$\mathbf{S}_x^{l-} = \begin{bmatrix} s_{00}^{lx-} & s_{01}^{lx-} & \cdots & s_{0(M-1)}^{lx-} \\ s_{10}^{lx-} & s_{11}^{lx-} & \cdots & s_{1(M-1)}^{lx-} \\ \vdots & \vdots & \ddots & \vdots \\ s_{(M-1)0}^{lx-} & s_{(M-1)1}^{lx-} & \cdots & s_{(M-1)(M-1)}^{lx-} \end{bmatrix} \tag{5.35}$$

where

$$s_{jk}^{lx+} = \begin{cases} \mu_j^{lx} + \sqrt{\Sigma_{jj}^{lx}} & j = k \\ \mu_j^{lx} & j \neq k \end{cases} \tag{5.36}$$

$$s_{jk}^{lx-} = \begin{cases} \mu_j^{lx} - \sqrt{\Sigma_{jj}^{lx}} & j = k \\ \mu_j^{lx} & j \neq k \end{cases} \tag{5.37}$$

where $j, k = 0, 1, \ldots, M-1$. Then, $\boldsymbol{\mu}_x^l$, $\mathbf{S}_x^{l+}$ and $\mathbf{S}_x^{l-}$ are transformed to the linear Mel-filterbank domain by taking exponent at each element as

$$\boldsymbol{\mu}_x^{fb} = \left[\mu_0^x, \mu_1^x, \ldots, \mu_{M-1}^x\right]^T \tag{5.38}$$

$$\mathbf{S}_x^{fb+} = \begin{bmatrix} s_{00}^{x+} & s_{01}^{x+} & \cdots & s_{0(M-1)}^{x+} \\ s_{10}^{x+} & s_{11}^{x+} & \cdots & s_{1(M-1)}^{x+} \\ \vdots & \vdots & \ddots & \vdots \\ s_{(M-1)0}^{x+} & s_{(M-1)1}^{x+} & \cdots & s_{(M-1)(M-1)}^{x+} \end{bmatrix} \tag{5.39}$$

$$\mathbf{S}_x^{fb-} = \begin{bmatrix} s_{00}^{x-} & s_{01}^{x-} & \cdots & s_{0(M-1)}^{x-} \\ s_{10}^{x-} & s_{11}^{x-} & \cdots & s_{1(M-1)}^{x-} \\ \vdots & \vdots & \ddots & \vdots \\ s_{(M-1)0}^{x-} & s_{(M-1)1}^{x-} & \cdots & s_{(M-1)(M-1)}^{x-} \end{bmatrix} \tag{5.40}$$

where

$$\mu_m^x = \exp\left(\mu_m^{lx}\right) \tag{5.41}$$

$$s_{jk}^{x+} = \exp\left(s_{jk}^{lx+}\right) \tag{5.42}$$

$$s_{jk}^{x-} = \exp\left(s_{jk}^{lx-}\right) \tag{5.43}$$

A brief outline of this transform where $M = 1$ is illustrated in Figure 5.3.

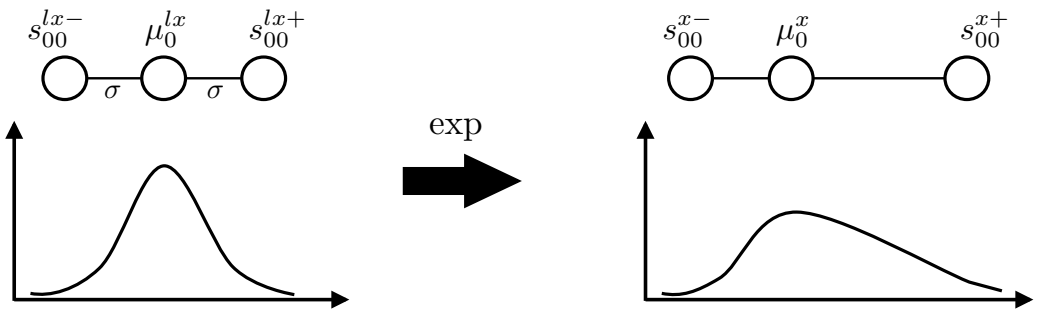

Figure 5.3: An brief outline of unscented transform ($M = 1$).

In the linear Mel-filterbank domain, sigma points from the noise model are also

extracted and then sigma point matrices, $\hat{\mathbf{S}}_d^{fb+}$ and $\hat{\mathbf{S}}_d^{fb-}$ are formed as

$$\hat{\mathbf{S}}_d^{fb+} = \begin{bmatrix} \hat{s}_{00}^{d+} & \hat{s}_{01}^{d+} & \cdots & \hat{s}_{0(M-1)}^{d+} \\ \hat{s}_{10}^{d+} & \hat{s}_{11}^{d+} & \cdots & \hat{s}_{1(M-1)}^{d+} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{s}_{(M-1)0}^{d+} & \hat{s}_{(M-1)1}^{d+} & \cdots & \hat{s}_{(M-1)(M-1)}^{d+} \end{bmatrix} \tag{5.44}$$

$$\hat{\mathbf{S}}_d^{fb-} = \begin{bmatrix} \hat{s}_{00}^{d-} & \hat{s}_{01}^{d-} & \cdots & \hat{s}_{0(M-1)}^{d-} \\ \hat{s}_{10}^{d-} & \hat{s}_{11}^{d-} & \cdots & \hat{s}_{1(M-1)}^{d-} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{s}_{(M-1)0}^{d-} & \hat{s}_{(M-1)1}^{d-} & \cdots & \hat{s}_{(M-1)(M-1)}^{d-} \end{bmatrix} \tag{5.45}$$

where

$$\hat{s}_{jk}^{d+} = \begin{cases} \hat{\mu}_j^d + \sqrt{\hat{\Sigma}_{jj}^d} & j = k \\ \hat{\mu}_j^d & j \neq k \end{cases} \tag{5.46}$$

$$\hat{s}_{jk}^{d-} = \begin{cases} \hat{\mu}_j^d - \sqrt{\hat{\Sigma}_{jj}^d} & j = k \\ \hat{\mu}_j^d & j \neq k \end{cases} \tag{5.47}$$

At this point the statistical distributions corresponding to $X_i^{fb}(m)$ and $\hat{D}_i^{fb}(m)$ in the mismatch function of Equation 5.6 are obtained, therefore, a set of parameters for the noisy speech distribution, i.e. the mean vector, $\hat{\boldsymbol{\mu}}_y^{fb}$, and sigma point matrices, $\hat{\mathbf{S}}_y^{fb+}$ and $\hat{\mathbf{S}}_y^{fb-}$, are derived as

$$\hat{\boldsymbol{\mu}}_y^{fb} = \boldsymbol{\mu}_x^{fb} + \hat{\boldsymbol{\mu}}_d^{fb} \tag{5.48}$$

$$= \left[\hat{\mu}_0^y, \hat{\mu}_1^y, \ldots, \hat{\mu}_{M-1}^y\right]^T \tag{5.49}$$

$$\hat{\mathbf{S}}_y^{fb+} = \mathbf{S}_x^{fb+} + \hat{\mathbf{S}}_y^{fb+} \tag{5.50}$$

$$= \begin{bmatrix} \hat{s}_{00}^{y+} & \hat{s}_{01}^{y+} & \cdots & \hat{s}_{0(M-1)}^{y+} \\ \hat{s}_{10}^{y+} & \hat{s}_{11}^{y+} & \cdots & \hat{s}_{1(M-1)}^{y+} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{s}_{(M-1)0}^{y+} & \hat{s}_{(M-1)1}^{y+} & \cdots & \hat{s}_{(M-1)(M-1)}^{y+} \end{bmatrix} \tag{5.51}$$

$$\hat{\mathbf{S}}_y^{fb-} = \mathbf{S}_x^{fb-} + \hat{\mathbf{S}}_y^{fb-} \tag{5.52}$$

$$= \begin{bmatrix} \hat{s}_{00}^{y-} & \hat{s}_{01}^{y-} & \cdots & \hat{s}_{0(M-1)}^{y-} \\ \hat{s}_{10}^{y-} & \hat{s}_{11}^{y-} & \cdots & \hat{s}_{1(M-1)}^{y-} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{s}_{(M-1)0}^{y-} & \hat{s}_{(M-1)1}^{y-} & \cdots & \hat{s}_{(M-1)(M-1)}^{y-} \end{bmatrix} \tag{5.53}$$

These are then transformed to the log Mel-filterbank domain as

$$\hat{\boldsymbol{\mu}}_y^l = \left[\hat{\mu}_0^{ly}, \hat{\mu}_1^{ly}, \ldots, \hat{\mu}_{M-1}^{ly}\right]^T \tag{5.54}$$

$$\hat{\mathbf{S}}_y^{l+} = \begin{bmatrix} \hat{s}_{00}^{ly+} & \hat{s}_{01}^{ly+} & \cdots & \hat{s}_{0(M-1)}^{ly+} \\ \hat{s}_{10}^{ly+} & \hat{s}_{11}^{ly+} & \cdots & \hat{s}_{1(M-1)}^{ly+} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{s}_{(M-1)0}^{ly+} & \hat{s}_{(M-1)1}^{ly+} & \cdots & \hat{s}_{(M-1)(M-1)}^{ly+} \end{bmatrix} \tag{5.55}$$

$$\hat{\mathbf{S}}_y^{l-} = \begin{bmatrix} \hat{s}_{00}^{ly-} & \hat{s}_{01}^{ly-} & \cdots & \hat{s}_{0(M-1)}^{ly-} \\ \hat{s}_{10}^{ly-} & \hat{s}_{11}^{ly-} & \cdots & \hat{s}_{1(M-1)}^{ly-} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{s}_{(M-1)0}^{ly-} & \hat{s}_{(M-1)1}^{ly-} & \cdots & \hat{s}_{(M-1)(M-1)}^{ly-} \end{bmatrix} \tag{5.56}$$

where

$$\hat{\mu}_m^{ly} = \log \hat{\mu}_m^y \tag{5.57}$$

$$\hat{s}_{jk}^{ly+} = \log \hat{s}_{jk}^{y+} \tag{5.58}$$

$$\hat{s}_{jk}^{ly-} = \log \hat{s}_{jk}^{y-} \tag{5.59}$$

The difference between $\hat{\boldsymbol{\mu}}_y^l$ and the diagonal vector of $\hat{\mathbf{S}}_y^{l+}$ derives standard deviations of the noisy speech distribution, $\hat{\boldsymbol{\sigma}}_y^{l+}$, as

$$\hat{\boldsymbol{\sigma}}_y^{l+} = \left[\hat{\sigma}_0^{ly+}, \hat{\sigma}_1^{ly+}, \ldots, \hat{\sigma}_{M-1}^{ly+}\right]^T \tag{5.60}$$

where

$$\hat{\sigma}_m^{ly+} = \hat{s}_{mm}^{ly+} - \hat{\mu}_m^{ly} \tag{5.61}$$

Similarly, the difference between $\hat{\boldsymbol{\mu}}_y^l$ and the diagonal vector of $\hat{\mathbf{S}}_y^{l-}$ also derives standard

deviations of the noisy speech distribution, $\hat{\boldsymbol{\sigma}}_y^{l-}$, as

$$\hat{\boldsymbol{\sigma}}_y^{l-} = \left[\hat{\sigma}_0^{ly-}, \hat{\sigma}_1^{ly-}, \ldots, \hat{\sigma}_{M-1}^{ly-}\right]^T \tag{5.62}$$

where

$$\hat{\sigma}_m^{ly-} = \hat{\mu}_m^{ly} - \hat{s}_{mm}^{ly-} \tag{5.63}$$

Covariance matrices of the noisy speech distribution, $\hat{\boldsymbol{\Sigma}}_y^l$, are then derived, using the average of $\hat{\boldsymbol{\sigma}}_y^{l+}$ and $\hat{\boldsymbol{\sigma}}_y^{l-}$ as

$$\hat{\boldsymbol{\Sigma}}_y^l = \hat{\boldsymbol{\sigma}}_y^l (\hat{\boldsymbol{\sigma}}_y^l)^T \tag{5.64}$$

where

$$\hat{\boldsymbol{\sigma}}_y^l = \left[\hat{\sigma}_0^{ly}, \hat{\sigma}_1^{ly}, \ldots, \hat{\sigma}_{M-1}^{ly}\right]^T \tag{5.65}$$

$$\hat{\sigma}_m^{ly} = \frac{1}{2}\left(\hat{\sigma}_m^{ly+} + \hat{\sigma}_m^{ly-}\right) \tag{5.66}$$

Finally, DCTs are applied to $\hat{\boldsymbol{\mu}}_y^l$ and $\hat{\boldsymbol{\Sigma}}_y^l$ in order to obtain the noise-adapted HMMs in the MFCC domain as

$$\hat{\boldsymbol{\mu}}_y = \mathbf{C}\hat{\boldsymbol{\mu}}_y^l \tag{5.67}$$

$$\hat{\boldsymbol{\Sigma}}_y = \mathbf{C}\hat{\boldsymbol{\Sigma}}_y^l \mathbf{C}^T \tag{5.68}$$

Parallel model combination using either the distribution mapping or the unscented transform can derive the noise-adapted HMMs as an online process by utilising clean speech HMMs and an estimate of the noise power spectrum without *a priori* knowledge as shown above. The only difference between the clean and noise-adapted HMMs is the probability distribution of observation vectors within each state, therefore, the state transition probabilities and the Gaussian mixture weights are unchanged. Thus, parallel model combination is expected to improve the decoding accuracy with less errors in temporal state allocation as shown in Figure 5.1 unlike noise-matched HMMs. Simultaneously, it is also effective to improve the performance of the HMM-based speech synthesis process.

5.3 Experimental Results and Analysis

To evaluate the effectiveness of speech adaptation this section examines HMM-based speech enhancement with noise-adapted HMMs. Experiments use speech from four speakers in the GRID database, two males and two females, which is downsampled to 8 kHz. From the 1000 utterances from each speaker, 800 are used for training and the remainder are for testing. Tests are carried out in white noise and babble noise at SNRs from -5 dB to 10 dB. In each experiment, the set of HMMs, Λ, is trained on an observation sequence, $\mathbf{O}$, that are extracted from clean speech. These are then adapted to model noisy speech in the decoding process using parallel model combination. The noise-adapted HMMs are exploited in the decoding process while the original clean speech HMMs are utilised in the synthesis process.

5.3.1 Feature Vectors

The feature vector is formed as a combination of the MFCCs, the log-Mel-filterbank aperiodicity (AP) coefficients and the fundamental frequency with the velocity and acceleration derivatives as shown in Figure 4.20. Different configurations of the MFCC coefficients are examined while the number of log-Mel-filterbank AP coefficients is fixed at 40, and Table 5.1 shows the feature vector configurations examined in the following experiments.

Config.	Mel-FBK	MFCCs	AP Coefs	Derivatives	Frame Shift
MFCC16-8	16	8	40	Δ & Δ^2	5 ms & 1 ms
MFCC23-23	23	23			

Table 5.1: Configurations of the acoustic features.

MFCC16-8 represents the case of using the truncation of high order coefficients, which correspond to high frequency cosine bases, that has given the best decoding performance in the previous experiments as shown in Figure 4.9. Alternatively, MFCC23-23 represents the case of no truncation applied, that has shown the best speech synthesis performance in a noiseless condition in the previous experiments as shown in Table 4.11.

Discrete-time speech is first divided into 25 ms-length frames by applying a Hamming window in which the frame shift is set equal to 5 ms and 1 ms. A 1024-point STFT is then

applied to obtain the power spectrum and AP measure which are then input into a Mel-filterbank. For MFCC calculation, the number of Mel-filterbank channels is set equal to 16 or 23 according to the test configurations in Table 5.1, and to 40 for the AP measure. A logarithm is then applied to the filterbank energies followed by a discrete cosine transform to produce 8-D or 23-D MFCC vectors, $\mathbf{x}_i$, and a 40-D aperiodicity vector, $\mathbf{a}_i$ at the i-th frame. A fundamental frequency contour, f_{0i}, is estimated with PEFAC followed by a logarithm to produce $\log f_{0i}$. These three components are assigned into a sequence of the augmented feature vectors, $\mathbf{O}$, along with their velocity derivatives, $\Delta\mathbf{x}_i$, $\Delta\mathbf{a}_i$, and $\Delta \log f_{0i}$, and acceleration derivatives, $\Delta^2\mathbf{x}_i$, $\Delta^2\mathbf{a}_i$, and $\Delta^2 \log f_{0i}$, as illustrated in Figure 4.20. The calculation of $\Delta\mathbf{x}_i$ and $\Delta^2\mathbf{x}_i$ follows Equations (4.109) and (4.110), and $\Delta\mathbf{a}_i$, $\Delta^2\mathbf{a}_i$, $\Delta \log f_{0i}$ and $\Delta^2 \log f_{0i}$ are also calculated in the same manner.

5.3.2 HMM training

CD-triphone HMMs with no language model applied are used in the experiments. In the case of 5 ms-frame shift, the model configurations examined are listed in Table 5.2.

Configurations	Frame Shift	Acoustic Feature	# States	# HMMs	LM
TRI_N/8 TRI_N/23	5 ms	MFCC16-8 MFCC23-23	12	150-250	NO

Table 5.2: Model configurations for the tests with feature vectors framed at 5 ms interval.

TRI_N/8, employs CD-triphone HMMs trained with a sequence of the feature vectors configured as MFCC16-8 whereas the other, TRI_N/23, uses CD-triphone HMMs which are based on the feature vectors formed as MFCC23-23. Each model in the set of the CD-triphone HMMs consists of 12 states and possible model sequences are constrained only by triphone context and no language model is applied. The training process first models around 700 CD-triphone HMMs for each speaker. Tree-based clustering is then applied state-by-state to reduce the number of models to the range between 150 and 250 for each speaker with the MDL criterion [94].

In the case of the tests with the feature vectors framed at 1 ms interval, HMMs of each configuration comprise 24 states as shown in Table 5.3.

Configurations	Frame Shift	Acoustic Feature	# States	# HMMs	LM
TRI_N/8 TRI_N/23	1 ms	MFCC16-8 MFCC23-23	24	150-250	NO

Table 5.3: Model configurations for the tests with feature vectors framed at 1 ms interval.

5.3.3 HMM Adaptation

At the begining of the decoding stage, the trained HMMs, $\mathbf{\Lambda}$, are modified to model noisy test speech using parallel model combination. The noise power spectrum in the noisy input speech at the i-th frame, $|\hat{D}_i(f)|^2$, is first estimated from the *a priori* SNR derived by the decision-directed method with the bias reducing algorithm in Equation (2.41), and it is then transformed to the linear Mel-filterbank coefficient vector, $\hat{\mathbf{D}}_i^{fb} = \left[\hat{D}_i^{fb}(0), \hat{D}_i^{fb}(1), \ldots, \hat{D}_i^{fb}(M-1)\right]^T$, where M is the number of the filterbank channels, in order to model the noise as a single state Gaussian distribution.

To apply parallel model combination, distribution parameters, i.e. mean vectors and covariance matrices, of only static MFCC features are extracted from the clean HMMs and then they are transformed to the linear Mel-filterbank domain, in which parallel model combination is applied by Equations (5.48) - (5.53), using the unscented transform. A preliminary experiment showed that the results using the log-normal transform are similar to the case of using the unscented transform. Therefore, this experiment uses only the unscented transform approach rather than doubling the results by also showing those for the log-normal transform. After the parameter sets of the noisy speech models are obtained by parallel model combination in the linear Mel-filterbank domain, they are then transformed back to the MFCC domain in order to be exploited as a set of the noise-adapted HMMs, $\hat{\mathbf{\Lambda}}$ at the decoding stage.

5.3.4 HMM Decoding

The HMM decoding process uses $\hat{\mathbf{\Lambda}}$ and a sequence of the static MFCC vectors extracted from $\mathbf{O}$ in order to obtain the most likely model sequence including their state sequences using the Viterbi algorithm discussed in Section 4.2.2.

5.3.5 Decoding Results

Figure 5.4 shows the results of HMM decoding with the noise-adapted HMMs, $\hat{\mathbf{\Lambda}}$, compared with the cases of using clean HMMs, $\mathbf{\Lambda}$, and noise-matched HMMs $\mathbf{\Lambda}_y$. Subplots

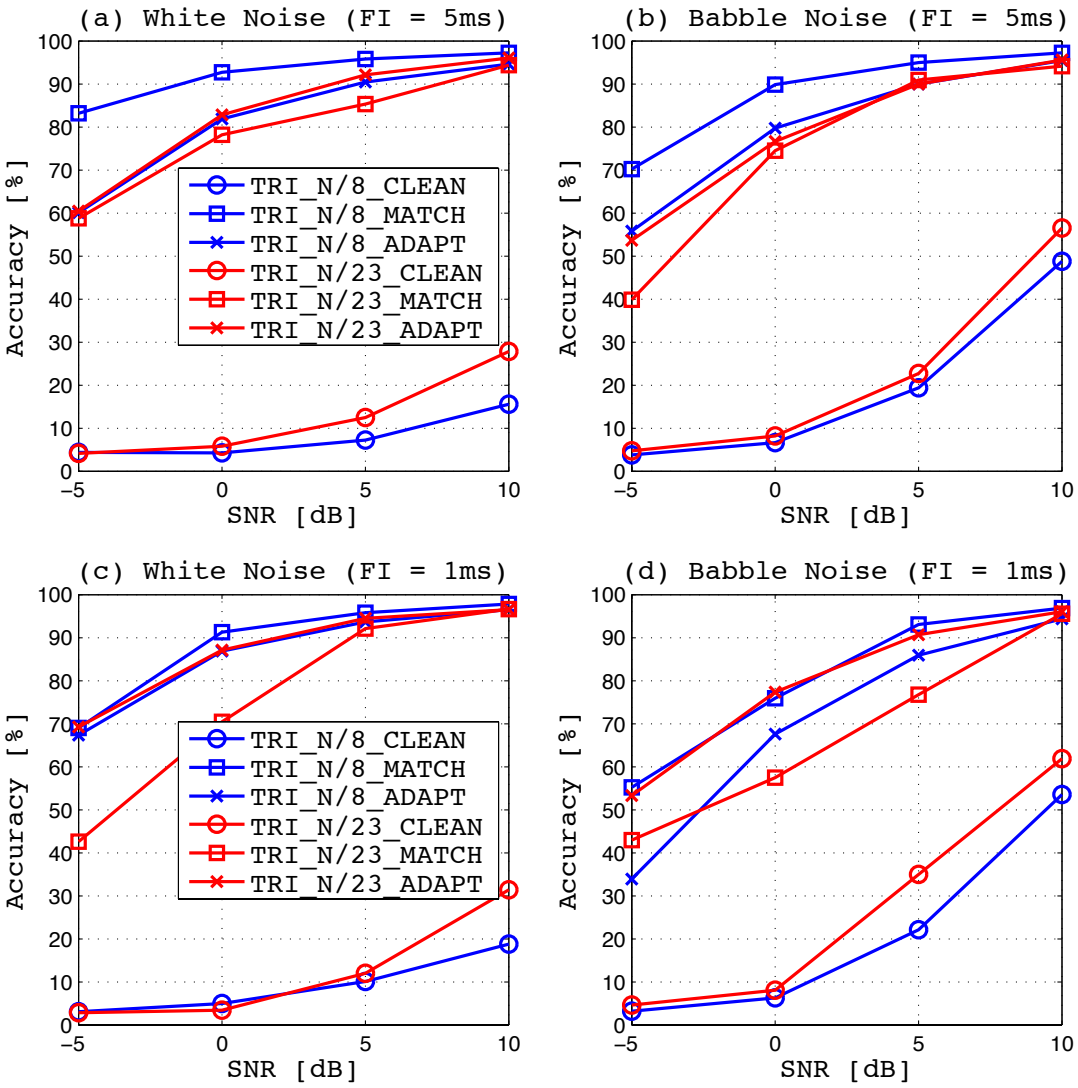

Figure 5.4: The results in decoding accuracy. a) and b) show the results in white noise and babble noise with the feature vectors framed at 5 ms interval. c) and d) show the results in white noise and babble noise with the feature vectors framed at 1 ms interval.

(a) and (b) show decoding accuracy with the feature vectors framed at 5 ms interval in white noise and babble noise while (c) and (d) show the results with the feature vectors framed at 1 ms interval. Noise-matched HMMs with the truncation (TRI_N/8_MATCH) still show the best decoding accuracy, but noise-adapted HMMs (TRI_N/8_ADAPT and

TRI_N/23_ADAPT) also show good noise robustness for practical use. Interestingly, noise-adapted HMMs with no truncation (TRI_N/23_ADAPT) have better noise robustness than noise-adapted HMMs with the truncation (TRI_N/8_ADAPT) in contrast to the case of noise-matched HMMs in which TRI_N/8_MATCH shows always higher decoding accuracy than TRI_N/23_MATCH. Synthesised speech by using clean HMMs with no MFCC truncation applied has shown higher PESQ scores than the case of using clean HMMs with the MFCC truncation in the preceding tests as shown in Table 4.11, but low noise robustness of TRI_N/23_MATCH in the decoding process has brought a crucial disadvantage in HMM-based speech enhancement as shown in Figure 4.28. However, TRI_N/23_ADAPT reduces that disadvantage and has the prospect of giving better noise robustness than TRI_N/23_MATCH to enhanced speech.

TRI_N/23_ADAPT does not bring deterioration in decoding accuracy even when the frame shift of the feature vectors change to 1 ms from 5 ms as opposed to the case of noise-matched HMMs. This also brings potential to achieve higher speech quality in enhanced speech as compared with the case of using feature vectors framed at 5 ms intervals.

5.3.6 HMM Synthesis and Speech Reconstruction

After obtaining model and state sequence of test speech by decoding, the clean HMMs can now synthesise the speech features of the clean speech according to this model and state sequence. The synthesised speech features, i.e. MFCC vectors, AP vectors and $\log f_0$, are then transformed to the spectral envelopes and the aperiodicity measure in the time-frequency domain and the fundamental frequency contour in order to reconstruct the enhanced speech with STRAIGHT following the same procedure as Section 4.4.2.

5.3.6.1 Speech Quality

Figure 5.5 shows the objective speech quality of resultant enhanced speech in terms of PESQ comparing with the enhanced speech using noise-matched HMMs in the decoding process, the log MMSE method and no noise compensation (NNC). Subplots (a) and (b) show PESQ scores of each HMM configuration with the feature vectors framed at 5 ms intervals in white noise and babble noise while (c) and (d) show the results with

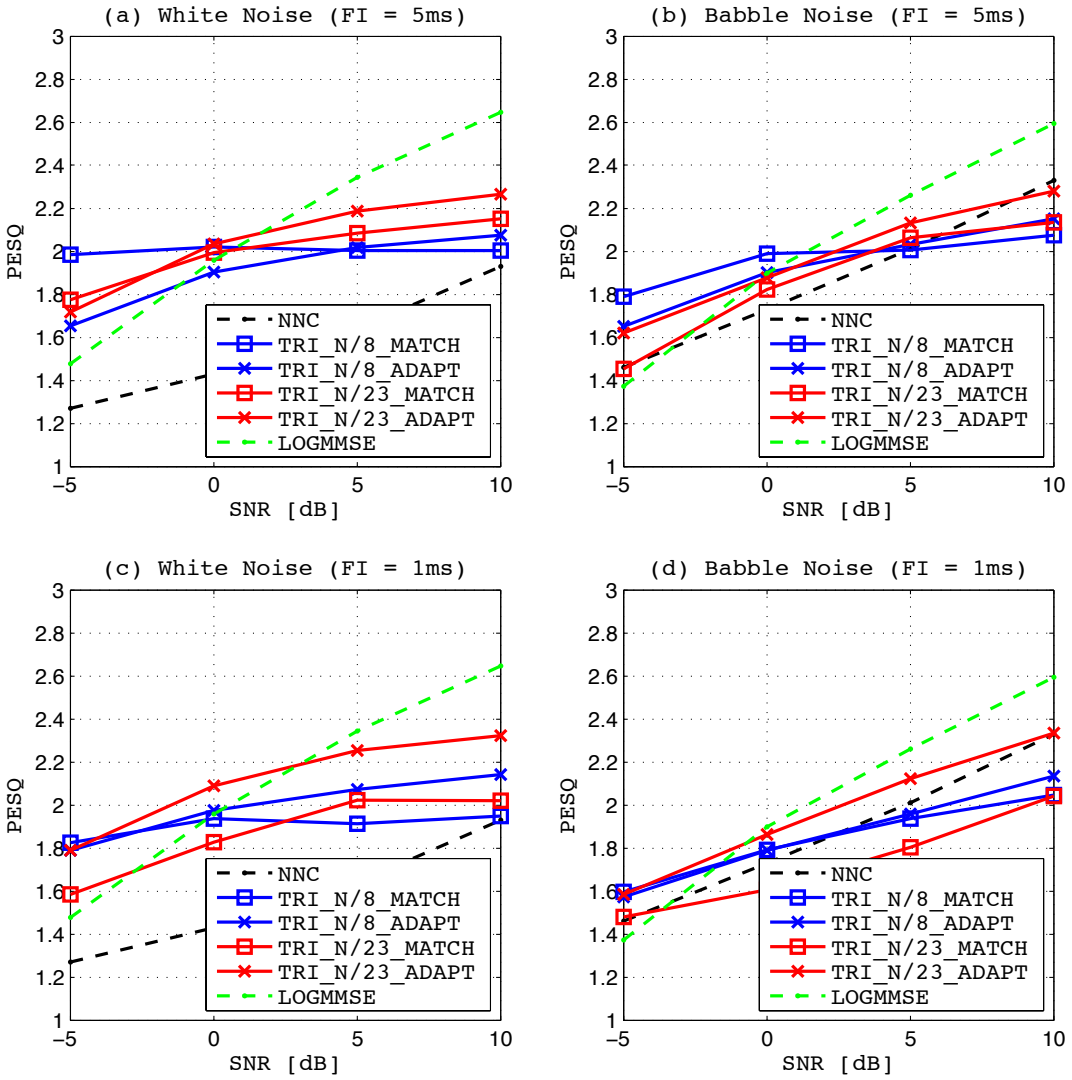

Figure 5.5: Objective speech quality of the enhanced speech in terms of PESQ. a) and b) show the results in white noise and babble noise with the feature vectors framed at 5 ms interval while c) and d) show the results using the feature vectors framed at 1 ms interval.

the feature vectors framed at 1 ms intervals. At an SNR of 10 dB with either noise type, PESQ scores of noise-adapted HMMs are always higher than noise matched HMMs although their decoding accuracy at that SNR are similarly high. This may be attributed to the fact that the model and state sequences derived by the noise-adapted HMMs, $\hat{\mathbf{\Lambda}}$, match the state allocation of the clean HMMs, $\mathbf{\Lambda}$, which are used in the speech synthesis process, whereas the sequences derived by the noise-matched HMMs, $\mathbf{\Lambda}_y$, bring time warping in reconstructed speech because of inconsistencies in state allocation between $\mathbf{\Lambda}_y$ and $\mathbf{\Lambda}$. Consequently, using $\hat{\mathbf{\Lambda}}$ raises the upper limit of PESQ scores, specifically, the PESQ scores of enhanced speech with TRI_N/23_ADAPT are competitive with the log MMSE method even at high SNRs of around 5 dB, especially in the case of using feature vectors framed at 1 ms interval.

Alternatively, TRI_N/23_ADAPT is not as robust as noise-matched HMMs with the MFCC truncation using 5 ms frame shifted feature vectors (TRI_N/8_MATCH) to noise.

Overall, TRI_N/23_ADAPT, which represents a practical system with no *a priori* knowledge about the noise, shows significant improvement in PESQ at SNRs of 0 dB and below in both white noise and babble noise.

5.3.6.2 Speech Intelligibility

Figure 5.6 shows the objective speech intelligibility of resultant enhanced speech in terms of NCM compared with the enhanced speech using noise-matched HMMs, $\mathbf{\Lambda}_y$, in the decoding process and the log MMSE method. Subplots (a) and (b) show NCM score of each HMM configuration with the feature vectors framed at 5 ms intervals in white noise and babble noise while (c) and (d) show the results with the feature vectors framed at 1 ms intervals. Noise-adapted HMMs, $\hat{\mathbf{\Lambda}}$, with no MFCC truncation applied (TRI_N/23_ADAPT) always show the best score in the HMM configurations over the test conditions, and $\hat{\mathbf{\Lambda}}$ with the MFCC truncation (TRI_N/8_ADAPT) also show better NCM scores than $\mathbf{\Lambda}_y$ in spite of lower decoding accuracy in the decoding process. This is again attributed to the match of state allocation between $\hat{\mathbf{\Lambda}}$ and $\mathbf{\Lambda}$.

NCM scores for the noise adapted HMM-based methods, i.e. TRI_N/8_ADAPT and TRI_N/23_ADAPT, are remarkably flat across SNRs between -5 dB and 10 dB and shows significant improvement at low SNRs between -5 dB and 0 dB.

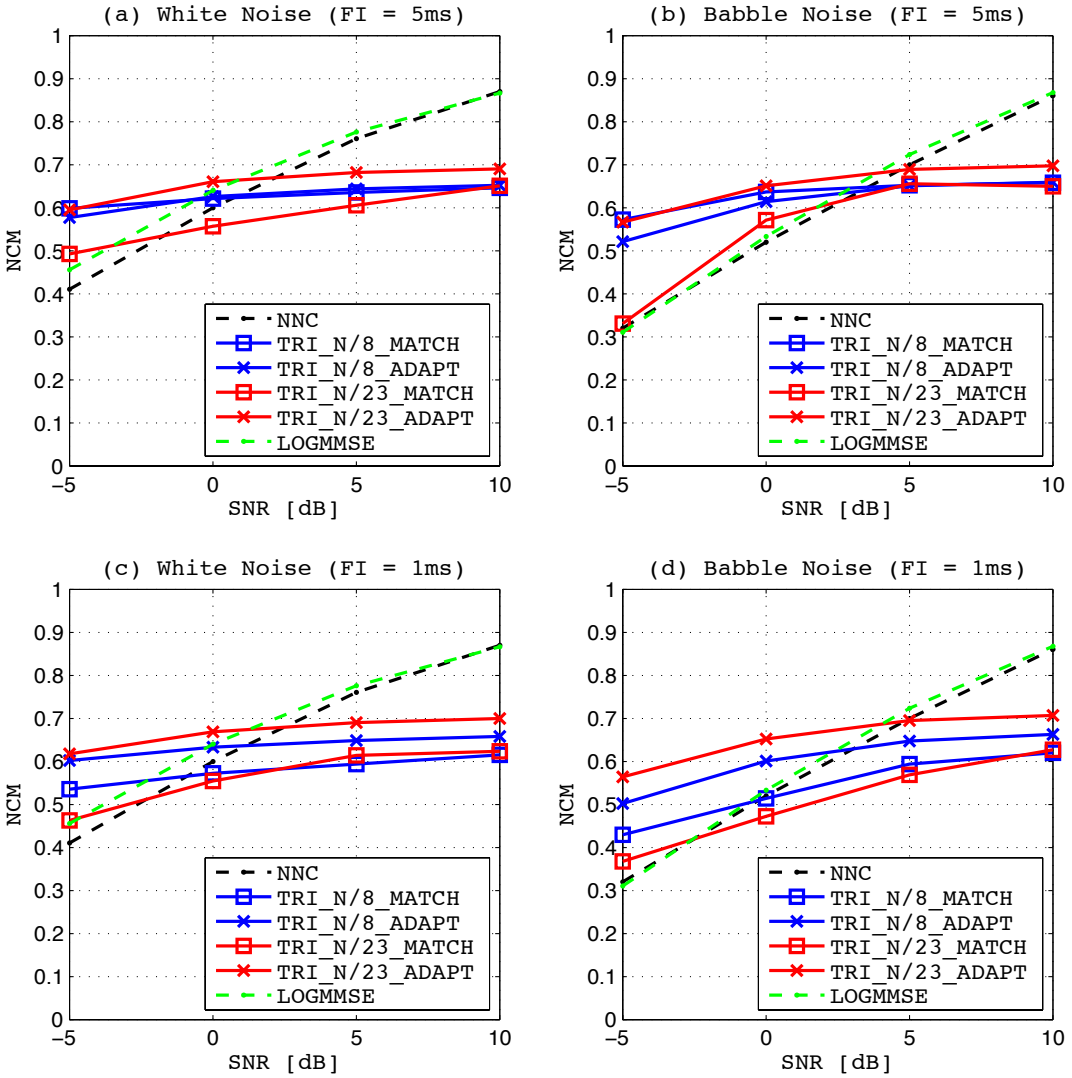

Figure 5.6: Objective speech intelligibility of the enhanced speech in terms of NCM. a) and b) show the results in white noise and babble noise with the feature vectors framed at 5 ms interval while c) and d) show the results using the feature vectors framed at 1 ms interval.

5.4 Conclusion of the Chapter

HMM-based speech enhancement using noise-matched HMMs, which is evaluated in the previous chapter, needs *a priori* noise information in the training process and creates inconsistency in state allocation between clean HMMs and noise-matched HMMs, and thus it cannot be used for practical applications of HMM-based speech enhancement. This chapter first discussed the theory of HMM adaptation to model noisy speech using parallel model combination to address the problems above. In parallel model combination, the mismatch function between clean speech and noisy speech is determined in the linear Mel-filterbank domain while HMMs had been modelled in the MFCC domain. Therefore, the non-linearity between these domains needs to be resolved and the distribution mapping between Gaussian and log-normal, and the unscented transform were discussed to tackle this problem.

Experiments then evaluated the performance of HMM-based speech enhancement with noise-adapted HMMs compared to the methods using noise-matched HMMs and log MMSE, which represents the conventional filtering methods, in terms of PESQ and NCM scores. The experimental analysis showed that HMM adaptation to model noisy speech with parallel model combination seems effective in obtaining state sequences which match the clean HMMs because the upper limits of PESQ and NCM scores were improved by that. In terms of noise robustness, however, the effectiveness of the HMM adaptation was limited to the configuration with no MFCC truncation because of difficulty in correctly estimating the noise.

In summary, applying the HMM adaptation to noisy speech using CD-triphone HMMs with no MFCC truncation (TRI_N/23_ADAPT) improved PESQ and NCM at high SNRs, e.g. 5 dB and above. Although the PESQ and NCM scores at SNRs below 0 dB of this configuration were lower than the configuration using noise-matched HMMs with MFCC truncation (TRI_N/8_MATCH), it still shows significant improvement as compared with log MMSE at those SNRs and the benefit of the use of HMM adaptation which eliminates the necessity of *a priori* knowledge about noise is advantageous.

Chapter 6

Improvement to Hidden Markov Model-Based Speech Enhancement

HMM-based speech enhancement reconstructs noise-free speech from the input noisy speech. The output speech, however, still has problems in terms of speech quality and intelligibility, which are brought by the characteristics of the techniques in HMM-based speech enhancement. This chapter first discusses problems attributed to the HMM decoding process and then the problems in the HMM synthesis process. A series of countermeasures are then proposed and experiments in each section examine these against the problems and effectiveness of those methods are evaluated prior to the final evaluation of the proposed method in the next chapter.

6.1 Introduction

The experimental results in the previous chapter, i.e. Figure 5.5 and 5.6, show that the proposed HMM-based speech enhancement under the real-world configurations can achieve better performance at low SNRs such as 0 dB and below than the log MMSE method representing the conventional filtering approaches. However, the PESQ and NCM scores gradually decrease at lower SNRs, though ideally they should be kept at the same level as at higher SNRs. This is caused by the decoding errors at the HMM

decoding stage, and thus, a method to detect erroneous frames from the result of HMM decoding needs to be applied so that the the speech segments which consist of erroneous frames in the output speech can be identified and then those segments replaced with the speech enhanced with a filtering method because the filtering method will produce more representative speech than an incorrectly decoded speech.

Another problem in the proposed method is also shown in the experimental results because the PESQ and NCM scores of the output speech at high SNRs are lower than the log MMSE method. This is attributed to the upper limit of PESQ in HMM-based speech synthesis shown in Table 4.15. Therefore, methods to improve the upper limit of HMM-based speech synthesis also need to be discussed to raise the performance of the proposed method over the noise conditions.

The remainder of this chapter first discusses a novel confidence measuring method to identify erroneous frames resulting from decoding errors and then evaluates the enhanced speech in which the speech segments comprising the low-confidence frames are replaced with the speech enhanced with log MMSE. Secondly, methods to refine HMM-synthesised speech are then discussed and evaluated prior to the conclusion of the chapter.

6.2 Confidence Measuring and Compensation for Decoding Errors

The proposed method of mitigating HMM decoding errors is a two stage process of first identifying errors and secondly applying compensation.

In the field of speech recognition, confidence measures have been utilised to evaluate the recognition results and it is known that accurate confidence measures bring practical application benefits to detect out-of-vocabulary words, non-speech noises, potential recognition mistakes and so on. [110] categorises various type of confidence measures into a combination of predictor features, e.g. [111], posterior probability and utterance verification. However, it reports that the overall performance of them remains fairly poor and it limits their applications. Therefore, a new method for confidence measuring that exploits a technique of HMM-based speech synthesis is first explored in this section and then a method to compensate for the erroneous frames detected by the confidence

measure is discussed in order to tackle the problem attributed to HMM decoding errors.

6.2.1 Overview of the Confidence Measure Estimation

Figure 6.1 illustrates the overview of the method to detect unreliable frames using a confidence measure. A sequence of MFCC vectors of the noisy speech, $\mathbf{C}_y$, is first extracted

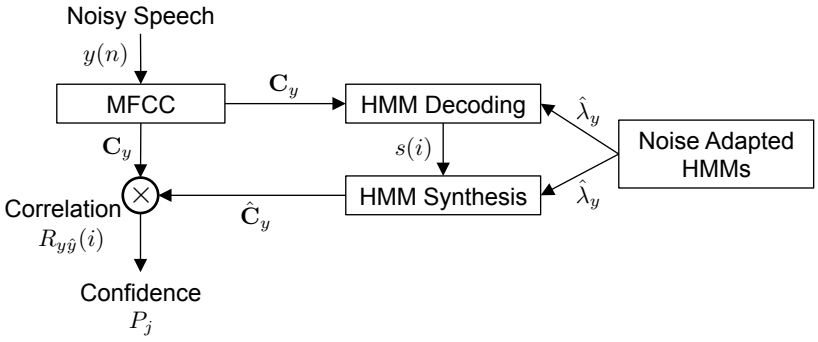

Figure 6.1: The overview of the proposed method for confidence measuring.

as

$$\mathbf{C}_y = \left[(\mathbf{c}_y^0)^T, (\mathbf{c}_y^1)^T, \ldots, (\mathbf{c}_y^{I-1})^T\right]^T \tag{6.1}$$

$$\mathbf{c}_y^i = [c_i(0), c_i(1), \ldots, c_i(M-1)]^T \tag{6.2}$$

where $c_i(m)$ denotes the m-th coefficient in the static MFCC vector of the noisy speech at frame i (i.e., $\mathbf{c}_y^i$). $\mathbf{C}_y$ is then decoded into a state sequence, $s(i)$, using the noise adapted HMMs, $\hat{\lambda}_y$. Although a set of clean HMMs, λ_x, is used for the speech synthesis process in the proposed HMM-based speech enhancement, $\hat{\lambda}_y$ is then used to synthesise an MFCC vector sequence of HMM-based noisy speech, $\hat{\mathbf{C}}_y$, from $s(i)$, and it is represented as.

$$\hat{\mathbf{C}}_y = \left[(\hat{\mathbf{c}}_y^0)^T, (\hat{\mathbf{c}}_y^1)^T, \ldots, (\hat{\mathbf{c}}_y^{I-1})^T\right]^T \tag{6.3}$$

$$\hat{\mathbf{c}}_y^i = [\hat{c}_i(0), \hat{c}_i(1), \ldots, \hat{c}_i(M-1)]^T \tag{6.4}$$

where $\hat{c}_i(m)$ denotes the m-th coefficient in the static MFCC vector of the HMM-based noisy speech at frame i (i.e., $\hat{\mathbf{c}}_y^i$).

The proposed confidence measure is based on a hypothesis that the decoding result

at frame, i, is reliable if $\hat{\mathbf{c}}_y^i$ is enough close to $\mathbf{c}_y^i$ while the frame is unreliable if $\hat{\mathbf{c}}_y^i$ is far from $\mathbf{c}_y^i$. Therefore, frame-by-frame confidence is determined by correlation coefficient, $R_{y\hat{y}}(i)$, as follows.

$$R_{y\hat{y}}(i) = \frac{\mathcal{E}\left[\left(c_i(m) - \mu_i\right)\left(\hat{c}_i(m) - \hat{\mu}_i\right)\right]}{\sqrt{\mathcal{E}\left[\left(c_i(m) - \mu_i\right)^2\right]\mathcal{E}\left[\left(\hat{c}_i(m) - \hat{\mu}_i\right)^2\right]}} \tag{6.5}$$

where μ_i and $\hat{\mu}_i$ are the mean values of $\mathbf{c}_y^i$ and $\hat{\mathbf{c}}_y^i$ respectively.

$R_{y\hat{y}}(i)$ determined by Equation (6.5) tends to be close to 1 for all the frames because the dynamic range of MFCC is constrained by log operation. Therefore, correlation coefficient, $R'_{y\hat{y}}(i)$, determined as follows is more effective to measure the frame-by-frame confidence.

$$\mathbf{b}_y^i = \mathbf{C}\exp\left(\mathbf{C}^{-1}\mathbf{c}_y^i\right) \tag{6.6}$$

$$= \left[b_i(0), b_i(1), \ldots, b_i(M-1)\right]^T \tag{6.7}$$

$$\hat{\mathbf{b}}_y^i = \mathbf{C}\exp\left(\mathbf{C}^{-1}\hat{\mathbf{c}}_y^i\right) \tag{6.8}$$

$$= \left[\hat{b}_i(0), \hat{b}_i(1), \ldots, \hat{b}_i(M-1)\right]^T \tag{6.9}$$

$$R'_{y\hat{y}}(i) = \frac{\mathcal{E}\left[\left(b_i(m) - \mu_i^b\right)\left(\hat{b}_i(m) - \hat{\mu}_i^b\right)\right]}{\sqrt{\mathcal{E}\left[\left(b_i(m) - \mu_i^b\right)^2\right]\mathcal{E}\left[\left(\hat{b}_i(m) - \hat{\mu}_i^b\right)^2\right]}} \tag{6.10}$$

where μ_i^b and $\hat{\mu}_i^b$ are the mean values of $\mathbf{b}_y^i$ and $\hat{\mathbf{b}}_y^i$, and $\mathbf{C}$ and $\mathbf{C}^{-1}$ denote the DCT and IDCT matrices. Now confidence of each frame is derived from $R'_{y\hat{y}}(i)$.

6.2.2 Compensation of the Unreliable Samples

After the frame-by-frame confidence measure of the decoding result is obtained, the phoneme-by-phoneme confidence measure, P_j, is then derived by taking the mean over the frames within the j-th phoneme in the utterance using the phoneme boundary from the HMM decoding as

$$P_j = \begin{cases} 1 & \frac{1}{i_{j+1}-i_j}\sum_{k=i_j}^{i_{j+1}} R'_{y\hat{y}}(k) \geqslant \beta \\ 0 & \text{otherwise} \end{cases} \tag{6.11}$$

where i_j represents the start frame of the j-th phoneme and β is the threshold between reliable (i.e., high confidence) and unreliable (i.e., low-confidence).

The time domain samples corresponding to the phonemes marked as unreliable possibly constitute wrong phonemes in the enhanced speech and thus, these samples are replaced with the corresponding samples in the speech enhanced with a filtering method such as log MMSE as illustrated in Figure 6.2. This operation may be effective to avoid

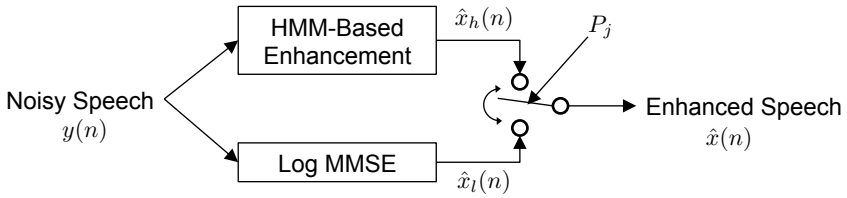

Figure 6.2: Compensation of the samples in the output speech corresponding to unreliable phonemes with the corresponding samples in log MMSE.

outputting wrong speech, but the replaced samples include residual and musical noise because of the characteristics of speech enhancement using filtering approach and the enhanced speech cannot be noise-free any more at this moment. This means trade-off between decoding errors and increased background noise, i.e. trade-off between speech intelligibility and quality. Therefore, the threshold, β, needs to be determined carefully.

After the reliable/unreliable classification is applied to each phoneme, the corresponding frames can be categorised into four conditions depending on an evaluation of the decision as shown in Table 6.1. If the threshold of the decision is too high, the ratio

	Low Confidence	High Confidence
Frames during Correctly Decoded Phoneme	False Positive	True Negative
Frames during Wrongly Decoded Phoneme	True Positive	False Negative

Table 6.1: Evaluation of the decision.

of "False Positive" increases as more phonemes become classed as erroneous. This results in the enhanced speech being contaminated with unnecessary residual and musical noise at low SNRs. However, if the threshold is too low, the ratio of "False Negative" is as high as the original decoding result and the output includes wrong speech at low SNRs.

6.2.3 Experimental Results

The confidence measure and the replacement of low confidence segments discussed above are examined by applying them to the results of the experiments in Section 5.3. Experiments use speech from four speakers in the GRID database, two males and two females, which is downsampled to 8 kHz. From the 1000 utterances from each speaker, 800 are used for training and the remainder are for testing. Tests are carried out in white noise and babble noise at SNRs from -5 dB to 10 dB. In each experiment, the feature vectors of speech, **O**, are configured as MFCC23-23 in Table 5.1 with a 25 ms Hamming window whose frame shift is set to 5 ms and extracted from the training set of clean speech. A set of 12 state CD-triphone HMMs, TRI_N/23, determined in Table 5.2 is trained on **O**. These are then adapted to model noisy speech in the decoding process using parallel model combination.

In the decoding process, only the static MFCC components in the noise adapted HMMs are used to obtain state sequence, $s(i)$, while the static and dynamic MFCC components in the noise adapted HMMs are used to synthesise noisy MFCC vectors which are compared with the MFCC vectors of original noisy speech for the confidence measure. The dynamic components in the noise adapted HMMs are unchanged from the clean HMMs.

6.2.3.1 Accuracy of Confidence Measure and Classification

Phoneme-by-phoneme confidence measure, P_j, is derived from the frame-by-frame confidence measure, $R'_{y\hat{y}}(i)$, calculated by Equation (6.10) and then the following performance measures are calculated at different thresholds to evaluate the effectiveness of the confidence measure.

$$\text{Correct Frame Rate} = \frac{(\text{Number of True Positive}) + (\text{Number of True Negative})}{\text{Number of Frames}}$$

$$\text{False Positive Rate} = \frac{\text{Number of False Positive}}{\text{Number of Frames}}$$

$$\text{False Negative Rate} = \frac{\text{Number of False Negative}}{\text{Number of Frames}}$$

Figure 6.3 shows the Correct Frame Rate (CFR) at different SNRs and threshold values. Subplot (a) shows the test results in white noise in which the CFR at SNR of -5 dB

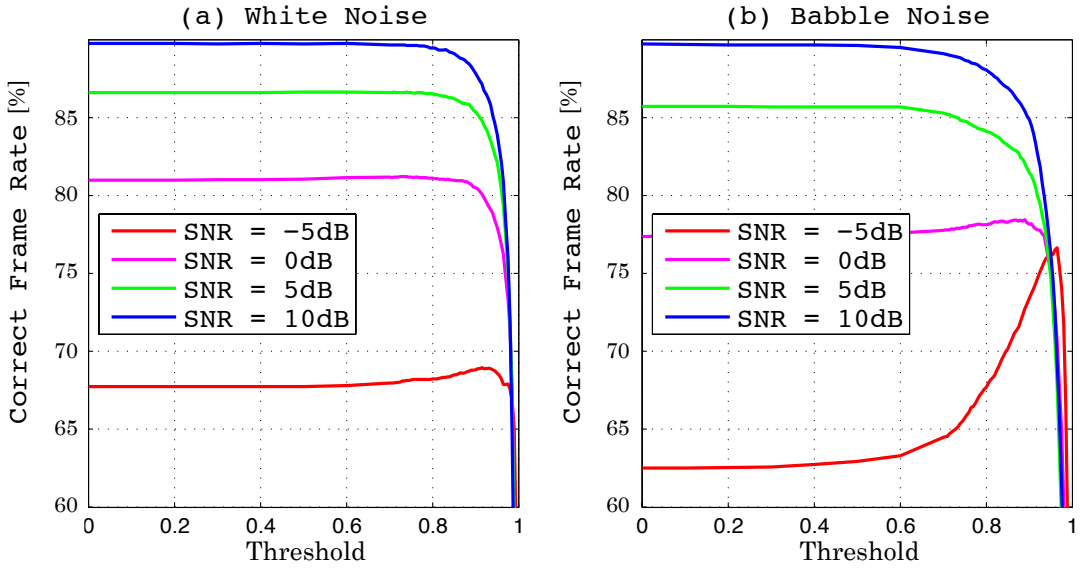

Figure 6.3: Correct frame rate at different thresholds. a) shows the result in white noise while b) is in babble noise.

is increased at the threshold between 0.7 and 0.92 while the CFR at higher SNRs is not improved. Most of erroneous frames at high SNRs such as 10 dB and 5 dB are on the boundaries between the phonemes, which are slightly shifted from the reference time label, and the majority of the frames within the phonemes are correct as Figure 5.4 shows the decoding accuracy at these SNRs amounts more than 90 %. Therefore, these partial erroneous frames in a phoneme are not identified by P_j because of the averaging operation over the phoneme in Equation (6.11), and consequently, the CFR is not improved from the original decoding result (i.e., threshold, $\beta = 0$). However, when SNR is equal to -5 dB, Figure 5.4 shows that 40 % of phonemes are substitution, insertion or deletion errors in which the majority of frames are incorrect and thus, they are detectable by P_j. When the threshold is set more than 0.8 at SNRs of 0 dB and above, the CFR falls exponentially because of a significant increase of the False Positive Rate (FPR). Similarly, the CFR at -5 dB substantially falls at the threshold more than 0.92.

Subplot (b) reports that the CFR at SNR of -5 dB is steeply raised at thresholds between 0.6 and 0.95, and the increase amounts 14 pts. at the peak brought by the

threshold set equal to 0.95. When SNR is equal to 0 dB, the CFR gradually increases while the threshold is less than and equal to 0.9 and then it steeply goes down. When SNR is 5 dB and 10 dB, the CFR keeps flat during the threshold being less than 0.7 and then exponentially falls. These results show that the proposed confidence measure is effective at detecting the decoding errors specifically at SNRs less than 0 dB in babble noise with very little deterioration in other noise conditions by setting the threshold around 0.8. This seems to be attributed to the fact that the decoding errors in the noise conditions of SNR less than 0 dB in babble noise are dominated by deletion errors from the middle to the end of utterances, which are more likely to be discriminated by comparing $\mathbf{b}_y^i$ with $\hat{\mathbf{b}}_y^i$.

Figure 6.4 shows the FPR and the False Negative Rate (FNR) at different threshold values and SNRs in white noise. Subplots (a), (b), (c) and (d) show the results at SNRs of -5 dB, 0 dB, 5 dB and 10 dB respectively and both of the FPR and FNR are kept almost flat during the threshold being less than and equal to 0.8. Therefore, the classification of the frames into reliable and unreliable with the threshold equal to 0.8 seems not to affect the decoding result.

Similarly, Figure 6.5 illustrates the FPR and FNR at different threshold values and SNRs in babble noise. Subplots (a), (b), (c) and (d) show the results at SNRs of -5 dB, 0 dB, 5 dB and 10 dB respectively. In the case of SNR of -5 dB, the FNR decreases by 7 pts. at the threshold set equal 0.8 while the FPR increases by 1 pt. When SNR is 0 dB, both the decrease of the FNR and the increase of the FPR are equal to 2 pts. Alternatively, when SNR is 5 dB and 10 dB, the FNR does not change while the FPR increases by 2 pts. Therefore, the accuracy of the frame classification into reliable and unreliable with the threshold value of 0.8 is poor at SNRs equal to 5 dB and above. In those noise conditions, however, both PESQ and NCM scores of the enhanced speech processed by the filtering methods are higher than the proposed method. Therefore, the sample replacement according to this frame classification seems not to affect the output even at high SNRs.

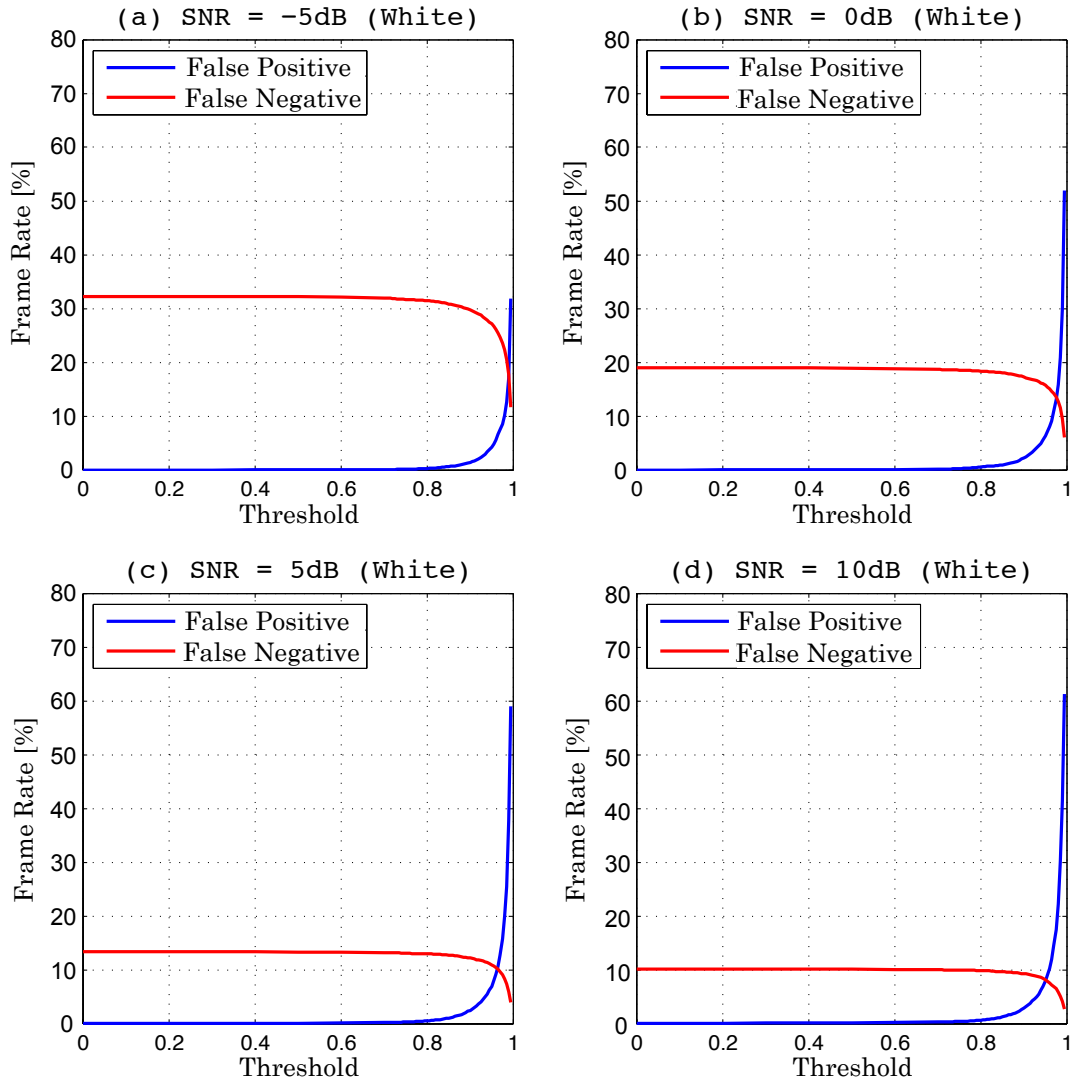

Figure 6.4: False positive rate and false negative rate with different threshold values at SNRs of a) -5 dB, b) 0 dB, c) 5 dB and d) 10 dB in white noise.

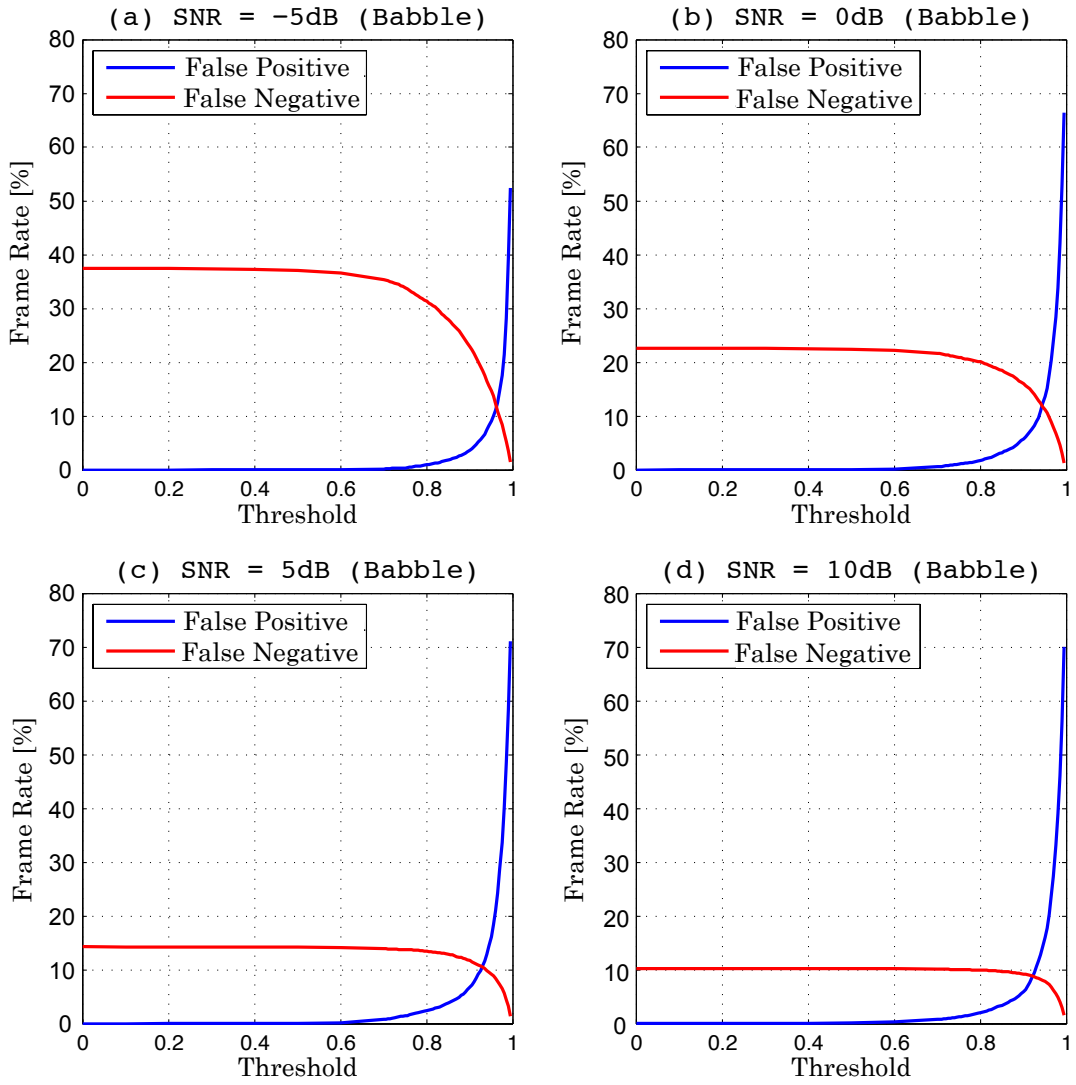

Figure 6.5: False positive rate and false negative rate with different threshold values at SNRs of a) -5 dB, b) 0 dB, c) 5 dB and d) 10 dB in babble noise.

6.2.3.2 Effectiveness of Replacement of the Samples corresponding to unreliable phonemes

After the phoneme classification with the threshold set equal to 0.8, the time-domain speech signal reconstructed with the proposed HMM-based speech enhancement is combined with the time-domain speech enhanced with the log MMSE method according to the result of the phoneme classification, i.e. the time-domain samples in the reconstructed speech corresponding to unreliable phonemes are replaced with the corresponding samples in enhanced speech processed with the log MMSE method.

Figure 6.6 compares the performance of combined speech with HMM-based speech and log MMSE in terms of PESQ and NCM. Subplot (a) compares PESQ scores in white noise. The difference of the performance between HMM-based speech and combined speech is very little because replaced phonemes are not many as it was expected from the results in which both of the CFR, FPR and FNR changed little in Figures 6.3 and 6.4. Subplot (b) compares PESQ scores in babble noise. PESQ score of combined speech at -5 dB is improved by 0.16. This is attributed to the increase of the CFR shown in Figure 6.3 and the decrease of the FNR shown in Figure 6.5. Therefore, the PESQ score seems more sensitive against decoding errors than residual and musical noise at this noise level. The PESQ scores of combined speech at 0 dB and above are not different from HMM-based speech though Figure 6.5 shows the FPR of approximately 2.5 %. Therefore, it seems sample replacement with this FPR does not give significant influence to PESQ score.

Alternatively, considering objective intelligibility, NCM scores of combined speech shown in subplots (c) and (d) are lower than HMM-based speech, and those differences become larger at lower SNRs. Specifically, at -5 dB in babble noise, NCM score of combined speech falls by 0.2 from HMM-based speech though the CFR increases by 5 pts. according to Figure 6.3. These results bring a notion that NCM score is more sensitive against background noise than decoding errors.

To illustrate decoding errors and compensation, Figure 6.7 shows narrowband spectrograms of female speech of "Bin Blue At L Three Again". Subplot (a) shows natural clean speech, Subplot (b) represents noisy speech contaminated with white noise at an SNR of -5 dB and Subplots (c), (d) and (e) show enhanced speech with HMM-based

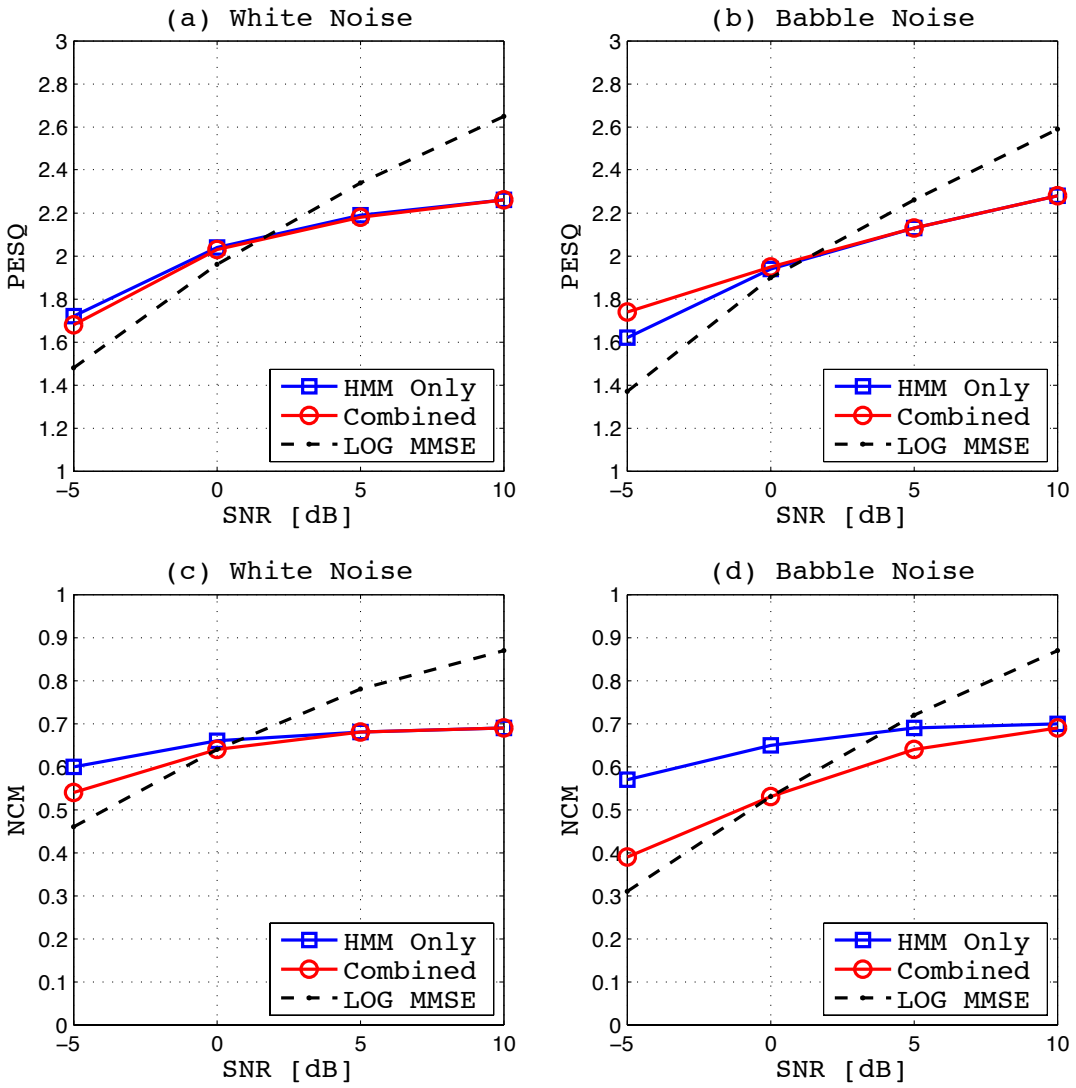

Figure 6.6: Performance of combined speech at different SNRs comparing with HMM-based speech and log MMSE. a) and b) compare PESQ scores at different SNRs in white noise and babble noise while c) and d) compare NCM scores at different SNRs in babble noise.

enhancement, log MMSE and combined speech respectively. This example shows that the combined speech contains residual noise of log MMSE though the HMM-based speech does not contain decoding errors. This is brought by false positive classification of the confidence measure.

Figure 6.8 shows narrowband spectrograms of the same speech as Figure 6.7. However, Subplot (b) shows noisy speech contaminated with babble noise at an SNR of -5 dB and Subplots (c), (d) and (e) show enhanced speech with HMM-based enhancement, log MMSE and combined speech respectively. In this example, the HMM-based speech contains a deletion error and the combined speech replaces the error segment with log MMSE by the true positive classification of the confidence measure. Unnecessary residual noise is, however, added at the tail of the combined speech due to the false positive classification.

In summary, the experiments in this section showed that combining HMM-based speech with log MMSE according to the proposed confidence measure is effective to raise PESQ scores in particular noise conditions. This process, however, decreases NCM scores by intaking residual and musical noise in speech processed with log MMSE.

6.3 Refinement of HMM-Based Speech Synthesis with Global Variance

Quality of HMM-based speech with no decoding errors is the baseline of the performance of the proposed HMM-based speech enhancement. Therefore, it is important to refine the process of HMM-based speech synthesis. Although statistical parametric speech synthesis, including HMM-based speech synthesis, has demonstrated various advantages such as flexibility and small footprint [9,50], the synthesised speech quality is still not as good as the quality of natural speech and unit selection TTS approaches [112]. The deterioration in quality of HMM-based speech are largely attributed to the process of vocoder, acoustic modelling and over-smoothing [9], and acoustic modelling has thoroughly been studied and evaluated in Chapter 4, including the whole-word/sub-word models, the number of states, the number of models clustered by MDL criterion , different configurations of the speech features with dynamic derivatives and framing interval of the speech

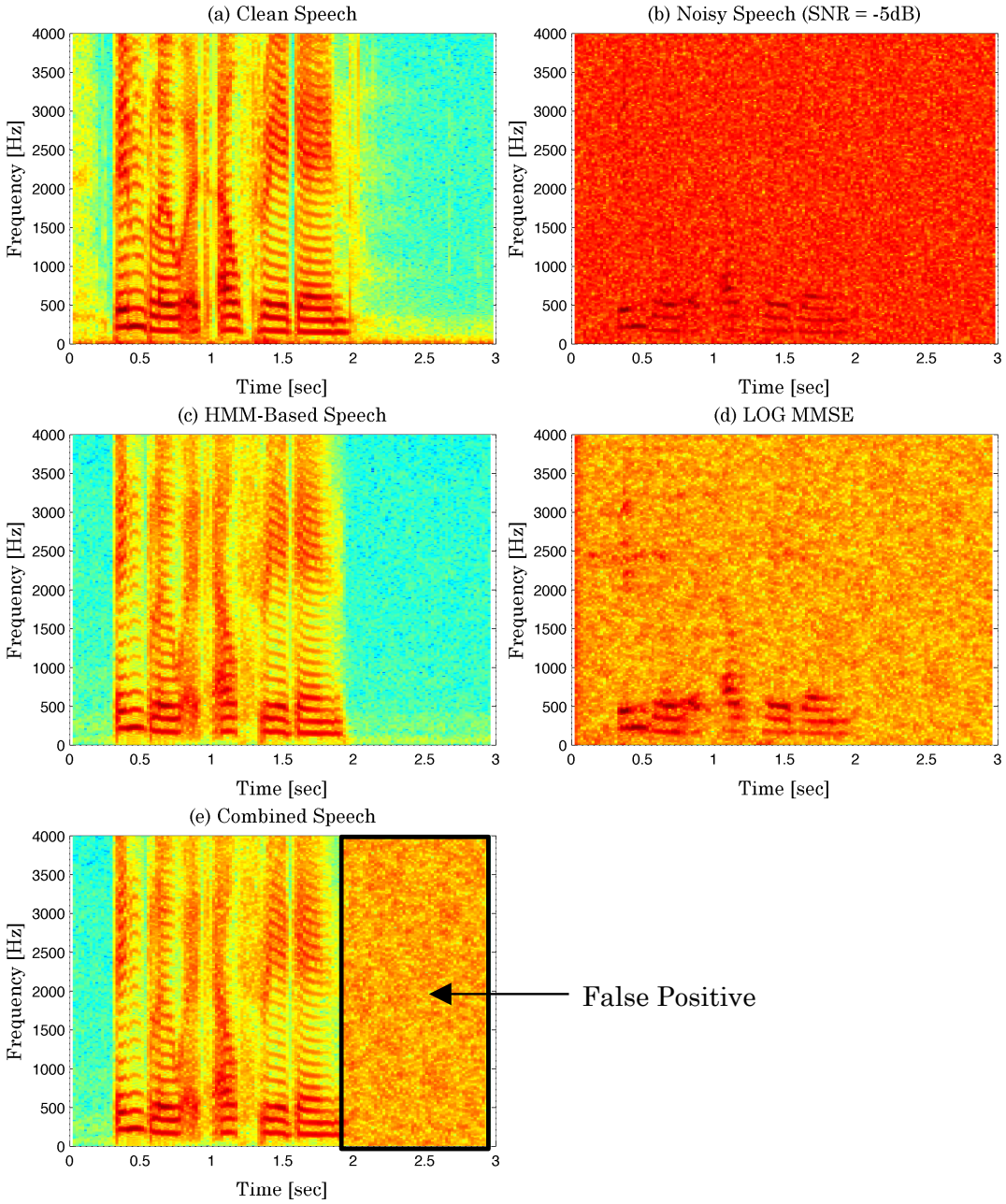

Figure 6.7: Narrowband spectrograms of female speech of "Bin Blue At L Three Again". Subplots (a), (b), (c), (d) and (e) show natural clean speech, noisy speech contaminated with white noise at SNR of -5 dB, enhanced speech with HMM-based enhancement, log MMSE and combined speech respectively.

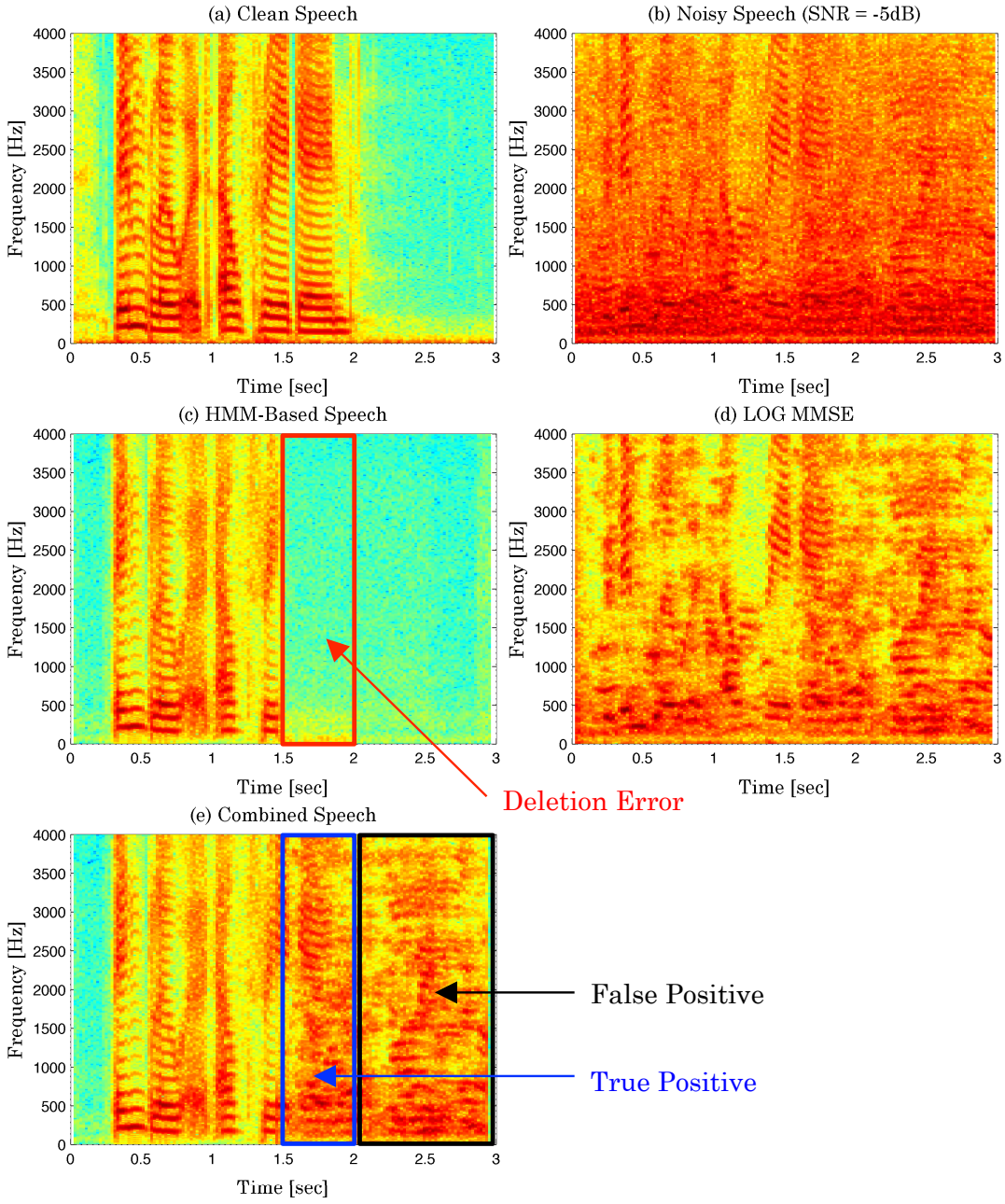

Figure 6.8: Narrowband spectrograms of female speech of "Bin Blue At L Three Again". Subplots (a), (b), (c), (d) and (e) show natural clean speech, noisy speech contaminated with babble noise at SNR of -5 dB, enhanced speech with HMM-based enhancement, log MMSE and combined speech respectively.

features. Therefore, this section discusses the problem associated with the vocoder and over-smoothing in order to improve the baseline performance of HMM-based speech.

6.3.1 Deterioration by STRAIGHT

The proposed method of HMM-based speech enhancement adopts STRAIGHT as a vocoder in its reconstruction process and it is true that the performance of STRAIGHT possibly limits the quality of enhanced speech. Therefore, the following experiment is conducted to separate the limitation brought by STRAIGHT from the deterioration caused by HMM-based speech synthesis.

The experiment uses clean natural speech from four speakers in the GRID database, two males and two females, which is downsampled to 8 kHz. From the 1000 utterances from each speaker, 800 are used for training and the remainder are for testing. Firstly, the spectral envelope and aperiodicity of the test set in the time-frequency domain are extracted with the setting of 25 ms Hamming window, 5 ms frame shift and 1024-point FFT while the fundamental frequency contour of the test set is obtained with PEFAC. This parameter set of natural speech is then retransformed to the time-domain by STRAIGHT to evaluate the influence that STRAIGHT gives.

Conversely, in HMM-based speech synthesis, STRAIGHT reconstructs speech from the spectral envelope and aperiodicity synthesised from the trained HMMs, error-free state sequences of the test set and the fundamental contour obtained by PEFAC. The configuration of acoustic feature vectors and HMMs is same as the experiments in Section 6.2.3, and error-free state sequences are obtained by forced alignment using reference transcription labels of the test set.

Table 6.2 compares PESQ and NCM scores between speech reconstructed by STRAIGHT from the natural speech parameter set and the HMM-based parameter set. The reconstructed speech from the natural speech parameter set scored 3.37 in PESQ and 0.96 in NCM while it would obtain 4.5 for PESQ and 1.0 for NCM if the reconstruction process of the STRAIGHT vocoder had no deterioration. Therefore, it is true that the process of STRAIGHT affects PESQ and NCM scores in HMM-based speech synthesis. However, the scores of speech reconstructed from the HMM-based parameter set are much lower than the natural speech parameter set, and this means that the con-

straint on the performance of HMM-based speech synthesis is dominated by the process to synthesise the HMM-based speech parameters rather than the reconstruction process of the STRAIGHT vocoder. Therefore, the remainder of this section focusses on mitiga-

	Natural speech features	HMM-based speech features
PESQ	3.37	2.45
NCM	0.96	0.70

Table 6.2: PESQ and NCM scores of speech reconstructed by STRAIGHT from natural speech parameters and HMM-based speech parameters

tion of over-smoothing, which is the other main cause of the deterioration in quality of HMM-based speech mentioned above, to refine the HMM-based speech.

6.3.2 Over-smoothing

The parameters of HMM-based speech are synthesised by maximising their output probabilities according to the statistically trained models with the constraints between static and dynamic features, i.e finding the parameters satisfying Equations (4.84), (4.86) and (4.90). The statistical averaging operation in this process often produces over-smoothing of the resulting parameters by which detailed characteristics of speech are missed. Consequently, the synthesised speech sounds muffled as compared with natural speech [9].

The speech synthesis algorithm considering global variance (GV) [113] has been reported as one of the most successful approaches to emphasise spectral formants from HMM-based parameters [9, 50, 112, 114]. Therefore, this method is explored as follows.

6.3.2.1 Global Variance

When a sequence of static features, $\mathbf{C}$, in a sequence of augmented observation features, $\mathbf{O}$, is determined as

$$\mathbf{C} = \left[\mathbf{c}_0^T, \mathbf{c}_1^T, \ldots, \mathbf{c}_{N-1}^T\right]^T \tag{6.12}$$

$$\mathbf{c}_n = \left[c_n(0), c_n(1), \ldots, c_n(M-1)\right]^T \tag{6.13}$$

where $c_n(m)$ represents the m-th coefficient of the static feature in the n-th frame, a GV vector of the static feature, $\mathbf{v}(\mathbf{C})$, is derived as

$$\begin{aligned}
\mathbf{v}(\mathbf{C}) &= [v(0), v(1), \ldots, v(M-1)]^T && (6.14)\\
v(m) &= \frac{1}{N-1}\sum_{n=0}^{N-1}\left(c_n(m) - \mu_c(m)\right)^2 && (6.15)\\
\mu_c(m) &= \frac{1}{N}\sum_{n=0}^{N-1} c_n(m) && (6.16)\\
& && (6.17)
\end{aligned}$$

GVs are calculated across all the training utterances and they are then modelled as a Gaussian distribution as

$$\begin{aligned}
P\left(\mathbf{v}(\mathbf{C}) \mid \lambda_{gv}\right) &= \mathcal{N}\left(\mathbf{v}(\mathbf{C}); \boldsymbol{\mu}_{gv}, \boldsymbol{\Sigma}_{gv}\right) && (6.18)\\
&= \frac{1}{\sqrt{(2\pi)^M |\boldsymbol{\Sigma}_{gv}|}} e^{\left(-\frac{1}{2}(\mathbf{v}(\mathbf{C})-\boldsymbol{\mu}_{gv})^T \boldsymbol{\Sigma}_{gv}^{-1}(\mathbf{v}(\mathbf{C})-\boldsymbol{\mu}_{gv})\right)} && (6.19)
\end{aligned}$$

where λ_{gv} represents the parameter set of the GV model which consists of the mean vector, $\boldsymbol{\mu}_{gv}$, and the diagonal covariance matrix, $\boldsymbol{\Sigma}_{gv}$. HMMs, λ, and GV model, λ_{gv}, are independently trained, and the HMM-based synthesis process generates the static feature, $\hat{\mathbf{C}}$, to satisfy the following equation instead of Equation (4.90).

$$\hat{\mathbf{C}} = \arg\max_{\mathbf{C}} \left\{\mathcal{N}\left(\mathbf{WC}; \boldsymbol{\mu}_{\hat{\mathbf{q}}}, \boldsymbol{\Sigma}_{\hat{\mathbf{q}}}\right) \cdot \mathcal{N}\left(\mathbf{v}(\mathbf{C}); \boldsymbol{\mu}_{gv}, \boldsymbol{\Sigma}_{gv}\right)^{\alpha}\right\} \tag{6.20}$$

where $\boldsymbol{\mu}_{\hat{q}}$ and $\boldsymbol{\Sigma}_{\hat{q}}$ are the mean vector and the diagonal covariance matrix of the Gaussian distribution at state, $\hat{q}$, of λ, matrix, $\mathbf{W}$, contains the regression coefficients to transform the static vectors into the augmented vectors and α is the GV weight, which is usually set equal to the ratio of the vector dimensions between $\mathbf{O}$ and $\mathbf{v}(\mathbf{C})$, to control the balance between the HMM and GV probabilities. The GV probability in Equation (6.20) plays a role to retain the dynamic range of $\hat{\mathbf{C}}$ close to the dynamic range of the training data set and thus, works as a penalty to prevent over-smoothing [9].

6.3.2.2 Experimental Results

HMM-based speech synthesis considering the GV probability, i.e Equation (6.20), is examined as follows. Experiments uses speech from four speakers in the GRID database, two males and two females, which is downsampled to 8 kHz. From the 1000 utterances from each speaker, 800 are used for training and the remainder are for testing. In the experiment, the feature vectors of speech, $\mathbf{O}$, are configured as MFCC23-23 in Table 5.1 with a 25 ms Hamming window whose frame shift is set to 5 ms and extracted from the training set of clean speech. A set of 12 state CD-triphone HMMs, TRI_N/23, determined in Table 5.2 is trained on $\mathbf{O}$ to form λ, and global variance model, λ_{gv}, is also formed by calculating $\boldsymbol{\mu}_{gv}$ and $\boldsymbol{\Sigma}_{gv}$ from variance vector, $\mathbf{v}(\mathbf{C})$, of all the utterances in the training set, which is determined by Equation (6.14).

State sequences of clean speech of the test set, which have no decoding errors, are then obtained by using forced alignment to synthesise the clean speech parameters which satisfy Equation (6.20).

Table 6.3 compares PESQ and NCM between HMM-based speech with and without the GV model. This result shows HMM-based speech using the GV model increases PESQ

	HMM	HMM+GV
PESQ	2.45	2.50
NCM	0.70	0.70

Table 6.3: PESQ and NCM scores of HMM-based speech with and without the GV model.

by 0.05, but the score of NCM is not improved by the GV model. The effectiveness of the GV model in this test may possibly be limited by the characteristics of the test set in which the length of each utterance is less than three seconds. In the case of using GV model, the algorithm tries to generate the parameters which always have a particular fixed variance. However, short time utterances such as GRID database generally have less variance in their speech features than longer utterances because they contain less variety of phonemes [115]. Consequently, the algorithm gives unmatched variance to the synthesised speech features and it causes the process not to work effectively. Figure 6.9 shows the spectral surface of female speech, "Bin Blue At L Three Again", in the

time-frequency domain. Subplots (a), (b) and (c) show natural speech, HMM-based

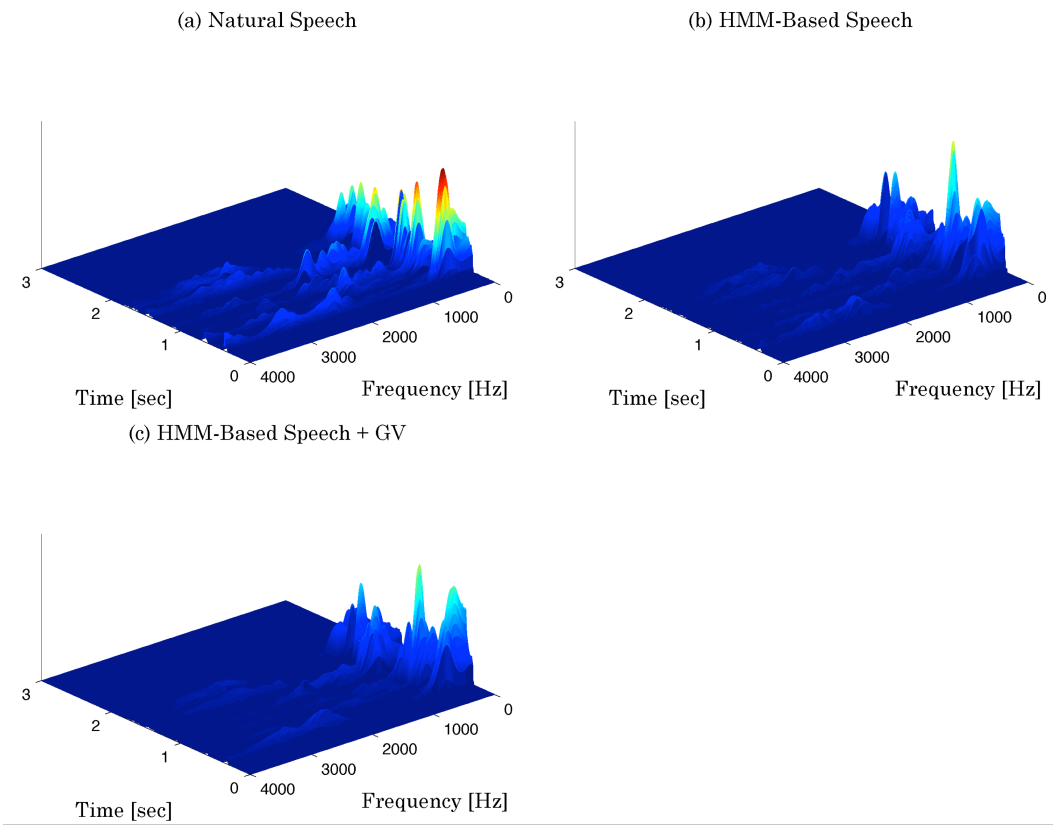

Figure 6.9: Spectral surface of female speech, "Bin Blue At L Three Again", in the time-frequency domain. a), b) and c) show natural speech, HMM-based speech and HMM-based speech with the GV model respectively.

speech and HMM-based speech with the GV model respectively. This shows the spectral envelopes of HMM-based speech are compensated to be closer to the trajectory of the natural speech by the GV model though over-smoothing still exists.

HMM-based speech with the GV model is then examined in white noise and babble noise at SNRs between -5 dB and 10 dB. State and model sequences are obtained by HMM decoding with the noise-adapted HMMs. Clean speech features are then synthesised by using the clean HMMs and the GV model. PESQ and NCM scores are compared with HMM-based speech without the GV model and with log MMSE in Figure 6.10. Subplots (a) and (b) show PESQ scores in white noise and babble noise at different SNRs whereas subplots (c) and (d) illustrate the performance in terms of NCM in white noise and babble noise. The effect of using the GV model on PESQ is more remarkable at higher SNRs

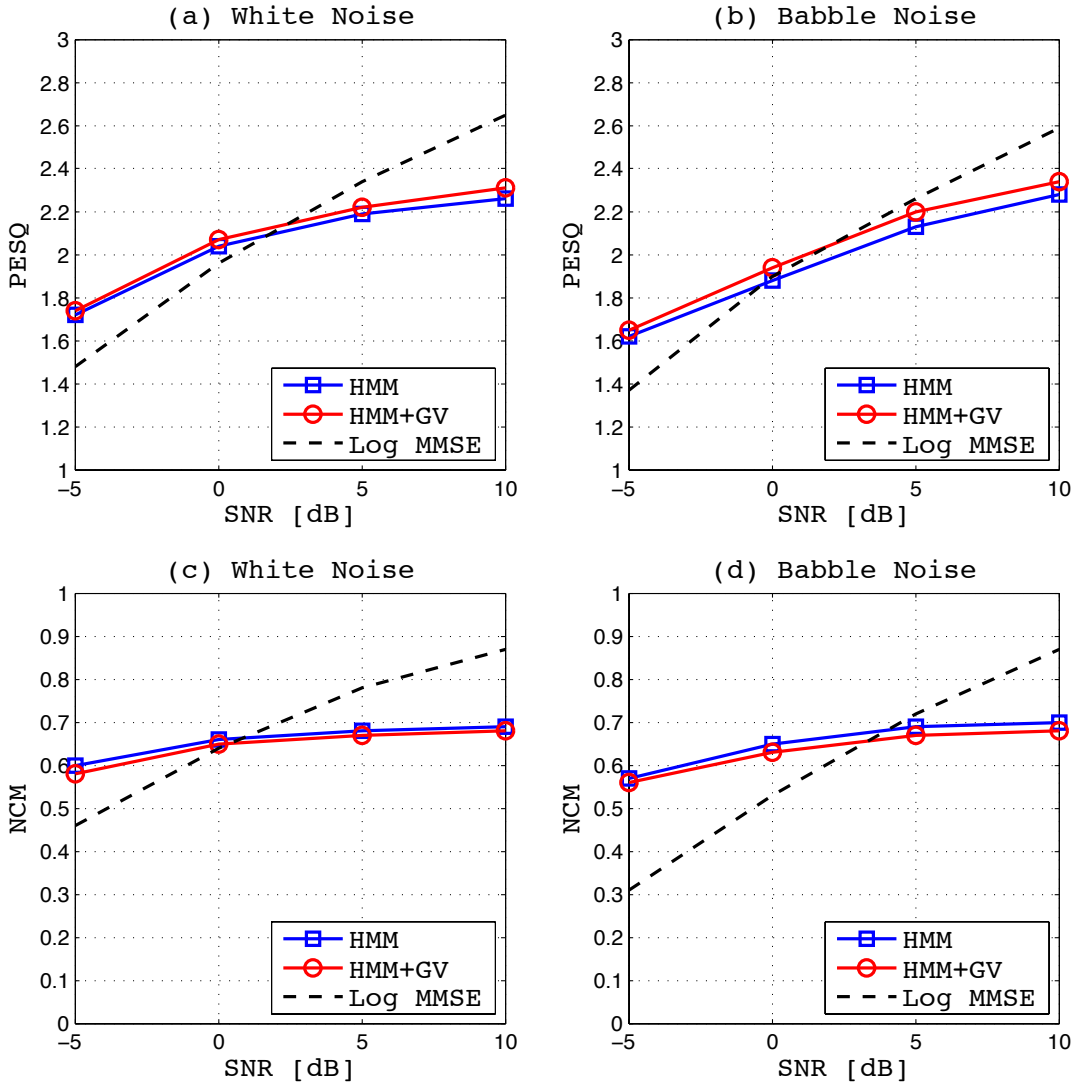

Figure 6.10: Performance of HMM-based speech with the GV model in noisy conditions compared with HMM-based speech without GV and Log MMSE. a) and b) show PESQ scores in white noise and babble noise at different SNRs while c) and d) illustrate NCM scores in white noise and babble noise.

because the compensation for over-smoothing in error frames are not useful. Conversely, NCM scores of HMM-based speech tend to become lower by keeping the variance of the synthesised speech the fixed value. HMM-based speech with GV, however, sounds explicitly clear in a subjective sense as compared with HMMs without GV. Moreover, PESQ scores applied to HMM-based speech empirically tend to be lower than subjective hearing impression. This seems to be attributed to a characteristic of reconstruction-based speech enhancement brought by the decoding process where the time allocation of each phoneme is possibly different from the original speech. Although the difference is not large in terms of human auditory perception, PESQ detects it and which results in lower scores. Therefore, it is also important to evaluate the performance with subjective listening tests which are carried out in the next chapter in addition to objective tests.

6.4 Conclusion of the Chapter

This chapter first discussed confidence measuring to first identify and then to compensate for the influence of decoding errors. A novel method to measure frame-by-frame and phoneme-by-phoneme confidence was studied and then enhanced speech was produced by combining HMM-based speech with log MMSE according to the phoneme-by-phoneme confidence and evaluated. The proposed confidence measure and combined speech improved PESQ scores at low SNRs, specifically in babble noise. The benefit was, however, limited and NCM scores were decreased by residual and musical noise brought by log MMSE segments.

In the latter part of the chapter, HMM-based speech synthesis using the GV model was explored to improve the baseline quality of the proposed HMM-based speech enhancement. The evaluation in clean and noisy conditions showed improvement in PESQ over noise conditions. Those rises, however, seem to be much less than a subjective hearing impression and thus, detailed subjective listening tests are carried out for further evaluation of the proposed HMM-based speech enhancement in the next chapter.

Chapter 7

Evaluation of the Proposed HMM-Based Speech Enhancement

The proposed method of HMM-based speech enhancement has firstly been discussed and evaluated with different acoustic and language models in Chapter 4. Secondly it has been put into more practical use by applying HMM adaptation in Chapter 5, and then the techniques to tackle HMM decoding errors and over-smoothing in the synthesised speech parameters are discussed in Chapter 6. Now it is time to carry out a full evaluation of speech quality and intelligibility of the proposed method. This chapter first evaluates the performance of the proposed method objectively in terms of PESQ and NCM as well as the previous chapters, and then carries out subjective listening tests.

7.1 Introduction

The fundamental process in the proposed HMM-based speech enhancement has been explored in Chapter 4 and HMM adaptation to noisy speech is then applied in Chapter 5 in order to improve HMM decoding accuracy and time warping in the resultant model and state sequences in practical conditions. Moreover, Chapter 6 discussed the methods to tackle decoding errors and to improve the baseline performance, and thus, the aim of this chapter is to carry out a comprehensive evaluation of those techniques.

The experiments in Chapter 4 have shown that HMM-based speech enhancement performs best with noise-matched whole-word HMMs and language model. Next best is noise-matched context-dependent-triphone CD-triphone HMMs with no grammar in terms of PESQ and NCM (Figures 4.28 and 4.29). They have also shown that acoustic models trained by sequences of MFCC vectors with truncation of the high order coefficients have a tendency to obtain higher PESQ scores at low SNRs than acoustic models trained with MFCC vectors with no truncation because of their superiority in noise robustness in HMM decoding. Conversely, acoustic models trained with non truncated MFCC vectors obtain higher scores at high SNRs than acoustic models with MFCC truncation because of their superiority in the performance of HMM-based speech synthesis with correct state sequences.

Then, the tests in Chapter 5, which focus on CD-triphone HMMs for practical use, have shown that HMM decoding accuracy using acoustic features of MFCC vectors with no truncation is improved to the same level as decoding with acoustic features of truncated MFCCs by applying HMM adaptation to noisy speech (Figure 5.4). Consequently, HMM-based speech enhancement with non truncated MFCCs performs better than using MFCC truncation over the SNR range between -5 dB and 10 dB (Figures 5.5 and 5.6). Therefore, for the tests in this chapter, the acoustic features are configured as shown in Table 7.1 considering the preceding empirical findings and practical use. The number of

Type of HMMs:	Context Dependent Triphone (CD-triphone HMMs)
HMMs for Decoding:	Noise-Adapted HMMs, $\hat{\mathbf{\Lambda}}$
HMMs for Synthesis:	Clean HMMs, $\mathbf{\Lambda}$
Number of HMM States:	12
Language Model:	No
Speech Features:	MFCCs, Aperiodicity and $\log f_0$
Filterbank Channels:	23
MFCC Truncation:	No
Aperiodicity Coefficients:	40
f_0 Estimation:	PEFAC
Dynamic Features:	Δ and Δ^2
Frame Length:	25 ms
Frame Interval:	5 ms
Window Type:	Hamming
FFT:	1024 Point FFT

Table 7.1: The common configuration of HMMs and acoustic features for the tests.

HMM states, the number of filterbank channels, the f_0 estimation method and the frame interval in Table 7.1 are determined considering the empirical results of Tables 4.8 and 4.11 and Figures 3.10, 5.5 and 5.6.

Furthermore, the test results in Chapter 6 show that combining HMM-based speech with log MMSE according to the phoneme-by-phoneme confidence measure improves PESQ at SNR of -5 dB in babble noise (Figure 6.6), and HMM-based speech synthesis considering the GV model also improves the PESQ scores over the SNR range between -5 dB and 10 dB in white noise and babble noise (Figure 6.10). Therefore, HMM-based enhanced speech with the GV model (HMM+GV) and HMM+GV combined with log MMSE according to the phoneme-by-phoneme confidence measure (HMM+GV+CMB) are also evaluated in addition to the basic HMM-based enhanced speech (HMM) as shown in Table 7.2.

HMM:	Basic HMM-Based enhanced speech
HMM+GV :	HMM-Based enhanced speech with the GV model
HMM+GV+CMB:	HMM+GV combined with the log MMSE method according to the phoneme-by-phoneme confidence measure

Table 7.2: Configurations of HMM-based speech enhancement for the tests.

The remainder of this chapter first describes the procedure to obtain enhanced speech by the proposed HMM-based speech enhancement for use in subsequent objective and subjective tests. This procedure has been discussed across several chapters and it is now presented in a clear single form. Next objective tests for each method in terms of PESQ and NCM are then carried out followed by subjective listening tests for speech quality and intelligibility.

7.2 Test Procedure

Tests in this chapter use speech from four speakers in the GRID database, two males and two females, which is downsampled to 8 kHz. From the 1000 utterances from each speaker, 800 are used for training and the remainder are for testing. The performance of the proposed HMM-based speech enhancement is examined in white noise and babble

noise at SNRs between -5 dB and 10 dB. The procedure to enhance noisy speech by the proposed HMM-based speech enhancement is formed by the following processes.

- Feature Extraction
- HMM Training
- HMM Adaptation
- HMM Decoding
- Speech Parameter Synthesis
- Confidence Measuring
- Speech Reconstruction

Each of these processes is explained in the following subsections.

7.2.1 Feature Extraction

At the first stage of the proposed HMM-based speech enhancement, sequences of MFCCs, $\mathbf{C}^x$, and sequences of aperiodicity coefficients, $\mathbf{C}^a$, are first extracted from the training speech and test speech as

$$\mathbf{C}^x = \left[(\mathbf{c}_0^x)^T, (\mathbf{c}_1^x)^T, \ldots, (\mathbf{c}_{I-1}^x)^T\right]^T \tag{7.1}$$

$$\mathbf{c}_i^x = [x_i(0), x_i(1), \ldots, x_i(22)]^T \tag{7.2}$$

$$\mathbf{C}^a = \left[(\mathbf{c}_0^a)^T, (\mathbf{c}_1^a)^T, \ldots, (\mathbf{c}_{I-1}^a)^T\right]^T \tag{7.3}$$

$$\mathbf{c}_i^a = [a_i(0), a_i(1), \ldots, a_i(39)]^T \tag{7.4}$$

where $x_i(l)$ and $a_i(m)$ are the l-th coefficient of the MFCCs and the m-th coefficient of the aperiodicity coefficients at the i-th frame of the speech. These speech feature sequences are then combined to construct a unified speech feature sequence, $\mathbf{C}$, as

$$\mathbf{C} = \left[\mathbf{c}_0^T, \mathbf{c}_1^T, \ldots, \mathbf{c}_{I-1}^T\right]^T \tag{7.5}$$

$$\mathbf{c}_i = \left[(\mathbf{c}_i^x)^T, (\mathbf{c}_i^a)^T\right]^T \tag{7.6}$$

Then the sequence of the static speech features, $\mathbf{C}$, derives a sequence of the augmented speech features, $\mathbf{O}$, by adding the velocity and acceleration derivatives of each feature as

$$\begin{aligned} \mathbf{O} &= \mathbf{WC} && (7.7) \\ &= \left[\mathbf{o}_0^T, \mathbf{o}_1^T, \ldots, \mathbf{o}_{I-1}^T\right]^T && (7.8) \\ & && (7.9) \end{aligned}$$

where $\mathbf{W}$ denotes a matrix of the regression coefficients referred to as Equation (4.79) to transform the sequence of the static features into the sequence of the augmented features including their first and second order dynamic features, and

$$\mathbf{o}_i = \left[(\mathbf{c}_i^x)^T, (\Delta\mathbf{c}_i^x)^T, (\Delta^2\mathbf{c}_i^x)^T, (\mathbf{c}_i^a)^T, (\Delta\mathbf{c}_i^a)^T, (\Delta^2\mathbf{c}_i^a)^T\right]^T \tag{7.10}$$

where Δ and Δ^2 are the notations of a velocity derivative and acceleration derivative. This structure of $\mathbf{O}$ is referred to as Figure 4.20 but the sequence of the fundamental frequencies is not included here because the fundamental frequency contour is extracted directly from the test speech by using PEFAC to reconstruct speech rather than synthesising it from the statistical model.

With the preceding procedure, the observation sequence, $\mathbf{O}$, of the training speech is extracted in the clean condition whereas $\mathbf{O}$ of the test speech is extracted in each noisy condition.

7.2.2 HMM Training

Observation sequences, $\mathbf{O}$, extracted from the training speech and their reference transcripts are then used to train a set of CD-triphone HMMs, $\mathbf{\Lambda} = \{\lambda_1, \lambda_2, \ldots, \lambda_D\}$. The Viterbi algorithm first renews the estimate of state sequence, $\hat{\mathbf{q}}$, as

$$\hat{\mathbf{q}} = \arg\max_{\mathbf{q}\in\mathbf{Q}} P(\mathbf{q} \mid \mathbf{O}, \boldsymbol{\lambda}) \tag{7.11}$$

where $\boldsymbol{\lambda}$ represents a sequence of the initial HMMs corresponding to the transcript of the training speech, $\mathbf{q}$ is the initial state sequence and $\mathbf{Q}$ denotes the group which comprises

all the possible state sequences to achieve $\boldsymbol{\lambda}$ during $\mathbf{O}$. Using the renewed state sequence, $\hat{\mathbf{q}}$, the Baum-Welch algorithm then renews the parameters of $\lambda_k \in \boldsymbol{\lambda}$ ($k \in \{1, 2, \ldots, D\}$) as

$$\tilde{\lambda}_k = \arg\max_{\lambda_k} P(\mathbf{O} \mid \lambda_k), \qquad \text{for } \forall \lambda_k \in \boldsymbol{\lambda} \tag{7.12}$$

After the initial HMM parameters and the initial state sequence are replaced with the renewed model parameters and state sequence, all the $\lambda_k \in \boldsymbol{\lambda}$ are optimised by an iteration of Equations (7.11) and (7.12).

Then the number of the HMMs, D, is reduced to around 200 for each speaker by MDL-based clustering [94] in order to avoid underfitting and overfitting, and finally, the clustered models are trained with the preceding procedure.

7.2.3 HMM Adaptation

The HMMs trained with the clean speech are then adapted to the noisy test speech at the next stage, HMM adaptation process. In this process, the noise power spectrum of the noisy test speech at i-th frame, $|\hat{D}_i(f)|^2$, is first estimated by unbiased MMSE estimation [116] and then it is transformed to the Mel-filterbank domain to form a sequence of the Mel-filterbank coefficient vectors of the noise, $\hat{\mathbf{D}}^{fb}$, as

$$\hat{\mathbf{D}}^{fb} = \left[(\hat{\mathbf{d}}_0^{fb})^T, (\hat{\mathbf{d}}_1^{fb})^T, \ldots, (\hat{\mathbf{d}}_{I-1}^{fb})^T\right]^T \tag{7.13}$$

$$\hat{\mathbf{d}}_i^{fb} = \left[\hat{D}_i^{fb}(0), \hat{D}_i^{fb}(1), \ldots, \hat{D}_i^{fb}(22)\right]^T \tag{7.14}$$

where $\hat{D}_i^{fb}(m)$ represents the m-th Mel-filterbank coefficient of $|\hat{D}_i(f)|^2$.

The noise model of the test speech $\hat{\lambda}_d^{fb}$ is configured as a single-state-single-gaussian model and the model parameters are derived as

$$\hat{\boldsymbol{\mu}}_d^{fb} = \frac{1}{I}\sum_{i=0}^{I-1} \hat{\mathbf{d}}_i^{fb} \tag{7.15}$$

$$\hat{\boldsymbol{\Sigma}}_d^{fb} = \frac{1}{I-1}\sum_{i=0}^{I-1} \left(\hat{\mathbf{d}}_i^{fb} - \hat{\boldsymbol{\mu}}_d^{fb}\right)\left(\hat{\mathbf{d}}_i^{fb} - \hat{\boldsymbol{\mu}}_d^{fb}\right)^T \tag{7.16}$$

where $\hat{\boldsymbol{\mu}}_d^{fb}$ and $\hat{\boldsymbol{\Sigma}}_d^{fb}$ are the mean vector and the covariance matrix of $\hat{\lambda}_d^{fb}$ respectively.

Simultaneously, the mean vectors and the covariance matrices of the static MFCC

feature in the s-th state ($s = 1, 2, \cdots, 12$) of the trained HMMs, λ_k ($k = 1, 2, \ldots, D$), are denoted as $\boldsymbol{\mu}_{s,k}$ and $\boldsymbol{\Sigma}_{s,k}$, and these are transformed to the linear Mel-filterbank domain by the unscented transform discussed in Section 5.2.3 to obtain $\boldsymbol{\mu}^{fb}_{s,k}$ and $\boldsymbol{\Sigma}^{fb}_{s,k}$.

The parallel model combination then adapts $\boldsymbol{\mu}^{fb}_{s,k}$ and $\boldsymbol{\Sigma}^{fb}_{s,k}$ to the noisy speech as

$$\hat{\boldsymbol{\mu}}^{fb}_{s,k} = \boldsymbol{\mu}^{fb}_{s,k} + \hat{\boldsymbol{\mu}}^{fb}_{d} \tag{7.17}$$

$$\hat{\boldsymbol{\Sigma}}^{fb}_{s,k} = \boldsymbol{\Sigma}^{fb}_{s,k} + \hat{\boldsymbol{\Sigma}}^{fb}_{d} \tag{7.18}$$

The noise-adapted mean vectors and covariance matrices in the linear Mel-filterbank domain, $\hat{\boldsymbol{\mu}}^{fb}_{s,k}$ and $\hat{\boldsymbol{\Sigma}}^{fb}_{s,k}$ are finally transformed back to the MFCC domain by the unscented transform in order to constitute the noise-adapted HMMs $\hat{\lambda}_k$.

7.2.4 HMM Decoding

The HMM decoding process uses a set of the CD-triphone HMMs adapted to the noisy test speech, $\hat{\boldsymbol{\Lambda}} = \{\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_D\}$, and a sequence of the static MFCC vectors, $\mathbf{C}^x$, extracted from the test speech in order to obtain the most likely model sequence including their state sequences, $\hat{s}(i)$ as.

$$\hat{s}(i) = \arg\max_{\mathbf{q} \in \mathbf{Q}_{\Pi}} P\left(\mathbf{q} \mid \boldsymbol{\Pi}, \mathbf{C}^x\right) \tag{7.19}$$

where $\boldsymbol{\Pi}$ is a group which consists of all the possible CD-triphone model sequences during the observation and $\mathbf{Q}_{\Pi}$ represents a group which comprises all the possible state sequences which achieve $\forall \hat{\boldsymbol{\lambda}} \in \boldsymbol{\Pi}$. Equation (7.19) is solved by the Viterbi algorithm discussed in Section 4.2.2.

7.2.5 Speech Parameter Synthesis

The next stage utilises $\hat{s}(i)$ and $\boldsymbol{\Lambda}$ to synthesise the clean speech features of the test speech, i.e. the sequences of the MFCCs, $\hat{\mathbf{C}}^x$, and the aperiodicity coefficients, $\hat{\mathbf{C}}^a$, statistically. This process is attained as follows. A sequence of the HMM-based speech parameters including their dynamic features, $\hat{\mathbf{O}}$, and a sequence of the static HMM-

based speech parameters, $\hat{\mathbf{C}}$, are first defined as

$$\hat{\mathbf{C}} = \left[\hat{\mathbf{c}}_0^T, \hat{\mathbf{c}}_1^T, \ldots, \hat{\mathbf{c}}_{I-1}^T\right]^T \tag{7.20}$$

$$\hat{\mathbf{c}}_i = \left[(\hat{\mathbf{c}}_i^x)^T, (\hat{\mathbf{c}}_i^a)^T\right]^T \tag{7.21}$$

$$\hat{\mathbf{O}} = \left[\hat{\mathbf{o}}_0^T, \hat{\mathbf{o}}_1^T, \ldots, \hat{\mathbf{o}}_{I-1}^T\right]^T \tag{7.22}$$

$$\hat{\mathbf{o}}_i = \left[(\hat{\mathbf{c}}_i^x)^T, (\Delta\hat{\mathbf{c}}_i^x)^T, (\Delta^2\hat{\mathbf{c}}_i^x)^T, (\hat{\mathbf{c}}_i^a)^T, (\Delta\hat{\mathbf{c}}_i^a)^T, (\Delta^2\hat{\mathbf{c}}_i^a)^T\right]^T \tag{7.23}$$

where

$$\hat{\mathbf{c}}_i^x = [\hat{x}_i(0), \hat{x}_i(1), \ldots, \hat{x}_i(22)]^T \tag{7.24}$$

$$\hat{\mathbf{c}}_i^a = [\hat{a}_i(0), \hat{a}_i(1), \ldots, \hat{a}_i(39)]^T \tag{7.25}$$

where $\hat{x}_i(l)$ and $\hat{a}_i(m)$ represent the l-th coefficient of the synthesised MFCCs and the m-th coefficient of the synthesised aperiodicity coefficients at the i-th frame. Then $\hat{\mathbf{C}}$ is synthesised as

$$\hat{\mathbf{C}} = \arg\max_{\mathbf{C}'} P\left(\mathbf{W}\mathbf{C}' \mid \hat{s}(i), \mathbf{\Lambda}\right) \tag{7.26}$$

$$= \arg\max_{\mathbf{C}'} \left\{\mathcal{N}(\mathbf{W}\mathbf{C}'; \boldsymbol{\mu}_{\hat{\mathbf{q}}}, \mathbf{\Sigma}_{\hat{\mathbf{q}}})\right\} \tag{7.27}$$

where

$$\boldsymbol{\mu}_{\hat{\mathbf{q}}} = \left[\boldsymbol{\mu}_{\hat{q}_0}^T, \boldsymbol{\mu}_{\hat{q}_1}^T, \cdots, \boldsymbol{\mu}_{\hat{q}_{I-1}}^T\right]^T \tag{7.28}$$

$$\mathbf{\Sigma}_{\hat{\mathbf{q}}} = \mathrm{diag}\left[\boldsymbol{\sigma}_{\hat{q}_0}^T, \boldsymbol{\sigma}_{\hat{q}_1}^T, \cdots, \boldsymbol{\sigma}_{\hat{q}_{I-1}}^T\right]^T \tag{7.29}$$

where $\boldsymbol{\mu}_{\hat{q}_i}$ is the mean vector of the HMM state at the i-th frame corresponding to $s(i)$ while $\boldsymbol{\sigma}_{\hat{q}_i}$ represents a diagonal elements of the covariance matrix of the HMM state at the i-th frame. Therefore, the Newton-Raphson algorithm [117] is applied to find $\hat{\mathbf{C}}$ satisfying

$$\frac{\partial \log \mathcal{N}(\mathbf{W}\hat{\mathbf{C}}; \boldsymbol{\mu}_{\hat{\mathbf{q}}}, \mathbf{\Sigma}_{\hat{\mathbf{q}}})}{\partial \hat{\mathbf{C}}} = 0 \tag{7.30}$$

Alternatively, for the test configurations of HMM+GV and HMM+GV+CMB in

Table 7.2, the following is solved instead of Equation (7.27) to synthesise $\hat{\mathbf{C}}$.

$$\hat{\mathbf{C}} = \arg\max_{\mathbf{C}'} \left\{ \mathcal{N}\left(\mathbf{W}\mathbf{C}'; \boldsymbol{\mu}_{\hat{\mathbf{q}}}, \boldsymbol{\Sigma}_{\hat{\mathbf{q}}}\right) \cdot \mathcal{N}\left(\mathbf{v}(\mathbf{C}'); \boldsymbol{\mu}_{gv}, \boldsymbol{\Sigma}_{gv}\right)^{\alpha} \right\} \tag{7.31}$$

where $\boldsymbol{\mu}_{gv}$ and $\boldsymbol{\Sigma}_{gv}$ are the mean vector and covariance matrix of the GV model, which has been discussed in Section 6.3.2.1, α is set equal to the ratio of the vector dimension between $\hat{\mathbf{O}}$ and $\hat{\mathbf{C}}$, and $\mathbf{v}(\mathbf{C}')$ is determined as

$$\mathbf{v}(\hat{\mathbf{C}}') = [v(0), v(1), \ldots, v(23+40-1)]^T \tag{7.32}$$

$$v(m) = \frac{1}{62} \sum_{i=0}^{I-1} \left(\hat{c}_i(m) - \mu_{\hat{c}}(m)\right)^2 \tag{7.33}$$

$$\mu_{\hat{c}}(m) = \frac{1}{63} \sum_{i=0}^{I-1} \hat{c}_i(m) \tag{7.34}$$

$$\hat{c}_i(m) = \begin{cases} \hat{x}_i(m) & m = 0, 1, \ldots, 22 \\ \hat{a}_i(m-22) & m = 23, 24, \ldots, 62 \end{cases} \tag{7.35}$$

After the most likely $\hat{\mathbf{C}}$ is obtained, $\hat{\mathbf{c}}_i^x$ and $\hat{\mathbf{c}}_i^a$ for $\forall i$, which are referred to as Equations (7.20) and (7.21), are then extracted from $\hat{\mathbf{C}}$ to construct the sequence of the HMM-based MFCCs, $\hat{\mathbf{C}}^x$ and the sequence of the HMM-based aperiodicity coefficients, $\hat{\mathbf{C}}^a$, as

$$\hat{\mathbf{C}}^x = \left[(\hat{\mathbf{c}}_0^x)^T, (\hat{\mathbf{c}}_1^x)^T, \ldots, (\hat{\mathbf{c}}_{I-1}^x)^T\right]^T \tag{7.36}$$

$$\hat{\mathbf{C}}^a = \left[(\hat{\mathbf{c}}_0^a)^T, (\hat{\mathbf{c}}_1^a)^T, \ldots, (\hat{\mathbf{c}}_{I-1}^a)^T\right]^T \tag{7.37}$$

Additionally, a sequence of the MFCCs of the HMM-based noisy speech, $\hat{\mathbf{C}}^y$, is also derived by replacing a set of clean HMMs, $\boldsymbol{\Lambda}$, with a set of noise-adapted HMMs, $\hat{\boldsymbol{\Lambda}}$, in the preceding procedure for the test configuration of HMM+GV+CMB in which $\hat{\mathbf{C}}^y$ is required for the confidence measuring.

7.2.6 Confidence Measuring

The test configuration of HMM+GV+CMB applies the confidence measure discussed in Section 6.2 to mitigate the influence of HMM decoding errors at this stage.

A sequence of MFCCs extracted from test speech, $\mathbf{C}^x$, is first compared with the

sequence of the MFCCs synthesised by the noise-adapted HMMs, $\hat{\mathbf{C}}^y$ to obtain frame-by-frame confidence of $\hat{s}(i)$ as

$$\begin{aligned}
\mathbf{u}_i &= \mathbf{C}\exp\left(\mathbf{C}^{-1}\mathbf{c}_i^x\right) && (7.38)\\
&= [u_i(0), u_i(1), \ldots, u_i(22)]^T && (7.39)\\
\hat{\mathbf{u}}_i &= \mathbf{C}\exp\left(\mathbf{C}^{-1}\hat{\mathbf{c}}_i^y\right) && (7.40)\\
&= [\hat{u}_i(0), \hat{u}_i(1), \ldots, \hat{u}_i(22)]^T && (7.41)\\
R_{u\hat{u}}(i) &= \frac{\mathcal{E}\left[(u_i(m)-\mu_i)\left(\hat{u}_i(m)-\hat{\mu}_i\right)\right]}{\sqrt{\mathcal{E}\left[(u_i(m)-\mu_i)^2\right]\mathcal{E}\left[(\hat{u}_i(m)-\hat{\mu}_i)^2\right]}} && (7.42)
\end{aligned}$$

where $\mathbf{C}$ and $\mathbf{C}^{-1}$ denote the DCT and IDCT matrices, and

$$\begin{aligned}
\mu_i &= \frac{1}{23}\sum_{m=0}^{22} u_i(m) && (7.43)\\
\hat{\mu}_i &= \frac{1}{23}\sum_{m=0}^{22} \hat{u}_i(m) && (7.44)
\end{aligned}$$

Phoneme-by-phoneme confidence is then derived as

$$P_j = \begin{cases} 1 & \frac{1}{i_{j+1}-i_j}\sum_{k=i_j}^{i_{j+1}} R_{y\hat{y}}(k) \geqslant \beta \\ 0 & \text{otherwise} \end{cases} \qquad (7.45)$$

where i_j represents the start frame of the j-th phoneme in $\hat{s}(i)$, and β is set equal to 0.8 with respect to the empirical results shown in Figures 6.3, 6.4 and 6.5.

7.2.7 Speech Reconstruction

At the final stage of the proposed HMM-based speech enhancement, the sequences of the synthesised speech features, $\hat{\mathbf{C}}^x$ and $\hat{\mathbf{C}}^a$, are first transformed to the linear Mel-filterbank

domain as

$$\hat{\mathbf{X}}^{fb} = \exp\left(\mathbf{C}^{-1}\hat{\mathbf{C}}^{x}\right) \tag{7.46}$$

$$= \left[(\hat{\mathbf{x}}_0^{fb})^T, (\hat{\mathbf{x}}_1^{fb})^T, \ldots, (\hat{\mathbf{x}}_{I-1}^{fb})^T\right]^T \tag{7.47}$$

$$\hat{\mathbf{A}}^{fb} = \exp\left(\mathbf{C}^{-1}\hat{\mathbf{C}}^{a}\right) \tag{7.48}$$

$$= \left[(\hat{\mathbf{a}}_0^{fb})^T, (\hat{\mathbf{a}}_1^{fb})^T, \ldots, (\hat{\mathbf{a}}_{I-1}^{fb})^T\right]^T \tag{7.49}$$

where

$$\hat{\mathbf{x}}_i^{fb} = [\hat{x}_i(0), \hat{x}_i(1), \ldots, \hat{x}_i(22)]^T \tag{7.50}$$

$$\hat{\mathbf{a}}_i^{fb} = [\hat{a}_i(0), \hat{a}_i(1), \ldots, \hat{a}_i(39)]^T \tag{7.51}$$

where $\hat{x}_i(l)$ represents the energy in the l-th Mel-filterbank channel of the HMM-based speech at the i-th frame while $\hat{a}_i(m)$ is the m-th coefficient of the aperiodicity coefficients of the HMM-based speech at the i-th frame in the linear Mel-filterbank domain. These are then transformed to the time-frequency domain to obtain the spectral surface, $\hat{X}(f,i)$, and the aperiodicity, $\hat{A}(f,i)$, where f denotes FFT bin index ($f = 0, 1, \ldots, 512$), by using the method applying channel normalisation and cubic spline interpolation which is referred to as Section 4.4.2.

Incidentally, the fundamental frequency contour, $f_0(i)$, is extracted from the test speech by using PEFAC because the fundamental frequency contour synthesised by HMMs cannot trace rapid changes as illustrated in Figure 4.25.

All the speech features required by the STRAIGHT vocoder, i.e. $\hat{X}(f,i)$, $\hat{A}(f,i)$ and $f_0(i)$, are now acquired and the enhanced time-domain speech, $\hat{x}(n)$, is reconstructed.

Additionally, the test configuration of HMM+GV+CMB combines $\hat{x}(n)$ with speech enhanced by log MMSE, $\hat{x}'(n)$, according to the phoneme-by-phoneme confidence, P_j, as

$$\hat{x}_{cmb}(n) = \begin{cases} P_1\hat{x}(n) + (1-P_1)\hat{x}'(n) & n_1 \leqslant n < n_2 \\ P_2\hat{x}(n) + (1-P_2)\hat{x}'(n) & n_2 \leqslant n < n_3 \\ \vdots & \vdots \\ P_J\hat{x}(n) + (1-P_J)\hat{x}'(n) & n_J \leqslant n \end{cases} \tag{7.52}$$

where n_j is the sample index corresponding to the beginning of the j-th phoneme in $\hat{x}(n)$ and J is the number of phonemes in $\hat{x}(n)$.

7.3 Objective Evaluation

Test speech enhanced by the preceding procedure is objectively evaluated in this section in terms of PESQ for speech quality and NCM for speech intelligibility comparing with baseline performance given by no noise compensation (NNC), and the conventional filtering approaches to speech enhancement represented by log MMSE (LOG) [36] and the subspace method (SUB) [40].

The test configurations for the proposed HMM-based speech enhancement are referred to as Tables 7.1 and 7.2, i.e. HMM, HMM+GV and HMM+GV+CMB.

7.3.1 Speech Quality

Table 7.3 and Figure 7.1 show the PESQ scores of the different configurations of the proposed method (HMM, HMM+GV and HMM+GV+CMB) at SNRs from -5 dB to 10 dB in white noise and babble noise comparing with baseline performance (NNC) and the conventional filtering approaches (LOG and SUB).

Noise	NNC	HMM	HMM+GV	HMM+GV+CMB	LOG	SUB
White Noise						
10 dB	1.93	2.26	2.31	2.30	**2.65**	2.57
5 dB	1.64	2.19	2.22	2.21	**2.34**	2.33
0 dB	1.43	2.04	**2.07**	2.04	1.96	2.04
-5 dB	1.27	1.72	**1.74**	1.69	1.48	1.59
Babble Noise						
10 dB	2.33	2.28	2.34	2.33	**2.59**	2.46
5 dB	2.01	2.13	2.20	2.18	**2.26**	2.12
0 dB	1.73	1.88	**1.94**	**1.94**	1.90	1.75
-5 dB	1.46	1.62	1.65	**1.73**	1.37	1.28

Table 7.3: PESQ scores at SNRs of 10 dB, 5 dB, 0 dB and -5 dB in white noise and babble noise

In white noise, NNC is always marked the lowest score over the SNR range, and the HMM-based Methods (HMM, HMM+GV and HMM+GV+CMB) are marked the highest

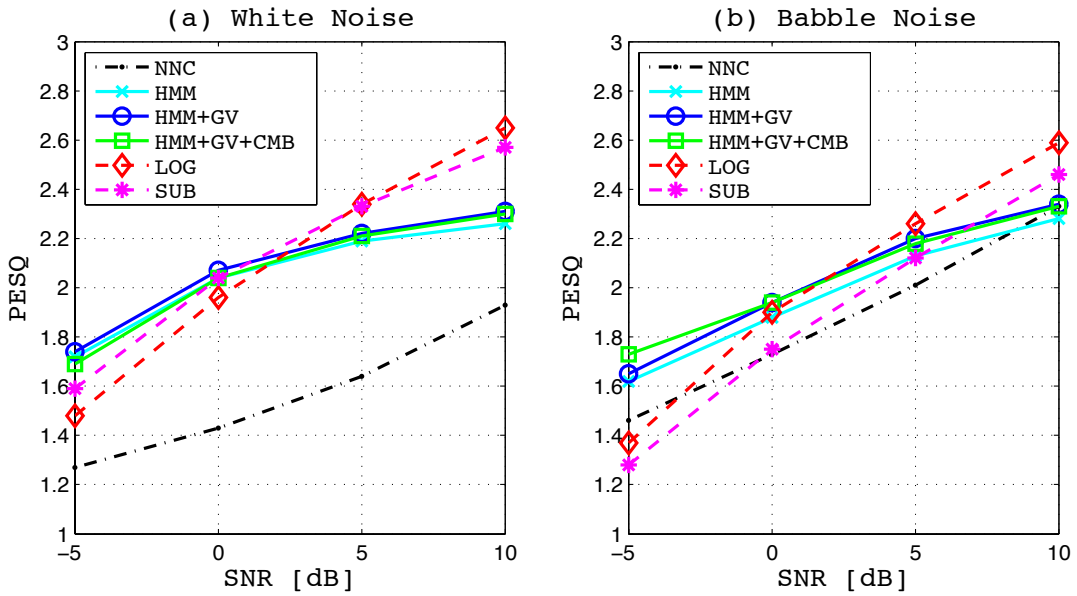

Figure 7.1: PESQ scores of the proposed HMM-based speech enhancement at different SNRs comparing with log MMSE and the subspace method. a) shows the performance in white noise while b) shows the performance in babble noise.

scores at SNRs of 0 dB and below whereas their scores are lower than the conventional approaches (LOG and SUB) at SNRs of 5 dB and above.

In babble noise, the HMM-based methods (HMM, HMM+GV and HMM+GV+CMB) show the highest performance at SNRs of 0 dB and below as well as the case of white noise. Specifically, the superiority to other methods becomes more remarkable at -5 dB while the scores of filtering methods (LOG and SUB) become lower than NC.

Comparing the performance among the configurations of HMM-based methods, HMM+GV always achieves higher scores than HMM in both white noise and babble noise. This is attributed to effectiveness of the global variance model in compensation for over-smoothing of the HMM-based speech parameters. The difference between HMM+GV and HMM+GV+CMB is not substantial at SNRs of 0 dB and above. The score of HMM+GV+CMB is, however, superior to HMM+GV at -5 dB in babble noise as opposed to the case of white noise in which the score of HMM+GV+CMB becomes lower than HMM+GV at -5 dB. This is interesting because combining the HMM-based speech with log MMSE in babble noise is more effective than in white noise even though

the performance of log MMSE is higher in white noise than in babble noise. This is attributed to the fact that the accuracy of HMM decoding at SNR of -5 dB is much lower in babble noise than in white noise (Figure 5.4) and it brings a loss of speech to the output which can effectively be supplemented with log MMSE with the confidence measure.

7.3.2 Speech Intelligibility

Table 7.4 and Figure 7.2 show the NCM scores of the different configurations of the proposed method (HMM, HMM+GV and HMM+GV+CMB) at SNRs from -5 dB to 10 dB in white noise and babble noise comparing with baseline performance (NNC) and the conventional filtering approaches (LOG and SUB).

Noise	NNC	HMM	HMM+GV	HMM+GV+CMB	LOG	SUB
White Noise						
10 dB	0.87	0.69	0.68	0.68	0.87	**0.93**
5 dB	0.76	0.68	0.67	0.67	0.78	**0.86**
0 dB	0.60	0.66	0.65	0.63	0.64	**0.74**
-5 dB	0.41	**0.60**	0.58	0.52	0.46	0.56
Babble Noise						
10 dB	0.86	0.70	0.68	0.67	0.87	**0.92**
5 dB	0.70	0.69	0.67	0.62	0.72	**0.80**
0 dB	0.52	**0.65**	0.63	0.51	0.53	0.62
-5 dB	0.32	**0.57**	0.56	0.38	0.31	0.38

Table 7.4: NCM scores at SNRs of 10 dB, 5 dB, 0 dB and -5 dB in white noise and babble noise

HMM and HMM+GV show a characteristic to have stable performance over the SNR range in both white noise and babble noise with slight reduction at -5 dB while the scores of other methods fall more rapidly. These characteristics bring the highest scores to HMM at lower SNRs, i.e. -5 dB in white noise and 0 dB and -5 dB in babble noise.

Comparing the performance among the configurations of HMM-based methods (HMM, HMM+GV and HMM+GV+CMB), HMM always shows slight higher scores than HMM+GV as opposed to the performance in terms of PESQ. The performance of HMM+GV+CMB is influenced by HMM+GV at higher SNRs whereas it is more dominated by the performance of log MMSE at lower SNRs. Consequently, the performance

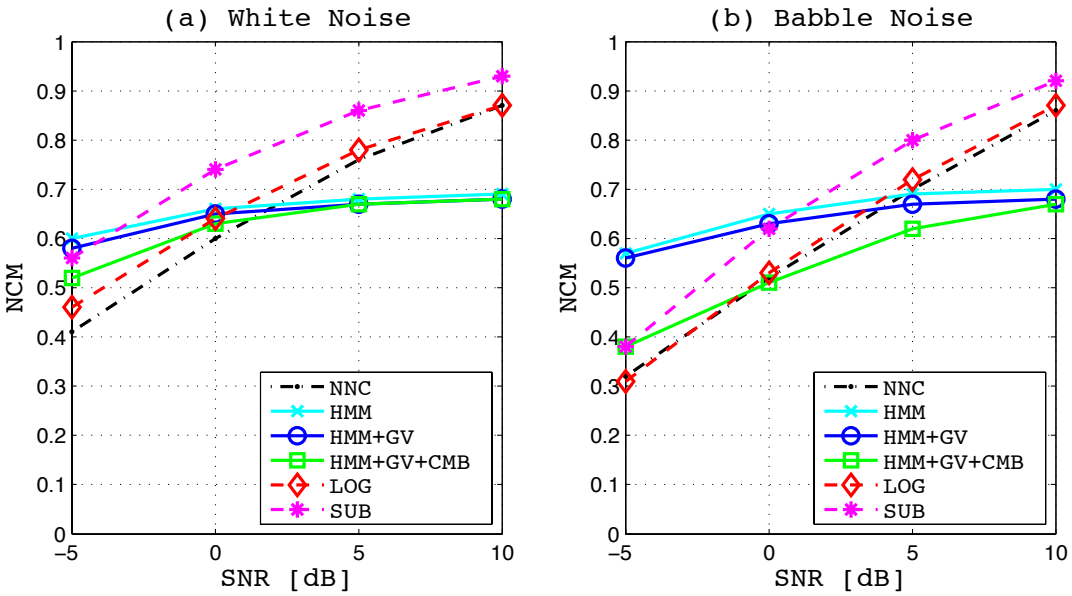

Figure 7.2: NCM scores of the proposed HMM-based speech enhancement at different SNRs comparing with log MMSE and the subspace method. a) shows the performance in white noise while b) shows the performance in babble noise.

of HMM+GV+CMB is always worse than HMM and HMM+GV in both white noise and babble noise in terms of NCM implying the compensation for over-smoothing with the global variance model and for decoding errors are not effective to improve NCM score.

7.4 Subjective Evaluation

Subjective listening tests are now carried out in addition to the objective comparative evaluation in the previous section. Test speech enhanced by the preceding procedure (HMM, HMM+GV and HMM+GV+CMB) is subjectively evaluated in this section by listening tests for speech quality and intelligibility comparing with baseline performance given by no noise compensation (NNC), and logMMSE (LOG), which represents the conventional filtering approach to speech enhancement.

7.4.1 Speech Quality

For subjective evaluation of speech quality, three-way mean opinion score (MOS) listening tests in which a subject listens to speech once and then grades it as 5 (Excellent), 4 (Good), 3 (Fair), 2 (Poor) or 1 (Bad) in terms of each of background noise, speech

distortion and overall quality are carried out for 10 subjects. An utterance is first randomly selected from the test speech (200 utterances × 4 speakers) and contaminated with white noise or babble noise at an SNR of -5 dB, 0 dB, 5 dB or 10 dB. The speech is then enhanced by one of HMM, HMM+GV, HMM+GV+CMB, LOG or NNC and then a subject listens to it through headphones (Sennheiser: HD-495) at a noise-free condition in a quiet room in order to grade the enhanced speech by the three-way MOS test. Test guidance including all the information needed for the test is given to the subjects prior to the start of the test and the subjects can adjust the volume level of speech to comfortable level during the test. This test is repeated for 120 utterances for each subject. Figure 7.3 shows the user interface of the three-way MOS listening test.

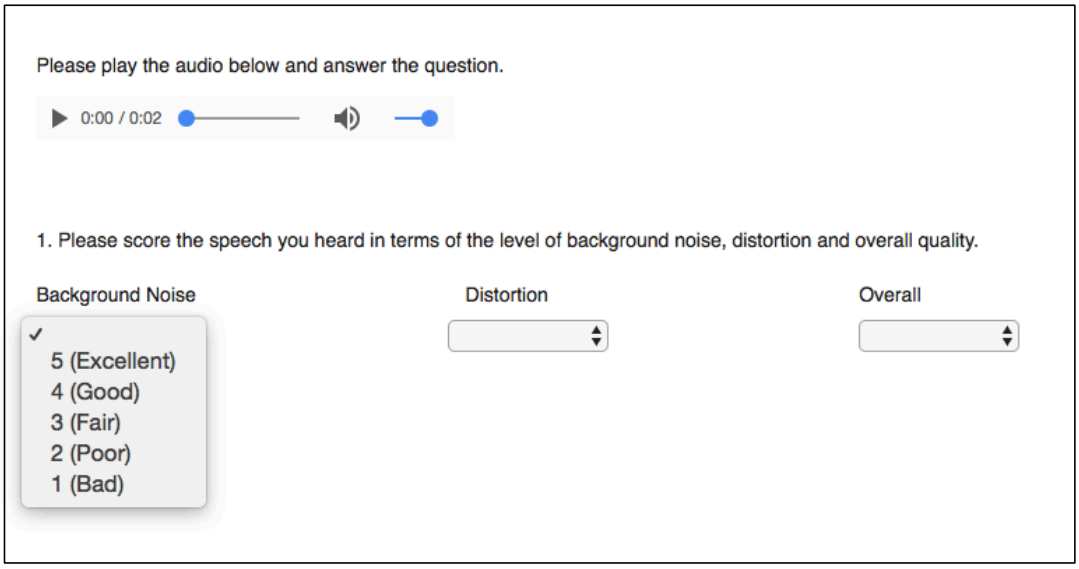

Figure 7.3: The user interface of the three-way MOS listening test.

Tables 7.5 - 7.7 and Figure 7.4 show the scores of the three-way MOS listening test for each configurations of the proposed HMM-based speech enhancement (HMM, HMM+GV and HMM+GV+CMB), log MMSE (LOG) and no noise compensation (NNC) in white noise and babble noise. The tables and figure also show the performance of LOG and no noise compensation NNC as a representative of the conventional filtering-based methods and the baseline performance with which the performance of the proposed methods are compared.

Noise	NNC	HMM	HMM+GV	HMM+GV+CMB	LOG
White noise					
10 dB	1.95	4.67	4.62	**4.71**	3.29
5 dB	1.33	4.57	**4.71**	4.43	2.81
0 dB	1.33	4.57	**4.62**	3.71	2.52
-5 dB	1.00	4.48	**4.57**	3.00	1.71
Babble Noise					
10 dB	2.29	4.24	**4.71**	4.29	2.86
5 dB	1.90	**4.57**	**4.57**	4.48	2.48
0 dB	1.57	4.43	**4.71**	4.57	2.33
-5 dB	1.29	4.24	**4.48**	3.90	1.86

Table 7.5: Subjective listening scores focused on background noise at SNRs from -5 dB to 10 dB in white noise and babble noise.

Noise	NNC	HMM	HMM+GV	HMM+GV+CMB	LOG
White noise					
10 dB	**3.86**	3.67	3.81	3.43	**3.86**
5 dB	3.48	**3.71**	3.52	3.33	3.24
0 dB	3.19	**3.38**	3.05	**3.38**	2.95
-5 dB	2.67	**3.19**	3.14	3.14	2.38
Babble Noise					
10 dB	**4.05**	3.57	3.81	3.43	3.57
5 dB	**4.05**	3.33	3.62	3.67	3.33
0 dB	3.29	3.71	**3.76**	3.71	3.05
-5 dB	3.29	**3.43**	3.30	3.24	2.43

Table 7.6: Subjective listening scores focused on signal distortion at SNRs from -5 dB to 10 dB in white noise and babble noise.

Noise	NNC	HMM	HMM+GV	HMM+GV+CMB	LOG
White noise					
10 dB	2.71	3.90	**4.05**	4.00	3.57
5 dB	2.10	**3.95**	**3.95**	3.52	3.00
0 dB	2.05	**3.86**	3.67	3.48	2.76
-5 dB	1.43	**3.67**	3.52	3.14	1.95
Babble Noise					
10 dB	3.14	3.62	**4.05**	3.86	3.14
5 dB	2.81	3.67	**4.10**	4.05	2.71
0 dB	2.33	3.86	**4.24**	4.00	2.71
-5 dB	2.10	3.57	**3.78**	3.48	2.00

Table 7.7: Subjective listening scores as the overall grade of speech at SNRs from -5 dB to 10 dB in white noise and babble noise.

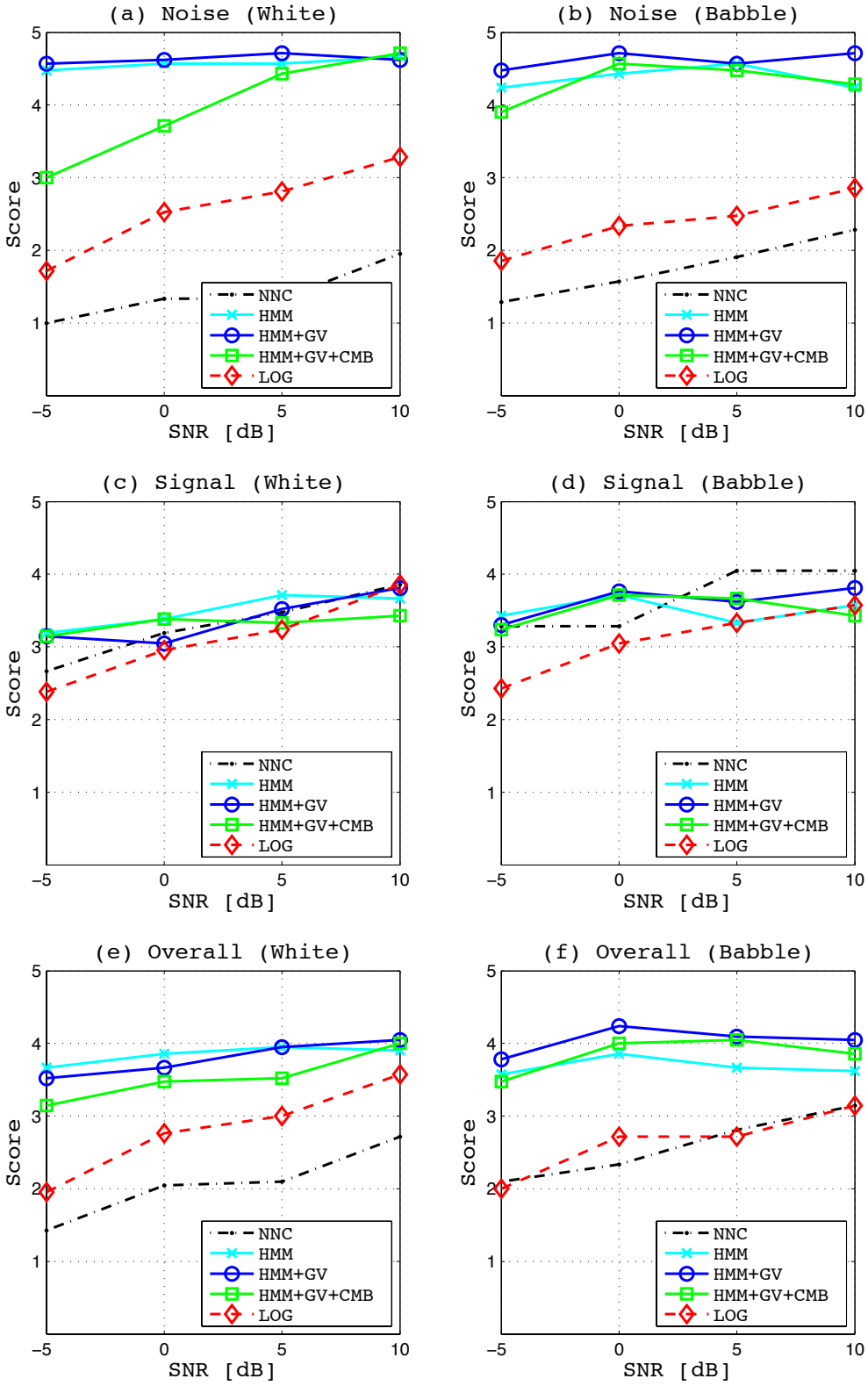

Figure 7.4: Test scores of the three-way MOS listening test with different configurations of speech enhancement. a) and b) show the scores with respect to background noise in white noise and babble noise. c) and d) show the scores focused on signal distortion while e) and f) represent overall speech quality.

Table 7.5 and Subplots (a) and (b) in Figure 7.4 show the listening scores focused on background noise of enhanced speech. In this criterion, HMM and HMM+GV keep very high grades between 5 (Excellent) and 4 (Good) over the range of SNRs. LOG performs explicitly lower than HMM and HMM+GV and the scores fall further at lower SNRs even though they are always higher than NNC over the SNR range. HMM+GV+CMB is graded as high as HMM and HMM+GV at SNRs of 5 dB and above but it reduces the score at lower SNRs by being combined with log MMSE specifically in white noise. This is attributed to a rise of false positive errors of confidence measure. False positive errors replace HMM-based speech with log MMSE at speech segments which do not need to be replaced. This is equivalent to adding noise into clean speech and consequently, it reduces the score.

Table 7.6 and Subplots (c) and (d) in Figure 7.4 show the scores with respect to the signal distortion of enhanced speech in white noise and babble noise. The grade of HMM-based speech enhancement, i.e. HMM, HMM+GV and HMM+GV+CMB, is comparable to NNC, which is not degraded in terms of speech distortion, even at high SNRs such as 5 dB and above in addition to showing the robustness against noise. The proposed methods are also superior to log MMSE over the SNR range, but 10 dB in white noise, in this criterion as well.

The overall evaluation is represented by Table 7.7 and Subplots (e) and (f) in Figure 7.4 showing that the performance of the proposed HMM-based speech enhancement surpasses log MMSE and NNC overwhelmingly over all the noise conditions. Specifically, HMM+GV keeps the performance in around the grade of 4 (Good) over the SNR range. The difference in overall performance between HMM and HMM+GV is not significant in white noise but more effectiveness of using the global variance model is shown in babble noise. It seems that the only factor which potentially gives different characteristics to the output, depending on noise conditions, is the HMM decoding results. It is, however, not identified from a comparison of the decoding results in white noise with those in babble noise shown in Figure 5.4. The scores of HMM+GV+CMB are always lower than HMM+GV but the significant decline in the noise intrusiveness score in white noise (Subplot A) is reduced in the overall evaluation.

In order to evaluate significant differences among the algorithms, two-way analy-

sis of variance (ANOVA) over the algorithms and noise is applied and then a multiple comparison tests according to Tukey's least significant difference (LSD) test [118] are examined. Table 7.8 shows pairwise comparison of p-values of the four algorithms (HMM / HMM+GV / HMM+GV+CMB / LOG) and NNC over all SNR conditions. The first column of the table shows each pair of the algorithms while the second, third and fourth columns correspond to the background noise scores, the distortion scores and the overall scores.

Pairs of Algorithms	Background Noise	Signal Distortion	Overall
NNC / HMM	0.0000	0.9997	0.0000
NNC / HMM+GV	0.0000	0.9860	0.0000
NNC / HMM+GV+CMB	0.0000	0.9549	0.0000
NNC / LOG	0.0000	0.0004	0.0000
HMM / HMM+GV	0.2094	0.9549	0.8627
HMM / HMM+GV+CMB	0.0000	0.8969	0.8939
HMM / LOG	0.0000	0.0002	0.0000
HMM+GV / HMM+GV+CMB	0.0000	0.9997	0.3214
HMM+GV / LOG	0.0000	0.0029	0.0000
HMM+GV+CMB / LOG	0.0000	0.0059	0.0000

Table 7.8: Pairwise p-values of the algorithms over all SNR conditions.

In terms of the subjective scores of background noise, all the combinations of the algorithms except the pair of HMM with HMM+GV show significant effect ($p < 0.005$). This implies that each algorithm can be effective in reducing background noise because p-values of each algorithm paired with NNC are nearly zero, but the effectiveness of HMM and HMM+GV is similar. Focusing on the scores of signal distortion, only the combinations paired with LOG show significant effect. This means speech enhanced by HMM-based algorithms can retain the speech signal effectively whereas the log MMSE algorithm produces more signal distortion. p-values of the overall scores show that HMM, HMM+GV and HMM+GV+CMB are not significantly different to each other though all of the four algorithms are significantly effective as enhancing the speech.

7.4.2 Speech Intelligibility

For subjective evaluation of speech intelligibility, subjective word recognition tests in which a subject listens to speech once and then selects words in the utterance are carried

out for 10 subjects. An utterance is first randomly selected from the GRID test speech (200 utterances × 4 speakers) and contaminated with white noise or babble noise at an SNR of -5 dB or 0 dB (Noise conditions at SNRs of 5 dB and 10 dB are omitted since a preliminary experiment has shown that speech intelligibility is not affected at SNRs of 5 dB and above). The speech is then enhanced by one of HMM, HMM+GV, HMM+GV+CMB, LOG or NNC and then a subject listens to it through headphones (Sennheiser: HD-495) at a noise-free condition in a quiet room in order to select words in the utterance to which he/she listened. Test guidance including all the information needed for the test is given to the subjects prior to the start of the test and the subjects can adjust the volume level of speech to comfortable level during the test. This test is repeated for 60 utterances for each subject. Figure 7.5 shows the user interface of the subjective word recognition test in which the word options are placed in accordance with the GRID grammar, i.e. *verb* → *blue* → *preposition* → *alphabet* → *number* → *adverb*.

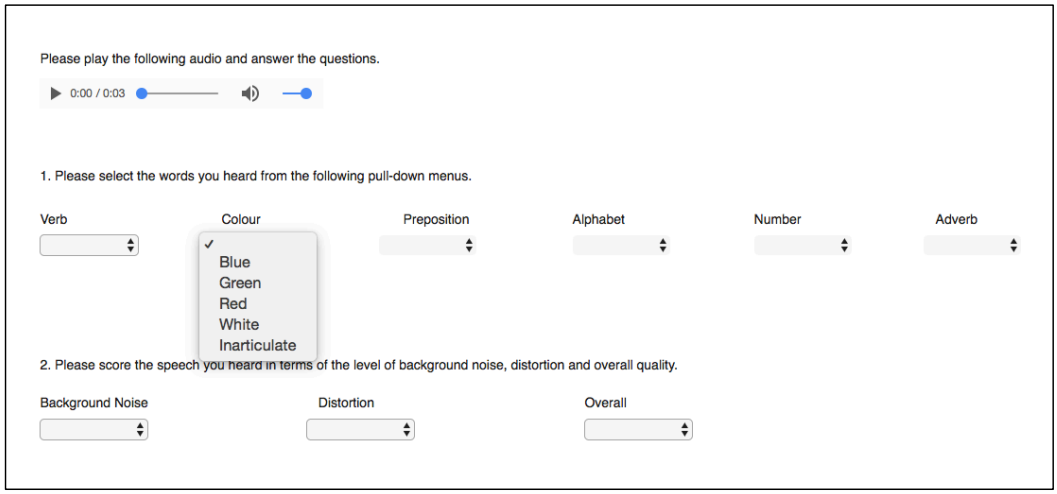

Figure 7.5: The user interface of the subjective word recognition test.

This user interface also includes questions for the speech quality test in addition to the questions for the speech intelligibility test to measure speech quality and intelligibility in parallel for rationalisation purpose. Therefore, the subjects measure only speech quality for 60 utterances whose SNR are 5 dB or 10 dB while they measure both speech quality and intelligibility for the other 60 utterances whose SNR are -5 dB or 0 dB. This makes different test conditions depending on SNR of utterances. However, at given SNR,

all methods of enhancement are evaluated identically.

The test results are evaluated as Correct Answer Rate, $W_{correct}$, which is determined as follows as a subjective speech intelligibility.

$$W_{correct} = \frac{\text{The number of words recognised correctly in a utterance}}{\text{The number of words in a utterance}} \tag{7.53}$$

Table 7.9 and Figure 7.6 show $W_{correct}$ of the subjective word recognition test for each configurations of the proposed HMM-based speech enhancement in different noise conditions. They also show the performance of LOG and NNC which represent the conventional filtering-based approach to speech enhancement and the baseline performance respectively for comparison.

Noise	NNC	HMM	HMM+GV	HMM+GV+CMB	LOG
White noise					
0 dB	0.87	0.88	0.84	0.78	**0.91**
-5 dB	0.73	0.78	**0.81**	0.80	0.77
Babble Noise					
0 dB	0.94	0.90	0.94	0.81	**0.95**
-5 dB	**0.88**	0.82	0.84	0.78	0.76

Table 7.9: Correct answer rates of the subjective word recognition test at SNRs of -5 dB and 0 dB in white noise and babble noise.

The test results show that HMM-based speech enhancement keeps intelligibility flat at SNRs between -5 dB and 0 dB as compared with LOG and NNC as well as the objective scores in NCM. This characteristic brings higher Correct Answer Rate of HMM-based methods at -5 dB in both white noise and babble noise than LOG, but lower at 0 dB.

The results also show that in babble noise, NNC shows better intelligibility than the enhanced speech at SNRs between -5 dB and 0 dB. It seems to be attributed to the difference of frequency bands between noise and speech. Specifically, test utterances of female speech are relatively easy to be recognised correctly even at -5 dB in babble noise since the babble noise in the test set is dominated by male voice.

Comparing the performance among HMM-based methods, HMM+GV obtains higher Correct Answer Rate than HMM at -5 dB in white noise and -5 dB and 0 dB in babble noise. The difference is, however, not significant. HMM+GV+CMB shows always lower

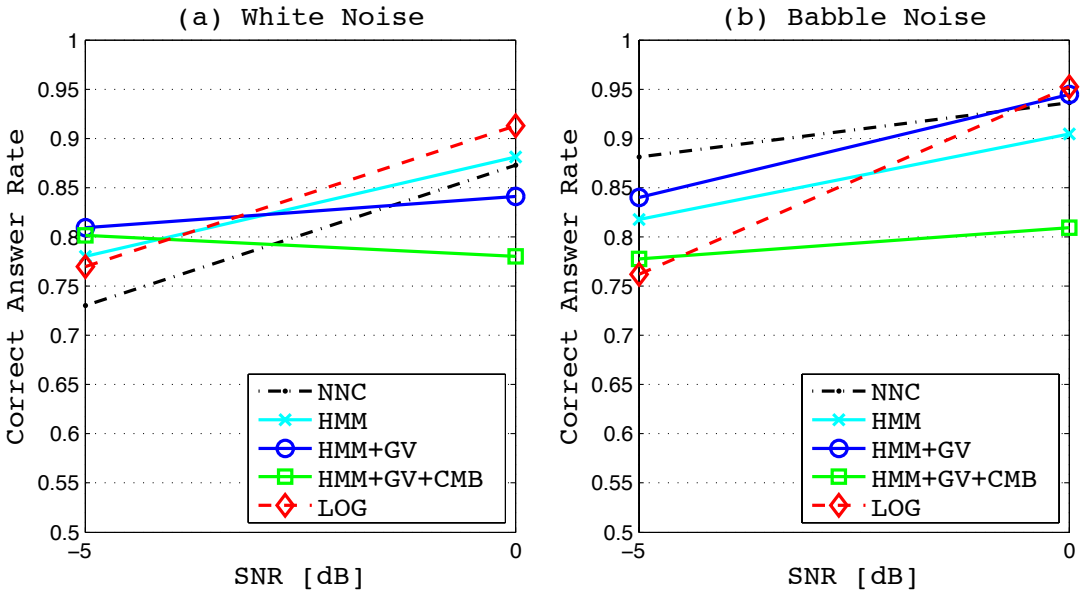

Figure 7.6: Correct answer rates of the subjective word recognition test at SNRs of -5 dB and 0 dB in a) white noise and b) babble noise.

intelligibility than HMM+GV in both white noise and babble noise. This gives a notion that to achieve higher decoding accuracy is more effective than to compensate the erroneous decoding results to improve the performance of HMM-based speech enhancement.

In order to evaluate significant differences among the algorithms, two-way ANOVA over the algorithms and noise is applied and then a multiple comparison tests according to Tukey's LSD test are examined. Table 7.10 shows pairwise comparison of p-values of the four algorithms (HMM / HMM+GV / HMM+GV+CMB / LOG) and NNC over all SNR conditions. p-values show that all of the algorithms except HMM+GV+CMB do not have significant difference in comparison to NNC. This implies that the intelligibility of the enhanced speech produced by HMM, HMM+GV and LOG is no better than NNC. However, the intelligibility using HMM+GV+CMB brings more deterioration in speech intelligibility.

7.5 Conclusion of the Chapter

This chapter has examined the proposed HMM-based speech enhancement by objective and subjective tests to evaluate total performance of the proposed method comparing

Pairs of Algorithms	p-values
NNC / HMM	0.4004
NNC / HMM+GV	0.5524
NNC / HMM+GV+CMB	0.0006
NNC / LOG	0.9993
HMM / HMM+GV	0.9993
HMM / HMM+GV+CMB	0.1674
HMM / LOG	0.5524
HMM+GV / HMM+GV+CMB	0.0969
HMM+GV / LOG	0.7060
HMM+GV+CMB / LOG	0.0017

Table 7.10: Pairwise p-values of the algorithms over all SNR conditions.

with baseline performance given by no noise compensation and methods representing the conventional filtering approach to speech enhancement.

The objective tests have shown that the proposed HMM-based speech enhancement has significant superiority to the conventional methods in terms of PESQ, which represents an objective score of speech quality, at low SNR conditions such as 0 dB and below in white noise and babble noise. Additionally, the performance in PESQ is further improved over the noise conditions by applying the compensation for over-smoothing in the HMM-based speech parameters by global variance model. Moreover, the compensation for decoding errors using the proposed confidence measure works effectively in terms of PESQ when the enhanced speech has losses of speech by decoding errors.

Alternatively, in terms of NCM which represents an objective score of speech intelligibility, the performance of the proposed HMM-based speech enhancement keeps stable over the noise conditions and consequently, it is superior to the conventional methods at low SNR conditions such as -5 dB in white noise and 0 dB and below in babble noise. NCM score is, however, reduced by applying the compensation for either decoding errors or over-smoothing.

The subjective tests for speech quality and intelligibility have also been carried out. These tests have revealed the superiority of the proposed speech enhancement more explicitly. The overall evaluation of the proposed HMM-based methods in the three-way MOS listening test surpass both log MMSE, which represents the conventional filtering-based methods, and no noise compensation over all the noise conditions. Similarly, the

subjective word recognition also showed the superiority of the proposed methods in speech intelligibility to log MMSE in low SNR conditions such as -5 dB.

Incidentally, it is discovered that PESQ tends to give relatively harsh scores to HMM-based speech enhancement as compared with the scores given to log MMSE from comparison of the test results between subjective and objective tests. This is considered to be attributed to the time alignment process in the PESQ computation in which time delay values between original and degraded signals are gauged [99]. In this sense of time allocation in the enhanced signals, reconstruction-based speech enhancement has disadvantageous nature as compared with the filtering-based approaches because the time allocation of enhanced speech is formed from only statistical information in the decoding process while the filtering-based methods exploit the original signal. However, at least it seems that the proposed HMM-based speech enhancement has overwhelming superiority to the conventional methods in enhancing speech which has been degraded by additive noise as shown by the subjective test results.

Chapter 8

Conclusions and Further Work

As stated at the beginning of the thesis, the purpose of this work was to develop a novel method to enhance speech signals degraded by additive noise in which the only accessible information is monaural noisy speech. This chapter concludes the thesis by first reviewing the work discussed in this thesis and then identifying the key findings. Finally, suggestions of further work are presented.

8.1 Review

This section reviews the work discussed in this thesis. Chapter 1 first introduced the area of speech enhancement problems that need to be addressed. The basic architecture of the proposed method was then presented.

A variety of the conventional methods for speech enhancement including the spectral subtraction, Wiener filters, statistical-model-based filtering methods and subspace algorithms which are based on the filtering approaches are discussed in Chapter 2. Experimental analysis has then shown that the log MMSE method, which represents the statistical model-based filtering methods, generally shows the best performance as the overall evaluation in terms of PESQ. The experiments also demonstrated that the filtering-based methods can leave musical noise, residual noise and distortion at low SNRs such as 0 dB and below which are attributed to underestimation and overestimation of the noise in the filtering-based methods. The reconstruction-based approaches to speech enhancement including corpus and inventory-based speech enhancement and model-based speech

enhancement are then discussed prior to Chapter 3.

The proposed HMM-based speech enhancement is achieved by using a speech production model to reconstruct clean speech from speech features synthesised statistically. Therefore, Chapter 3 first discussed the human speech production process and properties of the speech signal attributed to the production process. The source-filter model was then discussed with respect to the preceding human speech production process and the properties of speech signals. Alternatively, the sinusoidal model and the HNM, which is one of variants of the sinusoidal model, have also been studied as another approach to model voiced speech. These models are based on the notion where voiced speech is modelled as a summation of harmonic sinusoids. The sinusoidal model showed very good quality in speech production at a brief speech reconstruction experiment. However, it turned out that the features required by the sinusoidal model to produce speech are less suitable for building statistical models than the source filter model because of their variability. Consequently, the STRAIGHT vocoder, which is a variant of the source-filter model using the mixed-excitation model, was selected for the speech reconstruction process in the proposed HMM-based speech enhancement from the aspect of its good performance as a speech production model. At the end of the chapter, different methods to estimate the fundamental frequency of speech are explored and the experiments have shown that PEFAC has a distinct advantage over RAPT and YIN in performance for estimating the fundamental frequency of speech in noisy condition.

Then, the topic reached the detail of HMM-based speech enhancement in Chapter 4. The chapter first gave an overview of HMMs and the theories were then extended to the practical applications such as ASR and HMM-based speech synthesis. The latter part of the chapter explored HMM-based speech enhancement, achieved by combining the techniques of HMM training, HMM decoding and HMM synthesis with the STRAIGHT speech production model. Experiments evaluated the performance of speech enhancement with different sets of configurations, comparing with the log MMSE method which represents the conventional filtering methods. The experimental analysis has shown that using the context-dependent triphone HMMs with no grammar constraints with 5 ms-frame interval achieves quality and intelligibility sufficiently close to that with grammar constrained whole-word models in terms of PESQ and NCM, which represent objec-

tive measure of speech quality and intelligibility, but puts no restrictions on the input speech. Compared to conventional methods of speech enhancement, it turned out that HMM-based speech enhancement shows higher PESQ and NCM scores at lower SNRs.

The first experiment of HMM-based speech enhancement carried out in Chapter 4 had exploited *a priori* knowledge of noise at the HMM training stage in order to use noise-matched HMMs in the decoding process. Chapter 5 first discussed the problems brought by employing noise-matched HMMs and then studied the theory of HMM adaptation to model noisy speech using parallel model combination to address the problems. In parallel model combination, the mismatch function between clean speech and noisy speech is determined in the linear Mel-filterbank domain while HMMs had been modelled in the MFCC domain. Therefore, the non-linearity between these domains needs to be resolved and the distribution mapping between Gaussian and log-normal, and the unscented transform were discussed to tackle this problem. Experiments then evaluated the performance of HMM-based speech enhancement with noise-adapted HMMs compared to the methods using noise-matched HMMs and log MMSE. The experimental analysis showed that HMM adaptation to model noisy speech with parallel model combination is effective to achieve both noise robust decoding and to obtain state sequences which match the clean HMMs as a practical method. Consequently, the upper limits of PESQ and NCM scores were raised.

The method to improve the decoding accuracy was discussed in Chapter 5 but when decoding errors occur, wrong speech segments are produced in the reconstructed speech. This may reduce speech intelligibility of the output speech. Therefore, the former part of Chapter 6 discussed confidence measuring to first identify and then to compensate for the influence of decoding errors. A novel method to measure frame-by-frame and phoneme-by-phoneme confidence was studied and then enhanced speech was produced by combining HMM-based speech with log MMSE according to the phoneme-by-phoneme confidence and evaluated. The proposed confidence measure and combined speech improved PESQ scores at low SNRs, specifically in babble noise. The benefit was, however, limited and NCM scores were rather decreased by introducing log MMSE segments which include residual and musical noise. To improve the performance of the proposed HMM-based speech enhancement from another point of view, the chapter then explored

a method to reduce over-smoothing in the synthesised parameters by using the global variance model. The evaluation in clean and noisy conditions showed that the method using the global variance is effective to improve the baseline performance of the proposed method in terms of PESQ.

Finally, the proposed HMM-based speech enhancement was examined by objective and subjective tests in Chapter 7 to evaluate total performance of the proposed method comparing with baseline performance given by no noise compensation and log MMSE representing the conventional filtering approach. The objective tests have shown that the proposed HMM-based speech enhancement has significant superiority to the conventional methods in terms of PESQ at low SNR conditions in white noise and babble noise. Additionally, the performance in PESQ is further improved over the noise conditions by applying the global variance model. The objective tests were also evaluated in terms of NCM, which represents an objective score of speech intelligibility. The performance of the proposed HMM-based speech enhancement kept stable over the noise conditions and consequently, it brought the proposed method superiority to the conventional methods at low SNR conditions. Then the subjective tests for speech quality and intelligibility have been carried out. These tests have revealed the superiority of the proposed speech enhancement further explicitly. The overall evaluation of the proposed HMM-based methods in the three-way MOS listening test surpassed both log MMSE and no noise compensation over all the noise conditions. Similarly, the subjective word recognition test also showed an advantage of the proposed methods in speech intelligibility to log MMSE in low SNR conditions such as -5 dB.

8.2 Key Findings

The main finding of this work is that HMM-based speech enhancement can produce enhanced speech that is either better or comparable to speech produced from conventional methods of speech enhancement. Specifically, the HMM-based enhancement is more effective at low SNRs where conventional methods break down. Several other key findings have also been discovered and are highlighted below.

8.2.1 Speech Production Model and Features

The STRAIGHT vocoder was employed as the speech production model in this work and this model has shown good performance with the parameters synthesised from statistical models (i.e., HMMs). Originally, STRAIGHT requires a smoothed time-frequency spectral surface, a time-frequency aperiodicity measure and a fundamental frequency contour of speech for speech reconstruction [71]. To apply statistical models of speech to the STRAIGHT speech production model, Mel-cepstrum-based acoustic features, which are known to model speech effectively [88], such as Mel-Generalised Cepstrum-Based Line Spectrum Pair (MGC-LSP), have successfully been applied to various text-to-speech (TTS) applications, in which MGC-LSP vectors synthesised from trained HMMs are converted to the set of STRAIGHT features and then reconstructed by STRAIGHT [119]. The work proposed in this thesis, however, needs more statistically discriminative acoustic features because this work also requires the HMM decoding process to acquire a model and state sequence from noisy speech unlike TTS applications. Therefore, MFCCs, which have successfully been applied to practical ASR applications [89, 90], were employed as the spectral feature in this work to achieve good performance at low SNR conditions. As a result it was found that using MFCCs as a speech feature for accurate HMM decoding and using STRAIGHT for high quality speech reconstruction is effective to achieve high quality HMM-based speech enhancement even at low SNR conditions such as -5 dB.

8.2.2 Unconstrained Speech Input

Statistical models of speech were configured as context-dependent (CD)-triphone HMMs in this work to avoid any speech constraints placed on the enhancement system that whole-word models or explicit language models would impose. CD-triphone HMMs resolve the lack of variety which monophone models hold. Additionally, they can avoid either overfitting or underfitting by employing the tree-based clustering [94] which also enables untrained models to be classified into the clusters. Therefore, the restriction of vocabulary which is a one of the biggest problem on whole-word HMMs is also resolved. Since the possible sequences of CD-triphone HMMs are constrained by the context of speech, i.e. the previous phoneme and the next phoneme, the decoding accuracy of CD-

triphone HMMs with no language model is comparable to whole-word HMMs using a language model as shown in Figure 4.18. Moreover, Table 4.15 has reported that the performance of CD-triphone on speech synthesis is also the same level as whole-word HMMs. Therefore, using CD-triphone HMMs is an important choice from the aspect of both performance and practicality.

8.2.3 Noise Robustness

The techniques to adapt clean HMMs to model noisy speech, which include parallel model combination, the Gaussian-log normal mapping and the unscented transform are also vital to achieve good performance and practicality. HMM adaptation using parallel model combination has enabled this work to be robust to noise without noise information and noise matched models *a priori*, and it also raised the baseline performance by resolving the problem brought by inconsistency in the state allocation between clean HMMs and noise-matched HMMs which is referred to as Figure 5.1. The effectiveness of the HMM adaptation is shown in Figures 5.5 and 5.6.

8.2.4 Further Impruvement in Speech Quality

To give further refinement to the proposed method, compensation for over-smoothing in the synthesised parameters were then applied. Specifically, the speech synthesis algorithm using the global variance model, referred to as Equation (6.20), was implemented to emphasise the formants in the synthesised parameters and then examined. The test results have shown that this refinement is effective to raise the baseline performance of the system in terms of both PESQ and subjective quality tests as shown in Figure 7.1 and Table 7.7. Additionally, a new method to mitigate the influence of decoding errors was also applied. This method is a two stage process of first identifying errors and secondly applying compensation. The overview of the first stage is illustrated in Figure 6.1 while the stage 2 is depicted in Figure 6.2. As a total evaluation, this method improved neither objective nor subjective performance as shown Figures 7.1 - 7.6 because as long as some segments in HMM-based speech, which does not contain background noise, are replaced with non-clean speech, improvement of the quality and intelligibility is not be expected regardless of whether the confidence measure is true or false. However, the first stage of

the method, i.e. the process to identifying decoding errors, has shown meaningful results, especially at low SNRs such as -5 dB as shown in Figure 6.3, and it may have a use other applications.

In conclusion, this work achieved HMM-based speech enhancement which shows significantly high performance by integrating the preceding techniques.

8.3 Further Work

This section proposes some suggestions for further work which may improve the proposed method of HMM-based speech enhancement.

8.3.1 DNN-HMM

Recently, Deep Neural Network (DNN) with multiple hidden layers has successfully been applied to speech processing applications. For example, ASR which uses context dependent deep neural network hidden Markov Models (CD-DNN-HMMs) has been proposed and has shown that it significantly outperforms HMMs while statistical parametric speech synthesis which replaces the decision tree-based clustering process with DNN has been presented for text-to-speech applications [120, 121]. Applying these technique to this work can be challenging topic to achieve technical breakthrough.

8.3.2 Speech Production Model

Recent research has evaluated a wide range of vocoders and reported an experimental comparison [66]. The report was aimed at statistical parametric speech synthesis (SPSS) but it can be a good reference to examine and employ other speech production models apart from STRAIGHT for HMM-based speech enhancement. An improved speech production model is likely to improve the resulting speech at both high and low SNRs.

8.3.3 Non-Stationary Noise Model

In the noise adaptation process, the current work has assumed that the noise data is stationary over the entire length of the utterance. Therefore, the decoding accuracy may be improved by employing a time varying non-stationary noise model specifically in

babble noise or other non-stationary noise.

Bibliography

[1] P. C. Loizou, *Speech Enhancement: Theory and Practice*, 2nd ed. CRC Press, 2013.

[2] P. C. Loizou and G. Kim, "Reasons why current speech-enhancement algorithms do not improve speech intelligibility and suggested solutions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 47–56, March 2011.

[3] J. Meyer and K. U. Simmer, "Multi-channel speech enhancement in a car environment using Wiener filtering and spectral subtraction," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 1167–1170, April 1997.

[4] I. Almajai and B. Milner, "Enhancing audio speech using visual speech features," *Proceedings of INTERSPEECH*, pp. 1959–1962, September 2009.

[5] A. Kato and B. Milner, "Using hidden Markov models for speech enhancement," *Proceedings of INTERSPEECH*, pp. 2695–2699, 2014.

[6] A. Kato and B. Milner, "HMM-based speech enhancement using sub-word models and noise adaptation," *Proceedings of INTERSPEECH*, pp. 3748–3752, September 2016.

[7] P. Harding and B. Milner, "Reconstruction-based speech enhancement from robust acoustic features," *Speech Communication*, vol. 75, pp. 62–75, December 2015.

[8] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 373–376, May 1996.

[9] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, November 2009.

[10] J. Ming, R. Srinivasan, and D. Crookes, "A corpus-based approach to speech enhancement from nonstationary noise," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 822–836, May 2011.

[11] J. Ming and D. Crookes, "Speech enhancement from additive noise and channel distortion - a corpus-based approach," *Proceedings of INTERSPEECH*, pp. 2710–2714, September 2014.

[12] X. Xiao and R. M. Nickel, "Speech enhancement with inventory style speech resynthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1243–1257, August 2010.

[13] J. L. Carmona, J. Barker, A. M. Gómez, and N. Ma, "Speech spectral envelope enhancement by HMM-based analysis/resynthesis," *IEEE Signal Processing Letters*, vol. 20, no. 6, pp. 563–566, June 2013.

[14] D. Erro, I. Sainz, I. Saratxaga, E. Navas, and I. Hernaez, "MFCC+F0 extraction and waveform reconstruction using HNM: Preliminary results in an HMM-based synthesizer," *Proceedings of FALA*, pp. 29–32, 2010.

[15] K. El-Maleh and P. Kabal, "Comparison of voice activity detection algorithms for wireless personal communications system," *IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 470–473, May 1997.

[16] K. Srinivasan and A. Gersho, "Voice activity detection for cellular networks," *IEEE Speech Coding Workshop*, pp. 85–86, 1993.

[17] D. Freeman, G. Cosier, C. Southcott, and I. Boyd, "The voice activity detector for the Pan-European digital cellular mobile telephone service," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 369–372, 1989.

[18] S. Sasaki and I. Matsumoto, "Voice activity detection and transmission error control for digital cordless telephone system," *IEICE Transactions on Communications*, vol. E77-B, no. 7, pp. 948–955, July 1994.

[19] A. L. Floc'h, R. Salami, B. Mouy, and J.-P. Adoul, "Evaluation of linear and nonlinear spectral subtraction methods for enhancing noisy speech," *Proceedings of INTERSPEECH*, pp. 131–134, November 1992.

[20] J. A. Haigh and J. S. Mason, "Robust voice activity detection using cepstral features," *Proceedings of IEEE TENCON*, vol. 3, pp. 321–324, October 1993.

[21] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 504–512, July 2001.

[22] R. Martin, "An efficient algorithm to estimate the instantaneous SNR of speech signals," *Proceedings of EUROSPEECH*, pp. 1093–1096, 1993.

[23] R. Martin, "Spectral subtraction based on minimum statistics," *Proceedings of EUSIPCO*, pp. 1182–1185, 1994.

[24] L. Liu, J. He, and G. Palm, "Effects of phase on the perception of intervocalic stop consonants," *Speech Communication*, vol. 22, no. 4, pp. 403–417, September 1997.

[25] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pp. 208–211, 1979.

[26] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, no. 2, pp. 113–120, April 1979.

[27] P. Lockwood and J. Boudy, "Experiments with a nonlinear spectral subtractor (NSS), hidden Markov models and the projection, for robust speech recognition in cars," *Speech Communication*, vol. 11, no. 2-3, pp. 215–228, June 1992.

[28] S. V. Vaseghi and B. P. Milner, "Noise compensation methods for hidden Markov model speech recognition in adverse environments," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 1, pp. 11–21, January 1997.

[29] K. K. Paliwal and L. D. Alsteris, "On the usefulness of STFT phase spectrum in human listening tests," *Speech Communication*, vol. 45, no. 2, pp. 153–170, February 2005.

[30] P. Scalart and J. V. Filho, "Speech enhancement based on a priori signal to noise estimation," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 629–632, 1996.

[31] R. Martin, *Statistical Methods for the Enhancement of Noisy Speech*, ser. Speech Enhancement, J. Benesty, S. Makino, and J. Chen, Eds. Springer Berlin Heidelberg, 2005.

[32] Y. Hu and P. C. Loizou, "Speech enhancement based on wavelet thresholding the multitaper spectrum," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 1, pp. 59–67, January 2004.

[33] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, December 1979.

[34] R. J. McAulay and M. L. Malpass, "Speech enhancement using a soft-decision noise suppression filter," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 2, pp. 137–145, April 1980.

[35] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, December 1984.

[36] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, no. 2, pp. 443–445, April 1985.

[37] P. J. Wolfe and S. J. Godsill, "Simple alternatives to the Ephraim and Malah suppression rule for speech enhancement," *Proceedings of 11th IEEE Signal Processing Workshop on Statistical Signal Processing*, pp. 496–499, August 2011.

[38] T. Lotter and P. Vary, "Speech enhancement by MAP spectral amplitude estimation using a super-Gaussian speech model," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 1, pp. 1110–1126, January 2005.

[39] Y. Hu and P. C. Loizou, "Subjective comparison of speech enhancement algorithms," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 153–156, May 2006.

[40] Y. Ephraim and H. Van-Trees, "A signal subspace approach for speech enhancement," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 3, no. 4, pp. 251–266, July 1995.

[41] M. Dendrinos, S. Bakamides, and G. Carayannis, "Speech enhancement from noise; a regenerative approach," *Speech Communication*, vol. 10, no. 1, pp. 45–57, February 1991.

[42] Y. Hu and P. C. Loizou, "A generalized subspace approach for enhancing speech corrupted by colored noise," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 40, pp. 334–341, July 2003.

[43] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition (L)," *Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, November 2006.

[44] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 229–238, January 2008.

[45] J. Ma, Y. Hu, and P. C. Loizou, "Objective measures for predictiong speech intelligibility in noisy conditions based on new band-importance functions," *The Journal of the Acoustical Society of America*, vol. 125, no. 5, pp. 3387–3405, May 2009.

[46] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14, pp. 281–297, 1967.

[47] D. Mansour and B. H. Juang, "A family of distortion measures based upon projection operation for robust speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 11, pp. 1659–1671, November 1989.

[48] H. Kawahara, "Speech representation and transformation using adaptive interpolation of weighted spectrum: Voceder revisited," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 1303–1306, April 1997.

[49] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, A. V. Oppenheim, Ed. Englewood Cliffs, New Jersey 07632: Prentice-Hall, Inc., 1978.

[50] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden Markov models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, May 2013.

[51] H. Dudley, "Remaking speech," *Journal of the Acoustical Society of America*, vol. 11, no. 2, pp. 165–177, October 1939.

[52] B. S. Atal and S. L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *The Journal of the Acoustical Society of America*, vol. 50, no. 2B, pp. 637–655, 1971.

[53] C. Kwan-Un and D. T. Magill, "The residual-excited linear prediction vocoder with transmission rate below 9.6 kbits/s," *IEEE Transactions on Communications*, vol. 23, no. 12, pp. 1466–1474, December 1975.

[54] M. R. Schroeder and B. S. Atal, "Code-excited linear prediction (CELP): high-quality speech at very low bit rates," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 10, pp. 937–940, April 1985.

[55] A. V. McCree and T. P. Barnwell, III, "A mixed excitation LPC vocoder model for low bit rate speech coding," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 4, pp. 242–250, July 1995.

[56] P. Hedelin, "A tone-oriented voice-excited vocoder," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 6, pp. 205–208, April 1981.

[57] R. J. McAulay and J. Thomas F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 4, pp. 744–754, August 1986.

[58] D. Erro, I. Sainz, E. Navas, and I. Hernaez, "Harmonics plus noise model based vocoder for statistical parametric speech synthesis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 2, pp. 184–194, April 2014.

[59] G. S. Kang and S. S. Everett, "Improvement of the excitation source in the narrowband linear prediction vocoder," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, no. 2, pp. 377–386, April 1985.

[60] H. Kawahara, J. Estill, and O. Fujimura, "Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system STRAIGHT," *Proceedings of MAVEBA: Models and Analysis of Vocal Emissions for Biomedical Applications International Workshop*, pp. 59–64, September 2001.

[61] J. Makhoul, R. Viswanathan, R. Schwartz, and A. W. F. Huggins, "A mixed-source model for speech compression and synthesis," *The Journal of the Acoustical Society of America*, vol. 64, pp. 1577–1581, 1978.

[62] S. Y. Kwon and A. J. Goldberg, "An enhanced LPC vocoder with no voiced/unvoiced switch," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 4, pp. 851–858, August 1984.

[63] M. R. Portnoff, "Time-frequency representation of digital signals and systems based on short-time fourier analysis," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 1, pp. 55–69, February 1980.

[64] R. Veldhuis and H. He, "Time-scale and pitch modifications of speech signals and resynthesis from the discrete short-time fourier transform," *Speech Communication*, vol. 18, no. 3, pp. 257–279, May 1996.

[65] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for Mel-cepstral analysis of speech," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 137–140, March 1992.

[66] Q. Hu, K. Richmond, J. Yamagishi, and J. Latorre, "An experimental comparison of multiple vocoder types," *Proceedings of ISCA Workshop on Speech Synthesis*, pp. 135–140, August 2013.

[67] J. Blauert and P. Laws, "Group delay distortions in electroacoustical systems," *Journal of the Acoustical Society of America*, vol. 63, no. 5, pp. 1478–1483, May 1978.

[68] H. Kawahara, "STRAIGHT, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds," *Acoustical Science and Technology*, vol. 27, no. 6, pp. 349–353, 2006.

[69] H. Kawahara, M. Morise, T. Takahashi, R. Nishimura, T. Irino, and H. Banno, "TANDEM-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and aperiodicity estimation," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3933–3936, April 2008.

[70] H. Kawahara, T. Takahashi, M. Morise, and H. Banno, "Development of exploratory research tools based on TANDEM-STRAIGHT," *Proceedings of APSIPA ASC: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pp. 111–120, 2009.

[71] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigne, "Restructing speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, no. 3-4, pp. 187–207, April 1999.

[72] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Pearson Education, Limited, 2010.

[73] J. Laroche, Y. Stylianou, and E. Moulines, "HNS: Speech modification based on a harmonic + noise model," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 550–553, April 1993.

[74] G. Degottex and Y. Stylianou, "A full-band adaptive harmonic representation of speech," *Proceedings of INTERSPEECH*, pp. 382–385, 2012.

[75] Q. Hu, Y. Stylianou, K. Richmond, R. Maia, J. Yamagishi, and J. Latorre, "A fixed dimension and perceptually based dynamic sinusoidal model of speech," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6270–6274, May 2014.

[76] A. Pawi, S. Vaseghi, and B. Milner, "Pitch extraction using modified higher order moments," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5078–5081, March 2010.

[77] D. Talkin, "A robulst algorithm for pitch tracking (RAPT)," *Speech coding and synthesis*, pp. 495–518, 1995.

[78] B. G. Secrest and G. R. Doddington, "An integrated pitch tracking algorithm for speech systems," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 8, pp. 1352–1355, April 1983.

[79] H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, April 2002.

[80] A. M. Noll, "Cepstrum pitch determination," *Journal of the Acoustical Society of America*, vol. 41, pp. 293–309, February 1967.

[81] S. Ahmadi and A. S. Spanias, "Cepstrum-based pitch detection using a new statistical V/UV classification algorithm," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 333–338, May 1999.

[82] S. Gonzalez and M. Brookes, "PEFAC - A pitch estimation algorithm robust to high levels of noise," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 2, pp. 518–530, February 2014.

[83] G. L. Turin, "An introduction to digital matched filters," *Proceedings of the IEEE*, vol. 64, no. 7, pp. 1092–1112, July 1976.

[84] D. Byrne, H. Dillon, K. Tran, S. Arlinger, K. Wilbraham, R. Cox, B. Hagerman, R. Hetu, J. Kei, C. Lui, J. Kiessling, M. N. Kotby, N. H. A. Nasser, W. A. H. E. Kholy, Y. Nakanishi, H. Oyer, R. Powell, D. Stephens, R. Meredith, T. Sirimanna, G. Tavartkiladze, G. I. Frolenkov, S. Westerman, and C. Ludvigsen, "An international comparison of long-term average speech spectra," *Journal of the Acoustical Society of America*, vol. 96, no. 4, pp. 2108–2120, October 1994.

[85] H. Ney, "Dynamic programming algorithm for optimal estimation of speech parameter contours," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, no. 3, p. 208, March 1983.

[86] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, February 1989.

[87] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition," *The Bell System Technical Journal*, vol. 62, no. 4, pp. 1035–1074, April 1983.

[88] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, August 1980.

[89] X. Zhou, D. Garcia-Romero, R. Duraiswami, C. Espy-Wilson, and S. Shamma, "Linear versus Mel frequency cepstral coefficients for speaker recognition," *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 559–564, December 2011.

[90] *ETSI ES 202 212 V1.1.2 'Speech processing, transmission and quality aspects (STQ)'*, ETSI Standard, November 2005.

[91] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937.

[92] C. Yang, F. K. Song, and T. Lee, "Static and dynamic spectral features: Their noise robustness and optimal weights for ASR," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1087–1097, March 2007.

[93] R. Sundaram, A. Ganapathiraju, J. Hamaker, and J. Picone, "ISIP 2000 conversational speech evaluation system," *Speech Transcription Workshop, College Park, Maryland, USA*, 2000.

[94] K. Shinoda and W. Takao, "MDL-based context-dependent subword modeling for speech recognition," *The Journal of the Acoustical Society of Japan*, vol. 21, no. 2, pp. 79–86, 2000.

[95] J. Rissanen, "Universal coding, information, prediction and estimation," *IEEE Transactions on Information Theory*, vol. 30, no. 4, pp. 629–636, July 1984.

[96] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Duration modeling for HMM-based speech synthesis," *Proceedings of International Conference on Spoken Language Processing*, vol. 98, pp. 29–31, 1998.

[97] H. Zen, T. Masuko, K. Tokuda, T. Yoshimura, T. Kobayashi, and T. Kitamura, "State duration modeling for HMM-based speech synthesis," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 3, pp. 692–693, March 2007.

[98] C. De Boor, *A practical guide to splines*, 1st ed., ser. Applied Mathematical Sciences, J. E. Marsden and L. Sirovich, Eds. Springer, 1978, vol. 27.

[99] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 749–752, May 2001.

[100] V. L. Beattie and S. J. Young, "Noisy speech recognition using hidden Markov model state based filtering," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 917–920, 1991.

[101] V. L. Beattie and S. J. Young, "Hidden Markov model state-based cepstral noise compensation," *Proceedings of IEEE International Conference on Spoken Language Processing*, pp. 519–522, 1992.

[102] A. D. Berstein and L. D. Shallom, "An hypothesized Wienner filtering approach to noisy speech recognition," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 913–916, 1991.

[103] A. P. Varga and R. K. Moore, "Hidden Markov model decomposition of speech and noise," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 845–848, April 1990.

[104] M. J. F. Gales and S. J. Young, "Cepstral parameter compensation for HMM recognition in noise," *Speech Communication*, vol. 12, no. 3, pp. 231–239, July 1993.

[105] M. J. F. Gales and S. J. Young, "Robust speech recognition in additive and convolutional noise using parallel model combination," *Computer Speech and Language*, vol. 9, no. 4, pp. 289–307, October 1995.

[106] M. J. F. Gales and S. J. Young, "Robust continuous speech recognition using parallel model combination," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 5, pp. 352–359, September 1996.

[107] F. Faubel, J. McDonough, and D. Klakow, "A phase-averaged model for the relationship between noisy speech, clean speech and noise in the log-mel domain," *Proceedings of INTERSPEECH*, pp. 553–556, 2008.

[108] L. Deng, J. Droppo, and A. Acero, "Enhancement of log Mel power spectra of speech using a phase-sensitive model of the acoustic environment and sequential estimation of the corrupting noise," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 2, pp. 133–143, March 2004.

[109] L. Lu, A. Ghoshal, and S. Renals, "Joint uncertainty decoding with unscented transform for noise robust subspace Gaussian mixture models," *Proceedings of SAPASCALE wokshop*, 2012.

[110] H. Jiang, "Confidence measures for speech recognition: A survey," *Speech Communication*, vol. 45, no. 4, pp. 455–470, 2005.

[111] S. Cox and R. Rose, "Confidence measures for the SWITCHBOARD database," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 511–514, May 1996.

[112] S. Takamichi, T. Toda, A. W. Black, G. N. S. Sakti, and S. Nakamura, "Postfilters to modify the modulation spectrum for statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 24, no. 4, pp. 755–767, April 2016.

[113] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 5, pp. 816–824, May 2007.

[114] T. Toda, *Modeling of Speech Parameter Sequence Considering Global Variance for HMM-Based Speech Synthesis, Hidden Markov Models, Theory and Applications*, P. Dymarski, Ed. In Tech, April 2011.

[115] S. King, "An introduction to statistical parametric speech synthesis," *Sadhana*, vol. 36, no. 5, pp. 837–852, October 2011.

[116] T. Gerkmann and R. C. Hendriks, "Unbiased MMSE-based noise power estimation with low complexity and low tracking delay," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1383–1393, May 2012.

[117] T. J. Ypma, "Historical development of the Newton-Raphson method," *SIMA Review*, vol. 37, no. 4, pp. 531–551, 1995.

[118] M. N. Ghosh and D. Sharma, "Power of Tukey's test for non-additivity," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 25, no. 1, pp. 213–219, 1963.

[119] H. Zen, T. Toda, and K. Tokuda, "The Nitech-NAIST HMM-based speech synthesis system for the Bizzard Challenge 2006," *IEICE Transactions on Information and Systems*, vol. 91, no. 6, pp. 1764–1773, 2008.

[120] G. E. dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, January 2012.

[121] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7962–7966, May 2013.