Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project 2017: Final Report

Project Title:	**Quality-preserving Speech Intelligibility Enhancement using a Kalman Filter**
Student:	**Jia Ying Goh**
CID:	**00749529**
Course:	**4T**
Project Supervisor:	**Brookes, D.M.**
Second Marker:	**Evers, C.**

Abstract

Speech enhancement algorithms aim to reduce the background noise of a noise-corrupted speech input without distorting the original clean speech. In real-world situations, this can be very challenging. Although many algorithms have been developed to improve the Signal-to-Noise Ratio (SNR) of the noisy input, they also introduce speech distortion and artifacts such as musical noise, damaging speech quality and intelligibility. Recently, there has been growing psychoacoustic and physiological evidence to support the use of the modulation domain for speech enhancement, where the modulation domain is defined as the temporal variations of the acoustic spectral components. This report proposes modifications to existing modulation-domain speech processing methods, where an Ideal Binary Mask (IBM) will be applied to training samples of noisy speech to obtain averaged statistical information that can then be applied on new test samples. The goal is to use this data to improve the performance of an existing modulation-domain Kalman Filter (MDKF). The performance of these proposed modifications is assessed by measuring the segmental SNR (segSNR), speech quality (using Perceptual Evaluation of Speech Quality or PESQ) and speech intelligibility (Short-Time Objective Intelligibility or STOI) of the enhanced speech. Results show that the developed algorithms provide varying degrees of improvements in both speech quality and intelligibility over a range of input noise levels.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my project supervisor, Mr Mike Brookes. His patient guidance and understanding has been very helpful throughout the course of the project, and meetings with him have always produced useful ideas and feedback.

Next, I would like to express my appreciation to my family for their understanding and support, allowing me to successfully complete the project.

Last but not least, I would like to thank my friends who have been by my side to support and motivate me throughout the course of the project.

Contents

Abstract **i**

Acknowledgements **ii**

List of Figures **vi**

List of Tables **viii**

1 Introduction **1**

1.1 Motivation 1

1.2 Project Objectives 2

1.3 Project Scope 2

1.4 Report Overview 2

2 Background **4**

2.1 Enhancement Domains 4

2.1.1 Time Domain 4

2.1.2 Time-Frequency Domain 5

2.1.3 Modulation Domain 6

2.2 Noise Estimation 6

2.3 Spectral Subtraction 7

2.4 Ideal Binary Mask 8

2.4.1 Musical Noise 11

2.4.2 Practical Considerations 12

2.5 Linear Prediction Analysis 12

2.6 Kalman Filter 14

2.6.1 Modulation-Domain Kalman Filter 15

2.6.2 Comparison with Time-Domain Kalman Filter 18

2.6.3 Performance of MDKF 19

2.7 Speech Quality 20

2.7.1 Subjective Speech Quality Measures 20

2.7.2 Objective Speech Quality Measures 21

2.8 Speech Intelligibility 23

2.8.1 Short-Time Objective Intelligibility 23

3 Problem Analysis **24**

3.1 Deliverables 24

3.2 Implementation 25

3.2.1 Optimal Modulation Frame Length 25

3.3 Kalman Filter Framework 26

4 Testing Methodology **28**

4.1 Assessing Speech Quality 28

4.2 Assessing Speech Intelligibility 28

4.3 Speech Database: TIMIT 29

5 Modified Kalman Filter Inputs **30**

5.1 Introduction 30

5.2 Incorporating Binary Mask into Observation 30

5.2.1 Gaussian Product 31

5.2.2 Training Mask Statistics 32

5.2.3 Modifying Observation 32

5.3 Modified Kalman Filter Equations 33

5.3.1 Decoupling Kalman Filter Equations 33

5.3.2 Incorporating IBM into decoupled KF equations 34

5.4 Performance Results and Discussion 34

5.5 Conclusion 38

6 Improved LPC Coefficients 39

6.1 Introduction 39

6.2 Weighted LPC estimation 39

6.3 Performance Results and Discussion 40

6.4 Conclusion 44

7 IBM-improved Noise Estimation 45

7.1 Introduction 45

7.2 MMSE Noise Estimation 45

7.2.1 Assumptions in Unbiased MMSE Estimator 46

7.3 Modifying Noise Estimation 47

7.4 Optimal IBM-Modified Estimator Threshold 48

7.4.1 Estimated Global SNR 50

7.5 Performance Results and Discussion 51

7.6 Conclusion 55

8 Conclusion and Future Work 56

8.1 Future Work 57

8.1.1 LPC Estimation 57

8.1.2 Real Listening Tests 57

8.1.3 Probability Distribution 57

8.1.4 Combining Modifications 57

Bibliography 58

List of Figures

2.1 Top to bottom: clean speech, noisy speech, IBM and IBM-processed speech 10

2.2 Performance (percentage of words identified accurately) as a function of LC (dB) for two input SNR levels, masked in multitalker babble (replicated from [33]) 11

2.3 Top row: clean speech (left), speech corrupted with white Gaussian noise (right); bottom row: TDKF-enhanced speech (left), MDKF-enhanced speech (right) 19

2.4 Structure of PESQ model (taken from [61]) 22

2.5 Structure of STOI model (taken from [62]) 23

3.1 Plot of normalised excitation variance vs. modulation frame lengths 26

3.2 Average PESQ scores vs. modulation frame lengths 26

3.3 Average STOI scores vs. modulation frame lengths 27

3.4 Block diagram of baseline MDKF model 27

5.1 Block diagram of IBM-modified MDKF (BMMDKF) 35

5.2 Average segSNR values of BMMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels 36

5.3 Average PESQ values of BMMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels 36

5.4 Average STOI values of BMMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels 37

6.1 Block diagram of MDKF using IBM-enhanced LPCs (LMDKF) 41

6.2 Average segSNR values of LMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels 42

6.3 Average PESQ values of LMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels . . . 42

6.4 Average STOI values of LMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels . . . 43

7.1 Average segSNR ratios comparing IBM-modified noise estimation with original MMSE noise estimation vs. speech corrupted by white noise at varying SNR levels . . . 49

7.2 Average PESQ ratios comparing IBM-modified noise estimation with original MMSE noise estimation vs. speech corrupted by white noise at varying SNR levels . . . 49

7.3 Average STOI ratios comparing IBM-modified noise estimation with original MMSE noise estimation vs. speech corrupted by white noise at varying SNR levels . . . 50

7.4 Estimated global SNR of different noise estimation methods vs. actual global SNR of input speech . . . 51

7.5 Block diagram of MDKF using IBM-modified noise estimation (NMDKF) . . . 52

7.6 Average segSNR values of NMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels . . . 53

7.7 Average PESQ values of NMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels . . . 53

7.8 Average STOI values of NMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels . . . 54

List of Tables

2.1 Categories of MOS: Absolute Category Rating 20

5.1 List of parameters used to evaluate BMMDKF and other algorithms 35

6.1 List of parameters used to evaluate LMDKF and other algorithms 41

7.1 List of parameters used to evaluate NMDKF and other algorithms 52

Nomenclature

AMS	Analysis-Modification-Synthesis
fwSNRseg	Frequency-Weighted Signal-to-Noise Ratio
IBM	Ideal Binary Mask
KF	Kalman Filter
LMS	Least Mean Squares
LPC	Linear Prediction Coefficients
MDKF	Modulation-Domain Kalman Filter
ML	Maximum Likelihood
MMSE	Minimum Mean Squared Error
MOS	Mean Opinion Score
MS	Minimum Statistics
NLMS	Normalised Least Mean Squares
PDF	Probability Density Function
PESQ	Perceptual Evaluation of Speech Quality
PSD	Power Spectral Density
RLS	Recursive Least Squares
segSNR	Segmental Signal-to-Noise Ratio
SNR	Signal-to-Noise Ratio
SPP	Speech Presence Probability
STFT	Short-Time Fourier Transform
STOI	Short-Time Objective Intelligibility

TBM	Target Binary Mask
TDKF	Time-Domain Kalman Filter
VAD	Voice Activity Detector

Chapter 1

Introduction

1.1 Motivation

In today's highly interconnected world, communication between people, as well as with the world around them, is a major and critical aspect of their lives. Among the methods of communication (including but not limited to speech, text, images and bodily cues), speech generally stands out as the most efficient. Other methods such as visual indicators are sometimes useful to communicate ideas and thoughts, but a complex message is often best brought across via speech.

Applications utilising speech are thus widespread and numerous, and are generally designed to make use of clean speech. In a real-world environment, however, when speech is recorded, the recording inherently picks up not just the speech signal of interest, but also undesired background noise and channel noise. This damages the quality and intelligibility of the recorded speech, which poses a major problem for these applications requiring undamaged speech. Speech enhancement is hence often needed, with the goal of restoring the desired speech signal from the noisy mix, ideally by eliminating this noise while retaining the quality and intelligibility of the original speech signal.

There are various types of noise, including but not limited to additive noise, convolution noise and transcoding noise [1]. Additive acoustic noise that is uncorrelated with the speech signal generally degrades the intelligibility and quality of the perceived speech, and in cases of large noise may dominate and mask out the original speech. Convolution noise, on the other hand, manifests as reverberation, which is introduced by acoustic reflection, degrading intelligibility. Unlike additive noise, reverberation is highly correlated with the speech signal. Finally, transcoding noise can occur due to amplitude clipping in a microphone and appears as distortion. This project is concerned with the removal of additive acoustic noise.

Speech enhancement methods can be broadly classified into two types. Single-channel methods consider a single signal source. On the other hand, multi-channel methods consider multiple speech signals obtained from multiple microphones, where additional noise reduction can be achieved using information unavailable to a method relying on a single source, such as phase alignment from multiple microphones, leading to better overall noise reduction. However, this introduces additional costs and complexity, and in many applications such as hearing aids and mobile phones, single-channel methods are necessary due to constraints such as size. This project focuses on single-channel speech enhancement methods.

However, speech enhancement is complex. Traditional speech enhancement techniques such as spectral subtraction have very successfully improved speech quality by attenuating noise, but they tend to introduce speech spectral distortion [2], thus damaging its intelligibility. This project therefore aims to modify existing techniques to improve both the quality and intelligibility of speech.

1.2 Project Objectives

In this project, the objective is to improve both speech quality and intelligibility by modifying an existing speech enhancement algorithm. Standard tests for quality and intelligibility, include the Perceptual Evaluation of Speech Quality (PESQ, [3]) and Short-Time Objective Intelligibility (STOI, [4]), will be used to quantify the enhanced speech.

Specifically, this project aims to modify an existing speech enhancement algorithm based on a Kalman filter, by further incorporating additional information obtained from a so-called “ideal binary mask”. The proposed method is to scale the predicted value in the Kalman filter and modify its variance by an amount pre-determined from training data, with the goal that PESQ and STOI increase.

1.3 Project Scope

This project assumes the binary mask is already provided; how it is generated is therefore out of scope of this project. This project focuses on incorporating a given estimated binary mask into an existing Kalman filter speech enhancement implementation.

This project makes use of MATLAB and signal processing techniques. In particular, the project utilises VOICEBOX, a speech processing toolbox for MATLAB [5], which is included in the Imperial College London Software Library.

1.4 Report Overview

This report is categorised into a few main chapters. Chapter 1 introduces and provides context to the problem, while providing a high-level overview of the project objectives and scope. Chapter 2 provides detailed description of the required background information, particularly providing in-depth information regarding the relevant algorithms and performance evaluation methods.

Chapter 3 analyses the project, breaking down the requirements into specific deliverables. It also describes the implementation of the baseline algorithm and how the improvements will be made. Chapter 4 briefly covers the testing methodology and how the algorithms will be evaluated.

Chapter 5 covers the first proposed modification, whereby an ideal binary mask is applied to a set of training speech samples. This gives useful statistical information that is then used to improve on a baseline Kalman filter implementation. Its performance relative to existing methods is evaluated and discussed.

Chapter 6 explores a different approach. Here, the linear prediction coefficients used in a Kalman filter algorithm are modified by using a weighted sum to calculate the autocorrelation function, with weights determined from an ideal binary mask applied to the input signal. As before, this enhancer is evaluated and discussed relative to existing methods.

A third modification is proposed in Chapter 7. Noise estimation is an important part of speech enhancement, and a method is proposed to incorporate binary mask information to improve the accuracy of noise estimation in a Kalman filter-based speech enhancer. Finally, Chapter 8 concludes the project and briefly explores possible areas for future work.

Chapter 2

Background

The goal of a speech enhancement algorithm is to reduce the background noise of a noise-corrupted speech input without distorting the underlying clean speech [6]. Traditionally, speech enhancement algorithms for noise reduction can be grouped into three main categories: noise reduction via filtering techniques, noise reduction via spectral restoration, and speech-model-based noise reduction methods [7]. Overall, speech enhancement techniques aim to improve the speech using audio signal processing techniques.

Some widely-used speech enhancement methods are described in this chapter. This chapter also provides the background to some relevant aspects of speech enhancement, including possible operating domains, noise estimation, and performance evaluation measures.

2.1 Enhancement Domains

Speech enhancement can be performed in one of several domains. The following sections briefly describe these domains.

2.1.1 Time Domain

In the time domain, speech is usually enhanced using fixed or adaptive filtering techniques [8]. Fixed filters require prior knowledge of both the clean signal and noise, while this is not required for adaptive filters, which are able to adjust their parameters according to an optimisation algorithm, with little to no knowledge of the signal or noise characteristics, thus being more practical.

There are different approaches to adaptive filtering, one of which is the Least-Mean-Squares (LMS) filter. LMS algorithms aim to mimic a desired filter by finding a set of filter coefficients to minimise the mean squared error, where the error is the difference between the desired and actual signal [9]. The basic idea is to iteratively update the filter coefficients to approach the optimum coefficients, using a certain step size at each iteration. The LMS is a stochastic gradient descent approach, meaning that it is adapted based on the current error. It is, however, sensitive to input scaling, making it difficult to find an optimum step size to guarantee convergence. This limitation motivated the development of a variant, the Normalised Least Mean Squares (NLMS) algorithm, which is a variant of LMS that solves this problem by normalising with the power of the input [10].

Another popular approach is the Recursive Least Squares (RLS) algorithm, which recursively finds the filter coefficients to minimize a weighted least squares cost function relating to the input signal. This is unlike LMS and NLMS, which aim to reduce the mean squared error. Compared to LMS and NLMS, the RLS exhibits very fast convergence, but at the cost of higher computational complexity.

Generally, adaptive filters are used when there is some quantity to be minimised. For example, an adaptive filter can be implemented iteratively with a time delay to estimate and remove mains noise, which is periodic and thus can be estimated from previous samples. This is not as applicable in typical speech enhancement, which is concerned with reducing random noise.

2.1.2 Time-Frequency Domain

Speech enhancement can be performed in the time-frequency (T-F) domain, which analyses signals in both time and frequency domains simultaneously, using various T-F representations [11]. Assuming speech is quasi-stationary over sufficiently short periods [12], the noisy input speech signal is divided into overlapping short frames, typically using a Hamming window, where the frame length is a compromise between temporal and frequency resolution [13]. These frames will be called acoustic frames, and are separate from the modulation frames referred to in the modulation domain in Section 2.1.3. Performing the Fourier transform on these frames produces a T-F matrix, on which processing can then be done. This entire process is called the short-time Fourier Transform (STFT).

Generally, T-F enhancement methods apply a gain function to suppress T-F regions which are noise-dominated while preserving speech-dominated regions, typically on the magnitude spectrum only. Computing the gain function depends on the noisy power spectrum, which needs to be estimated separately. After processing, inverse STFT followed by overlap-add reconstruction is performed to produce the enhanced time-domain speech signal. This approach works because speech is relatively sparse, due to limitations of the human ability in terms of speaking and listening i.e. with reasonable levels of noise, the speech can be divided into speech-dominated and noise-dominated regions.

Although this approach can improve the calculated signal-to-noise (SNR) of noisy speech, it can lead to undesired “musical noise” artifacts. These appear as isolated spectral components of noise and manifest as brief tones in the enhanced speech, which are generally deemed unnatural and disturbing [14]. This is because the amplitude of the short-time spectrum exhibits large fluctuations in noisy regions. After processing, the enhanced spectrogram consists of randomly located spectral peaks corresponding to the maxima of the original spectrogram, where the regions between these peaks have been suppressed as they are close to or below the averaged estimated noise spectrum. The result is residual noise comprising of sinusoids of random frequencies between each time frame.

This is used in the setup of the algorithm described in Section 2.6.1.

2.1.3 Modulation Domain

Modulation-domain processing starts off similarly to T-F processing, in that the noisy input signal undergoes STFT analysis to produce time-varying frequency components.

For speech enhancement, the amplitudes envelope of each frequency band is regarded as one modulation signal; the spectral amplitudes of each frequency band are windowed into overlapping modulation frames, with a separate modulation frame length and frame overlap compared to the acoustic frame length and overlap of STFT, where each acoustic frame provides one modulation-domain sample for each frequency bin. If each modulation frame contains M samples (i.e. M acoustic frames form one modulation frame) and each acoustic frame contains N time-domain samples, each modulation frame is constructed from MN time-domain samples. The modulation-domain signal has a frequency determined by the acoustic frame increment: since each acoustic frame provides one modulation sample, successive modulation samples are spaced apart by the acoustic frame shift. If the time-domain signal has a sampling frequency of f_s Hz and the acoustic frame shift is L Hz, the modulation-domain sampling frequency is (f_s/L) Hz.

A processing algorithm then estimates the modulation frames of clean speech, which are then overlap-added to reconstruct the modified modulation signals. Combining this with the phase spectrum of the noisy input signal and performing the inverse STFT then produces the enhanced time-domain speech signal.

Even though the acoustic envelope directly contains the speech information, the temporal dynamics of the envelope better represent the information contained in speech [15]. These dynamics, which are at significantly lower frequencies than the speech signal itself, are provided in the modulation spectrum, suggesting that working in the modulation domain for speech processing can produce better results. More detailed description is provided in Section 2.6.

2.2 Noise Estimation

Noise estimation is an important part of speech processing. In many algorithms including those described in this report, performance is heavily affected by the accuracy of the noise estimation. The algorithms proposed in this project are rely on the estimation of the spectral noise power.

When estimating noise power, because the noise power may change rapidly over time, the spectral estimate has to be updated as often as possible. In speech enhancement, an overestimate or underestimate of the noise power will lead to an over-suppression or under-suppression of the noisy signal, which can lead to excessive residual noise or reduced intelligibility in the enhanced signal.

One way to estimate spectral noise power is to exploit the time periods where speech is absent, which requires detection of speech presence using a voice activity detector (VAD) [16]. This method encounters problems when the noise is non-stationary, however, as a sudden increase in noise power may instead be interpreted as the onset of speech. Additionally, if the noise power varies during speech presence, this change can only be detected with some time delay.

To improve noise estimation, methods have been proposed based on minimum statistics (MS) [17]. The method in [17] does not use a VAD. Instead, it tracks spectral minima in each frequency band without distinguishing between between speech presence and absence. The power spectrum of the noisy signal is tracked frame-by-frame and observed over a short time period. A general assumption is that within the observed time frame, speech is absent for a non-zero portion of the total time period. The spectral noise power is then obtained from the minimum of the estimated power spectrum of the noisy signal. Similar to VADs, if the noise power rises within the observed time period, it can be tracked with some time delay. Due to the delay, the local noise power estimates tend to be underestimated. This results in residual noise and musical noise artifacts when the noise estimation is used in a speech enhancer.

A more recent method to estimate the noise power spectral density (PSD) is to use a minimum mean-squared-error (MMSE) optimal estimation method [18], which can be interpreted as a VAD-based noise power estimator [19]. Generally, MMSE-based estimators are more robust to non-stationary noise and are less computationally intensive as compared to MS-based methods [18]. The estimator in [19] replaces the hard VAD of the estimator in [18] with a soft speech presence probability (SPP) with a fixed *a priori* SNR as a parameter of the likelihood of speech presence, using a value typical in speech presence. This soft estimation overcomes the issue of random fluctuations in the estimated SNR. This modification automatically makes the estimator unbiased, and it retains similar performance while achieving lower computational complexity compared to [18].

2.3 Spectral Subtraction

Spectral subtraction is a widely-used filtering technique which operates in the time-frequency domain. In this method, stationary or slowly-varying noise is attenuated from noisy speech by subtracting the magnitude noise spectrum, estimated during periods where speech is absent [20]. It is also possible to estimate the noise using a secondary sensor [8]. The estimated noise spectrum is then subtracted from the noisy spectrum to produce an approximated spectrum of the clean speech. The spectral error can then be computed and reduced separately. The algorithm can be further refined by incorporating residual noise reduction and non-speech signal attenuation [20].

Spectral subtraction works on the back of a few assumptions: firstly, that the background noise is additive to the clean signal [20]. This assumption means that the complex spectrum of the input noisy signal can be expressed as the sum of the speech spectrum and the noise spectrum. Next, it is assumed that the noise is a stationary or a slowly varying process (locally stationary). This allows the algorithm enough time to accurately formulate an updated estimate for the new noise magnitude spectrum before speech activity starts again. Lastly, the underlying assumption is that noise can be significantly reduced by removing its effect in the magnitude spectrum only i.e. phase spectrum is untouched, and the estimate of the clean speech magnitude spectrum is combined with the phase spectrum of the noisy input signal [21].

As mentioned in Section 2.1.2, the local stationarity assumption means the processing should be done on small-enough chunks of the input. Therefore, the input must first be split into overlapping frames using overlap-add processing. In the final step after processing, these frames are reassembled to form the continuous output signal.

To avoid signal distortion introduced by data segmentation [22], each frame is first multiplied by a windowing function before performing the Fourier Transform (typically using the Fast Fourier

Transform or FFT). The output signal is then formed by the sum of these overlapping windowed frames. After processing, when the signal is being reassembled, the window is applied again.

For the signal to remain undistorted, multiplying by these windows should not change its magnitude. To achieve this, particular overlap factor/window pairs must be used; for example, if a Hamming window is chosen, applying the window twice requires that the overlapped windows approximately sum to unity for an overlap factor of 4 i.e. each windowed frame overlaps each of its neighbours by 50%, ensuring the output signal remains undistorted.

Spectral subtraction is popular largely because it is simple and easy to implement, requiring mainly the forward and inverse Fourier Transforms. However, this comes at a cost to performance. Subtracting the noise spectrum from the noisy input spectrum introduces distortion in the signal, known as musical noise [1], as mentioned in Section 2.1.2. Variations have been developed in attempts to mitigate this. A common variation involves over-subtraction and a noise floor. This method involves an over-subtraction factor, whereby an overestimate of the noise power spectrum is subtracted from that of the input, and using a noise spectral floor, which prevents the processed spectrum from going below a preset minimum value, to control both the amount of residual noise and musical noise [23]. However, it is generally evaluated that these modifications improve speech quality further but do not significantly affect the intelligibility of the input signals [1].

2.4 Ideal Binary Mask

Sound is generated by acoustic sources, and these sources are typically complex, containing multiple frequency components. In a typical environment, multiple acoustic sources are simultaneously active, including undesired background noise, and a listener's ear will pick up only the sum of all these sources. There are various types of corrupting background noise, including but not limited to acoustic noise (e.g. vehicle vibration), speech-shaped noise, industrial noise and multi-talker babble (e.g. noisy cafeteria with other speakers) [24]. For the listener to distinguish between the different sounds in the incoming mix, such as picking out a particular speaker in a busy supermarket, the incoming audio signal has to be partitioned and categorised accurately into individual sounds.

Human beings have auditory systems that are remarkably capable at doing this, and are thus able to understand speech in many of these noisy conditions. The signal separation process, known as auditory scene analysis, is typically performed in two stages, to understand the message spoken by the target speaker. Firstly, the input sound is decomposed into a matrix of time-frequency (T-F) units, where each unit represents the signal occurring at a particular instance in time with a particular frequency component. These T-F units are then analysed, and the auditory system utilises a combination of cues, learned patterns and other prior knowledge about the target to pick out the T-F units of the target signal, and group these individual units into a single recognisable "image" of the desired signal [25]. Essentially, the auditory system employs an analysis-synthesis strategy to organise the input into separate streams corresponding to different audio sources.

To model the human auditory system, computational auditory scene analysis (CASA) was proposed to approach sound separation in two stages: segregation and grouping [26]. The aim of using these CASA techniques was to pick out the target signal from the noisy mix, and the computational method of choice was the ideal T-F binary mask [27].

The ideal binary mask (IBM) is defined in the T-F domain as a matrix of binary numbers, and is constructed by comparing the local signal-to-noise ratio (SNR), defined as the difference between the target signal energy and the noise energy, in each T-F unit against a threshold known as a local criterion (LC). In the IBM, the T-F units with local SNR exceeding the LC (in decibels) are assigned 1, and 0 otherwise. If a 0 dB SNR threshold is used to generate the mask, a T-F unit being assigned 1 indicates that the energy of the target signal is stronger than that of the interference (masker) within that particular T-F unit, which is a particularly intuitive implementation. Let $T(t, f)$ and $M(t, f)$ denote the target and masker signal power measured in dB respectively, at time t and frequency f; the IBM is then defined as

$$IBM(t, f) = \begin{cases} 1 & \text{if } T(t, f) - M(t, f) > LC \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

This mask can then be applied to the T-F representation of the incoming noisy signal; it acts as a selective filter, allowing some parts of the signal to pass through (those T-F units assigned to 1) while eliminating other parts (those assigned to 0). This means that at each T-F unit, the IBM either retains target energy or discards interference energy. The IBM therefore offers an indication of the T-F areas of audible target speech, and offers significant improvements in intelligibility [28].

An example of the IBM at work is shown in Figure 2.1, with a clean sentence obtained from the TIMIT database [29]. From top to bottom, the spectrograms shown are that of: a) clean speech; b) clean speech corrupted with white Gaussian noise at 5 dB SNR; c) IBM constructed using LC threshold of 0 dB, where black pixels denote 1 (target stronger than interference masker) and white pixels denote 0 (target weaker than masker); d) segregated mixture obtained with the 0 dB LC IBM, obtained by multiplying the spectrograms in (b) and (c), one T-F unit at a time.

The 0 dB LC IBM, a particularly simple and intuitive comparison, is theoretically optimal in terms of SNR gain ([30], [31]); Figure 2.1 shows its good performance, whereby the spectrogram of the processed speech is nearly identical to that of clean speech. It was later shown that while it is not optimal due to certain constraints, it performs almost as well as the proposed alternative, and is in fact more practical for real-world implementation [32]. Multiple studies have examined further the effects of the LC, input SNR level and masker type on the performance of the IBM. For example, a technique called ideal T-F segregation (ITFS) has been effective in making use of the IBM to improve the intelligibility of human speech masked by competing voices [28]. It is argued that the ITFS removes informational masking caused by the IBM-eliminated T-F units with large masker energy, where informational masking refers to the inability to accurately distinguish the target signal from the noisy mixture.

To demonstrate the benefits of IBM processing, various studies carried out intelligibility tests, in which listeners listen to a set of IBM-processed sentences and write down the words they hear; results produced are in terms of the percentage of words identified correctly.

A typical IBM intelligibility test result is shown in Figure 2.2, where UN represents the unprocessed noisy speech (replicated from [33]). Here, STFT was used to process the input noisy signal, using multitalker babble as the masker. As shown, the performance peaks out between roughly −20 dB and 5 dB for an input SNR of −5 dB, with a slightly smaller range for an input SNR of −10 dB.

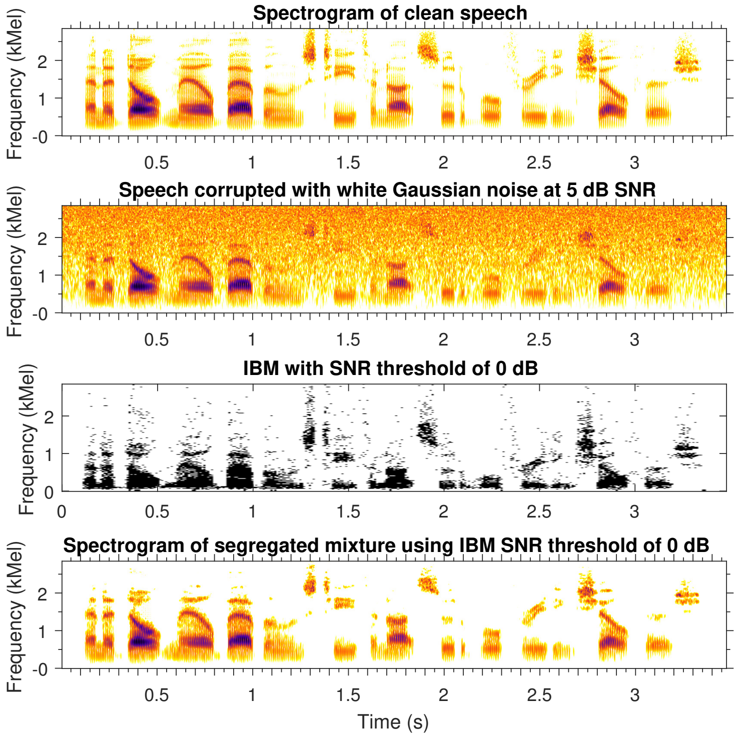

Figure 2.1: Top to bottom: clean speech, noisy speech, IBM and IBM-processed speech

Large intelligibility gains were demonstrated in [33], but the range of LC values for near-perfect intelligibility (performance plateaus of ≈ 100% accuracy) were different to that in [28]. Attributing this to differences in the setup and signals used, it was suggested that the pattern of the IBM was the critical factor for intelligibility, rather than the local SNR of individual T-F units [33].

The significant improvements to intelligibility made IBM a notable candidate for speech enhancement applications such as hearing aids, provided the IBM could be approximated to a high degree of accuracy. However, to apply it, it is important to understand how IBM enhances intelligibility. In [28], it is argued that the IBM suppresses informational masking by directing the listener's attention to the T-F units containing target information i.e. *where* the target signal is, in a T-F auditory space [33]. This led to the conclusion that listeners need not extract specific knowledge from individual T-F units, but rather the overall pattern of the IBM, i.e. pattern of target-dominated and masker-dominated T-F units, was the most important factor for intelligibility, which was also concluded in [33]. However, this interpretation is limited to the range of LCs where the IBM pattern represents the T-F units that are audible to normal human listeners i.e. LCs close to 0 dB [34].

An alternative ideal mask definition was proposed in [35], which also produced large intelligibility improvements. This alternative mask was named the target binary mask (TBM), as the mask was

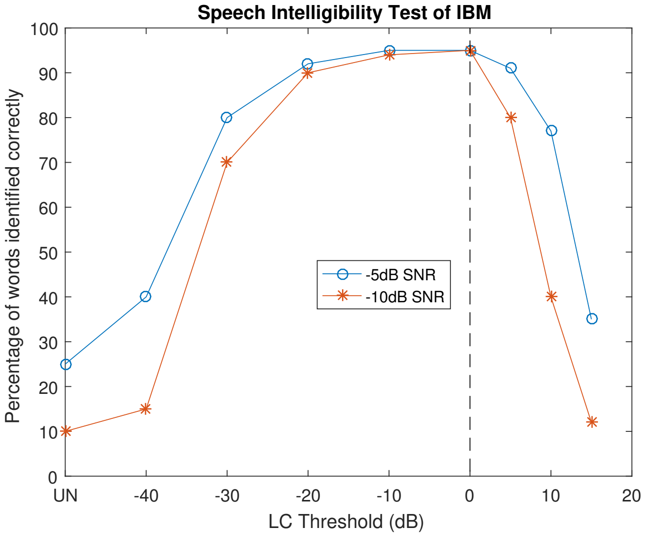

Figure 2.2: Performance (percentage of words identified accurately) as a function of LC (dB) for two input SNR levels, masked in multitalker babble (replicated from [33])

calculated based on the target signal only. TBM depends on the long-term spectrum of the target speaker, and compares with an average spectrum i.e. a time-invariant threshold. The mask pattern naturally resembles the target signal and is unaffected by the masker specifically. Instead, the TBM generated in this manner can be applied to a mixture of the target signal and a different masker. On the other hand, the IBM pattern depends on the masking signal; IBM compares with the actual noise in the T-F units, which is time-dependent.

In certain applications, it may be easier to estimate the TBM than the IBM, and so it was of interest to investigate the intelligibility performance of the TBM: it was shown that the TBM has comparable performance to the IBM [36]. A noise-robust method based on target sound estimation to estimate the TBM was proposed in [37].

2.4.1 Musical Noise

The largest calculated SNR gain is achieved by using a fully-binary mask of 1s and 0s. However, doing so generally degrades the quality of the enhanced speech due to the introduction of musical noise, which refers to random, short tone-like bursts that, in some situations, can be more bothersome than the original noise. In [38], the mask values were instead 1 and 0.1: if the mask indicates that speech is present, the gain is 1, otherwise the gain is set to 0.1 instead of cutting completely. By reducing the severity of the mask switching, the amount of musical noise artifacts introduced by the mask was reduced overall.

2.4.2 Practical Considerations

By definition, the IBM depends on oracle knowledge, as the mask is constructed based on the target and interfering signals before mixing. In a real-world situation, the target signal is of course unavailable, meaning the IBM has to be estimated from noisy data only. In the presence of significant noise, this can be a difficult task, and it is impossible to compute the IBM for all T-F units with complete accuracy. The effect of overall binary mask estimation error was investigated further in [33], and it was demonstrated that the estimation needs to be very accurate overall. As an example, $> 90\%$ accuracy is required to estimate the IBM for the case of -5 dB input masked with multitalker babble to yield significant gains in intelligibility.

While it is of interest to further investigate the effects of estimation uncertainty and error on speech intelligibility improvements, this project focuses on the Kalman filter algorithm, and assumes that an ideal or estimated binary mask has already been computed and is available.

2.5 Linear Prediction Analysis

In linear prediction, future values of a signal are estimated as a linear combination of previous samples. In speech enhancement, the temporal variation of a speech signal is modelled using a linear predictive model. Equation 2.2 shows a p-th order all-pole linear predictor (autoregressive model), where the current state $\hat{s}(n)$ is estimated as a linear combination of p previous states [39].

$$\hat{s}(n) = -\sum_{k=1}^{p} a_k s(n-k) \tag{2.2}$$

where $s(n)$ is a speech sequence that is windowed to ensure the quasi-stationarity of the speech. In speech enhancement, obtaining an accurate model of the speech is useful for noise attenuation. This entails obtaining accurate estimates of the coefficients a_k in Equation 2.2, known collectively as the linear prediction coefficients (LPCs). This method is known as linear prediction analysis, where the goal is to obtain estimates of a_k that minimise the mean squared error [39].

The prediction error (prediction residual) is computed as the difference between the actual sample $s(n)$ and the predicted sample $\hat{s}(n)$:

$$e(n) = s(n) - \hat{s}(n) = s(n) + \sum_{k=1}^{p} a_k s(n-k) \tag{2.3}$$

A popular method to minimise the mean squared error is the least squares autocorrelation method, achieved by minimising the total prediction error E over all samples:

$$E = \sum_{n} e^2(n) = \sum_{n} [s(n) + \sum_{k=1}^{p} a_k s(n-k)]^2 \tag{2.4}$$

The coefficients a_k that minimise Equation 2.4 can be found by taking the derivative of E with respect to a_k and setting it to 0, thus solving:

$$\frac{\partial E}{\partial a_i} = 0, \quad i = 1, 2, ..., p \tag{2.5}$$

From Equations 2.4 and 2.5, we have

$$\sum_n s(n-i)s(n) = \sum_{k=1}^{p} a_k \sum_n s(n-i)s(n-k), \quad i = 1, 2, ..., p \tag{2.6}$$

which produces a set of p linear equations and p unknowns which can be solved for the LPC coefficients $\{a_k, k = 1, 2, ..., p\}$ that minimise E in Equation 2.4.

In the autocorrelation method, the error is assumed to be minimised over the infinite duration $-\infty < n < \infty$ [39]. In practice, there is typically a finite interval of interest: $0 < n < N - 1$. Equation 2.6 can then be written as:

$$R(i) = \sum_{k=1}^{p} a_k R(i-k), \quad i = 1, 2, ..., p \tag{2.7}$$

where

$$R(i) = \sum_{n=i}^{N-1} s(n-i)s(n) \tag{2.8}$$

is the autocorrelation function of the signal $s(n)$. The coefficients $R(i-k)$ form the autocorrelation matrix $\mathbf{R}$, which is a symmetric Toeplitz matrix [39] since the autocorrelation function is even i.e. $R(i) = R(-i)$. In matrix notation, Equation 2.7 can be written as:

$$\mathbf{RA} = -\mathbf{r} \tag{2.9}$$

where $\mathbf{A}$ contains the desired LPC coefficients $\{a_k, k = 1, 2, ..., p\}$. These coefficients can then obtained by solving the matrix inverse equation:

$$\mathbf{A} = -\mathbf{R}^{-1}\mathbf{r} \tag{2.10}$$

Finally, the prediction residual (Equation 2.3) can be obtained by inverse filtering the noisy speech signal using the computed LPCs [40].

LPC analysis is based on the source-filter model of speech production, which models speech as a combination of a source of sound (vocal cords) and a linear filter (vocal tract) [41]. This assumes that speech is produced by a buzzer at the end of a tube, which is a close approximation of real speech production [42]. The glottis, which produces the buzz, is the opening between the vocal folds; as the vocal folds vibrate, a “buzzing” sound is produced, which is what we term “pronunciation” [43]. This buzz is characterised by its intensity (loudness) and frequency (pitch). The vocal tract, comprising the throat and mouth, forms the tube, which is characterised by its resonances, which produce formants (enhanced frequency harmonics) in the speech. The vocal tract transfer function can be modelled by an all-pole filter, which we use to estimate the current state of speech as a linear function of previous states.

In this project, LPC analysis is performed on modulation frames in the modulation domain, using the autocorrelation method described in [39]. The prediction residual is used to calculate the excitation variance in the algorithm described in Section 2.6.1, which is the basis of the algorithms discussed in this report.

2.6 Kalman Filter

The Kalman filter [44] is a recursive optimal data processing algorithm. Under certain assumptions, it is optimal with respect to any practical measure. This is because the Kalman filter (KF) makes use of all data available to it, processing all available information to estimate the current value of the desired variables. In the context of speech enhancement, speech signals are modelled as autoregressive processes using the state space method, where the processed speech is recursively estimated, one sample at a time [45].

The filter has a recursive “predictor-corrector” structure [46]; firstly, a prediction of the desired variable at the next measurement time is made, based on all previously available data, producing a prediction value and its associated uncertainty. When the next measurement is actually taken, the difference between the measurement and the predicted value is used to “correct” the prediction, to produce the new estimate. Note that this recorded measurement comes with its associated uncertainty, arising from imperfections of measuring instruments. The new estimate is thus updated using a linear combination of the prediction and the measurement, with more weight given to estimates with lower uncertainty.

The KF was initially proposed for speech enhancement by Paliwal and Basu in 1987 [47], where excellent noise reduction was achieved when linear prediction coefficients (LPCs) were estimated from clean speech. The KF is of particular interest for speech enhancement, as the speech model is inbuilt into the KF recursion equations, and the enhanced speech contains little to no musical noise, assuming clean LPCs are available [48]; the performance of the KF is highly dependent on the accurate estimation of LPCs. However, for practical use, these parameters have to be estimated from noisy speech since the clean speech is not known *a priori*, causing a significant drop in performance. Better performance has been demonstrated in variations of the original KF algorithm, such as a cascaded estimator/encoder structure which improves LPC estimates [49].

In recent years, the focus has shifted away from the traditional KF methods which utilise the acoustic domain, defined as the short-time Fourier Transform (STFT) of the signal. Instead, there has been growing interest in the modulation domain, defined as the variation over time of the magnitude spectrum at all acoustic frequencies [50]. Studies have increasingly shown the importance of the modulation domain for speech analysis; for example, very low frequency modulations of sound have been shown to be the fundamental carriers of information in speech [50], due to physiological limitations on how rapidly the vocal tract is able to change with time [51]. The slowly-varying modulation domain hence represents how the vocal tract changes over time [52].

The KF is capable of handling non-stationary signals as well as estimating both magnitude and phase spectra [53], which puts it at an advantage over STFT-based, acoustic domain-based methods for speech processing, as phase information has been shown to be more important in the modulation domain than in the acoustic domain [54]. It was also noted in [52] that the low order linear predictor KF was more appropriate for enhancing slower-varying modulating signals than for enhancing time-domain speech, as the time-domain signals contain long-term correlation which the low order linear predictor cannot capture. This is important for the KF, as its optimality works on the basis of incorporating and using all data available to the algorithm. These results suggest the use of the KF in the modulation domain as an improved method of speech enhancement [52].

2.6.1 Modulation-Domain Kalman Filter

The modulation-domain KF (MDKF) is an adaptive minimum mean-squared error (MMSE) estimator that uses the statistics of time-varying changes in the magnitude spectrum of both speech and noise [52]. In the MDKF, an analysis-modification-synthesis (AMS) framework is used to obtain the modulation domain in three steps. In the analysis stage, the input speech signal is processed using STFT; next, the noisy input spectrum undergoes some modification or processing; and lastly, the output processed signal is synthesised by inverse STFT followed by the overlap-add method.

Analysis-modification-synthesis framework in the acoustic domain

Considering an additive noise model, where $y(n)$, $x(n)$ and $v(n)$ represent zero-mean signals of noisy speech, clean speech and noise respectively:

$$y(n) = x(n) + v(n) \tag{2.11}$$

Assuming speech is quasi-stationary means that it can be analysed in frames using the STFT (analysis), thus obtaining the STFT of the noisy signal $y(n)$:

$$Y(n,k) = \sum_{l=-\infty}^{\infty} y(l)w(n-l)e^{-j\frac{2\pi kl}{N}} \tag{2.12}$$

which can be represented using STFT analysis as Equation 2.13:

$$Y(n,k) = X(n,k) + V(n,k) \tag{2.13}$$

where $Y(n,k)$, $X(n,k)$ and $V(n,k)$ denote the STFTs of noisy speech, clean speech and noise respectively and k refers to the discrete acoustic frequency index, N is the acoustic frame duration in number of samples and $w(n)$ is a window analysis function. For speech enhancement, a Hamming window is typically used. Note that this model is noise-additive in the complex STFT domain.

Each one of $Y(n,k)$, $X(n,k)$ and $V(n,k)$ is a complex spectrum, and can be expressed in terms of their acoustic magnitude and acoustic phase spectra. For example, $Y(n,k)$ can be represented as:

$$Y(n,k) = |Y(n,k)|e^{j\angle Y(n,k)} \tag{2.14}$$

where $|Y(n,k)|$ is the acoustic magnitude spectrum and $\angle Y(n,k)$ is the acoustic phase spectrum.

Traditionally, AMS-based methods only modify the noisy acoustic magnitude spectrum $|Y(n,k)|$ to obtain a processed magnitude spectrum $|\hat{X}(n,k)|$; the modified spectrum is thus obtained by combining the enhanced magnitude spectrum with the original noisy phase spectrum $\angle Y(n,k)$:

$$\hat{X}(n,k) = |\hat{X}(n,k)|e^{j\angle Y(n,k)} \tag{2.15}$$

The enhanced speech $\hat{x}(n)$ is then reconstructed by performing the inverse STFT of the enhanced acoustic spectrum $\hat{X}(n,k)$ followed by synthesis windowing and overlap-add [55].

Kalman filter model in the modulation domain

As briefly introduced in Section 2.1.3, in the modulation domain, the acoustic magnitude spectrum of noisy speech is interpreted as a series of modulating signals spanning across time, where each modulating signal $|Y(n,k)|$ represents the variation of one frequency component over time, with $k = 1,2,...,N$ where N is the number of frequency bins. Each modulating signal is individually processed with a separate KF [52].

To visualise this, imagine that a time-domain noisy speech signal sampled at 8 kHz is windowed with a 64 ms frame (window) length and 4 ms frame shift. Taking the STFT, each window is analysed individually: the samples within a 64 ms window are viewed as a frequency-domain signal with (for example) 256 frequency bins. When the next window is taken (original window shifted by 4 ms), the samples are again analysed into a set of 256 frequency bins. Doing this for the entire signal produces 256 time-varying signals (modulating signals), one for each frequency component and processed with its own KF, where the samples in each signal are 4 ms apart. Within each KF, the modulating signal is further windowed, but the signal now has a much lower frequency: in this case, $\frac{1}{0.004} = 250$ Hz. Assuming a modulating window of 64 ms, each window only contains $\frac{64}{4} = 16$ samples, compared to 512 samples for a 64 ms window of a 8 kHz time-domain signal.

Returning to the model, an additive noise model is assumed for each modulating signal, assuming white Gaussian noise (Equation 2.16). Recall that the noisy phase spectrum is left untouched.

$$|Y(n,k)| = |X(n,k)| + |V(n,k)| \tag{2.16}$$

In the KF autoregressive model, a p-order linear predictor is used to model the evolution of speech over time (Equation 2.17), where $a_{j,k}; j = 1,2,...,p$ are the LPCs and $W(n,k)$ is a random white excitation with a variance of $\sigma^2_{W(k)}$.

$$|X(n,k)| = -\sum_{j=1}^{p} a_{j,k}|X(n-j,k)| + W(n,k) \tag{2.17}$$

Including the noise signal, the overall state space representation for noisy speech can be written as:

$$\mathbf{X}(n,k) = \mathbf{A}(k)\mathbf{X}(n-1,k) + \mathbf{d}W(n,k) \tag{2.18}$$

$$|Y(n,k)| = \mathbf{d}^T\mathbf{X}(n,k) + |V(n,k)| \tag{2.19}$$

where $\mathbf{X}(n,k) = [|X(n,k)|, |X(n-1,k)|, ...|X(n-p+1,k)|]^T$ is the clean speech modulation state vector, $\mathbf{d} = [1,0,...,0]^T$ is the measurement vector for both the excitation noise $W(n,k)$ and observation, and $\mathbf{A}(k)$ is the state transition matrix utilising the LPCs:

$$\mathbf{A}(k) = \begin{bmatrix} -a_{1,k} & -a_{2,k} & \dots & -a_{p-1,k} & -a_{p,k} \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} \tag{2.20}$$

The Kalman filter recursively calculates a linear unbiased MMSE estimate $\hat{\mathbf{X}}(n|n,k)$ of the k-th modulation state vector at time n, given the noisy modulating signal up to and including time n (i.e. $|Y(1,k)|, |Y(2,k)|, ...|Y(n,k)|$) using the following equations:

$$\mathbf{P}(n|n-1,k) = \mathbf{A}(k)\mathbf{P}(n-1|n-1,k)\mathbf{A}(k)^T + \sigma^2_{W(k)}\mathbf{d}\mathbf{d}^T \tag{2.21}$$

$$\hat{\mathbf{X}}(n|n-1,k) = \mathbf{A}(k)\hat{\mathbf{X}}(n-1|n-1,k) \tag{2.22}$$

$$\mathbf{K}(n,k) = \mathbf{P}(n|n-1,k)\mathbf{d}[\sigma^2_{V(k)} + \mathbf{d}^T\mathbf{P}(n|n-1,k)\mathbf{d}]^{-1} \tag{2.23}$$

$$\mathbf{P}(n|n,k) = [\mathbf{I} - \mathbf{K}(n,k)\mathbf{d}^T]\mathbf{P}(n|n-1,k) \tag{2.24}$$

$$\hat{\mathbf{X}}(n|n,k) = \hat{\mathbf{X}}(n|n-1,k) + \mathbf{K}(n,k)[|Y(n,k)| - \mathbf{d}^T\hat{\mathbf{X}}(n|n-1,k)] \tag{2.25}$$

where $\sigma^2_{V(k)}$ is the variance of the corrupting noise and $\mathbf{P}(n|n,k)$ is the error covariance matrix. These equations can be categorised into two main steps: prediction and updating. Equations 2.21 and 2.22 predict the error covariance and state based on past samples respectively, while the other equations update the Kalman gain, error covariance and state based on the predicted values.

In particular, Equation 2.25 is the main updating step, whereby a linear combination of the estimate based on previous samples $|\hat{X}(n|n-1,k)|$ and the current measurement $|Y(n,k)|$ is used to compute the current estimate $|\hat{X}(n|n,k)|$. To view this more clearly, we can rewrite Equation 2.25 as:

$$\hat{\mathbf{X}}(n|n,k) = [\mathbf{I} - \mathbf{K}(n,k)\mathbf{d}^T]\hat{\mathbf{X}}(n|n-1,k) + \mathbf{K}(n,k)|Y(n,k)| \tag{2.26}$$

The accuracy of the weighted sum producing the updated state is critical in determining the correctness of the algorithm. Hence, it is imperative that the noise power is estimated as accurately as possible. There are a number of ways to do so, as discussed in Section 2.2.

As the algorithm is running, each modulating signal $|Y(n,k)|$ is windowed into modulation frames, and the LPCs and excitation variance $\sigma^2_{W(k)}$ are estimated. Within each frame, the LPCs are kept constant, whereas the Kalman gain $\mathbf{K}(n,k)$, error covariance matrix $\mathbf{P}(n|n,k)$ and estimated state vector $\hat{\mathbf{X}}(n|n,k)$ are updated every sample, regardless of frame.

2.6.2 Comparison with Time-Domain Kalman Filter

For comparison purposes, the time-domain Kalman filter (TDKF) equations are shown below:

$$\mathbf{P}(n|n-1) = \mathbf{A}\mathbf{P}(n-1|n-1)\mathbf{A}^T + \sigma^2_w\mathbf{d}\mathbf{d}^T \tag{2.27}$$

$$\hat{\mathbf{x}}(n|n-1) = \mathbf{A}\hat{\mathbf{x}}(n-1|n-1) \tag{2.28}$$

$$\mathbf{K}(n) = \mathbf{P}(n|n-1)\mathbf{d}[\sigma^2_v + \mathbf{d}^T\mathbf{P}(n|n-1)\mathbf{d}]^{-1} \tag{2.29}$$

$$\mathbf{P}(n|n) = [\mathbf{I} - \mathbf{K}(n)\mathbf{d}^T]\mathbf{P}(n|n-1) \tag{2.30}$$

$$\hat{\mathbf{x}}(n|n) = \hat{\mathbf{x}}(n|n-1) + \mathbf{K}(n)[y(n) - \mathbf{d}^T\hat{\mathbf{x}}(n|n-1)] \tag{2.31}$$

where $\hat{\mathbf{x}}(n|n-1)$ and $\hat{\mathbf{x}}(n|n)$ are the *a priori* (predicted) and *a posteriori* (updated) state vectors respectively, $\mathbf{P}(n|n-1)$ and $\mathbf{P}(n|n)$ are the *a priori* and *a posteriori* error covariance matrices respectively, $\mathbf{K}(n)$ is the Kalman gain, and σ^2_v, σ^2_w are the noise and excitation variances respectively.

Figure 2.3 compares the spectrograms of TDKF (order 18) and MDKF (order 2) applied on speech from the TIMIT database [29], corrupted by white Gaussian noise at 5 dB SNR, and sampled at 16 kHz. For the purposes of comparing performance limits, clean speech LPCs were used in the filters. Generally, both algorithms perform well in removing noise, especially when speech is absent. However, there is visibly some noise in the TDKF-enhanced speech; particularly, frequency components above 1.8 kHz have been noticeably degraded by noise. A listening test confirmed this, detecting the presence of high-frequency artifacts.

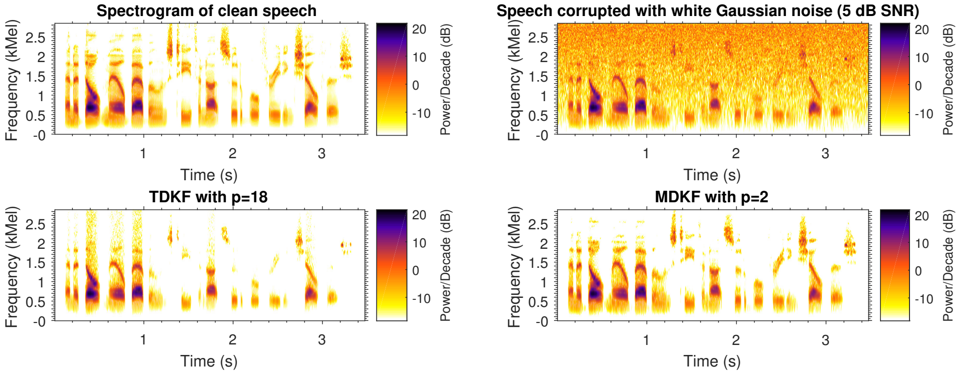

Figure 2.3: Top row: clean speech (left), speech corrupted with white Gaussian noise (right); bottom row: TDKF-enhanced speech (left), MDKF-enhanced speech (right)

This TDKF noise is a limitation of using the KF for speech enhancement. Similarly to how we rearranged Equation 2.25, Equation 2.31 can be rewritten to show that the enhanced output is a weighted combination of the estimated speech and measured speech, where the relative weight depends on the Kalman gain $\mathbf{K}(n)$. When speech is absent, $\mathbf{P}(n|n-1) = \mathbf{0}$, meaning that $\mathbf{K}(n) = \mathbf{0}$ and the estimated state is equal to the predicted state, being unaffected by the noisy measurement.

When speech is present, however, the algorithm does not work quite so perfectly. Using a low model order means that the TDKF linear predictor cannot fully replicate the harmonic structure of speech, which requires autocorrelation lags in the order of the number of samples in a pitch period [52]. The prediction thus has unvoiced and noise-like characteristics, and the result is that the updated output only preserves the speech component below 1.8 kHz [52]. The resultant noise will be especially prevalent in regions of low SNR, where the prediction is weighted more heavily due to Equation 2.29 producing a smaller $\mathbf{K}(n)$.

MDKF has an inherent advantage over the TDKF due to the linear predictor model used. Compared to the TDKF, the MDKF models the temporal variation of the acoustic magnitude spectrum of speech, which represents the changes in the vocal tract over time. This is more comprehensive since the low-order MDKF linear predictors are able to model the modulating signal dynamics, due to physiological limitations of how quickly the vocal tract can change [51]. This also better represents speech information overall, as it has been shown that low-frequency modulations of sound are the fundamental carriers of speech information [50].

2.6.3 Performance of MDKF

Overall, experimental results from the TIMIT corpus (Figure 2.3) show that under ideal conditions where clean speech LPCs can be obtained accurately, the linear predictor is sufficient to model the modulating signals of clean speech. As described earlier, the vocal tract tends to change slowly due to physiological constraints, and thus low LPC orders ($p = 2$) were found to be sufficient. Using this, the MDKF was by far the best performing algorithm, doing better than all acoustic and time-domain methods tested, including the TDKF, for both white and coloured noise [52]. This was despite both algorithms having access to clean speech LPCs.

However, clean speech is not available in reality; the presence of noise generally degrades the LPC estimates, worsening the performance of the MDKF algorithm. In [52], a practical MDKF algorithm was evaluated, which used an acoustic-domain pre-processor for LPC estimation to reduce the effect of noise degradation.

2.7 Speech Quality

The perceived overall speech quality is how “good” the quality of the speech is. The definition of “good” is typically left to the listener, who then gives a score to the speech. Methods to assess speech quality can be grouped into subjective and objective measures.

2.7.1 Subjective Speech Quality Measures

Subjective quality measures typically compare the original and processed speech by a listener or a group of listeners. The listeners subjectively rate or rank the speech according to a predetermined scale. Since every listener is unique, their ratings will vary; this variation in results can be reduced by averaging the scores from a group of listeners.

Mean Opinion Score

A widely used subjective quality measure is the Mean Opinion Score (MOS) [56]. Each listener gives a numeric MOS score, typically in the range $1 - 5$, where 1 is the lowest perceived quality and 5 is the highest perceived quality. The “Absolute Category Rating” scale is commonly used, as shown in Table 2.1 [57]. The overall score is obtained by averaging the ratings from all listeners, representing an overall perceived quality of the speech. With a large number of speech files, this test can be costly and time consuming.

Rating	**Label**
1	Excellent
2	Good
3	Fair
4	Poor
5	Bad

Table 2.1: Categories of MOS: Absolute Category Rating

2.7.2 Objective Speech Quality Measures

On the other hand, objective speech quality use physical measurements and some calculated values from these measurements. Typically, these calculations compare objective measurements for the reference clean speech and the distorted speech.

Many of the objective measures are highly correlated with subjective measures; it is thus common for a test to use objective measures to estimate subjective methods, which are usually more time-consuming and costly as they involve human listeners. However, as noted previously, there are situations in which high objective scores do not produce high subjective scores and vice versa.

SNR

Signal-to-Noise Ratio (SNR) is one of the oldest and most widely used objective quality methods. It has low computational complexity, but requires both clean and distorted speech. The classic formula is calculated (in dB) as:

$$SNR = 10\log_{10}\frac{\sum\limits_{n=1}^{N} x^2(n)}{\sum\limits_{n=1}^{N}\{x(n)-\hat{x}(n)\}^2} \tag{2.32}$$

where $x(n)$ is the clean speech, $\hat{x}(n)$ is the distorted speech and N is the number of time-domain samples.

However, this formula does not represent actual speech quality well as it averages over the entire signal even though speech is non-stationary. Speech energy fluctuates over time, and this formula is dominated by parts where speech energy is large and noise energy is small, which is not representative of the entire signal.

Modifications have thus been proposed. To better represent the temporal variation of speech, segmental SNR (segSNR) was proposed to calculate SNR in short frames and take the average:

$$SNR_{seg} = \frac{10}{M}\sum_{m=0}^{M-1}\log_{10}\frac{\sum\limits_{n=Lm}^{Lm+L-1} x^2(n)}{\sum\limits_{n=Lm}^{Lm+L-1}\{x(n)-\hat{x}(n)\}^2} \tag{2.33}$$

where L is the frame length in number of samples and M is the number of frames in the signal ($N = ML$). The logarithm of the ratio is computed before averaging, which means that frames with unusually large ratios are weighted less while frames with lower ratios are weighted more. This matches the perceptual quality better, ensuring that frames with large speech and low noise do not unfairly dominate the overall ratio.

However, if the speech contains too much silence, the overall segSNR value decreases significantly as silent frames usually give large negative segSNR values. In this case, silent frames should be excluded from the averaging by using speech activity detectors. Similarly, frames which exhibit excessively large or small speech values should also be excluded. These modifications give segSNR values that match the subjective quality better. As a result, segSNR often has upper and lower bounds of 35 dB and −10 dB respectively [58].

A separate variation of SNR is the frequency-weighed SNR (fwSNRseg), which weights the contribution of the different frequency bands. The fwSNRseg can be defined as:

$$fwSNR_{seg} = \frac{10}{M} \sum_{m=0}^{M-1} \frac{\sum\limits_{j=0}^{K-1} W(j,m) \log_{10} \frac{X(j,m)^2}{\{X(j,m) - \hat{X}(j,m)\}^2}}{\sum\limits_{j=0}^{K-1} W(j,m)} \tag{2.34}$$

where $W(j,m)$ is the weight of the j^{th} frequency band in the m^{th} frame, K is the number of frequency bands and $X(j,m)$, $\hat{X}(j,m)$ are the spectral amplitude of the clean and distorted speech respectively. The weights can be chosen in many ways, one of which is the ANSI SII Standard [59].

Perceptual Evaluation of Speech Quality

One of the most popular objective speech quality measures is the ITU-T P.862: Perceptual Evaluation of Speech Quality (PESQ) [3].

PESQ was developed to model subjective tests commonly used to assess the voice quality by human beings (e.g. MOS), using true voice samples as test signals. It is designed for use over a wide range of conditions. A mapping from PESQ to MOS scores was standardised, allowing PESQ results to model MOS scores that range from 1 (Bad) to 5 (Excellent) (typical of Table 2.1). The average correlation between PESQ-mapped MOS scores and subjective MOS for a number of tests was a high score of 0.935 [60]. The block diagram of PESQ is shown in Figure 2.4 (taken from [61]).

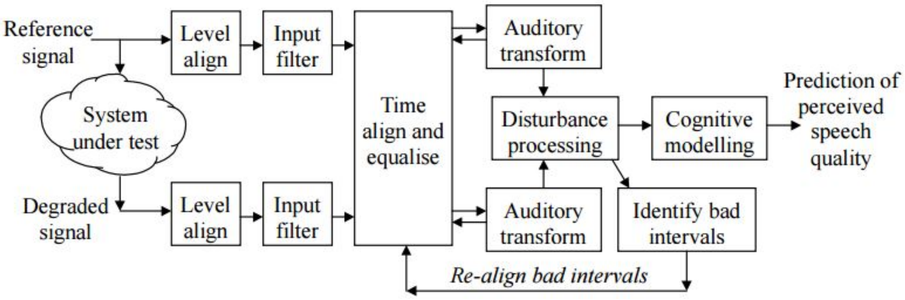

Figure 2.4: Structure of PESQ model (taken from [61])

2.8 Speech Intelligibility

Speech intelligibility is the accuracy with which we can hear what is being said, and is a different performance measure as compared to perceived speech quality. Specifically, it is measured as the percentage of correctly identified words relative to the number of words. Instead of words, one may also use phonemes or syllables as the test unit. If words or complete sentences are used, they typically encompass linguistically meaningful units, and thus the choice of test words is important to ensure a fair assessment.

Although there does not exist a completely clear relationship between perceived speech quality and intelligibility, there exists some correlation between the two. Generally, “good” quality speech also gives high intelligibility and vice versa. However, this generalisation does not always hold; there are some samples that are highly intelligible yet are perceived as “poor” quality and vice versa.

2.8.1 Short-Time Objective Intelligibility

A widely-used method to evaluate intelligibility is the Short-Time Objective Intelligibility (STOI) measure [4]. The basic structure is shown in Figure 2.5.

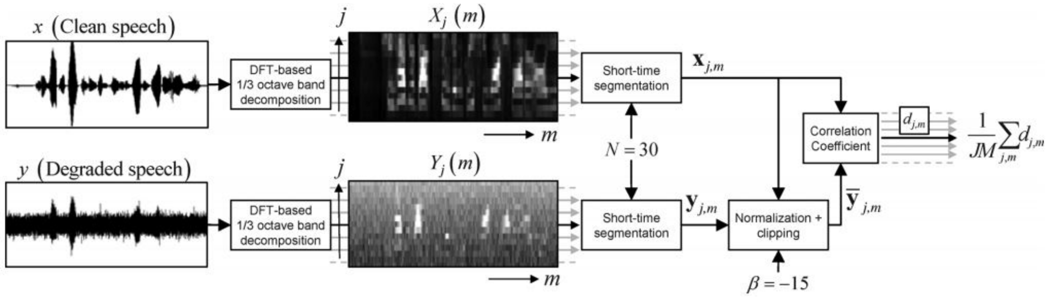

Figure 2.5: Structure of STOI model (taken from [62])

STOI requires both clean and noisy speech. Both of these are first time-frequency (T-F) decomposed to obtain a representation resembling the transform properties of the auditory system [62].

STOI contains an intermediate-stage intelligibility measure, which compares the temporal envelopes of the clean and degraded speech in short-time segments using a correlation coefficient. Before this comparison, the temporal envelopes are first normalised and clipped. The normalisation process compensates for global level differences, which should not dominate the speech intelligibility, while clipping prevents over-sensitivity of the model towards excessively degraded T-F units. This ensures that if a T-F unit is already deemed unintelligible, further corruption of this unit does not lead to a lower intelligibility prediction.

These intermediate intelligibility measures are then averaged to a single value ranging from 0 to 1, where 1 represents perfect intelligibility i.e. 100% of words can be detected accurately. Results have demonstrated that this value is highly correlated with the true speech intelligibility of noisy speech from multiple listening experiments [62].

Chapter 3

Problem Analysis

In this chapter, the project aims are analysed and broken down into specific deliverables. Following that is a discussion on the implementation of the baseline algorithm and how the proposed modifications will be made. Subsequent chapters discuss these modifications and the performance evaluation in greater detail.

3.1 Deliverables

The goal is to modify a Modulation-Domain Kalman Filter (KF) speech enhancement method by integrating data from an ideal binary mask (IBM). This project proposes a few methods to do so.

In the first method, IBM statistics are used to scale the predicted value and variance in the MDKF. This requires an implementation of the MDKF, and also involves obtaining the IBM statistical information. This can be broken down into a few deliverables, as follows:

1) Implement a working IBM (algorithm from [33])

2) Obtain statistical information from IBM from training data

3) Implement a working MDKF (algorithm from [52])

4) Apply IBM information in a useful way to improve MDKF iterations

A second proposed method is to improve the linear prediction coefficients (LPCs) using the IBM information. In the MDKF, LPCs are estimated in each modulation frame. The goal is to study if these LPCs can be improved by incorporating data provided by an IBM:

1) Apply weighted sum to LPC estimation using weights determined from IBM

The final proposed method is to improve the noise estimation of the MDKF with the IBM, using a weighted sum to tweak parts of the noise estimation algorithm, including the Signal-to-Noise Ratio (SNR) and speech presence probability (SPP):

1) Apply weighted sum to SNR and SPP in noise estimation using weights determined from IBM

After implementing these methods, their performance will be evaluated using the Signal-to-Noise Ratio (SNR), Perceptual Evaluation of Speech Quality (PESQ) and Short-Time Objective Intelligibility (STOI) measures:

1) For a range of input noise SNR levels, run all algorithms

2) Evaluate using measures SNR, PESQ, STOI

This set of requirements and tests are necessary to evaluate if the proposed algorithms are able to provide improvements over existing methods. For evaluation, the testing methodology is standardised to ensure a fair assessment, and will be discussed later. All methods are also evaluated and averaged over multiple input speech samples to reduce the effect of outliers.

3.2 Implementation

In this project, the MDKF in Section 2.6.1 is used as a baseline algorithm. For the base MDKF, an acoustic frame length of 16 ms was used with a 4 ms frame shift. For each frequency bin, a modulation frame of 24 ms was used with a 4 ms frame increment to determine the LPC coefficients. An MDKF model order of $p = 2$ was used. In most experiments, these settings are used for all algorithms to compare and evaluate performance.

3.2.1 Optimal Modulation Frame Length

In [63], it was shown that short modulation frame durations of $10-32$ ms retain good intelligibility overall as compared to longer frame lengths. This is briefly verified below.

Using clean speech as the “noisy” input speech to the MDKF filter, a plot of the average LPC excitation variance per modulation frame against a range of modulation frame lengths is shown in Figure 3.1, where the error is normalised against the length of the modulation frame and averaged over 38 randomly chosen speech samples from the TIMIT dataset.

Figure 3.1 clearly shows that the error decreases as the modulation frame length increases. This makes sense intuitively as the modulation domain represents the variation in speech, which changes much more slowly than the actual speech itself.

However, the numerical error does not fully represent the human perception of speech. Indeed, Figures 3.2 and 3.3 show that the PESQ and STOI scores decrease as the modulation frame length increases. It was empirically found through informal listening tests that a $16 - 24$ ms modulation frame length produced acceptable levels of noise, while being more intelligible than speech enhanced using longer frame lengths, which agrees with the conclusions in [63]. This project thus uses a modulation frame length of 24 ms with a 4 ms frame shift for an overall balance of quality and intelligibility. Given a 4 ms acoustic frame shift, the modulation domain sampling frequency is 250 Hz, which gives 6 samples per modulation frame.

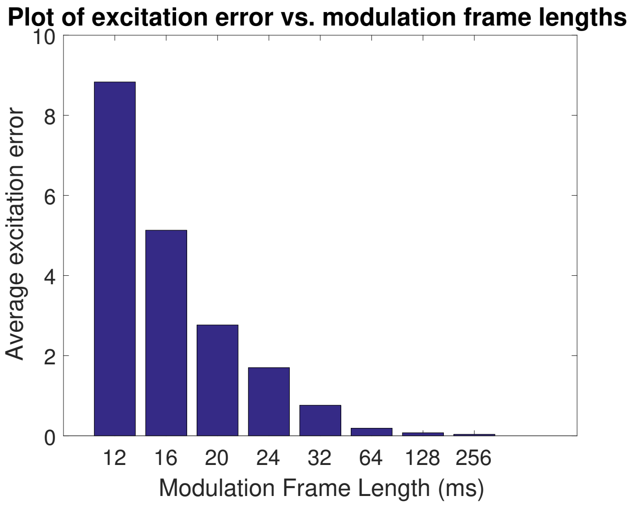

Figure 3.1: Plot of normalised excitation variance vs. modulation frame lengths

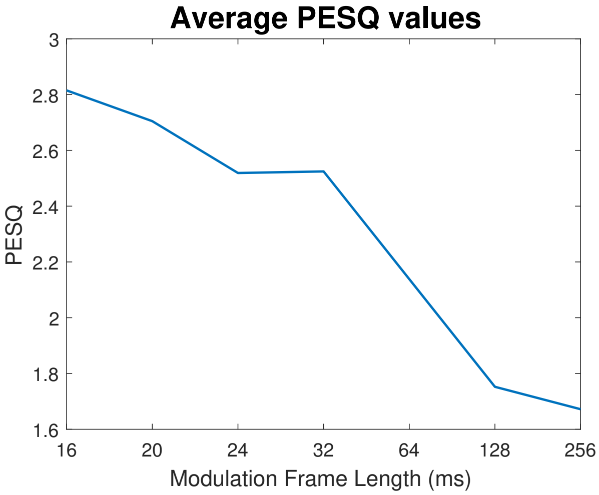

Figure 3.2: Average PESQ scores vs. modulation frame lengths

3.3 Kalman Filter Framework

The base framework for the proposed speech enhancement algorithms in this project are shown in Figure 3.4, which is based on a Modulation-Domain Kalman Filter (MDKF) as described in Section 2.6.1. Modifications are then proposed to this baseline algorithm.

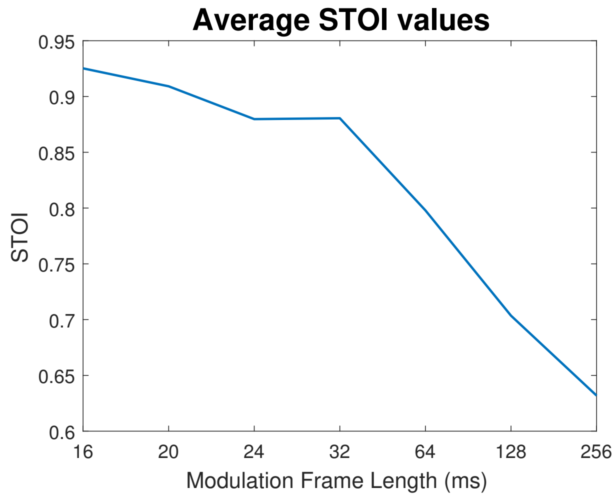

Figure 3.3: Average STOI scores vs. modulation frame lengths

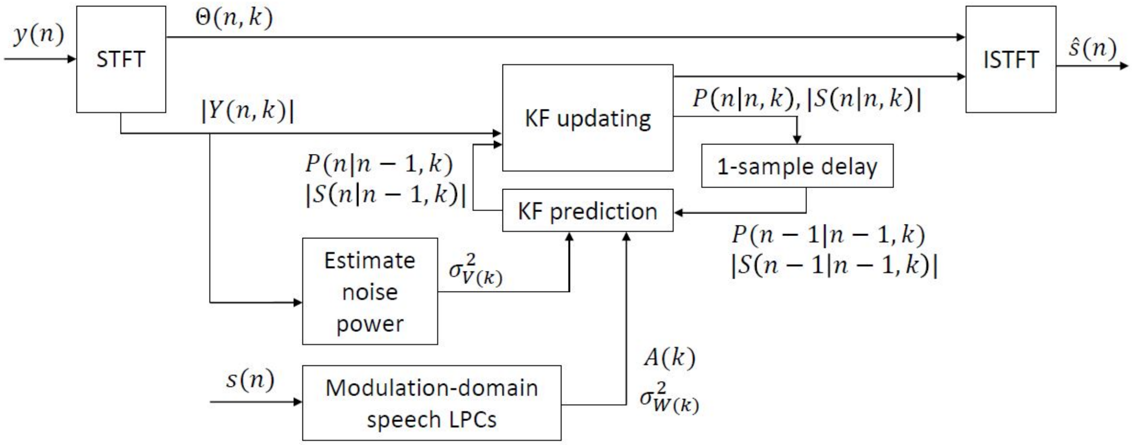

Figure 3.4: Block diagram of baseline MDKF model

The noisy input time-domain speech signal $y(n)$ first undergoes the STFT transform. The amplitude spectrum $|Y(n,k)|$ is used as input to the MDKF, and is also used to estimate the noise power spectrum. For a theoretical investigation of best performance possible, the LPC coefficients are estimated clean speech $s(n)$; specifically, they are estimated from the spectral amplitudes of the clean speech in each frequency bin. After processing through the Kalman filter, the output enhanced magnitudes are combined with the noisy phase spectrum $\theta(n,k)$, then processed through an inverse STFT to produce the output processed speech.

Chapter 4

Testing Methodology

In this project, the objective is to propose modifications to existing speech enhancement algorithms, with the goal of improving the two aspects of speech quality: the overall perceived speech quality, and the speech intelligibility [64].

4.1 Assessing Speech Quality

The perceived overall speech quality is how “good” the quality of the speech is. The assessment is left to the listener who scores the speech according to a pre-defined rating system.

Methods to assess speech quality can be grouped into subjective and objective measures. As discussed before, many of the objective measures are highly correlated with subjective measures; it is thus common for a test to use objective measures, which are less time-consuming and cheaper, to approximate subjective methods.

In this report, the quality of the enhancement algorithms will be assessed using segSNR and PESQ. The effect of the enhancements on noise level will be assessed by segmental SNR (segSNR) while speech quality will be evaluated by Perceptual Evaluation of Speech Quality (PESQ).

4.2 Assessing Speech Intelligibility

On the other hand, speech intelligibility is the accuracy with which we can hear and identify what is being said. Typically, it is measured as the percentage of correctly identified words relative to the total number of words. In this report, speech intelligibility of the proposed speech enhancers will be evaluated using Short-Time Objective Intelligibility (STOI).

4.3 Speech Database: TIMIT

The TIMIT Acoustic-Phonetic Continuous Speech Corpus of read speech was designed to provide speech data for acoustic-phonetic studies as well as the development and evaluation of automatic speech recognition systems [29]. It is widely used in the research and testing of speech enhancement algorithms. A combined effort between the Massachusetts Institute of Technology (MIT), SRI International (SRI) and Texas Instruments, Inc. (TI), the TIMIT database contains recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences, each of which is a few seconds long. The recordings are 16-bit resolution, 16 kHz rate speech waveform files, and the database also includes time-aligned phonetic and word transcriptions.

Some of the proposed algorithms require training samples; a set of 38 randomly chosen samples from the TIMIT training database was used for such training purposes. For general performance testing, a separate set of 38 randomly chosen speech samples are used. These speech samples are corrupted by white noise at SNRs between -20 and 20 dB.

Chapter 5

Modified Kalman Filter Inputs

5.1 Introduction

From the modulation-domain Kalman filter (MDKF) iteration equations (Equations 2.21 to 2.26), we know that two variables are used to form the updated state: the prediction of the current state and the observation of the current state. The observation is noisy; we can improve the accuracy of the updated state, and hence the algorithm performance, if the observation is modified to better represent the actual speech.

In this chapter, we propose to incorporate information from an ideal binary mask (IBM) to improve the accuracy of the updated state. This can be done in two ways: in the first method, the observation is tweaked directly, while in the second method, the updated state is instead constructed by combining information from the observation, the prediction and the mask altogether. Both methods are detailed and evaluated in this chapter.

5.2 Incorporating Binary Mask into Observation

From Section 2.4, we know that the 0 dB-threshold IBM produces a mask of 1s and 0s, where 1s represent T-F units where the signal has higher energy than noise, and vice versa for 0s, and that this threshold gives the best-performing mask overall. The observation used in the MDKF equations (Section 2.6.1) is the original input noisy observation; if it is modified to better represent the inherent underlying speech (or silence in absence of speech), the algorithm can perform better. One way to do so is to incorporate the statistical quantities of an ideal binary mask, by multiplying together the probability density functions (PDFs) of the observation and the mask.

5.2.1 Gaussian Product

Let $f(x)$ and $g(x)$ be two Gaussian PDFs with arbitrary means a and b and variances A and B, which we represent as $f(x) \sim \mathcal{N}(x; a, A)$ and $g(x) \sim \mathcal{N}(x; b, B)$ respectively. Their product is also a product of two Gaussian PDFs [65], of which one term is dependent on x while the other is independent of x:

$$\mathcal{N}(x; a, A)\ \mathcal{N}(x; b, B) = \mathcal{N}(a; b, A + B)\ \mathcal{N}(x; c, C) \tag{5.1}$$

where C and c are defined as:

$$C = (A^{-1} + B^{-1})^{-1} = \frac{AB}{A + B} \tag{5.2}$$

$$c = C(\frac{a}{A} + \frac{b}{B}) = \frac{Ab + Ba}{A + B} \tag{5.3}$$

In the context of the MDKF, we are interested in $\mathcal{N}(x; c, C)$, the PDF dependent on x. Given an observation y and clean speech s, we can represent their joint probability $p(s, y)$ in two ways:

$$p(s, y) = p(s)\ p(y|s) = p(y)\, p(s|y) \tag{5.4}$$

Using Bayes' Theorem, we can express this as:

$$p(s|y) = \frac{p(y|s)\ p(s)}{p(y)} \tag{5.5}$$

where $p(s)$ is the prior probability of the predicted speech, $p(y)$ is the probability of the observation, $p(s|y)$ is the conditional probability of the speech given the observation and $p(y|s)$ is the conditional probability of the observation given speech.

In the MDKF, we want $p(s|y)$ i.e. we wish to estimate the clean speech given the noisy measurement. To compute this, we require the probabilities $p(y|s)$ and $p(s)$ (Equation 5.5). We already have $p(y|s)$ as the noisy observation given clean speech, which comes from the measurement and the assumption of an additive noise model $y = s + n$, where n is the noise. The other term, $p(s)$, can be obtained from an IBM, where we assume that the mask is as accurate as possible; how such a mask is generated in a real-world scenario is not discussed in this report.

Mapping the available and desired probabilities to Equation 5.1, we have available $f(s) = p(s)$ and $g(s) = p(y|s)$. By multiplying these two terms together, we expect a term independent of s and a term dependent on s, which are $p(y)$ and $p(s|y)$ respectively. The latter term, which is the conditional probability of the (true) clean speech given the noisy observation, is what we propose to replace the noisy observation term in the MDKF equations.

5.2.2 Training Mask Statistics

The method that we propose to obtain $p(s)$ requires statistical information from an IBM. To obtain the PDF, we need its mean and variance. The mean and variances were calculated separately for mask 1s and 0s i.e. separate PDFs were generated for when the mask indicates that speech is dominant and when noise is dominant. This was done separately for each frequency bin.

To generate the IBM, the STFT of the signal is first taken, producing a matrix of time-frequency (T-F) units, and the unit-wise signal-to-noise ratio (SNR) is then computed to produce a binary-valued matrix. In the MDKF setup, the same STFT of the signal is taken; the mask thus provides a speech-dominant/noise-dominant indicator for each T-F unit of the STFT-processed signal.

To generate the speech-dominant PDF for one frequency bin in a speech signal, we picked out the samples of the amplitude spectrum of this frequency bin which had corresponding (the same T-F location) mask values indicating 1. The mean and variance of these samples was then computed. The same process was done to compute the noise-dominant PDF for the same frequency bin. This was done for each frequency bin, and the values were averaged over all speech samples in the training dataset, which were corrupted by white noise at 5 dB SNR. To save time, training of mask statistics was done once, with its values stored offline to pull when needed.

5.2.3 Modifying Observation

With the IBM statistics, the observation can then be modified. For reference, the MDKF equations are replicated from Section 2.6.1 and shown below:

$$\mathbf{P}(n|n-1,k) = \mathbf{A}(k)\mathbf{P}(n-1|n-1,k)\mathbf{A}(k)^T + \sigma^2_{W(k)}\mathbf{d}\mathbf{d}^T \tag{5.6}$$

$$\hat{\mathbf{X}}(n|n-1,k) = \mathbf{A}(k)\hat{\mathbf{X}}(n-1|n-1,k) \tag{5.7}$$

$$\mathbf{K}(n,k) = \mathbf{P}(n|n-1,k)\mathbf{d}[\sigma^2_{V(k)} + \mathbf{d}^T\mathbf{P}(n|n-1,k)\mathbf{d}]^{-1} \tag{5.8}$$

$$\mathbf{P}(n|n,k) = [\mathbf{I} - \mathbf{K}(n,k)\mathbf{d}^T]\mathbf{P}(n|n-1,k) \tag{5.9}$$

$$\hat{\mathbf{X}}(n|n,k) = \hat{\mathbf{X}}(n|n-1,k) + \mathbf{K}(n,k)[|Y(n,k)| - \mathbf{d}^T\hat{\mathbf{X}}(n|n-1,k)] \tag{5.10}$$

where the observation mean is $|Y(n,k)|$ and the observation variance is in $\sigma^2_{V(k)}$. As the MDKF loops through each sample in each modulating signal, it tweaks these parameters for every sample. In each iteration, the algorithm checks the corresponding T-F position in the mask if it indicates a 1 or 0, then brings up the relevant mask mean and variance. Using Equations 5.2 and 5.3, a scaled mean and variance is obtained for the current observation input by multiplying with the relevant mask PDF. This process replaces the original observation mean $|Y(n,k)|$ and variance $\sigma^2_{V(k)}$ in the

MDKF equations with $|Y\_scaled|$ and $\sigma^2_{V(k)\_scaled}$ respectively. We thus get modified versions of Equations 5.8 and 5.10:

$$\mathbf{K}(n,k) = \mathbf{P}(n|n-1,k)\mathbf{d}[\sigma^2_{V(k)\_scaled} + \mathbf{d}^T\mathbf{P}(n|n-1,k)\mathbf{d}]^{-1} \tag{5.11}$$

$$\hat{\mathbf{X}}(n|n,k) = \hat{\mathbf{X}}(n|n-1,k) + \mathbf{K}(n,k)[|Y\_scaled| - \mathbf{d}^T\hat{\mathbf{X}}(n|n-1,k)] \tag{5.12}$$

5.3 Modified Kalman Filter Equations

The IBM can be used in a slightly different way; instead of tweaking the observation itself, information from the mask can be directly used in the KF equations. Now, the KF equations use three pieces of information (prediction, observation, binary mask) to estimate the current state, rather than just the former two as in the original KF equations.

5.3.1 Decoupling Kalman Filter Equations

To insert the mask information, the KF equations first need to be decoupled. For the scalar output case i.e. where $|Y(n,k)|$ is a scalar, with $\mathbf{d} = [1, 0, ..., 0]^T$, the observation can be decorrelated from the rest of the state vector. Since each modulating signal has a separate Kalman filter, we remove the frequency bin subscript k in this section, and also indicate the time sample index in the subscript for clarity e.g. we represent $\mathbf{P}(n|n-1,k)$ as $\mathbf{P}_{n|n-1}$. After obtaining the *a priori* error covariance matrix (Equation 5.6), we can decompose it as:

$$\mathbf{P}_{n|n-1} = \begin{bmatrix} g_n & \mathbf{g}_n^T \\ \mathbf{g}_n & \mathbf{G}_n \end{bmatrix} \tag{5.13}$$

noting that the covariance matrix is symmetric. We then apply a transformation $\mathbf{R}_n$ to the state vector $\mathbf{X}_{n|n-1}$ to obtain:

$$\mathbf{z}_{n|n-1} = \mathbf{R}_n\mathbf{X}_{n|n-1} = \begin{bmatrix} 1 & \mathbf{0}^T \\ -g_n^{-1}\mathbf{g}_n & \mathbf{I} \end{bmatrix} \mathbf{X}_{n|n-1} \tag{5.14}$$

The covariance matrix of $\mathbf{z}$ is given by (omitting subscript n for clarity):

$$\begin{aligned} \langle \mathbf{z}\mathbf{z}^T \rangle &= \mathbf{R}\langle \mathbf{X}\mathbf{X}^T \rangle \mathbf{R}^T = \mathbf{R}\mathbf{P}\mathbf{R}^T \\ &= \begin{bmatrix} 1 & \mathbf{0}^T \\ -g^{-1}\mathbf{g} & \mathbf{I} \end{bmatrix} \begin{bmatrix} g & \mathbf{g}^T \\ \mathbf{g} & \mathbf{G} \end{bmatrix} \begin{bmatrix} 1 & -g^{-1}\mathbf{g} \\ \mathbf{0}^T & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} g & \mathbf{0}^T \\ \mathbf{0} & \mathbf{G} - g^{-1}\mathbf{g}\mathbf{g}^T \end{bmatrix} \end{aligned} \tag{5.15}$$

which shows that the first element of $\mathbf{z}$ is uncorrelated with the rest of $\mathbf{z}$, and is distributed as $\mathcal{N}(\mathbf{d}^T\mathbf{z}_{n|n-1}, g)$. This can then be combined with the observation to obtain the distribution of the updated state, which produces the same formulation as the original Kalman filter equations.

5.3.2 Incorporating IBM into decoupled KF equations

With the mask information available, the mask can be combined with the observation and decoupled state $\mathbf{z}$ to improve the KF iterations. Unlike Section 5.2, now the product of three distributions has to be taken, which is done pairwise using the steps in Section 5.2.1.

Assume that the observation has been combined with the relevant mask statistics (recall that speech-dominant "1"s and "0"s are distributed separately) using the steps covered in Section 5.2.1, and assume that the result is distributed as $\mathcal{N}(y, r)$. This is then combined with the first element of $\mathbf{z}$ to obtain the posterior distribution $\mathcal{N}(\mathbf{d}^T\mathbf{z}_n; \frac{gy_n + r\mathbf{d}^T\mathbf{z}_{n|n-1}}{g+r}, \frac{gr}{g+r})$.

Its mean value is

$$\mathbf{d}^T\mathbf{z}_{n|n} = \frac{1}{g+r}(gy + r\mathbf{d}^T\mathbf{z}_{n|n-1}) = \mathbf{d}^T\mathbf{z}_{n|n-1} + \frac{g}{g+r}(y - \mathbf{d}^T\mathbf{z}_{n|n-1}) \tag{5.16}$$

For the entire transformed state $\mathbf{z}_n$, we can write

$$\mathbf{z}_{n|n} = \mathbf{z}_{n|n-1} + \frac{g}{g+r}(y - \mathbf{d}^T\mathbf{z}_{n|n-1})\mathbf{d} \tag{5.17}$$

Finally, the original state $\mathbf{X}_{n|n}$ can be obtained with the reverse transformation:

$$\begin{aligned}\mathbf{X}_{n|n} &= \mathbf{A}^{-1}\mathbf{z}_{n|n} \\ &= \begin{bmatrix} 1 & \mathbf{0}^T \\ g^{-1}\mathbf{g} & \mathbf{I} \end{bmatrix} \mathbf{z}_{n|n}\end{aligned} \tag{5.18}$$

5.4 Performance Results and Discussion

In this section, the IBM-modified MDKF algorithm uses the modification described in Section 5.3, and will be termed BMMDKF. The block diagram for the BMMDKF is shown in Figure 5.1, highlighting the IBM modification in bold.

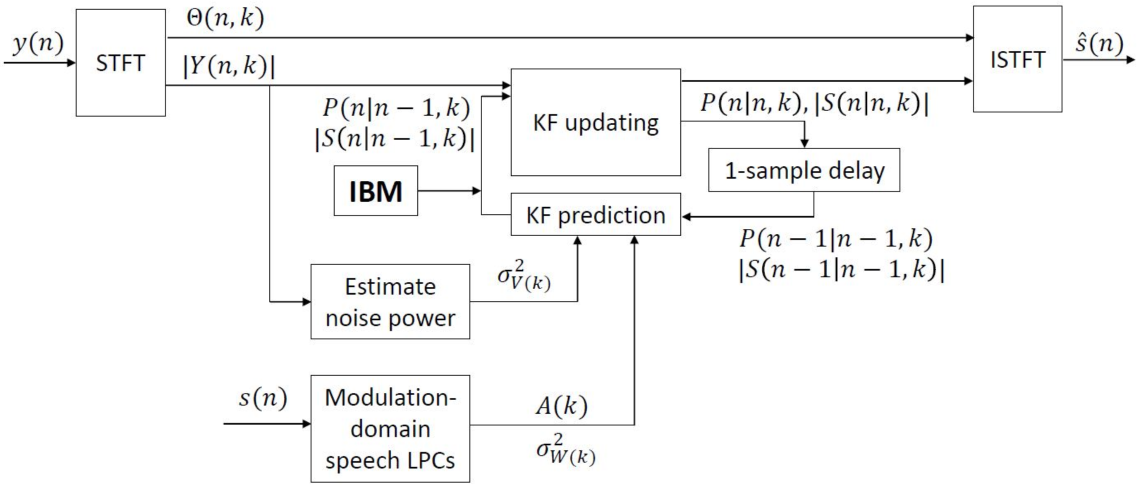

Figure 5.1: Block diagram of IBM-modified MDKF (BMMDKF)

To evaluate the modified algorithm, the BMMDKF will be compared to a few control algorithms: the original MDKF of Section 2.6.1, the MMSE speech enhancement algorithm in [66] and the original input speech corrupted by white noise. These are respectively named MDKF, MMSE and Noisy in the plots that follow. All methods are evaluated for segSNR, PESQ and STOI, and are tested over the input training dataset described in Section 4.3, and the results are normalised over all input samples. To standardise testing, the parameters used for all algorithms generally follow those described in Chapter 3.2. They are shown in Table 5.1.

Parameter	**Value**
Sampling frequency	16 kHz
Acoustic frame length	16 ms
Acoustic frame shift	4 ms
Modulation frame length	24 ms
Modulation frame shift	4 ms
Windowing function	Hamming window
MDKF model order	2
LPCs generated from	clean speech
Input speech corrupted by	white noise
IBM SNR threshold (LC)	0 dB

Table 5.1: List of parameters used to evaluate BMMDKF and other algorithms

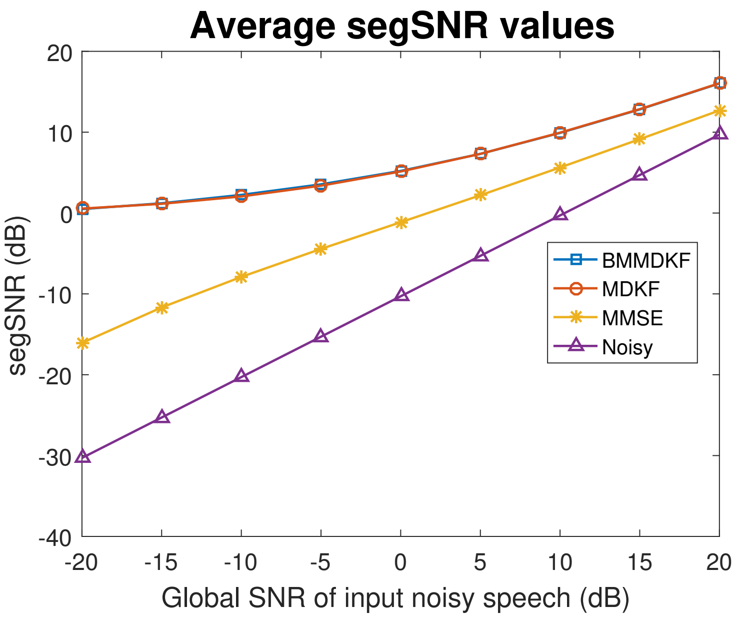

Figure 5.2: Average segSNR values of BMMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels

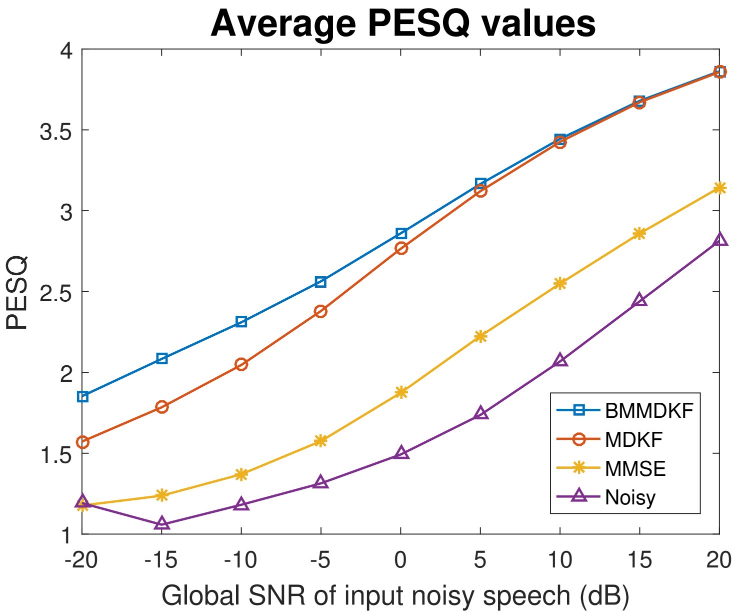

Figure 5.3: Average PESQ values of BMMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels

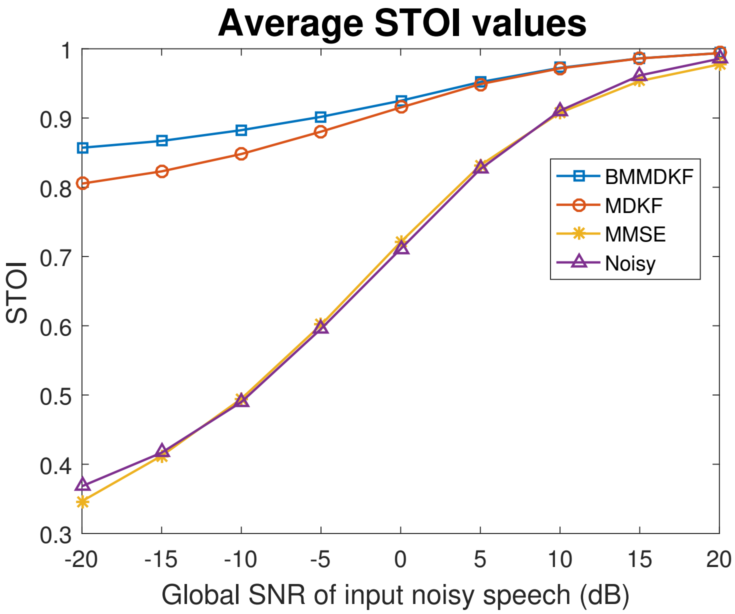

Figure 5.4: Average STOI values of BMMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels

From Figure 5.2, the segSNR of BMMDKF and MDKF are very similar. However, it is generally true that the segSNR of BMMDKF is slightly larger at input SNRs from -20 to 0 dB.

The enhancement provided by BMMDKF is more obvious in Figures 5.3 and 5.4, in the PESQ and STOI scores respectively. The average PESQ scores of BMMDKF are higher than that of MDKF across all input SNR, and show particularly large increase over the -20 to 5 dB input SNR range: the BMMDKF PESQ scores are on average higher than that of MDKF by 0.1945, a 10.0174% increase over the MDKF. This improvement rises to 0.2245 or 11.7359% if we consider the -20 to 0 dB input SNR range. Intelligibility scores also show advancement. Over the -20 to 0 dB input SNR range, STOI scores of BMMDKF are on average higher than that of MDKF by 0.0323 or 3.8646%. Overall, these results show a significant improvement in both perceived quality and intelligibility of enhanced speech.

It is clear from Figures 5.3 and 5.4 that the refinement provided by the IBM modification rises with a reduced global input SNR. This trend is unsurprising, as it is generally much more difficult to improve on a very good score as compared to a low score. When the input SNR is high or satisfactory, including the range 0 to 20 dB, the baseline MDKF already performs very well. For example, PESQ scores are in the range $3 - 4$, and STOI scores are at 0.94 and above; these are very high scores and are difficult to significantly improve on, and thus the BMMDKF shows only small improvements when the input SNR is high. When the input SNR is very low, however, the BMMDKF performance is markedly more enhanced; at -20 dB input, its PESQ and STOI scores are 17.86% and 6.43% better than the MDKF respectively.

5.5 Conclusion

This chapter proposes applying an optimal IBM to a training dataset to obtain averaged statistical information corresponding to speech-dominant and noise-dominant regions of white noise-corrupted speech. It is proposed to use this information to improve the parameters used in the MDKF iteration equations. This modification is termed the BMMDKF algorithm.

Performance results show that the BMMDKF shows very similar segSNR results to the MDKF across the board. However, it provides significant improvements over the MDKF in terms of perceived quality and intelligibility when applied on speech corrupted by white noise for a large range of input SNR. The performance gains are much more significant for input signals at much lower SNR. Overall, the BMMDKF has proven to be theoretically preferable to the MDKF in tracking and recovering the underlying clean speech in a noisy input over a range of input noise levels.

With separate PDFs for the signal-dominant and noise-dominant portions of the mask, a condition needs to be checked for each iteration to pull the correct mask statistics. This check, along with the modification of the observation mean and variance, slows the algorithm down by a significant amount. Currently, the enhancer is not being used in a real-time scenario, and thus speed is not crucial. If it is used in a real-time context in the future, however, the code will need to be modified so as to minimise the time delay of the enhancement.

Chapter 6

Improved LPC Coefficients

6.1 Introduction

The framework for the estimation of the linear prediction coefficients (LPCs) used in the MDKF was described earlier in Section 2.5. Since the LPCs aim to mimic the linear prediction model of speech, the accuracy of LPC estimation is critical to the performance of the enhancer. In this section, we proposed a simple modification to the LPC estimation by incorporating information from an available IBM, with the aim of improving the coefficients generated.

6.2 Weighted LPC estimation

As described in Section 2.5, LPC estimation using the autocorrelation method minimises the mean squared error by minimising the total prediction error E over all samples.

Instead of a simple summation of all errors, we can instead have a weighted sum that better represents the speech; this information can be provided by a binary mask first applied to the speech. We can then modify Equation 2.4 to get:

$$E = \sum_{n} w(n)e^2(n) = \sum_{n} w(n)[s(n) + \sum_{k=1}^{p} a_k s(n-k)]^2 \tag{6.1}$$

where $w(n)$ represents the weight attached to each speech sample $s(n)$.

This modification essentially represents a multiplication of the autocorrelation function by a scaling factor dependent on an IBM. In this algorithm, unlike the BMMDKF, the binary mask is specifically applied to the input of interest only. In the modulation-domain LPC estimation used in the MDKF, the LPCs are estimated from each modulating signal along with the corresponding binary mask frame, where the modulating signal and the binary mask are windowed in the same way. The windowed mask has values from 0 to real numbers less than 1, due to the windowing.

The weights tested were of the form:

$$w(n) = \begin{cases} 1 + th, & \text{if } IBM(n) == 0 \\ 1 - th, & \text{otherwise} \end{cases} \tag{6.2}$$

where th represents a positive threshold value. The optimal threshold was found to be 0.15 i.e. the optimal weights were determined to be $1 + 0.15 = 1.15$ when the mask is zero and $1 - 0.15 = 0.85$ when the mask shows a non-zero value. For example, a mask frame of $[0, 0, 0.48, 0]$ gives the weight vector $[1.15, 1.15, 0.85, 1.15]$, which is then multiplied element-wise with the input modulating signal frame. All other steps in the LPC estimation are the same as in Section 2.5.

It should be noted that these values imply that when the mask indicates noise, the error is given a greater weight as compared to when the mask indicates speech. A possible explanation is that generally, the amplitude of the signal during speech activity is greater than that of speech absence, assuming that the noise is random but always present in the signal. As such, if the mask indicates that a certain time-frequency (T-F) unit contains only noise, the LPC modelled signal should also have noise at the same T-F unit to minimise the overall mean squared error, and likewise for when the mask indicates speech presence. However, speech can generally have a larger variation in amplitude as compared to that of noise only. As such, it can be argued that if the model needs to predict speech absence (i.e. speech absence), it has to do so with greater accuracy than if it needs to predict speech presence. This leads to the need for a greater weight when the mask indicates speech absence as compared to speech presence. Results agree with this weighting, showing some improvements over the original MDKF.

6.3 Performance Results and Discussion

In this section, the enhanced-LPC algorithm will be termed LMDKF, with the other algorithms named as before. The block diagram for the LMDKF is shown in Figure 6.1.

Similarly to previous experiments, the LMDKF is run against other control algorithms, with the parameters used shown in Table 6.1.

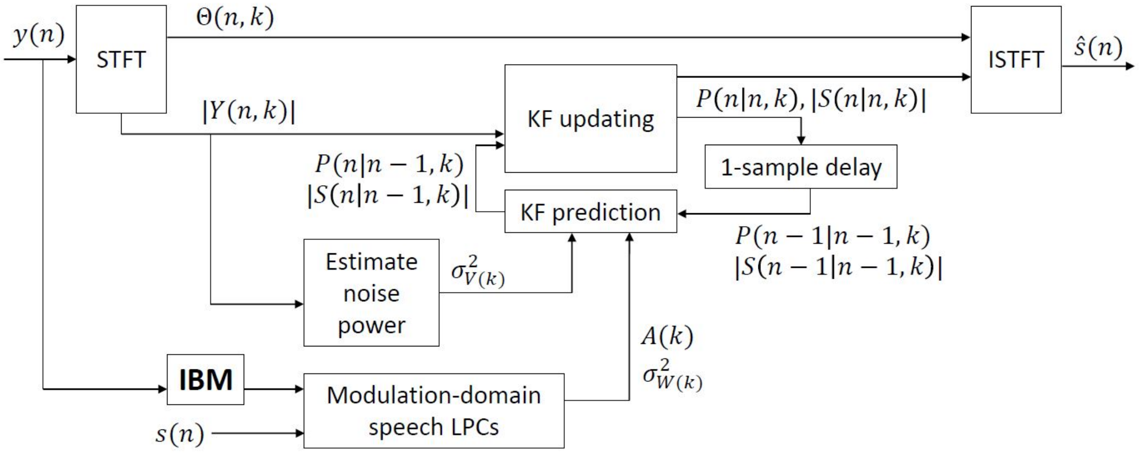

Figure 6.1: Block diagram of MDKF using IBM-enhanced LPCs (LMDKF)

Parameter	Value
Sampling frequency	16 kHz
Acoustic frame length	16 ms
Acoustic frame shift	4 ms
Modulation frame length	24 ms
Modulation frame shift	4 ms
Windowing function	Hamming window
MDKF model order	2
LPCs generated from	clean speech
Input speech corrupted by	white noise
IBM SNR threshold (LC)	0 dB
LPC weight threshold	0.15

Table 6.1: List of parameters used to evaluate LMDKF and other algorithms

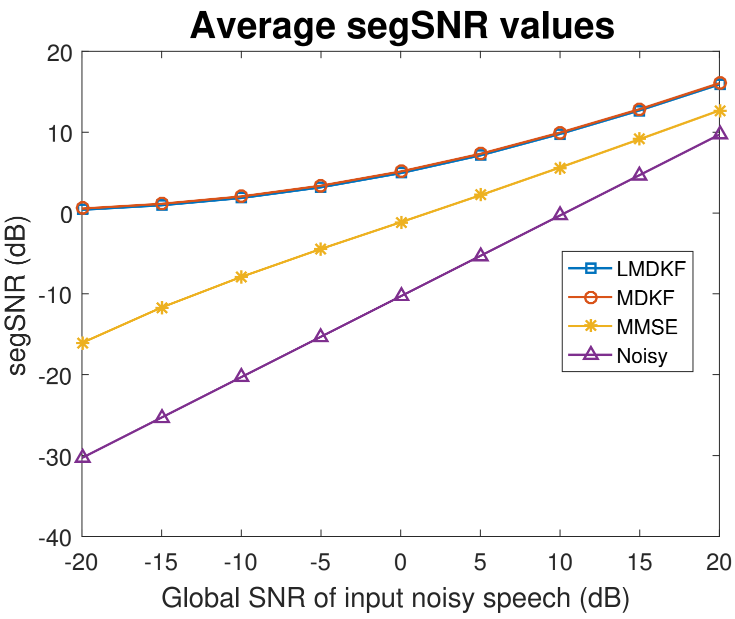

Figure 6.2: Average segSNR values of LMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels

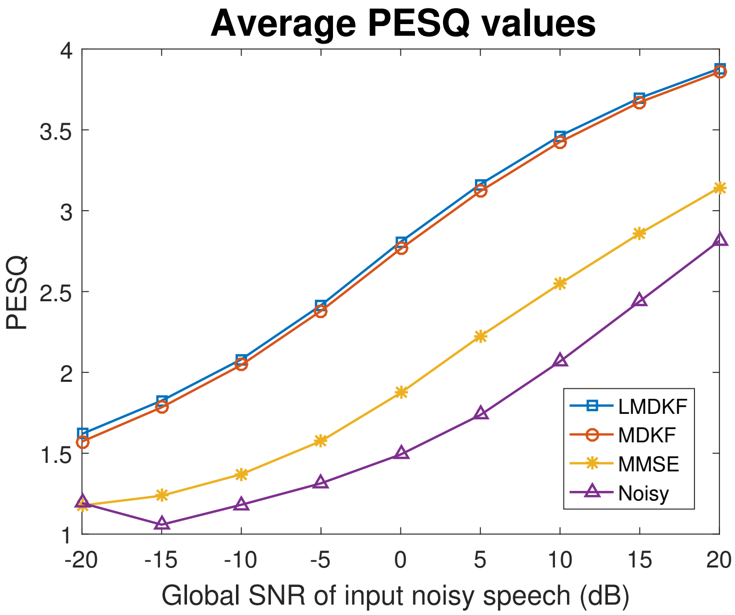

Figure 6.3: Average PESQ values of LMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels

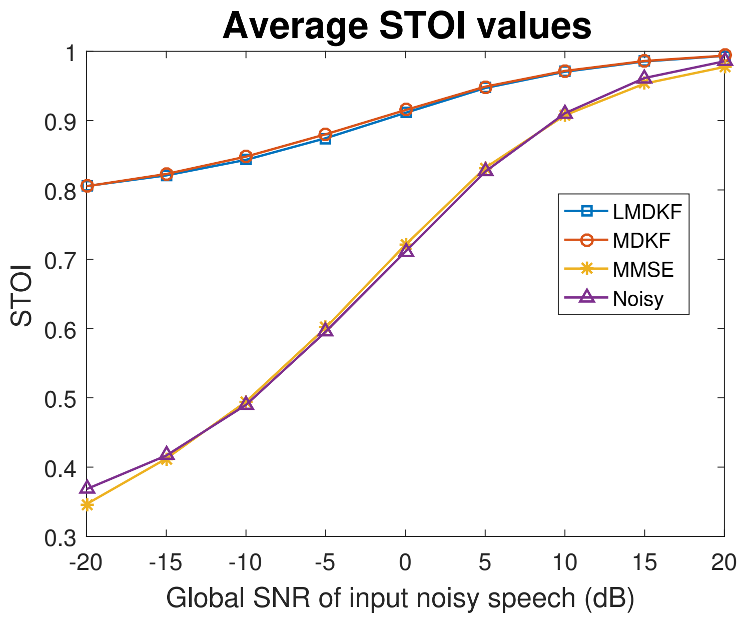

Figure 6.4: Average STOI values of LMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels

Figure 6.2 shows that the segSNR performance of the LMDKF and MDKF are very similar. In fact, the LMDKF performs very slightly worse, with segSNR scores an average of −0.1797 dB relative to the MDKF.

The STOI scores of LMDKF are also generally lower than MDKF (Figure 6.4). This decrease is small, but peaks out at inputs between −10 and 0 dB, with an average reduction of 0.5418%. Overall, the average STOI scores of LMDKF are lower by 0.2514%.

Across the board, Figure 6.3 demonstrates that the PESQ scores of the LMDKF are higher than MDKF by an average of 1.5349%. For 0 to 20 dB input SNR, the LMDKF shows PESQ scores 1.06% better than MDKF, while the improvement rises to 2.01% for −20 to 0 dB input. As with the BMMDKF, the LMDKF performs shows greater performance gains over the MDKF for a lower input SNR, with an improvement of 3.0224% at −20 dB input and 0.5585% at 20 dB.

On the surface, it appears that the changes to segSNR, PESQ and STOI balance out overall. However, it is arguable that segSNR is relatively unimportant compared to the other two measures when it comes to human listeners and their subjective opinions. Furthermore, the largest decrease in STOI score compared to the MDKF is 0.0092, which is a 1% decrease in intelligibility scores i.e. MDKF is estimated to accurately detect 1% more words than LMDKF, which can be regarded as almost negligible. On the other hand, while the improvement in PESQ score is not huge, it is still significant, and possibly more important to a real listener than a 1% reduction in words recognised. Overall, if we look past the raw numbers and consider what they constitute, the LMDKF is very similar to the MDKF, but it is very possible that the LMDKF is regarded slightly more favourably than the original MDKF in real listening tests.

6.4 Conclusion

This chapter discusses modifying the LPC estimation part of the MDKF by using a weighted sum to compute the mean squared error, using an IBM applied on the input signal to determine these weights. Unlike the BMMDKF of Chapter 5, the IBM here is specifically applied to the signal of interest, to determine the time-frequency (T-F) areas of the signal to modify in order to come up with useful weighting.

This modification shows very minor degradation in segSNR and STOI scores, and a small but moderately significant improvement in PESQ scores. Overall, the LMDKF is very similar to the MDKF. However, the various numerical scores were analysed further, and it is conceivable that the LMDKF would be preferred over the MDKF.

Chapter 7

IBM-improved Noise Estimation

7.1 Introduction

Noise estimation is a critical part of speech enhancement, and the performance of enhancement algorithms is heavily affected by the accuracy of the noise estimation. In this chapter, a modification of an existing noise spectral estimation method based on an ideal binary mask (IBM) is proposed.

A popular method to estimate the noise power spectral density (PSD) is to use a minimum mean-squared-error (MMSE) optimal estimation method, which can be interpreted as a VAD-based noise power estimator. This chapter proposes a few tweaks to the parameters used in the algorithm described in [19], based on the information provided by an IBM.

7.2 MMSE Noise Estimation

To aid the understanding of the modifications made in this chapter, we first present a summary of the MMSE noise estimator in [19]. A majority of the equations are left out for clarity, and only those critical to the proposed modification will be inserted in this report.

Let y, s and n be the noisy speech, clean speech and noise respectively, and assume that they are additive in the short-time Fourier domain, giving the noisy observed speech as $Y = S + N$, where time and frequency indices are omitted for convenience and S, N are the complex spectral speech and noise components respectively. The clean speech and noise are assumed independent, such that $\mathbb{E}(|Y|^2) = \mathbb{E}(|S|^2) + \mathbb{E}(|N|^2)$. Define the spectral speech and noise power as $\mathbb{E}(|S|^2) = \sigma_S^2$ and $\mathbb{E}(|N|^2) = \sigma_N^2$ respectively, and the *a priori* and *a posteriori* Signal-to-Noise Ratio (SNR) by $\zeta = \sigma_S^2/\sigma_N^2$ and $\gamma = |y|^2/\sigma_N^2$ respectively.

It is assumed that the noise and speech spectral coefficients ($p_N(n)$ and $p_S(s)$ respectively) have complex Gaussian distributions. This gives a complex Gaussian distribution for the noisy speech $p_Y(y)$ which depends on the true SNR. The noise power estimator shown in [18] is based on an

MMSE estimate of the noise periodogram, which can be obtained by calculating the conditional expectation $\mathbb{E}(|N|^2|y)$ (equation omitted for clarity), which is a function of the power of the noisy observation, the *a priori* SNR and the spectral noise power.

In practice, the *a priori* SNR and the spectral noise power have to be estimated. When estimating noise power, it is a common assumption that the noise signal varies more slowly than speech [67]. Therefore, [18] uses the spectral noise power estimate of the previous time frame $(l-1)$ i.e. $\hat{\sigma}_N^2 = \hat{\sigma}_N^2(l-1)$, assuming some correlation between the noise in adjacent frames of speech.

Estimating the *a priori* SNR is more complicated, as speech tends to vary more rapidly between successive frames. In [18], the proposed method to estimate $\hat{\zeta}$ uses a maximum-likelihood (ML) estimate followed by bias compensation. When doing so, the MMSE estimator can be viewed as a hard-threshold voice activity detector (VAD) based decision between the noisy observation and the spectral noise power estimate. The resultant estimator is biased and requires bias compensation.

Finally, after estimating the noise periodogram from $\mathbb{E}(|N|^2|y)$, the noise PSD is obtained via recursive smoothing with a parameter $\alpha_{pow} = 0.8$ [18]:

$$\hat{\sigma}_N^2(l) = \alpha_{pow}\hat{\sigma}_N^2(l-1) + (1-\alpha_{pow})\mathbb{E}(|N|^2|y(l)) \tag{7.1}$$

7.2.1 Assumptions in Unbiased MMSE Estimator

The previous section dealt with the original noise power MMSE estimator as described in [18]. A modification was made in [19] to produce an unbiased estimator based on a soft speech presence probability (SPP) estimate with fixed priors, instead of a hard VAD threshold. This section covers part of the modification, and in particular highlights certain assumptions made that can be improved using IBM information.

When speech presence is uncertain, an MMSE estimator for the noise periodogram is given by:

$$\mathbb{E}(|N|^2|y) = P(H_0|y)\mathbb{E}(|N|^2|y, H_0) + P(H_1|y)\mathbb{E}(|N|^2|y, H_1) \tag{7.2}$$

where H_0 and H_1 indicate speech absence and speech presence respectively. Using Bayes' Theorem, the *a posteriori* SPP can be expressed as:

$$P(H_1|y) = \frac{P(H_1)p_{Y|H1}(y)}{P(H_0)p_{Y|H0}(y) + P(H_1)p_{Y|H1}(y)} \tag{7.3}$$

Thus, computing the *a posteriori* SPP requires the *a priori* probabilities $P(H_1) = 1 - P(H_0)$ and the likelihood functions for speech presence $p_{Y|H1}(y)$ and speech absence $p_{Y|H0}(y)$. Without an observation, [19] assumes that a time-frequency point being considered is equally likely to contain speech or not contain speech i.e. uniform priors $P(H_1) = P(H_0) = 0.5$ were chosen, independent of the observation. Given an IBM, this can be improved.

The likelihood functions $p_{Y|H1}(y)$ and $p_{Y|H0}(y)$ in Equation 7.3 indicate how well the observation y fits the modelling parameters for speech presence and absence respectively, and can be modelled with complex Gaussian distributions (equations omitted). The likelihood under speech presence depends on the *a priori* SNR. While the algorithm in [18] uses a complex Gaussian distribution for the noisy observation $p_Y(y)$ which depends on the true local *a priori* SNR, the likelihood under speech presence is instead a function of a parameterised *a priori* SNR, reflecting the typical SNR when speech is present. In [19], this typical SNR is fixed at 15 dB. This is another area in which information from an IBM can be useful.

Substituting the likelihood functions for speech absence and speech presence (equations omitted in this report) into Equation 7.3, an expression for the *a posteriori* SPP can be obtained as:

$$P(H_1|y) = \left(1 + \frac{P(H_0)}{P(H_1)}(1+\zeta_{H_1})e^{-\frac{|y|^2}{\hat{\sigma}_N^2}\frac{\zeta_{H_1}}{1+\zeta_{H_1}}}\right)^{-1} \tag{7.4}$$

where the spectral noise power estimate of the previous time frame is used i.e. $\hat{\sigma}_N^2 = \hat{\sigma}_N^2(l-1)$ as before.

If the noise power estimate $\hat{\sigma}_N^2$ underestimates the true noise power σ_N^2, the *a posteriori* SPP in Equation 7.4 will be overestimated, and the noise power will not be tracked as quickly as needed. In the worst case, the noise power might remain the same. To avoid this stagnation due to underestimated noise power, a check is done in [19] to verify if the *a posteriori* SPP has been close to 1 for a long time. The *a posteriori* SPP is first recursively smoothed over time: $\bar{P}(l) = 0.9\bar{P}(l-1) + 0.1P(H_1|y(l))$. If this smoothed quantity is larger than 0.99, the update is deemed to have stagnated; the current *a posteriori* SPP estimate $P(H_1|y(l))$ is then forced to be lower than 0.99.

The noise power spectrum can then be computed from the estimated *a posteriori* SPP $P(H_1|y(l))$, which is used to weight the noisy input power spectrum $|Y|^2$ and the previous estimate of the noise power $\hat{\sigma}_N^2(l-1)$ accordingly:

$$\mathbb{E}(|N|^2|y(l)) = P(H_1|y(l))\hat{\sigma}_N^2(l-1) + [1 - P(H_1|y(l))]|Y(l)|^2 \tag{7.5}$$

Finally, this raw noise estimate is smoothed as in Equation 7.1 to obtain the estimated noise PSD.

7.3 Modifying Noise Estimation

In the previous section which briefly reviewed the MMSE noise estimators of [18] and [19], some parameters are notably fixed. In this section, we use an IBM to provide information that allows us to vary these parameters.

Firstly, a typical speech-present SNR of 15 dB is used when estimating the noise power for each frame. This provides an overall optimal result, but can be improved if a binary mask can indicate speech absence or presence. This SNR was therefore adaptively modified to be increased or decreased depending on what is indicated by the IBM for each frame. The IBM frame was not used directly, but mapped to a set of weights determined by the formula:

$$w(n) = \begin{cases} th, & \text{if } IBM(n) == 0 \\ 1.5 + th, & \text{otherwise} \end{cases} \tag{7.6}$$

where th is an empirically-determined positive constant. Notice that if the mask indicates speech presence, the corresponding weight is larger than 1, essentially boosting the speech. On the whole, this was found to produce better results.

The speech-present SNR was then multiplied by the mean of the current IBM frame, and this modified SNR was used for the current frame to calculate the *a posteriori* SPP.

However, this alone was found to be insufficient in improving the noise estimation. Thus, the IBM frame was additionally used to weight the *a posteriori* SPP found in Equation 7.4. Unlike the adjustment to the SNR, the SPP frame, which is used to obtain the noise estimate, was multiplied element-wise by the weighting IBM with values as determined in Equation 7.7. Combining these two adjustments showed the best results.

7.4 Optimal IBM-Modified Estimator Threshold

Given the modification, an optimal threshold for the binary mask needed to be found. A few threshold values were tested and compared in the plots below, using a 0 dB LC threshold IBM. The following plots compare the ratio of an MDKF using the modified noise estimation and the original MDKF, plotted over a range of input SNR values and comparing segSNR, PESQ and STOI: a ratio of > 1 means that the IBM-modified noise estimation performs better than the original.

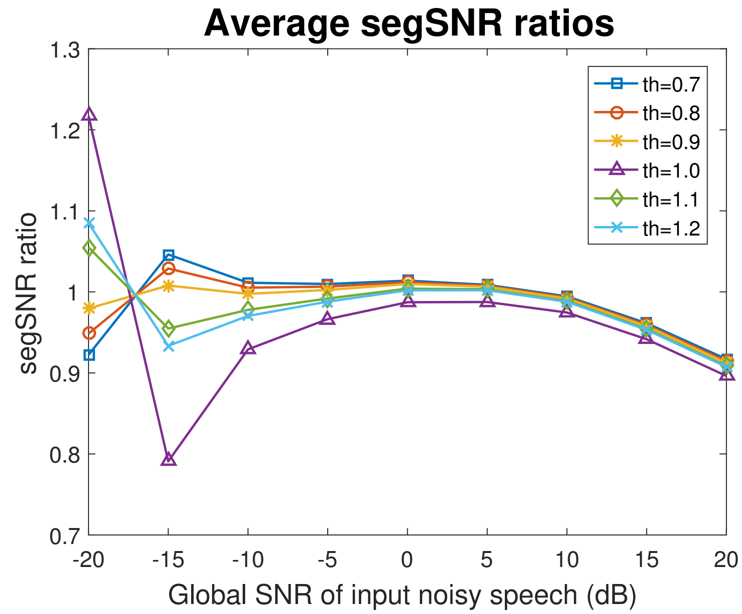

Figure 7.1: Average segSNR ratios comparing IBM-modified noise estimation with original MMSE noise estimation vs. speech corrupted by white noise at varying SNR levels

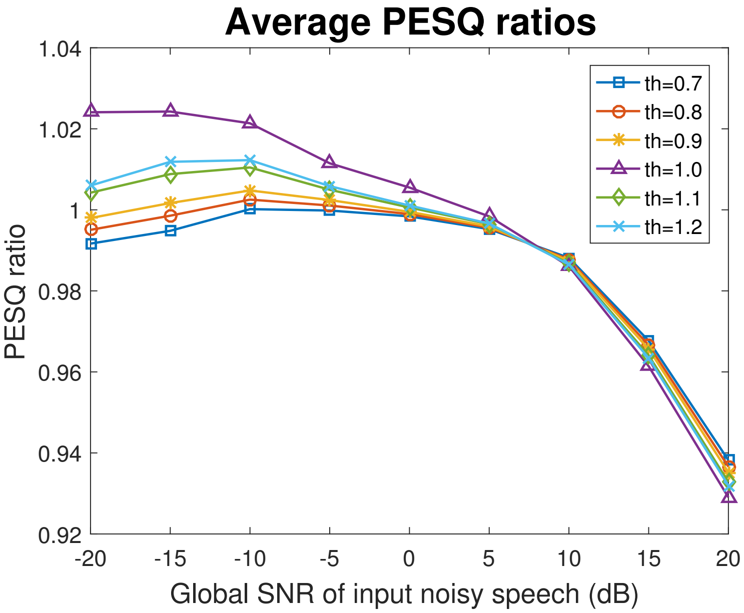

Figure 7.2: Average PESQ ratios comparing IBM-modified noise estimation with original MMSE noise estimation vs. speech corrupted by white noise at varying SNR levels

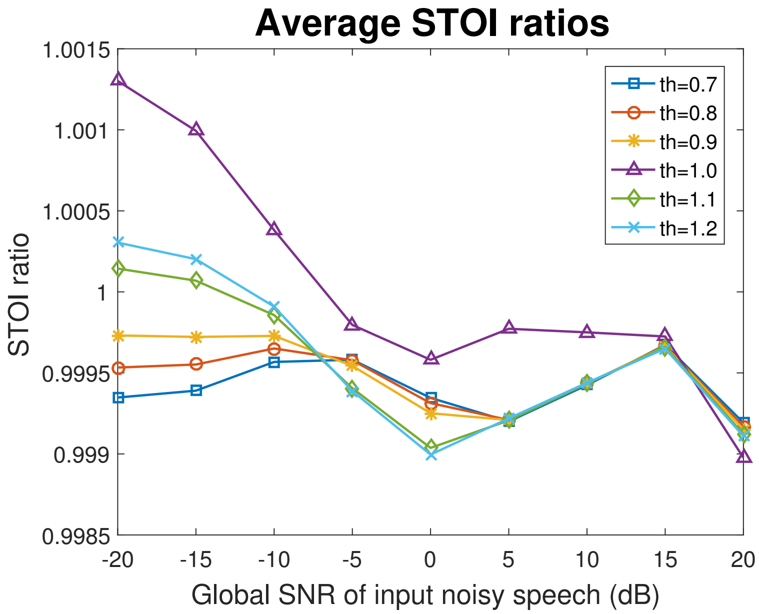

Figure 7.3: Average STOI ratios comparing IBM-modified noise estimation with original MMSE noise estimation vs. speech corrupted by white noise at varying SNR levels

Figures 7.1 to 7.3 show that $th = 1.0$ is the best value. Although it has the worst-performing segSNR over most of the input SNR range tested, it performs best for PESQ and STOI over most of the input SNR range, and these measures are arguably more representative of how a human listener perceives the enhanced speech as compared to an SNR measure such as segSNR. Therefore, $th = 1.0$ was used as the threshold, producing the IBM-mapped weights:

$$w(n) = \begin{cases} 1.0, & \text{if } IBM(n) == 0 \\ 2.5, & \text{otherwise} \end{cases} \tag{7.7}$$

7.4.1 Estimated Global SNR

In Figure 7.1, we saw that the segSNR of the optimal-threshold ($th = 1.0$) IBM-modified noise estimator is largely lower than that of the original MMSE noise estimator of [19]. This is verified in Figure 7.4.

Figure 7.4 compares the averaged estimated global SNR of the IBM-modified noise estimation algorithm and the original MMSE noise estimator, plotted against the actual input global SNR. In the plot, the true 45° line is represented by a green dotted line i.e. the closer to the green line, the more accurate the SNR estimation. The plot shows that both algorithms tend to underestimate the noise level (overestimate SNR) when the noise is very large (-15 dB input SNR or lower), and vice versa when the noise is small (high input SNR). However, an across-the-board feature is that

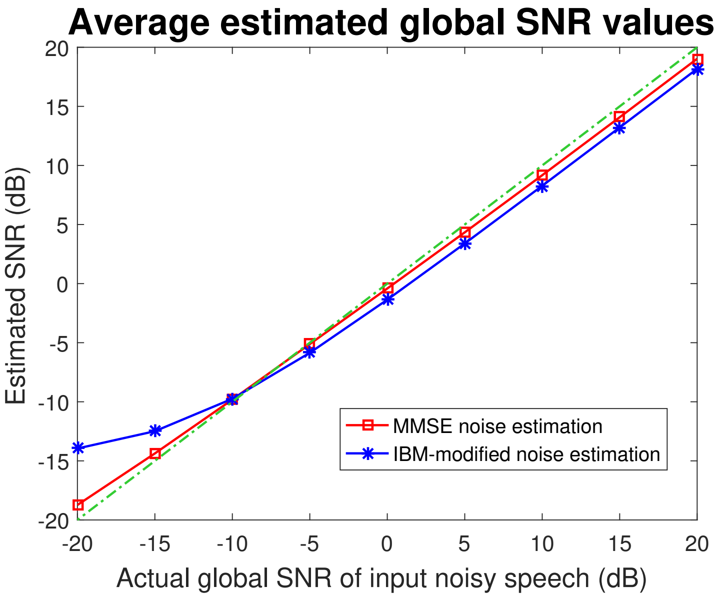

Figure 7.4: Estimated global SNR of different noise estimation methods vs. actual global SNR of input speech

the original MMSE noise estimator largely matches the real value better than the IBM-modified noise estimator.

A possible explanation for this is that the noise is random, and the MMSE noise estimator in [19] uses only a fixed *a priori* SNR value typical of speech. However, incorporating IBM data meant tweaking the original spectral components, which meant modifying this SNR value adaptively. Using the threshold values determined in Figures 7.2 and 7.3 better represented the actual speech information, but might not represent the numerical SNR as well as the original algorithm.

7.5 Performance Results and Discussion

In this section, the algorithm using the IBM-enhanced noise estimate will be termed NMDKF, with the other algorithms named as before. The block diagram for the proposed modification is shown in Figure 7.5.

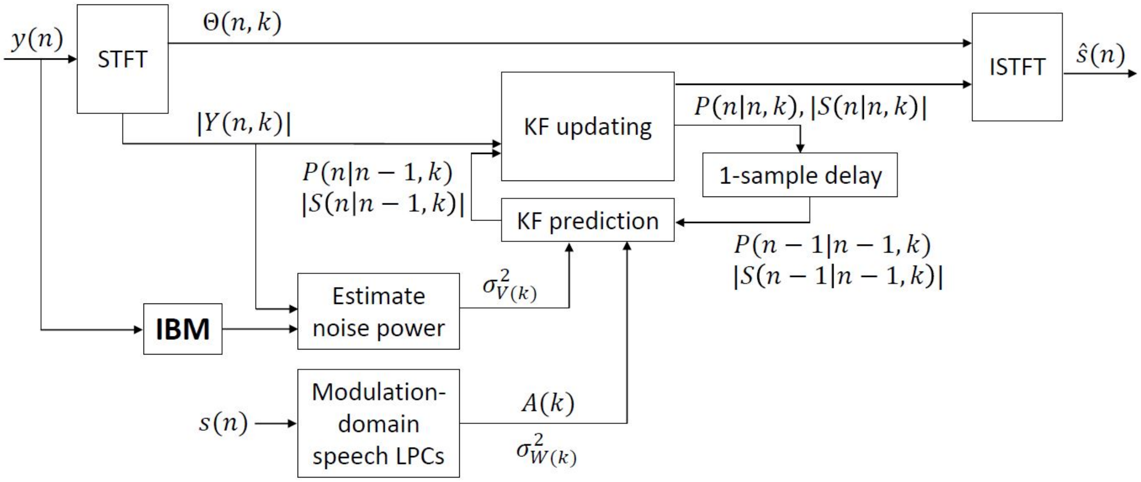

Figure 7.5: Block diagram of MDKF using IBM-modified noise estimation (NMDKF)

As with all previous experiments, the algorithm discussed in this section is run against other control algorithms, with the parameters used shown in Table 7.1.

Parameter	**Value**
Sampling frequency	16 kHz
Acoustic frame length	16 ms
Acoustic frame shift	4 ms
Modulation frame length	24 ms
Modulation frame shift	4 ms
Windowing function	Hamming window
MDKF model order	2
LPCs generated from	clean speech
Input speech corrupted by	white noise
IBM SNR threshold (LC)	0 dB
IBM zeros threshold	1.0
IBM ones threshold	2.5

Table 7.1: List of parameters used to evaluate NMDKF and other algorithms

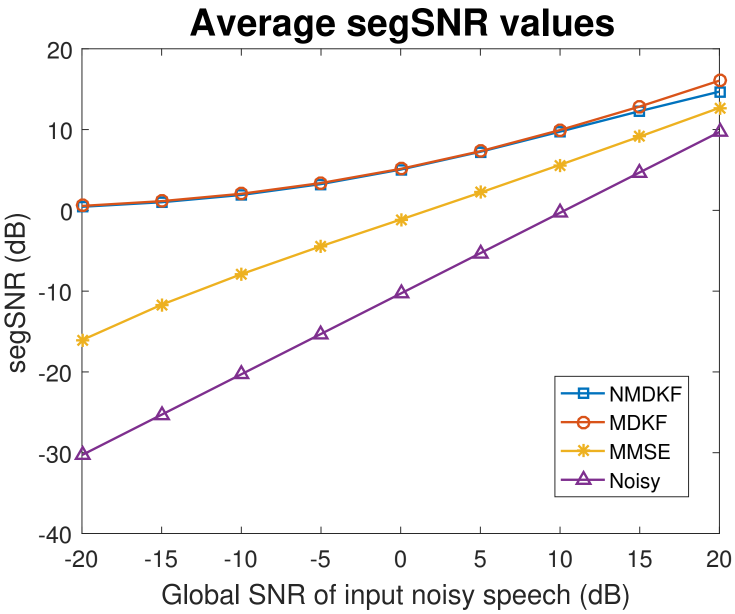

Figure 7.6: Average segSNR values of NMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels

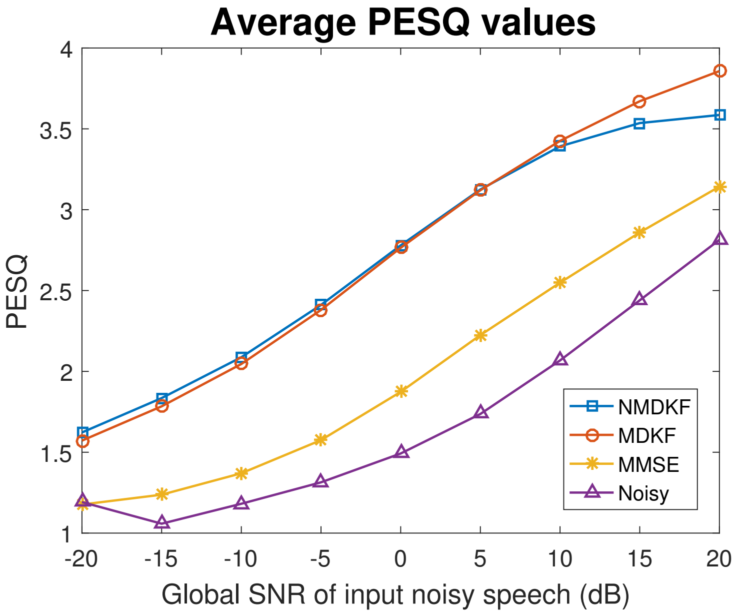

Figure 7.7: Average PESQ values of NMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels

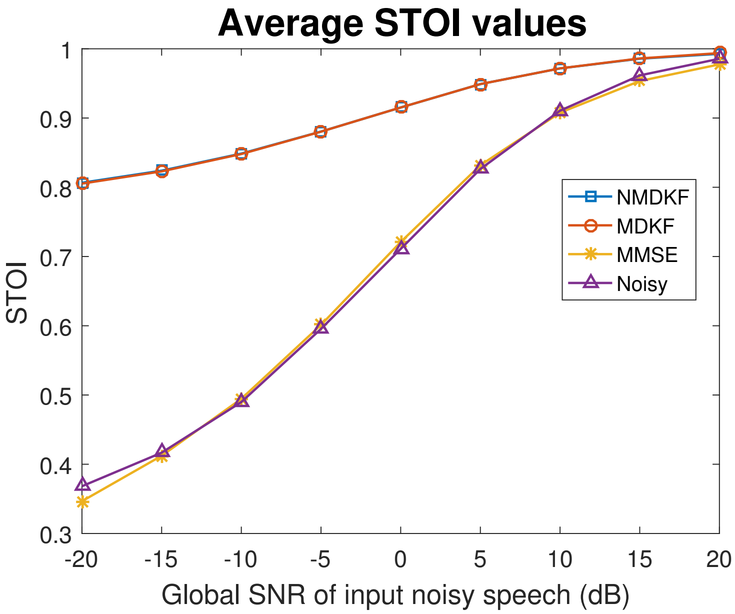

Figure 7.8: Average STOI values of NMDKF and other algorithms vs. speech corrupted by white noise at varying SNR levels

The NMDKF performs very similar to the MDKF in terms of segSNR, as shown by Figure 7.6. Up to and including 10 dB input, its segSNR is marginally worse than MDKF by 0.1244 dB on average. From 10 to 20 dB input, the segSNR of NMDKF is lower by 0.7076 dB on average.

In terms of PESQ scores, the NMDKF shows improvements over the input SNR range of −20 to 5 dB compared to the MDKF by an average of 0.0323 or 1.68% (Figure 7.7). This is similar to the LMDKF. However, above 5 dB input, the NMDKF PESQ scores begin to peak out, unlike the MDKF which continues to steadily rise. From 10 to 20 dB input, the average NMDKF PESQ scores are 0.1468 lower than the MDKF, with the discrepancy at −0.2714 for 20 dB input.

From Figure 7.8, the STOI scores of NMDKF and MDKF are very similar: the largest difference is at −20 dB input, where the NMDKF score is 0.0012 higher than MDKF. Overall, the discrepancies are minor, but the general trend is that the NDMKF performs slightly better at low input SNR (−20 to 0 dB), while it performs similarly or marginally worse at higher input SNR (0 to 20 dB).

For the most part, the NMDKF performs similarly or shows small improvements over the MDKF at low input SNR, while it performs similarly or worse at higher input SNR. This can lead to the conclusion that the NMDKF would not be preferred over the MDKF. However, consider that at high input SNR, the MDKF already posts very high scores by all measures. While the PESQ score of NMDKF is lower by MDKF by 0.2714 at 20 dB input, the scores are 3.5857 and 3.8572 for the NMDKF and MDKF respectively. On the PESQ scale, these are fairly high scores, and while the reduction in PESQ admittedly hurts quality and user experience, it is still a generally satisfactory score. Furthermore, the NMDKF PESQ scores are better where it is arguably more important to improve the scores: at lower input SNR. As mentioned previously, the NMDKF shows improvements from −20 to 5 dB input SNR as compared to the MDKF by an average of 0.0323. Informal listening tests showed a general preference for the NMDKF at these lower input SNR, while the difference at higher input SNR was significantly less perceptible.

7.6 Conclusion

This chapter proposes a modification to the MMSE noise estimation method done in [19], by incorporating information from an IBM to tweak certain fixed parameters such as SPP and *a priori* SNR. By and large, performance results show that the NMDKF performs better than the MDKF at lower input SNR and vice versa at higher input SNR. Overall, the modified algorithm performs similarly to the MDKF.

However, the NMDKF generally performs better than the original MDKF up to an input signal SNR of approximately 5 dB. It is possible that an adaptive algorithm could be used to apply the noise estimation that performs better depending on the input noise level. Doing so will take extra time, and thus this selective framework could be used if speed was not of concern e.g. this might not be suitable for a real-time processing application.

Chapter 8

Conclusion and Future Work

In this project, we explored three approaches to modifying a speech enhancement method based on a modulation-domain Kalman filter (MDKF), by using information extracted from an ideal binary mask (IBM). The aim was to improve the perceived quality and intelligibility of the enhanced noisy speech. The first approach is to include IBM information into the parameters of the MDKF iteration equations directly, covered in Chapter 5. The next method, proposed in Chapter 6, proposes to modify estimation of the linear prediction coefficients (LPCs) by using a weighted sum to calculate the total error. In Chapter 7, the final method uses the IBM to tweak noise estimation parameters.

In Chapter 5, the IBM is applied to a set of training input samples to obtain separate averaged probability distribution statistics corresponding to speech-dominant and noise-dominant regions of the mask. This information is used to modify the observation mean and variance of an input noisy test signal to better track the underlying clean speech (BMMDKF method). Performance results show that the BMMDKF shows very similar segSNR results to the MDKF. However, it demonstrates significant improvements over the MDKF in terms of both perceived quality (PESQ) and intelligibility (STOI) when applied on noisy speech for a large range of input SNR, with larger performance gains at lower input SNR. Overall, the BMMDKF has proven to be preferable to the MDKF in enhancing noisy speech for a large range of input SNR.

The next modification, discussed in Chapter 6, is concerned with tweaking the LPC estimation, which is used to obtain LPC coefficients used in the MDKF filter equations. The original LPC estimation aims to minimise the sum of errors; this method (LMDKF method) suggests to instead minimise a weighted sum of errors, with the weights determined from an IBM applied directly to the test signal. Unlike the method in Chapter 5, the IBM here is applied to the specific input test signal only. The LMDKF shows very minor degradation in segSNR and STOI scores compared to the MDKF, and a small but moderately significant improvement in PESQ scores. Overall, it is similar to the MDKF, but when the various numerical scores were analysed further, it is conceivable that the LMDKF would be preferred over the MDKF due to the importance of the PESQ improvement.

Finally, Chapter 7 investigates a method (NMDKF method) to improve the MMSE noise estimation of [19], which is used to determine the error covariance matrix in the MDKF equations. The modification incorporates information from an IBM to tweak certain fixed parameters such as SPP and *a priori* SNR, with the aim of calculating the actual noise in each frame more accurately. On the whole, the modified algorithm performs similarly to the MDKF. However, the NMDKF shows improvements over the MDKF up to an input SNR of 5 dB. An adaptive algorithm could be used to apply the better-performing noise estimation method depending on the input noise level. Doing so will take more time, and so this may not be acceptable for a real-time speech enhancer.

8.1 Future Work

8.1.1 LPC Estimation

In the “oracle” baseline model used for all modifications, shown in Figure 3.4, LPC coefficients are calculated from the clean speech, as this project is more concerned with investigating the theoretical upper bound performance. In a real-world context, the clean speech is not available. A possible area for further study is to instead estimate the LPCs from the noisy speech passed through the MMSE speech enhancement algorithm used in [66]. Although not as accurate as clean-speech LPCs, the effect of noise on the model can be greatly reduced when calculating LPCs from MMSE-enhanced speech rather than from the original noisy speech. If the results are favourable, this can be used for practical applications as it does not involve clean speech.

8.1.2 Real Listening Tests

In this project, PESQ and STOI routines were used to try and mimic the perceived quality and intelligibility measures of human listeners. While these routines have been proven to be highly correlated with the subjective scores, there is still merit to holding listening tests with real human listeners, to provide a more accurate performance measures and also allow for individual feedback.

8.1.3 Probability Distribution

In Chapter 5, the BMMDKF algorithm assumes that the observation, prediction and IBM statistics are all Gaussian distributed. This simplifies the computation when combining these three parameters, and is a fairly accurate assumption. However, if allowed for additional computational complexity and algorithm runtime, it might be possible to investigate further the properties of these parameters, and find a better matching distribution. It is possible that these provide better performance gains that the BMMDKF, but at the end of the day the trade-offs between performance gains and algorithm complexity must be weighted and balanced.

8.1.4 Combining Modifications

This report discusses a few modifications to a baseline MDKF that provide varying degrees of improvements. It is possible that combining two or more of them could produce results better than any individual modification could achieve, and could be further investigated.

Bibliography

[1] Philipos C. Loizou. *Speech Enhancement: Theory and Practice*. CRC Press, 2007.

[2] A. R. Fukane and S. L. Sahare. “Different approaches of spectral subtraction method for enhancing the speech signal in noisy environments”. In: *International Journal of Scientific & Engineering Research* 2.5 (2011), p. 1.

[3] ITU-T P.862. *Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs.* 2001.

[4] C. H. Taal et al. “A short-time objective intelligibility measure for time-frequency weighted noisy speech”. In: *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. Mar. 2010, pp. 4214–4217.

[5] Mike Brookes. *VOICEBOX: Speech Processing Toolbox for MATLAB*. 2012. URL: `http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html`.

[6] J. Benesty, S. Makino, and J. Chen (Eds.) *Speech Enhancement.* Springer, 2005.

[7] J. Benesty, M. M. Sondhi, and Y. Huang (ed). *Springer Handbook of Speech Processing.* Springer, 2007.

[8] B. Widrow et al. “Adaptive noise cancelling: Principles and applications”. In: *Proceedings of the IEEE* 63.12 (Dec. 1975), pp. 1692–1716.

[9] B. Widrow and M. E. Hoff. “Neurocomputing: Foundations of Research”. In: ed. by James A. Anderson and Edward Rosenfeld. Cambridge, MA, USA: MIT Press, 1988. Chap. Adaptive Switching Circuits, pp. 123–134.

[10] J. Dhiman, S. Ahmad, and K. Gulia. “Comparison Between Adaptive Filter Algorithms (LMS, NLMS and RLS)”. In: *International Journal of Science, Engineering and Technology Research (IJSETR)* 2 (May 2013).

[11] Leon Cohen. *Time Frequency Analysis: Theory and Applications.* Prentice-Hall, 1994.

[12] Michael D. Riley. *Speech Time-Frequency Representations.* Springer, 1989.

[13] D. Griffin and Jae Lim. “Signal estimation from modified short-time Fourier transform”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32.2 (Apr. 1984), pp. 236–243.

[14] O. Cappe. “Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor”. In: *IEEE Transactions on Speech and Audio Processing* 2.2 (Apr. 1994), pp. 345–349.

[15] Hynek Hermansky. *Modulation Spectrum in Speech Processing.* Springer, 1998.

[16] Jongseo Sohn, Nam Soo Kim, and Wonyong Sung. “A statistical model-based voice activity detection”. In: *IEEE Signal Processing Letters* 6.1 (Jan. 1999), pp. 1–3.

[17] R. Martin. “Noise power spectral density estimation based on optimal smoothing and minimum statistics”. In: *IEEE Transactions on Speech and Audio Processing* 9.5 (July 2001), pp. 504–512.

[18] R. C. Hendriks, R. Heusdens, and J. Jensen. “MMSE based noise PSD tracking with low complexity”. In: *2010 IEEE International Conference on Acoustics, Speech and Signal Processing.* Mar. 2010, pp. 4266–4269.

[19] T. Gerkmann and R. C. Hendriks. “Unbiased MMSE-Based Noise Power Estimation With Low Complexity and Low Tracking Delay”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.4 (May 2012), pp. 1383–1393.

[20] S. Boll. “Suppression of acoustic noise in speech using spectral subtraction”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 27.2 (Apr. 1979), pp. 113–120.

[21] Saeed V. Vaseghi. *Advanced Digital Signal Processing and Noise Reduction.* Wiley, 2009.

[22] N. Aydin and H. S. Markus. “Optimization of processing parameters for the analysis and detection of embolic signals”. In: *European Journal of ultrasound* 12.1 (2000), pp. 69–79.

[23] M. Berouti, R. Schwartz, and J. Makhoul. “Enhancement of speech corrupted by acoustic noise”. In: *ICASSP ’79. IEEE International Conference on Acoustics, Speech, and Signal Processing.* Vol. 4. Apr. 1979, pp. 208–211.

[24] H. Kozou et al. “The effect of different noise types on the speech and non-speech elicited mismatch negativity”. In: *Hearing research* 199.1 (2005), pp. 31–39.

[25] Albert S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound.* The MIT Press, Cambridge, MA, 1990.

[26] DeLiang Wang and Guy J. Brown. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications.* Wiley, 2006.

[27] DeLiang Wang. “On Ideal Binary Mask As the Computational Goal of Auditory Scene Analysis”. In: *Speech Separation by Humans and Machines.* Ed. by Pierre Divenyi. Boston, MA: Springer US, 2005, pp. 181–197.

[28] D. S. Brungart et al. “Isolating the energetic component of speech-on-speech masking with ideal time-frequency segregation”. In: *The Journal of the Acoustical Society of America* 120.6 (2006), pp. 4007–4018.

[29] J. S. Garofolo et al. *TIMIT Acoustic-Phonetic Continuous Speech Corpus.* 1993. URL: `https://catalog.ldc.upenn.edu/ldc93s1`.

[30] G. Hu and D. Wang. “Monaural speech segregation based on pitch tracking and amplitude modulation”. In: *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing.* Vol. 1. May 2002, pp. 553–556.

[31] D. P.W. Ellis. “Model-based scene analysis”. In: *Computational auditory scene analysis: Principles, algorithms, and applications* (2006), pp. 115–146.

[32] Y. Li and D. L. Wang. “On the optimality of ideal binary time–frequency masks”. In: *Speech Communication* 51.3 (2009), pp. 230–239.

[33] N. Li and P. C. Loizou. "Factors influencing intelligibility of ideal binary-masked speech: Implications for noise reduction". In: *The Journal of the Acoustical Society of America* 123.3 (2008), pp. 1673–1682.

[34] U. Kjems et al. "Speech intelligibility of ideal binary masked mixtures". In: *2010 18th European Signal Processing Conference*. Aug. 2010, pp. 1909–1913.

[35] M. C. Anzalone et al. "Determination of the potential benefit of time-frequency gain manipulation". In: *Ear and hearing* 27.5 (2006), pp. 480–492.

[36] U. Kjems et al. "Role of mask pattern in intelligibility of ideal binary-masked noisy speech". In: *Acoustical Society of America* 126 (3 Sept. 2009).

[37] S. G. Karadogan et al. "Robust isolated speech recognition using binary masks". In: *2010 18th European Signal Processing Conference*. Aug. 2010, pp. 1988–1992.

[38] T. Stokes, C. Hummersone, and T. Brookes. "Reducing Binary Masking Artifacts in Blind Audio Source Separation". In: *Audio Engineering Society Convention 134*. May 2013.

[39] J. Makhoul. "Linear prediction: A tutorial review". In: *Proceedings of the IEEE* 63.4 (Apr. 1975), pp. 561–580.

[40] B. Yegnanarayana et al. "Speech enhancement using linear prediction residual". In: *Speech Communication* 28.1 (1999), pp. 25–42.

[41] Tsutomu Chiba. *The Vowel: Its Nature and Structure*. Phonetic Society of Japan, 1958.

[42] Kenneth N. Stevens. *Acoustic Phonetics*. The MIT Press, 1999.

[43] Bruce Hayes. *Introductory Phonology*. Wiley-Blackwell, 2008.

[44] R. E. Kalman et al. "A new approach to linear filtering and prediction problems". In: *Journal of basic Engineering* 82.1 (1960), pp. 35–45.

[45] Wen-Rong Wu and Po-Cheng Chen. "Subband Kalman filtering for speech enhancement". In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 45.8 (Aug. 1998), pp. 1072–1083.

[46] Peter S. Maybeck. *Stochastic Models, Estimation, and Control*. Vol. 1. Academic press, Inc., 1979.

[47] K. Paliwal and A. Basu. "A speech enhancement method based on Kalman filtering". In: *ICASSP '87. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 12. Apr. 1987, pp. 177–180.

[48] N. Ma, M. Bouchard, and R. A. Goubran. "Speech enhancement using a masking threshold constrained Kalman filter and its heuristic implementations". In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.1 (Jan. 2006), pp. 19–32.

[49] J. D. Gibson, B. Koo, and S. D. Gray. "Filtering of colored noise for speech enhancement and coding". In: *IEEE Transactions on Signal Processing* 39.8 (Aug. 1991), pp. 1732–1742.

[50] L. Atlas and S. A. Shamma. "Joint Acoustic and Modulation Frequency". In: *EURASIP Journal on Advances in Signal Processing* (July 2003).

[51] K. Paliwal, B. Schwerin, and K. Wójcicki. "Single-channel speech enhancement using spectral subtraction in the short-time modulation domain". In: *Speech Communication* 52.5 (2010), pp. 450–475.

[52] S. So and K. K. Paliwal. “Modulation-domain Kalman filtering for single-channel speech enhancement”. In: *Speech Communication* 53.6 (2011), pp. 818–829.

[53] C. J. Li. “Non-Gaussian, Non-stationary, and Nonlinear Signal Processing Methods”. PhD thesis. Aalborg University, Denmark, 2006.

[54] S. Greenberg and T. Arai. “The relation between speech intelligibility and the complex modulation spectrum.” In: *2nd INTERSPEECH Event, Aalborg, Denmark*. 2001, pp. 473–476.

[55] T. Quatieri. *Discrete-Time Speech Signal Processing: Principles and Practice*. Prentice-Hall, 2002.

[56] ITU-T P.830. *Subjective performance evaluation of telephone band and wideband codecs*. 1996.

[57] ITU-T P.800. *Methods for subjective determination of transmission quality*. 1996.

[58] J. H.L. Hansen and B. L. Pellom. “An effective quality evaluation protocol for speech enhancement algorithms”. In: *ICSLP*. Vol. 7. 1998, pp. 2819–2822.

[59] American National Standards Institute (ANSI). *Methods for calculation of the speech intelligibility index*. 1997.

[60] A. W. Rix et al. “Objective Assessment of Speech and Audio Quality: Technology and Applications”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.6 (Nov. 2006), pp. 1890–1901.

[61] A. W. Rix et al. “Perceptual Evaluation of Speech Quality (PESQ) - A New Method for Speech Quality Assessment of Telephone Networks and Codecs”. In: *Proceedings of the Acoustics, Speech, and Signal Processing, 200. On IEEE International Conference - Volume 02*. ICASSP ’01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 749–752.

[62] C. H. Taal et al. “An Algorithm for Intelligibility Prediction of Time-Frequency Weighted Noisy Speech”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.7 (Sept. 2011), pp. 2125–2136.

[63] K. Paliwal, B. Schwerin, and K. Wójcicki. “Role of modulation magnitude and phase spectrum towards speech intelligibility”. In: *Speech Communication* 53.3 (2011), pp. 327–339.

[64] Kondo Kazuhiro. *Subjective Quality Measurement of Speech*. Springer, 2012.

[65] M. Brookes. *The Matrix Reference Manual*. 1998-2011. URL: `http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/intro.html`.

[66] Y. Ephraim and D. Malah. “Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32.6 (Dec. 1984), pp. 1109–1121.

[67] R. Martin. “Noise power spectral density estimation based on optimal smoothing and minimum statistics”. In: *IEEE Transactions on Speech and Audio Processing* 9.5 (July 2001), pp. 504–512.