Deep Learning Approach to Speech Recognition:
A Signal Extractor & Producer for Artificial General Intelligence

by
Ahmad Zuhair S. Hasanain

Master of Science
Electrical Engineering
Florida Institute of Technology
2017

Bachelor of Science
Electrical Engineering
King Abdulaziz University
2009

A dissertation
submitted to the College of Engineering and Science
at Florida Institute of Technology
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy
in
Electrical Engineering

Melbourne, Florida
December, 2022
We the undersigned committee hereby approve the attached dissertation

Deep Learning Approach to Speech Recognition: A Signal Extractor & Producer for Artificial General Intelligence

by

Ahmad Zuhair S. Hasanain

Veton Z. Këpuska, Ph.D.  
Associate Professor  
Electrical Engineering  
Committee Chair

Marius C. Silaghi, Ph.D.  
Professor  
Computer Science  
Outside Committee Member

Georgios C. Anagnostopoulos, Ph.D.  
Associate Professor  
Electrical Engineering  
Committee Member

Ivica N. Kostanic, Ph.D.  
Associate Professor  
Electrical Engineering  
Committee Member

Philip J. Bernhard, Ph.D.  
Associate Professor & Head  
Computer Engineering And Sciences
The efficient use of a communication bandwidth starts with the data source. The features of the speech signals can be extracted and reconstructed to lower the Internet traffic of the acoustic artificial agents and to increase the quality of the automatic speech recognition systems. The Speech Quefrency Transform (SQT) is hereby introduced in the work to enrich the communication space between the artificial agents and mankind. We describe the motivation, methodology, and deep learning approach in detail as we apply the SQT technology to several applications: sharp pitch track extraction, real-time speech communications, and emotion recognition. The results were excellent. The work proves that the acceleration is the unit of quefrency and advocates for the adoption of the geometric scale for the cepstrum domain. It also proposes spectral banking to model the quefrency filters by the means of controlling the spectral leakage. This dissertation shows how to generate, combine, and apply the filters.
# Table of Contents

Abstract ........................................................................................................ iii

List of Figures .................................................................................................. vi

List of Tables ................................................................................................... ix

Acknowledgments ............................................................................................ x

Dedication .......................................................................................................... xi

1 Introduction .................................................................................................. 1
   1.1 Speech Signals ......................................................................................... 5
   1.2 Speech Perception .................................................................................. 11
   1.3 Web Clients ............................................................................................ 13
   1.4 Speech Depth ......................................................................................... 15

2 Review .......................................................................................................... 17
   2.1 Extraction Domains ............................................................................... 18
   2.2 Extraction Methodologies ...................................................................... 19
   2.3 Paralinguistic Intonation ....................................................................... 23
   2.4 Language Intelligence ............................................................................ 25

3 Methodology ................................................................................................. 29
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Theory</td>
<td>31</td>
</tr>
<tr>
<td>3.2</td>
<td>Unit</td>
<td>34</td>
</tr>
<tr>
<td>3.3</td>
<td>Scale</td>
<td>36</td>
</tr>
<tr>
<td>3.4</td>
<td>Modeling</td>
<td>39</td>
</tr>
<tr>
<td>3.5</td>
<td>Phase</td>
<td>45</td>
</tr>
<tr>
<td>3.6</td>
<td>Quefrency Demodulation</td>
<td>51</td>
</tr>
<tr>
<td>3.7</td>
<td>Angular Window</td>
<td>57</td>
</tr>
<tr>
<td>3.8</td>
<td>Multidimensional Quefrency Transform</td>
<td>67</td>
</tr>
<tr>
<td>3.9</td>
<td>Speech Reproduction</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>Results</td>
<td>84</td>
</tr>
<tr>
<td>4.1</td>
<td>Proof of Concept</td>
<td>85</td>
</tr>
<tr>
<td>4.2</td>
<td>Minimal Viable Product</td>
<td>93</td>
</tr>
<tr>
<td>4.3</td>
<td>Speech Emotion Detection</td>
<td>100</td>
</tr>
<tr>
<td>4.4</td>
<td>Summary</td>
<td>116</td>
</tr>
<tr>
<td>5</td>
<td>Conclusion</td>
<td>119</td>
</tr>
<tr>
<td>5.1</td>
<td>Statement</td>
<td>120</td>
</tr>
<tr>
<td>5.2</td>
<td>Implications</td>
<td>121</td>
</tr>
<tr>
<td>5.3</td>
<td>Novelty</td>
<td>127</td>
</tr>
<tr>
<td>5.4</td>
<td>Contributions</td>
<td>129</td>
</tr>
</tbody>
</table>

References 130
List of Figures

1.1 Spectrograms of Multi- and Mono-Resonance Communication Systems . . 6
1.2 Stationarity in Speech Frames . . . . . . . . . . . . . . . . . . . . . . . . 7
1.3 Phoneme Representation in the Spectrum . . . . . . . . . . . . . . . . . . 9
1.4 Pulse Trains . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
1.5 Web Client Flowchart . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

2.1 Pitch Perplexity Challenge [67] . . . . . . . . . . . . . . . . . . . . . . . 20
2.2 Mel Double-Sided Effect on MFCC . . . . . . . . . . . . . . . . . . . . . 21
2.3 Multi Pitch Detection Result . . . . . . . . . . . . . . . . . . . . . . . . . 22
2.4 Our Visualization of the Variable Length Auto-Encoder . . . . . . . . . . 27

3.1 A Cosine Function and Its Third-Order Approximation . . . . . . . . . . . 33
3.2 Gaussian Frequency Banks . . . . . . . . . . . . . . . . . . . . . . . . . . 33
3.3 Quefrency Scales . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
3.4 Frequency Scales . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
3.5 Sinc Frequency Banks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44
3.6 Detection Phase . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
3.7 Aliasing in a 1-Hz Triangular Wavelet . . . . . . . . . . . . . . . . . . . . 48
3.8 The Effect of the Harmonic Order on the Quefrency Approximation: $\gamma = 2$ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
3.9 Discrete Derivative Functions . . . . . . . . . . . . . . . . . . . . . . . . . 55
3.10 Angular Window . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.11</td>
<td>AngWin Speech Demodulation Space</td>
<td>61</td>
</tr>
<tr>
<td>3.12</td>
<td>AngWin Modulation Effect</td>
<td>64</td>
</tr>
<tr>
<td>3.13</td>
<td>SQT Negative Cases</td>
<td>68</td>
</tr>
<tr>
<td>3.14</td>
<td>SQT Positive Cases</td>
<td>69</td>
</tr>
<tr>
<td>3.15</td>
<td>Componentwise Quefrency Filter</td>
<td>71</td>
</tr>
<tr>
<td>3.16</td>
<td>Quefrency Filter: $\gamma = 4$</td>
<td>73</td>
</tr>
<tr>
<td>3.17</td>
<td>Forming Harmonic Vectors by Map-Reduce</td>
<td>74</td>
</tr>
<tr>
<td>3.18</td>
<td>The Effect of the Number of Cycles on Rectangular Windows: $M = 4$</td>
<td>77</td>
</tr>
<tr>
<td>3.19</td>
<td>Adjustment of the Additive Spectrum of Multiple Rectangular Windows</td>
<td>78</td>
</tr>
<tr>
<td>3.20</td>
<td>Variable Window Overlapping with Fixed Frame Rates</td>
<td>81</td>
</tr>
<tr>
<td>4.1</td>
<td>GPEs of Pitch Tracks</td>
<td>87</td>
</tr>
<tr>
<td>4.2</td>
<td>Error Rates by Levels of Additive White Noise</td>
<td>88</td>
</tr>
<tr>
<td>4.3</td>
<td>Error Rates by Levels of Additive Turbine Noise</td>
<td>89</td>
</tr>
<tr>
<td>4.4</td>
<td>Mean Squared Error (MSE) by Noise Type and Level</td>
<td>91</td>
</tr>
<tr>
<td>4.5</td>
<td>SQT.m Quantization Error</td>
<td>92</td>
</tr>
<tr>
<td>4.6</td>
<td>Waveform Recovered from Undistorted SQT Features</td>
<td>92</td>
</tr>
<tr>
<td>4.7</td>
<td>Speech Normalization by Formant Alignment</td>
<td>95</td>
</tr>
<tr>
<td>4.8</td>
<td>Capped Buffers of Speech Features Chunks in Serverless Streaming</td>
<td>96</td>
</tr>
<tr>
<td>4.9</td>
<td>SQT.JS Result of Streaming</td>
<td>97</td>
</tr>
<tr>
<td>4.10</td>
<td>SQT.JS Result of Repeated Word with Different Intonations</td>
<td>99</td>
</tr>
<tr>
<td>4.11</td>
<td>Results of Complex Extraction by SQT.PY</td>
<td>101</td>
</tr>
<tr>
<td>4.12</td>
<td>Feature Engineering Procedure of SQT.JS and SQT.PY</td>
<td>104</td>
</tr>
<tr>
<td>4.13</td>
<td>SpeechEmotions Architecture Layout &amp; Training Approach</td>
<td>105</td>
</tr>
<tr>
<td>4.14</td>
<td>Exploratory Data Analysis (EDA) of SQT RAVDESS</td>
<td>108</td>
</tr>
<tr>
<td>4.15</td>
<td>F1-Score Progress on RAVDESS by Label</td>
<td>110</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>4.16 Training Progress on RAVDESS on Average</td>
<td>111</td>
<td></td>
</tr>
<tr>
<td>4.17 The ROC Results of the SpeechEmotions RAVDESS by Label</td>
<td>112</td>
<td></td>
</tr>
<tr>
<td>4.18 The F1 Results of the SpeechEmotions RAVDESS by Label</td>
<td>113</td>
<td></td>
</tr>
<tr>
<td>4.19 Bagging Ensemble by Single Classifiers</td>
<td>114</td>
<td></td>
</tr>
<tr>
<td>4.20 Summary of the SpeechEmotions RAVDESS Results</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td>5.1 Multi-Speaker Data Augmentation</td>
<td>122</td>
<td></td>
</tr>
<tr>
<td>5.2 Economical Unfiltered Cepstrograms</td>
<td>123</td>
<td></td>
</tr>
<tr>
<td>5.3 Spectrograms of Reconstructed Order-12 Harmonic Sequences ($M = 12$)</td>
<td>124</td>
<td></td>
</tr>
<tr>
<td>5.4 Effect of Economical Parameters on Balanced Accuracy</td>
<td>125</td>
<td></td>
</tr>
</tbody>
</table>
## List of Tables

3.1 The Components of the Cumulative Demodulation .......................... 65  
3.2 Bitrates of Raw Features by Frame Rate and Harmonic Order .......... 79  
4.1 Confusion Matrices of the RAVDESS Training & Validation Samples ... 115
Acknowledgements

Thanksgiving is owed to God the best of the creators and the Lord of universe, without Whom never can an existence be, and to my parents, who cherished my learning opportunities. I am also thankful for Umm Al-Qura University and King Abdullah Scholarship Program for sponsoring the research; the Saudi Digital Academy and SDAIA Academy for their amazing bootcamps; Prof Judith B Strother and Dr Tawfiq Hasanin for proofreading the dissertation; MiSK SPARK, KAUST Game Jam, Saudi Space Apps Challenge for their proceedings; as well as Dr Anthony O’Neil Smith, Muntaser Syed, Prof Debasis Mitra, Dr Chul-Ho Lee, Prof Jean-Paul Pinelli, Prof Chelakara Subramanian, and colleagues for their collaborations.
In the Name of God the Beneficent & Merciful
Chapter 1

Introduction

• *Half of all Internet, new Internet connections are coming to mobile devices at one kind or another.*
  
  – Eric Emerson Schmidt, former Google CEO

• *Inspection to improve quality is too late, ineffective, costly. Quality comes not from inspection, but from the improvement of the production process.*
  
  – William Edwards Deming, PhD

• *And this of course will add trillions of new devices to the network within the next few years, not to mention the huge volumes of information - data actually first - that must be stored and analyzed and managed and shared, because of course, the goal is to transform data into information and information into insight.*
  
  – Cara Carleton Fiorina, first female CEO of HP

• *More data beats clever algorithms, but better data beats more data.*
  
  – Peter Norvig, PhD
Even though Machine Learning (ML) is theoretically capable of modeling complex systems, its methods are not plug-and-play solutions that can readily solve any mathematical problem given raw data. Several parameters have to be configured. Data pipelines must be engineered for each and every ML application for Artificial Intelligence (AI) to work promptly. For example, the input features’ range and distributions influence the quality of the clusters of the k-means-based methods. If an unsupervised model is not designed intelligently, complex ML systems have little chance to become intelligent where there are infinite hypotheses to explore and when it converges to local solutions. Similarly, Artificial Neural Networks (ANN) incompletely search hypotheses’ spaces and converge promptly to a very good solution only when the relevant hyperparameters are fine tuned to the specific application and when an outstandingly reachable goal solution can be expressed by the feature space. The approach to having good speech models starts by and is mostly made of extracting high quality information from speech data. This is true in multiple data domains. "It is about transforming data from passive to active, from static to dynamic - transforming data into insight. Now, all of this demands a new approach to information technology from the approaches of the 80s or the 90s," Carly Fiorina said at Oracle OpenWorld 2004. Hereby the robust data representation of the Speech Quefrency Transform (SQT) is happily introduced — mainly for speech features extraction. It can certainly be valuable in speech synthesis and sound engineering; however, boosting Artificial General Intelligence (AGI) is the purpose of the SQT model so that AI may perceive and communicate emotions and, ultimately, the human intent which underlies Wake-Up-Word (WUU) and generally the speech commands. In other words, the SQT feature engineering is expanding the Automatic Speech Recognition (ASR) capabilities for Explainable, Cognitive AI.

ML-based optimizations may not be as efficient as engineered optimizations since the latter ensures the integration of the relevant theorems. A possible alternative to this work is to have ML algorithms pre-trained on feature engineering, but it is tremendously difficult
for ML models to select features and find scientific theorems correctly without consuming many resources. "The transformation can only be accomplished by man, not by hardware (computers, gadgets, automation, new machinery). A company can not buy its way into quality," W Edwards Deming wrote in 1982 Out of the Crisis. Since the feature state space heavily influences the behavior of the ML algorithms, the initial conditions stimulate the random processes' output quality. The informed search is obviously faster than the uninformed search. For example, an engineered model needs resources that are far fewer than the filters of its Convolutions Neural Networks (CNN) counterpart need for signal anti-aliasing. There is no need to add many ML filters and layers to rediscover the equations of the well-established theorems such as the Fourier Theorem. They are simply integrated in Automatic Speech Recognition (ASR) systems. Even the state of the art language models struggle with efficiency. Human beings require training examples that are much fewer than the Generative Pre-trained Transformer 3 (GPT-3) needs to leverage simple tasks [14].

Efficiency is becoming more necessary as the Internet traffic and number of network nodes are increasing. The number of connected Internet of Things (IoT) devices is expected to double within five years from the 2020 estimated figure of 11.3 billion [61], generating a predicted value of more than seventy Zettabytes ($73.1 \times 10^{21}$ bytes) of data by 2025 [37]. This is not surprising since heavy signal processing tasks are moving to cloud computing for scalability. For instance, the Google speech services have been duplicating voice data in multiple cloud clusters for voting ensembles. "An Exabyte is roughly a million gigabytes. We generate [roughly five Exabytes ($5.0 \times 10^{18}$ bytes)] in every two days now," Eric Schmidt said at the ASNE NewsNow 2010 Ideas Summit. By the same token, excluding its client-side WUW detection, the Amazon Alexa Assistant has been sending raw data to cloud computing for signal processing [8]. Unlike Free Lossless Audio Codec (FLAC), lossy client-based encoders can significantly lower not only the network traffic but also the round-trip time (RTT) because there is a large degree of redundancy in the raw speech data. The encoded
speech data can be decoded also at the client. With a multi-device compatible, open-source encoding unit, the AI services may become faster, and the network bandwidth utilization as well as the functioning costs may be minimized while the high quality Natural Language Processing (NLP) services are maintained.

“We don’t have better algorithms than anyone else; we just have more data,” Peter Norvig said at Google Zeitgeist 2011. AI would exhibit naturally reflexive word recognition when the voice characteristics of emotions in human speech are preserved in the speech feature. If an AI system does not receive speech features that capture emotions, and if its model fails to find such features, AI would have to rely on unreliable secondhand clues for emotion detection, such as explicit vocal expressions, whose meaning can be negated. Speech AI engineers have been having to choose from two compromising speech extraction procedures, namely the Mel-Frequency Cepstral Coefficients (MFCC) and the spectrogram of the Fast Fourier Transform (FFT). While the former selects the features but does not preserve crucial information for ML, the latter preserves the crucial information but does not select the features. An alternative approach that strikes a balanced trade-off is the Speech Quefrency Transform, which we are presenting in our work. The SQT feature space is more expansive than the spectrogram and more preservative than the MFCC features as shown in the second and fourth chapters.

The first chapter continues with a gentle introduction to speech signals, web clients, and vocal depths. Chapter 2 on page 17 sums up a brief literature review. Chapter 3 (page 29) sets forth the methodology of the SQT model, which consists of a feature extractor and a signal producer and normalizes speech formant features for human-machine interactions, constituting the minimal feature representation that is common for both natural intelligence and AI to perceive and produce speech signals. The methodology chapter includes discussions about the most appropriate unit, scale, model, and spectral filtering of the quefrency. It also describes a cepstral filtering (liftering) procedure that easily removes perplexity from the
pitch tracks. Although the SQT model has been designed meticulously for Artificial General Intelligence (AGI), so far it has only been tested with the resources that were available. Chapter 4 (page 84) provides in-depth analysis and tangible results in several applications of the SQT technology: pitch analysis, speech streaming, and deep learning. We showcase a small representative set of illustrations to avoid information overload, hoping the highlighted points are informative. The contributions of the work are summarized after the conclusion in Chapter 5, which is on page 119.

1.1 Speech Signals

At first glance at Figure 1.1, the reader may notice the parallel curves in the spectrogram of the human voice [50] (Figure 1.1a) but not in the bird chirrup (Figure 1.1b). The salient curly harmonics of the voice render a hidden state that appears contentious when connecting the dots. The speech producer has a hidden state because it is measured indirectly. The speech spectrogram is a graph of the power or energy distribution along audio frequencies (i.e., the spectrum) versus time. Power is the time rate of energy and is usually in decibels (dB); however, power and energy may be used interchangeably since the intensity contrast may undergo Gamma Correction and Min-Max Normalization in various devices. The contrasts of the pictured graphs are adjusted per the printing norms; the higher the energies, the darker the pixels, but the color scheme is usually inverted when printed on diagnostic monitors to save energy. The human speech can be captured from the spectrogram using two features: the pitch and the harmonic intensities (coefficients or power components). In order to have artificial agents processing (or understanding) spoken languages in a human manner, it is crucial to realize a mathematical representation of speech that is attuned accordingly, especially because human intelligence and language are tangled up during development.

The linearly-spaced curves in the spectrogram are the outcome of a periodic multitone
signal that is locally stationary. In other words, the periodicity and the repeated pattern fluctuate slowly relatively to the sampling rate. Figure 1.2 depicts an example of a discrete time series that captured the speech state transitions between a voiced phoneme, a speech stop, and a speech pulse. The speech code correlates with the shape of one period of the fundamental waveform. It can be modeled in the time domain, in which it exhibits a variation of a sine cardinal (sinc) function, or modeled in the frequency domain, in which it can be approximated by Gaussian Mixture Models (GMMs). The more the signal is

![Spectrograms of Multi- and Mono-Resonance Communication Systems](image)

(a) Child Babble

(b) Canary Twitter

Figure 1.1: Spectrograms of Multi- and Mono-Resonance Communication Systems
locally stationary, the sharper the curves are. Speech producers flap in response to internal air pressure, air molecules are compressed and released periodically, and the pulse shape makes the speech signal transmittable through air. The time distance between two adjacent compressions (bursts, pulses, or cycles) is the wave period \( T_0 \), measured in seconds per cycle, or one over Hertz (1/Hz). The wave-interval is the reciprocal of the minimum frequency shift between two harmonic curves, as in Equation 1.1. This minimal shift is the speech fundamental frequency (FF or \( f_0 \)). Per context, it is also the pitch and the frequency carrier. However, being in an air channel as its communication medium, the periodicity of the signal is in meters per cycle. The \( \lambda_0 \) and \( \nu \) in the equation are the corresponding wavelength and the speed of sound in the medium, respectively. Although the variables are time variants, the \( \nu \) is usually assumed to be constant, but the temperature, humidity, and wind speed, all of which slightly affect \( \nu \), are not constant along the air paths from the speech producer (vocal folds, cords, or glottis) to the speech receivers (eardrums’ cochleas, microphones, and acoustic beamformers).

\[
T_0 = \lambda_0 / \nu = 1 / f_0 \text{ (second-per-cycle)} \tag{1.1}
\]

In Figure 1.1b, the transitioning of the birdsong \( f_0 \) is constrained in the producer’s hypercoordinates, but the \( f_0 \) observations are projected onto the two-dimensional spectrogram. The
projection onto the periodicity space is non-linear since the $f_0$ teleports in the spectrogram as though two frequencies (such as 1 kHz and 4 kHz) are identical because there are unaccounted independent axes. For example, the fundamental waveform of the canary is visually rotated around a time-variant axis parallel to the time axis, and its perimeter path renders a visual effect of cylinders that are visible. Assuming the bird’s mono $f_0$ was traveling with a constant angular velocity in a polar coordinate, the inferred radius of a pictured cylinder is about 1 kHz and centered at 2 kHz. Similarly, the infant voice in Figure 1.1a appears with a deeper voice during an emotional outburst, between Second 3.25 and 3.6. The event is noticeable in the figure and also in the audio playback. The tone-change phenomenon, which usually happens during puberty, doubles the fundamental interval and folds up the spectral code bandwidth. This creates the speech resolutions, which are important for speakers’ normalization. It is commonly known that deep and high human voices are not represented equally in a telephone bandwidth (within 4 kHz). Additionally, the variable scaling of the spectral bandwidth has been one of the main challenges in Automatic Speech Recognition (ASR). In order to normalize the speech features, the speaker’s pitch must be considered during the feature extraction process.

Per the juxtaposition of the two spectrograms, the human voice has a fundamental waveform, whose shape transformed in the figure at a relatively slow pace, and has parallel spectral curves, each of which produced a component. The spectral energies of the harmonic components are mainly the speech features. The harmonic components can also be called timbre or overtone series. They function as the frequency-modulating signal and the vocal tract response, denoted $H_m$ in this work for modeling the speech system. According to a 1989 study [65], the human perception of speech is similar frequency demodulation. If the parallel curves were the openings of window blinds, a few shaded patterns would appear behind the blinds. The shaded patterns are the formants, and their mixtures’ variations compose the phonemes. A phoneme is a distinctive sound, resulting from convolving the multitone signal
with the formant system, as illustrated in Figure 1.3. The human speech consists of these components, which are conveying the hidden shape of the spatial cavities of the vocal tract (the nasal and oral cavities). The collective shape of the tract is a system through which the molecules’ excitations of the fundamental waveform pass. The output of the modulating system is the speech signal, which, due to its local stationarity, consists of recognizable time units.

Unlike the birdsong but like many mammals’ sounds, human speech consists primarily of the multitone signal that travels through a multi-resonance vocal system, and it is characterized by local periodicity as sketched in Figure 1.4. Equivalently, in the frequency domain, as shown in Figure 1.3, the glottal train of the multitone signal is multiplied by a Gaussian Mixture Model of the formants. The periodicity, hence the fundamental frequency $f_0$, controls the
frequency spacing between the elements of the speech code in the channel bandwidth, and high speech components are attenuated. Two commonplace approaches to pitch and speech feature extractions are modulation-based and are applied to the frequency domain. The Fast Fourier Transform, albeit invaluable for several applications, can complicate some speech processing tasks such as $f_0$ tracking. Also applying the Fourier and/or the Cosine Transforms twice for the quefrency domain does not normalize the speech features. The term quefrency was coined in 1963 and was derived from the term frequency; compare qu-e-fr-ency and fr-e-qu-ency [12]. The quefrencies are sometimes treated as the inverse of frequencies. However, the quefrencies are actually frequencies of frequencies, and the cepstrogram are spectrograms of spectrograms. The next two chapters address the misconception since there is a need for a robust acoustic front-end that effortlessly extracts and produces speech utterances. The speech front-end represents spoken utterance by the speech feature space, which is important for generating acoustic and language models. Language modeling is a different topic but is briefly mentioned in the next chapter to put the feature engineering front-end into context.
1.2 Speech Perception

Originally, speech signals are continuous functions with an infinite sampling rate \( f_s \). They become time-discrete when sampled. Low sampling rates consume less storage space than high sampling rates do since, according to the Nyquist Theorem, the bandwidth must be less than or equal to half the sampling rate. Frequencies outside the bandwidth interfere if not filtered out. The sampling rate \( f_s \) is 8 kHz in narrowband or landline telephony and is 16 kHz in wideband or High-Definition (HD) voice. For capturing sound effects, higher rates such as 22.05 kHz and 44.1 kHz are used in video gaming and full-resolution audio. Even higher \( f_s \) can be used; for example, Apple iTunes Store handles 96, 176.4, and 192 kHz. For human-level AGIs, the organic characteristics are heuristics that may be used to simplify the search (or prune the hypothesis space).

Although \( f_s \) is proportional to the frequency range, it does not have a direct relationship with its resolution when the sampled audio is again discretized (enframed, filtered, or windowed) to time frames for spectral analysis. The frequency resolution relies on the window interval (span or length). In other words, one needs to increase the window length to measure lower frequencies, and an infinitely long window is hypothetically required to measure the almost zero frequency of the direct current (dc) signals.

Human hearing may perceive frequencies down to 20 Hz and may not discriminate between acoustic echos lagging less than 100 milliseconds (ms), according to psychoacoustic experiments [69]. That is, the organic window must be greater than or equal to 50 ms for the 20 Hz perception and greater than or equal to 100 ms for its anti-aliased detection. In practice, however, the window is usually selected between 15 ms and 50 ms. However, it cannot be shorter than 20 ms since the spectral range of the audio instruments is above 50 Hz [59] and since the speech volume at 50 Hz is considered the lowest level in the standardized acoustic loudness contours [63].
Generally, the human auditory perception is most receptive of the frequency band between 1 kHz and 5 kHz, and the minimal hearing threshold increases as one advances in age. Furthermore, the frequency components that are less than 60 Hz may not be perceived by everyone. It could have a relationship with the visual perception. For videos, 60 frames per second (fps) is considered a high frame rate standard, while 24 fps is the lowest. The two rates respectively correspond to 16ms and 42 ms of stillness between the frames. For audio, $f_s$ has an inverse relationship with the information gain, the bitrate and the storage. When each frame is an M-sized array with a datatype, which sets the bit depth or the number of bits to represent each element in the array, the bitrate is defined in Equation 1.2 and is measured in bit-per-second (bps). In the equation, the frame rate ($R_s$) is $f_s$ when $M = 1$ for raw samples. The raw samples are usually stored with the "wav" filename extension, and its coding is recognized as Pulse-Code Modulation (PCM). The bitrate decreases with lossless and lossy codecs. Respectively, FLAC and MPEG-1 Audio Layer 3 (MP3) are examples of the two types of compression. Being the lossy format, MP3 was standardized in 1993 [64] and is commonly used in the Internet for the relative quality despite the files’ sizes.

$$\text{Bitrate} = \text{Bit\_depth} \cdot M \cdot R_s \text{ (bps)} \quad (1.2)$$

The application of the Nyquist Theory can be extended to the window length. Consider an 8 kHz rate channel case with a secret 40 Hz tone signal or subchannel, which only youngsters can hear, where $\lambda_0 = 25$ ms or 200 samples. One would need to capture 200 samples to recognize the tone pattern and then another 200 samples to recognize the event repetitiveness; hence, the event is not interference from a higher voice or subchannel. Consequently, 400 subsequent time samples are needed in order to unambiguously observe the event that repeats every 200 samples or less, as in Figure 1.4. This case relates to the Nyquist Theory since the bandwidth of the window has to be less than or equal to 20 Hz,
which is half the subchannels’ minimum rate. In other words, to recognize the tone signal, the receiver simply needs to listen to at least twice the tone wavelength: $2 \cdot \lambda_0$ seconds, as further explained in the third chapter.

### 1.3 Web Clients

Web clients use Hypertext Transfer Protocol (HTTP) and HTTP Secure (HTTPS), which use the application-layer Transmission Control Protocol (TCP), to communicate to web servers over the Internet. For cybersecurity reasons, HTTPS has been the standard for transmitting users’ data, which is captured by wearable sensors and media devices.

New JavaScript (JS) applications are developed on top of existing software using their Application Programming Interfaces (APIs). Web Audio API is the interface of the audio software supported in common web browsers by software communities, usually affiliated with software companies such as Microsoft, Mozilla, and Google. The audio software has a JavaScript-syntax interface but is written in lower-level languages such as C++ and Rust [52]. It has built-in signal processing essentials, such as digital buffering, bi-quad filtering, and oscillator synthesizing [62]. Web browsers differ in implementation. For example, downsampling is not supported in every web browser, and Safari uses a few uncommon standards for the high sampling rates.

The web client fetches HTTP commands, mainly GET and POST for downloading and uploading. GET can operate in chunks for streaming, whereby the server sets the number of chunks per second for the client. Alternatively, POST can have an outbound function in a timed loop and an inbound function in the HTTP response. When each client has a timed loop that sets its request rate, the server needs only to track the clients’ refresh rates. Furthermore, an error-control buffer is necessary at reception for both the server and the client sides since the transmitted chunks could be routed differently by the network.
A flowchart of a client example is sketched in Figure 1.5 to enumerate the basic blocks. While the nodes from the Start to the Download Track are for the streaming loop, the blocks from the Speech Features to the Speech Extractor are for the audio processing. AudioContext is a pipeline instance that processes media from the source streams to the track destinations. Worklet is a threaded module extending the AudioContext, while Web Worker is a threaded JS script. WebGL works in GPU-enabled devices with modern web browsers. Anti-alias downsampling is simply low-pass filtering (LPF) followed by decimation, which is the weighted averages of the two samples that are nearest to the multiples of the output sampling interval.

The Speech Extractor and Producer map between the speech frames and the audio samples. The producer is relatively simpler than the extractor, but it needs to track the phase to avoid generating clicking sounds. The extractor and producer are explained in detail when revisited.

Figure 1.5: Web Client Flowchart
1.4 Speech Depth

The speech depths are commonly associated with masculinity, femininity, and infancy, but the overall majority of human beings naturally produce the resolution that is also a common characteristic of the youth voice. Since the medium of the air particles is equivalent to a low-pass channel, it attenuates the high frequencies of the utterances [36, 39]. Speech audio is also low-pass filtered (or anti-aliased) while it is acquired and stored. Because a higher $f_0$ spreads the spectral code to higher frequencies, each speech depth naturally has a speech resolution ($M$), not to be confused with the frequency resolution ($1/\lambda_w$). The variability of the speech depth may have been fostering communications within household members and facilitating language learning along one of the speech dimensions. Since each speech depth happened to have a speech resolution and a cepstral band, and since a simple projection can align the speech features of different speakers’ characteristics, the speech depth must be one of the $f_0$ coordinates.

Newborns first encounter blurred speech, and the low resolution may help human beings acquire and model languages in a gradual general-to-specific heuristic search, guiding the internal neural systems. Generally, increasing $f_0$ prioritizes the high voices over the deeper voices; this is ascribable to the voice masking phenomena, which is caused by the fact that two cycles of a high $f_0$ resemble one cycle of a low $f_0$, but not vice versa. Some may argue otherwise using the Fourier model, but the utterances are wavelet pulses, and the pulses of the high voices are more than the pulse of the deep voices. For instance, listening to a conversation whose background is voices of unsatisfied dependents requires extra processing to filter out the high voice. The unpleasant noise may further the survival of the species, preventing child neglect, so high priority members naturally have high voices. On the other hand, lowering or deepening the voice expands its vocal coverage since signals traveling on lower frequencies optimize their energy for long distance communications. It may have
been useful while hunting in open fields. Although gender may have a correlation because of the folds’ lengths [23], the association does not hold because of the existence of sizable minorities, and so the majority of human beings have high voices. Moreover, the voice generator can transit (or teleport) between the speech resolutions regardless of sex and age. For instance, parents tend to use infant-directed speech to promote communication and learning. Recent findings showed that infants who babble at early age receive contingent feedback from social interactions that foster language learning [2]. Likewise, infants can produce adolescent-directed speech. The illustration of Figure 1.1 about the phenomena confirms previous speculations in literature. For example, Hoeschele [35] and Warren and others [68] mentioned that the consensus of the two-dimension view of $f_0$ was motivated by expertise.

One may speculate about the underlying physical constraints that prompt the $\lambda_0$ halving or doubling; however, the teleport path indicates that $f_0$ has an additional dimension. In literature, a rotation around the second axis was called pitch height and was believed to be 110 Hz, although this number was not exactly a constant maybe because humans lived in different environments. A 2017 Bernhardsson survey [10] showed that the arithmetic population mean of $f_0$ varies per language. Moreover, similar to speech production, human perception of speech does not considerably discriminate between the intonations that are one octave apart [20]. The term pitch may imply the angular value while the $f_0$ refers to the actual value or measurement. The relation between the two terms is that the pitch equals $f_0$ modulo height. Speech is central to humankind, so there are several sciences that study speech, and each field projects a perspective to the SQT scope. The related literature intersection is presented in the next chapter.
Chapter 2

Review

This chapter briefly reviews the related work. The features of the speech signals can be extracted from the temporal, spectral, and cepstral domains. There are many methods, some of which may complement others’ weaknesses, and so they can be combined. The leading methods include Normalized Correlation Function (NCF), Pitch Estimation Filter (PEF), Cepstrum Pitch Determination (CPD), and Mel-Frequency Cepstral Coefficients (MFCC). The premised transform needs to filter the pitch track and not distort the power spectrum since there are subtle yet crucial fine speech details that are crucial for language emotion, emphasis, and grammar. Artificial General Intelligence (AGI) could be achieved by wide-span context models.
2.1 Extraction Domains

The glottal and format features can be extracted from speech signals in several ways, and even though there are many pitch extraction techniques [57], it has still been a challenging task [16, 33]. The main speech domains are three: time, frequency, and quefrency.

In the time domain, the signal can be matched with its lagged versions using auto-correlation, which is one of the basic methods for pitch extraction [56]. Since the speech signal is assumed to be stationary, it can be compared with itself, and the self similarity is measured high when the lag matches the wavelength. Another general way to estimate the pitch is from the number of the zero crossings or the sign flipping. In the frequency domain, the spectral components of the signal are obtained first, as has been shown in the spectrogram. Since the speech signals have overtones, the frequency components can indicate the fundamental frequency. For example, the instantaneous frequency and phase can be combined with the time analysis to extract the pitch track [40, 15] although reacquiring high frequency resolutions. Pitch Estimation Filter (PEF) uses the frequency domain.

In the cepstral domain, which is commonly used as the power spectrum of the logarithm of the power spectrum [11, 53, 54], the pitch track and its overtones become separated from the formant features, and cepstral filtering, also known as liftering, is applied to identify the pitch track from the overtone lookalikes. Examples of this approach are Cepstrum Pitch Determination (CPD) and Mel-Frequency Cepstral Coefficients (MFCC). The equation for calculating the cepstrum is defined as \( \mathcal{F}^{-1}\log|\mathcal{F}\{\bullet\}| \) where \( \mathcal{F}\{\bullet\} \) denotes a forward Fourier transformation, \( e^{-j\theta} = \cos(\theta) - j \cdot \sin(\theta) \), and \( j = \sqrt{-1} \) [58, 44]; hence, Equation 2.1.

\[
p[n] = \frac{1}{2c+1} \sum_{m=0}^{2c} e^{j2\pi \frac{mn}{2c+1}} \log \left| \sum_{u=0}^{2c} e^{-j2\pi \frac{mu}{2c+1}} s[u] \right|
\]

This section outlines the extraction domain categories: temporal, spectral, and cepstral. Our SQT methodology, which is presented in Chapter 3, calculates the cepstrum from the
spectrum domain, so the conventional definition and equation of cepstrum have to undo the log and Mel scales. In other words, quefrency is simply frequency of frequencies, and as the frequency is the independent variable of the spectrum, quefrency is similarly the independent variable of the cepstrum. The quefrency theory of Section 3.1 obtains the cepstral domain from the power spectrums, and this allows us to perform cepstral filtering on the factorized cepstral components in parallel. It makes sense to filter the cepstral components instead of the cepstrum. This concept is elaborated towards the end of the Methodology chapter.

2.2 Extraction Methodologies

The best pitch extraction methods that are most relevant to SQT are Normalized Correlation Function (NCF) [4], Pitch Estimation Filter (PEF) [25], and Mel-Frequency Cepstral Coefficients (MFCC) [49]. There are many other methods for pitch extraction, and their points of strength vary per application. Some methods are very simple but do not work well in noisy environments. For example, the Harmonic Product Spectrum [19] is common because it is used for adjusting acoustic instruments, so it fits non-human voice applications. Aiming for continuous improvement, the section focuses solely on the drawbacks of the previous methods and the solutions with which they were addressed in literature.

The main concern of the pitch track methods is differentiating between the pitch and its first overtone. The tones that are one unit octave apart from the true pitch track are lookalikes since they have common components. In literature, the lookalike noise was presented using the Normalized Cross-Correlation (NCF) method as in Figure 2.1 [67]. The cepstrogram of the figure shows a 140 Hz voice and its lookalikes at 70 and 210 Hz. Several methods [67, 22, 5, 45] were not instantaneously able to filter the pitch from its partial harmonic components.
Another challenge is the formant alignment. Mel-Frequency Cepstral Coefficients (MFCC) is considered one of the best speech feature extractions as it has been proven robust in various speech applications [24]. However, its model reliance on the Mel-Scale diminishes the fine speech details as illustrated in Figure 2.2. While the Mel Scale partly aligns and also increases the weight of the first speech formant, it ruins the remaining higher formants, which convey the most fine speech features. In other words, it reduces the weights of the high speech components since it heavily relies on one format observation that is at the lower side of the spectrum. Additionally, because of the MFCC spectral compression, it is too lossy for any meaningful, telligible MFCC feature reconstruction. There may be a way for MFCC to mitigate its Mel double-sided effect, but MFCC appears to be fundamentally based on the format misalignment.

There are two strategies to mitigate the drawbacks: combining and smoothing. In the first strategy, the pitch ambiguity are circumvented by putting together the pitch tracks of various methods [51]. Since the various features of the methods may be partially independent, putting them together may reduce the collective number of their blind spots. For example, "Yet Another Algorithm for Pitch Tracking" (YAAPT) [72] combines observation candidates from more than one pitch detection approach. The other workaround is smoothing. Since the pitch track is likely not to have abrupt changes due to the stationarity, multiple observations may stabilize the reading and filter out the outliers. Although the temporal averaging has been used to address the main concern [41], it lowers the detection quality, especially at the edges of the voice activity intervals. The original characteristics of the data get corrupted when
Figure 2.2: Mel Double-Sided Effect on MFCC
mean filters are applied on the pitch track. To conclude, combining multiple methods may not result in the most optimal way, but it may suggest that there has not been a comprehensive method that resolves the pitch ambiguity for good. Furthermore, median filters appeared to have a better effect than the mean filter. Although both workarounds are not optimal, never does it make sense to smooth the pitch patterns since the edges may carry vital speech features.

Several cutting-edge approaches have not been as robust as the natural speech agents. This was surprising since the natural auditory processors in mammals demodulate and compose speech signals effortlessly even though the natural agents operate at much slower processing speeds than today’s supercomputers. Using Matlab, the performance of the Normalized Correlation Function (NCF) and Pitch Estimation Filter (PEF) methods is shown in Chapter 4. In previous work [47], the error rates of NCF and PEF appeared lower than the error rates of several methods, such as Cepstrum Pitch Determination [53], Log-Harmonic Summation [32], and Summation of Residual Harmonics by [21]. The results show that it is possible to extract the pitch track in a way that makes it possible not only to rely only on one method but also to have the one method detect multiple pitch tracks instantaneously as shown in Figure 2.3. Using the SQT features, the voices of men, women, and children were separable and reconstructable. Apparently, SQT is relatively unique as it easily removes the overtone noise. Consequently, it may facilitate distant, multi-speaker, and simultaneous speech recognition in the AGI agents.

Figure 2.3: Multi Pitch Detection Result
2.3 Paralinguistic Intonation

Speech utterances sound natural when they are carried on the $f_0$ signal, which is a continuous function. The fundamental frequency ($f_0$) is the main speech feature and is vital in natural language processing, especially in languages where speech stresses play a major role in defining the speech parts and grammar. For example, modeling English grammar must lean on the voice patterns.

The intonation patterns are sometimes marked with diacritics since they are essential in word recognition and are common in Eastern languages. According to Albert Mehrabian [48], the nonverbal human tone conveys shades of meanings. For instance, the only difference between the two Korean words for sit and hug is that the emphasis makes the second "anta" slightly longer. Beside the double consonants in Korean [66], other examples are the acute accentuation in Greek, the compound words in English, and the tashdíd emphasis in Arabic. For instance, the stressed syllable in "thermometer" is just as important as its phoneme sequence. Moreover, the regular pattern of the $f_0$ is a reduction since it correlates with the breathing pattern. For instance, adults breath slower and so naturally do their glottises. The general pattern was shown in previous publications, according to which the emotions, such as happiness and sadness, also correlate with the pitch pattern [27]. Utterances normally de-accelerate as air pressure reduces. In contrast, $f_0$ increments appear while modeling upcoming emotional expressions and emphasis.

The $f_0$ function holds important clues for several grammar components: punctuation periods, clauses, and stresses. A persistent upward trend may imply a preparation; the accelerating is salient in the regularly de-accelerating pattern. Exclamation points and question marks are slightly similar; the former is a (linear) trend across the utterance, while the latter is an $f_0$ suffix, which is usually exponential. Additionally, the speech emphasis is a short up-down bounce. The emphasis appears to be the most frequent pattern. A varying
$f_0$ can possibly grasp attention as the auditory focuses of the listeners may intersect with the speaker’s tone. Such an intersection maximizes the reception of the speech. That is, words uttered with bouncing tones do sound emphasized as they widen in the quefrency domain.

Attention is a well-known concept in computer vision and deep learning [7, 71]. For instance, saliency maps are used to highlight specific spatial or temporal portions of the input sample. In speech recognition and translation, increasing the weight of some parts of speech may imitate the biological cognitive attention, so the important temporal parts are softly selected while the unimportant parts are softly discarded. For example, Head Fusion [70] improved the classification accuracy in the RAVDESS emotion recognition task. The attention technique could be related to our work. However, our approach is based on feature engineering and does not explicitly select nor weight the temporal portions, so the two are inherently different. We believe the $f_0$ features can help deep learning locate the importance of the portions by itself. The feature selection of the SQT is done on the frequency domain via the quefrency domain, not on the time domain as attention techniques do. Our approach simply gives the ML algorithms the needed features, and the attention concept may be achieved more naturally since the $f_0$ pattern can grasp attention. The SQT approach was proposed to several publishers in 2019, 2020, and 2021, and was accepted in 2022 [30]. Receiving several peer-reviews responses, we added further supportive arguments to the proof of the quefrency unit in Sections 3.2 and 3.5, which had been included since the unit is important to the field of signal processing and engineering systems. A quefrency value represents a set of frequency values.

The syllables of verbs are emphasized differently than those of nouns. The utterances of neutral statements and wake-up-word requests diverge naturally in the $f_0$ pattern. The $f_0$ highlights some parts of speech, and this is crucial for machine learning and language understanding. This is because the $f_0$ is a premier component in natural languages.
2.4 Language Intelligence

The generation and recognition of spoken languages are sophisticated processes. Since the intelligence of a species function as a means of its survival, the two processes may have been genetically optimized during a lengthy selective reproduction phase, as languages interconnect with and boost intelligence. For example, having been bestowed control over breaths, species, like camels, dolphins, elephants, and birds, have been able to extend their senses beyond the line of sight and share alerts and information within the kind. For instance, whales communicate at long distances and navigate their surroundings, transmitting sound units and receiving sonar echos. The utilization of the spoken units is a founding module for intelligence just as the human utilization of written alphabetic and numerical symbols is a basis for written knowledge, commerce, and civilization. Human languages come in several forms, and each has a countable number of units (or letters). In the spoken one, the phoneme is the unit of speech, and the sense of speech units may be directly linked to intelligence.

The ability of comprehending logic and sequential series of events must have been, to a certain degree, built upon the primal ability of recognizing sensory data, one of which is speech. The former is equivalent to the latter when the domain of the speech spans multiple days as opposed to minutes. Some may argue that some individuals learn to walk first while some talk first. Even so, human language acquisition begins much earlier with quasi-resonant vocalizations according to psychologist Rachel Albert [3]. Furthermore, the essence of the language may be stored in the genetic structure. According to linguist Noam Chomsky [17], there exists an innate Language Acquisition Device (LAD) in the brain that prepositions concepts, such as nouns and verbs, in the human languages. It is true that the tuneful pattern is not exclusive to humans; however, the speech ability has been vital to humankind [55]. The reasoning is that graceful recognition of temporal sequences also enabled the cognition of long-term chronological occurrences, which, when optimized in succeeding iterative genetic
mutations, may have facilitated the emergence of intellect in the species. Constituting a self-aware agent, hierarchical spans of language are levels for intelligence. Additionally, Magnetic Resonance Imaging (MRI) revealed that both the processing of language and the ability of using tools stimulate similar neural areas [34]. Therefore, speech signal processing is part of decision-making processes.

Artificial systems equipped with digital artificial front ends may have a human-like learning phase as well as subjectivity and perhaps artificial feelings. Acoustic, language, and transition models can be generated directly from the auto-encoding of the produced speech frames. The time compression layers may handle time warping. An auto-encoder network consists of two sets of layers for dimension reduction and dimension expansion [26]. The number of the layers and the time compression rate of each layer set the maximum span the network can memorize, which can be minutes, weeks, or years. The intermediate outputs across the network would look similar to the processing hierarchy in Figure 2.4. In the hierarchy, the encoding layers form acoustic models, while the decoding layers form generative models, and at the top is the most time-compressed output of the auto-encoder, which may represent the world state of artificial agent and include its maximum unconscious capacity. The stacked encoding/decoding layers can be trained at the same time by averaging the decoded signals. Once the encoding and decoding layers are trained, fully connected layers may be added between the stacked layers for language, inference, and transition modeling. The fixed window [9] neural language layers can be trained to map from the intermediate encoding outputs to the agent world state, so it may estimate the world state from the given observation. The inverse of the language layers is the generative inference layers at the decoding side. Finally, the transition may be modeled by a layer that maps between the world states. For example, it may translate from questions to answers, today to tomorrow, or from speech to text. Determining the outputs’ lengths may require weighted similarities.

Some people may have doubts or concerns about intelligent machines; however, corre-
Figure 2.4: Our Visualization of the Variable Length Auto-Encoder

...
The general intelligence agents are very likely to sense and express emotions when the emotion features are preserved in the speech feature space, and the SQT transform does that by providing a reconstructable speech space that is comprehensible by both AI and humankind in a means of communication that is more direct than using the intermediate textual phoneme representations.
Chapter 3

Methodology

Since the acoustic perception is receptive to frequency-modulated signals, and since the vocal features are separable in the quefrency domain, the cepstrum is a sophisticated speech feature space. Some may argue that not all speech phonemes are periodical because there are phonemes that are unvoiced. However, the unvoiced features have a spectral presence and so could still be detected by cepstral filters. Additionally, the unvoiced features are not entirely unvoiced. They are usually coupled by voiced components to strengthen their air transmission. The unvoiced speech units are variations of broadband noise, such as the violet noise, and can be modeled by low-order Finite Impulse Response (FIR) filters since they have proportionally low frequency resolution, which may not detect the glottal voice. One way to highlight the voiced and unvoiced units is by stacking two differently configured spectrograms. However, joining the wide- and narrow-band spectrograms would require adjusting two frame rates since the former has more temporal resolution than the latter. Another method to increase the unvoiced phonemes is by setting the frame rate \( R_x \) of the narrow-band spectrogram beyond 90 frames per second (fps) — or equivalently the frame stride to less than \( \lfloor 0.0\bar{1} \cdot f_s \rfloor \) samples. In samples, milliseconds, or percentage, the frame
stride (or step) is the interval between the beginning (or any similar) points of two adjacent frames. Out of the frame length, its percentage is the complement of the frame overlapping. In other words, the voiced features are salient in the narrow-band spectrograms, and the unvoiced features, like the features of the speech pulses and noises, are salient in the wide-band spectrograms and can be made salient in the narrow-band spectrograms by increasing the number of frames. Note that, once the signal is in frames, the speech time series is re-sampled from the sampling rate \( f_s \) to the frame rate \( R_s = 1/\text{stride} \) (fps). The frames can be transformed to spectrograms with several spectral configurations; therefore, the frequency domain of the frames can be sampled in a way that transforms the samples of the frames directly to the cepstrogram.

The chapter realizes the speech data transformation gradually from concept to practice, axiomatizing the appropriate unit, scale, and model of quefrency and the cepstral filtering. Using frequency sampling, the quefrency domain can be obtained with one direct transformation in several ways. The chapter concludes by formulating the multi-dimensional transformation whose expansive method is theoretically more efficient than the combinative method of the wavelet filters.
3.1 Theory

While the speech time samples have positive and negative values oscillating around a zero mean, the speech frequency samples are represented with positive energy magnitudes, each of which has an angular phase whose range is in $[0, 2\pi)$ radians (rad). The real part of the energy vector is considered negative in the third and fourth quarters: $[\pi, 2\pi)$ rad. The negative energy occurs when the signal and the extracting real-part filter have opposite signs; that is, when the signal is in the first angular half while the filter is in the second angular half, or vice versa. This is generally true for sinusoidal filters but not for wavelet filters, which may need more than two halves.

One possible way to obtain a quefrency value directly from the time samples is by designing an all-pass filter whose phase curve approximates a cosine function. Another hypothetical way is to design an all ripple frequency response. However, both of them are not straightforward as we found out. The practical method is to obtain one component at a time. To elaborate, applying a few complex filters is much simpler than aligning the phases of the multi-component wavelets. The computational complexity of the multi-dimensional method increases linearly, while the other two methods factorially, which is faster than exponentially. For example, for four harmonic components, only eight filters are required in total for the phase synchronization when applied individually, but 24 filters are required when the components are combined — for the harmonic combinations. Wavelet filters are not practical even though they may look like a shortcut. This is because their phase alignment or synchronization is more demanding than the frequency decomposition. In addition, their high coefficients may have to be weighted less than their lower coefficients. Last but not least, the cosine function is applicable along the time axis but not directly applicable to the frequency axis. The quefrency model, therefore, should be constructable by a set of temporal sinusoidal filters whose frequency responses can assemble a spectral sinusoidal filter.
Fortunately, this is achievable using windowed cosine filters and Stone-Weierstrass Theory, which states the fact that time-bounded signals can be approximated by sums of polynomials. In order to approximate the trigonometric function in the frequency domain, there are a few constraints. Since the samples have to osculate in the frequency domain, we have to designate positive and negative frequency regions (or banks) for each cepstral filter. Also the polynomial function has to be construable in the frequency domain. For example, Gaussian, Chebyshev, and Sinc windows meet the criteria. In other words, the cosine function can be approximated by its first $M$ exponential terms as expressed in Equation 3.1. Figure 3.1 demonstrates that there is little difference between the cosine function and its approximation when the independent variable is in the range $(0.5, M)$. The approximation was achieved using a sum of an alternating sign sequence of Gaussian functions that are linearly spaced along the independent axis between 0.5 and 3. The Gaussian frequency responses are available, and the frequency shift is possible by convolution. Therefore, the spectral cosine filter is feasible by a windowed set of filters.

$$\cos(2\pi \lambda_0 f) \approx \sum_{m=1}^{M} e^{-\left(f-(m\cdot f_0)\right)^2/\sigma^2} - e^{-\left(f-(m-0.5\cdot f_0)\right)^2/\sigma^2}$$ (3.1)

In Equation 3.2, a quefrency filter is defined by a convolution between the Gaussian frequency response of the window and an alternating sign impulse train, where an impulse function ($\delta$) is the frequency response of a temporal cosine function. Figure 3.2 illustrates an instance of the convolution when $\tau \in [0, 2.25]$. For example, the value at 2.25 is zero since it is midway between the in-window positive and negative impulses. The outcome of the convolution assembles a quefrency filter, which is a spectral cosine function. The quefrency filter is thus realized by finding the variance of the window and the frequency spacing between the alternating sequence of impulses of the temporal cosine filters. Another way to look at it is that the positive impulses are for measuring the existence of a quefrency...
and that the negative impulses are for measuring the ambient or non-existence of a quefrency. The distance between each of the two adjacent positive impulses is $f_0$. The quefrency model is repeated $N$ times for an $N$-order cepstrum.

To recap, SQT consists of quefrency filters that transform a speech signal from the time domain to the quefrency domain, generating the cepstrogram. The SQT model is sound theoretically. The section presents the quefrency concept and supports it by existing theories. The quefrency transform is explainable by the Fourier properties when they are applied twice: on the time and on the frequency domains, mainly with the Stone-Weierstrass and Fourier Convolution theorems. The proof was conducted using the Gaussian window since its exponent terms can be easily noticed in the cosine exponential approximation, but proof
could have been conducted by other windows as revealed in the following sections.

\[ q_{f_0}(f) = \int e^{-(f-\tau)^2/\sigma^2} \sum_{m=1}^{M} \delta(\tau - m \cdot f_0) - \delta(\tau - (m - 0.5) \cdot f_0) \, d\tau \]  

(3.2)

3.2 Unit

The rate of acceleration is what the cepstrum measures since its independent axis, quefrency, is the change in frequency (Hz) per time (\(\mu s\)). In other words, the cepstrum measures the acceleration of the cycles. The proof that quefrency is the rate of change can be derived directly from the relevant definitions. Let the scalar quantities \(\lambda_0\), second, and second' be sample measurements of the corresponding units: cycle, s, and s', in a \(\sigma\) time unit.

Then, obviously, quefrency is \(\text{second} \frac{\text{second}}{\sigma / \lambda_0}\) (in Hz-per-s' unit) since it is the frequency of frequencies, since frequency is the rate of cycle per the standardized second per definition. Hence, it is the rate of occurrence with respect to time, and since frequency also is the ratio of the standardized second to the comparable \(\lambda_0\) interval, hence, \(\frac{\text{second}}{\lambda_0}\) (in cycles per second or Hz unit). That is, quefrency is the rate of (wave-period per the standardized second) per the other constant second'. Intuitively, the unit of the quefrency is cycles per second squared when \(\text{second}' = \text{second}\). For convenience, driving distances are sometimes normalized to mere hours, but not everyone can see the implications. Because units are sometimes normalized by the sampling rate [38]), some people may assert that the cepstrum looks like the time space as it is commonly used for lossy compression, and thus its unit must be seconds, but this is not necessarily true. Some spaces retain similar characteristics; however, they have different units. For example, the second derivative of \(f(x) = x^3\) is monotonically increasing just like the original function, yet \(f(x)\) and \(f''(x)\) have different units. However, if the unit of \(f(x)\) is a meter and the unit of \(x\) is a second, then the unit of \(f''(x)\) is meters per second squared, which is a unit of acceleration. This completes the proof. The following elaborations reflect
on the conclusion.

Rates are usually averaged by the harmonic mean formula, from which one can infer the harmonic scale in Equation 3.3. It can be used to scale the frequency axis. In the formula, $f_{\text{min}}$ and $f_{\text{max}}$ are the hyperparameters of the minimum and maximum $f_0$ detection boundaries. ($n \in \{0, 1, \cdots, N - 1, N\}$.)

\[
R(n) = R(n, f_{\text{min}}, f_{\text{max}}) = \frac{1}{\frac{1-n/N}{f_{\text{min}}} + \frac{n/N}{f_{\text{max}}}} \quad \text{(Hz/s')}
\] (3.3)

Consequently, one possible way to space the quefrency axis is by $1/\Delta q = \frac{f_{\text{max}} f_{\text{min}} \cdot N \times 10^{-6}}{(f_{\text{max}} - f_{\text{min}})}$ Hertz per microsecond (or cycle per second per $\mu$s).

Additionally, since

\[
\frac{1}{R_n} (s'/Hz) = \frac{1}{R_{n+1}} (s'/Hz) + \frac{1}{\Delta q} (\mu s/Hz)
\] (3.4)

then, the equivalent temporal unit is

\[
s' = \frac{(f_{\text{max}} - f_{\text{min}}) \cdot 10^6}{f_{\text{max}} \cdot f_{\text{min}} \cdot N} \quad (\mu s)
\] (3.5)

The factor $s'$ and the cepstral order ($N$) are inversely proportional. When the speed of sound is 343 meters per second, the value of $s'$ becomes $(f_{\text{max}} - f_{\text{min}}) \cdot 343 \times 10^2 / (f_{\text{max}} \cdot f_{\text{min}} \cdot N)$ centimeter. The unit of the quefrency can be converted to samples given the sampling rate as cited earlier.
3.3 Scale

Figure 3.3a shows the harmonic scale as defined in the previous section. The quefrency independent variable is spaced harmonically such that $\Delta q = \text{Hz}/(36 \mu s)$. On the other hand, Figure 3.3b shows the regular arithmetic scale, where $\Delta q$ varies from $\text{Hz}/(392 \mu s)$ to $\text{Hz}/(3 \mu s)$. The two cepstrgorams were produced with the same time series, cepstral resolution, and time frames. The harmonic scale appears better than the arithmetic scale for several reasons, but the two are not as practical as the geometric scale.

Like the Mel scales, the harmonic scale bends the frequency axis to provide more cepstral resolution for low pitches at the expense of the high pitches, and this may give a humanizing perception as was argued in related scales.

On the other hand, applying the Mel Scale on the arithmetic extraction after the fact adds uncontrollable noise. For example, the frequencies around 191 Hz/s’ of Figure 3.3b were represented in a few pixels and received low intensity, so they are prone to pixelization and underflow when enlarged. Furthermore, even though the harmonic is better than the arithmetic scale, the harmonic is not practical from the information theory perspective since it designates a large storage space for very deep voices, whose occurrences are rare.

As was stated in the introduction, anti-aliasing attenuates the high frequencies of the utterances, the speech signals have voice depths, and the voice depth is believed to be one of the voice coordinates. Based on the voice statistics [10], the quefrency scale can become statistically representative when it represents the speakers equally. To do that, the frequency may have to be scaled in accordance with the pitch notion. From the relation between the pitch and $f_0$ in Equation 3.6, one can derive the geometric scale.

$$f_0 \equiv \text{pitch (mod height)}$$
$$= \text{pitch} + \text{height} \cdot \text{depth}_{\text{depth}\in\{0.5,1,2,4\}}$$

(3.6)
Figure 3.3: Quefrency Scales

(a) Harmonic Scale

(b) Arithmetic Scale
The formula for the geometric scale that is adjusted to the "A" tone is thus in Equation 3.7. $R$ is an array of fundamental frequencies, and $N$ is the number of quefrency levels. Figure 3.4 compares the five scales. The geometric scale is obviously different from the Mel and Bark Scales. For example, the geometric scale, along with the the harmonic scale, has a sharp curve since it fits the quefrency range, which is set by the minimum and maximum hyperparameters. The geometric scale is better than the other scales because it equalizes the vocal representations of the main voice categories. For example, to represent the [55, 880] Hz pitch range in one byte (or 256 levels), the geometric scale designates 60 levels for the range [100, 200] Hz and 60 levels for the range [200, 400] as shown in the figure. While the Mel and Bark scales are not applicable to SQT because they are defined over the complete frequency range rather than the specific quefrency range, the geometric scale is between the linear and harmonic scales. It is less biased than the other scales, and this is vital since it equalizes the voice ranges, so the machine learning algorithms do not have to normalize cepstral resolutions for two most likely speaker categories.

$$R[n]_{0\leq n<N} = f_{\min} \cdot (f_{\max}/f_{\min})^{n/(N-1)} \text{ (Hz/s') }$$

(3.7)
3.4 Modeling

As mentioned in the Speech Signals chapter (Section 1.1), the speech signal is a random process generated by the hidden states of the vocal tracts and folds, which have physical boundaries that lower the state transition probabilities and thus make the signal presentable in the discrete-time frames. The maximum interval of the window (or the length of the frame) is reciprocally related to the minimal detectable fundamental frequency $f_{\text{min}}$, and the spectral filters have to be windowed in a way that creates the cepstral filter. The optimal number of the sinusoidal cycles ($\gamma$) inside the window depends on the selection of the window type, but generally, the observation of at least one cycle ($\gamma \geq 1$) is necessary for detecting uni-tone signals. Moreover, at least two cycles are essential for measuring the stationarity.
of periodic signals. Setting $\gamma$ to include more than two cycles increases the stationarity measurement per the law of large numbers, but for continuous spectral coverage, the frames have to include exactly four cycles ($\gamma \leq 4$) or less to form the SQT model. Furthermore, small additive leakages can be caused by the non-ideality of the pulse train, so the length in practice may be set slightly shorter than the theoretical one. Therefore, Equation 3.8 is the general formula of the window length ($\lambda_w$). The upper boundary is annotated $f_0/4$ earlier in Figure 3.2 for clarification. Additionally, to convert the window length from seconds to samples, let $c = \lfloor f_s \lambda_w [0]/2 \rfloor$; then the discrete-time, independent variable of the frame is $u \in \{0, 1, \cdots, 2c\} = \mathcal{W}_{[0,2c]}$, and the number of samples in the window is $(2c + 1)$, where the exact length is $2c$ samples. The intervals are easier to track with the odd window lengths in deep layer structures than with even window lengths. When there is an abundance of memory, the windows should not be applied to every frame while they can simply be applied directly to the transform once generated. The window length ($\lambda_w$) plays a crucial part in the definition of the quefrency model, which is based on a speech signal model that can be used to explain the extractions and reconstructions.

$$\lambda_w[n] = \frac{\gamma}{R[n]} \bigg|_{1 \leq \gamma \leq 4}$$

Multiple frequency transforms can form a quefrency transform when they are windowed correctly. For SQT, each set of harmonics has a window length, the minimal output resolution is $M \times N$, and the transform ($T$) maps a $(2c+1)$-sample series (or vector) to an $N$-cepstral $\times M$-spectral matrix; $T : \mathbb{R}_{[-1,+1]}^{2c+1} \rightarrow (\mathbb{R}_{[f_{\min},f_{\max}]}^N, \mathbb{R}_+^M)$. The $M$ and $N$ hyperparameters are in $\mathbb{Z}_+$. The complex SQT version requires twice the size. While the cepstral resolution ($N$) sets the number of the quefrency quantization levels per any scale, the spectral resolution ($M$) sets the harmonic order for the extractions and reconstructions. The wavelength of the $m^{th}$ harmonic component is $\frac{\lambda_0}{2^m}$. For example, the wavelength of the third harmonic term
is 0.25,λ0. Therefore, the high components capture the fine periodic details from the λ0 waveform while the low components capture the more dominant speech feature details.

The glottal signal on the time domain (g[u]) is a pulse train and on the frequency domain (G[U]) is a pulse train. Without loss of generality, consider an ideal case whereby the glottal signal is a unit impulse train. In the frequency domain, the frequency response of the ideal impulse train is also an impulse train, which would carry the harmonic series. As demonstrated earlier in the spectrogram, the parallel lines of G[U] are multiplied by the hidden state series (H[U]) of the tract system. Keep the ideal case in mind. In the non-ideal case, the parallel lines are bold since g[u] consists of imperfect pulses rather than impulses. Additionally, because the waveform periods do not necessarily fit the frames ideally, the window (w_n) causes spectral leakage or thickens the spectral parallel curves. The spectral leakage may tell microscopic details about the speaker such as his or her glottal health and age. In both cases, a phoneme sound is encoded in the contrast between the values of the harmonic series and not the audio volume. Most importantly, when a signal travels through a system, a convolution process (denoted by *) occurs between the signal and the impulse response, as mathematically shown in Equation 3.9. In communications engineering, the system of the vocal channel can be represented by an impulse response function h, which is a time sequence that describes how the output of the channel responds to an impulse stimulus. The system can also be represented by its frequency response function H, which is its Fourier transform. That is, the producer system may be represented by a frequency multiplier that simply places the speech code of the tract on the harmonic series of the folds.

\[
s[u] = \mathcal{F}^{-1}\{ H[U] \cdot G[U] \} = h[u] * g[u] = h[u] * w_n[u] \cdot \sum_{m=1}^{M} \cos(2\pi m f_0 \cdot (u - c)/f_s + \varphi[d])
\]

(3.9)
\[ S[f] = \max_{\varphi} H[f] \cdot \left( W_n[f] \ast \sum_{m \in \mathbb{N}} \delta(f - mf_0) \right) \cdot e^{i\varphi} \]  

(3.11)

\[ s[t] \approx \sum_{m=1}^{M} H_m[t] \cdot \cos(2\pi \cdot (\Phi[m] - [\Phi[m]]) - \pi/2) \]  

(3.12)

\[
\begin{align*}
\Phi[m] &= \frac{m}{f_s} \cdot \sum_{\tau=0}^{t} f_0[\tau] \\
2mf_0[\tau] &\leq f_s
\end{align*}
\]

The frequency-division multiplexing in Equation 3.10 is the extraction speech model. In the context of frequency modulation, the frequency values \( mf_0 \) are called instantaneous frequencies, and their amplitudes are in an \( H_m \) array. For dimensionality expansion, a quefrency transform consisting of a 2\( MN \) matrix filters the positive and negative parts of the harmonic components from the speech signal \( (s[u]) \). The detection can be further enhanced by additional phase channels using the phase shift parameter \( \varphi \). In power engineering, three-phase systems ensure the continuity of the reception, but in communications engineering, the phase synchronizes the receivers with the transmitters. For the speech extractor, the phase is included to track the transmission with the real and imaginary channels, so the number of the phase channels \( (D) \) is two. However, one-phase and four-phase transformers may be used in special cases; thus, the general formula is Equation 3.13.

\[ \varphi[d] \bigg|_{0 \leq d < D} = \frac{\pi d}{2 \cdot D} \]  

(3.13)

To fix the quefrency filter on the spectral domain, the window lengths must be exact multiples of \( 1/R[n] \) since the spectral leakage has to be centered regardless of the phase difference. The length of the frame has to be exactly four times the \( \lambda_0 \) wavelength as in Equation 3.14 to construct the quefrency filters by employing the spectral leakages of the rectangular windows. The Fourier transform of a rectangular function is a sinc function, and the harmonic sinc series converges to a cosine function as depicted in Figure 3.5. As noted
earlier, the negative valued frequencies are off-phase real-valued frequencies. The sinc mixture is defined in Equation 3.15. The approximation of the cosine function using the sinc series is similar to using the exponential terms of the Gaussian mixture depicted in Figure 3.2. Per Equation 3.11, the Fourier transform of the window is convoluted on the frequency domain since the window is multiplied on the time domain. The main-lobe width of the frequency response of the window must correspond to \( f_0 \) (or \( R[n] \)), and the convolution causes controlled spectral leakages while many spectral values are added and subtracted.

\[
\begin{align*}
  w_n[u] &= \begin{cases} 
    1/(f_s \lambda_w[n]), & \text{if } |u - c| \leq f_s \lambda_w[n]/2 \\
    0, & \text{otherwise}
  \end{cases} \\
  \cos(2\pi f_0, f) \bigg|_{0.5 < f_0 f < M} &\approx \sum_{\tau=1}^{2M} (-1)^\tau \cdot \text{sinc}((f - \tau/2) \cdot \lambda_w[n]) \\
  \text{sinc}(x) &= \sin(\pi x)/ (\pi x)
\end{align*}
\] (3.14)  

Instead of leaving the spectral leakage unspecified, the SQT extractor employs it to model the cosine-shaped frequency banks. Each bank samples a frequency, and the collection of the frequency filters constructs the quefrequency filter. The filters that are constructed in this section use the sinc frequency response, which is the window that is used in many topics, such as the sampling theory of the time samples. The section shows that it is suitable also for the frequency samples because the quefrequency filter is basically spectral sampling. Also the rectangular window mitigates phase distortions.
Figure 3.5: Sinc Frequency Banks
3.5 Phase

The frequency responses of the signals and systems are generally complex, but they can be real-valued. For example, the phase response of the rectangular window (Figure 3.5) alternates between 0 and 180 degrees, both of which are on the real axis, so the rectangular window has a real-valued frequency response. When the window filter is applied, the characteristic of the 180-degree-alternating phase is passed to the output response of the convolution. Additionally in the figure, the phase response is linear, which means the frequency components have the same delay. If they were delayed differently, the signal could be distorted by the system, and if the window is centered, as in the example, the derivative of its phase becomes zero. A non-zero derivative (or slope) or generally any linear phase system is equivalent to a delay buffer, but when the phase response is non-linear, the frequency components would have different slopes and so different phase shifts. The distortion happens when the group delay does not equal the phase delay. Understanding the phase or the frame angle is important to avoid the spectral distortion that can delay the frequency components differently and may affect the features’ extractions and reconstructions.

The windowing operator may affect the coordinate system of the sinusoidal extractor rather than the signal. In Figure 3.6, a 360-degree rotation is equivalent to one wavelength of a sinusoidal function. The speech frame vector has a variable magnitude, and it travels counter clock wise in the figure when the third perpendicular axis is the phase coordinate. The signal vector is represented by the real and imaginary parts, so the two-phase extractor suffices while the one- and three-phase extractors may not. For the two-phase extractor, the frame is projected on cosine and sine functions for the duration of the window length before the filters can locate the phase of the signal. In other words, the phase shift may be measured with multiple cycles. The phase measurement is cumulative with a temporal integration variable, so the \( \Re \) and \( \Im \) axes reference the phase of the transform at the initiation moment.
Figure 3.6: Detection Phase

Usually, the frequency components inside the frame are viewed as being affected by the window since the windowing is usually applied first. However, as discussed in the previous section, SQT applies the window to its filters. Even though the two orders of operations are equivalent mathematically because of the convolution properties, it makes sense when it is applied first to the transform because it eventually affects how the components are being captured rather than the actual components. Accordingly, the windowing operator can cause the coordinate systems of the transform to rotate.

The phase of the windowed filter oscillates since it is delayed and expedited. The phase delay happens in a similar manner to the Doppler effect although the two are not exactly the same. In the Doppler effect, the frequency of the signal that is transmitted to and from a mobile object can increase and decrease whether it is moving inward or outward. Similarly in most windows, the signal is amplified in the first half of the window and is reduced in the
second half. Even if the zero-crossing rate is not changed, the spectra of the signal can be aliased or leaked to adjacent frequencies. When both the signal and the audio volume are increasing or decreasing, the phase appears slower and delayed. On the other hand, when the signal has a positive slope while the volume has a negative slope, the phase appears faster and early. For example, in the one-Hz triangular wavelet of Figure 3.7, when the derivatives of the signal and the window have the same sign, the frequency of the signal is mostly shifted up to high frequencies, but when the two are moving in opposite directions, the frequency of the sine function is mostly shifted down to frequencies that are lower than one Hertz.

The spectral response of the triangle window widens the spectral response of the sinusoidal function. The spectral leakage appears symmetric when the number of the enclosed cycles is even. Because the window alternates the shape of the sine function, the function of the filter becomes complex. To mitigate the interference, the filter function may be altered to reverse the group delay effect by adding the missing slope that is removed by the window operator, so its result becomes less complex. By doing so, any signal that has the values supposedly removed by widowing measures less similarly than a signal with the expected magnitude distortion. The phase \( \Phi \) function in the figure was obtained by subtracting the derivative of the sine function \( \frac{d}{dx} \sin(2\pi x) \) from the derivative of the window \( \frac{d}{dx} \sin(2\pi x) \cdot \text{tri}(x - 1) \).

In other words, the windowed filter can be adjusted to reduce the phase distortion.

A non-linear window can become linear directly using its phase response. The phase is proportional to the wavelength, and the phase in each filter is passed at a constant velocity, so the difference can be added or subtracted to fix the real frequency filter to the real axis. The phase of any window can be fixed or normalized by using FFT as in Equation 3.16. The equation returns the most similar window that is feasible and has equal increments and decrements. The new version of the window may be \( 180^\circ \)-flipping and complex, preventing the group delay distortion and minimizing the spectral leakage. The equation simply uses the Fourier multiplication and frequency-shift properties, dividing the time samples by the
Figure 3.7: Aliasing in a 1-Hz Triangular Wavelet
magnitude and subtracting the phase by an equivalent angle to each filter. The complex window is not confined to two-phase filters. For a three-phase filter, the magnitude of the window $|\overline{W_n[u]}|$ would have to be projected into three linearly-angled axes, whose equation was provided. That is, the exact reversal of the group delay of the window is possible although it may require more than two channels per filter.

$$\overline{W_n[u]} = W_n[u] \cdot e^{-j\angle W_n[U]} / |W_n[U]|$$

(3.16)

$$W_n[U] = \mathcal{F}\{W_n[u]\}$$

Speech signals, like any digital signal, are vectors because they reside in an orthogonal space. Because the angular distance equals one wavelength, time or distance may be the third perpendicular axis in Figure 3.6. In the complex space, stationary signals have a constant circular speed, and the harmonic components are accelerated versions of the pitch cycles. When harmonic number one completes one cycle, harmonic number two completes two cycles, and number three, three cycles. The harmonic acceleration of a high voice is greater than the acceleration of a deep voice. A phase is a cycle, and a frequency is the derivative of the phase and thus the speed of cycles per second. Even when the frequency (speed) is multiplied by time and converted to radians ($\omega_0 t$ or $2\pi f_0 t$), the phase becomes a measurement of polar distance — think of it as a distance in miles or meters that is converted to a time span or interval or simply to angular units for convenience. Speech is recognized as a signal once it is in a medium. In other words, because the speed of sound can be assumed constant, the relevant distance is conveniently counted in cycles, but that does not change the fact of the origin characteristics. Similarly, a quefrency is the derivative of the frequency and is the acceleration of the cycles per second. The speed difference between two deep harmonic oscillations is less than the speed difference between two high harmonic oscillations. For example, when the fundamental frequency is 50 Hz, the 50 Hz is thus the
cycle acceleration; that is, the difference between the speed of the cycles at the first and second harmonic frequencies, the second and the third, and so on. Even if a group of frequencies travels at a constant speed, the frequencies may get delayed at different rates to the observing filters. When they do so, the signal gets distorted since the windowing can shift the filtered signal, or equivalently, the observer. The phase responses of several windows are studied similarly. Different window types have different slight advantages when implementing the SQT model: The Gaussian window requires a few observation cycles, the Chebyshev window boosts the convolutions, and the rectangular window is directly compatible with the Parseval Theory. Because of the exact equality equation, quality speech feature reconstructions may be achieved using rectangular windows.
3.6 Quefrency Demodulation

The implementation of the quefrency detection is based on frequency demodulation, sinusoidal frequency banking, and de-multiplexing. They are for applying the window length to approximate the frequency banking given the harmonic order, whose effect on the quefrency filter with rectangular window is shown in Figure 3.8. The figure demonstrates the applicability of the Nyquist theorem in the quefrency domain that two cycles are sufficient for the anti-alias modeling as the harmonic order goes to infinity.

The main concept in the frequency demodulation is that the amplitude that is carried on a high frequency component is transferred to dc by multiplying the signal by a matching frequency and phase filter; hence, \( f_1 = f_2 \) and \( \phi_1 = \phi_2 \) in Equation 3.17. Note that: \( \cos(x) = [\exp(ix) + \exp(-ix)]/2 \), \( \cos(0) = 1 \), and the dc component is the arithmetic mean. Additionally, there are positive and negative frequencies in the two-sided power spectrum. The power or energy that is added or subtracted to or from the positive frequency creates an equal amount of energy pulling the negative frequency toward its opposite direction. For example, when sand is grabbed from one flat ground to another, every mass that is taken out leaves a pit behind. The energies can travel between the two frequency sides, but the total energy in the positive and negative frequency sides are always equal and cancel each other out or sum to zero.

\[
\text{filter} \cdot \text{tone} = 2A \cos(\alpha f_1 + \phi_1) \cdot 2B \cos(\alpha f_2 + \phi_2)
\]
\[
= (A e^{-i\alpha f_1} e^{-i\phi_1} + A e^{i\alpha f_1} e^{i\phi_1})(B e^{-i\alpha f_2} e^{-i\phi_2} + B e^{i\alpha f_2} e^{i\phi_2})
\]
\[
= \left|_{f_1=m,f_2} AB e^{-i\alpha f_1 (1-m)} e^{-i(\phi_1-\phi_2)} + AB e^{i\alpha f_1 (1-m)} e^{i(\phi_1-\phi_2)}
\]
\[
+ AB e^{-i\alpha f_1 (1+m)} e^{-i(\phi_1+\phi_2)} + AB e^{i\alpha f_1 (1+m)} e^{i(\phi_1+\phi_2)}
\]
\[
= 2AB \cos(\alpha f_1(m-1) + \phi_2 - \phi_1) + 2AB \cos(\alpha f_1(m+1) + \phi_2 + \phi_1)
\]
Figure 3.8: The Effect of the Harmonic Order on the Quefrency Approximation: $\gamma = 2$
Accordingly, Equation 3.18 is the baseline formula to add the components of the quefrency model and extract the quefrency given a signal frame at time $t$. However, the main drawback of the baseline formula is that it does not consider the phase shift of the different components and the overtone lookalikes. While model have equal spectral coefficients, implementable filters, like the saw function approximation in Equation 3.20, have variant spectral coefficients, so the angles may have to be aligned.

\[
\begin{align*}
\Re \{ w[u] \} &= \sum_{m=1}^{2M} (-1)^{-1} \cos(\pi mu/c)/(2c + 1) \\
\Im \{ w[u] \} &= \sum_{m=1}^{2M} (-1)^{-1} \sin(\pi mu/c)/(2c + 1)
\end{align*}
\]

PitchTrack: $q[t] = \argmax_{R[n]} \left( \Re \{ w[u] \} \cdot s[u] \right)^2 + \left( \Im \{ w[u] \} \cdot s[u] \right)^2$ \hspace{1cm} (3.18)

\[
M \leq \left\lfloor 0.5R[n]/f_s \right\rfloor, \ c = \left\lfloor f_s/R[n] \right\rfloor, \ \text{and} \ 0 \leq u \leq 2c
\]

The angles could be aligned using the geometric tangent (tan) function since it can regulate the phase response of its filtered signal. For demonstrating the underlying mathematics of the relevant implementations, a new window operator is defined in this section and is elaborated on in the next section. Maybe appearing as a shortcut, the angular window (AngWin) is based on the tan function. Half of the tangent and the squared secant functions are implementable as in Figure 3.9, noting that $\tan(x) = \sin(x)/\cos(x)$ and that $\frac{d}{dx} \tan(x) = \sec^2(x)$. The AngWin definition and depiction are respectively in Equation 3.19 and Figure 3.10. The tangent window can be used to perform sharp quefrency band-pass filters; however, the
windows have poles and thus require attentive implementation.

\[ w_{tan}[u] = \tan(0.5\pi \cdot (u - 0.5)/c)/(2c + 1) \]

\[ w_{sec^2}[u] = \sec^2(0.5\pi \cdot (u - 0.5)/c)/(2c + 1) \]

AngWin\[u\] : \(-\tan(0.5\pi f_0 \cdot (u + 0.5)/c)\cdot \\sin(2\pi\psi f_0 \cdot (u + 0.5)/(2c))/(2c + 1) \right|_{c=[f_s/f_0]} \approx (\cos(2\pi f_0 t) - 2\cos(\pi f_0 t) + 1)/\alpha \]

\[ Saw(u) = 0.5 - (u/c \ (mod\ 1)) \]

\[ \approx 0.1\pi \sum_{m=1}^{M} \sin(2\pi mu/c)/m \] (3.20)

The angular window (AngWin) is the special case of the multiplication of the sine and tangent functions when the window length is exactly two cycles ($\lambda_w = 2$). The window has two non-zero frequency components; its amplitude at $f_0$ is half its amplitude at $0.5f_0$; and the phase of its $0.5f_0$ component is shifted by 180 degrees, hence, the negative sign. The general description of the window is that it has very smooth edges, which rapidly go to and return from the other side of infinity, and it can be repeated as long as it has a near-zero mean. The first half is to shift the negative part down to -90 degrees, and the second half is to shift the positive part up to 90 degrees, or vice versa. In the AngWin case, the window has one pole roughly at the center.

Infinity exists, but most hardware cannot handle it. AngWin may be handy because it is easier to center one pole than multiple poles. Combining the filter system with FFT may become unstable if the size of the window is even, even though its output is real. The infinity pole from the tan function is multiplied by the zero value of the sine function. Not only in theoretical closed-form solutions, but infinity may also have to be centered between two sampling indices, so the system does not overflow. The SQT filtration using AngWin is
Figure 3.9: Discrete Derivative Functions
Figure 3.10: Angular Window

\[
\text{AngWin: } \frac{\cos(2\pi t) - 2 \cos(\pi t) + 1}{\alpha}
\]
done by the operations: magnitude FFT AngWin. As is illustrated in the figure, the window alternates between the positive and negative imaginary axes. The positive and negative frequencies are at 90 and -90 degrees as explained earlier. The window may measure the value of two SQT frequency parts, so its filtering may be equivalent to \( N \) FFT operations. Twenty to fifty quefrency features could be suitable to start with.

### 3.7 Angular Window

The overall perspectives to crystallize the SQT methodology were provided in the previous sections, and they can be applied preferably by the multi-dimensional SQT implementation using the next section, which is on page 67. There are further details to present before we recommend an SQT implementation for the ASR front-end functionalities. Most readers may want to skip the section which briefs the AngWin alternative implementation.

For the AngWin window, we derived Equation 3.24 for the case when there is a detectable harmonic tone without loss of generality. (Note \( \sin(x) = [\exp(ix) - \exp(-ix)] / (2i) \) and \( \tan(x) = -i[\exp(ix) - \exp(-ix)] / [\exp(ix) + \exp(-ix)] \)). The proof concludes with a sequence formula that predicts the pattern of the frequency demodulation:

\[
\text{Tangent} \cdot \text{Cosine} \cdot \text{Tone} =
\]

\[
i \tan(\pi f_1) \cdot 2A \cos(2\pi f_1 + \phi_1) \cdot 2B \cos(2\pi f_2 + \phi_2) =
\]

\[
( e^{i\pi f_1} [e^{i\pi f_1} + e^{-i\pi f_1}]^{-1} - e^{-i\pi f_1} [e^{i\pi f_1} + e^{-i\pi f_1}]^{-1} )
\]

\[
( A e^{-i2\pi f_1} e^{-i\phi_1} + A e^{i2\pi f_1} e^{i\phi_1} )
\]

\[
( B e^{-i2\pi f_2} e^{-i\phi_2} + B e^{i2\pi f_2} e^{i\phi_2} )
\]

\[
= ABe^{-i2\pi f_1} e^{-i\phi_1} e^{-i2\pi f_2} e^{-i\phi_2} e^{i\pi f_1} [e^{i\pi f_1} + e^{-i\pi f_1}]^{-1}
\]

\[
+ ABe^{-i2\pi f_1} e^{-i\phi_1} e^{i2\pi f_2} e^{i\phi_2} e^{i2\pi f_1}/2 [e^{i\pi f_1} + e^{-i2\pi f_1}]^{-1}
\]

57
+ \text{AB}e^{2\pi f_1}e^{i\phi_1}e^{-2\pi f_2}e^{-i\phi_2}e^{i\pi f_1}[e^{i\pi f_1} + e^{-i\pi f_1}]^{-1} \\
+ \text{AB}e^{2\pi f_1}e^{i\phi_1}e^{2\pi f_2}e^{i\phi_2}e^{i\pi f_1}[e^{i\pi f_1} + e^{-i\pi f_1}]^{-1} \\
- \text{AB}e^{-2\pi f_1}e^{-i\phi_1}e^{-2\pi f_2}e^{-i\phi_2}e^{-i\pi f_1}[e^{i\pi f_1} + e^{-i\pi f_1}]^{-1} \\
- \text{AB}e^{-2\pi f_1}e^{-i\phi_1}e^{2\pi f_2}e^{i\phi_2}e^{-i\pi f_1}[e^{i\pi f_1} + e^{-i\pi f_1}]^{-1} \\
- \text{AB}e^{2\pi f_1}e^{i\phi_1}e^{-2\pi f_2}e^{-i\phi_2}e^{-i\pi f_1}[e^{i\pi f_1} + e^{-i\pi f_1}]^{-1} \\
- \text{AB}e^{2\pi f_1}e^{i\phi_1}e^{2\pi f_2}e^{i\phi_2}e^{i\pi f_1}[e^{i\pi f_1} + e^{-i\pi f_1}]^{-1} \\
= \left[ a_i \right] \left\{ -\text{AB}(e^{-i\pi f_1} - e^{i\pi f_1})(e^{-i(2\pi f_1+\pi)} + e^{i(2\pi f_1+\pi)}) \\ \\
(e^{-2\pi f_2-i\phi_2} + e^{2\pi f_2+i\phi_2})[e^{-i\pi f_1} + e^{i\pi f_1}]^{-1} \right\} \\
= \text{sec}(\pi f_1)i\text{AB} \\
\left[ -\sin(\pi f_1 - 2\pi f_2 + \phi_1 - \phi_2) - \sin(\pi f_1 + 2\pi f_2 + \phi_1 + \phi_2) \\
+ \sin(3\pi f_1 - 2\pi f_2 + \phi_1 - \phi_2) + \sin(3\pi f_1 + 2\pi f_2 + \phi_1 + \phi_2) \right] \quad (3.22) \\
= \left|_{\phi_1=\phi_2=0, A=-i} \right. B \left[ \sin(\pi f_1 - 2\pi f_2) + \sin(\pi f_1 + 2\pi f_2) \\
- \sin(3\pi f_1 - 2\pi f_2) - \sin(3\pi f_1 + 2\pi f_2) \right] \quad (3.23) \\
= \left|_{f_2=m f_1, m \in \mathbb{R}} \right. 2B\cos(2\pi mf_1)[\sin(\pi f_1) - \sin(3\pi f_1)]
\[
B \left[ \sin(\pi 0 f_1) - \sin(\pi f_1) + \sin(3\pi f_1) - \sin(5\pi f_1) \right], \quad m = 1 \\
B \left[ \sin(\pi f_1) - \sin(3\pi f_1) + \sin(5\pi f_1) - \sin(7\pi f_1) \right], \quad m = 2 \\
B \left[ \sin(3\pi f_1) - \sin(5\pi f_1) + \sin(7\pi f_1) - \sin(9\pi f_1) \right], \quad m = 3 \\
B \left[ \sin(5\pi f_1) - \sin(7\pi f_1) + \sin(9\pi f_1) - \sin(11\pi f_1) \right], \quad m = 4 \\
B \left[ \sin(7\pi f_1) - \sin(9\pi f_1) + \sin(11\pi f_1) - \sin(13\pi f_1) \right], \quad m = 5 \\
\ldots \\
B \left[ \sin(15\pi f_1) - \sin(17\pi f_1) + \sin(19\pi f_1) - \sin(21\pi f_1) \right], \quad m = 9 \\
B \left[ \sin(61\pi f_1) - \sin(63\pi f_1) + \sin(65\pi f_1) - \sin(67\pi f_1) \right], \quad m = 32 \\
\ldots \\
B \left[ \sin(61\pi f_1) - \sin(63\pi f_1) + \sin(65\pi f_1) - \sin(67\pi f_1) \right], \quad m = 32 \\
B \left[ \sin(1991\pi f_1) - \sin(1993\pi f_1) + \sin(1995\pi f_1) - \sin(1997\pi f_1) \right], \quad m = 991 \\
\ldots \\
= B \sum_{n=-2}^{1} (-1)^n \sin(\pi (2m + n + 1)f_1) \quad m > 0 \quad (3.24)
\]

The derivation could have continued from the Equation 3.23 line with an additive assumption \( f_2 = f_1 + b \), which ends in \( 2B(\sin(\pi f_1) - \sin(3\pi f_1))\cos(2\pi (b + f_1)) \), but the additive replacement was not suitable for complex numbers as it would have weakened the constraints. Also the cosine function obviously removed the denominator of the tangent function and affected the angular effect needed by the SQT model, so the modulation needs to shift to an imaginary axis to keep the tan in the equation.

The derivation can resume at the phase line from Equation 3.22 by adding the \(-90\) degrees to make the cosine a sine function:
\[ \tan(\pi f_1) \cdot 2A \sin((2 + 2n)\pi f_1) \cdot 2B \cos(2\pi f_2 + \phi_2) \]

\[ = \sec(\pi f_1)AB \begin{bmatrix} \cos(\pi f_1 - 2\pi f_2 + 2\pi fn - \phi_2) + \cos(\pi f_1 + 2\pi f_2 + 2\pi fn + \phi_2) \\
- \cos(3\pi f_1 - 2\pi f_2 + 2\pi fn - \phi_2) - \cos(3\pi f_1 + 2\pi f_2 + 2\pi fn + \phi_2) \end{bmatrix} \]

\[ = \sec(\pi f_1)AB \begin{bmatrix} \cos((1 + 2n)\pi f_1 - (2m\pi f_1 + \phi_2)) + \cos((1 + 2n)\pi f_1 + (2m\pi f_1 + \phi_2)) \\
- \cos((3 + 2n)\pi f_1 - (2m\pi f_1 + \phi_2)) - \cos((3 + 2n)\pi f_1 + (2m\pi f_1 + \phi_2)) \end{bmatrix} \]

\[ = \sec(\pi f_1)AB \begin{bmatrix} 2\cos((2n + 1)\pi f_1)\cos(2m\pi f_1 + \phi_2) - 2\cos((2n + 3)\pi f_1)\cos(2m\pi f_1 + \phi_2) \end{bmatrix} \]

\[ = \sec(\pi f_1)AB \cdot \cos(2m\pi f_1 + \phi_2) \begin{bmatrix} 2\cos(-\pi f_1) - 2\cos(\pi f_1) \end{bmatrix} = 0 \]

\[ \therefore -\tan(\pi f_0) \cdot \sin(n\pi f_0) \cdot 4B \cos(2\pi f_0 + \phi) = \Bsec(\pi f_0) = \begin{bmatrix} \cos(2\pi f_0(m - [0.5 + \psi]) + \phi) + \cos(2\pi f_0(m + [0.5 + \psi]) + \phi) \\
- \cos(2\pi f_0(m - [0.5 - \psi]) + \phi) - \cos(2\pi f_0(m + [0.5 - \psi]) + \phi) \end{bmatrix} \]

Equation 3.25

A visual description of the demodulation function is depicted in Figure 3.11. Equation 3.25 can safely be used when \( f_0 \) is a positive integer or \( n \) is even. The sine function stabilized the tan function by constraining the poles of the tan function. There is a region of instability because of the variable shift by the variable phase (\( \phi \)), so it is mostly stable when \( n \) is even or \( f_0 \) is in \([\mathbb{N} - 1/5, \mathbb{N} + 1/5]\), but the zeros are exactly at \( f_0 \in \mathbb{N} \), to which the \( f_0 \) selection must be restricted to make the system less variant on the phase diffidence or synchronization.

In Equation 3.25, \( \pi \) was found numerically to be one of the optimal values of \( n \) to
Figure 3.11: AngWin Speech Demodulation Space
maximize the spectral demodulation to the dc component, but the proof continues with the rational numbers to approximate the expected result since π is not rational and may destabilize the system. Also the fractions of π make the roots hard to see, which is not optimal for simplification. Recall that the energy of the positive frequency part is at \( m > 0 \) while the energy of the negative frequency part is at \( m < 0 \) and that \( m \) and \( f_0 \) are integer numbers. The frequency roots are obtained from the AngWin derivation, and the detailed steps are in Equation 3.26. Therefore, the cumulative sum of the energies of the frequency multiplicatives \( (mf_0) \) are adding at \( m = \pm[\psi + 0.5] \) and subtracting at \( m = \pm[\psi - 0.5] \). Although the \( m \) values may become fractions and so not affect the dc component directly, the window makes a cumulative sum at the dc component, so the dc component of the speech signal may have to be filtered before its multiplication with the window. The components of the selected \( f_0 \) are going to be filtered by bandpassed then added and subtracted to and from the cumulation. For a \( kf_1 \) frequency set with a \( B \) magnitude of energy at the \( k = 3 \) term, another set of roots is cascaded from the previous roots. The detectable multiplicative values of \( f_1 \) are outlined, as in the previous detailed steps, in Equation 3.27.

Frequency Roots of the Positive Terms:

\[
\begin{align*}
    m - [0.5 + \psi] &= 0 & \iff & & m = \psi + 0.5 \\
    m + [0.5 + \psi] &= 0 & \iff & & m = -\psi - 0.5
\end{align*}
\]

Frequency Roots of the Negative Terms:

\[
\begin{align*}
    m - [0.5 - \psi] &= 0 & \iff & & m = -\psi + 0.5 \\
    m + [0.5 - \psi] &= 0 & \iff & & m = \psi - 0.5
\end{align*}
\]

(3.26)
Cascaded Roots of the Positive Terms:

\[ k - [0.5 + \psi] = -\psi - 0.5 \quad \Leftrightarrow \quad k = 0 \]
\[ k - [0.5 + \psi] = -\psi + 0.5 \quad \Leftrightarrow \quad k = 1 \]
\[ k - [0.5 + \psi] = \psi + 0.5 \quad \Leftrightarrow \quad k = 2\psi + 1 \]
\[ k - [0.5 + \psi] = \psi - 0.5 \quad \Leftrightarrow \quad k = 2\psi \]
\[ k + [0.5 + \psi] = \psi + 0.5 \quad \Leftrightarrow \quad k = 0 \]
\[ k + [0.5 + \psi] = \psi - 0.5 \quad \Leftrightarrow \quad k = -1 \]
\[ k + [0.5 + \psi] = -\psi - 0.5 \quad \Leftrightarrow \quad k = -2\psi - 1 \]
\[ k + [0.5 + \psi] = -\psi + 0.5 \quad \Leftrightarrow \quad k = -2\psi \]

Cascaded Roots of the Negative Terms:

\[ k - [0.5 - \psi] = \psi - 0.5 \quad \Leftrightarrow \quad k = 0 \]
\[ k - [0.5 - \psi] = \psi + 0.5 \quad \Leftrightarrow \quad k = 1 \]
\[ k - [0.5 - \psi] = -\psi - 0.5 \quad \Leftrightarrow \quad k = -2\psi \]
\[ k - [0.5 - \psi] = -\psi + 0.5 \quad \Leftrightarrow \quad k = -2\psi + 1 \]
\[ k + [0.5 - \psi] = -\psi + 0.5 \quad \Leftrightarrow \quad k = 0 \]
\[ k + [0.5 - \psi] = -\psi - 0.5 \quad \Leftrightarrow \quad k = -1 \]
\[ k + [0.5 - \psi] = \psi + 0.5 \quad \Leftrightarrow \quad k = 2\psi \]
\[ k + [0.5 - \psi] = \psi - 0.5 \quad \Leftrightarrow \quad k = 2\psi - 1 \quad (3.27) \]

Although most of the new added and subtracted terms summed to zero, four terms remained from the result. When \( \psi = 2 \), they are \( n \) (or \( 2\psi \)) apart from \( f_0 \) as depicted in Figure 3.12. On the positive frequency axis, the terms of the summation translate to real
vectors at $(2\psi + 1)f_0$ and $(2\psi - 1)f_0$. In the case of $n = 2$, the spectral cumulation can add only half the components whose harmonic numbers are odd, as expanded in Table 3.1. Luckily, the frequency axis can shift, so the formula can be simplified and approximated as in Equation 3.28. Its spectrum can be approximated by an even square wave, whose imaginary part is a version of its real part time-shifted by half the squares’ widths.

\[
AF_{odd}[u] = 0.5\tan(2\gamma \pi u/c) \sin(4\gamma \pi u/c) + j 0.5\tan(2\gamma \pi u/c + \pi/4) \sin(4\gamma \pi u/c + \pi/2)
\]

\[
= \sin^2(2\gamma \pi f_0 u/c) + j \sin^2(2\gamma \pi f_0 u/c + \pi/4) \quad u \in [0, \gamma f_0/f_s]
\]

\[
AF_{even}[u] = |AF_{odd}[u]| \cdot \left[ \cos(4\gamma \pi u/c) + j \sin(4\gamma \pi u/c) \right]
\]

\[
= \sin^2(2\gamma \pi u/c) \cdot e^{j4\gamma \pi u/c} \quad u \in [0, \gamma f_0/f_s]
\]

\[
\approx (1 + \sin(4\gamma \pi u/c) + j(1 + \cos(4\gamma \pi u/c))
\] (3.28)

As in the table, the filter equation adds the odd half two times the even half when $n = 1 = 2\psi$, and the sequence may be added with some noise when $n = 0.5 = 2\psi$. A similar behavior can be observed by the saw function, which can be approximated by a weighed harmonic average, as in Equation 3.20. The result verifies that it is necessary to add the
Table 3.1: The Components of the Cumulative Demodulation

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\psi$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>$2\psi - 1$</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>13</td>
<td>15</td>
<td>17</td>
<td>19</td>
</tr>
<tr>
<td>$2\psi + 1$</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>13</td>
<td>15</td>
<td>17</td>
<td>19</td>
<td>21</td>
</tr>
<tr>
<td>$\psi$</td>
<td>0.5</td>
<td>1</td>
<td>1.5</td>
<td>2</td>
<td>2.5</td>
<td>3</td>
<td>3.5</td>
<td>4</td>
<td>4.5</td>
<td>5</td>
</tr>
<tr>
<td>$2\psi - 1$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>$2\psi + 1$</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>13</td>
<td>15</td>
<td>17</td>
<td>19</td>
</tr>
<tr>
<td>$\psi$</td>
<td>0.25</td>
<td>0.5</td>
<td>0.75</td>
<td>1</td>
<td>1.25</td>
<td>1.5</td>
<td>1.75</td>
<td>2</td>
<td>2.25</td>
<td>2.5</td>
</tr>
<tr>
<td>$2\psi - 1$</td>
<td>-0.5</td>
<td>0</td>
<td>0.5</td>
<td>1</td>
<td>1.5</td>
<td>2</td>
<td>2.5</td>
<td>3</td>
<td>3.5</td>
<td>4</td>
</tr>
<tr>
<td>$2\psi + 1$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

$0.5f_0$ when measuring $f_0$. Including the frequency samples between the harmonic samples is necessary to fix the sign of the harmonic components in the spectral summation. The sharp edges of the saw function can be smoothed by the sign function: $\text{sgn}(\text{Saw}[u]) \cdot |\text{Saw}[u]|^{0.5}$.

Thus, the ideal window shape for the SQT transform is a sinc function, but since the speech signal has a random phase and the sinc time window is narrow, the synchronization of the sinc involves convolution to accommodate a phase per time sample. The number of harmonic samples ($M$) controls the width (or variance) of the sinc function, so the number of the independent axes can be less than the number of time samples. For example, when the convolution step is $0.5/M$, the number of matrix multiplications is about four times the required harmonic order as in Equation 3.29, which is the two-dimensional formula for a $4M$-phase quefrency filter. Similar formulas can be derived to measure the cycle accelerations by a complex square train window, the convolutions of multiple tangent windows ($\tan(\alpha t)$ functions) with the first derivative of the signal ($\alpha s'[u] \approx s[u] - s[u - 1]$), and by a distant measure of multiple squared secant windows ($\sec^2(\alpha t)$ functions) with the second derivative of the signal($\beta s''[u] \approx s'[u] - s'[u - 1]$). The tan and sec types have impulses,
and aligning multiple pulses is not an easy task; however, one cycle can be generated and repeated or concatenated for multiple-cycle windows. However, the methods that computed the harmonic energies collectively appeared computationally as complex as measuring the individual harmonic components separately, but one of the two options comes with extra features.

\[
SQT_{2D}^2[n] = \frac{1}{M^2} \sum_{i=0}^{4M} \left( s[u] \cdot sinc\left( \gamma M \pi \left( \left( u - 0.5i/M \right) \mod c \right) - c \right)/c \right)^2 \quad (3.29)
\]

\[
| c = \left[ 0.5 \gamma f_s/R[n] \right] \]

In conclusion, complexity is needed to have the spectral harmonic energies counted equally. The algorithms in the next chapter not only have several practical applications, but they are for identifying the post processes that would have to be included in any SQT filter design. When we define the basic elements of the process and make an implementation that provides high quality and high volume data for machine learning, perhaps an optimal formula may be derived. At this time, devices with Graphical Processing Units (GPUs) are becoming commonplace, and parallel processing is mostly available in the devices that are capable of machine learning. There is no doubt that measuring the harmonic components individually yields the highest quality analysis.
3.8 Multidimensional Quefrency Transform

The SQT filters make the frequency bands of noise signals cancel out while the bands of specific signals add up or down, but the added signals are not always the target speech signals. Figures 3.13 and 3.14 summarize the key negative and positive cases of the SQT model. The absolute means of the filtered signals in the negative cases are roughly zeros while in the positive cases are non-zeros. For example, both the white noise and the undertone signal are negative cases because their negative parts negate their positive parts. On the other hand, the mean values of the resonant tones whose quefrency matches the SQT filter are large during the voice activity events. Because the voice of a few speakers is shifted, the large mean values can be negative sometimes. The first three cases were either true negative or true positive, but the last one is false negative because the overtones partially share similar components with the quefrency of the filter. The section shows how to filter out the false negative lookalikes from the cepstrum space using the dimensionality expansion of the multi-dimensional SQT implementation.

The speech dimensionality expansion is achieved by extracting the individual harmonic components, in which the speech signals and the lookalikes are separable, and the finely detailed speech space of the expansive transform can produce pitch tracks that are smoothed naturally. The expanded space also produces extra information that can be used for several by-products: signal reproduction, emotion detection, and multi-pitch tracking. In fact, the highest quality quefrency detection was achieved by the dimension expansion since the magnitude and direction of each component vector are accounted to identify the random phase of the noise signals. A time frame is usually represented as a column vector \( \overline{z} \in \mathbb{R}^{2c+1} \). Mapping it to \( \mathbb{R}^{2MND} \), which is considered a four dimensional speech space \( \Omega[k, l, m, n] \), the formula of the speech quefrency transform (\( T_{SQT} \)) is defined in Equation 3.30 — for the four dimensions such that the phase shift index \( k \in [0, D) \), the index of the positive and negative
Figure 3.13: SQT Negative Cases
Figure 3.14: SQT Positive Cases
parts \( l \in [0, 2) \), the harmonic number \( m \in [0, M) \), and the quefrency index \( n \in [0, N) \).

Figure 3.15 illustrates the partitioned matrix transpose of the real and imaginary part filters for a 55 Hz fundamental frequency and its first seven harmonic components. Once the speech frames are transformed to the quefrency space, the pitch track is found when the projected quantities are: map reduced, convolved, and argumented by the maximum.

\[
\Omega[k, l, m, n] = \sum_{u} T_{SQT}[u, v] \cdot (s[u] - \mu_s)/\sigma_s
\]

\[
T_{SQT}[u, v] = \begin{cases} 
\cos(\pi k/D + 2\pi (u - c) f_i)/(2c + 1), & \text{if } f_i \leq 0.5 \text{ and } |u - c| \leq c \\
0, & \text{otherwise}
\end{cases}
\]

\[
k = \lfloor v/N/M/2 \rfloor \mod D, \quad f_i = (m - 0.5l + 1) R[n]/f_s
\]

\[
l = \lfloor v/N/M \rfloor \mod 2, \quad m = \lfloor v/N \rfloor \mod M, \quad n = v \mod N,
\]

\[
v \in \{0, 1, \cdots, 2MNfD - 1\}, \quad u \in \{0, 1, \cdots, 2c\}, \quad \text{and } c = 0.5\gamma f_s/R[0]
\]

The complex cepstrum (C) can be obtained by summing the harmonic vectors of the complex spectra \( \Psi \) as in Equations 3.31 and 3.32. The equation applies the Euclidean norm and the Arc-Tangent function on the projected quantities \( \Omega \), which is determined by subtracting the negative parts from the neighboring positive parts as shown in Figure 3.16. These processes can be done in one step in parallel by map-reduce as illustrated in Figure 3.17. From this point, a few steps can be followed to remove the noise.
Figure 3.15: Componentwise Quefrency Filter
\[ C[n] = \sum_{m=0}^{M-1} \Psi[m,n] \]  

\[ |\Psi[m,n]|^2 = \Omega^2[0, 0, m, n] - \Omega^2[0, 1, m, n] + \Omega^2[1, 0, m, n] - \Omega^2[1, 1, m, n] \]

\[ \angle \Psi[m,n] = \begin{cases} 
\tan^{-1}(\Omega[1, 0, m, n]/\Omega[0, 0, m, n]), & \text{if } \Omega[0, 0, m, n] \geq 0 \\
\pi - \tan^{-1}(\Omega[1, 0, m, n]/|\Omega[0, 0, m, n]|), & \text{otherwise} 
\end{cases} \]  

The magnitudes of the spectral vectors are for the observations of the harmonic components, and the angles are for checking the source of the components. The occurrence of two harmonic components with high magnitudes and low phase difference increases the probability that they are from the same source. In particular, the spectrum of the noise may be constant in magnitude, as in the white noise case, but random in phase. Meanwhile the adjacent components of the human speech are not random but actually correlated. Accordingly, Equation 3.33 introduces the cepstral similarity (or negative distance) measurement \( D \), which is based on the two joint probability estimations of the magnitudes and the angles. The cepstral similarity generates a filtered cepstrum that can be used for the pitch track extraction. The filtering (or liftering) is basically a size-two mean convolution operator applied along the harmonic dimension. The multiplication between the adjacent components decreases the similarity score of the overtones because the overtones tend to have low magnitude values in their sequences. Also the angle differences can be applied to the local adjacency level and not to the whole set of elements at once because there may be a slight drift in the phase sequence, which may have a large phase gap between the angles of the first and the last components. Once the pitch track is extracted, it is used for the spectral feature selection.

\[ D[n] = D^2_{\text{mag}}[n] \cdot D_{\text{ang}}[n] \]  

72
On-Phase Components

Off-Phase Components

Arithmetic Result: Real Part

Imaginary Part

Figure 3.16: Quefrency Filter: $\gamma = 4$
Finally, the quefrency index is the index of the harmonic set that maximizes the cepstral similarity, so the spectral features of the vividest voice are selected using the argument of the maximum. Two indices can be added as in Equation 3.34. The raw spectral matrix can then be sliced by the $q$ index to obtain the feature representation of the tract hidden state $H_m$, which can be a real vector as in Equation 3.35 or a complex vector in the general case. Further processing might be added but is not necessary since the presented equations capture the minimally needed features for the feature reconstruction. For streaming, an unsigned 8-bit integer (uint8), which is one byte or an ASCII character, per harmonic encoding is possible when the input frame is normalized by the moving average of the extracted energy ($\sigma_z$) to produce the coded frame as in Equation 3.36 where 256 is the byte number of quantization levels. The bitrate of the output depends on the bit depth or the value of eight for one byte, the harmonic order $M$, and the frame rate $R_s$, which can be 30 but is usually between 24 and
120 frame per second. The quefrency index \( q \) can be stored in less than eight bits. At the receiver, the scale \( R \) is stored to decode the quefrency indices back to their corresponding values of the fundamental frequency for the voice reproduction.

\[
q_t = \text{argmax}_n ( D[n] + D[n + 1] ) 
\]

\[
H_m[t] = \alpha \sqrt{\Omega^2[0, 0, m, q_t] + \Omega^2[1, 0, m, q_t] + (1 - \alpha) \sqrt{\Omega^2[0, 0, m, q_t + 1] + \Omega^2[1, 0, m, q_t + 1]}}
\]

\[
F_0[t] = \alpha R[q_t] + (1 - \alpha) R[q_t + 1]
\]

\[
\alpha = D[q_t] / (D[q_t] + D[q_t + 1])
\]

\[
\text{Code}[t] = \text{uint8} \left\lfloor \frac{q_t}{255 \cdot H_m^2[t]} \right\rfloor
\]

\[
\text{bit-rate} = \text{bit-depth} \cdot R_s \cdot (M + 1)
\]

It is worth mentioning that the window length cannot be increased simplistically since the number of cycles within the window is maximally four, as shown in Figure 3.18, but the window length is sometimes too short to detect a waning voice with a slow frame rate due to the small sample size. Nonetheless, the spectrum of multiple rectangular windows can be combined. For example, the multiple of four cycles can be realized by averaging the spectra of correctly sized and concatenated windows. Figure 3.19 shows the process of dividing a large window into proper rectangular windows for increasing the collective window length beyond four times the wavelength using the Fourier time-shift property. The exact implementation of the large windows is possible by partitioning a sparse matrix such that a set of adjacent windows is designated per quefrency filter. Alternatively, the extraction of the frame can be averaged with the adjacent frames. The weighted average of two or three frames’ spectra is sufficient. Equivalently, the moving average, like in Equation 3.37, is an alternative workaround that may simplify the expressions. The angles in the equation...
are included for the compatibility with the angular distance, but the magnitude distance is competent.

\[
\Omega_t[k, l, m, n] = \alpha \Omega_{t-1}[k, l, m, n] \exp\{-j\pi (m - 0.5l)\} \\
+ (1 - \alpha) \Omega[k, l, m, n] \exp\{j\pi(m - 0.5l)\}
\] (3.37)

The section explains the multidimensional SQT methodology for extracting the speech features from a time frame. Because of its geometrical scale, the computations may have common factors and so may be optimized as in the FFT algorithm, and the processing of multiple frames can be in sequence or parallel. The bitrate summary of selected extraction settings is printed in Table 3.2. The speech signal representation of the expanded feature space \(\Omega\) can be useful in several machine learning applications. For example, the naturally fine pitch track may capture the voice emotions and characteristics. Additionally, the number of the voices can be traced in the \(D[n]\) space, and consequently, when the cepstral similarity is coupled with a microphone set, there is a possible market for tracing the features of multi-speaker speech signals. Furthermore, the extractor is attuned to the human speech and so can be used for distant speech recognition. As is seen later, the extracted spectral features generate responsive spectrograms and so are normalized by the speakers’ characteristics to a degree. The next section concludes the chapter by reconstructing the extracted stream of features to produce a speech signal that is linguistically similar to the original human speech.
Figure 3.18: The Effect of the Number of Cycles on Rectangular Windows: $M = 4$
Figure 3.19: Adjustment of the Additive Spectrum of Multiple Rectangular Windows
Table 3.2: Bitrates of Raw Features by Frame Rate and Harmonic Order

<table>
<thead>
<tr>
<th>Bit Rate (bps)</th>
<th>Bit Depth</th>
<th>Frame Rate</th>
<th>Harmonic Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,536</td>
<td></td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>3,072</td>
<td></td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4,608</td>
<td>8</td>
<td>23</td>
<td>15</td>
</tr>
<tr>
<td>6,144</td>
<td></td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>7,680</td>
<td></td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>1,920</td>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>3,840</td>
<td>15</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>5,760</td>
<td></td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>7,680</td>
<td></td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>9,600</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3,840</td>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>7,680</td>
<td></td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>11,520</td>
<td></td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>15,360</td>
<td></td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>19,200</td>
<td></td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>7,680</td>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>15,360</td>
<td></td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>23,040</td>
<td></td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>30,720</td>
<td></td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>38,400</td>
<td></td>
<td>39</td>
<td></td>
</tr>
</tbody>
</table>
3.9 Speech Reproduction

In the previous section, Figure 3.18 illustrates that multiple windows may have to be applied separately in order to fix the frame rate and the window size. In other words, when the frame rate is constant, high-voiced signals may require more frames since the number of the window splits is proportional to the quefrency value. Therefore, making the overlapping percentage a constant could increase the computational complexity linearly by the scale as the number of the windows in the quefrency assembly varies between $R_s \cdot f_{\text{min}}$ and $R_s \cdot f_{\text{max}}$. However, the quefrency theory works comfortably even when its transform is implemented partially. Additionally, the window overlapping is variable when the number of the joint windows is fixed. Figure 3.20 illustrates the variable overlapping percentage and the samples’ reconstruction for the zero-split window feature selection and extraction. For example, the fourth sample in the third frame ($\hat{s}[t=2, u=3]$) would be mapped to the value at $vt + u$ on the reconstruction domain. The extracted speech frames can generally be synthesized by Equation 3.38 whether the overlapping is constant or variable.

Equation 3.38 can be used to interpolate the time samples linearly between two speech frames. For convenience, the frame rate ($R_s$) is fixed assuming a persistent communication bandwidth. The synthesis process transfers the representation of the speech frames, which can be generated by ML, back to the representation of the time samples. The new sampling rate does not have to match the original sampling rate; the channel has the capacity for the harmonic components as long as the individual frequencies are less than half the sampling rate. Because of the constant frame rate, the length of the audio chunk between two frames is $1/R_s$ seconds. In the equation, the time frame and time sample indices are respectively denoted by the discrete independent variables $t$ and $u$. The summation of the harmonic signals is simply the estimation of the original signal, $h_m$ is the amplitude of the $m^{th}$ component, and $p_m$ is the phase in cycle unit. We recommend the addition of the $\pi/4$ shift when the
values of the extracted features $H_m[t]$ are not complex, so the envelope of the original signal becomes centered. However, when the extracted features $H_m[t]$ are complex, the quarterly shifted sines are substituted by cosine and sine functions. Moreover, $f_m[t,u]$ is the normalized instantaneous frequency of the $m$th harmonic components at frame number $t$ and sample number $u$. The number of samples in the synthesized frame $v$ is consequently $f_s/R_s$, which may not be an integer, so it is floored or rounded down to the nearest integer, and the remainder of the rounding is accumulated in $\epsilon$ to periodically compensate for the omitted sample fractions. In the equation, both the responsive spectrum ($h_m[t,u]$) and the quefrency value ($f_m[t,u]$) translate linearly from the frame number $t - 1$ to frame number $t$ using the threshold function $a[u]$.

$$
\hat{s}[t,u] = \begin{cases} 
\sum_{m=1}^{M} h_m[t,u] \cdot \sin( 2\pi \cdot p_m[t,u] + \pi/4 ), & \text{if } f_m[t,u] \leq 0.5 \\
0, & \text{otherwise}
\end{cases}
$$
\[ h_m[t, u] = \left( a[u] \cdot H_m[t] + (1 - a[u]) \cdot H_m[t - 1] \right) \cdot 1/\sqrt{2} \]

\[ p_m[t, u] = \left( p_m[t - 1, v] - \lfloor p_m[t - 1, v] \rfloor \right) + \sum_{\tau=0}^{u} f_m[t, \tau] \]

\[ f_m[t, u] = \left( a[u] \cdot f_0[t] + (1 - a[u]) \cdot f_0[t - 1] \right) \cdot m/f_s \]

\[ a[u] = \min(\alpha u/v, 1) \]

\[ u = \{ 0, 1, \cdots, v \} \]

\[ v = \lceil f_s/R_s + \epsilon \rceil - 1 \]

Also the accumulation of the phase is periodically reset (or wrapped) to avoid any unnecessary overheating when applying the modulating formula, so the modulo operator (mod 1) is included in the last sample of the previous frame \( p_m[t - 1, v] \). The phase distance is the integration or the area between the \( f_m \) curve and the time axis, and the transition function does not have to be linear as long as the traveled time distance is added to the value of the accumulation function as in Equation 3.39. The summation is the general case that is sometimes substituted by multiplication when using the step function for the transitioning representation. In other words, the synthesizer has to track the phase state for each active speaker since the phase of the frames must continue from where the phase of the preceding frames left; otherwise, the sharp edges of the discontinuities may add high-frequency or clicking noises.

The quality of the pitch extraction correlates with the hyperparameters. For example, the number of the quantization levels of the bit-depth could cause robotic sound effects. Similarly, the number of the quefrency pixels could cause vinyl effects when the high variance pitch track is interpolated due to steep derivative curves of the pitch track. Therefore, the choice of the hyperparameters controls the quality of the feature selection, extraction, and
reconstruction. For instance, the interpolation can be adjusted by the compression threshold \( \alpha > 1 \). Furthermore, the output quality is also influenced by how the signal is handled. For example, standardizing the variance of the input signal can increase the quality of the extraction and reconstruction of the harmonic components. The speech signal standardizing can be applied based on the voice activity detection because the ratio of the sum of the extracted energies to the total frame energy and the pitch pattern can differentiate between the frames of noise and the frames of voice activity.

In summary, the section elaborated on one of the most crucial SQT engineering applications, which is producing the speech signals from its ML friendly feature space. The formula of the speech feature reconstruction incorporates the harmonic frequencies of the speech streams, whose phase states need to be accumulated during the reproduction. The reconstructed signal is indistinguishable from the original signal when the imaginary part is added to the reconstruction formula [29]. Further processing can accompany the audio synthesizer, so dedicated threads are needed to elevate the user experience. For example, the reconstruction of multiple speech signals may require audio compressing as it is commonly used in speech synthesis and gaming. For Internet streaming, also buffering may be required to handle network errors. Nonetheless, the spectral domain energy mixing of the SQT method is potent, so it is not necessary to use median filters nor combine multiple extraction methods to enhance the pitch track. The proof is shown in the next chapter, which investigates the further results on performance and machine learning applications.
Chapter 4

Results

The previous chapter showed that the quefrency filters can be modeled by several windows. While the minimal required window length was $2\lambda_0$ for Gaussian, and Chebyshev, and rectangular windows, the rectangular windows can be implemented with the $4\lambda_0$, which is a relatively long length. In the previous chapter, we explain that rectangular windows do not meld the spectrum like the other windows. The chapter also shows that SQT has parameters that affect the size of the feature space or resolution. Consequently, there are multiple SQT configurations, each of which may have its own applications since there is always a trade-off between speed and quality and between costs and gains. While the fast extractors are best suited for economical processing, the refined extractors are best suited for analytical processing. This chapter presents the results by three applications: the SQT concept by pitch track extraction, online speech communications by features streaming and reconstruction, and audio emotion detection by machine leaning.
4.1 Proof of Concept

This section is to prove the concept of the SQT methodology to empirically know whether it is practically robust in Central Processing Unit (CPU) environments when its quality parameters are downgraded. Unfortunately, there was not an abundance of reliable labeled data when the experiments were conducted, so the verification test was narrowed to a low-volume dataset known as the Fundamental Frequency Determination Algorithm (FDA) evaluation database [6] although it had missing labels. The labels of the transitioning states to and from the speech segments were also missing in several other datasets. The basic SQT.m is the downgraded version implemented in Matlab with 1.34-second average processing time, which approximately matches the time complexities of similar practical approaches, namely the Matlab implementation of the Pitch Estimation Filter (PEF) [25] with a 3.24-second average processing time and the Matlab implementation of the Normalized Correlation Function (NCF) [4] with a 1.16-second average processing time.

The FDA evaluation database, prepared by an independent source, was labeled when the sampling rate was 20 kHz and included less than six minutes of speech. Since the algorithms were expected to generate the 20 kHz labels from relatively little acoustic evidence, the difficulty of the test was increased by resampling the audio that was given to the algorithms to 8 kHz. The SQT version was one of our early Matlab implementations with the real Gaussian windows, the unfiltered cepstra, the fifth harmonic order \( M = 5 \), and the magnitude distance, so it may have held the bare minimal resolutions, and it was expected to be as good as its counterparts. The test platform was Matlab Online 2019b (9.7.0), and the NCF and PEF implementations were built-in libraries in the official Matlab toolboxes; however, Matlab disclaimed that its PEF implementation may not fully represent of the PEF Amplitude Compression method. The test procedure was outlined in the Matlab documentation [47], which was another independent source. The outputs of some of the methods had outliers,
so a size-three median filter was applied to the outputs. The outputs were also time-shifted to minimize the libraries’ offsets if there were any. However, the two versions were the best open-source available solutions that we were able to find, and so was the database, so the tests were conducted to the best of our knowledge. The error functions of the performance metrics are defined in Equation 4.1.

$$\text{GPE-}\xi = P(|\hat{f}_0 - f_0| > f_0 \cdot \xi/100)$$

$$= \frac{1}{T} \sum_{t=1}^{T} x[t] \quad |x[t]| = \begin{cases} 
1, & \text{if } \frac{|\hat{f}_0[t] - f_0[t]|}{f_0[t]} > \xi/100 \\
0, & \text{otherwise}
\end{cases}$$

(4.1)

$$\text{MSE} = \frac{1}{T} \sum_{t=1}^{T} |\hat{f}_0 - f_0|^2$$

Figure 4.1 compares the error of the three methods on the FDA test. The metric of the Matlab test for evaluating the pitch extractions is the statistical significance of the Gross Pitch Error (GPE) with an observed result value of 20%. The observed values of 10% and 5% are extra to show the performance trends. The results are p-values, which are the probability of the null hypothesis. The null hypothesis is the assumption of false methodologies and are compared with a statistical significance threshold, whose default value is 5%, meaning that the test is significant if the probability of error is unlikely or the p-value is less than the threshold. The common threshold for the pitch extraction application is 10%. The figure shows that the p-value of the three methods was less than 5% when the observed value was 20%, and was less than 10% when the observed value was 10%, but the significance of the three methods was not as expected when the observed value was 5%. However, the figure also shows that the SQT null hypothesis is less likely than the PEF and NCF null hypotheses. So we conclude that the SQT method is as significant as the well established methods if not greater. The difficulty of the task was further increased by adding additive noise. The noise
was generated and mixed also by predefined Matlab functions.

Figures 4.2 and 4.3 repeat the previous test but with two types of additive background audio: white and turbine noises. The tests were repeated with different levels of the signal to noise ratios: 20dB, 10dB, and 0dB, where 0dB is the loudest noise when the ratio is one. For the 10% observed value case, the figures show that the p-value for SQT remained less than 10% under the 20 and 10dB additive noise cases. The same can be said for PEF and NCF but only under the white noise cases and the 20dB turbine environment condition noise. It may be safe to say that while the SQT hypothesis is wrong less than 10% of the times when the signal to noise ratio is greater than 10dB, the PEF and NCF hypotheses are wrong less than 10% of the times when the signal to noise ratio is greater than 20dB. The bar charts in the two figures show that white noise was easier to deal with than the turbine noise.

Figure 4.4 summarizes the Mean Square Error (MSE) results for the seven cases. The tests of significance are widely accepted by scientific communities, but in comparative analyses, the test of significance may not be as important as the MSE metric because the p-value gives binary results while the MSE metric measures the variance of the error. Accordingly, the MSE calculations were included to show the scale of the error. For example, although the
Figure 4.2: Error Rates by Levels of Additive White Noise
Figure 4.3: Error Rates by Levels of Additive Turbine Noise
performance of PEF and NCF had similar p-values, NCF appeared better than PEF by the MSE figures. Additionally, even though PEF had better MSE than NCF had in the 0dB turbine condition, the MSE error rate of NCF was obviously less affected by the white noise than PEF was. These results would have required more significance tests to confirm them. In either case, as expected, SQT generated the least MSE since its error is mostly quantization errors as illustrated in Figure 4.5.

Beside minimizing MSE, SQT-based methods have other advantages over the regular pitch track methods. First, the SQT method is expected to perform better than the early SQT.m version, which did not include linear interpolation and liftering and whose error was high mainly because of the quantization steps. Second, the SQT method may determine the expected MSE range beforehand since its parameter $N$ is proportional to its accuracy: $N \propto \frac{1}{3\sqrt{MSE}}$. Third, while the other pitch extract methods have one task, the SQT methods preserve the original information in its expanded space and provide a comprehensive front-end for the speech modules. SQT serves more than the pitch track. For example, the original signal can be recovered during the pitch track extraction with no computational overhead as shown in Figure 4.6, which was a twelve-order reconstruction. Most importantly, since both PEF and NCF were considered best in the field, and since SQT had fewer errors than their implemented versions, we conclude that SQT has great potential and that the SQT proof of concept is thus completed. The basic SQT model was slightly faster in Matlab than NCF was when the SQT transform was cashed. The next section elaborates on the result of the signal reconstruction.
Figure 4.4: Mean Squared Error (MSE) by Noise Type and Level
Figure 4.5: SQT.m Quantization Error

Figure 4.6: Waveform Recovered from Undistorted SQT Features
4.2 Minimal Viable Product

We introduce SQT.JS, an open source JavaScript library. It has been tested on a web application called Speech.land, which utilizes the SQT methodology to extract, upload, stream, and play the extracted features in real time. The viable product is online communications using the feature representation. The motivation is that not only does it save communication bandwidth, but also artificial agents can join its conversations. Since both natural intelligence and AI can perceive and produce speech signals directly from and to the SQT feature spaces, the SQT technology constitutes a common ground between the human agents and the artificial agents. In addition to a microphone and speakers, a machine learning algorithm would only need the SQT extractors and signal producers for the human-machine interactions. No further intermediate textual representation is needed, and that allows the artificial agents to sense the firsthand speech since the intermediate texts may not be adequate to represent the unspoken emotions. This section shows the effect of the formant normalization, the serverless streaming, and the linear interpolation on the results of the SQT implementations in Matlab, JavaScript, and Python.

Figure 4.7 compares the spectrogram of an input signal and the harmonic detection of its responsive spectrogram. It is very easy to notice that the spectrogram does not select the harmonic components, so it must end up requiring a complex classifier. Likewise, the MFCC simply assigns less weights to high frequency components to reduce the complexity, but its classifier may still perplex because of the unaligned high formants. Additionally, the triangular frequency banks, which are applied with the Mel-Frequency scale, blend the formants; hence, its features are simply unreconstructable. On the other hand, although the SQT method may not align the formants perfectly because the tract systems appear nonlinear, its alignment seems perfectly handable by Convolutional Neural Networks because the harmonic shift can be made within the reach of a few layers of fixed size filters. To put it
another way, the SQT methods are responsive to the speakers’ voices because the voice depth is inversely proportional to the \( f_0 \) value. Even though high voices may sometimes have fewer harmonic components since large \( f_0 \) values need wider channels, the extra formants can be safely represented by zeros, and the extracted features are compatible with several sampling rates. The worst case scenario is when the \( f_0 \) is not extracted correctly; when this is the case, the features are reconstructed to a random speech resolution.

The speech network traffic is optimized when SQT.JS is included on the client side although the SQT.JS software can be placed on servers for remote processing, which lowers the load on the client devices. Network traffic, user requests, and data handling are the main tasks of the server as shown in Figure 4.8. The speech extractor transforms the microphone signal to the speech frames, and multiple speech frames can be encapsulated in a chunk. The web clients exchange keys with the servers for data security and user identification. The HTTP clients may send requests with a rate that is slower than the frame rate depending on the chunk size. The chunks are counted, and their numbers are monotonically increasing but can be reset periodically as in the circular buffers. The communications may follow complicated protocols, which may be out of the feature engineering scope. For simplicity, the clients send POST requests and receive POST responses in a roughly fixed rate, whose average can be tracked for validity. In the figure, Client1 sends one chunk to the primary database server at the roughly constant request rate, and Client2 receives one chunk at roughly a similar rate from the secondary database of the server cluster. For the Speech.land web application, the serverless functions assign buffers to the virtual areas by the users. The buffers are important because of the round-trip times and the network routing, which could cause chunks to arrive earlier than previous chunks.

The size of the buffer defines a live window for the receiving clients to catch the stream and for the restoration of the order of the chunks. Also at the receiving ends, the clients may arrange the frames that are in the clients’ track queues to protect the audio outputs from
Figure 4.7: Speech Normalization by Formant Alignment
Figure 4.8: Capped Buffers of Speech Features Chunks in Serverless Streaming

the network volatility. Figure 4.9 shows speech features recorded at Client1 and played back at Client2. There was a slight difference between the two versions of the features, and the error increases proportionally to the server routing distance and bandwidth and the response time. The network error can also be reduced by increasing the buffer size and the track delay. In the figure, the first speech chunk was sent at time 4.4 seconds and was received at 6.2, which is slightly less than two seconds. Meanwhile, the last chunk in the figure was sent at 5 and received at 6.4. The difference between the two was slightly less than 1.5 seconds. Therefore, the round-trip time at the start of the connection was larger than at the end of the connection. The environment of the result was M0 Atlas Cluster of MongoDB 5.1.1 (Realm 10.10.1) at Amazon AWS us-east-1. The web clients were Firefox 97.1.0 (#2015860771) running on Android 10.0.0.196 and Google Chrome Browser 97.0.4692.102 on Chrome OS 97.0.4692.102. The chunk size was three frames (100 ms), and the server buffer size was approximately 270 ms. The bit depth was ten.

As was described in the previous chapter, the linear interpolation can increase the bit depth. The computational complexity can be significantly reduced when the pitch track is interpolated along the frequency axis during the extraction. For example, the pitch track in Figure 4.10, which has sharp edges yet is smooth when the voice has high local stationarity,
Figure 4.9: SQT.JS Result of Streaming
was constructed with only $N = 50$ quefrency filters. The pitch track can also be further interpolated along the time axis so that its synthesized waveform restores the amplitude envelope. In the figure, the signals received from the microphone and played on the speakers sounded similar although they may look different because they have different phase information. The reconstructed signal was real while the input was complex. In other words, the signal was reproduced in one axis although it was extracted from two axes. Storing the phase data doubles the size but may not increase its speech quality significantly. Since only sizable information gain should be included in online processing, it was excluded from the Matlab and JavaScript implementations. This section showed the results of the online implementations which are applicable on real time processing. More complex processing can be handled by Python. Lastly, the pitch track of the six utterances in Figure 4.10 conveyed different meanings even though they were the same word: "one, oaaaaanne, one!, one?, one?!, oahne."

The next section shows the results of the emotion detection by Artificial Neural Networks using Python. Written words are sometimes not sufficient to communicate meanings, so they are usually texted with marks and emojis.
Figure 4.10: SQT.JS Result of Repeated Word with Different Intonations
4.3 Speech Emotion Detection

If emojis were unmuted, they may sound different, just like the human voice conveys emotions. Figure 4.11 depicts the pitch intonations of the previous four words when they are extracted by complex assembled SQT windows. There are several takeaways from the previous section, the first of which is that emotions are sometimes so vivid in the pitch track, especially when \( f_0 \) is extracted correctly. Second, the quefrency-based feature engineering not only preserves the naturally smooth curves of the four voices but also the \( f_0 \) sharp edges as well. Third, the harmonic filtering clears up the cepstrogram, so that it has significantly lower perplexity than the unfiltered cepstrogram, whose detection ambiguity was pointed out earlier. Additionally, because of the usage of the rectangular windows, the 30-frame-per-second extractor preserves the actual frame energies as was expected by Parseval’s theorem. Lastly, the complex extraction may convey subtle narrow-band features, but splitting the magnitude into real and imaginary parts may not have significantly impacted the feature reconstruction. The previous section proved that SQT can improve the bandwidth usage of the online speech communications; however, voice synthesis is a byproduct of the enriching of the quality of the speech features. Since the SQT features are comprehensible to human agents, they are expected to be comprehensible to Artificial Neural Networks (ANN) and auto encoders as well. We obtained positive preliminary results in several ML domains, but due to the shortness of time, the ML results are narrowed to the speech emotion challenge. This section briefly demonstrates the extent of which the SQT features can safely be used with ANN and Convolutional Neural Networks (CNN) challenging applications. A training procedure and CNN design are introduced in the next paragraph, and the following paragraphs describe the data, the metrics and the training progress, and the results of the SpeechEmotions project.

The flowchart of Figure 4.12 summarizes the feature extraction procedure, and the block
Figure 4.11: Results of Complex Extraction by SQT.PY
diagram of Figure 4.13 summarizes the training approach. The feature extraction is mostly covered in the previous section. The distance can be measured in magnitudes and angles, as in the previous section, but it should be kept in real and imaginary parts or a magnitude to make it suitable for machine learning in this section. The harmonic components can optionally be normalized by a log operator or a fractional exponent of $1/4$ and so is formalized in the approach. Furthermore, the extracted $f_0$ is repeated $M$ times to form a vector, so the network weights it like the harmonic vector. The redundancy also simplifies the complexity of the network. The quefrency is best kept in its index format, and it is divided by the number of the quefrency pixels $N$. Its additive inverse can optionally be stacked to supply the network with constant energy regardless of the value of the fundamental frequency. The features are concatenated per frame. For our training approach, the data that has similar labels are grouped together during the training phase of the network as shown at the top of Figure 4.13. Fixed-size segments of the training samples are then drawn randomly from the label categories and then from the aggregated samples. The segment size was 75, which is the number of consecutive frames that are drawn out per training sample and is equivalent to 2.5 seconds of time span, not to be confused with the audio sample and the batch size, which is the number of samples drawn at each iteration. The SpeechEmotions network is designed with Rectified Linear Unit (ReLu) activation functions since deep learning increases the network complexity. The number of nodes is inversely proportional to the number of layers. Neural networks are able to formulate derivatives, compression, and logic gates. The first four layers of the SpeechEmotions network are for time and spectral compressions. The second and fourth layers in particular convolve with 0-percent overlapping for dimension halving, while the first and third layers are there so the network would be able to detect slopes and edges. The two dimension-reduction layers can be replaced by max-pooling for slightly lower performance. The fifth convolutional layer is there to allow the network to detect and memorize speech patterns. The sixth layers flattens the output for the last two
layers, the seventh and eighth layers, which may formulate XOR logic gates. The activation function of the eighth layer is Softmax since it is the output layer of choice for discrete labels. The SpeechEmotion model is for a categorical classification problem, and the multi-labels are handled by one-hot encoding vectors. The next paragraph describes the data and the metrics.

In summary, the previous paragraph describes the design for a 2.5-second span network architecture. The data is re-balanced during the training phase by concatenating the audio samples by labels, then a sample is drawn from the label and the concatenated audio files. The network performs three tasks. The first half of the network consists of pattern compression layers while the second half is for pattern selection and logic detection. The hyperparameters of the first two layers can be repeated for deeper speech spans while the hyperparameters of the second half can be repeated at specific depths for multi-span artificial intelligence. For the vocal emotion prediction, the 2.5 seconds appeared fine for the demonstration.

As shown in the previous paragraph, the approach modifies the data during the training phase to account for the random timing of the real-time processing environments. The metrics during training are expected to be lower than the actual result since the model trains on augmented data that is harder than the actual data. The data of this section is the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [46], containing 1,440 speech and 1,012 song samples of two statements performed by 24 male and female actors. Another similar 2,453 audio samples are available in its Audio-Visual files. The 4,905 samples contain two statements, and the North American voices expressed eight emotion labels ($Y$): 1-Neutral, 2-Calm, 3-Happy, 4-Sad, 5-Angry, 6-Fearful, 7-Disgust, and 8-Surprised. The SQT features were extracted from the audio samples at a 44.1 kHz sampling rate. At the 90 Hz frame rate, the extraction of the database aggregates 5.49 hours of data and is summarized in Figure 4.14. The batch size was 24, which is three times the number of classes. The networks weights were initiated by Normal distribution with a standard
Figure 4.12: Feature Engineering Procedure of SQT.JS and SQT.PY
Training Data Concatenated by Label:

<table>
<thead>
<tr>
<th>Label 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label 1</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>Label 7</td>
</tr>
</tbody>
</table>

Strided Convolutional Architecture:

- **Layer One**
  - L Relu Conv2D Filters
  - Size: $3 \times 5$
  - $73 \times 36 \times L$

- **Layer Two**
  - L Relu Strided Conv2D Filters
  - Size: $2 \times 2$
  - Steps: $2 \times 2$
  - $36 \times 18 \times L$

- **Layer Three**
  - L Relu Conv2D Filters
  - Size: $3 \times 5$
  - $34 \times 14 \times L$

- **Layer Four**
  - L Relu Strided Conv2D Filters
  - Size: $2 \times 2$
  - Steps: $2 \times 2$
  - $17 \times 7 \times L$

- **Layer Five**
  - L Relu Conv2D Filters
  - Size: $6 \times 7$
  - $12 \times 1 \times L$

- **Layer Six**
  - Flatten
  - $12 \cdot L$

- **Layer Seven**
  - L Relu Dense Nodes
  - $L$

- **Layer Eight**
  - 8 Softmax Dense Nodes
  - $8 (\hat{Y})$

**Data Augmentation:**

- Randomly Drawn Label & Consecutive Frames

**Batch Size:** 24

**Gaussian Initial:**

$$\sigma_{He} = \sqrt{2/T_{in}}$$

**Entropy Loss:**

$$- \sum_{i=1}^{8} y_{i} \log(\hat{y}_{i})$$

**AdaM Update:**

$$\epsilon : 10^{-6}$$

---

Figure 4.13: SpeechEmotions Architecture Layout & Training Approach
deviation of $\sigma_{He} = \sqrt{2/\text{Number of Inputs}}$, known as the initiation of He [31]. The weights can be updated using Adaptive Delta (AdaDelta) [73] with $\alpha$: 0.5 and $\beta$: 0.99 or using Adaptive Momentum (AdaM) [43] with the most default parameters ($\alpha$: 0.001, $\beta_1$: 0.9, $\beta_2$: 0.999). The metrics, defined in Equations 4.2-4.8, were calculated by Python libraries: TensorFlow-gpu (v2.4.1), TensorFlow Addons (v0.13.0), Keras (v2.6.0), scikit-learn (v1.0.2), and Numpy (1.21.2). The samples of the balanced accuracy [13], as well as the averaged accuracy, were without further weighting and adjustments. The balancing of the multiclass Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROP) was obtained by a One-Versus-One [28] strategy. The accuracy and the precision metric measure the correctness of the model output, while the recall metric measures the conditional probability of the true label incidents that are being detected. The accuracy metric does not account for the class imbalance, and the precision and recall metrics have an inverse relationship, so the F1 score is defined per label as the harmonic average between the precision and recall metrics, and it basically measures the partial accuracy per label. Furthermore, the selectivity metric is the probability complement of the recall metric, and the two are sometimes called specificity and sensitivity. The five metrics are sensitive to the threshold selection, so the last two are combined in the AUC area, which is one way to compare between the overall performances of several models. Figures 4.15 and 4.16 plot the progress of the metrics during training, where the epoch is equivalent to 1.00 and 0.27 of training and validation data hours, the model complexity parameter $L$ is 28, and the validation set split was roughly 25% of the audio samples. The plots in the first figure shows the F1 score, and it indicates that some emotion were learned earlier than others. For example, the Sad label converged later than the Supervised label. Meanwhile, the second figure shows the first three Top-K scores and the AUC area. It shows that the validation accuracy plateaued while the averages of the Top$_K$-2 and AUC metrics overfitted. However, the overfitting of the AUC did not affect the individual F1 scores. The extra training time ensured that all labels converged. With
the modified concatenated samples, the highest validation accuracy was 82.5\% at training accuracy of 96.3\%. The results of the progress are satisfying. Furthermore, better results were expected after re-applying the model at its best validation accuracy checkpoint on the original database samples.

While the previous section shows the progress of the training approach, this section shows the validation results of the ROC, the F1, and the accuracy metrics in Figures 4.17, 4.18, and 4.20 respectively. The range and domain of the ROC plots are in [0.8, 1.0]. Figure 4.19 shows how the trained network is applied to classify the original database samples. A couple of segments are drawn randomly from each dataset sample, and the emotion classifier decides the label class based on the normalized mean of the scores of the segments. The plots show almost ideal AUC areas in the training set and slight difference between the ROC curves of the labels in the validation set. In the ROC range \([0.925, 0.975]\), the curve of the Happy label appears linear. Additionally, the bar charts of the second figure (4.18) suggest the model learned the training data. The AUC areas were above 98\% in the both training and validation sets of the RAVDESS dataset. As expected, the validation F1 scores were slightly higher than they appeared during the training approach. At the lower bounds, the Happy, Sad, and Fearful labels had more than 88\% F1 scores. Meanwhile, at the higher bounds, the Disgust, Calm, Neutral, and Surprised labels had less than 93\% F1 scores. The Recall scores show that the model is more likely to detect the Calm and Angry labels than the other labels. The Type-1 and Type-2 errors were roughly evenly distributed as shown in upper right half and the lower left half of the validation confusion matrix of Table 4.1.
Figure 4.14: Exploratory Data Analysis (EDA) of SQT RAVDESS
Accuracy = \( P( \hat{Y} = Y ) \)
= Probability of Correct Classification
= \( \frac{\text{Number of True Samples}}{\text{Number of Samples}} \)

Top-K: True samples are within K most probable predictions \( (4.2) \)

Precision = \( P( \hat{Y} = Y \mid \hat{Y} = \text{Positive} ) \)
= Probability of Correct Positive Classification
= \( \frac{\text{Number of True Positive Samples}}{\text{Number of Positive Predictions}} \) \( (4.3) \)

Recall = \( P( \hat{Y} = Y \mid Y = \text{Positive} ) \)
= Sensitivity = True Positive Rate (TPR)
= \( \frac{\text{Number of True Positive Samples}}{\text{Number of Positive Targets}} \) \( (4.4) \)

Selectivity = \( P( \hat{Y} = Y \mid Y = \text{Negative} ) \)
= Specificity = True Negative Rate (TNR) \( (4.5) \)
= \( \frac{\text{Number of True Negative Samples}}{\text{Number of Negative Targets}} \) \( (4.6) \)

F1 Score = \( 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \) \( (4.7) \)

ROC Area = \( \sum_{\text{Specificity}} \text{Sensitivity} \cdot \Delta \text{Specificity} \) \( (4.8) \)
Figure 4.15: F1-Score Progress on RAVDESS by Label
Figure 4.16: Training Progress on RAVDESS on Average
Figure 4.17: The ROC Results of the SpeechEmotions RAVDESS by Label
Figure 4.18: The F1 Results of the SpeechEmotions RAVDESS by Label
Finally, the summarized results are in Figure 4.20. The validation Top$_K$-2 score was 96.11%, suggesting that the chance of having the correct label in the second prediction is about 5% when it is not the first. The balanced validation accuracy was 90.90%, and the average validation accuracy was 91.22%. The two results are excellent; therefore, the relatively high results may have been due to the excellent SQT feature engineering. Most importantly, the SpeechEmotion model is safe for use with the first-three predictions strategy since it yielded 98.34% validation Top$_K$-3 accuracy. The model may have learned the subtle genuinity of the expressed emotions, and the audio samples would probably have expressed multiple emotions. The SpeechEmotions results of other datasets may be presented in future work since more time is needed to encompass several emotion hypotheses. For example, a consensus about the emotion labels may be needed. On the one hand, the expression labels appeared to have at least 25 discrete clusters [18]. On the other hand, the expressions appeared to be projectable in a way similar to the color models such as the three RGB independent axes. For instance, the Circumplex model [60] included two independent emotion axes: the Valence axis is horizontal and indicates whether the emotion is commonly felt as pleasant,
Table 4.1: Confusion Matrices of the RAVDESS Training & Validation Samples

<table>
<thead>
<tr>
<th>Count Per Target Label $Y$</th>
<th>$y_1$</th>
<th>$y_2$</th>
<th>$y_3$</th>
<th>$y_4$</th>
<th>$y_5$</th>
<th>$y_6$</th>
<th>$y_7$</th>
<th>$y_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Set:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{y}_1$</td>
<td>295</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{y}_2$</td>
<td>0</td>
<td>571</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{y}_3$</td>
<td>1</td>
<td>0</td>
<td>563</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{y}_4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>563</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{y}_5$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>560</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{y}_6$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>566</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{y}_7$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>294</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{y}_8$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>281</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Count Per Predicted Label $\hat{Y}$</th>
<th>$\hat{y}_1$</th>
<th>$\hat{y}_2$</th>
<th>$\hat{y}_3$</th>
<th>$\hat{y}_4$</th>
<th>$\hat{y}_5$</th>
<th>$\hat{y}_6$</th>
<th>$\hat{y}_7$</th>
<th>$\hat{y}_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation Set:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{y}_1$</td>
<td>73</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{y}_2$</td>
<td>2</td>
<td>171</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{y}_3$</td>
<td>3</td>
<td>2</td>
<td>173</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$\hat{y}_4$</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>166</td>
<td>0</td>
<td>9</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$\hat{y}_5$</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>179</td>
<td>0</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$\hat{y}_6$</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>169</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\hat{y}_7$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>83</td>
<td>1</td>
</tr>
<tr>
<td>$\hat{y}_8$</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>88</td>
</tr>
</tbody>
</table>
and the Arousal axis is vertical and indicates the activation intensity. Generally, the speech emotion recognition within the SQT scope has been presented to support the SQT feature engineering. The main scope of SQT is speech feature extraction, which is relevant to the levels of emotion and intelligence because the procedure tries to tune to the vividest voice and extracts the speech features even during the speech stops and breath inhales.

### 4.4 Summary

In summary, the chapter includes several topics to support the highly novel theoretical parts of the SQT approach that is described in detail in the previous chapter. The first section proved the concept with Gaussian windows since the results of an early SQT Matlab implementation was as good as the results of some common libraries on the FDA dataset if not not better. The chapter also studied the effect of the signal to noise ratio on the pitch track and its quantization error, which can be minimized by increasing the quefrency resolution, the frame rate interpolation, or other quantization error handling methods. The quefrency resolution is increased by the $N$ parameter. The second section introduced a gpu-enabled JavaScript implementation that works in modern web browsers for a real time streaming of the SQT features. It showed that rectangular windows can significantly reduce the Internet speech communication bandwidths since it reserves the spectral energies. A premier survey suggested that 1,920 was the minimal comfortably audible bitrate of the raw SQT features. NoSQT buffers were implemented with scalable Realm serverless functions. The minimum viable product was a multiplayer outer-space exploration simulation game where players may opt-in speech surveys to help science. The second section also demonstrated examples that show a correlation between the pitch track and speech emphasis, exclamation, and other emotion related patterns. Not only can the SQT features be used by human agents but also by artificial agents; the third section concluded that SQT can be safely be used in machine
Figure 4.20: Summary of the SpeechEmotions RAVDESS Results
learning applications. As expected, the fine pitch track and responsive spectrogram helped in reducing the observation perplexity and the prediction confusion in the highly challenging emotion detection task. The SpeechEmotions model was able to achieve excellent accuracy in the RAVDESS dataset. Its validation accuracy was roughly 90%, and its training accuracy was more than 99%. Therefore, the model appeared able to generalize the well. For these reasons, we can safely say that the SQT features are robust as they enable even further speech applications, such as distant speech and automated multi-, instantaneous-, and distant-speech extraction, recognition, as well as reconstruction. Finally, the chapter mainly details the special training approach and architecture of the SpeechEmotions network.
Chapter 5

Conclusion

In summary, the Speech Quefrency Transform (SQT) methodology has been introduced. It is aptly named. SQT models quefrency filters and extracts speech features that are ready-to-use in deep learning. Our work has explained its responsive spectral selection approach, which expands the speech dimensionality in a way that aligns the speech formants and thus reduces the acoustic perplexity for the ML models. In it, the quefrency domain is obtained from a modulation-based spectral filtering in an unconventional way for speech feature engineering, resulting in a hyperdimensional speech space. Then a specific projection and filtering procedure produces its synthesizable responsive spectrogram. This chapter restates the statement of the problem and the implications of the results in simple terms to highlight the takeaways for the developers who are adopting the SQT approach. For convenience, also the novelty of the dissertation and the enumeration of our research contributions are included. Our work is presented in the hope that it will be useful but without any warranties, including without limitation warranties of fitness for a particular purpose and merchantability.
5.1 Statement

Optimizing the feature extraction for Artificial General Intelligence (AGI) is the problem statement of the work. There is a chance that network upgrades are not catching up with the information explosion of the Information Age even with the latest advancements in communications. On massive scales, new Artificial Intelligence (AI) capabilities are being rolled out in online education, banking, entertainment, and e-commerce. With more Internet of Things (IoT) products and streaming services, what is considered big data at the moment may be considered regular in the near future. Additionally, the bandwidth is not a cheap commodity in long distance communications, low power wireless networks, and multi-region data centers, where data entries are replicated for data security and scalability. There is an obvious demand in several contexts for sending more information with less bandwidth to accommodate more devices, avoid future network congestion and data center outage incidents, and optimize telecommunications.

Speech is the most natural form of communication, but it consumes unnecessarily wide bandwidths because the speech space is sparse. Although zero-energy frequencies occupy most of the speech channel, it is easier to transmit the empty spectra than to transmit the minimal extract because the speech quality deteriorates for any data loss in the minimal extract. However, as this study shows, it is possible to select the spectral speech features and reproduce the speech signal while maintaining the regularly high speech recognition performance but with lower storage and communication capacities.
5.2 Implications

It is safe to say the SQT features are generally suitable for online applications since the outlined procedure has been tested remotely in several operating systems in three programming languages: Matlab, JavaScript, and Python. SQT may hopefully be useful for machine learning emotional diagnosis in the medical field and for scarce-bandwidth telecommunication in deep sea and outer space exploration fields. Our presented SQT technology outperforms the other techniques but not necessarily the optimal solution. Since never is there enough time to explore every potential variation, there might be better versions to come in the future.

One limitation worth noting is that the SQT bitrate is calculated per speaker. Therefore, the required storage and bandwidth specifications are proportional to the number of the concurrent speakers. Anyhow, the maximum of which may be two or three since there may be little value in playing back overly contested crosstalks. One way to train the multi-speech classifiers is by generating samples from the existing single-speech datasets, whereby samples are drawn and scaled, and the labels of the samples are randomly concatenated, as shown in Figure 5.1.

Figures 5.2 and 5.3 illustrate the pixel resolutions of the quefrency and of the speech voice versus time. The unfiltered cepstrograms were displayed with the same WUW Corpus [42] samples, so the images can be visually analyzed with Figures 2.2 and 4.11 of MFCC and SQT. The images may be self-explanatory. Most of all, Figure 5.4 concludes the emotion detection results of nine economical extraction cases in which the map-reduce procedure filtered and projected the hyperdimensionality of the $3 \times 3$ grid-search resolutions extracted from the RAVDESS dataset. The results in the figure were obtained with the frame rate of 30 frames per second. As the speech frame size decreased from 1024 to 64 pixels, the validation accuracy decreased only by 5% — from 90% to 85%. Overall, the nine models fitted the training and validation datasets well; their generalizations were acceptable. Figure 5.4
suggests that deep learning still makes the best out of the SQT energy-saving cases. As expected, the lower validation accuracy resulted when the number of harmonics \((M)\) and the number of quefrecies \((N)\) were the lowest at eight.

As shown in the Methodology chapter (3), the general formula for the size of the SQT transforms is \(4f_s/f_{min} \times 2MND\), so the number of the complex frequency filters is \(2MN\). In other words, in the case when both \(M\) and \(N\) were eight, the total number of filters was 256, which exceeded a very early baseline expectation when the pitch detection task needed twice that number of filters with two high resolution Fourier transforms. Therefore, the computational complexity of that SQT case may be made similar to the computational complexity of MFCC.

The sized-eight \((M)\) power spectrogram can be stored in 64 or 128 bits, and the eight quefrecies \((N)\) can be stored in 3 or 6 bits. Thus, its transmission bitrate can be between 2k and 4k bits per second. The bitrates of the raw data may be further decreased using auto-encoding compression. The SQT bandwidth consumption apparently needs less than

\[\text{122}\]
Figure 5.2: Economical Unfiltered Cepstrograms
The $H_m$ Harmonic Sequence

Figure 5.3: Spectrograms of Reconstructed Order-12 Harmonic Sequences ($M = 12$)
Figure 5.4: Effect of Economical Parameters on Balanced Accuracy
the lowest possible bitrate of MP3.

As Figure 5.4 shows relatively high validation accuracy at $M = 8$, the first eight harmonic components may be more than enough for the $f_0$ estimation as shown at the beginning of the Results chapter (4). To put it another way, it yielded very good emotion recognition accuracy maybe because the first formant mostly resides in the fifth order harmonic sequence ($M = 5$). The results appear phenomenal when compared with the attention mechanism results of Head Fusion.

For these reasons, SQT may provide Artificial General Intelligence (AGI) with a competent speech space that is efficient in bandwidth usage and vital to artificial emotional intelligence. Efficiency implies scalability, and automatic emotion recognition implies Automatic Speech Recognition (ASR), so high information gain is inferred. It gives AGI deep learners the tools they need to produce high quality speech signals and analyze the fine details of the speech signals.
5.3 Novelty

The quefrency transform is original because it is distinct. We were not able to find the SQT model in literature, so it had to be conceptualized. The SQT methodology is unique in terms of its windows and filters design, feature normalization, and noise filtering.

In terms of window, the quefrency transform uses windowing to optimize the quefrency filters. SQT is different from MFCC, Fourier, and other approaches. MFCC uses triangular banks, while QT uses sinusoidal banks. The two are different in principle since the former uses positive parts while the latter uses the positive and negative parts. The quefrency theory (QT) is different from Fourier Theory (FT) because FT uses a fixed length window, while QT defines the window of its filters, which are centered and varied in lengths and types, like rectangular windows.

In terms of features, while MFCC does not apply speaker normalization, the SQT features respond to the speakers’ fundamental frequencies, so it aligns the formants in its responsive spectrograms using the harmonic sequences. While FT does not select the speech features, QT selects the features. The extracted quefrency space is comprehensive, and its feature selection tunes to the speaker. The SQT speech features are remarkable.

In terms of overtone filtering, other methods do not adequately lifter the overtones while the SQT-based procedures uniquely convolve the harmonic sequence to filter the overtones. It is truly amazing that much noise can be simply removed by as small as a size-two mean filter when applied along the harmonic dimension of the expanded feature space. It must be a great addition.

Our quefrency transform was not copied from sources other than Divine Providence. It was illuminated and conceptualized based on digital speech signal processing and engineering disciplines. Like any novel piece of science does, it spared no effort to shed light on and correct widespread misconceptions. The importance of data is obvious. Whether for a
classification or a regression task, it is crucial that speech data be transformed to decrease the features’ overlapping between the different distributions. Increasing the separability of the data samples greatly simplifies the classifier. The SQT methodology was designed elegantly to preserve the fine features, which reduce the ML variance. Also the introduced tangent windows and the data augmentation, to name a few, may be unprecedented but not as novel as the SQT methodology. It is truly an all-in-one pioneering solution since it addresses several challenges in signal processing, such as sequence analysis, live streaming, network communications, and language intelligence.
5.4 Contributions

The contributed theoretical and practical aspects of the dissertation research are:

1. Definition of the Speech Quefrency Transform (SQT) approach.
   (a) Proof of the unit of quefrency.
   (b) Definition of quefrency filters.
   (c) Proper window filtering.
   (d) Filtration of cepstral noise.
   (e) Formula for speech formants alignment.
   (f) Parallel extractions of multi-speech features.
   (g) Reproducibility of the SQT features.
   (h) Bandwidth reduction of raw speech data to 1.9kbps.

2. Implementations of SQT Extractor and Producer.
   (a) Loop and loopless Matlab classes.
   (b) CPU and GPU JavaScript classes.
   (c) CPU/GPU Python class.

   (a) Pitch track assessment on the FDA Evaluation.
   (b) Machine Learning emotion detection on the RAVDESS dataset.
   (c) Emotion intensity classification problem of the MuSe 2021 challenge.

4. 2018 Publication at Florida Conference on Recent Advances in Robotics (FCRAR),
    which is a conference-proceedings journal.
Bibliography


133


