
**Budapest University of Technology and Economics**
Faculty of Electrical Engineering and Informatics
Department of Telecommunications and Media Informatics

Pengyu Dai

# INVESTIGATION OF F0 ESTIMATION ALGORITHMS AND THEIR APPLICATIONS IN TEXT-TO-SPEECH AND ULTRASOUND-TO-SPEECH SYNTHESIS

SUPERVISOR

Dr. Tamás Gábor Csapó

CO-SUPERVISOR

Dr. Mohammed Salah Al-Radhi

BUDAPEST, 2020

# Contents

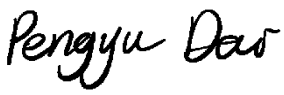
# STUDENT DECLARATION

I, **Pengyu Dai**, the undersigned, hereby declare that the present MSc thesis work has been prepared by myself and without any unauthorized help or assistance. Only the specified sources (references, tools, etc.) were used. All parts taken from other sources word by word, or after rephrasing but with identical meaning, were unambiguously identified with explicit reference to the sources utilized.

I authorize the Faculty of Electrical Engineering and Informatics of the Budapest University of Technology and Economics to publish the principal data of the thesis work (author's name, title, abstracts in English and in a second language, year of preparation, supervisor's name, etc.) in a searchable, public, electronic and online database and to publish the full text of the thesis work on the internal network of the university (this may include access by authenticated outside users). I declare that the submitted hardcopy of the thesis work and its electronic version are identical.

Full text of thesis works classified upon the decision of the Dean will be published after a period of three years.

Budapest, 18 December 2020



.....
Pengyu Dai

# Abstract

The estimated fundamental frequency (F0) of the speech signal is useful for several speech technologies. During text-to-speech, the goal is to synthesize human-like speech from text input. Articulatory-to-speech synthesis has the aim to generate intelligible speech from the recorded movement of the articulatory organs, e.g. using ultrasound tongue imaging. This thesis first shows my recent progress of a proposed F0 estimation algorithm called as “PnYIN” which is based on YIN that yields good results in the experiments. Second, the next experiment used an open-source Merlin toolkit which based on deep neural network and a vocoder that can be used for text-to-speech (TTS) synthesis. Proposed algorithm PnYIN and another five F0 estimation algorithms were applied in Merlin to train the F0 parameter. Experimental results show that the baseline algorithm of Merlin (DIO) does not perform the best in all scenarios, wheras PnYIN shows a slightly better result in an objective indicator when using female speech as input. Finally, the last three experiments were implemented on an Ultrasound-based Silent Speech Interfaces (SSI) using Hungarian and English corpus separately. This SSI uses deep neural networks to perform articulatory-to-acoustic conversion directly from ultrasound images which do not contain direct measurements of the vocal fold vibration. I investigated the effects of five different discontinuous F0 estimation algorithms in such system. I found that these discontinuous F0 algorithms are predicted with lower error, and they result in slightly more natural synthesized speech than the Idiap baseline continuous F0 algorithm. The results confirmed that discontinuous algorithms (e.g. Yin) are closer to original speech in objective metrics and subjective listening test.

# 1 Introduction

Nowadays, speech and music technologies are receiving increasing attention. Applications such as speech recognition and automatic music transcription play an essential role in human-computer interactions and are widely used in a large number of mobile devices. For example, some mobile applications are able to find the song the user sings to his/her phone. Some speech-based emotion classification systems use the statistics of acoustic feature statistics of speech samples to classify the emotion of a speech sample. Those translation and input methods applications which can take the user speech as input were widely used. Extracting accurate acoustic features such as fundamental frequency (F0) from signals is crucial for the functionalities of these kinds of applications. However, there are various negative factors on speech and music that will reduce accuracy. For instance, the speech signal varies with time, and acoustic signal is not always voiced. A number of technologies for extracting accurate acoustic features have described in the literature. This thesis introduces several widely used F0 extraction algorithms and in section 2 introduces one proposed algorithm which are slightly improved on the basis of an existing algorithm.

Extracting accurate F0 feature is a crucial task for speech recognition and speech synthesis related technologies. In this work, F0 estimation algorithms were integrated with text-to-speech (TTS) [1] and articulatory-to-speech applications. During TTS, the goal is to synthesis human understandable speech from text input. Articulatory-to-speech synthesis has the aim to generate intelligible speech from the recorded movement of the articulatory organs, e.g. Silent Speech Interface (SSI) using ultrasound tongue images [2].

This section introduced basic knowledge of F0 estimation algorithms, text-to-speech synthesis and Ultrasound-to-speech synthesis. These experiments are detailed reported begin from section 2 in the following order:

- Section 2: Proposed computational feasible solution of F0 estimation algorithm.
- Section 3: Investigation of 5 F0 estimation algorithms in Merlin (a neural network based speech synthesis system)
- Section 4: Effects of F0 Estimation Algorithms on Ultrasound-based Silent Speech Interfaces Using Hungarian Corpus
- Section 5: Extended experiment of section 4 while using English corpus UXTD.
- Section 6: Extended experiment of section 5 while using English corpus TaL1.

## 1.1 F0 Estimation Algorithms

The fundamental frequency (F0) of a periodic signal is the inverse of its period. For a perfectly periodic signal, the periodic is the smallest positive member of the set of time shifts that leave the signal invariant. For human voiced speech, F0 is usually defined as the rate of vibration of the vocal folds. Periodic vibration at the glottis may produce speech that is less perfectly periodic because of movements of the vocal tract that filters the glottal source waveform. Glottal vibration itself may also show periodicities, such as changes in amplitude, rate or glottal waveform shape, or intervals where the vibration seems to reflect several superimposed periodicities, or where glottal pulses occur without an obvious regularity in time or amplitude [3]. These factors conspire to make the task of obtaining a useful estimate of speech F0 rather difficult. Although many F0 estimation methods have been proposed, it is still a topic that attracts much effort and ingenuity.

The general procedure of estimating F0 is shown in figure 1.1. First, the input acoustic signal will be processed by a pre-processing technology, which usually aims to reduce the input domain or increase the frequency or time resolution. Second, a generator follows to estimate candidate from the true period sought and select the final sequence. In most cases, post-processing will apply to refine the F0 estimation. Figure 1.2 shows an example figure of the F0 curve. On the upper part of the figure is the speech signal of "Hello world" recorded from a male speaker. Its spectrum is shown below, and the blue line is the corresponding F0 curve.

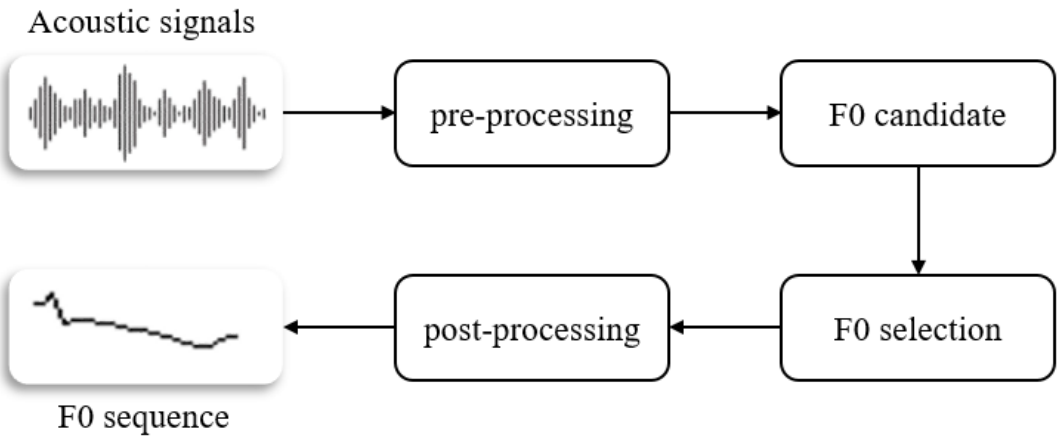



**Figure 1.1: Common work flow of F0 estimation algorithms**

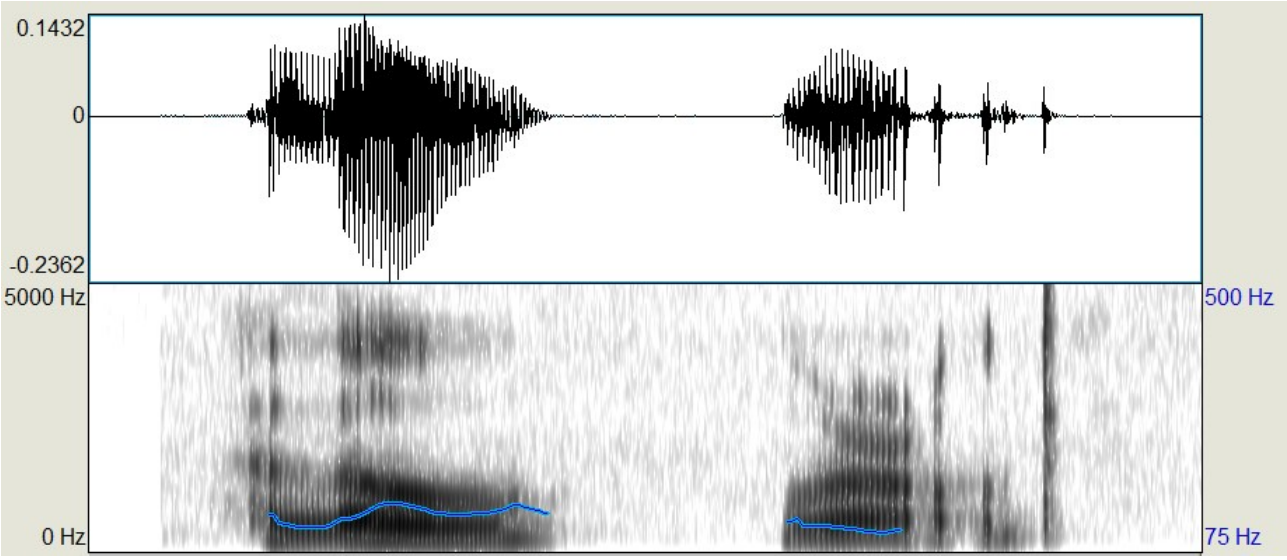



**Figure 1.2: Speech signal of "Hello world" and its F0 curve**

## 1.2 Text-to-Speech

Text-to-speech (TTS) is a technology that converts a written text into understandable human voice. A TTS synthesizer is a computer-based system that can be able to read any text aloud that is given through standard input devices. In general, a TTS system can be broken down into two main parts. The first is text analysis, where the input text is transcribed into a phonetic or some other linguistic representation, and the second one is the generation of speech waveforms, where the output is produced from this phonetic and prosodic information [1].

A general workflow of the TTS system is shown in figure 1.3.

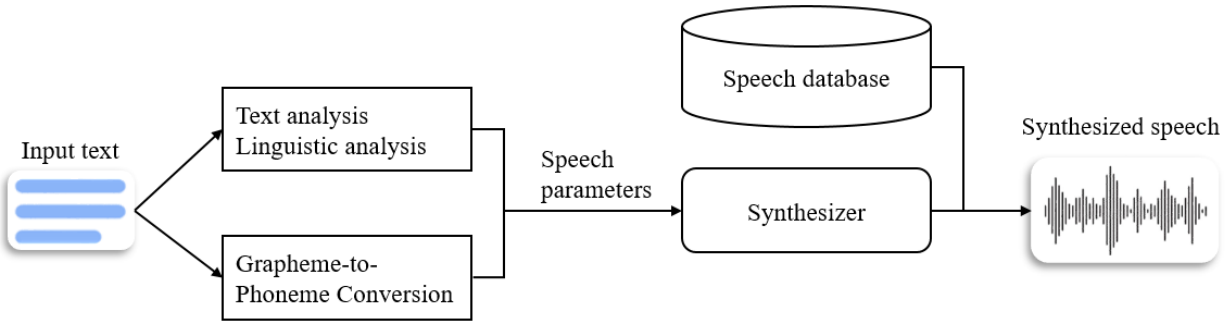



**Figure 1.3: Common workflow of TTS**

Text processing and speech generation are two main components of such TTS system. The objective of the text processing component is to process the given input text and produce appropriate sequence of phonemic units. These phonemic units are realized by the speech generation component

either by synthesis from parameters or by the selection of a unit from a large speech corpus. For natural sounding speech synthesis, it is essential that the text processing component produce an appropriate sequence of phonemic units corresponding to an arbitrary input text[4].

## 1.3 Ultrasound-to-Speech Synthesis

During the past few years, there has been a significant interest in articulatory-to-acoustic conversion, which is often referred to as "Silent Speech Interface" (SSI) [5]. This has the main idea of recording the soundless articulatory movement, and automatically generating speech from the movement information, without the subject actually producing any sound. Such an SSI system can be highly useful in many scenarios:

- For the speaking impaired people (e.g. after laryngectomy);
- In extremely noisy environments where regular speech is not feasible but the information should be transmitted;
- Silent calls in order to preserve privacy when making phone calls in public areas, or in some police actions.
- In military applications

For this automatic conversion task, typically ultrasound tongue imaging (UTI) [2, 6, 7, 8, 9], permanent magnetic articulography (PMA) [10], electromagnetic articulography (EMA) [11], electromyography (EMG) [12] or multimodal approaches [13] are employed. Ultrasound imaging of the tongue is an attractive solution because the images could be recorded in a frame rate up to 100 Hz, which will record subtle and swift movements [14]. Figure 1.4 [15] shows an example of ultrasound tongue images. The top figures show video images of the lips, and the bottom figures show the corresponding ultrasound images of the tongue.

In this work, ultrasound tongue imaging was employed in experiments. The input tongue images were recorded in midsagittal orientation using a "Micro" ultrasound system (Articulate Instruments Ltd.). Speech signals were also recorded synchronously with the ultrasound images. The workflow of the UTI system are shown in figure 1.5. In this system, speech parameters are extracted from speech signals first, and then it will feed to a training model. After training the system will predict speech parameters such as F0 from ultrasound images.

**Figure 1.4: The top figures show video images of the lips and the bottom figures show the corresponding ultrasound images of the tongue [15]**

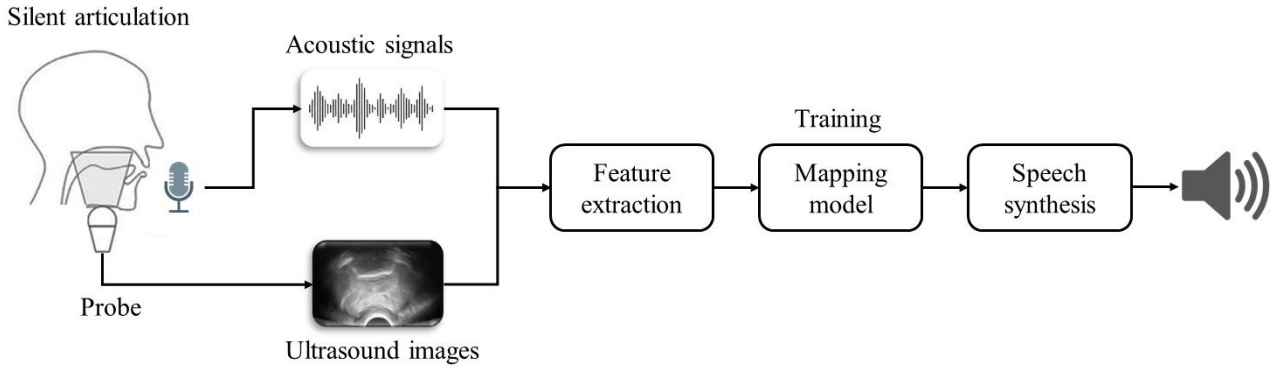



**Figure 1.5: Work flow of an ultrasound-based silent speech**

# 2 Proposed F0 Estimation Algorithm

A number of F0 estimation algorithms are described in the literature and their accuracy are differing on clean and noisy speech. It is hard to design a fully new F0 estimation algorithm. However, most F0 estimation algorithms are improved by applying pre-processing and post-processing techniques. In this work, four approached are introduced and their performance with various F0 estimation algorithms are conducted in experiments. A computationally feasible solution was proposed on the basis of existing algorithm.

Section 2.1 introduces the algorithms implemented in the experiment and section 2.2 introduces the suggested one. Performance of experimental algorithms were evaluated by several objective measurements.

## 2.1 Experimented F0 Estimation Algorithms

### 2.1.1 Yaapt

It is a Yet another algorithm for pitch tracking (Yaapt) developed in [16]. The “kernel” of Yaapt is based on the “Robust Algorithm for Pitch Tracking (RAPT) [17]”. However, both the signal processing and the tracking algorithms are very different. One of the key contributions is the extensive use of spectrographic information to guide the tracking. That is, gross errors in F0 tracking can often be identified, by overlaying pitch tracks with the low frequency part of a spectrogram. This algorithm use method for extracting this spectrographic information, and combine it with pitch estimates from correlation methods, in order to create a robust overall pitch track. Another innovation is to separately compute pitch candidates from both the original speech signal, and a nonlinearly processed version of the signal, and then to find the “lowest cost” track from among the candidates using dynamic programming.

The entire F0 tracking algorithm can be divided into five main steps,

1. Pre-processing.
2. F0 candidate selection based on normalized cross-correlation function (NCCF).
3. Candidate refinement based on spectral information (both local and global).
4. Candidate modifications based on plausibility and continuity constraints.
5. Final path determination using dynamic programming.

On the second step, the basic idea of correlation-based F0-tracking is that the correlation signal will have a peak of large magnitude at a lag corresponding to the pitch period. If the magnitude of the largest peak is above some threshold (about 0.6), then the frame of speech is usually voiced. Yaapt applied a modification to the basic autocorrelation, that is the normalized cross correlation function (NCCF) defined in [17].

Figure 2.1 shows an example of F0 curved estimated by Yaapt.

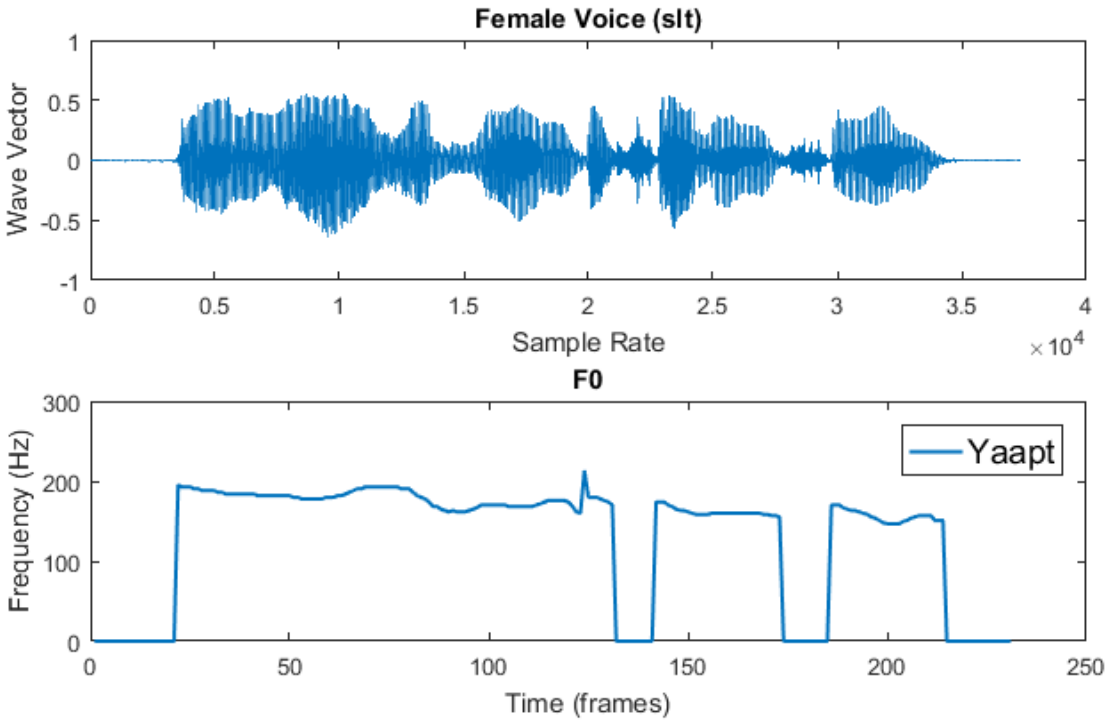



**Figure 2.1: Input audio ("Manuel had one besetting sin") with F0 of Yaapt**

## 2.1.2 YIN

The YIN algorithm uses a different function based on the autocorrelation method. While the autocorrelation function aims to maximize the product between the waveform and its shifted version, the difference function dt(τ) aims to minimize the difference between the waveform and the shifted version [19].

$$d'_t(\tau) = \begin{cases} 1 & if\,\tau = 0 \\ d_t(\tau)/[(1/\tau)\sum_{j=1}^{\tau} d_t(j)] & otherwise \end{cases} \quad (2.1)$$

Where W is the size of the window. In order to handle the quasiperiodic nature of the pitch in real signals, the YIN algorithm normalizes the difference function by its cumulative mean and sets a value of 1 for τ = 0, as

$$d_t(\tau) = \sum_{j=1}^{W} (x_j - x_{j+\tau})^2 \tag{2.2}$$

The last three steps involve placing a threshold on the smallest value of τ that is accepted. Also, parabolic interpolation is used to refine the peak location and searching around the initial pitch markers to refine the estimate further. Figure 2.2 shows an example of input female speech signal, and it's F0 curve estimated by Yin.

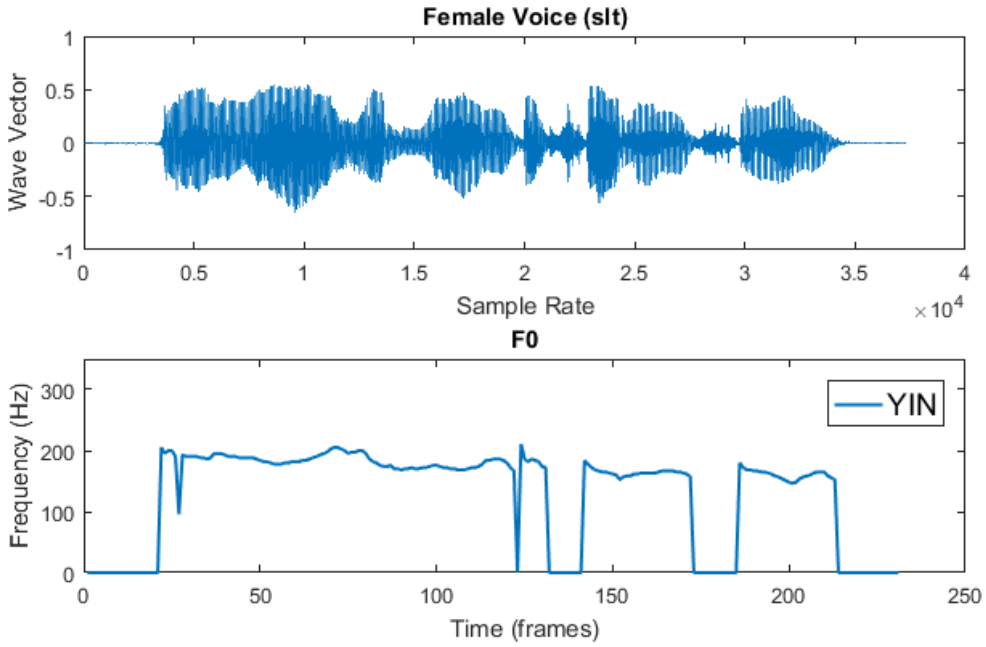



**Figure 2.2: Input female speech signal with its F0 estimated by Yin**

### 2.1.3 Swipe

A sawtooth waveform inspired pitch estimator (Swipe) was developed for speech and music [20]. Swipe estimates the fundamental frequency of the sawtooth waveform whose spectrum best matches the spectrum of the input signal. The comparison of the spectra is made by computing a normalized inner product between the spectrum of the signal and a modified cosine. The size of the analysis window is chosen appropriately to make the width of the main lobes of the spectrum match the width of the positive lobes of the cosine [20].

Swipe is similar to short-term autocorrelation function (ACF) [21], it using a cosine as a kernel to performs and integral transform of the spectrum. Instead of using the square of the magnitude of the spectrum, it used its square root. Also, it introduces some modifications to the cosine kernel to avoid some problems of autocorrelation.

The general procedure of Swipe is as follows. First, it zeroes the first quarter of the first cycle of the cosine to avoid the maximum that autocorrelation has at zero lag. Second, it multiplies the kernel by an envelope that decays as 1/f to avoid the periodicity of the autocorrelation function for periodic signals. Third, it normalizes the kernel and uses a pitch-dependent window size to make the width of the main spectral lobes match the width of the positive cosine lobes. This last step is done to avoid the tendency that autocorrelation has to give higher values to periodic signals with high F0 than to periodic signals with low F0. It can be shown that the type of signals that maximizes the inner product between the spectrum and the kernel are periodic signals whose spectral envelope decay as 1/f [22]. Figure 2.3 shows an input female speech signal, and it's F0 estimated by Swipe.

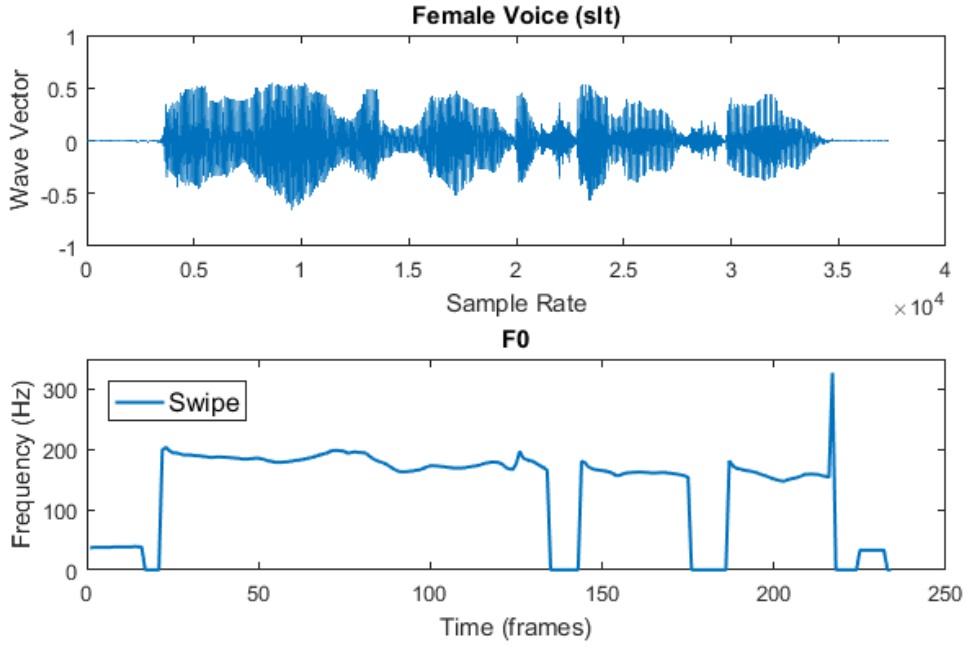



**Figure 2.3: Input female speech signal with its F0 estimated by Swipe**

## 2.1.4 ACF

Autocorrection calculates the dot-product of the original signal and a shifted version [21]. The autocorrelation function $r(\tau)$ of a signal with time lag $\tau$ is defined as follows,

$$r(\tau) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x(n+\tau) \tag{2.3}$$

The autocorrelation function always has a global maximum for $\tau = 0$. If the signal is periodic, the autocorrelation function should have global maxima at multiples of the period of the signal $T_0$ such that $rx(nT_0) = rx(0)$, n = 1, 2, 3... . In practice, $x(\tau)$ is usually a non-periodic windowed signal. Hence, no global maxima can be found outside $\tau = 0$. However, there can still be some local maxima.

If the highest of the local maxima is at a time lag τ , and the value at this point is above a threshold, the signal is said to have a periodic part. The fundamental frequency F0 is estimated to be 1/τ [22].

## 2.2 Pre-processing Methodologies

### 2.2.1 Pre-normalize

SHRP [23] is an F0 estimation algorithm used the subharmonic-to-harmonic ratio. A pre-processing technology was applied in this algorithm which aims to reduce the effect of frames with very high frequency. The function defined in (2.6).

$$X = \frac{x - \mathrm{E}(x)}{\max\left|x - \mathrm{E}(x)\right|} \tag{2.4}$$

where x is the input single-channel audio wave (a column vector).

This function used in this experiment and was named “pre-normalize”.

### 2.2.2 Nebula

This function was applied in Nebula (F0 estimation and voicing detection by modeling the statistical properties of feature extractors) [24]. The pre-processing part used in Matlab is shown in annex.

### 2.2.3 Low pass filter

A low-pass filter is a filter that passes signals with a frequency lower than a selected cutoff frequency. It also attenuates signals with frequencies higher than the cutoff frequency. A low pass filter function was applied in this experiment. The code is shown in annex.

### 2.2.4 Harmonic

This function is a critical part of Yang vocoder [54], which developed to refine F0 detection algorithms. Yang vocoder is a state-of-the-art vocoder that parameterizes the speech signal into a parameterization that is amenable to statistical manipulation. Function “RefineF0byHarmonics” [54] was used in this experiment. This function aim to refine F0 using harmonic components.

## 2.3 Error Metric

To compare the differences of estimated F0 between original algorithm with refined algorithm and how they closed to the baseline algorithm, RMSE (Root Mean Square Error) or NMSE (Normalized RMS) were used.

RMSE has been used as a standard statistical metric to measure model performance in many research studies. The RMSE represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences. Thus it measures the distance between predictions and the expected outputs. In this experiment, it used to measure the distance between the proposed algorithm with target algorithm. The smaller value will be better.

Formally it is defined in (2.6) where $Z_f$ is actual value series, $Z_o$ is estimated value series. $N$ is a sample size.

$$RMSE = \sqrt{\sum_{i=1}^{N} (Z_{f_i} - Z_{o_i})^2 / N} \tag{2.5}$$

## 2.4 Results

In the experiments, previous pre-processing methods were inserted into very beginning of each algorithm. For instance, the pre-normalize function was inserted into the beginning of Swipe; thus the input signal will be processed first by pre-normalize function. In this case, the F0 estimated by the refined Yaapt will be marked as Refined_ Swipe (see below), while the F0 estimated by original Yaapt will be marked as Original_ Swipe. The others are named in the same form.

The input speech data was from English corpus. Each measurement was conducted with 20 samples (10 male and 10 female speakers). And the final results are the average value. Since these F0 estimation algorithms work as continuous and discontinuous separately, they were compared to different baseline algorithms. F0 estimated by Yaapt and SHRP was seen as the target curve for discontinuous and continuous algorithms separately. For instance, original swipe (Original_Swipe) and refined Swipe (Swipe) will compare to Yaapt separately. The smaller RMSE value means it is more close to the target curve.

### 2.4.1 Results of pre-normalize

Table 2.1 list the results of the pre-normalize function. Please note that the goal is to minimize the average value, and the best one will be bold. From the results, we observe that:

**a)** Swipe

There almost no differences between original and refined value.

**b)** Yin

It is obvious that refined YIN gets a smaller value which means it is closer to the target curve, and it successfully refined the original YIN.

**c)** ACF

There is no difference between the original and refined value.

**Table 2.1: Objective measurement metrics of pre-normalize method**

| Metrics | Comparison pair | Average value |
|---|---|---|
| RMSE | (Yaapt, Original_Swipe) | 42.5344 |
| RMSE | (Yaapt, Refined_Swipe) | 42.5343 |
| RMSE | (Yaapt, Original_YIN) | 37.6074 |
| RMSE | (Yaapt, Refined_YIN) | **31.3928** |
| NRMS | (SHRP, Original_ACF) | 23.2474 |
| NRMS | (SHRP, Refined_ACF) | 23.2474 |

Further experiments were applied with refined YIN. As shown in figure 2.4, we can see that compared to the baseline Yaapt (black curve) the Original-YIN makes two big errors, while refined-YIN fixed it. This makes better results.

However, these better results only obviously performed in female input speech. The possible reason may be that the pre-normalize function reduces the impact of those frames which have extremely high frequencies. And usually female voice frequency is higher than the male voice.

This combination of pre-normalize function with YIN got slightly better results than the original YIN. This refined YIN algorithm was proposed as a "new" algorithm, and I was named it as a "PnYIN" bellow.

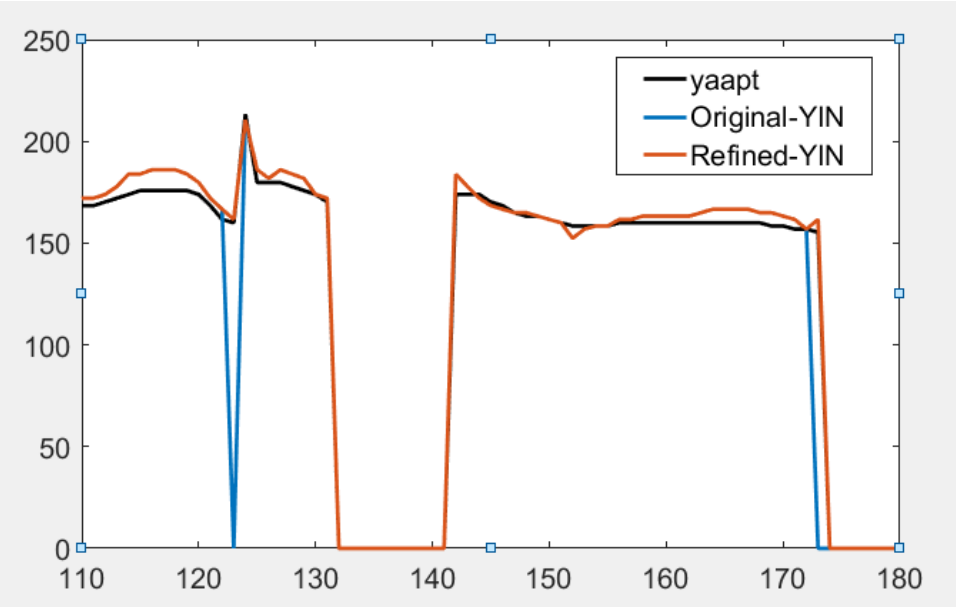



**Figure 2.4: Figure of target curve (Yaapt) in black, original YIN in blue and refined YIN in red**

### 2.4.2 Results of Nebula

Table 2.2 list the results of Nebula function, and figure 2.5 ~ 2.7 shows some example of their F0 curve. The bold value in table 2.2 is the better one of the comparison pair. We observe that:

**a)** Swipe

Swipe refined by Nebula function and the refined one is closer to target curve Yaapt than the original one. Hence, more experiments were conducted with their F0 curve. One example is shown in figure 2.5. We see that curve of original YIN makes two big "mistake" which is far different with target curve (Yaapt). However, these two "mistake" are fixed in the refined YIN curve. Thus the performance of refined YIN is more closed to the target algorithm. We can say that Nebula function slightly improved Swipe in current experiments.

**b)** YIN

Nebula also improved YIN. We see that the refined YIN have better value than the original one. An example of their F0 curve is shown in figure 2.6. We observe that both original and refined YIN overestimated the F0 value. However, the overestimated part of refined YIN is smaller than the original one, which makes it closer to the target curve. So Nebula function also improved YIN slightly.

**c)** ACF

Refined YIN has smaller value as well. An example of comparison of their F0 curve is shown in figure 2.7. We observe that in the begin of the curve, both original and refined YIN are far different with the target curve. In the rest part, refined YIN are closer to the target curve. Especially in end of the curve, original YIN goes in the opposite direction which makes wrong F0 estimation.

**Table 2.2: Objective measurement metrics of Nebula method**

| Metrics | Comparison pair | Average value |
|---|---|---|
| RMSE | (yaapt, Original_Swipe) | 47.8825 |
| RMSE | (yaapt, Refined_Swipe) | **43.9488** |
| RMSE | (yaapt, Original_YIN) | 17.4952 |
| RMSE | (yaapt, Refined_YIN) | **14.7378** |
| NRMS | (SHRP, Original_ACF) | 17.4952 |
| NRMS | (SHRP, Refined_ACF) | **14.7378** |

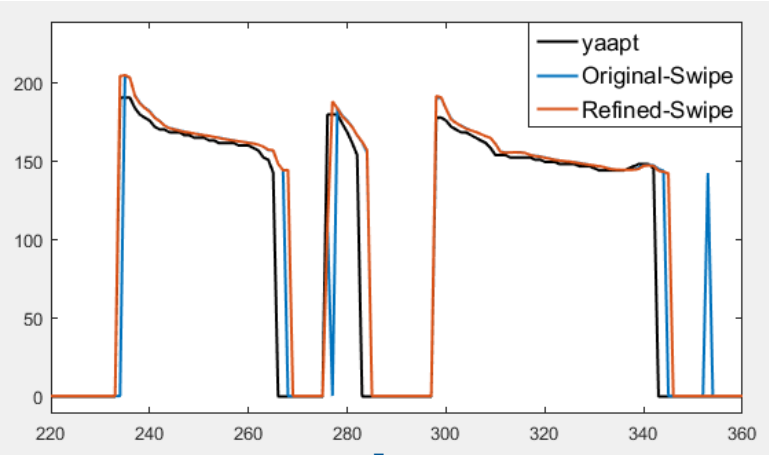



**Figure 2.5: Figure of target curve (Yaapt), original Swipe and refined Swipe**

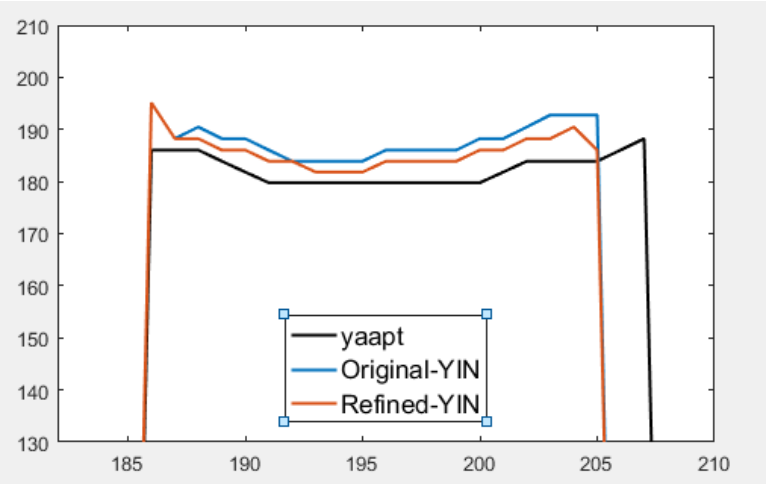



**Figure 2.6: Figure of target curve (Yaapt), original YIN and refined YIN**

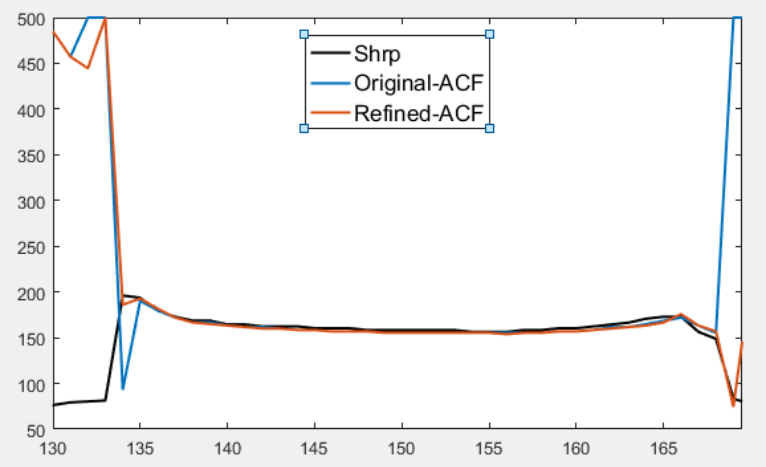



**Figure 2.7: Figure of target curve (SHRP), original ACF and refined ACF**

### 2.4.3 Results of low pass filter

Table 2.3 shows the experimental results of a low pass filter. We observe that:

a) Swipe

Low pass filter failed to improve Swipe. The combination of Swipe with low pass filter makes the results worse.

b) YIN

Refined YIN get the smaller value of RMSE which means it is slightly improved by low pass filter. However, the improved part is too small and it is not a convincing result.

c) ACF

The same with Swipe, adding low pass filter makes ACF works worse.

**Table 2.3: Objective measurement metrics of low pass filter**

| Metrics | Comparison pair | Average value |
|---|---|---|
| RMSE | (yaapt, Original_Swipe) | **43.1005** |
| RMSE | (yaapt, Refined_Swipe) | 43.6665 |
| RMSE | (yaapt, Original_YIN) | 33.9421 |
| RMSE | (yaapt, Refined_YIN) | **33.7100** |
| NRMS | (SHRP, Original_ACF) | **23.2474** |
| NRMS | (SHRP, Refined_ACF) | 24.3678 |

In general, low pass filter didn't perform well in this experiment. Although an improved example was found in figure 2.8, it might be a particular case.

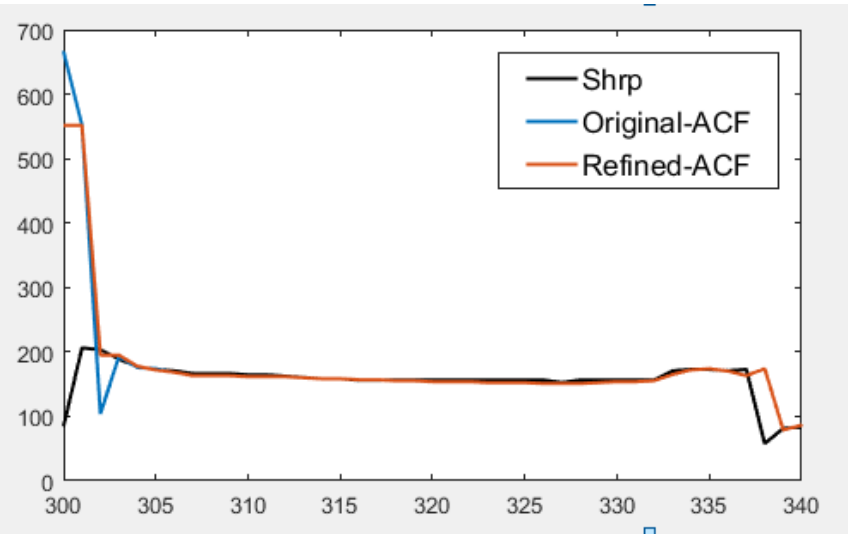



**Figure 2.8: Figure of target curve (SHRP), original ACF and refined ACF**

### 2.4.4 Results of Harmonic

Experiments of harmonic function only implemented with YIN and ACF, because the harmonic function is not fit for Swipe. In below table 2.4, we observe that:

a) YIN

YIN is improved by harmonic function; however, the improved part is so small that can't be convincing evidence.

b) ACF

Although the result is not bad, it is not convincing enough as well.

**Table 2.4: Objective measurement metrics of harmonic**

| Metrics | Comparison pair | Average value |
|---|---|---|
| RMSE | (yaapt, Original_YIN) | 33.8149 |
| RMSE | (yaapt, Refined_YIN) | **33.7687** |
| NRMS | (SHRP, Original_ACF) | 21.2310 |
| NRMS | (SHRP, Refined_ACF) | **21.064** |

Further experiments conducted in their F0 curve. An example shown in figure 2.9, there are almost no difference between original and refined ACF curve. Although NRMS shows refined ACF is the better one, the improved part is so small that could be ignored. Hence, the harmonic function fails to improve these algorithms.

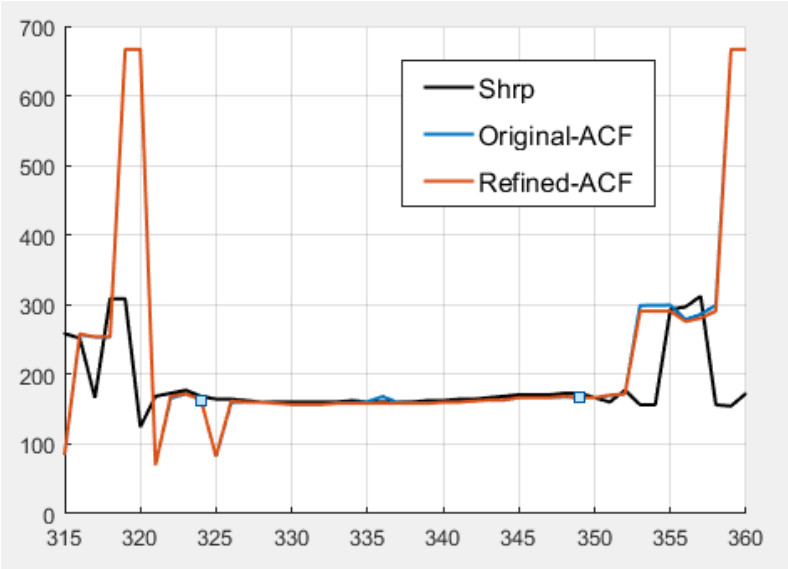



**Figure 2.9: Figure of target curve (SHRP), original ACF and refined ACF**

## 2.5 Conclusion

Four pre-processing technologies have been implemented with Swipe, YIN and ACF. Both pre-normalize and Nebula shows some convincing results in objective metrics. YIN was slightly improved by pre-normalize function. Although this better results only occur on a female voice, it is a good result. Nebula successfully improved Swipe, YIN and ACF; however, the improvement is not convinced enough from more experiment with F0 figures.

## 2.6 PnYIN

In the objective evaluation, the combination of pre-normalize function with YIN shows convinced results. Thus this combination is the proposed F0 estimation algorithm and it is named “PnYIN” (Pre-normalized YIN) in the later experiments.

# 3 Investigation of F0 Estimation Algorithms in Merlin

Merlin is a powerful toolkit for building Deep Neural Network models for statistical parametric speech synthesis. The system takes linguistic features as input and employs neural networks to predict acoustic features, which are then passed to a vocoder to produce the speech waveform. In these experiments, 5 existing F0 estimation algorithms and the proposed algorithm PnYIN were integrated into Merlin. Their performances were assessed by objective measurements of predicted F0 features.

## 3.1 Investigated F0 Estimation Algorithms

Except for Yaapt, YIN, and Swipe introduced in section 2.1, another two F0 estimation algorithm Rapt and DIO were investigated in this experiment.

### 3.1.1 DIO

DIO [25] is proposed for real-time interactive applications using a singing voice. It does not require expensive computation such as autocorrelation. DIO consists of three steps. The first step is low-pass filtering with different cutoff frequencies. If the filtered signal only consists of the fundamental component, it forms a sine wave with a period of T0, which is the fundamental period. Since the target F0 is unknown, many filters with different cutoff frequencies are used in this step. The second step is to calculate the F0 candidates and their reliabilities in each filtered signal. Since a signal that consists of only the fundamental component form a sine wave, the four intervals of the waveform, i.e., the positive and negative zero-crossing intervals and peak and dip intervals have the same value. Their standard deviation is therefore associated with the reliability measure, and their average is defined as an F0 candidate. In the third step, the candidate with the highest reliability is selected [26].

Figure 3.1 shows an input female speech signal, and it’s F0 estimated by DIO.

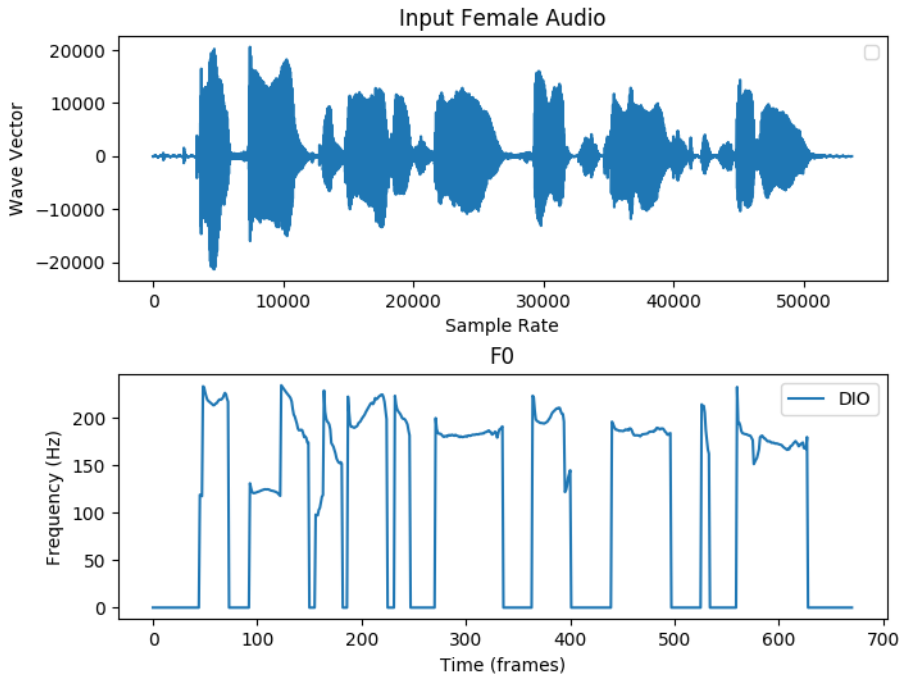



**Figure 3.1: Input female speech signal with its F0 estimated by Dio**

### 3.1.2 Rapt

A robust algorithm for pitch checking (Rapt) [27] is designed to work at any sampling frequency and frame rate over a wide range of possible F0, speaker and noise conditions. The primary aim of Rapt F0 estimator was to obtain the most robust and accurate estimates possible, with little thought to computational complexity, memory requirements or inherent processing delay. Rapt significantly reduce computational cost while maintaining the desired accuracy by incorporating several efficiency enhancements [28].

The steps that constitute Rapt are shown below:

- Provide two versions of the sampled speech data; one at the original sample rate; another at a significantly reduced rate.
- Periodically computes the normalized cross correlation function (NCCF) of the low sample rate signal for all lags in the FO range of interest. Record the locations of local maxima in this first-pass NCCF.
- Compute the NCCF of the high sample-rate signal only in the vicinity of promising peaks found in the first pass. Search again for local maxima in this refined NCCF to obtain improved peak location and amplitude estimates.

- Each peak retained from the high-resolution NCCF generates a candidate FO for that frame. At each frame, the hypothesis that the frame is unvoiced is also advanced.
- Dynamic programming is used to select the set of NCCF peaks or unvoiced hypotheses across all frames that best match the characteristics mentioned above.

In the second step, the basic idea of correlation-based F0-tracking is that the correlation signal will have a peak of large magnitude at a lag corresponding to the pitch period. If the magnitude of the largest peak is above some threshold (about 0.6), then the frame of speech is usually voiced. Instead of basic autocorrelation, the normalized cross correlation function (NCCF) is applied.

Figure 3.2 shows an input female speech signal, and it's F0 estimated by Rapt.

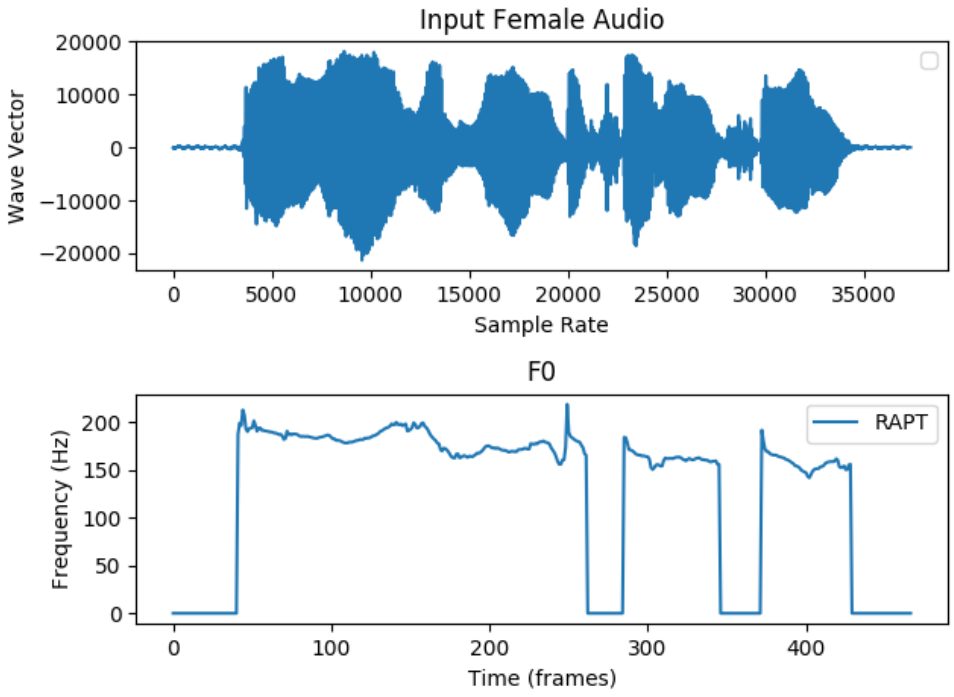



**Figure 3.2: Input female speech signal with its F0 estimated by Rapt**

## 3.2 Acoustic Modeling

### 3.2.1 WORLD Vocoder

A vocoder is a category of voice codec that analyzes and synthesizes the human voice signal for audio data compression, multiplexing, voice encryption or voice transformation. WORLD is a vocoder-based high-quality speech synthesis system developed in an effort to improve the sound quality of real-time applications using speech [29]. In many applications such as singing synthesizers and voice conversion systems, high-quality speech synthesis systems are the curial part. Also speech

analysis, manipulation, and synthesis based on the idea of the vocoder are widely used. Such systems consist of F0 and spectral envelope estimation algorithms and a synthesis algorithm that takes the estimated speech parameters. However, the speech synthesized by most of the conventional vocoder systems is inferior to that of waveform-based systems. An exception is a vocoder-based system called STRAIGHT [30], which is capable of high-quality speech synthesis. High-quality speech synthesis remains a popular research topic.

Real-time processing is another topic of speech synthesis research. For example, voice conversion for Karaoke requires real-time analysis and synthesis. Real-time STRAIGHT has been proposed as a way to meet the demand for real-time processing, but the simplified algorithm it used degrades the quality of the synthesized speech. Real-time singing morphing has the same problem. TANDEM-STRAIGHT [31, 32] is supposed to be a simplified version that outputs almost all the same parameters as STRAIGHT. The system works well, but it is hard to use it for real-time speech analysis and synthesis.

Even several high-quality speech synthesis systems have been developed, real-time processing has been complicated with them because of their high computational costs. In contrast, WORLD is a high-quality speech synthesis system developed in an effort to improve the sound quality of real-time applications using speech. This system has not only sound quality but also quick processing. The effectiveness of the system was evaluated by comparing its output with against natural speech, including consonants. Its processing speed was also compared with those of conventional systems. The results showed that WORLD was superior to the other systems in terms of both sound quality and processing speed. In particular, it was over ten times faster than the conventional systems, and the real-time factor (RTF) indicated that it was fast enough for real-time processing.

WORLD consists of three algorithms for obtaining three speech parameters and a synthesis algorithm that takes these parameters as input. Figure3.3 illustrates the processing of the system. First, the f0 contour is estimated with DIO. Second, the spectral envelope is estimated with CheapTrick, which uses not only the waveform but also the F0 information. Third, the excitation signal is estimated with PLATINUM and used as an aperiodic parameter.

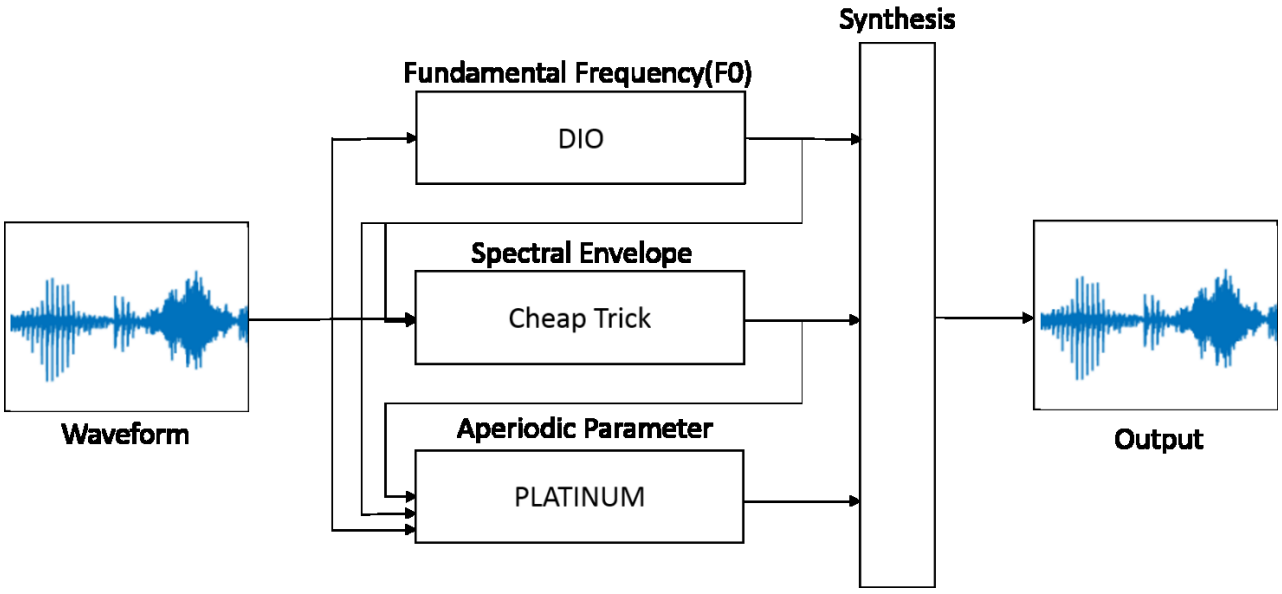



**Figure 3.3: WORLD vocoder workflow**

### 3.2.2 Deep Learning

Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces. Most modern deep learning models are based on artificial neural networks, such as Deep Feed-forward Neural Network, although they can also include propositional formulas or latent variables organized layer-wise in deep generative models [33].

In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input may be a matrix of pixels; the first representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode a nose and eyes; and the fourth layer may recognize that the image contains a face [34]. Importantly, a deep learning process can learn which features to optimally place in which level on its own. The deep learning architectures can be constructed with a greedy layer-by-layer method. Deep learning helps to disentangle these abstractions and pick out which features improve performance.

Deep learning architectures have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, etc., where they have produced results comparable to and in some cases superior to human experts.

In this experiment, a Deep Feed-forward Neural Network was applied. A neural network is a series of algorithms that endeavours to recognize underlying relationships in a set of data through a

process that mimics the way the human brain operates. It can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. A "neuron" in a neural network is a mathematical function that collects and classifies information according to a specific architecture [35]. A neural network works similarly to the human brain's neural network, and each neural network contains layers of interconnected nodes. Each node is a perceptron and is similar to multiple linear regression. The perceptron feeds the signal produced by a multiple linear regression into an activation function that may be nonlinear.

In this work, experiments were conducted with Merlin, which uses Deep feed-forward neural networks (DNNs). DNN as a deep conditional model are the model popular model to map linguistic features to acoustic features directly. The DNNs can be viewed as replacement for the decision tree used in the HMM-based speech as detailed in. It can also be used to model high-dimensional spectra directly. In the feedforward framework, several techniques such as multitask learning, minimum generation error, have been applied to improve the performance. However, DNNs perform the mapping frame by frame without considering contextual constraints, even though stacked bottleneck features can include some short-term contextual information [36].

A feedforward neural network is the simplest type of network. With enough layers, this architecture is usually called a Deep Neural Network (DNN). The input is used to predict the output via several layers of hidden units, each of which performs a nonlinear function, as follows [36]:

$$\begin{aligned} h_t &= K(W^{xh}x_t + b^h) \\ y_t &= W^{hy}h_t + b^y \end{aligned} \tag{3.2}$$

Where $K(\cdot)$ is a nonlinear activation function in a hidden layer, $W^{xh}$ and $W^{hy}$ are the weight matrices, $b^h$ and $b^y$ are bias vectors, and $W^{hy}h_t$ is a linear regression to predict target features from the activations in the preceding hidden layer. Figure 3.4 shows an example workflow of feedforward neural network. In this experiment, the network takes linguistic features as input and predicts the vocoder parameters through several hidden layers.

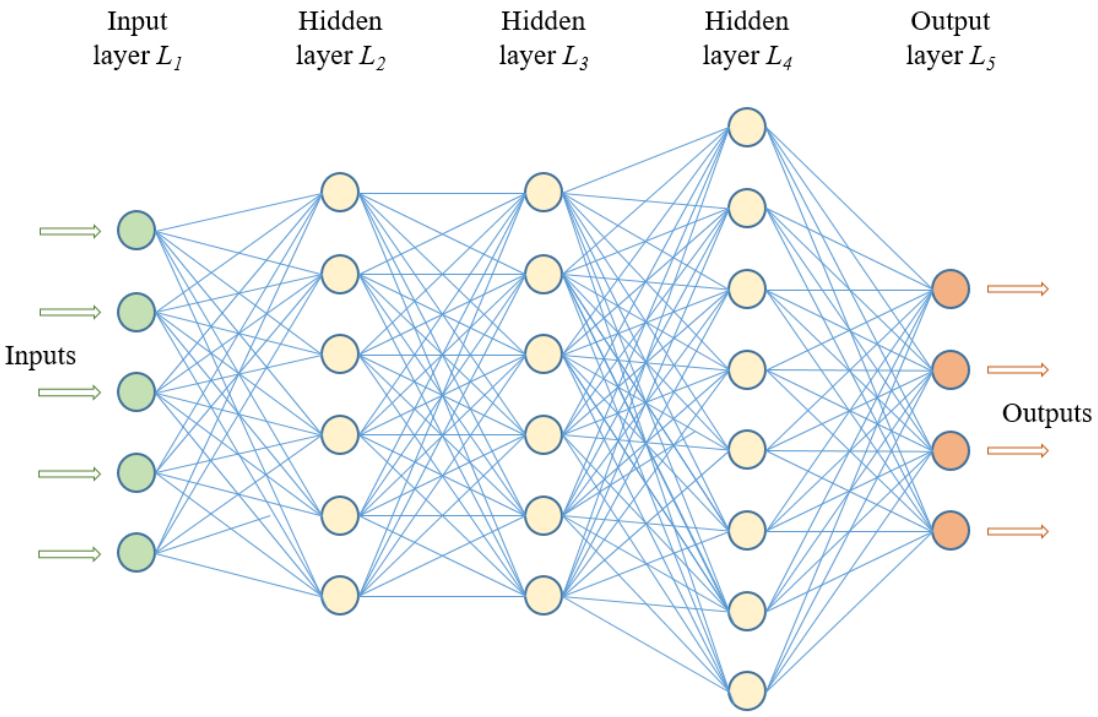



**Figure 3.4: Sample workflow of feed-forward neural network**

## 3.3 Experimental Conditions

### 3.3.1 Merlin: The Neural Network (NN) based Speech Synthesis System

Merlin [37] is a toolkit for building Deep Neural Network models for statistical parametric speech synthesis. The system takes linguistic features as input, and employs neural networks to predict acoustic features, which are then passed to a vocoder to produce the speech waveform. Various neural network architectures are implemented, including a standard feedforward neural network, mixture density neural network, recurrent neural network, long short-term memory (LSTM) [38] recurrent neural network, amongst others. It is developed at the Centre for Speech Technology Research (CSTR), University of Edinburgh. It must be used in combination with a front-end text processor (e.g., Festival) and a vocoder (e.g., STRAIGHT or WORLD).

Text-to-speech (TTS) synthesis involves generating a speech waveform, given textual input. Freely-available toolkits are available for two of the most widely used methods: waveform concatenation, and hidden Markov models (HMM) [39] based statistical parametric speech synthesis, or simply Statistical Parametric Speech Synthesis (SPSS) [40]. Even though the naturalness of good waveform concatenation speech continues to be generally significantly better than that of waveforms

generated via SPSS using a vocoder, the advantages of flexibility, control, and small footprint mean that SPSS remains an attractive proposition.

In SPSS, one of the most important factors that limit the naturalness of the synthesized speech is the acoustic model. For the past decade, hidden Markov models (HMMs) have dominated acoustic modelling. The way that the HMMs are parametrized is critical, and almost universally this entails clustering groups of models for acoustically- and linguistically-related contexts, using a regression tree. However, the necessary across-context averaging considerably degrades the quality of synthesized speech. In this case, Merlin used a more powerful regression model than a tree. Thus, Merlin has more training data, more advanced computational resource, more advanced training algorithms, and significant advancements in the various other techniques needed for a complete parametric speech synthesizer: the vocoder, and parameter compensation, enhancement and post-filtering techniques.

### 3.3.2 DNN configuration

The DNN model of Merlin has a lot parameter; hence we can configure it freely to fit our demands. And there are some significant configurations such as the number of layers and neurons which highly influence the performance of DNN.

In my case, I used the default configuration first, which uses 6 hidden layers and every layer contains 1024 neuron. Kind of hidden layer can be indicated in Merlin, and I used TANH type. TANH means Hyperbolic tangent as shown in figure 3.5. The output ranges of TANH from -1 to 1 and having an equal mass on both the sides of zero-axis so it is zero centered function. So TANH overcomes the non-zero centric issue of the logistic activation function. Hence optimization becomes comparatively easier than logistic, and it is always preferred over logistic.

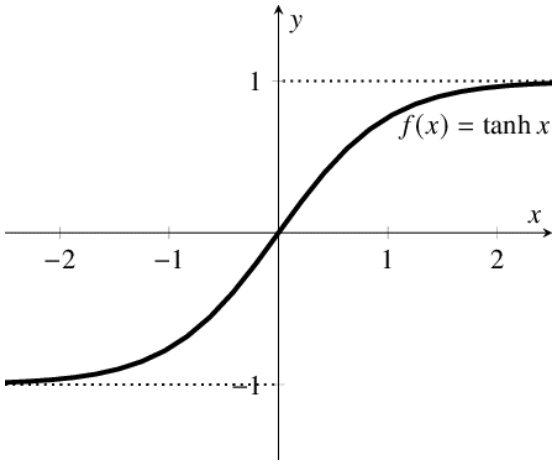



**Figure 3.5: Figure of TANH function**

## 3.4 Objective measurement metrics

### 3.4.1 MCD

Mel cepstral distortion (MCD) is a measure of how different two sequences of Mel cepstra are. It is used in assessing the quality of parametric speech synthesis systems, including statistical parametric speech synthesis systems, the idea being that the smaller the MCD between synthesized and natural Mel cepstral sequences, the closer the synthetic speech is to reproducing natural speech [41]. It is by no means a perfect metric for assessing the quality of synthetic speech but is often a useful indicator in conjunction with other metrics.

### 3.4.2 BAP

This is a band aperiodicity (BAP) of speech signals, where “aperiodicity” is defined as the power ratio between the speech signal and the aperiodic component of the signal. This power ratio depends on the frequency band, so the aperiodicity should be given for several frequency bands. It is Aperiodic energy and Typically around 3 to 5 bands (on a Mel scale).

### 3.4.3 RMSE

Root mean square error (RMSE) measures the distance between predictions and the expected outputs. For more detail, see section **错误!未找到引用源。**.

### 3.4.4 VUV

The voiced/unvoiced (VUV) analysis metric is designed to estimate a cut-off frequency for voiced and unvoiced part of a signal, in analogy with the production mode of vocal sounds. For evaluation rule, if one frame in the VUV reference is voiced while the output of the corresponding frame is unvoiced (or vice-versa), then it is counted as one error. Thus the smaller of VUV, the better of F0 estimation.

### 3.4.5 Training Time

Training time represents the training time of DNN model.

### 3.4.6 Validation error

The validation error gives us an idea about how well our model does on data used to train it. So smaller validation error is better.

## 3.5 Results and Evaluation

Usually, females have higher voice frequency than male when speaking the same sentences. As shown in figure 3.6, in most interval the frequency of female are higher than male. Hence, my experiments are conducted in two separate speech database. One from female speaker SLT while another from male speaker BDL.

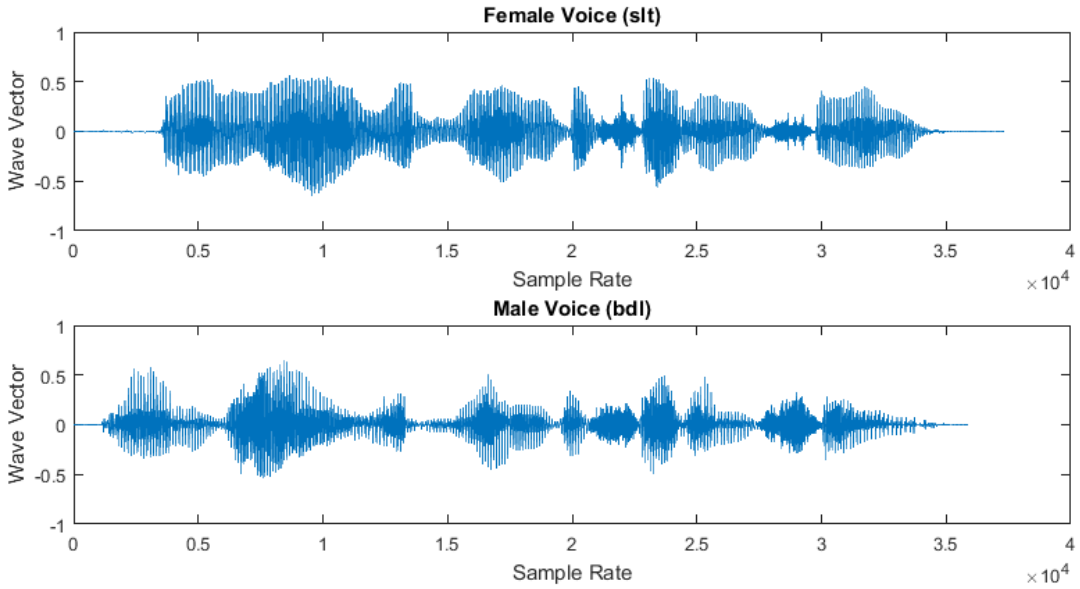



**Figure 3.6: Female and male speech signal in the same sentences**

### 3.5.1 Feamal speaker: SLT

The results are shown in table 3.1. We notice that:

- The MCD of RAPT is the smallest, which means the generation voice of RAPT is much more close to natural speech than others.
- The BAP of these algorithms are almost the same, but DIO is the smallest. Thus DIO is more accurate than others in estimating periodic.
- Swipe has the smallest RMSE. It means for the accuracy Swipe is better than DIO and others.
- For the VUV parameter, MYIN has the best performance. It means MYIN is much more accurate in estimating voice and unvoiced frames than others.
- For the train time YIN is the smallest, which says it running faster.

In conclusion, Swipe has the best performance at F0 estimation, while Swipe and DIO take much more time than others in training time. MYIN get a good result at VUV, which means accurate estimation at voiced and unvoiced frames.

**Table 3.1: SLT objective metrics**

| | MCD (dB) | BAP (dB) | F0 RMSE (Hz) | VUV (%) | Train Time (m) | Validation error |
|---|---|---|---|---|---|---|
| DIO | 4.908 | **0.232** | 11.787 | 5.286% | 148.11 | **155.879** |
| RAPT | **4.901** | 0.233 | 15.602 | 4.827% | 133.95 | 157.064 |
| Yaapt | 4.908 | 0.233 | 17.487 | 5.371% | 138.92 | 156.020 |
| Swipe | 4.909 | 0.233 | **10.340** | 5.516% | 141.88 | 156.104 |
| YIN | 4.906 | 0.233 | 17.879 | 5.088% | **132.45** | 156.334 |
| MYIN | 4.907 | 0.233 | 19.259 | **4.564%** | 133.25 | 156.498 |

### 3.5.2 Male speaker: BDL

Although Merlin comes with female waveform, male waveform also could be managed. The objective measurements results are shown in table 3.2. We observe that:

- The MCD of DIO is the smallest, which means the generation voice of DIO is much more close to natural speech than others.
- For BAP, Swipe is the smallest. Thus Swipe is more accurate than others in estimating periodic.
- For RMSE, Swipe still has the best result. It means for the accuracy Swipe is better than others.
- For VUV, RAPT is best, which means accurate estimation at voiced and unvoiced frames.
- YIN has the shortest train time while Yaapt has the smallest validation error.
- Although RMSE of DIO is bigger than Swipe, it is smaller than others. And the result of Yaapt, YIN and MYIN is obvious bigger than DIO.
- For YIN and MYIN, the RMSE of MYIN is smaller than YIN, it means MYIN have better performance.

In conclusion, Swipe still has the best performance of F0 estimation and it is better than DIO in both male and female speech voice. RAPT has obviously much more accurate in voiced and unvoiced estimation. However, MYIN has no good performance in a male voice.

**Table 3.2: BDL objective metrics**

| | MCD (dB) | BAP (dB) | F0 RMSE (Hz) | VUV (%) | Train Time (m) | Validation error |
|---|---|---|---|---|---|---|
| DIO | **5.185** | 0.224 | 14.761 | 10.548% | 147.86 | 160.357 |
| RAPT | 5.217 | 0.213 | 15.964 | **7.676%** | 129.65 | 160.688 |
| Yaapt | 5.191 | 0.225 | 20.380 | 10.477% | 128.64 | **160.100** |
| Swipe | 5.225 | **0.212** | **12.541** | 13.610% | 138.71 | 167.804 |
| YIN | 5.215 | 0.224 | 21.628 | 11.224% | **127.52** | 160.959 |
| MYIN | 5.216 | 0.224 | 23.690 | 10.178% | 128.68 | 160.912 |

## 3.6 Conclusion

In this experiment, 5 F0 estimation algorithms were implemented with Merlin. Although DIO is the baseline F0 estimation of Merlin, this experiment results show that DIO not perform best in all the metrics. Generally, Swipe has slightly better performance than DIO. My proposed algorithm PnYIN works in the experiment, and it is slightly better than others in VUV measurements when using female speaker voice as input.

# 4 Effects of F0 Estimation Algorithms on Ultrasound-based Silent Speech Interfaces Using Hungarian Corpas

## 4.1 F0 Estimation on Slient Speech Interfaces

State-of-the-art SSI systems use the 'direct synthesis' principle, where the speech signal is generated directly from the articulatory data, using vocoders [2, 8, 9, 10, 13, 42]. Most of these approaches focus on predicting just the spectral features of the vocoder (e.g. Mel-Generalized Cepstrum, MGC). The reason for this is that while there is a direct relation between tongue movement and the spectral content of speech, the F0 value depends on the vocal fold vibration, which has no direct connection with the movement of the tongue and face or the opening of the lips [43]. However, there is some evidence that tongue shapes differ in the case of voiced and unvoiced sounds; for example, the vibration of the vocal folds may slow down during consonant articulation [44]. Along with other factors, these changes correlate with the specific articulatory configuration of the obstruents; that is, the volume of space between the glottis and the obstacle [45]. In spite of these facts, most authors studying SSI systems take the unpredictability of F0 for granted and use the original F0, a constant F0 or white noise as excitation.

Only a few studies attempted to predict the voicing feature and the F0 curve using articulatory data as input. Nakamura et al. utilized EMG data, and they divided the problem into two steps. First, they used a support vector machines (SVM) for voiced/unvoiced (V/U) discrimination, and in the second step they applied a Gaussian mixture model (GMM) for generating the F0 values. According to their results, EMG-to-F0 estimation achieved a correlation of 0.5, while the V/U decision accuracy was 84% [12]. Lorenz et al. also utilized EMG data, and they applied a quantization approach to generating F0 in an EMG-to-Speech Conversion SSI. This approach quantizing the EMG-to-F0 mappings target values, and thus turning a regression problem into a recognition problem. This new F0 generation method achieves a significantly better performance than a baseline approach [46]. Hueber et al. experimented with predicting the V/U parameter along with the spectral features of a vocoder, using ultrasound and lip video as input articulatory data. They applied a feed-forward deep neural network (DNN) for the V/U prediction and attained an accuracy score of 82%, which is very similar to the result of Nakamura et al. [2].

Another two studies experimented with EMA-to-F0 prediction. Liu et al. compared DNN, RNN and LSTM neural networks for the prediction of the V/U flag and voicing. They found that the

strategy of cascaded prediction, namely using the predicted spectral features like auxiliary input increases the accuracy of excitation feature prediction [47]. Zhao et al. found that the velocity and acceleration of EMA movements are effective in articulatory-to-F0 prediction and that LSTMs perform better than DNNs in this task. Although their objective F0 prediction scores were promising, they did not evaluate their system in subjective human listening tests [48].

Another two deep learning experiments for estimating the F0 curve from ultrasound tongue images alone are proposed [49, 50]. In the literature, they presented their results for DNN-based F0 estimation from ultrasound images [50]. In contrast with others worked with EMG signals, the input articulatory representation contains no information directly related to vocal fold vibration. They applied a 2-stage DNN-based approach where one machine learning model seeks to estimate the voicing feature, while another one seeks to predict the F0 value for voiced frames. During the evaluation (synthesis) step, the outputs of the two DNNs are merged. It was achieved by taking the output value of the F0 predictor network where the voicing network decided in favor of voicing and returning a constant value for frames judged to be unvoiced. In the experiments, they attained a correlation rate of 0.74 between the original and the predicted F0 curve. And in subjective listening tests the subjects could not distinguish between the sentences synthesized using the DNN-estimated or the original F0 curve and ranked them as having the same quality. However, in the previous experiments, only a single F0 estimation algorithm based on Idiap [51] was implemented [49].

Here I extended the study by investigating different robust F0 estimation algorithms: Yaapt, Rapt, DIO and YIN. In contrast with previous work where Idiap worked as a continuous pitch algorithm that implemented with a continuous vocoder, the new four algorithms are discontinuous and implemented with a discontinuous vocoder. I discovered in my experiments that all discontinuous algorithms got better values than Idiap (being the baseline of the current thesis) in objective and subjective measurements.

## 4.2 Methodology

### 4.2.1 Data Acquisition

Two Hungarian male and two female subjects with normal speaking abilities were recorded while reading sentences aloud (altogether 209 sentences each); and the data of a female speaker was used in my current experiments. The sentences are divided into two distinct sets, 200 were selected for training and validation sets, 9 for the test set. The tongue movement was recorded in midsagittal orientation using the “Micro” ultrasound system of Articulate Instruments Ltd. at 82 fps. The speech

signal was recorded with a Beyerdynamic TG H56c tan omnidirectional condenser microphone. The ultrasound data and the audio signals were synchronized using the tools provided by Articulate Instruments Ltd. In the experiments below, the raw scanline data of the ultrasound was used as input data for the DNNs. The images were reduced to 64×128 pixels.

### 4.2.2 Feature Extraction and Speech Synthesis

The general workflow of ultrasound-based silent speech interface is shown in Figure 4.1. I applied the SPTK vocoder for the analysis and synthesis of speech (http://sp-tk.sourceforge.net). The speech signal was low-pass filtered and resampled to 22 050 Hz. The F0 curve was extracted by Idiap, Yaapt, Rapt, Dio and Yin, respectively. I extracted 12 Mel-Generalized Cepstrum-based Line Spectral Pair (MGC-LSP) features along with the gain, which resulted in a 13-dimensional feature vector. This vector served as the training target during DNN training. In the synthesis phase, I replaced all parameters required by the synthesizer by the estimates produced by the DNN. The vocoder generated an impulse-noise excitation according to the F0 parameter and applied spectral filtering using the MGC-LSP coefficients and a Mel-Generalized Log Spectral Approximation (MGLSA) filter [55] to reconstruct the speech signal.

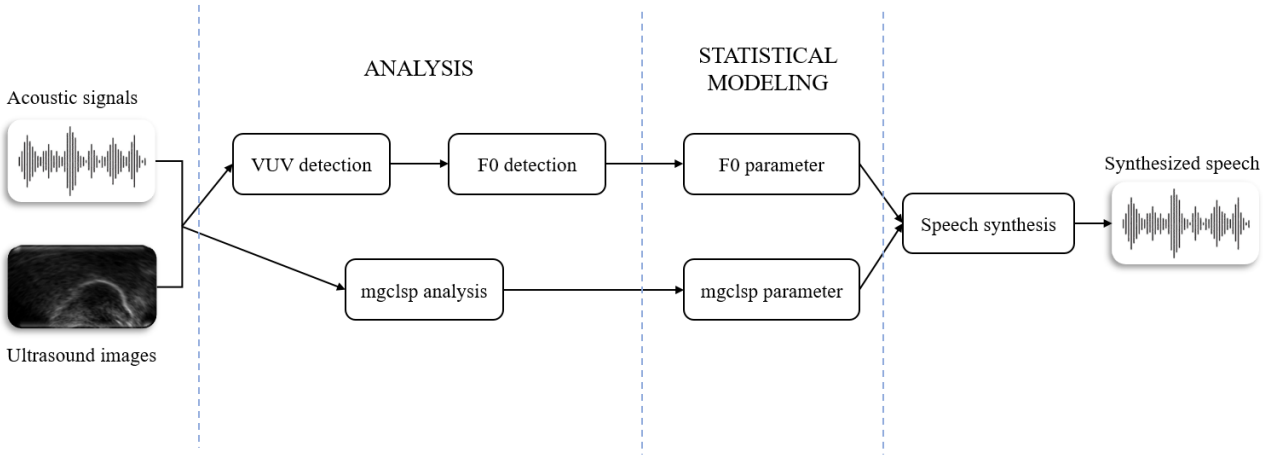



**Figure 4.1: General workflow of UTI system**

### 4.2.3 DNN-based Fundamental Frequency Estimation

DNNs were used in two major machine learning components, one dedicated to making the voiced/unvoiced decision, while the role of the second was to estimate the actual F0 value for voiced frames. The first tasks, since V/U decision for each frame has a binary output, it was treated as a classification task. While working on the same input images, the second DNN seeks to learn the F0 curve. This second task was viewed as a regression problem, and it was trained with the voiced segments from the training data. The outputs of the two DNNs were merged during the evaluation

(synthesis) step. For Idiap, this is achieved by taking the output value of the F0 predictor network where the voicing network decided in favor of voicing and returning a constant value for frames judged to be unvoiced. For Yaapt and another three algorithms, only those predicted F0 values from voiced frames are used.

I trained DNNs with 5 hidden layers of 1000 ReLU neurons. The F0 parameter was predicted together with the gain and the 12 LSP parameters. This DNN contained 14 linear neurons in its output layer. The network trained for the binary U/V decision task had the same structure, but with a binary classification output layer.

## 4.3 Objective and Subjective Measurements

### 4.3.1 Objective measurements

In order to measure how synthesized speech closed to original recorded speech, five objective measurement methods were applied. The synthesized speech used predicted F0 and predicted mgclsp parameter. 5 metrics were selected to be the measurement index. A short introduction of them are shown below.

**a)** IS

IS (Itakura–Saito) is an LPC-based (linear predictive coding) measure. It is a measure of the difference between an original spectrum and an approximation of that spectrum. The IS measure is defined as [56]

$$d_{IS}(\vec{a}_p, \vec{a}_c) = \frac{\sigma_c^2}{\sigma_p^2}\left(\frac{\vec{a}_p R_c \vec{a}_p^T}{\vec{a}_c R_c \vec{a}_c^T}\right) + \log\left(\frac{\sigma_c^2}{\sigma_p^2}\right) - 1 \tag{4.1}$$

where $\sigma_p^2$ and $\sigma_c^2$ are the LPC gains of the clean and processed signals, respectively. The smaller value is better.

**b)** LLR

LLR (log-likelihood ratio) [56] is also an LPC-based measure. It is the spectral envelope difference between the input signal and the predicted signal. It defined as

$$d_{LLR}(\vec{a}_p, \vec{a}_c) = \log\left(\frac{\vec{a}_p R_c \vec{a}_p^T}{\vec{a}_c R_c \vec{a}_c^T}\right) \tag{4.2}$$

where $\vec{a}_c$ is the LPC vector of the clean speech signal, $\vec{a}_p$ is the LPC vector of the processed enhanced speech signal, and $R_c$ is the autocorrelation matrix of the noise-free speech signal. The smaller value is better.

**c)** CEP

CEP (cepstrum distance measures) [58] provides an estimate of the log spectral distance between two spectra and it computed as follows

$$d_{CEP}(\vec{c}_c, \vec{c}_p) = \frac{10}{\log 10}\sqrt{2\sum_{k=1}^{p}[c_c(k) - c_p(k)]^2} \quad (4.3)$$

Where $\vec{c}_c$ and $\vec{c}_p$ is are the CEP coefficient vectors of the noise-free and processed signals, respectively.

The smaller value is better.

**d)** fwSNRseg

FWSEG (frequency-weighted segmental SNR) [59]. It is a time-domain measure. It computed using the following equation:

$$fwSNRseg = \frac{10}{M}\sum_{m=0}^{M-1}\frac{\sum_{j=1}^{K} W(j,m)\log_{10}\frac{X(j,m)^2}{(X(j,m)-\hat{X}(j,m))^2}}{\sum_{j=1}^{K} W(j,m)} \quad (4.4)$$

where W(*j, m*) is the weight placed on the *j*th frequency band, K is the number of bands, M is the total number of frames in the signal, X(*j, m*) is the critical-band magnitude (excitation spectrum) of the clean signal in the *j*th frequency band at the *m*th frame, and $\hat{X}(j, m)$is the corresponding spectral magnitude of the enhanced signal in the same band. The bigger value is better.

**e)** Estoi

ESTOI (Extended ShortTime Objective Intelligibility) [60]. It calculates the correlation between the temporal envelopes of clean and processed speech. The smaller value is better.

### 4.3.2 Subjective listening test

In order to find out which investigated model is closer to natural speech, I conducted an online MUSHRA-like (Multi-Stimulus test with Hidden Reference and Anchor) listening test [60]. The advantage of MUSHRA is that it allows the evaluation of multiple samples in a single trial without breaking the task into many pairwise comparisons. My aimed to compare natural and synthesized baseline sentences with the synthesized sentences using another four discontinuous F0 extraction algorithms.

2 reference variants were synthesized for each sentence of the listening test. To have an upper glass ceiling, I synthesized sentences using the original F0 curve (natural in figure 4.2). To have a

benchmark lower anchor version, I synthesized sentences using a constant F0 (const F0 in figure 4.2), where the V/U network predicted the voicing of the actual ultrasound images.

Five sentences were selected for the test, which is not included in the training database. All sentences appeared in randomized order (different for each listener). In the MUSHRA test, the listeners had to rate the naturalness of each stimulus in a randomized order relative to the reference (which was the natural sentence), from 0 (highly unnatural) to 100 (high natural).

## 4.4 Results of Objective Measurements

Table 4.1 list the results of objective measurements (note that our goal is to minimize IS, LLR and CEP, while maximizing fwSNRseg and ESTOI. The bold value is the best one of the method). This objective evaluation was done on 9 test data which are not included in the training data.

Comparing the baseline with others, we observe that:

a) All discontinuous algorithms get better value than the baseline in each metrics, which means that F0 predicted by discontinuous algorithms with discontinuous vocoder have better performance than the baseline.

b) Yaapt performs well in each metrics. Although Yaapt has shown it's strong performance in many applications, it is still surprise to see that Yaapt are the best one in each metric.

c) We see that Rapt is the second-best one.

**Table 4.1: Results of objective metrics**

| Method | Evaluation Metric | | | | |
|---|---|---|---|---|---|
| | IS | LLR | CEP | fwSNRseg | ESTOI |
| Idiap (baseline) | 4.4821 | 0.6078 | 4.5801 | 5.7718 | 0.3645 |
| Rapt | 1.1673 | 0.5014 | 3.9928 | 6.9196 | 0.3897 |
| Yaapt | **0.5664** | **0.4772** | **3.8166** | **7.1242** | **0.4134** |
| DIO | 1.4039 | 0.5103 | 3.9604 | 7.0647 | 0.3881 |
| YIN | 3.0025 | 0.5397 | 4.0710 | 6.8494 | 0.3754 |
| PnYIN | 1.3579 | 0.48318 | 3.8808 | 4.969 | 0.39275 |

## 4.5 Results of Subjective Listening Test

Altogether 16 listeners participated in the main test (6 females, 10 males). None of them indicated any hearing loss. The subjects were between 21-47 years (mean 24 years). On average, the whole test took 12 minutes to complete. Figure 4.2 shows the average naturalness score for these experimented algorithms. The benchmark version (const F0) achieved the lowest score, while the natural sentences (natural) were rated the highest, as expected. Comparing with other discontinuous algorithms, the baseline Idiap get the lowest score, which means all discontinuous algorithms based on predicted sentences sound more natural than baseline. We also noticed that the scores of four discontinuous algorithms are very similar. The reason might be their synthesized sentences are relatively close, and it is hard for a human being to distinguish their subtle differences. To check the statistical significance of the differences, I conducted Mann-Whitney-Wilcoxon rank-sum tests with a 95% confidence level, showing that the result of the Yin algorithm was significantly different from the baseline, while the other differences are not significant.

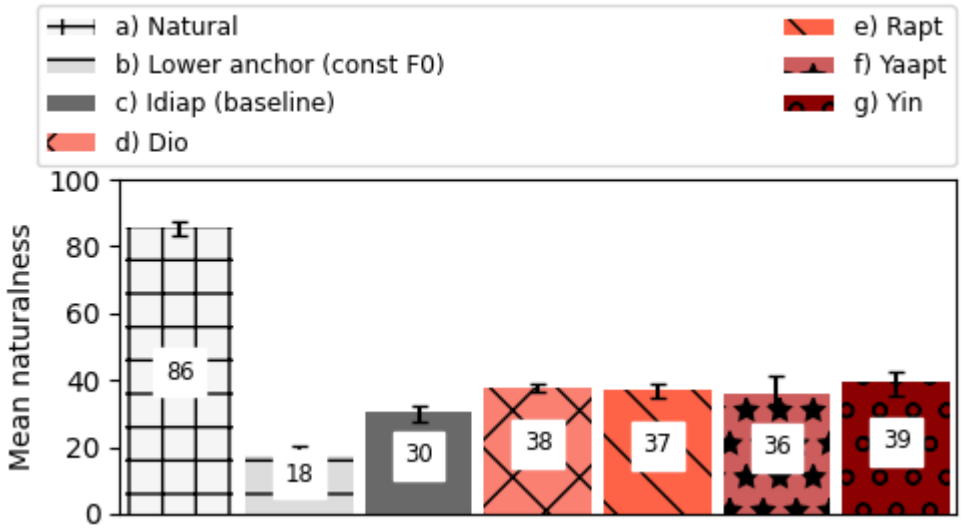



**Figure 4.2: Results of the subjective listening test. The error bars show the 95% confidence intervals.**

## 4.6 Conclusion

Here I described my experiments for comparing several discontinuous F0 estimation algorithms with a continuous baseline one in ultrasound-based articulatory-to-acoustic mapping. I used four accurate discontinuous F0 estimation algorithms to predict the F0 value of voiced frames. The results of objective and subjective evaluation demonstrated that F0 predicted by discontinuous algorithms and the synthesized sentences outperform the one based on continuous F0 (baseline). The experiments were run on the voice of only one Hungarian female speaker. In the future, I plan to repeat the experiments with more speakers (both male and female) and also with English data. Besides, it will be worth to apply recurrent neural networks to take into account the sequential nature of articulatory and speech data. For a practical Silent Speech Interface, it will be necessary to apply speaker adaptation techniques, i.e. in the future I plan to test how the UTI-to-F0 algorithms trained on one speaker work with other speakers or with real silent articulation.

# 5 Effects of F0 Estimation Algorithms on Ultrasound-based Slient Speech Interfaces Using English (UXTD) Corpus

In the previous experiment, the effects of 5 F0 estimation algorithms on UTI system were conducted and got convincing results. However, the previous experiment only implemented based on one Hungarian male speech data. The synthesized speech data based on Hungarian is a hindrance for a non-Hungarian researcher. And it is worth to extend the input experiment data to gather more convincing evidence. Thus in this work, English corpus Ultrax Typically Developing (UXTD) was used in the input data.

## 5.1 Methodology

### 5.1.1 Data acquisition

The dataset used in experiments is come from UltraSuite Repository (A repository of ultrasound and acoustic data from child speech therapy sessions) [61]. In this experiment only Ultrax Typically Developing (UXTD) was used. UXTD is a dataset of 58 typically developing children between 11/2011–10/2012. The data was recorded in the laboratory using the Articulate Assistant Advanced software (AAA). All sessions were conducted by a speech and language therapist (SLT), and both the children and the therapists spoke English with a standard Scottish accent. All therapists were female.

In this experiment, speech data of 3 children were selected from the UXTD databases. The sentences are divided into two distinct sets, 200 were selected for training and validation sets, 10 for the test set. The images were reduced to 64×103 pixels.

### 5.1.2 Feature Extraction and Speech Synthesis

The workflow is the same with the previous experiment (see section 4.2.2)

### 5.1.3 DNN-based Fundamental Frequency Estimation

DNN configuration also the same with previous experiment (see section 4.2.3).

## 5.2 Objective Measurements

In order to measurement quality of synthesized speech, the same objective measurement methods were conducted in this experiment as well. Table 5.1 listed the results. Please note that our

goal is to minimize IS, LLR and CEP, while maximizing fwSNRseg and ESTOI. The bold value is the best one of that column.

**Table 5.1: Results of objective metrics**

| Method | Evaluation Metric | | | | |
|---|---|---|---|---|---|
| | IS | LLR | CEP | fwSNRseg | ESTOI |
| Idiap (baseline) | 4.2747 | 0.77296 | 4.9900 | 4.8240 | 0.10973 |
| Rapt | 4.3325 | 0.74821 | 4.9462 | 4.7909 | 0.10275 |
| Yaapt | 7.5721 | 0.77864 | 5.0437 | 4.8209 | 0.09277 |
| DIO | 3.3168 | 0.75483 | **4.8970** | **5.0135** | **0.12290** |
| YIN | **2.8590** | 0.74334 | 4.9608 | 4.8080 | 0.10912 |
| PnYIN | 5.1035 | **0.73297** | 4.9386 | 4.8827 | 0.08815 |

We observe that:

a) Yaapt is not the best one this time.

b) In general, DIO could be seen as the best one. It has the best value in 3 metrics.

c) PnYIN got the best value in LLR

d) Compare to Hungarian corpus, the differences between the result value are smaller. Except IS, their results of all other metrics are pretty close.

## 5.3 Conclusion

Experiment with English corpus UXTD was successful. The objective metrics demonstrate pretty different results with a previous experiment with Hungarian corpus. And their value of objective metrics is similar. One of the reason might be related to the training data. Since the English corpus used from UTXD were recorded with children, the quality of speech waveform are not the same with Hungarian corpus. In this English corpus, the recording scenario was that the children were practicing the articulation of English words. One recorded speech waveform only recorded one word or sometimes only reading a single vowel or consonant. However, in Hungarian corpus, the speaker is speaking a completed sentence. So this makes the total valid voiced frames of English training data are not the same with Hungarian training data.

It is possibly affected by the quality of the training data, the synthesized speech is not that clear and natural. So the subjective listening test was not conducted. The experiment also only used waveform of a female speaker, in the future we could conduct experiment on both male and female speaker dataset.

# 6 Effects of F0 Estimation Algorithms on Ultrasound-based Slient Speech Interfaces Using English (TaL1) Corpus

In section 5, English corpus UXTD was used in the experiment. However, speech waveform in UXTD only contain one word or a single vowel. In the end of my thesis work, I was notified that the UltraSuite repository updated with data from adult speakers and in this dataset each waveform contains a whole sentence. This dataset is called Tongue and Lips corpus (TaL1). It is worth to extend my experiment with this dataset.

In this experiment, Tongue and Lips corpus (TaL1) were used in the input data. TaL1 is a single-speaker dataset with data of one professional voice talent, a male native speaker of English. In this dataset, every recorded speech waveform is a whole sentences.

## 6.1 Methodology

### 6.1.1 Data acquisition

The Tongue and Lips corpus (TaL1) is come from UltraSuite Repository [61] as well. TaL1 is a single-speaker dataset with data of one professional voice talent, a male native speaker of English. The speaker was fitted with the UltraFit stabilising helmet, which held the video camera and the ultrasound probe. Data was recorded using the Articulate Assistant Advanced (AAA) software. Ultrasound was recorded using Articulate Instruments'Micro system at ~80fps with a 92$^{o}$ field of view. A single B-Mode ultrasound frame has 842 echo returns for each of 64 scan lines, giving a 64 × 842 “raw” ultrasound frame that captures a midsagittal view of the tongue. The speaker was seated in a hemi-anechoic chamber and audio was captured with a Sennheiser HKH 800 p48 microphone with a 48KHz sampling frequency at 16 bit [62].

In the experiment, the recorded audios were resampled to 22KHz and the ultrasound images were resized to 64 × 128. There are six recording sections in TaL1. I only used dataset in section “day2”, where 181 sentences were used for training and 10 for testing.

### 6.1.2 Feature Extraction and Speech Synthesis

The workflow is the same with the previous experiment (see section 4.2.2)

### 6.1.3 DNN-based Fundamental Frequency Estimation

DNN configuration also the same with previous experiment (see section 4.2.3).

## 6.2 Objective Measurements

The same objective measurement methods were conducted in this experiment as well. Please note that our goal is to minimize IS, LLR and CEP, while maximizing fwSNRseg and ESTOI. The bold value is the best one of that column.

**Table 6.1: Results of objective metrics**

| Method | Evaluation Metric | | | | |
|---|---|---|---|---|---|
| | IS | LLR | CEP | fwSNRseg | ESTOI |
| Idiap (baseline) | 6.7277 | 0.67272 | 4.4453 | 5.3645 | 0.27118 |
| Rapt | **4.6177** | **0.63142** | **4.2897** | **5.4879** | 0.27834 |
| Yaapt | 9.1106 | 0.68699 | 4.5444 | 5.2307 | 0.28533 |
| DIO | 18.4918 | 0.90911 | 5.4055 | 4.3135 | 0.28099 |
| YIN | 4.6803 | 0.63690 | 4.3115 | 5.4340 | 0.28521 |
| PnYIN | 12.7444 | 0.76592 | 4.8667 | 4.8330 | **0.28662** |

We observe that:

a) In general, Rapt could be seen as the best one.

b) PnYIN has the best value in ESTOI

c) In every metric, YIN is very close to Rapt

## 6.3 Conclusion

Experiment with English corpus TaL1 was successful. The male speaker in TaL1 speaks much more words in each waveform than UXTD dataset. In the objective measurements results we see that Rapt is the best one, and the score of YIN are very close to Rapt. In contrast with previous experiment PnYIN only shows slightly better performance when using female speech as input, this time PnYIN get the best value in ESTOI when using male speech data as input.

The quality of synthesized speech by all these algorithms are very close as all of them are discontinuous algorithm. In the objective metrics Rapt get better value than the baseline algorithm in each metrics. However, when listening to the synthesized speech, personally I could find sentences that Rapt is better than the baseline and I could find sentences the baseline is better than Rapt. The reason might be that the words the speaker saids also kind of influenced these F0 algorithms' performance.

In the future, it's worth to repeat the experiments with more speakers (both male and female), and test how the UTI-to-F0 algorithms trained on one speaker work with other speakers or with real silent articulation.

# 7 Summary

This thesis shows my recent results about several robust F0 estimation algorithms and proposed a computational feasible algorithm PnYIN on the basis of YIN. Further experiments were conducted on evaluating F0 estimation algorithms performances on two state-of-the-art speech synthesis applications Merlin and an ultrasound-based silent speech interface. On the experiments with Merlin, I found that the baseline F0 estimation algorithm DIO does not perform well in all the objective indicators while Swipe shows slightly better results than DIO. The proposed algorithm PnYIN also get a good result in one of the objective indicators. The following experiments investigated the effects of F0 estimation algorithms in the articulatory-to-acoustic conversion from ultrasound images. The results of objective and subjective evaluation demonstrated that F0 predicted by discontinuous algorithms and the synthesized sentences outperform the one based on continuous F0 (baseline). These experiments were conducted on both Hungarian corpus and English corpus.

In the future, it will be worth to repeat all experiments with more speech data (both male and female). It will be worth to investigate more F0 estimation algorithms with vocoders and their performance in text-to-speech synthesis and articulatory-to-acoustic conversion.

# 8 Acknowledgement

I would first thank my thesis supervisor Tamás Gábor Csapó and co-supervisor Mohammed Salah Al-Radhi. Thanks for their help and advices when I stuck in any kind of issue. Their guidance and support is indispensable along the way.

I thank the listeners for participating in the subjective listening test.

The Titan X GPU for the deep learning experiments was donated by the NVIDIA Corporation.

I was supported by the Stipendium Hungaricum scholarship and the Chinese Scholarship Council.

# References


[1] Dutoit, Thierry. *An introduction to text-to-speech synthesis*. Vol. 3. Springer Science & Business Media, 1997.

[2] Thomas Hueber, Elie-laurent Benaroya, Bruce Denby, and Gérard Chollet, “Statistical Mapping Between Articulatory and Acoustic Data for an Ultrasound-Based Silent Speech Interface,” in *Proc. Interspeech*, Florence, Italy, pp. 593–596, 2011.

[3] Cheveigné, Alain de, and Hideki Kawahara. "Comparative evaluation of F0 estimation algorithms." *Seventh European Conference on Speech Communication and Technology*. 2001.

[4] Isewon, Itunuoluwa, O. J. Oyelade, and O. O. Oladipupo. "Design and implementation of text to speech conversion for visually impaired people." *International Journal of Applied Information Systems* 7.2 (2012): 26-30.

[5] Kawahara, Hideki, and Masanori Morise. "Technical foundations of TANDEM-STRAIGHT, a speech analysis, modification and synthesis framework." *Sadhana* 36.5 (2011): 713-727.

[6] Bruce Denby, Tanja Schultz, Kiyoshi Honda, Thomas Hueber, James M. Gilbert, and Jonathan S. Brumberg, “Silent speech interfaces,” *Speech Communication*, vol. 52, no. 4, pp. 270–287, 2010.

[7] Bruce Denby and Maureen Stone, “Speech synthesis from real time ultrasound images of the tongue,” in *Proc. ICASSP*, Montreal, Quebec, Canada, pp. 685–688, 2004.

[8] Thomas Hueber, Gérard Bailly, and Bruce Denby, “Continuous Articulatory-to-Acoustic Mapping using Phone-based Trajectory HMM for a Silent Speech Interface,” in *Proc. Interspeech*, Portland, OR, USA, pp. 723–726, 2012.

[9] Aurore Jaumard-Hakoun, Kele Xu, Clémence Leboullenger, Pierre Roussel-Ragot, and Bruce Denby, “An Articulatory-Based Singing Voice Synthesis Using Tongue and Lips Imaging,” in *Proc.Interspeech*, pp. 1467–1471, 2016.

[10] Jun Wang, Ashok Samal, and Jordan Green, “Preliminary Test of a Real-Time, Interactive Silent Speech Interface Based on Electromagnetic Articulograph,” in *Proc. SLPAT*, pp. 38–45, 2014.

[11] Florent Bocquelet, Thomas Hueber, Laurent Girin, Christophe Savariaux, and Blaise Yvert, “Real-Time Control of an Articulatory-Based Speech Synthesizer for Brain Computer Interfaces,” *PLOS Computational Biology*, vol. 12, no. 11, pp. e1005119, nov 2016.

[12] Keigo Nakamura, Matthias Janke, Michael Wand, and Tanja Schultz, “Estimation of fundamental frequency from surface electromyographic data: EMG-to-F0,” in *Proc. ICASSP*, Prague, Czech Republic, pp. 573–576, 2011.

[13] João Freitas, Artur Ferreira, Mário A T Figueiredo, António Teixeira and Miguel Sales Dias, “Enhancing multimodal silent speech interfaces with feature selection,”in *Proc. Interspeech*, Singapore, Singapore, pp. 1169–1173, 2014.

[14] Csapó, Tamás Gábor, et al. "Synchronized speech, tongue ultrasound and lip movement video recordings with the “Micro” system." in *Proc. CAPSS2017*, pp. 48, 2017.

[15] Ribeiro, Manuel Sam, et al. "TaL: a synchronised multi-speaker corpus of ultrasound tongue imaging, audio, and lip videos." *arXiv preprint arXiv:2011.09804*, 2020.

[16] Zahorian, Stephen A., and Hongbing Hu. "A spectral/temporal method for robust fundamental frequency tracking." *The Journal of the Acoustical Society of America* 123.6 (2008): 4559-4571.

[17] Talkin, David, and W. Bastiaan Kleijn. "A robust algorithm for pitch tracking (RAPT)." *Speech coding and synthesis* 495 (1995): 518.

[18] Kasi, Kavita, and Stephen A. Zahorian. "Yet another algorithm for pitch tracking." *2002 ieee international conference on acoustics, speech, and signal processing*. Vol. 1. IEEE, 2002.

[19] De Cheveigné, Alain, and Hideki Kawahara. "YIN, a fundamental frequency estimator for speech and music." *The Journal of the Acoustical Society of America* 111.4 (2002): 1917-1930.

[20] Camacho, Arturo, and John G. Harris. "A sawtooth waveform inspired pitch estimator for speech and music." *The Journal of the Acoustical Society of America* 124.3 (2008): 1638-1652.

[21] Xiao-Dan Mei, Jengshyang Pan, Sheng-He Sun, “Efficient algorithms for speech pitch estimation”, *Intelligent Multimedia, Video and Speech Processing, 2001*, pp. 421-424, 2001

[22] Hui, Li, Bei-qian Dai, and Lu Wei. "A pitch detection algorithm based on AMDF and ACF." *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. Vol. 1. IEEE, 2006.

[23] Sun, Xuejing. "Pitch determination and voice quality analysis using subharmonic-to-harmonic ratio." *2002 IEEE international conference on acoustics, speech, and signal processing.* Vol. 1. IEEE, 2002

[24] Hua, Kanru. "Nebula: F0 estimation and voicing detection by modeling the statistical properties of feature extractors." *arXiv preprint arXiv*:1710.11317 (2017).

[25] Morise, M., H. Kawahara, and T. Nishiura. "Rapid F0 estimation for high-SNR speech based on fundamental component extraction." *Trans. IEICEJ 93 (2010)*: 109-117, 2010.

[26] Kawahara, Hideki, Yannis Agiomyrgiannakis, and Heiga Zen. "Using instantaneous frequency and aperiodicity detection to estimate F0 for high-quality speech synthesis." *arXiv preprint arXiv*:1605.07809, 2016.

[27] Talkin, David, and W. Bastiaan Kleijn. "A robust algorithm for pitch tracking (RAPT)." *Speech coding and synthesis* 495 (1995): 518.

[28] Azarov, Elias, Maxim Vashkevich, and Alexander Petrovsky. "Instantaneous pitch estimation based on RAPT framework." *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO). IEEE*, 2012.

[29] Morise, Masanori, Fumiya Yokomori, and Kenji Ozawa. "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications." *IEICE TRANSACTIONS on Information and Systems* 99.7 (2016): 1877-1884.

[30] Kawahara, Hideki, Ikuyo Masuda-Katsuse, and Alain De Cheveigne. "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds." *Speech communication* 27.3-4 (1999): 187-207.

[31] Kawahara, Hideki, et al. "Tandem-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and aperiodicity estimation." *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008.

[32] Kawahara, Hideki, and Masanori Morise. "Technical foundations of TANDEM-STRAIGHT, a speech analysis, modification and synthesis framework." *Sadhana* 36.5 (2011): 713-727.

[33] Goodfellow, Ian, et al. *Deep learning*. Vol. 1. No. 2. Cambridge: MIT press, 2016.

[34] Le, Quoc V., et al. "On optimization methods for deep learning." *ICML*. 2011.

[35] Müller, Berndt, Joachim Reinhardt, and Michael T. Strickland. *Neural networks: an introduction*. Springer Science & Business Media, 2012.

[36] Cichocki, Andrzej, Rolf Unbehauen, and Roman W. Swiniarski. *Neural networks for optimization and signal processing*. Vol. 253. New York: wiley, 1993.

[37] Zhizheng Wu, Oliver Watts, Simon King, "Merlin: An Open Source Neural Network Speech Synthesis System" in *Proc. 9th ISCA Speech Synthesis Workshop (SSW9)*, September 2016, Sunnyvale, CA, USA.

[38] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

[39] Rabiner, Lawrence, and B. Juang. "An introduction to hidden Markov models." *ieee assp magazine* 3.1 (1986): 4-16.

[40] Black, Alan W., Heiga Zen, and Keiichi Tokuda. "Statistical parametric speech synthesis." *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP*'07. Vol. 4. IEEE, 2007.

[41] Kubichek, Robert. "Mel-cepstral distance measure for objective speech quality assessment." *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing.* Vol. 1. IEEE, 1993.

[42] Jose A. Gonzalez, Lam A. Cheah, Phil D. Green, James M. Gilbert, Stephen R. Ell, Roger K. Moore, and Ed Holdsworth, “Evaluation of a silent speech interface based on magnetic sensing and deep learning for a phonetically rich vocabulary,” In *Proc. ISCA, Interspeech*, Stockholm, Sweden, pp. 3986–3990, 2017.

[43] Jintao Jiang, Abeer Alwan, Lynne E. Bernstein, Patricia Keating, and Ed Auer, “On the correlation between facial movements, tongue movements, and speech acoustics,” *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 11, pp. 174–1188, 2002.

[44] Corine A. Bickley and Kenneth N Stevens, “Effects of a vocal tract constriction on the glottal source: experimental and modeling studies,” *Journal of Phonetics*, vol. 14, pp. 373–382, 1986.

[45] John R. Westbury and Patricia A. Keating, “On the naturalness of stop consonant voicing,” *Journal o Linguistics*, vol. 22, pp. 145–166, 1986.

[46] Diener, Lorenz, Tejas Umesh, and Tanja Schultz. "Improving Fundamental Frequency Generation in EMG-to-Speech Conversion Using a Quantization Approach." *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019.

[47] Zheng-Chen Liu, Zhen-Hua Ling, and Li-Rong Dai, “Articulatory-to-acoustic conversion with cascaded prediction of spectral and excitation features using neural networks,” in *Proc. Interspeech*, San Francisco, CA, USA, pp. 1502–1506, 2016.

[48] Cenxi Zhao, LongbiaoWang, Jianwu Dang, and Ruiguo Yu, “Prediction of F0 based on articulatory features using DNN,” in *Proc. ISSP*, Tienjin, China, 2017.

[49] Tamás Gábor Csapó, Mohammed Salah Al-Radhi, Géza Németh, Gábor Gosztolya, Tamás Grósz, László Tóth, Alexandra Markó, “Ultrasound-based Silent Speech Interface Built on a Continuous Vocoder”, in *Interspeech*, pp. 894-898, 2019.

[50] Tamás Grósz, Gábor Gosztolya, László Tóth, Tamás Gábor Csapó, and Alexandra Markó, “F0 Estimation for DNN-Based Ultrasound Silent Speech Interfaces,” in *Proc. ICASSP*, pp. 291-295, 2018.

[51] Philip N. Garner, Milos Cernak, Petr Motlicek, “A simple continuous pitch estimation algorithm,” *IEEE Signal Processing Letters*, vol. 20, no. 1, pp. 102–105, 2013.

[52] Rogers, Matthew, et al. "Systems and methods for speech preprocessing in text to speech synthesis." *U.S. Patent Application* No. 12/240,397.

[53] Hua, Kanru. "Nebula: F0 estimation and voicing detection by modeling the statistical properties of feature extractors." *arXiv preprint arXiv:1710.11317,* 2017.

[54] Hideki Kawahara, Yannis Agiomyrgiannakis, Heiga Zen. YANG VOCODER: Yet-ANother-Generalized VOCODER. https://github.com/google/yang_vocoder

[55] Imai, Satoshi, Kazuo Sumita, and Chieko Furuichi. "Mel log spectrum approximation *(MLSA) filter for speech synthesis." Electronics and Communications in Japan (Part I: Communications)* 66.2 (1983): 10-18.

[56] Schuyler R. Quackenbush, Thomas Pinkney Barnwell and Mark A. Clements, *Objective Measures of Speech Quality*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1988.

[57] Kitawaki Nobuhiko, Nagabuchi Hiromi, and Itoh Kenzo, “Ob-jective quality evaluation for low bit-rate speech coding sys-tems”, *IEEE Journal on Selected Areas in Communications*, vol. 6, pp. 262–273, 1988

[58] Tribolet, J. M., Peter Noll, B. McDermott, and R. Crochiere. "A study of complexity and quality of speech waveform coders." In *ICASSP'78. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 586-590. IEEE, 1978.

[59] Jesper Jensen and Cees H. Taal, "An Algorithm for Predicting the Intelligibility of Speech Masked by Modulated Noise Maskers," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2009–2022, 2016.

[60] "ITU-R Recommendation BS.1534: Method for the subjective assessment of intermediate audio quality," 2001.

[61] Eshky, Aciel, et al. "UltraSuite: a repository of ultrasound and acoustic data from child speech therapy sessions." *arXiv preprint arXiv:1907.00835*, 2019.

[62] Ribeiro, Manuel Sam, et al. "TaL: a synchronised multi-speaker corpus of ultrasound tongue imaging, audio, and lip videos." *arXiv preprint arXiv:2011.09804*, 2020.

[63] Jianfen Ma, Yi Hu and Philipos C. Loizou, "Objective measures for predicting speech intelligibility in noisy condi-tions based on new band-importance functions", Journal of the Acoustical Society of America, vol. 125, no. 5, pp. 3387-3405, 2009.

# List of Figures

# List of Tables

# Annex

## 1. Code of Nebula

```
function y = preprocess(x, dither_level = 0.05, dc_cutoff = 50 / 4000)
  xsqr_intg = cumsum(x .^ 2);
  xrms = sqrt((xsqr_intg(257:end) - xsqr_intg(1:end - 256)) / 256);
  xrms = [ones(128, 1) * xrms(1); xrms; ones(128, 1) * xrms(1)];
  peak = max(xrms);
  thrd = peak * dither_level;
  x = dcnotch(x, dc_cutoff);
  y = x + (xrms < thrd) .* randn(size(x)) .* (thrd - xrms);
end


function y = dcnotch(x, cutoff)
  a1 = - 2.0 * cos(pi * cutoff);
  a0 = 8.0 * cos(pi * cutoff) - 7.0;
  r = (-a1 - sqrt(a1 ^ 2 - 4.0 * a0)) / 2.0;
  a = [1.0, -r];
  b = [1.0, -1.0];
  y = filtfilt(b, a, x);
end
```

Where x is input single-channel audio wave.

## 2. Code of low pass filter

```
function y = lowp(x, f1, f3, rp, rs, Fs)
  wp = 2*pi*f1/Fs;
  ws = 2*pi*f3/Fs;
  [n,~] = cheblord(wp/pi, ws/pi, rp, rs);
  [bx1,az1] = cheby1(n, rp, wp/pi);
  [h,~] = freqz(bz1, az1, 256, Fs);
  h = 20*log10(abs(h));
  y = filter(bz1, az1, x)
end
```

where

x: input single-channel audio wave

Fs: sample rate

f1: passband cutoff frequency

f3: stopband cutoff frequency

rp: an attenuation of frequencies in the passband

rs: an attenuation of frequencies in the stopband