Efficient Initialization of Linear System for Parameter Estimation of HNM Speech Model

Kanru Hua

March 16, 2015

Abstract

In the context of speech synthesis, HNM (Harmonic Noise Model) models speech signal with a harmonic component and a filtered noise component. The common approach to estimate the harmonic component involves solving a linear system comprising of a Toeplitz matrix and a vector. However the initialization of this linear system is relatively computationally expensive, comparing to the matrix inverse operation. This paper presents an optimized initialization method. By exploiting the similarity between the matrix elements and FT (Fourier Transform), the time complexity is reduced from quadratic to linearithmic, without sacrificing precision.

Background

This essay is based on part of the author's personal research in an attempt to build a singing voice synthesis system. The author, Kanru Hua (Candidate number: 001458-0045, Shanghai Pinghe School), has been self-teaching speech signal processing since 4 years ago. His research interests are speech analysis, speech synthesis and machine learning.

Readers who are not familiar with the context of this essay may refer to [1] and [4]. In addition, graphical visualization of each step in the proposed method is included in the Appendix to help understand.

1 Introduction

HNM[1] is a widely used signal model for speech synthesis. In HNM model, time domain speech signal is represented as the sum of a time-varying harmonic signal and a time-varying filtered white noise signal. The former comprises of a few (around 30 to 80) sinusoidal harmonics, each of different amplitudes and phases, while the later can be modeled by filter coefficients, or more generally spectral envelope. Once the harmonic and noise parameters are found, the synthesis of speech is simply adding up sinusoids and filtered white noise. The model is noted for its flexibility and synthesis quality. Pitch and time-scale modification can be achieved by scaling the harmonic frequencies and amplitudes. Applications in speech synthesis system have shown that HNM synthesis backend is able to produce high quality speech superior to other techniques[2].

This paper focuses on the estimation of harmonic parameters, in the analysis process of a HNM model. The original approach estimates harmonic phases and amplitudes under a least-squares criterion which minimizes the local square error between synthesized harmonic signal and the given speech signal. The analysis is carried out frame-wise, i.e., around a chain of time instants.

Since the speech is assumed to be harmonic, the frequency of each harmonic is an integer multiple of the fundamental frequency. Under this assumption the least-squares problem can be described by a linear system of the form $\mathbf{R}x = \mathbf{b}$, where $\mathbf{R}$ is a Toeplitz matrix, $\mathbf{b}$ is a column vector, and x is the unknown vector containing amplitudes and phases information. As pointed out in the original paper[1], this equation can be solved efficiently using Levinson-Durbin recursion. However, the initialization of $\mathbf{R}$ and $\mathbf{b}$ is often more time-consuming than solving the matrix inverse.

We propose an efficient method to initialize the above matrix and vector elements. The elements of $\mathbf{R}$ and $\mathbf{b}$ are interpreted as sampled spectrums of the DTFT (Discrete Time Fourier Transform) of the analysis window and windowed signal, respectively. Then elements of $\mathbf{R}$ can be computed by sampling the DTFT of a squared Hamming window, and $\mathbf{b}$ can be efficiently computed by applying NUFFT (Non-Uniform Fast Fourier Transform)[3] to the windowed signal, with proper normalization.

The proposed method is able to produce the identical result as the original approach, with totally negligible numerical errors. For the computation of $\mathbf{R}$, time complexity is reduced from $O(n^2)$ to $O(n)$; for computing $\mathbf{b}$, time complexity is reduced from $O(n^2)$ to $O(n \log n)$.

The rest of this paper is organized as follows. Section 2 reviews the original approach of the parameter estimation problem. Section 3 reformulates the problem in terms of resampled DTFT spectrums. The fast initialization of matrix $\mathbf{R}$ is derived in Section 4. This is done by using the closed-form of the DTFT of a squared Hamming window. Initialization of vector $\mathbf{b}$ is also presented in Section 4, with a brief description of NUFFT. Section 5 summarizes the proposed method and suggests some future research directions.

2 The Original Approach

This section briefly reviews the least-square problem formulation and the original initialization method by directly evaluating matrix elements. To keep it consistent, same notations as in [1] are used throughout this paper.

In a HNM model, the sinusoidal components are assumed to be harmonic and stationary around local time instants,[1]

$$a_k(t) = a_k(t_a^i) \tag{1}$$

$$f_0(t) = f_0(t_a^i) \tag{2}$$

where $a_k(t)$ is a function of the amplitude of the k-th harmonic at time t and $f_0(t)$ is the function of fundamental frequency at time t. t_a^i is the time center of the i-th analysis frame.

The harmonic component around time instant t_a^i is,

$$\hat{h}(t) = \sum_{k=1}^{L} a_k(t_a^i) \cos(2\pi k f_0(t_a^i)(t - t_a^i) + \phi_k(t_a^i)) \tag{3}$$

where L is the number of harmonics; $\phi_k(t_a^i)$ is the phase of the k-th harmonic at time instant t_a^i in radians.

The optimization objective is to minimize the square error between harmonic signal $\hat{h}(t)$ and a given speech signal $s(t)$. Since the noise component in HNM is defined as the difference between speech signal and harmonic signal, i.e., $s(t) - \hat{h}(t)$, they are not considered in the estimation of harmonic component[2]. In practice, the error needs to be windowed around t_a^i because HNM model assumes local stationary characteristic of speech. Hamming window is chosen in order to put more emphasis on the error close to t_a^i. The objective is defined as,

[1] Uniform sampling rate is assumed.

[2] In other words, the noise component is minimized under least squares criterion.

$$\{a_k^*(t_a^i), \phi_k^*(t_a^i)\} = \underset{a_k(t_a^i),\phi_k(t_a^i)}{\arg\min} \sum_{t=t_a^i-N}^{t_a^i+N} \Big(w_{2N+1}(t)(s(t)-\hat{h}(t))\Big)^2 \tag{4}$$

$$w_N(t) = \alpha + 2\beta\cos(\frac{2\pi t}{N-1}) \tag{5}$$

where N is the period length (the nearest integer to $\frac{1}{f_0(t_a^i)}$); $w_N(t)$ is the function of a Hamming window of length N. According to the definition of a Hamming window, $\alpha = 0.54$ and $\beta = 0.23$. Note that the analysis window has an odd length of $2N+1$.

Setting the derivative of (4) to 0, we can obtain a set of equations which lead to the optimal parameters. However, it is suggested to modify (3) by taking out the phase term and replacing $a_k(t_a^i)$ with a complex number $A_k(t_a^i)$ to simplify the solving step,

$$\hat{h}(t) = \sum_{k=-L}^{k=L} A_k(t_a^i)e^{j2\pi k f_0(t_a^i)(t-t_a^i)} \tag{6}$$

and $A_k(t_a^i) = \frac{1}{2}a_k(t_a^i)e^{j\phi k(t_a^i)}$. It's worth pointing out that $A_k(t_a^i) = A_{-k}^*(t_a^i)$ and when $k=0$, an additional DC term is introduced.

Here we leave out the intermediate steps in [1] and present the final linear system derived from (4) and (6),

$$\mathbf{R}x = \mathbf{b} \tag{7}$$

where $\mathbf{R}$ is a $(2L+1)\times(2L+1)$ Toeplitz matrix with elements r_{ik}. To keep it simple, the t_a^i terms are taken away,

$$r_{ik} = \sum_{t=-N}^{N} w_{2N+1}^2(t)e^{j2\pi(i-L-1)f_0 t - j2\pi(k-L-1)f_0 t} \tag{8}$$

It is easy to show that $r_{ik} = r_{i+p,k+p}$, so $\mathbf{R}$ is a Toeplitz matrix.

$\mathbf{b}$ in (7) is a $(2L+1)\times 1$ vector with elements b_k as

$$b_k = \sum_{t=-N}^{N} w_{2N+1}^2(t)s(t)e^{-j2\pi(k-L-1)f_0 t} \tag{9}$$

The linear system in (7) can be initialized by computing (8) and (9) for all i and k. Because $\mathbf{R}$ is a Toeplitz matrix, it has only $2L+1$ unique elements that can be denoted as

$$r_{ik} = r_l = \sum_{t=-N}^{N} w_{2N+1}^2(t)e^{-j2\pi t f_0 l} \mid_{l=k-i,-2L\leq l\leq 2L} \tag{10}$$

Thus the initialization in the original approach requires $2L+1$ iterations for both $\mathbf{R}$ and $\mathbf{b}$, each of which involves $2N+1$ complex exponential operations. Hence the time complexity for initialization is of the order $O(LN)$, which is approximately the same order as that for inversing a $(2L+1)\times(2L+1)$ Toeplitz matrix using Levinson-Durbin algorithm.

3 Fourierian Interpretation of Matrix Elements

Based on the observation that equation (9) and (10) show similarity to the Discrete Fourier Transform of $w_{2N+1}^2(t)$ and $w_{2N+1}^2(t)s(t)$, respectively, this section establishes an interpretation of r_l and b_k as resampled DFT spectrums.

The $2N+1$ points DFT of a squared Hamming window is,

$$W_{2N+1}(k) = \sum_{t=-N}^{N} w_{2N+1}^2(t) e^{-j2\pi \frac{kt}{2N+1}} \tag{11}$$

Here the frequency index k is assumed to be continuous, in the DTFT sense. However the unit of k is in bins instead of radians, so that we can easily compare this equation to (10). The only difference between (10) and (11) is the exponent part. By scaling k, r_l can be represented using $W_{2N+1}(k)$,

$$r_l = W_{2N+1}\left((2N+1)f_0 l\right) \tag{12}$$

which means that r_l is equivalent to resampling $W_{2N+1}(k)$ by a factor of $(2N+1)f_0$. Note that $N \approx \frac{1}{f_0}$, so r_l is roughly a half-downsampled version of $W_{2N+1}(k)$ when $f_0 \ll 1$, which holds in most of the cases. But this approximation lacks in precision and is simply presented to help get the picture.

In the same manner b_k is derived as

$$b_k = X_{2N+1}\left((2N+1)f_0(k-L-1)\right) \tag{13}$$

$$X_{2N+1}(k) = \sum_{t=-N}^{N} w_{2N+1}^2(t) s(t) e^{-j2\pi \frac{kt}{2N+1}} \tag{14}$$

where $X_{2N+1}(k)$ is the DFT (assumed continuous frequency) of the speech signal multiplied by a squared Hamming window.

4 Efficient Initialization based on Fourier Transform

In this section efficient methods for computing r_l and b_k are presented respectively. Although applying FFT to (possibly zero-padded) $w_{2N+1}^2(t)$ and $w_{2N+1}^2(t)s(t)$ signals and then using sinc interpolation to compute r_l and b_k is possible, very high order sinc kernel is needed to get a reasonable precision, at the expense of time complexity. For r_l and b_k, we propose different methods for their initialization in the following text.

4.1 Computation of r_l

Since equation (11) is independent of the input signal $s(t)$ and fundamental frequency f_0, and a Hamming window is the sum of a constant term and a trigonometry function, it is easy to derive the closed form of the DFT of a squared Hamming window. Then the closed-form is plugged into (12) so that r_l can be directly computed by evaluating the closed-form expression.

Recall the definition (5) of a symmetric Hamming window,

$$w_N(t) = \alpha + 2\beta \cos(\frac{2\pi t}{N-1}) \tag{15}$$

Then $w_N^2(t)$ is,

$$\begin{aligned}
w_N^2(t) &= \alpha^2 + 4\beta^2 \cos^2(\frac{2\pi t}{N-1}) + 4\alpha\beta \cos(\frac{2\pi t}{N-1}) && (16)\\
&= \alpha^2 + 4\beta^2 \frac{\cos(\frac{4\pi t}{N-1}) + 1}{2} + 4\alpha\beta \cos(\frac{2\pi t}{N-1}) && (17)\\
&= \alpha^2 + 2\beta^2 + 2\beta^2 \cos(\frac{4\pi t}{N-1}) + 4\alpha\beta \cos(\frac{2\pi t}{N-1}) && (18)
\end{aligned}$$

Let $\omega = 2\pi \frac{k}{N}$, the DTFT of $w_N^2(t)$ (assume N is *odd* in our case) is,

$$W_N(\omega) = \sum_{t=-(N-1)/2}^{(N-1)/2} [\alpha^2 + 2\beta^2 + 2\beta^2 \cos(\frac{4\pi t}{N-1}) + 4\alpha\beta \cos(\frac{2\pi t}{N-1})]e^{-j\omega t} \tag{19}$$

$$\begin{aligned} &= (\alpha^2 + 2\beta^2) \sum_t e^{-j\omega t} + 2\beta^2 \sum_t \cos(\frac{4\pi t}{N-1}) e^{-j\omega t} \\ &\qquad + 4\alpha\beta \sum_t \cos(\frac{2\pi t}{N-1}) e^{-j\omega t} \end{aligned} \tag{20}$$

$$\begin{aligned} &= (\alpha^2 + 2\beta^2) \sum_t e^{-j\omega t} + \beta^2 \sum_t \left(e^{j4\pi \frac{t}{N-1}} + e^{-j4\pi \frac{t}{N-1}}\right) e^{-j\omega t} \\ &\qquad + 2\alpha\beta \sum_t \left(e^{j2\pi \frac{t}{N-1}} + e^{-j2\pi \frac{t}{N-1}}\right) e^{-j\omega t} \end{aligned} \tag{21}$$

$$\begin{aligned} &= (\alpha^2 + 2\beta^2) \sum_t e^{-j\omega t} + \beta^2 \left(\sum_t e^{-j(w - \frac{4\pi}{N-1})t} + \sum_t e^{-j(w + \frac{4\pi}{N-1})t}\right) \\ &\qquad + 2\alpha\beta \left(\sum_t e^{-j(w - \frac{2\pi}{N-1})t} + \sum_t e^{-j(w + \frac{2\pi}{N-1})t}\right) \end{aligned} \tag{22}$$

where all t range from $-\frac{N-1}{2}$ to $\frac{N-1}{2}$.

At the core of (22) is an aliased sinc function, whose closed form can be derived using properties of a geometric sequence[4],

$$\sum_{t=-(N-1)/2}^{(N-1)/2} e^{-j\omega t} = \frac{e^{j\omega \frac{N-1}{2}} - e^{-j\omega \frac{N+1}{2}}}{1 - e^{-j\omega}} \tag{23}$$

$$= \frac{e^{-j\omega \frac{1}{2}} \left(e^{j\omega \frac{N}{2}} - e^{-j\omega \frac{N}{2}}\right)}{1 - e^{-j\omega}} \tag{24}$$

$$= \frac{2j \sin(\omega \frac{N}{2})}{e^{\frac{1}{2} j\omega} - e^{-\frac{1}{2} j\omega}} \tag{25}$$

$$= \frac{\sin(N \frac{\omega}{2})}{\sin(\frac{\omega}{2})} = \frac{\sin(\pi k)}{\sin(\pi \frac{k}{N})} \tag{26}$$

which is a real number and can be easily computed. We can define the above equation as the "discrete aliased sinc function",

$$\text{dasinc}_N(k) \triangleq \sum_{t=(N-1)/2}^{(N-1)/2} e^{-j\omega t} = \frac{\sin(\pi k)}{\sin(\pi \frac{k}{N})} \tag{27}$$

By plugging (27) into (22), we can finally get the closed form of $W_N(k)$, the spectrum of an odd-length squared Hamming window,

$$\begin{aligned} W_N(k) &= (\alpha^2 + 2\beta^2)\text{dasinc}_N(k) \\ &\quad + \beta^2 \left(\text{dasinc}_N(k - \frac{2N}{N-1}) + \text{dasinc}_N(k + \frac{2N}{N-1})\right) \\ &\quad + 2\alpha\beta \left(\text{dasinc}_N(k - \frac{N}{N-1}) + \text{dasinc}_N(k + \frac{N}{N-1})\right) \end{aligned} \tag{28}$$

Thus, r_l can be efficiently computed in linear time by substituting (28) into (12). Note that the above derivation can be easily adapted to any member of Generalized Hamming Window Family and Blackman-Harris Window Family, provided that the time domain signal consists of only constant term and sinusoids, or power of sinusoids. Note that when $k = nN, \forall n \in \mathrm{Z}$, the denominator of $\mathrm{dasinc}(k)$ becomes zero and the function turns invalid. This is solved by adding a small constant to k. Also note that $W_N(k)$ is real because $\mathrm{dasinc}_N(k)$ is always real. So the computation of the inverse of $\mathbf{R}$ does not have to involve complex arithmetics. This is even true for the original approach (8) because two methods produce the same result.

4.2 Computation of b_k

As mentioned before, based on the resampled spectrum property of b_k elements, applying sinc interpolation to the FFT spectrum of $w^2_{2N+2}(t)s(t)$ to get b_k is viable but computationally expensive. Another idea to achieve this is to modify the FFT algorithm to support non-integer-sampled frequencies. However, non-integer frequencies do not have the odd and even property which is the key to conventional radix-based FFT algorithms[5].

This idea led us to use NUFFT (Non-Uniform sampled Fast Fourier Transform)[3] that is a generalization of this case. NUFFT supports both time and frequency domain non-uniform and non-integer sampled inputs/outputs. To illustrate its principle, recall the definition of b_k in (9), with changes on index k to simplify the exponent,

$$b_{k+L+1} = \sum_{t=-N}^{N} w^2_{2N+1}(t)s(t)e^{-j2\pi k f_0 t} \tag{29}$$

If t were to be scaled by $\frac{1}{f_0 N}$, (29) becomes a time-domain non-integer sampled but frequency-domain integer sampled DFT. Thus if interpolation is carried directly on the input signal in time domain, a standard FFT routine can be applied to give the non-integer sampled spectrum b_k. Aliasing is negligible since $\frac{1}{f_0 N} \approx 1$.

In NUFFT both time and frequency domains are allowed to be non-uniform sampled, and the interpolation is done by convolving the input with a heat kernel, followed by sampling at the desired resolution. Then a standard FFT is applied on the resampled signal. The output is multiplied by the inverse Fourier Transform of the heat kernel so as to cancel the convolution effect.

Let the time domain sampling be uniform and frequency domain sample points be $2\pi k f_0 t, t \in [-N, N]$, we have successfully utilized NUFFT to compute b_k elements. Normalization by $2N+1$ is probably needed due to implementation differences. Since the standard FFT runs in $O(n \log n)$, while the preprocessing step (convolution with heat kernel) is negligible as a result of the fast decaying property of heat kernel (so only low order kernels are required to achieve reasonable precision), the optimized computation of b_k has a time complexity of $O(n \log n)$.

5 Conclusion

An efficient method to initialize the linear system used in the least-squares estimation of HNM harmonic parameters is proposed. The Toeplitz matrix elements are computed from a symbolic DFT of squared analysis window; the vector elements are computed using NUFFT. Time complexity for initialization has been reduced by roughly a logarithmic order (a speed up between n and $\log n$ times).

For even better performance, square rooted Hamming window and MLT sines window[4] are suggested because $W_N(k)$ would then have fewer terms. It's worth pointing out that r_{ik} can also be computed using NUFFT, in the same manner as b_k. This enables us to use arbitrary window which may not be sinusoidal.

Comparing to ABS/OLA[6], the proposed method runs at a comparable speed but is able to find the global minimum of the objective function (4), and thus achieves better modeling accuracy.

The proposed method can be easily adapted for the least-squares estimation of the more general sinusoidal model or Deterministic plus Stochastic Model[7], given frequency estimate for each sinusoid. The harmonic frequency term kf_0 should be replaced by a set of sinusoid frequencies; type-3 NUFFT[3] should be used for efficient computation of b_k. However, $\mathbf{R}$ would no longer be a Toeplitz matrix. This indicates that $\mathbf{R}$ would have a quadratic number of unique elements comparing to a linear number in a Toeplitz matrix. The proposed method will then reduce the complexity for initializing $\mathbf{R}$ from $O(n^3)$ to $O(n^2)$. In such case tremendous performance improvement is achieved. But on the other hand, the inverse of $\mathbf{R}$ will also have higher complexity. Our next step is to compare the efficiency of the proposed method with a peak-picking combined with gradient descent-based parameter estimation method for sinusoidal models[8].

Appendix - Graphical Visualization of Parameter Estimation

This appendix shows an example of parameter estimation for a harmonic model to give the readers an intuition for better understanding. Part of a speech signal (figure 1) is input into the parameter estimation algorithm. The sampling frequency is 44100Hz; the fundamental at the analysis instant is 142.3Hz. The period is hence $\frac{44100}{142.3} \approx 310$ samples long and the frame size is twice the period length, as shown in figure 1.

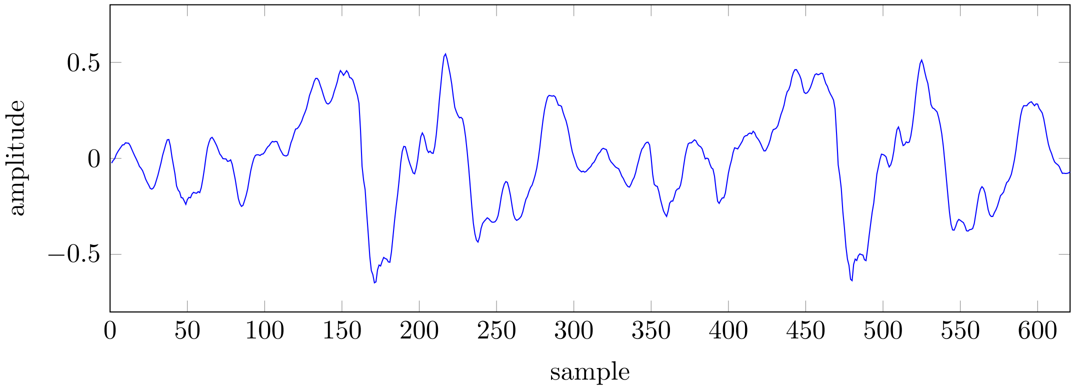


Figure 1: input signal around an analysis time instant

The next step is to generate $\mathbf{R}$ by resampling the DTFT of a squared analysis window. We first show the DTFT of a squared hamming window before resampling in figure 2.

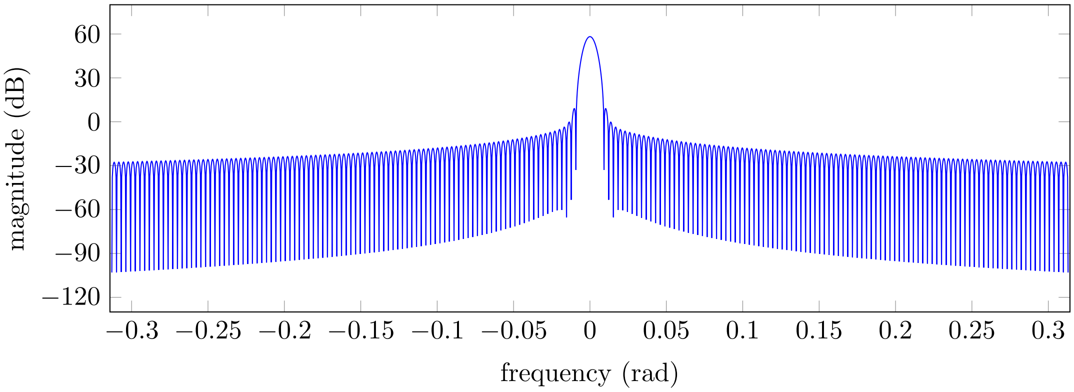


Figure 2: DTFT of a squared hamming window

Then the resampled DTFT spectrum r_l based on (12) is shown in figure 3 below where the number of harmonics is set to 80. We can see that r_l are chosen such that the side lobes are suppressed.

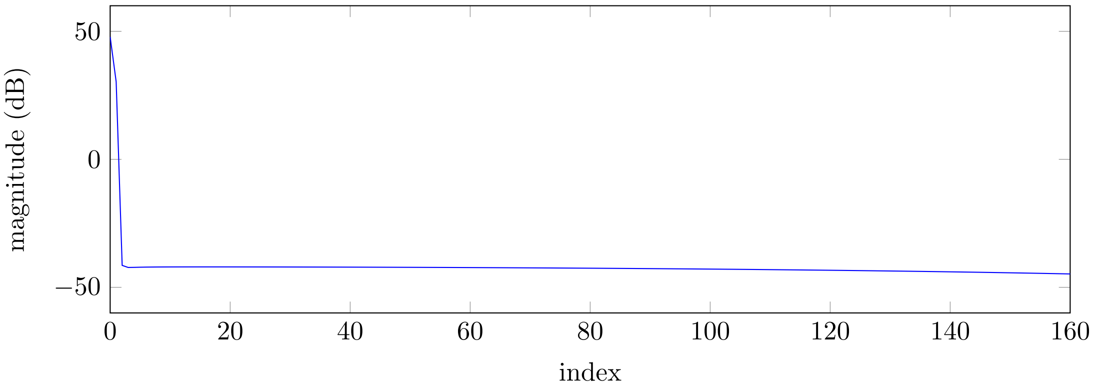


Figure 3: resampled DTFT of a squared hamming window

While b_k are computed using NUFFT as a non-integer frequency resampled DFT of the windowed speech signal. Accordingly, we show the DFT spectrum and the resampled DFT spectrum in figure 4 and figure 5 respectively.

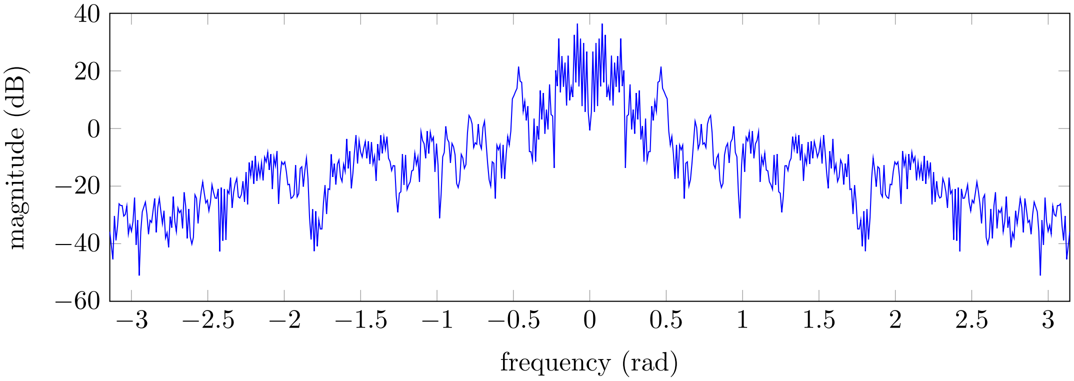


Figure 4: DFT of the windowed speech signal

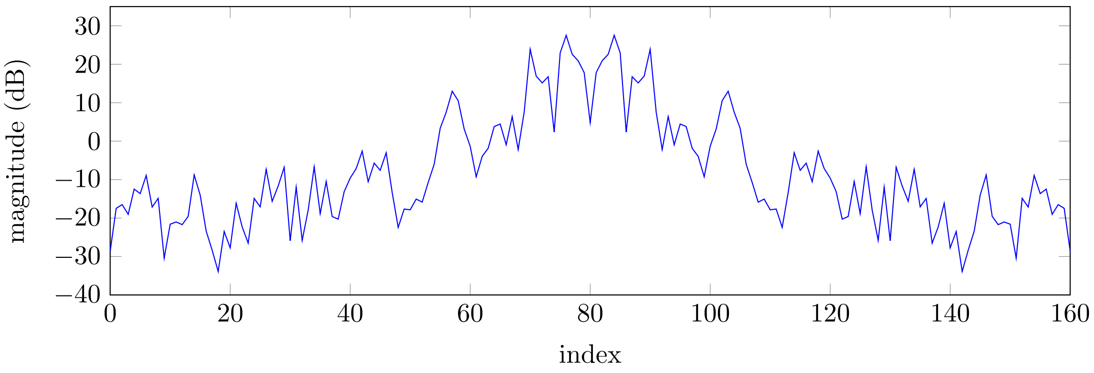
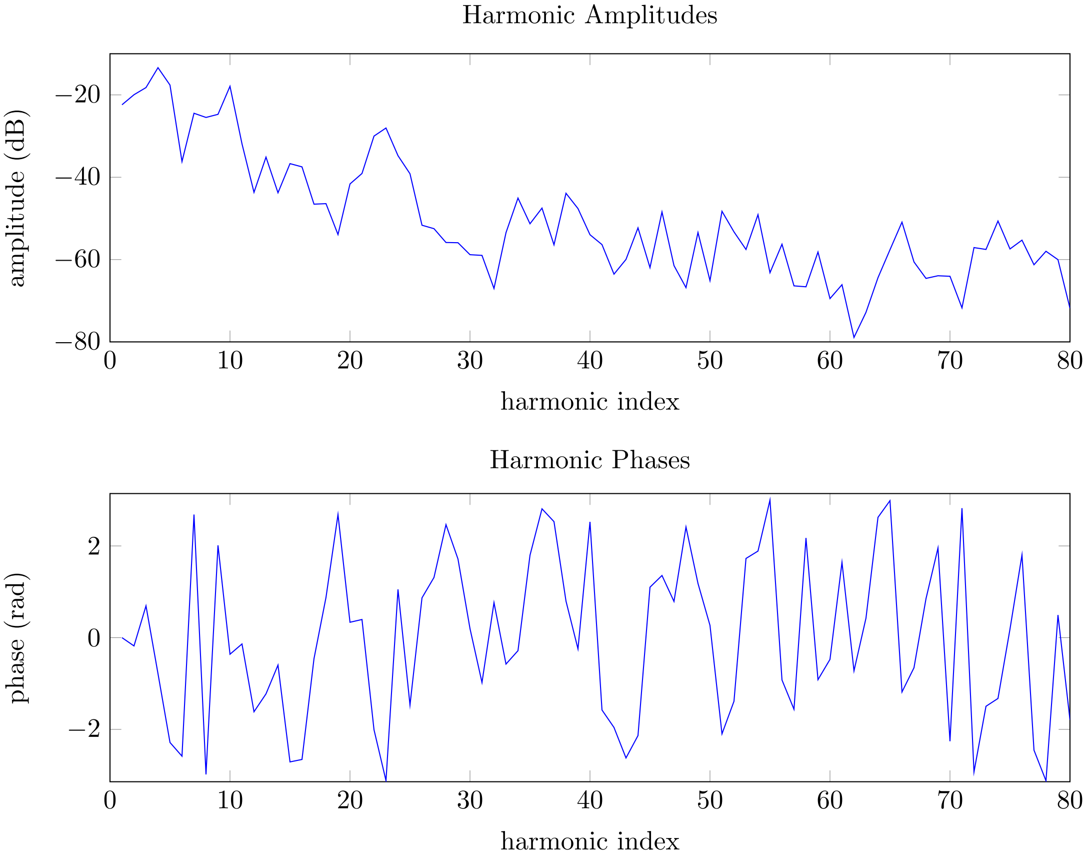


Figure 5: resampled DFT of the windowed speech signal

Finally, solving the linear system (7) we get the amplitude and phase of each harmonic as shown in figure 6.


Harmonic Amplitudes

Harmonic Phases

Figure 6: estimated parameters of harmonics

References

[1] Stylianou, Yannis. “Harmonic plus noise models for speech, combined with statistical methods, for speech and speaker modification.” Diss. Ecole Nationale Suprieure des Tlcommunications, 1996.

[2] Syrdal, Ann, et al. “TD-PSOLA versus harmonic plus noise model in diphone based speech synthesis.” Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on. Vol. 1. IEEE, 1998.

[3] Greengard, Leslie, and June-Yub Lee. “Accelerating the nonuniform fast Fourier transform.” SIAM review 46.3 (2004): 443-454.

[4] Smith, Julius O. “Spectral Audio Signal Processing.” W3K Publishing. ISBN 978-0-9745607-3-1.

[5] Cooley, James W., and John W. Tukey. “An algorithm for the machine calculation of complex Fourier series.” Mathematics of computation 19.90 (1965): 297-301.

[6] George, E. Bryan, and Mark JT Smith. “Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model.” Speech and Audio Processing, IEEE Transactions on 5.5 (1997): 389-406.

[7] Serra, Xavier, and Julius O. Smith. “Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition.” Computer Music Journal (1990): 12-24.

[8] Hua, Kanru. “A method to improve the extraction quality of periodic component of speech”. Patent Application. CN201410457379. 2014.