Abstract

This paper presents a detailed analysis and comparison of the Short-Time Fourier Transform (STFT) and Thomson’s multitaper method for frequency estimation of musical signals from a classical guitar. We show that more accurate frequency estimates can be obtained by taking into account the frequencies in a small neighbourhood around the identified frequency peaks. We also demonstrate that the multitaper method yields better frequency estimates than those from the STFT while the extra computation time required is almost negligible.

1. Introduction

The estimation of frequencies in sound signals is one of the important areas in signal processing. For musical signals in particular, of interest is not only the estimation of the frequencies of the musical notes being played but also the start and finish times of these notes. The output of this frequency estimation process is vital in many musical signal processing applications, such as automatic music transcription [6, 8, 1, 4], timbre analysis (e.g., [2]), melody extraction (e.g., [12]). The Short-time Fourier transform (STFT) is currently the most popular method employed in musical signal analysis. It involves firstly splitting the musical signals into (often overlapping) time segments; the discrete Fourier transform (DFT) is then applied to each segment independently. The end result is a spectrogram on which further analysis can be carried out. With a sampling rate of 44.1 kHz in most current digital musical signals, the highest frequency value that can be recovered, according to the Nyquist-Shannon sampling theory, is 22.05 kHz. If a segment of the musical signal has $N$ samples, then the DFT returns the Fourier transform of $N$ frequency values that are equi-distant apart over the interval $(-22.05, 22.05)$ kHz. Thus, the longer is the segment the better is the frequency resolution and the poorer is the time resolution. One must therefore do a trade-off between the accuracy of frequency estimates and the accuracy of time estimates by settling on a suitable value for $N$.

The splitting of the input signal into segments is equivalent to overlaying a rectangular window over the signal. It is well known that the Fourier transform of a rectangular window is a sinc function. Multiplying the signal by a rectangular window function in the time domain is equivalent to convolving the Fourier transform of the signal by the sinc function, resulting in the problem known as spectral leakage. In 1982, Thomson [11] proposed a nonparametric technique known as the multitaper method (MTM) to improve the estimation of the power spectral density (PSD) of signals (see also [7]). The MTM uses a bank of carefully designed optimal bandpass filters that have some nice properties. The method involves a time-bandwidth parameter that determines the number of tapers to be used.

In this paper, we investigate the use of Thomson’s MTM on musical signals and present a detailed analysis and comparison of its performance with the standard STFT. We test both methods on synthesized sine waveforms as well as pre-recorded musical signals played on a classical guitar. In Section 2, we present background information about STFT and MTM. In Section 3, we outline the steps involved in processing for the dominating frequencies of the signal in the frequency domain. Experimental results are presented in Section 4. Discussion and conclusion are given in Sections 5 and 6.

2. Background

The equal temperament scale that is commonly adopted in Western music has each octave divided into 12 semitones equidistant apart on the logarithmic scale. The frequency value (in Hz), $\omega$, of each semitone is determined by an integer value $n$ via the formula $\omega = 2^{n/12} \times 440$, where $n = 0$ gives the note A4 (440 Hz, the frequency of standard tuning forks). For the classical guitar, $n$ varies from $-29$, giving the note E2 of 82.41 Hz (the 6th open string of the classical guitar), to about 14 (note B5 of 987.77 Hz). Although
the range of frequencies of the classical guitar is small compared to other instruments such as the piano, when each fundamental note is played, the first three overtones are usually clearly audible, giving a higher range of frequencies.

Let \( f(t_n) \), where \( t_n = n\Delta t \), for \( n = 0, 1, \ldots, M \) be the musical signal sampled at \( \Delta t \) seconds apart. For a sampling rate \( s = 44.1 \text{ kHz} \), \( \Delta t \approx 2.267 \times 10^{-5} \) second. In order to estimate the start and finish times of musical notes in the signal, it is necessary to divide the signal into segments and process them separately. From here on, we assume that the entire signal is divided into segments of \( N \) samples, where \( N << M \). Adjacent segments may be disjoint; however, it is common to have a percentage of overlap between them.

### 2.1. Short-time Fourier Transform

WLOG, the sampled time instants in each segment of the musical signal can be relabelled as \( t_n \), for \( n = 0, 1, \ldots, N - 1 \). For each segment of \( N \) samples, the Discrete Fourier transform can be written as

\[
F(\omega) = \sum_{n=0}^{N-1} f(t_n) \exp[-i2\pi \omega(t_n - t_{N-1}/2)],
\]

where \( \omega \) is the frequency variable, which takes on discrete values equally spaced over the interval \((-s/2, s/2]\). The amplitude, \( |F| \), is called the Fourier spectrum (or simply the spectrum).

### 2.2. Multitapers

In this subsection, a brief review of the multitaper method is presented. Interested readers should refer to the classical paper by Thomson [11] for more detail. For signals that satisfy the condition of being wide-sense stationary (WSS), i.e., the first and second moments of the signal do not vary with respect to time, the multitaper method can be applied to estimate the power spectral density of the signal for further analysis. In general, if we look at the entire piece of music, the signal does not satisfy the condition of WSS; however, if we look at a small window only then this condition is satisfied. This means that the multitaper method can be applied in musical signal analysis and \( f(t) \) can be expressed as

\[
f(t) = \int_{-s/2}^{s/2} \exp[i2\pi \nu(t - t_{N-1}/2)] d\nu,
\]

where \( d\nu \) has zero first moment while its second moment is related to the power spectral density \( S(\nu) \) via

\[
E([d\nu(\nu)]^2) = S(\nu) d\nu.
\]

Equation (2) is called the spectral representation of \( f(t) \).

To understand how the multitaper method works, we need to employ the well-known Dirichlet kernel, which is defined by

\[
D(\nu) = \sum_{n=0}^{N-1} \exp[-i2\pi \nu(t_n - t_{N-1}/2)].
\]

Substituting (2) into (1) gives

\[
F(\omega) = \int_{-s/2}^{s/2} \sum_{n=0}^{N-1} \exp[-i2\pi(\omega - \nu)(t_n - t_{N-1}/2)] d\nu.
\]

The key idea here is that the Dirichlet kernel, \( D(\cdot) \), has eigenfunctions \( \{ U_k(N,W; \nu) \mid k = 1, \ldots, N \} \) for a given value of \( N \) and \( W \), where \( N \) is the number of samples and \( W \) is the width of the main lobe (which is normally of the order \( 1/N \)). As a result, the parameters \( N \) and \( NW \) (the product of \( N \) and \( W \)) are often used to describe the kernel and its eigenfunctions. Typical values for \( NW \) are 4s, 2.5s, etc. These eigenfunctions, defined in the frequency domain, are known as the discrete prolate spheroidal wave functions (DPSWFs). The corresponding eigenvalues are:

\[
1 > \lambda_0(N,W) > \cdots > \lambda_{N-1}(N,W) > 0:
\]

\[
\int_{-W}^{W} D(\omega - \nu) U_k(N,W; \nu) d\nu = \lambda_k(N,W) U_k(N,W; \omega),
\]

for \( k = 0, 1, \ldots, N - 1 \). Furthermore, and somewhat surprisingly, the DPSWFs are orthogonal over both the ranges \((-s/2, s/2]\) and \([-W, W]\). That is,

\[
\int_{-W}^{W} U_k(N,W; \nu) U_l(N,W; \nu) d\nu = \delta_{kl}
\]

(7)

\[
\int_{-s/2}^{s/2} U_k(N,W; \nu) U_l(N,W; \nu) d\nu = \delta_{kl}.
\]

(8)

It should now be clear why this is called the multitaper method – the DPSWFs defined a set of orthogonal tapers to provide protection against leakage. The zero-th order taper \( U_0(\nu) \) has the greatest concentration of energy. The more tapers that are used, the more estimates of the power spectrum are available, resulting in a reduction of variance. However, larger \( NW \) values can cause more spectral leakage and thus more biased spectral estimates.

In practical implementation, often used are the real discrete prolate spheroidal sequences (DPSSs), \( u_k(N,W,t) \), which are the normalized inverse Fourier transform of the DPSWFs:

\[
u_k(N,W,t) = \frac{1}{\epsilon_k \lambda_k(N,W)} \int_{-W}^{W} U_k(N,W; \nu) \exp[i2\pi \nu(t - t_{N-1}/2)] d\nu,
\]

where \( \epsilon_k = 1 \) if \( k \) is even, \( \epsilon_k = i \) otherwise. The DPSSs are also known as the Slepian sequences, named after Slepian [10]. Being the inverse Fourier transform of the DPSWFs, the Slepian sequences are real functions defined in the time domain. The first three Slepian sequences,
$u_k(N, W; t)$ with $NW = 4s$, for $k = 0, 1, 2$, are shown in Fig. 1. The subscript $k$ is known as the order of the sequence. Each Slepian sequence acts as a window function over the signal before the Fourier transform is applied. We can see from Fig. 1 that Slepian sequences of higher orders cover regions away from the centre of the signal. The number of zero-crossings of each sequence is governed by the order of the sequence.

![Figure 1: The first three Slepian sequences for NW = 4s. All the three eigenvalues $\lambda_k$, for $k = 0, 1, 2$, are very close to 1. The horizontal axis in the plot corresponds to the time duration of one segment of the musical signal.](image)

For a given value of $NW$, the total number of tapers is $2NW - 1$. Thus, one needs to select an $NW$ value that best suits the type of applications at hand. Large $NW$ values produce power spectra with wider and fewer peaks, making it easy to detect the dominant frequencies present in the input signal. The downside of using large $NW$ values is that the power spectra of nearby frequencies may not be distinguishable because they are merged into a single peak. Small $NW$ values, on the other hand, improve the resolution and allow the frequencies to be estimated more accurately. However, the power spectra will have many local maximum peaks, making it difficult to isolate those prominent ones. For automatic detection of frequencies present in the input signal, one can combine a small set of $NW$ values ranging from large to small.

One way to combine the power spectra from the tapers to get the resultant PSD is to compute the weighted sum of the power spectra with the weighting factors being the eigenvalues of the corresponding tapers. It is an expensive process to compute the Slepian sequences. Fortunately, the Slepian sequences can be pre-computed once the values of $N$ and $NW$ are determined.

### 3. Spectrum and PSD processing

Whether the STFT or the multitaper method is used, further processing must be conducted to detect the prominent frequencies and their amplitudes in each segment of the signal. In this paper, we propose the following processing steps to be carried out to improve the frequency estimate. Although the description below refers to the spectrogram only, if the STFT is used, then the processing is applied to the spectrogram; however, if the multitaper method is used, then the processing is applied to the PSD over each segment.

- **Thresholding.** This step suppresses the insignificant spectral values. A threshold value, $v$, is automatically selected using the following criterion:

$$v = \max_{\omega} \{|F(\omega)|\} - c/100,$$

where $|F(\omega)|$ denotes the amplitude of the Fourier transform at $\omega$; $r = \max_{\omega} \{|F(\omega)|\} - \min_{\omega} \{|F(\omega)|\}$; $c$ is a percentage value, typically 99.0 to 99.9. All the spectral values below $v$ are set to zero. The thresholded spectrogram is used for the subsequent steps.

This way of defining the threshold in terms of a percentage is adopted in this paper because such a threshold is more flexible and adaptable to absolute change in the spectrum.

- **Locating local maxima.** A frequency $\omega_0$ is marked as a local maximum point on the spectrogram if its Fourier spectrum is larger than its two immediate values. A thresholding value around this marked local maximum point, define the neighbourhood, $\alpha(\omega_0)$, around the local maximum point $\omega_0$ as follows:

  - Given that $\omega_0$ is already marked as a local maximum point on the spectrum, define the neighbourhood, $\alpha(\omega_0)$, around $\omega_0$ as a set as follows:
    $$\alpha(\omega_0) = \{\omega_0, \omega_0 - \Delta\omega, \omega_0 + \Delta\omega\}.$$

  - Set $k \leftarrow 1$. While $|F(\omega_0 - (k+1)\Delta\omega)| < |F(\omega_0 - k\Delta\omega)|$ do:
    (i) $\alpha(\omega_0) \leftarrow \alpha(\omega_0) \cup \{\omega_0 - (k+1)\Delta\omega\}$
    (ii) $k \leftarrow k + 1$.

  - A similar operation is carried out on the other side of $\omega_0$ to produce the neighbourhood. It is not necessary that the neighbourhood is symmetric about $\omega_0$, as $\omega_0$ is determined by the uniform sampling (see Section 1) of the discrete Fourier transform and is often at an offset from the true frequency.

- **Frequency estimation.** The spectral values of the frequencies within the neighbourhood, $\alpha(\omega_0)$, around $\omega_0$ identified in the step above can be used as weighting factors for frequency estimation. The frequency value around the local maximum point $\omega_0$ is estimated via

$$\hat{\omega}_0 = \frac{\sum_{\omega \in \alpha(\omega_0)} \omega |F(\omega)|}{\sum_{\omega \in \alpha(\omega_0)} |F(\omega)|}.$$


4. Experiments

Experiments on both synthetic musical signals and real musical signals have been conducted. Each input signal was first divided into segments of $N$ samples. In all the experiments described, consecutive segments had 50% overlap. As the same segments were passed to both the STFT and MTM, the time resolutions of the identified notes were the same for both methods. The research focus is therefore on comparison of the accuracies of the frequency estimates and the computation time taken by each method. The processing steps described in the previous section were applied to each segment of the spectrogram and PSD independently. In all the experiments, the parameter $c$ in Eq. (10) used for the thresholding step (see the previous section) was set to 99.9. Various Matlab functions in the Signal Processing Toolbox were used in the experiments. The spectrum and PSD processing steps were also written in Matlab.

4.1. Experiments on synthetic data

Synthetic musical signal can be easily produced using a combination of sine functions of various known frequency values. We have tested both methods on many synthesized data. Because of the space limit, only two experiments are presented here. The first experiment involved notes in the lower frequency range present simultaneously in the signal, whereas the second experiment involved notes in the higher frequency range. Specifically, the notes C3, D3, E3, F3, and G3 (ranging from 130.81 to 196 Hz) were synthesized in Experiment 1, while C4, D4, E4, F4, and B3 (ranging from 246.94 to 349.23 Hz) were used in Experiment 2. Table 1 summarizes the root-mean-squared errors (RMSEs) of the frequency estimates from both methods for the two experiments with various segment lengths. The frequency estimates from both methods improved as the length ($N$) of the signal increased; however, the MTM consistently gave smaller RMSEs than the STFT.

<table>
<thead>
<tr>
<th>N</th>
<th>Experiment 1</th>
<th></th>
<th>Experiment 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE (Hz)</td>
<td></td>
<td>RMSE (Hz)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STFT</td>
<td>MTM</td>
<td>STFT</td>
<td>MTM</td>
</tr>
<tr>
<td>9000</td>
<td>7.5406</td>
<td>0.6695</td>
<td>1.3458</td>
<td>0.4462</td>
</tr>
<tr>
<td>10000</td>
<td>1.3665</td>
<td>0.9733</td>
<td>0.8905</td>
<td>0.2673</td>
</tr>
<tr>
<td>11000</td>
<td>0.8708</td>
<td>0.7005</td>
<td>0.5789</td>
<td>0.1855</td>
</tr>
<tr>
<td>12000</td>
<td>0.3531</td>
<td>0.2772</td>
<td>0.5199</td>
<td>0.1386</td>
</tr>
</tbody>
</table>

Table 1: Synthetic test results for the two experiments.

4.2. Experiments on real data

Two experiments on real musical data are presented. Experiment 1 was a waltz taken from a CD that comes with a classical guitar method book by Waldron [13]. As the book contains instructions on guitar tuning, one may assume that all the recorded songs were played on a properly tuned guitar. The entire piece contains 16 bars, lasting for 28 seconds (see Fig. 2. Each segment had $N = 9000$ samples and $NW$ was set to 1.5. Note that the value of $N$ is not large enough to distinguish the low frequency notes, such as B2 and C3; however, in this waltz example, such notes were not played temporally close to each other. For visualization purposes, both the spectrogram generated by STFT and the PSD from MTM have been enhanced and are shown in Figs. 3 and 4 respectively. It is evident that, even though $NW$ was set to a small value, the profile at each column of the PSD in Fig. 4 has wider and flatter peaks than the Fourier spectrum at the corresponding column in Fig. 3.

Figure 5 illustrates the Fourier spectrum (blue curve) from STFT and the PSD from MTM (red curve) at time $t = 0.81$ seconds (within the first bar). At this time, the active notes were C3 (130.81 Hz), G3 (196 Hz), and G4 (392 Hz). In the figure, the note C4 (the first harmonic of C3) at frequency value 261.63 Hz is clearly audible. The black dashed lines indicate the true frequencies for C3, G3, and C4. It is clear that the blue curve at around C3 and C4 is slightly off to the left. Use of the peaks from the STFT.
alone definitely gave very poor frequency estimates; however, when the neighbouring frequencies were incorporated, better frequency estimates were produced for both methods. Even though the note B₂ (123.47 Hz) is not present in the first bar of the waltz, it appears as a small peak in both the spectrogram and the PSD whenever the nearby note C₃ is played. This may be a unique signature of classical guitar music that can be used for instrument classification applications. The frequency estimate from the STFT weighted average for note C₃ was clearly affected by the presence of the spurious note B₂; however, the frequency estimate from MTM for this note was much more accurate, demonstrating the advantage of using multiple tapers. Table 2 summarizes this research finding. In this particular example, the frequency estimate from the STFT deviated from the true value 130.81 Hz by 2.51 Hz whereas the frequency estimate from MTM deviated by only 1.05 Hz.

It is important to note that even though the piece was played by an expert, some notes of equal time value varied significantly in played length. Furthermore, since it is unnatural for the guitarist to deliberately time and terminate each note after it is played, the ground truth information needed for error evaluation must be manually obtained. To do so, the software Audacity was used to assist the manual marking of the start and finish times of each note. The RMSEs of the frequency estimates from STFT and MTM in this experiment were 1.53 Hz and 1.29 Hz respectively.

As expected, the MTM required extra computation time over the STFT. From the Matlab profreport function, for STFT, the spectrogram took 11.43 seconds to compute and the spectrum processing took 4.26 seconds; for MTM, the PSD took 13.83 seconds and the PSD processing took 5.98 seconds.

Experiment 2 involved a more polyphonic piece (see Fig. 6) taken from the same book. The 8 bars shown in the Figure lasted for 13.5 seconds. The values of N and N_W were set to 11800 and 1.5, respectively. The two methods took 2.06 and 2.55 seconds in total for the generation and processing of the spectrogram and the PSD. The RMSEs of frequency estimates from the two methods were 0.79 Hz and 0.77 Hz, respectively.

Table 2: The ground truth frequency values, the frequency values at the Fourier spectrum peaks, and the frequency values estimated using (11) for both methods at t = 0.81 seconds.

<table>
<thead>
<tr>
<th></th>
<th>true freq. (Hz)</th>
<th>peak freq. (Hz)</th>
<th>est. freq. (Hz)</th>
<th>est. freq. (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C₃</td>
<td>130.81</td>
<td>127.40</td>
<td>128.63</td>
<td>129.75</td>
</tr>
<tr>
<td>G₃</td>
<td>196.00</td>
<td>191.10</td>
<td>193.42</td>
<td>193.70</td>
</tr>
<tr>
<td>C₄</td>
<td>261.63</td>
<td>295.70</td>
<td>259.70</td>
<td>259.88</td>
</tr>
</tbody>
</table>

5. Discussion

The experiments presented in the previous section show that the MTM can achieve the same level of accuracy as the STFT with much smaller N values, thus allowing a better time resolution for the detected notes. From the experiments conducted, the extra computation time required for MTM is almost negligible.

The spectral leakage problem is not unrecognized in musical signal analysis. Traditionally, whenever the STFT is used to generate a spectrogram for further analysis, a window function, such as the Hamming or Hann window, is
often employed to yield a more accurate spectrum estimate. Such window functions have the shape of a scaled cosine function and behave exactly like a zero-th order taper. Indeed, if a Hamming window was used for the STFT then its RMSEs shown in Table 1 for the synthesized data would reduce. For instance, for \( N = 9000, 10000, \) and \( 12000 \) in Experiment 1, the RMSEs of STFT with a Hamming window reduced to \( 1.1079, 0.9956, \) and \( 0.1146 \) Hz, respectively. An inspection of Table 1 reveals that the last RMSE (for \( N = 12000 \)) is smaller than that from the MTM. This demonstrates that the advantage of the MTM for protecting spectral leakage at the high frequency area is not fully exposed when only the (low) fundamental frequencies are the main focus of the analysis. Thus, for low frequency analysis, it would be more advantageous to use only a few low-ordered tapers. In the example given above, if only the zero-th order taper was used, then the RMSE from the MTM for \( N = 12000 \) reduced to \( 0.1143 \) Hz, which is slightly better than that from the STFT. For this same reason, the parameter \( NW \) should be set to a small value (e.g., 1.5).

A common problem to both the STFT and MTM is that when a segment covers the boundary between two notes, both methods may detect an average frequency instead of the individual frequencies. The problem can be alleviated by using longer segments with a trade-off of reducing accuracy of start and finish times of the notes. This problem is in fact common to all other methods as well.

If our aim is to detect dominating frequencies in a signal, one may question why other approaches are not employed. Indeed, methods such as the MUSIC (MUltiple SIgnal Classification) algorithm [9] and eigenvector (EV) algorithm [3] (see also [5, Ch.13]) may be more relevant since each individual musical note produces a line (i.e., an isolated peak) on the frequency axis. Both MUSIC and EV work well when the exact dimension of the signal space is known and when the underlying noise is white. Unfortunately, our preliminary tests show that these two methods gave very poor results. The problems are twofold: (i) The exact dimension of the signal space is usually unknown in real musical signals; (ii) High frequency noises in real musical signals are always present and they are correlated with the true (and lower) frequency values in the signal. Both MUSIC and EV were badly influenced by the high frequency coloured noise in the signal and failed to detect the true lower frequency values. These two methods may still be useful if they are combined with other non-parametric methods and a pre-processing noise filtering step. Analysis on these methods is postponed to a future report. Undoubtedly, the spectrum and PSD processing described in Section 3 also plays an important role in frequency estimation, e.g., how the frequency estimates are affected by varying the parameter \( c \) in the thresholding step or by defining the neighbourhood (see Section 3) differently. Other possible future works include designing \( N \) to be a dynamic value and taking into account temporal coherence of the musical notes across segments.

6. Conclusion

We have presented in this paper a detailed analysis and comparison of the STFT and MTM for frequency estimation in musical signals. While the STFT has been the traditional method for frequency analysis, we have shown that the multitaper method is a useful alternative for musical signal analysis. The extra computation time involved in computing the PSD using the multitaper method has been demonstrated to be negligible.

References


