Abstract

In this paper, we analyze the rhythmic structures of several pieces of music. We use two different types of data reduction to reduce the signals, eliminating higher frequency information. Using algorithms written for MATLAB by Sethares and Staley, we identify dominant frequencies and periodicities using the capabilities of the Discrete Fourier Transform and the Periodicity Transform. While both the DFT and PT have certain difficulties, we conclude that each transform provides us with important information. We also touch on other applications of the analysis of rhythm.

1 Introduction

Even to the untrained eye (or ear), it is quite apparent that mathematics is at play in music. As one delves deeper, one realizes that not only is math involved in music, but that there is an inextricable connection between the world of mathematics and every single element of music—whether it be in the theory of sound waves, the physics of instruments or the structure of musical rhythm.

1.1 The Wave Equation

As a spring-board for this discussion, we will begin with what is arguably the most basic element of music: pitch. Let us, for the sake of example, consider a stringed instrument. When a violinist plucks the string of the instrument, the string is set in motion. Let \( u(x, t) \) be the function that represents the displacement of the violin string at a position \( x \) and a time \( t \). Since both ends of a violin string are anchored, for a string of length \( L \), \( u(0, t) = 0 \) and \( u(L, t) = 0 \). These are called the boundary conditions.
The motion of the string is governed by the one-dimensional wave equation which states that the second derivative of displacement with respect to position is proportional to the second derivative of displacement with respect to time. That is,

\[ a^2 u_{xx} = u_{tt} \]

where \( a \) is a constant that depends on tension and the composition of the string. Although the physical principles that govern the behavior of wind instruments are different, it is worth noting that the one-dimensional wave equation is also at work—this is why wind and stringed instruments produce similar sounds.

Through the process of separation of variables (which can be found in any standard textbook—try [1]), we can show that solutions to the wave equation are linear combinations of solutions of the form:

\[ u(x, t) = \sin\left(\frac{m\pi x}{L}\right)\left[a_m \cos\left(\frac{am\pi t}{L}\right) + b_m \sin\left(\frac{am\pi t}{L}\right)\right] \]

where \( m = 1, 2, 3, \ldots \) and \( a_m \) and \( b_m \) are constants. That is, \( u(x, t) \) can be written as a linear combination of sines and cosines. If we fix \( x \) at a value \( x_0 \), we get a function of \( t \) that predicts how the string moves at a particular position. The resulting function shows us that the movement of the string is periodic:

\[ u(x_0, t) = c_m \cos\left(\frac{am\pi t}{L}\right) + d_m \sin\left(\frac{am\pi t}{L}\right) \]

and hence the resulting sound wave is periodic. The human ear perceives periodic sound waves of frequency 20-20,000 Hz (cycles per second) as pitched sound.

### 1.2 Musical Rhythm

Our project involves the study of periodic rhythms in music. The main difference between pitch and rhythm is a difference in scale—pitches are typically measured in hundreds or thousands of cycles per second, while rhythms are measured in hundreds of cycles per minute! However, both pitch and rhythm are periodic phenomena and so we can borrow some of the traditional methods used to study pitch. As we will see later, with the help of heavy-duty data reduction, these methods will prove rather useful.

### 1.3 Discrete vs. Continuous

Here is a good point at which to stop and make an important distinction. The type of sound that we will be analyzing here is sound taken from a CD, that is, a digital recording. In other words, we will be studying discrete signals (and therefore discrete functions). Were we to analyze LP records, we would need to deal with continuous functions. Digital recordings consist of a certain amount of discrete samples taken, as opposed to LP records, which
consist of one continuous track of sound. The more samples that are taken per second when a CD is recorded, the more accurately the discrete function will resemble the continuous function. In general, we will assume that the sampling is sufficiently frequent that we don’t lose much information by using the discrete approximation. It is also worth noting that many of the techniques developed in this article (such as DFT analysis) have an analog in the continuous world.

2 A Closer Look at Periodic Functions

In this section, we will develop some general theory of discrete, periodic functions needed in our musical analysis.

2.1 An Inner Product Space

Let’s investigate discrete periodic functions of a fixed period $N$. Any discrete periodic function is of the form $f[n]$ where $n \in \mathbb{Z}$ and $f[n+N] = f[n]$ for some integer $N$, which is referred to as the period of $f$. We claim that, for a fixed period $N$, the set of all $N$-periodic discrete functions forms an inner product space with the inner product defined as:

$$\langle f, g \rangle = \frac{1}{N} \sum_{n=0}^{N-1} f[n] \overline{g[n]}$$

where $\overline{g[n]}$ indicates the complex conjugate of $g[n]$.

The reader is invited to verify that this is indeed an inner product. For a minute, let’s entertain ourselves by looking at a seemingly unrelated set, $S$, where

$$S = \{ e^{2\pi i kn/N} | k = 0, \ldots, N-1 \}.$$

First notice that the elements of $S$ are $N$-periodic. In addition, $S$ is orthonormal. To see this, take two arbitrary elements of $S$, $e^{2\pi i kn/N}$ and $e^{2\pi i ln/N}$. For $k \neq l$,

$$\langle e^{2\pi i kn/N}, e^{2\pi i ln/N} \rangle = \frac{1}{N} \sum_{n=0}^{N-1} e^{2\pi i kn/N} e^{-2\pi i ln/N} = \frac{1}{N} \sum_{n=0}^{N-1} e^{2\pi i (k-l)n/N}$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} \omega^n$$

(where $\omega^n = e^{2\pi i (k-l)/N}$)

$$= \frac{1}{N} (1 + \omega + \omega^2 + \ldots + \omega^{N-1}).$$

Notice that $(1+\omega+\omega^2+\ldots+\omega^{N-1}) = (\omega^{N-1})/(\omega - 1)$ (one can verify this by long division). Therefore, $\langle e^{2\pi i kn/N}, e^{2\pi i kl/N} \rangle = (1/N)(\omega^{N-1})/(\omega - 1) = (1/N)(e^{2\pi i (k-l)} - 1)/(\omega - 1) = 0$
because $k$ and $l$ are integers. We have shown that $S$ is an orthogonal set. In addition, $S$ is an orthonormal set, since $\langle e^{2\pi i kn/N}, e^{2\pi i kn/N} \rangle = 1$. Continuing with this apparent non sequitur, we claim that we can write any $N$-periodic discrete function $f$ as a linear combination of elements of $S$. We claim:

$$f[n] = \sum_{k=0}^{N-1} F[k] e^{2\pi i kn/N}, \quad (2)$$

where

$$F[k] = \frac{1}{N} \sum_{n=0}^{N-1} f[n] e^{-2\pi i kn/N}. \quad (3)$$

To verify this, observe that

$$\sum_{k=0}^{N-1} F[k] e^{2\pi i kn/N} = \sum_{k=0}^{N-1} \left( \frac{1}{N} \sum_{m=0}^{N-1} e^{-2\pi i km/N} \right) e^{2\pi i kn/N}$$

$$= \frac{1}{N} \sum_{m=0}^{N-1} f[m] \left( \sum_{k=0}^{N-1} e^{-2\pi i km/N} e^{2\pi i kn/N} \right)$$

By our previous argument, the expression in parentheses equals 0 if $m \neq n$ and 1 if $m = n$. Therefore the right hand side of Equation 2 equals $f[n]$.

In conclusion, we see that $S$ is an orthonormal basis for the $N$-periodic discrete functions. Since $S$ is orthonormal, this representation of each discrete function is a unique representation.

Here is an example. Let $f[n]$ be the discrete 4-periodic function defined by


Then


where

$$F[0] = \frac{1}{4} (a + b + c + d)$$
$$F[1] = \frac{1}{4} (a + bi - c - di)$$
$$F[2] = \frac{1}{4} (a - b + c - d)$$
$$F[3] = \frac{1}{4} (a - bi - c + di)$$

Although upon first glance the equations (2) may not appear to be related to the solution of the wave equation (1), Euler’s formula ($e^{i \theta} = \cos \theta + i \sin \theta$) can be used to rewrite them in a form similar to (1). Also recall that we are using discrete approximations of continuous functions.
2.2 The Discrete Fourier Transform

The representation (2) is called the Discrete Fourier Transform (DFT). The equation $F[k]$ gives the coefficients of the different frequencies represented in the musical sound. The magnitude $|F[k]| = (F[k]\overline{F[k]})^{1/2}$ of each coefficient is the strength of each frequency component.

Let’s examine these coefficients more closely. Take the four-periodic function (4) as in our first example. Since neither $F[0]$ nor $F[2]$ contain complex components, $|F[0]| = F[0]$ and $|F[2]| = F[2]$. We can also see that $|F[1]| = |F[3]| = (1/4)((a-c)^2 + (b-d)^2)^{1/2}$ So, if $f$ is 2-periodic ($a = c$ and $b = d$), then $|F[0]| = F[0] = (1/2)(a+b)$, $|F[2]| = F[2] = (1/2)(a-b)$, and $F[1] = F[3] = 0$. Likewise, if $f$ is approximately 2-periodic, that is if $a$ is close to $c$ and $b$ is close to $d$, then $F[1]$ is close to $F[3]$ which is approximately 0.

As seen in these two examples, the DFT can be used to identify prominent frequencies (pitches) in a signal, $f$, by recovering the coefficients of the basis elements. We can graph this information to get a clear picture of the different frequencies present in the signal, as in Figure 1.

3 Analyzing Musical Rhythm Using the DFT

Up until this point, we have been analyzing frequencies (pitch) using a standard tool known as the DFT. We can also employ the capabilities of the DFT to analyze rhythm. In order to do this, however, a good deal of data reduction must take place. By removing the higher frequencies (usually the melody and instrumentation), we are left with the rhythmic components of the musical piece (the percussion and strong pulses). When the DFT is applied to these lower frequencies, much information about the rhythmic structure of the piece will be revealed.

3.1 Binary Representation

The simplest form of data reduction is “binary representation,” a method created by Rosenthal [6]. Any piece of music can be rhythmically represented using a sequence of 0s and 1s, where a “1” represents the onset of a note and a “0” represents either the length of note or a rest. Take a look at the binary reduction of the childhood favorite “Yankee Doodle” to get a clearer idea:
Yankee Doodle went to town, riding on a pony.
1 0 1 0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 1 0 1 0 0 0 1 0 0 0
Put a feather in his cap and called it macaroni.
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 0 1 0 0 0
Yankee Doodle keep it up, Yankee Doodle dandy
1 0 0 1 1 0 1 0 1 0 1 0 1 0 0 0 1 0 0 1 1 0 1 0 1 0 0 0 1 0 0 0
Mind them music and the step and with the folks be handy.
1 0 0 1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 0 1 0 0 0 1 0 0 0

Notice that the quarter note (or the pulse) is represented by the four-bit string 1000. Likewise, the eighth note is the two-bit string 10, the dotted eighth note is the three-bit string 100 and the sixteenth note is the one-bit string 1. Figure 1 shows the magnitude of the coefficients in the DFT of this 128-bit binary string representation of the rhythm of “Yankee Doodle.”

![Figure 1: DFT for binary representation of “Yankee Doodle”](image)

The largest spike occurs at 32 (that is, it occurs 32 times in the piece) and represents the most prominent rhythmic structure in the song—the quarter note. The second largest spike occurs at 8, representing the eight measures the song is broken up into. Since there are 4 quarter notes in each measure and the quarter note is the pulse, we can see that “Yankee Doodle” is in 4/4 (common time).

**DFT analysis of Rejoice in the Lamb** Binary reduction can be further exploited to gain rhythmic information about music much more complex than the old standby “Yankee Doodle.” In the following example, we used a passage from Benjamin Britten’s *Rejoice in the Lamb* [2], a festival cantata which features 4 soloists, a full choir and an organ.
We translated 48 measures of the piece and wrote the treble accompaniment, the bass accompaniment, the men’s vocal part and the women’s vocal part into four separate binary vectors \( a, ab, m \) and \( w \), each of length 1152. (See Appendix A for these vectors as well as a copy of the score.) Figure 2 represents the superimposed DFTs of vectors \( a, ab, m \) and \( w \).

First observe the two largest spikes are at 288 and 192. Since our sampling of the piece contains 48 measures, frequency 288 corresponds to a rhythm that occurs six times in one measure (the eighth-note triplet) and frequency 192 corresponds to a rhythm that occurs four times in a measure (the eighth note). Similarly, the other relevant rhythmic divisions are represented here by the spikes at 144 (the quarter-note triplet), 96 (the quarter note) and 48 (the measure).

You will notice that the DFT is very adept at identifying smaller rhythmic divisions—that is, ones that occur within the measure. But what about ones with larger periods (and therefore smaller frequencies) such as melodic phrases, which may span several measures? For example, in this piece there is a rather prominent four-bar phrase which repeats throughout. This phrase is represented on the graph by the spike at the frequency 12 mark, but its magnitude is less than that of the 240 spike. There is no division of the measure into five written in the actual score, so the 240 spike is a result of the interplay between the quarter-note triplet and the eighth note—an audio-illusion of sorts. But doesn’t the much more prominent four-bar phrase deserve a larger spike?

In conclusion, these smaller frequencies are detected by the DFT analysis, but their prominence cannot be correctly inferred from the graph since they are all trapped in the tiny
region on the far left. We will use a different transform in a bit which will address this problem.

3.2 Data Reduction for Recorded Music

As mentioned earlier, CDs are a discrete representation of sound, and so, at least in theory, the DFT should prove useful in analysis. It is useful, however, only after considerable data reduction (otherwise the high frequency would mask the rhythm). By using a more sophisticated method of data reduction, we can use the DFT to analyze the musical rhythm of many musical pieces recorded on CD.

The method we will use is the “psychoacoustically motivated data reduction” described in the article *Meter and Periodicity in Musical Performance* by William A. Sethares and Thomas W. Staley [8]. The goal is “to drastically reduce the amount of data in a perceptually relevant way [8]”. Consider an arbitrary signal recorded on a CD, s, sampled at a rate of 44.1 kHz. We then partition the audio data into twenty-three 1/3 octave bands (this number was chosen because it nicely partitions the sound that humans can detect). For each of these partitions, take the first 1024 samples (remember pitch and rhythm differ in frequency by about a factor of $10^3$) and find the average magnitude. Do this for the entire song, making sure there is some overlap between “bundles of samples” to increase the precision of the approximation. In the end, we will have successfully reduced a three-minute track from over 8,000,000 bits to about 10,000 (rhythmically relevant) bits! That’s quite an improvement. For reasons that will be clear later, we will normally choose the number of samples to be a highly divisible number. The data reduction algorithm we used is found in Appendix B.

DFT analysis of “Saltarello”  The example of “Saltarello” by Dead Can Dance offers us a very clear representation of the DFT. The piece is highly synthesized, using drum machines and other electronic instruments to create a crisp, clean sound. As we will see from this example, the stricter and more exact the rhythm is, the clearer the picture of the DFT will be. The most prominent spike in this graph occurs at 1138. We can clearly see that those spikes with less frequency than 1138 are divided into 8. This is a dead giveaway that the song is either in 4 or 8. The next most prominent spike is what appears to be 1/8 the frequency of 1138 (or 142.25). If you listen to the track, you can figure out that the 1138 spike represents the rhythm created by the tambourine and frame drum (the pulse). The 142.25 spike represents the rhythm created by the bass drum that falls on each down beat (this is most likely the measure). If we make this the measure, and the 1138 spike into the eighth note, then the song can be understood to be in 4/4, common time, with the eighth note predominating.
DFT analysis of “Bird”  We can also use the DFT to analyze pieces that are not in common time. Figure 4 is the representation of the DFT for “Bird,” also by Dead Can Dance. The reader is invited to try to figure out the time signature of the piece.

The largest spike in the DFT occurs at 850. The second to next spike is the 1700, which is either an overtone of the 850 spike or another, less prominent beat that is twice as frequent at the 850 spike. The next most prominent spike occurs at 85. If the 850 spike represents the beat, the 85 spike most likely represents a measure—of 10. If we make the beat a quarter note, we can predict that the song is in 10/4. Upon hearing the song, we will find out that our prediction was indeed correct. It is interesting to note that the long birdsong introduction does not affect the clarity of the picture.

DFT analysis of “Fish”  So far, we have only used the DFT to analyze songs that are either strictly binary or highly electronic—and have produced quite tidy graphs. What about more organic, acoustic songs that may waver in rhythmic exactness? Take this rather organic track in 7/4 time called “The Fish” by the progressive rock band Yes. Figure 5 shows the DFT. It is not a simple to analyze this graph. We know the song is in 7, and so we can work backwards to try and find the relevant frequencies (the pulse is at the 413 spike and the measure is at the 59 spike). But without this knowledge, the DFT is not of great assistance. The imperfection of humans and the “give and take” of organic musicianship is at fault here, creating non-perfect periodic frequencies which in turn create the “noise”
seen in the DFT.

**DFT analysis of “America”** An extreme example of this noise can be seen in the DFT of the raucous dance number “America” from Leonard Bernstein’s West Side Story in Figure 6. There is substantial tempo variation, as well as much shouting, hooting and clapping, all of which lead to this “sloppy” DFT.

**Limitations of the DFT** The limitations of the DFT are beginning to be unveiled. In addition to its problems in dealing with non-exact music, the DFT makes it difficult to observe those periodic rhythmic structures that are not as frequent as the beat—such as phrases. Sethares and Staley [7] have provided us with a new tool, which they call the Periodicity Transform, which, while not a panacea, does address the latter problem.

### 4 The Periodicity Transform

Let $x$ be our signal. The idea behind the Periodicity Transform (PT) is that we want to find the “closest” periodic vector to $x$. We will call this closest periodic vector $x^*$. By subtracting $x^*$ from $x$, we get another vector which we will call $r$. This residual signal will
now be searched for periodicity; that is, we will now search for the “closest” periodic vector to \( r \). And so on. Finally, we will have a decomposition of \( x = x^* + r_1^* + r_2^* + \ldots \) into periodic vectors. Like the basis elements in the DFT, these periodic vectors give us an idea of the relative strengths of periodicities within \( x \).

### 4.1 The space of \( p \)-periodic vectors

Recall that \( x[k], k \in \mathbb{Z} \) is \( p \)-periodic if \( x[k + p] = x[k] \) for all \( p \). Let \( \mathcal{P} \) = all periodic vectors and let \( \mathcal{P}_p = \) all \( p \)-periodic vectors. Notice that both \( \mathcal{P} \) and \( \mathcal{P}_p \) form vector spaces since they are both closed under addition and scalar multiplication.

We now need to define a basis vector for \( \mathcal{P}_p \). The following sequence is a fitting choice:

\[
\delta^i_p[i] = \begin{cases} 
1, & \text{if } (i - s) = 0 \pmod{p} \\
0, & \text{otherwise}
\end{cases}
\]

\( \text{e.g. } \delta^1_p = \ldots, 1, 0, 0, 0, 1, 0, 0, 0 \ldots \)  Note that \( \delta^1_p, \delta^2_p \) and \( \delta^3_p \) will all just be shifts of \( \delta^0_p \).
Consider the following:

\[
\langle x, y \rangle = \lim_{k \to \infty} \frac{1}{2k+1} \sum_{i=-k}^{k} x[i] y[i]
\]

for arbitrary elements \( x, y \) in \( \mathcal{P} \). We claim that this is an inner product on \( \mathcal{P} \). The limit will always exist since if \( x \in \mathcal{P}_{p_1} \) and \( y \in \mathcal{P}_{p_2} \), \( x[i] y[i] \in \mathcal{P}_{p_1 p_2} \) since it is now \( p_1 p_2 \)-periodic. The inner product now becomes

\[
\langle x, y \rangle = \frac{1}{p_1 p_2} \sum_{i=0}^{p_1 p_2 - 1} x[i] y[i]
\]

or the average of the \( p_1 p_2 \)-periodic vector over a single period. We now have a way to measure distance: \( ||x|| = \langle x, x \rangle^{1/2} \).

Signals \( x \) and \( y \) in \( \mathcal{P} \) are orthogonal if \( \langle x, y \rangle = 0 \), and two subspaces are orthogonal if every vector in one is orthogonal to every vector in the other. Notice, however, that no two periodic subspaces \( \mathcal{P}_p \) are orthogonal since \( \mathcal{P}_{1} \subset \mathcal{P}_{p} \) for every \( p \). Moreover, \( \mathcal{P}_{np} \cap \mathcal{P}_{mp} = \mathcal{P}_p \) when \( n \) and \( m \) are mutually prime. As an example, take \( \mathcal{P}_4 \) and \( \mathcal{P}_6 \). If \( x \in \mathcal{P}_4 \cap \mathcal{P}_6 \), then \( x \in \mathcal{P}_4 \) and \( x \in \mathcal{P}_6 \). For this to be true, \( x \) must also be 2-periodic (indeed, \( p = 2 \) and \( n = 2 \), \( m = 3 \)).
4.2 Projection onto $p$-periodic subspaces

The purpose of considering an inner product space here is so we can invoke that Projection Theorem. Let $x \in \mathcal{P}$ be an arbitrary signal. A minimizing vector in $\mathcal{P}_p$ is an $x^* \in \mathcal{P}_p$ such that $\|x - x^*\| \leq \|x - x_p\|$ for all $x_p \in \mathcal{P}_p$. Therefore, $x^*_p$ is the “closest” $p$-periodic vector to the original $x$ (to which we referred earlier). We are now ready to state the Projection Theorem.

**Theorem 1** The vector $x^*$ given by

$$x^* = \alpha_0 \delta^0_p + \alpha_1 \delta^1_p + \ldots + \alpha_{p-1} \delta^{p-1}_p,$$

where $\alpha_i = p \langle x, \delta^i_p \rangle$ for $0 \leq i \leq p - 1$ is the unique minimizing vector in $\mathcal{P}_p$.

**Proof.** First, observe that $x - x^*$ is orthogonal to $\mathcal{P}_p$: For any basis element $\delta^i_p \in \mathcal{P}_p$,

$$
\langle x - x^*, \delta^i_p \rangle = \langle x, \delta^i_p \rangle - \langle x^*, \delta^i_p \rangle = \langle x, \delta^i_p \rangle - \sum_{j=0}^{p-1} \alpha_j \langle \delta^i_p, \delta^j_p \rangle \\
= \langle x, \delta^i_p \rangle - p \langle x, \delta^i_p \rangle (1/p) = 0
$$
since $\langle \delta^i_p, \delta^j_p \rangle = 0$ if $i \neq j$ and $1/p$ if $i = j$.

Let $w$ be any vector in $\mathcal{P}_p$. Then

$$
||x - w||^2 = \langle x - w, x - w \rangle \\
= \langle (x - x^*) + (x^* - w), (x - x^*) + (x^* - w) \rangle \\
= \langle x - x^*, x - x^* \rangle + 2 \langle x - x^*, x^* - w \rangle + \langle x^* - w, x^* - w \rangle \\
= ||x - x^*||^2 + ||x^* - w||^2
$$

Then $||x - w|| \geq ||x - x^*||$, with equality only if $w = x^*$. Therefore, $x^*$ is the unique minimizing vector. \qed

We will also use the notation $\pi(x, \mathcal{P}_p)$ to represent the projection of $x$ onto $\mathcal{P}_p$.

**Example** Let $x = \ldots 1, 1, 0, 1, 1, 4, 0, 2, \ldots \in \mathcal{P}_8$. The projection of $x$ onto $\mathcal{P}_2$ is the vector

$$x^*_2 = \ldots \frac{1}{2}, 2, \frac{1}{2}, 2, \frac{1}{2}, 2, \frac{1}{2}, 2, \ldots$$

and the residual is

$$r_2 = x - x^*_2 = \ldots \frac{1}{2}, -1, -\frac{1}{2}, -1, \frac{1}{2}, 2, -\frac{1}{2}, 0, \ldots$$
The projection of \( x \) onto \( P_4 \) is
\[
x_4^* = \ldots 1, \frac{5}{2}, 0, \frac{3}{2}, 1, \frac{5}{2}, 0, \frac{3}{2}, \ldots,
\]
and the residual is
\[
r_4 = x - x_4^* = \ldots 0, -\frac{3}{2}, 0, -\frac{1}{2}, 0, \frac{3}{2}, 0, \frac{1}{2}, \ldots
\]
Projecting \( r_4 \) onto \( P_2 \) gives the zero vector.

This makes sense, though, because \( r_4 \) is the original signal with all 4-periodic subsignals removed. All 4-periodic signals are necessarily 2-periodic, and so \( \pi(r_4, P_2) = 0 \). In fact, we have the following theorem, due to Sethares and Staley [7]:

**Theorem 2** Let \( r_p = x - \pi(x, P_p) \) be the residual after projecting \( x \) onto \( P_p \) and \( r_{np} = x - \pi(x, P_{np}) \) be the residual after projecting \( x \) onto \( P_{np} \). Then \( r_{np} = r_p - \pi(r_p, P_{np}) \).

**Proof.** It is clear that \( \pi(\pi(x, P_p), P_{np}) = \pi(x, P_p) \). Then
\[
r_{np} = x - \pi(x, P_{np}) = x - \pi(x, P_p) - [\pi(x, P_{np}) - \pi(\pi(x, P_p), P_{np})] = r_p - \pi(r_p, P_{np}).
\]
\[ \square \]

Sethares and Staley have come up with the following additional theorem whose proof can be found in their article.

**Theorem 3** Let \( x \) be a periodic vector and \( p \) and \( n \) be positive integers. Then
\[
\pi(x, P_p) = \pi(\pi(x, P_p), P_{np}) = \pi(\pi(x, P_{np}), P_p).
\]
\[ \square \]

The theorem shows that the order of projection of a periodic vector \( x \) onto subspaces \( P_p \) and \( P_{np} \) does not matter. This is a result of thinking of \( \pi(x, P_p) \) as an average over every \( np \)th entry in \( x \).

**Corollary 1** The projection of \( r_{np} \) onto \( P_p \) is the zero vector.

**Proof.** Using Theorems 2 and 3, we have
\[
\pi(r_{np}, P_p) = \pi(r_p, P_p) - \pi(\pi(r_p, P_{np}), P_p) = \pi(r_p, P_p) - \pi(r_p, P_p) = 0.
\]
\[ \square \]

It is advantageous at this point to take a step back and think about what it is we are actually doing here. When we project our signal \( x \) onto \( P_p \), we are stripping it of all its \( p \)-periodic components. However, the residual may still have other relevant periodicities, and so we should project this “new signal” onto other subspaces (perhaps \( P_q, P_s \ldots \)) to extract them as well.
Truncation  A word needs to be mentioned about those signals that are of non-divisible periodicities. For example, consider

$$s = \ldots, 1.9, 3, -1, 0, 2.1, 3, -0.7, 0, 3, 3, -1, 0, 2.1, \ldots \in \mathcal{P}_{13}.$$  

This periodic sequence is "almost" 4-periodic, and so it would seem to make sense to project $s$ onto $\mathcal{P}_4$. Using the technique for projection just described, we would get

$$s_4^* = \ldots, 1.2, 1.2, 1.2, 1.2, \ldots.$$  

This does not jive well with our perception of $s$ as "almost" 4-periodic. Sethares suggests truncating the signal to be of a length that is a multiple of the periodicity for which we are searching. In this example, we would truncate $s$ to now be

$$\bar{s} = \ldots, 1.9, 3, -1, 0, 2.1, 3, -0.7, 0, 3, 3, -1, 0, \ldots \in \mathcal{P}_{12}$$

and

$$\bar{s}_4^* = \ldots, 2.333, 3, -0.9, 0, \ldots.$$  

This is a much more satisfying result.

Nonuniqueness  Before going on, it is necessary to note something about the uniqueness of projection. We have seen above that as long as the periodicities are not mutually prime, the order of projection does not matter. This is not true in general. While the DFT deals with subspaces which are orthogonal, the periodic subspaces $\mathcal{P}_p$ are not orthogonal to each other. Therefore, the representation of an arbitrary signal $s$ as a linear combination of the basis elements is not unique. Furthermore, there is not a unique order to choose projection onto periodic subspaces, since different orders may yield different results.

4.3 Algorithms

At the heart of the PT is its ability to choose among these subspaces and determine the most relevant order in which to project. Sethares and Staley have put forth four algorithms: the Small-to-Large algorithm, the $M$-Best algorithm, the Best Correlation algorithm and the Best Frequency algorithm. Just as its name suggests, the Small-to-Large algorithm scans a signal for relevant periodicities beginning at $p = 2$ and continuing up to larger ones. If the percent of the total energy removed by projection onto $\mathcal{P}_{p_i}$ is greater than a given threshold, the projection is carried out. Otherwise, that periodic space $\mathcal{P}_{p_i}$ is skipped and projection onto $\mathcal{P}_{p_{i+1}}$ is attempted. Observe that a "Large-to-Small" algorithm would be useless. Using the results of Corollary 1, if we first project a signal onto a subspace $\mathcal{P}_{n_p}$, the residual will not contain any of the smaller periodicities which are its divisors, $p$. This would yield misleading data. We used this algorithm exclusively in our calculations primarily because it was the one that required the least amount of time to run. The other three algorithms took upwards of an hour to complete the calculations. We are investigating the problem.
PT analysis of *Rejoice in the Lamb*  For this example, we used the small-to-large algorithm, with binary reduction, taking 1152 samples and our threshold defined at 0.1. The following are the results:

\[ 2 \quad 4 \quad 12 \quad 96 \quad 478 \]

While the PT does not give all the inter-measure divisions offered by the DFT, it does give some. Period 4 represents the eighth-note triplet (which is very prominent in the piano accompaniment) while period 12 represents the quarter note. The exciting result is the period 96—or the four-bar phrase! The other number, 478, represents noise explained by the fact that most of the important periodicities have already been stripped from the signal.

PT analysis of “Saltarello”  For this example, we used the small-to-large algorithm, taking 9104 samples. The maximum period searched for was 800 and our threshold was defined as 0.01. The following are the results:

\[ 1 \quad 8 \quad 16 \quad 32 \quad 64 \]

and five periodicities over 500 (529, 555, 659, 681, and 775) that do not seem musically significant.

The percent energy removed after projecting onto the respective subspaces are

\[ 0.5919 \quad 0.0272 \quad 0.0195 \quad 0.0129 \quad 0.0278 \]

The norms of the other 5 residuals are less than 0.012.

Notice that periodicity represents the measure (occurs 9104/64 = 142.25 times in the piece) and periodicity 8 represents the quarter note (occurs 1138 times in the piece). From the DFT graph, we can see that there is also a 4-bar phrase. We did not yield a periodicity of 256, but this probably just requires one of the three other algorithms.

PT analysis of “Bird”  For this example, we used the small-to-large algorithm, taking 15300 samples. The maximum period searched for was 800 and our threshold was defined as 0.01. The following are the results:

\[ 1 \quad 9 \quad 18 \quad 180 \quad 701 \]
The percent energy removed after projecting onto the respective subspaces are

\[
0.4228 \quad 0.0198 \quad 0.0343 \quad 0.0139 \quad 0.0110
\]

Here, periodicity 18 represents the beat where 180 is a measure containing ten beats. Again, we can deduce that our 9-periodic rhythmic element is the eighth note. With this song, there are no real prominent multi-bar phrases, so we need not search for another threshold or algorithm.

**PT analysis of “Fish”**  For this example, we used the small-to-large algorithm, taking 9939 samples. The maximum period searched for was 800 and our threshold was defined as 0.005. The following are the results:

There are 20 periods found, the most prominent being 1, 24, 169, 172, 335, and 337. The periods are plotted versus the percent of energy they remove from the residual signals in Figure 7.

![Figure 7: PT for “Fish”](image)

The period 24 represents the prominent beat, periods 169 and 172 dance around 168, the measure and 335 and 337 straddle 336, a two-bar phrase. The reader may notice that the number of samples is not divisible by 7 (the prominent beat). This, in addition to the non-exactness of the relevant periodicities, is explained by the fact that this piece, as mentioned before, is more organic with much more rhythmic “give and take.”

**Difficulties of PT analysis**  There are some obvious problems with the PT. As mentioned earlier, many of the algorithms take a long time to complete their calculations. Another major problem, which we encountered in the example of “America,” is the fact that the signal to be analyzed by the PT must have a relatively constant tempo. Sethares is currently trying to improve his method of data reduction with “beat tracking” capabilities.
5 Conclusion

For the past year, we have been studying two different, yet intimately related transforms: the DFT and the PT. With the aid of these two transforms, we have been able to analyze the internal rhythmic structure of many pieces of music, both directly from the score and from a recording on compact disc. While all of this is exciting (at least to us), there are actually other further applications of the PT. So many things in our universe are periodic in nature: from the pulsations of stars, to the ebb and flow of the tides; from the beating of our hearts, to the rising and setting of the sun. In addition to being able to find out more about the periodicity of these phenomena, we can use these transforms to predict future behavior by searching for patterns.

References


A  Data for Britten’s *Rejoice in the Lamb*

A.1  Britten Score
A.2 Binary Reduction for Britten Score
B Data Reduction Algorithm