Chapter 2: Fourier Analysis of Signals

2.1 The Fourier Transform in a Nutshell
2.2 Signals and Signal Spaces
2.3 Fourier Transform
2.4 Discrete Fourier Transform (DFT)
2.5 Short-Time Fourier Transform (STFT)
2.6 Further Notes

Important technical terminology is covered in Chapter 2. In particular, we approach the Fourier transform—which is perhaps the most fundamental tool in signal processing—from various perspectives. For the reader who is more interested in the musical aspects of the book, Section 2.1 provides a summary of the most important facts on the Fourier transform. In particular, the notion of a spectrogram, which yields a time–frequency representation of an audio signal, is introduced. The remainder of the chapter treats the Fourier transform in greater mathematical depth and also includes the fast Fourier transform (FFT)—an algorithm of great beauty and high practical relevance.

Chapter 3: Music Synchronization

3.1 Audio Features
3.2 Dynamic Time Warping
3.3 Applications
3.4 Further Notes

As a first music processing task, we study in Chapter 3 the problem of music synchronization. The objective is to temporally align compatible representations of the same piece of music. Considering this scenario, we explain the need for musically informed audio features. In particular, we introduce the concept of chroma-based music features, which capture properties that are related to harmony and melody. Furthermore, we study an alignment technique known as dynamic time warping (DTW), a concept that is applicable for the analysis of general time series. For its efficient computation, we discuss an algorithm based on dynamic programming—a widely used method for solving a complex problem by breaking it down into a collection of simpler subproblems.
Fourier Transform

Idea: Decompose a given signal into a superposition of sinusoids (elementary signals).

Each sinusoid has a physical meaning and can be described by three parameters:

\[ f(t) = A \cdot \sin(2\pi \omega t + \varphi) \]

\[ \omega = \text{frequency} \]
\[ A = \text{amplitude} \]
\[ \varphi = \text{phase} \]

**Interpretation:**
The amplitude \( A \) reflects the intensity at which the sinusoidal of frequency \( \omega \) appears in \( f \).
The phase \( \varphi \) reflects how the sinusoidal has to be shifted to best correlate with \( f \).

**Example:** Superposition of two sinusoids

\[ f(t) = A_1 \cdot \sin(\omega_1 t + \varphi_1) + A_2 \cdot \sin(\omega_2 t + \varphi_2) \]

\[ A_1 = 1 \]
\[ \omega_1 = 1 \]
\[ \varphi_1 = 0 \]

\[ A_2 = 0.6 \]
\[ \omega_2 = 3 \]
\[ \varphi_2 = -0.2 \]

\[ A_3 = 0.4 \]
\[ \omega_3 = 7 \]
\[ \varphi_3 = 0.4 \]

**Example:** C4 played by piano

The Fourier transform of \( f \) shows the frequency content of the signal.
Fourier Transform
Example: C4 played by trumpet

\[ f \]
| Time (seconds) |
\[ |\hat{f}| \]
| Frequency (Hz) |

Fourier Transform
Example: C4 played by violine

\[ f \]
| Time (seconds) |
| Frequency (Hz) |

Fourier Transform
Example: C4 played by flute

\[ f \]
| Time (seconds) |
| Frequency (Hz) |

Fourier Transform
Example: Speech "Bonn"

\[ f \]
| Time (seconds) |
| Frequency (Hz) |

Fourier Transform
Example: Speech "Zürich"

\[ f \]
| Time (seconds) |
| Frequency (Hz) |

Fourier Transform
Example: C-major scale (piano)

\[ f \]
| Time (seconds) |
| Frequency (Hz) |
**Fourier Transform**

Example: Chirp signal

![Chirp signal example](image)

Analysis using sinusoid with 262 Hz
→ high correlation
→ large Fourier coefficient

Analysis using sinusoid with 400 Hz
→ low correlation
→ small Fourier coefficient

Analysis using sinusoid with 523 Hz
→ high correlation
→ large Fourier coefficient

**Role of phase**

![Phase analysis](image)
Fourier Transform

Role of phase

Analysis with sinusoid having frequency 262 Hz and phase $\phi = 0.05$

Analysis with sinusoid having frequency 262 Hz and phase $\phi = 0.24$

Analysis with sinusoid having frequency 262 Hz and phase $\phi = 0.45$

Analysis with sinusoid having frequency 262 Hz and phase $\phi = 0.6$

Each sinusoid has a physical meaning and can be described by three parameters:

- $\omega = \text{frequency}$
- $A = \text{amplitude}$
- $\phi = \text{phase}$

Complex formulation of sinusoids:

$$ s(t) = A \cdot \exp(2\pi i \omega t) = A \cdot (\cos(2\pi \omega t) + i \cdot \sin(2\pi \omega t)) $$

where

- $\omega = \text{frequency}$
- $A = \text{amplitude} = |c|$
- $\phi = \text{phase} = \arg(c)$

Polar coordinates:

$$ c = |c| \cdot \exp(2\pi i \phi) $$

Signal

$ f : \mathbb{R} \rightarrow \mathbb{R} $

Fourier representation

$$ f(t) = \int_{\omega \in \mathbb{R}} c_{\omega} \exp(2\pi i \omega t) \, d\omega $$

Fourier transform

$$ c_{\omega} = f(\omega) = \int_{t \in \mathbb{R}} f(t) \exp(-2\pi i \omega t) \, dt $$
**Fourier Transform**

**Signal**

$f : \mathbb{R} \rightarrow \mathbb{R}$

**Fourier representation**

$f(t) = \int_{\omega \in \mathbb{R}} c_{\omega} \exp(2\pi i \omega t) d\omega$

**Fourier transform**

$c_{\omega} = \hat{f}(\omega) = \int_{\mathbb{R}} f(t) \exp(-2\pi i \omega t) dt$

- Tells which frequencies occur, but does not tell when the frequencies occur.
- Frequency information is averaged over the entire time interval.
- Time information is hidden in the phase.

---

**Short Time Fourier Transform**

Idea (Dennis Gabor, 1946):

- Consider only a small section of the signal for the spectral analysis → recovery of time information.
- Short Time Fourier Transform (STFT)
- Section is determined by pointwise multiplication of the signal with a localizing window function.

---

**Graphical Illustration**

- Time-frequency representation showing the transition from continuous Fourier transform to short-time Fourier transform.
Window functions

- Rectangular window
- Triangular window
- Hann window
Short Time Fourier Transform

**Window functions**

- Trade off between smoothing and “ringing”

---

**Definition**

- **Signal** $f : \mathbb{R} \to \mathbb{R}$
- **Window function** $g : \mathbb{R} \to \mathbb{R}$ ($g \in L^2(\mathbb{R})$, $\|g\|_2 \neq 0$)
- **STFT**
  \[ \hat{f}_g(t, \omega) = \int_{u \in \mathbb{R}} f(u)g(u-t) \exp(-2\pi i \omega u) du = \langle f, \gamma_t, \omega \rangle \]
  with $\gamma_t, \omega(u) = \exp(2\pi i \omega(u-t))g(u-t)$ for $u \in \mathbb{R}$

---

**Intuition:**
- $R_{t, \omega}$ is “musical note” of frequency $\omega$ centered at time $t$
- Inner product $\langle f, R_{t, \omega} \rangle$ measures the correlation between the musical note $R_{t, \omega}$ and the signal $f$

---

**Time-Frequency Representation**

**Spectrogram**

- Chirp signal and STFT with Hann window of length 50 ms
Time-Frequency Representation
Chirp signal and STFT with box window of length 50 ms

- Size of window constitutes a trade-off between time resolution and frequency resolution:
  - Large window: poor time resolution, good frequency resolution
  - Small window: good time resolution, poor frequency resolution
- Heisenberg Uncertainty Principle: there is no window function that localizes in time and frequency with arbitrary position.

Time-Frequency Representation
Signal and STFT with Hann window of length 20 ms

Time-Frequency Representation
Signal and STFT with Hann window of length 100 ms

Audio Features
Example: C-major scale (piano)

Audio Features
Example: Chromatic scale

Spectrogram
**Audio Features**

Example: Chromatic scale

Model assumption: Equal-tempered scale
- MIDI pitches: \( p \in [1 : 128] \)
- Piano notes: \( p = 21 \) (A0) to \( p = 108 \) (C8)
- Concert pitch: \( p = 69 \) (A4) \( \cong 440 \) Hz
- Center frequency: \( F_{\text{pitch}}(p) = 2^{(p-69)/12} \cdot 440 \) Hz

\( \rightarrow \) Logarithmic frequency distribution
Octave: doubling of frequency

---

**Audio Features**

Idea: Binning of Fourier coefficients

Divide up the frequency axis into logarithmically spaced “pitch regions” and combine spectral coefficients of each region to a single pitch coefficient.

---

**Audio Features**

Time-frequency representation

Windowing in the time domain

---

**Audio Features**

Example: Chromatic scale

Spectrogram
### Audio Features

**Example: Chromatic scale**

#### Log-frequency spectrogram

![Log-frequency spectrogram](image)

- **C6**: 4186 Hz
- **C5**: 2093 Hz
- **C4**: 1046 Hz
- **C3**: 523 Hz
- **C2**: 262 Hz

#### Frequency ranges for pitch-based log-frequency spectrogram

<table>
<thead>
<tr>
<th>Note</th>
<th>MIDI pitch</th>
<th>Center [Hz]</th>
<th>Left [Hz] boundary</th>
<th>Right [Hz] boundary</th>
<th>Width [Hz]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2</td>
<td>57</td>
<td>220.0</td>
<td>213.7</td>
<td>226.4</td>
<td>12.7</td>
</tr>
<tr>
<td>A#2</td>
<td>58</td>
<td>233.1</td>
<td>226.4</td>
<td>239.9</td>
<td>13.5</td>
</tr>
<tr>
<td>B2</td>
<td>59</td>
<td>246.9</td>
<td>239.9</td>
<td>254.2</td>
<td>14.3</td>
</tr>
<tr>
<td>C4</td>
<td>60</td>
<td>261.6</td>
<td>254.2</td>
<td>269.3</td>
<td>15.1</td>
</tr>
<tr>
<td>C#4</td>
<td>61</td>
<td>277.2</td>
<td>269.3</td>
<td>285.3</td>
<td>16.0</td>
</tr>
<tr>
<td>D4</td>
<td>62</td>
<td>293.7</td>
<td>286.3</td>
<td>302.3</td>
<td>17.0</td>
</tr>
<tr>
<td>D#4</td>
<td>63</td>
<td>311.1</td>
<td>302.3</td>
<td>319.9</td>
<td>18.0</td>
</tr>
<tr>
<td>E4</td>
<td>64</td>
<td>329.6</td>
<td>320.2</td>
<td>339.3</td>
<td>19.0</td>
</tr>
<tr>
<td>F4</td>
<td>65</td>
<td>349.2</td>
<td>339.3</td>
<td>359.5</td>
<td>20.2</td>
</tr>
<tr>
<td>F#4</td>
<td>66</td>
<td>370.0</td>
<td>360.8</td>
<td>380.8</td>
<td>21.4</td>
</tr>
<tr>
<td>G4</td>
<td>67</td>
<td>392.0</td>
<td>380.8</td>
<td>403.5</td>
<td>22.6</td>
</tr>
<tr>
<td>G#4</td>
<td>68</td>
<td>415.3</td>
<td>403.5</td>
<td>427.5</td>
<td>24.0</td>
</tr>
<tr>
<td>A4</td>
<td>69</td>
<td>440.0</td>
<td>427.5</td>
<td>452.9</td>
<td>25.4</td>
</tr>
</tbody>
</table>

### Audio Features

**Chroma features**

- Human perception of pitch is periodic in the sense that two pitches are perceived as similar in color if they differ by an octave.
- Separation of pitch into two components: tone height (octave number) and chroma.
- Chroma: 12 traditional pitch classes of the equal-tempered scale. For example: Chroma C = \{C, C\#, D, E, F, G, A, B, C\#, D\#, E\#, F\#, G\#, A\#\}
- Computation: pitch features → chroma features

Add up all pitches belonging to the same class

Result: 12-dimensional chroma vector.
Audio Features
Chroma features

- Sequence of chroma vectors correlates to the harmonic progression
- Normalization $x \rightarrow x/|x|$ makes features invariant to changes in dynamics
- Further denoising and smoothing
- Taking logarithm before adding up pitch coefficients accounts for logarithmic sensation of intensity

Audio Features
Logarithmic compression

For a positive constant $\gamma \in \mathbb{R}_{>0}$, the logarithmic compression

$$f_\gamma : \mathbb{R}_{>0} \rightarrow \mathbb{R}_{>0}$$

is defined by

$$f_\gamma(v) := \log(1 + \gamma \cdot v)$$

A value $v \in \mathbb{R}_{>0}$ is replaced by a compressed value $f_\gamma(v)$.

The higher $\gamma \in \mathbb{R}_{>0}$, the stronger the compression.
Audio Features

**Logarithmic compression**

A value \( v \in \mathbb{R}^{>0} \) is replaced by a compressed value \( \gamma v \). The higher \( \gamma \in \mathbb{R}^{>0} \) the stronger the compression.

Audio Features

**Normalization**

Replace a vector by the normalized vector \( x/\|x\| \) using a suitable norm \( \|\cdot\| \).

Example: Chroma vector \( x \in \mathbb{R}^{12} \)

Euclidean norm

\[ \|x\| := \left( \sum_{n=0}^{11} |x(n)|^2 \right)^{1/2} \]

Audio Features

**Chroma features** (normalized)

Karajan

Scherbakov
Audio Features

- There are many ways to implement chroma features
- Properties may differ significantly
- Appropriateness depends on respective application

Additional Material

Chroma Toolbox: Pitch, Chroma, CENS, CRP

- http://www.mpi-inf.mpg.de/resources/MIR/chromatoolbox/
- MATLAB implementations for various chroma variants

Inner Product

\[ (x|y) := \sum_{n=0}^{N} x(n)y(n) \] for \( x, y \in \mathbb{C}^N \)

- Length of a vector: \( ||x|| = \sqrt{(x|x)} \)
- Angle between two vectors: \( \cos(\phi) = \frac{|(x|y)|}{||x|| \cdot ||y||} \)
- Orthogonality of two vectors: \( (x|y) = 0 \)

Measuring the similarity of two functions

- \( \int_{t \in \mathbb{R}} f(t) \cdot g(t) \, dt \)
- Area positive and negative
- Integral small
- Similarity low

Discretization
Discretization

**Sampling**

\[ f : \mathbb{R} \rightarrow \mathbb{R} \]
\[ T > 0 \]
\[ x(n) := f(n \cdot T) \quad \text{equidistant sampling, } n \in \mathbb{Z} \]
\[ x: \mathbb{Z} \rightarrow \mathbb{R} \]
\[ x(n) \]
\[ F_s := \frac{1}{T} \quad \text{sampling rate} \]

CT-signal

Sampling period

DT-signal

Sample taken at time \( t = n \cdot T \)

Discretization

**Quantization**

**Sampling**

**Aliasing**

Original signal
Discretization

Aliasing

Original signal
Sampled signal using a sampling rate of 12 Hz
Reconstructed signal

Discretization

Aliasing

Original signal
Sampled signal using a sampling rate of 6 Hz
Reconstructed signal

Discretization

Aliasing

Original signal
Sampled signal using a sampling rate of 3 Hz
Reconstructed signal

Discretization

Integrals and Riemann sums

CT-signal \( f \)

\( \int_{\mathbb{R}} f(t) \, dt \)
Discretization
Integrals and Riemann sums

CT-signal $\int_{t \in \mathbb{R}} f(t) \, dt$

DT-signals (obtained by 1-sampling) $x$

Integral (total area)

Riemann sum (total area) $\approx$ Approximation of integral

Discretization
Integrals and Riemann sums

First CT-signal and DT-signal

Second CT-signal and DT-signal

Product of CT-signals and DT-signals

Integral $\approx$ Riemann sum $\int_{t \in \mathbb{R}} f(t)g(t)\,dt \approx \sum_{n \in \mathbb{Z}} x(n)y(n)$

Exponential Function
Polar coordinate representation of a complex number

$\exp(i\gamma) = \cos(\gamma) + isin(\gamma)$

$|\exp(i\gamma)| = 1$

$\exp(i\gamma) = \exp(i(\gamma + 2\pi))$
Exponential Function
Complex conjugate number
\( \exp(i\gamma) = \exp(-i\gamma) \)

Exponential Function
Additivity property
\( \exp(i(\gamma_1 + \gamma_2)) = \exp(i\gamma_1) \exp(i\gamma_2) \)

Fourier Transform
Chirp signal with \( \lambda = 0.003 \)
\[ f(t) := \begin{cases} 
\sin(\lambda \cdot t^2), & \text{for } t > 0 \\
0, & \text{for } t < 0 
\end{cases} \]

Fourier Transform
Chirp signal with \( \lambda = 0.004 \)
\[ f(t) := \begin{cases} 
\sin(\lambda \cdot t^2), & \text{for } t > 0 \\
0, & \text{for } t < 0 
\end{cases} \]

Fourier Transform
DFT approximation of Fourier transform

Fourier Transform
DFT approximation of Fourier transform
Fourier Transform

Discrete STFT

\[ \mathcal{X}(m, k) := \sum_{n=0}^{N-1} x(n + mH)w(n) \exp(-2\pi j kn/N) \]

- DT-signal: \( x : \mathbb{Z} \rightarrow \mathbb{R} \)
- Window function of length \( N \in \mathbb{N} \)
- Hop size \( H \in \mathbb{N} \)
- \( K = N/2 \)
- Fourier coefficient for frequency index \( k \in [0; K] \) and time frame \( m \in \mathbb{Z} \)

Parameters

- \( N = 64 \)
- \( H = 8 \)
- \( F_s = 32 \text{ Hz} \)

Computational world

Physical world

Index (frames)

Time (seconds)

Frequency (Hz)

Frequency (Hz)

Log-Frequency Spectrogram

Pooling procedure for discrete STFT

Parameters

- \( N = 4096 \)
- \( H = 2048 \)
- \( F_s = 44100 \text{ Hz} \)

Fast Fourier Transform

Signal Spaces and Fourier Transforms

<table>
<thead>
<tr>
<th>Signal space</th>
<th>( L^2(\mathbb{R}) )</th>
<th>( L^2([0,1]) )</th>
<th>( L^2([1]) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inner product</td>
<td>( \langle f,g \rangle = \int_{-\infty}^{\infty} f(x)\overline{g(x)} , dx )</td>
<td>( \langle f,g \rangle = \int_{0}^{1} f(x)\overline{g(x)} , dx )</td>
<td>( \langle f,g \rangle = \int_{1}^{1} f(x)\overline{g(x)} , dx )</td>
</tr>
<tr>
<td>Norm</td>
<td>( | f |_2 = \sqrt{\langle f,f \rangle} )</td>
<td>( | f |_2 = \sqrt{\langle f,f \rangle} )</td>
<td>( | f |_2 = \sqrt{\langle f,f \rangle} )</td>
</tr>
<tr>
<td>Definition</td>
<td>( L^2(\mathbb{R}) = { f : \mathbb{R} \rightarrow \mathbb{C} \mid | f |_2 &lt; \infty } )</td>
<td>( L^2([0,1]) = { f : [0,1] \rightarrow \mathbb{C} \mid | f |_2 &lt; \infty } )</td>
<td>( L^2([1]) = { f : [1] \rightarrow \mathbb{C} \mid | f |_2 &lt; \infty } )</td>
</tr>
<tr>
<td>Elementary frequency function</td>
<td>( \delta_{\xi}(t) = \overline{\exp(2\pi j \xi t)} )</td>
<td>( \delta_{\xi_1}(t) = \overline{\exp(2\pi j \xi_1 \cdot t)} )</td>
<td>( \delta_{\xi_2}(t) = \overline{\exp(2\pi j \xi_2 \cdot t)} )</td>
</tr>
<tr>
<td>Frequency parameter</td>
<td>( \xi \in \mathbb{R} )</td>
<td>( \xi \in [0,1] )</td>
<td>( \xi \in (1,1) )</td>
</tr>
<tr>
<td>Fourier representation</td>
<td>( f(t) = \int_{-\infty}^{\infty} f(t) \delta_{\xi}(t) , dt )</td>
<td>( f(t) = \int_{0}^{1} f(t) \delta_{\xi_1}(t) , dt )</td>
<td>( f(t) = \int_{1}^{1} f(t) \delta_{\xi_2}(t) , dt )</td>
</tr>
<tr>
<td>Fourier transform</td>
<td>( \hat{f} )</td>
<td>( \hat{f} )</td>
<td>( \hat{f} )</td>
</tr>
</tbody>
</table>

Algorithm: FFT

Input:
- The length \( N = 2^p \) with \( N \) being a power of two.
- The vector \( (x(0), \ldots, x(N-1)) \) in \( \mathbb{C}^N \).

Output:
- The vector \( (X(0), \ldots, X(N-1)) = \text{FFT}(x(0), \ldots, x(N-1)) \).

Procedure:
- Let \( (X(0), \ldots, X(N-1)) = \text{FFT}(x(0), \ldots, x(N-1)) \) denote the general form of the FFT algorithm.
- If \( N = 1 \) then
  - \( X(0) = x(0) \).
- Otherwise compute recursively:
  - \( X(k) = X(k) + X(N/2 - k) \) for \( k \in [0; N/2 - 1] \).
  - \( X(k) = X(k) - X(N/2 - k) \) for \( k \in [0; N/2 - 1] \).

\[ X(k) = A(k) + C(k) \] for \( k \in [0; N/2 - 1] \).

\[ X(N/2 + k) = A(k) - C(k) \] for \( k \in [0; N/2 - 1] \).