Introduction to **Audio Content Analysis**

module B.6: frequency resolution & instantaneous frequency

alexander lerch

Georgia Tech Center for Music Technology
College of Design

introduction

overview

corresponding textbook section

- section 7.3.1
- appendix B.6

- **lecture content**
  - frequency detection error for sampled signals
  - instantaneous frequency/frequency reassignment

- **learning objectives**
  - list the factors influencing frequency resolution in time and frequency domains
  - explain the frequency error in Cent
  - implement an instantaneous frequency estimate

introduction
overview

corresponding textbook section

- section 7.3.1
- appendix B.6

- **lecture content**
  - frequency detection error for sampled signals
  - instantaneous frequency/frequency reassignment

- **learning objectives**
  - list the factors influencing frequency resolution in time and frequency domains
  - explain the frequency error in Cent
  - implement an instantaneous frequency estimate

pitch detection resolution

introduction

- (fundamental) frequency detection on digital signals (discrete in time and frequency)

$\Rightarrow$ quantized result

error being made due to *discrete* signal processing

- **time domain**:
  - detection of *period length*

  $\Rightarrow$ maximum error depends on distance between two samples (sample rate)

- **frequency domain**:
  - detection of *bin frequency*

  $\Rightarrow$ maximum error depends on distance between two frequency bins (block length and sample rate)

pitch detection resolution
introduction

- (fundamental) frequency detection on digital signals (discrete in time and frequency)

⇒ quantized result

error being made due to *discrete* signal processing

- **time domain**:
  - detection of *period length*

  ⇒ maximum error depends on distance between two samples (sample rate)

- **frequency domain**:
  - detection of *bin frequency*

  ⇒ maximum error depends on distance between two frequency bins (block length and sample rate)

pitch detection resolution
introduction

- (fundamental) frequency detection on digital signals (discrete in time and frequency)

$\Rightarrow$ quantized result

error being made due to *discrete* signal processing

- **time domain**:
  - detection of *period length*

  $\Rightarrow$ maximum error depends on distance between two samples (sample rate)

- **frequency domain**:
  - detection of *bin frequency*

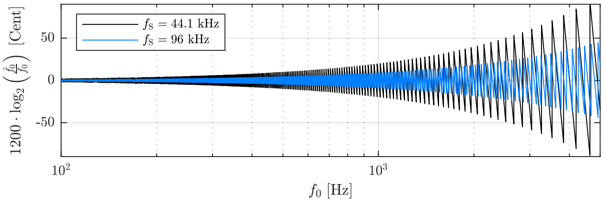
  $\Rightarrow$ maximum error depends on distance between two frequency bins (block length and sample rate)

pitch detection resolution

time domain (e.g., ACF)

period length quantized to multiple of inter-sample interval T_S

$$
\begin{aligned}
T_\mathrm{Q} &= j \cdot T_\mathrm{S} \\
\Rightarrow \quad f_\mathrm{Q} &= \frac{1}{j \cdot T_\mathrm{S}}
\end{aligned}
$$



matlab source: plotPitchErrorTimeDomain.m

pitch detection resolution
frequency domain (e.g., HPS)

frequency quantized to multiple of inter-bin interval

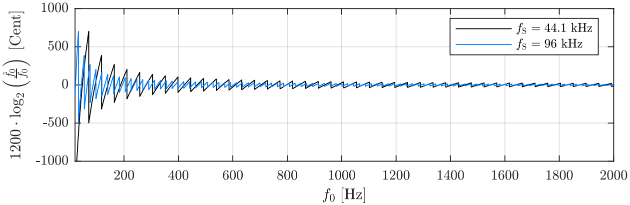
$$f_{\mathrm{Q}} = k \cdot \frac{f_{\mathrm{S}}}{\mathcal{K}}$$

$\mathcal{K}$	Δf [Hz]	k_{ST}	$f(k_{\mathrm{ST}})$ [Hz]
256	187.5	35	6562.5
512	93.75	35	3281.25
1024	46.875	35	1640.625
2048	23.4375	35	820.3125
4096	11.7188	35	410.1563
8192	5.8594	35	205.0781
16384	2.9297	35	102.5391

pitch detection resolution
frequency domain (e.g., HPS)

frequency quantized to multiple of inter-bin interval

$$f_\mathrm{Q} = k \cdot \frac{f_\mathrm{S}}{\mathcal{K}}$$



matlab source: plotPitchErrorFreqDomain.m

pitch detection resolution

simple fix

- **assumption**: pitch is stationary with minor deviations over time
- **simple solution**:
  - average pitch observations over blocks
  - the more blocks are averaged, the more result might approximate the *real* (population) mean
- **problems**:
  1. adds significant latency (non-realtime)
  2. will not work for time-variant signals (speech, music)

pitch detection resolution

simple fix

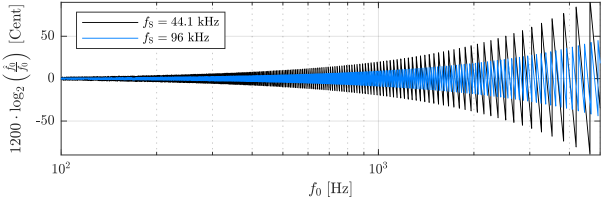
- **assumption**: pitch is stationary with minor deviations over time
- **simple solution**:
  - average pitch observations over blocks
  - the more blocks are averaged, the more result might approximate the *real* (population) mean
- **problems**:
  1. adds significant latency (non-realtime)
  2. will not work for time-variant signals (speech, music)

pitch detection resolution
simple fix

- **assumption**: pitch is stationary with minor deviations over time
- **simple solution**:
  - average pitch observations over blocks
  - the more blocks are averaged, the more result might approximate the *real* (population) mean
- **problems**:
  1. adds significant latency (non-realtime)
  2. will not work for time-variant signals (speech, music)

pitch detection resolution

time domain observations

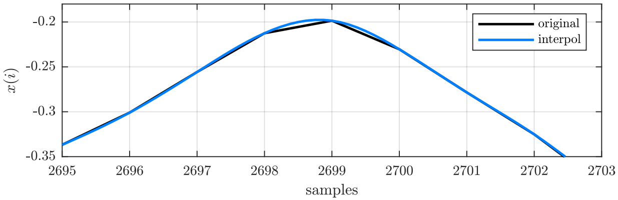


- error depends on fundamental frequency
- error depends on sample rate

matlab source: plotPitchErrorTimeDomain.m

pitch detection resolution
time domain workarounds

virtually increase time resolution by upsampling



+ higher virtual resolution
− significant workload increase

matlab source: plotTimeInterp.m

pitch detection resolution

frequency domain workarounds

different ways of increasing frequency resolution in the frequency domain

1. increasing the FFT window length (decreases time resolution)
2. interpolating the spectrum
3. applying frequency reassignment

pitch detection resolution

frequency domain workarounds

different ways of increasing frequency resolution in the frequency domain

1. increasing the FFT window length (decreases time resolution)
2. interpolating the spectrum
3. applying frequency reassignment

pitch detection resolution

frequency domain workarounds

different ways of increasing frequency resolution in the frequency domain

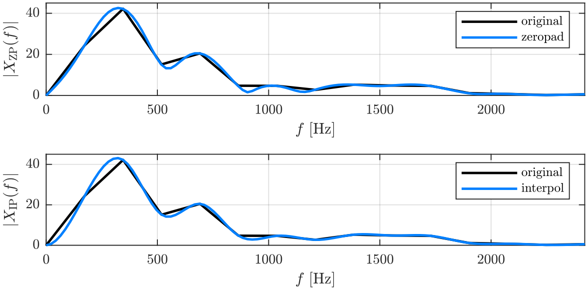
1. increasing the FFT window length (decreases time resolution)
2. interpolating the spectrum
3. applying frequency reassignment

pitch detection resolution

spectrum interpolation

1. zeropad in time domain

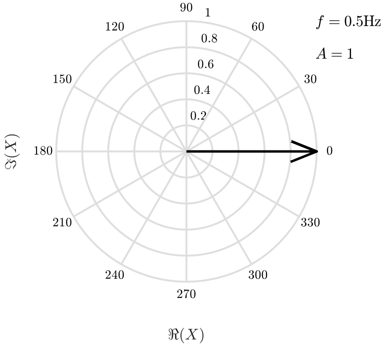
2. use standard interpolation on magnitude spectrum



matlab source: plotFreqInterp.m

pitch detection resolution

frequency reassignment: relation of phase and frequency 1/2

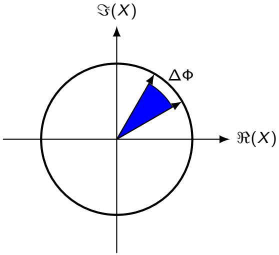


- phasor representation:
  1. sine value is defined by magnitude and phase
  2. decreasing the amplitude ⇒ shorter vector
  3. increasing the frequency ⇒ increasing speed

matlab source: matlab/animatePhasor.m

pitch detection resolution

frequency reassignment: relation of phase and frequency 2/2



- relation of frequency and phase change:
  - time for full rotation is period length T with

$$f = \frac{1}{T}$$

  - time for fractional rotation $\Delta\Phi$ is corresponding fraction of period length

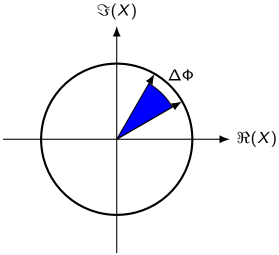
$$f = \frac{\Delta\Phi}{\Delta t}$$

  - in other words:

$$\begin{aligned} \Phi(t) &= \omega \cdot t \\ \Rightarrow \frac{d\Phi(t)}{dt} &= \omega = 2\pi f \end{aligned}$$

pitch detection resolution

frequency reassignment: relation of phase and frequency 2/2



- relation of frequency and phase change:
  - time for full rotation is period length T with

$$f = \frac{1}{T}$$

  - time for fractional rotation $\Delta\Phi$ is corresponding fraction of period length

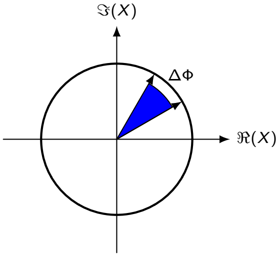
$$f = \frac{\Delta\Phi}{\Delta t}$$

  - in other words:

$$\begin{aligned} \Phi(t) &= \omega \cdot t \\ \Rightarrow \frac{d\Phi(t)}{dt} &= \omega = 2\pi f \end{aligned}$$

pitch detection resolution

frequency reassignment: relation of phase and frequency 2/2



- relation of frequency and phase change:
  - time for full rotation is period length T with

$$f = \frac{1}{T}$$

  - time for fractional rotation $\Delta\Phi$ is corresponding fraction of period length

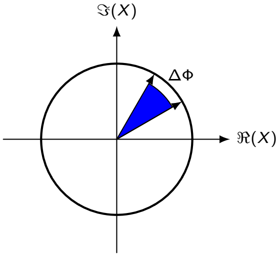
$$f = \frac{\Delta\Phi}{\Delta t}$$

  - in other words:

$$\begin{aligned} \Phi(t) &= \omega \cdot t \\ \Rightarrow \frac{d\Phi(t)}{dt} &= \omega = 2\pi f \end{aligned}$$

pitch detection resolution

frequency reassignment: relation of phase and frequency 2/2



- relation of frequency and phase change:
  - time for full rotation is period length T with

$$f = \frac{1}{T}$$

  - time for fractional rotation $\Delta\Phi$ is corresponding fraction of period length

$$f = \frac{\Delta\Phi}{\Delta t}$$

  - in other words:

$$\begin{aligned} \Phi(t) &= \omega \cdot t \\ \Rightarrow \frac{d\Phi(t)}{dt} &= \omega = 2\pi f \end{aligned}$$

pitch detection resolution

frequency reassignment: principles

frequency domain:

- instead of using the bin frequency

$$f(k) = k \cdot \frac{f_\mathrm{S}}{\mathcal{K}}$$

- we use the phase of each bin $\Phi(k, n)$
- to compute the frequency from the phase difference of neighboring blocks

$$\omega_\mathrm{I}(k, n) \propto \Phi(k, n) - \Phi(k, n-1)$$

- $\omega_\mathrm{I}(k, n)$ is called **instantaneous frequency** per block per bin

pitch detection resolution

frequency reassignment: principles

frequency domain:

- instead of using the bin frequency

$$f(k) = k \cdot \frac{f_\mathrm{S}}{\mathcal{K}}$$

- we use the phase of each bin $\Phi(k, n)$
- to compute the frequency from the phase difference of neighboring blocks

$$\omega_\mathrm{I}(k, n) \propto \Phi(k, n) - \Phi(k, n-1)$$

- $\omega_\mathrm{I}(k, n)$ is called **instantaneous frequency** per block per bin

pitch detection resolution

frequency reassignment: scaling factor

- instantaneous frequency calculation has to take into account
  - hop size $\mathcal{H}$
  - sample rate f_S

$$\omega_\mathrm{I}(k, n) = \frac{\Delta\Phi_\mathrm{u}(k, n)}{\mathcal{H}} \cdot f_\mathrm{S}$$

- problem: phase ambiguity

$$\Phi(k, n) = \Phi(k, n) + j \cdot 2\pi$$

$\Rightarrow$ *phase unwrapping*

pitch detection resolution

frequency reassignment: scaling factor

- instantaneous frequency calculation has to take into account
  - hop size $\mathcal{H}$
  - sample rate f_S

$$\omega_\mathrm{I}(k,n) = \frac{\Delta\Phi_\mathrm{u}(k,n)}{\mathcal{H}} \cdot f_\mathrm{S}$$

- problem: phase ambiguity

$$\Phi(k,n) = \Phi(k,n) + j \cdot 2\pi$$

$\Rightarrow$ *phase unwrapping*

pitch detection resolution

frequency reassignment: phase unwrapping

1. compute unwrapped phase $\Phi_\mathrm{u}(k,n)$
   - estimate unwrapped bin phase

$$\hat{\Phi}(k,n) = \Phi(k,n-1) + \underbrace{2\pi k \cdot \frac{\mathcal{H}}{\mathcal{K}}}_{=\omega_k \cdot \frac{\mathcal{H}}{f_\mathrm{S}}}$$

   - unwrap phase by shifting current phase to estimate's range

$$\Phi_\mathrm{u}(k,n) = \hat{\Phi}(k,n) + \mathrm{princarg}\left[\Phi(k,n) - \hat{\Phi}(k,n)\right]$$

2. compute unwrapped phase difference

$$\begin{aligned}\Delta\Phi_\mathrm{u}(k,n) &= \Phi_\mathrm{u}(k,n) - \Phi(k,n-1)\\ &= \hat{\Phi}(k,n) + \mathrm{princarg}\left[\Phi(k,n) - \hat{\Phi}(k,n)\right] - \Phi(k,n-1)\\ &= \frac{2\pi k}{\mathcal{K}}\mathcal{H} + \mathrm{princarg}\left[\Phi(k,n) - \Phi(k,n-1) - \frac{2\pi k}{\mathcal{K}}\mathcal{H}\right]\end{aligned}$$

pitch detection resolution

frequency reassignment: phase unwrapping

1. compute unwrapped phase $\Phi_\mathrm{u}(k,n)$
   - estimate unwrapped bin phase

$$\hat{\Phi}(k,n) = \Phi(k,n-1) + \underbrace{2\pi k \cdot \frac{\mathcal{H}}{\mathcal{K}}}_{=\omega_k \cdot \frac{\mathcal{H}}{f_\mathrm{S}}}$$

   - unwrap phase by shifting current phase to estimate's range

$$\Phi_\mathrm{u}(k,n) = \hat{\Phi}(k,n) + \mathrm{princarg}\left[\Phi(k,n) - \hat{\Phi}(k,n)\right]$$

2. compute unwrapped phase difference

$$\begin{aligned}\Delta\Phi_\mathrm{u}(k,n) &= \Phi_\mathrm{u}(k,n) - \Phi(k,n-1)\\ &= \hat{\Phi}(k,n) + \mathrm{princarg}\left[\Phi(k,n) - \hat{\Phi}(k,n)\right] - \Phi(k,n-1)\\ &= \frac{2\pi k}{\mathcal{K}}\mathcal{H} + \mathrm{princarg}\left[\Phi(k,n) - \Phi(k,n-1) - \frac{2\pi k}{\mathcal{K}}\mathcal{H}\right]\end{aligned}$$

pitch detection resolution
frequency reassignment: phase unwrapping

1. compute unwrapped phase $\Phi_\mathrm{u}(k,n)$
   - estimate unwrapped bin phase

$$\hat{\Phi}(k,n) = \Phi(k,n-1) + \underbrace{2\pi k \cdot \frac{\mathcal{H}}{\mathcal{K}}}_{=\omega_k \cdot \frac{\mathcal{H}}{f_\mathrm{S}}}$$

   - unwrap phase by shifting current phase to estimate's range

$$\Phi_\mathrm{u}(k,n) = \hat{\Phi}(k,n) + \mathrm{princarg}\left[\Phi(k,n) - \hat{\Phi}(k,n)\right]$$

2. compute unwrapped phase difference

$$\begin{aligned}
\Delta\Phi_\mathrm{u}(k,n) &= \Phi_\mathrm{u}(k,n) - \Phi(k,n-1) \\
&= \hat{\Phi}(k,n) + \mathrm{princarg}\left[\Phi(k,n) - \hat{\Phi}(k,n)\right] - \Phi(k,n-1) \\
&= \frac{2\pi k}{\mathcal{K}}\mathcal{H} + \mathrm{princarg}\left[\Phi(k,n) - \Phi(k,n-1) - \frac{2\pi k}{\mathcal{K}}\mathcal{H}\right]
\end{aligned}$$

pitch detection resolution

frequency reassignment: problems

- **overlapping spectral components**
  - sinusoidal components often overlap (spectral leakage, several instruments playing the same pitch, ...)
    - $\Rightarrow$ "incorrect" phase estimate
    - spectrum should be as sparse as possible, increase STFT length
- **inaccurate phase unwrapping**
  - unwrapping algorithm is based on assumption of similarity between predicted and measured phase
  - decrease hop size

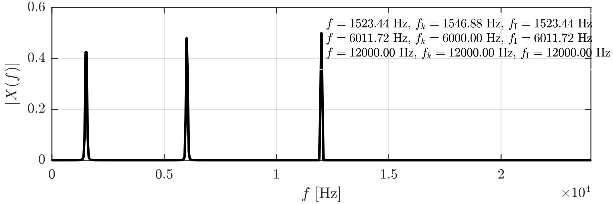
pitch detection resolution

frequency reassignment: problems

- **overlapping spectral components**
    - sinusoidal components often overlap (spectral leakage, several instruments playing the same pitch, ...)
        - ⇒ "incorrect" phase estimate
        - ▶ spectrum should be as sparse as possible, increase STFT length
- **inaccurate phase unwrapping**
    - unwrapping algorithm is based on assumption of similarity between predicted and measured phase
    - decrease hop size

pitch detection resolution

frequency reassignment: example



- FFT length: 1024
- sample rate: 48 kHz

- selected frequencies:
  - between bins (0.5)
  - between bins (0.25)
  - on bin

matlab source: plotInstantaneousFreq.m

pitch detection resolution

frequency reassignment: applications

- **improving frequency resolution**
  - e.g., for detecting signal frequencies when using a filter bank
- **improving phase extrapolation**
  - e.g., for accurate phase estimation in the *phase vocoder*
- **grouping spectral bins**
  - spectral leakage sidelobes have the same instantaneous frequency
- **tonalness detection**
  - the instantaneous frequency should be reasonably close to the bin frequency for the component to be considered tonal

pitch detection resolution

frequency reassignment: applications

- **improving frequency resolution**
  - e.g., for detecting signal frequencies when using a filter bank
- **improving phase extrapolation**
  - e.g., for accurate phase estimation in the *phase vocoder*
- **grouping spectral bins**
  - spectral leakage sidelobes have the same instantaneous frequency
- **tonalness detection**
  - the instantaneous frequency should be reasonably close to the bin frequency for the component to be considered tonal

pitch detection resolution
frequency reassignment: applications

- **improving frequency resolution**
  - e.g., for detecting signal frequencies when using a filter bank
- **improving phase extrapolation**
  - e.g., for accurate phase estimation in the *phase vocoder*
- **grouping spectral bins**
  - spectral leakage sidelobes have the same instantaneous frequency
- **tonalness detection**
  - the instantaneous frequency should be reasonably close to the bin frequency for the component to be considered tonal

pitch detection resolution

frequency reassignment: applications

- **improving frequency resolution**
  - e.g., for detecting signal frequencies when using a filter bank
- **improving phase extrapolation**
  - e.g., for accurate phase estimation in the *phase vocoder*
- **grouping spectral bins**
  - spectral leakage sidelobes have the same instantaneous frequency
- **tonalness detection**
  - the instantaneous frequency should be reasonably close to the bin frequency for the component to be considered tonal

summary
lecture content

- **frequency resolution of sampled signals depends on**
  - time domain: sample rate
  - freq domain: sample rate, block size
- **pitch detection error in Cent also depends on input frequency**
  - time domain: high error at high frequencies
  - freq domain: high error at low frequencies
- **possible solutions**
  - time domain:
    - upsampling/interpolation
  - freq domain:
    - zeropadding/interpolation
    - frequency reassignment (instantaneous frequency)