Real-time monophonic pitch estimation for guitar driven sound synthesis

Luc de Jonckheere

Supervised by:
Erwin Bakker
Michael Lew

LIACS
Leiden University

29 Augustus 2022

Universiteit Leiden
The Netherlands

Introduction

Real-time monophonic pitch estimation

- Pitch estimation: Measure pitch of signal/played note
- Real-time: While the musician is playing it
  - ↳ Real-time constraint: 20 ms
- Monophonic: One note at the time

Motivations

- Drive synthesizer with a guitar
- No open source real-time pitch estimation applications available

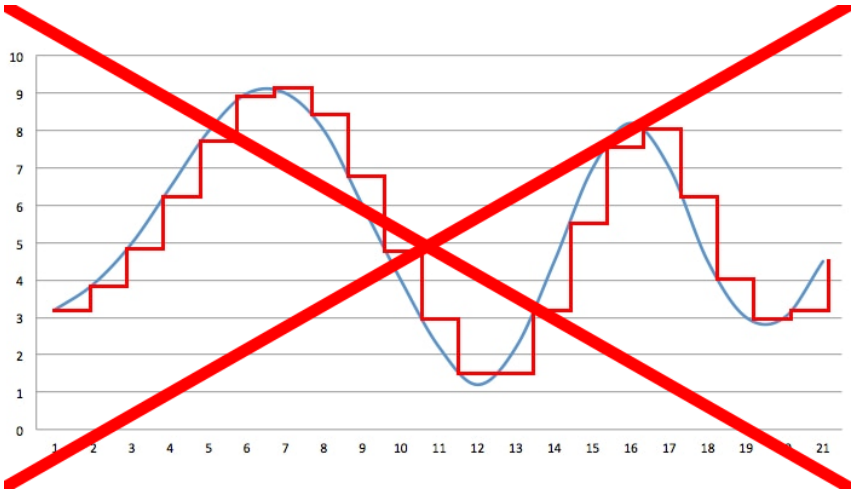
Goals

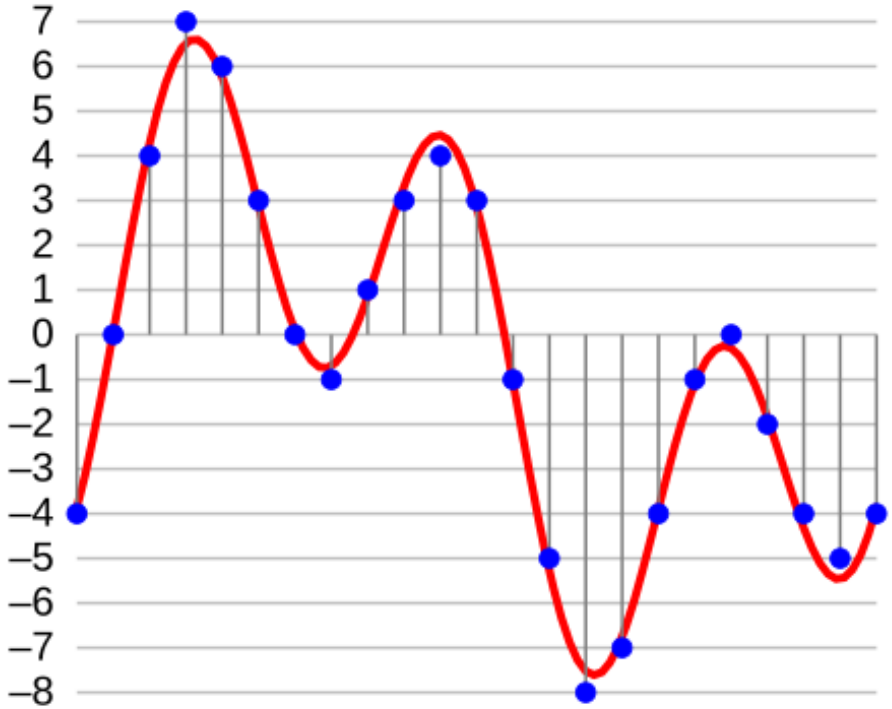
1. Create a real-time pitch estimation algorithm
2. Synthesize audio based on estimation

Universiteit Leiden
The Netherlands

Audio in computers

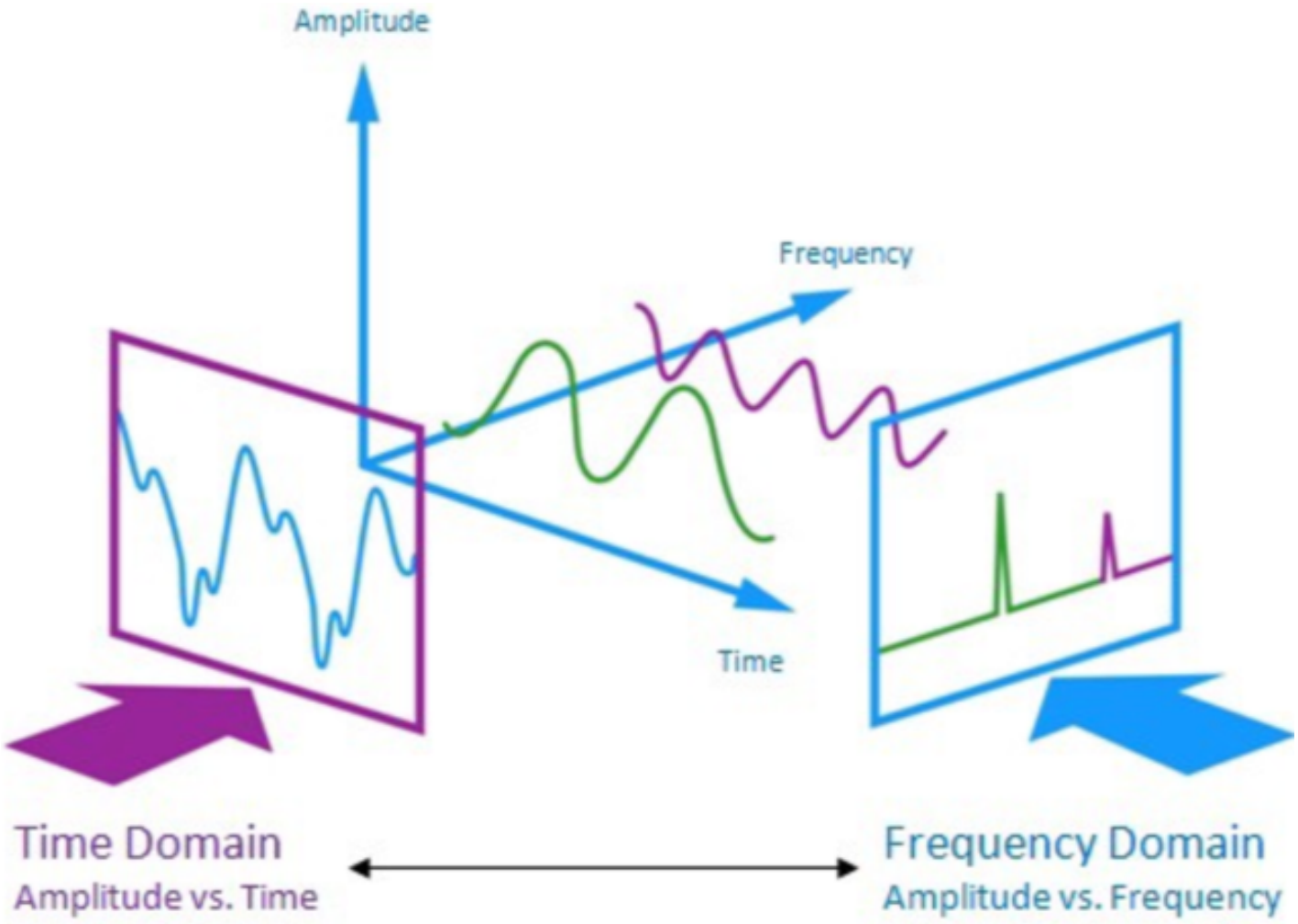
- Audio format
- Sample rate (f_{SR})





Universiteit Leiden
The Netherlands

Fourier transform



Preliminary knowledge

Fourier transform

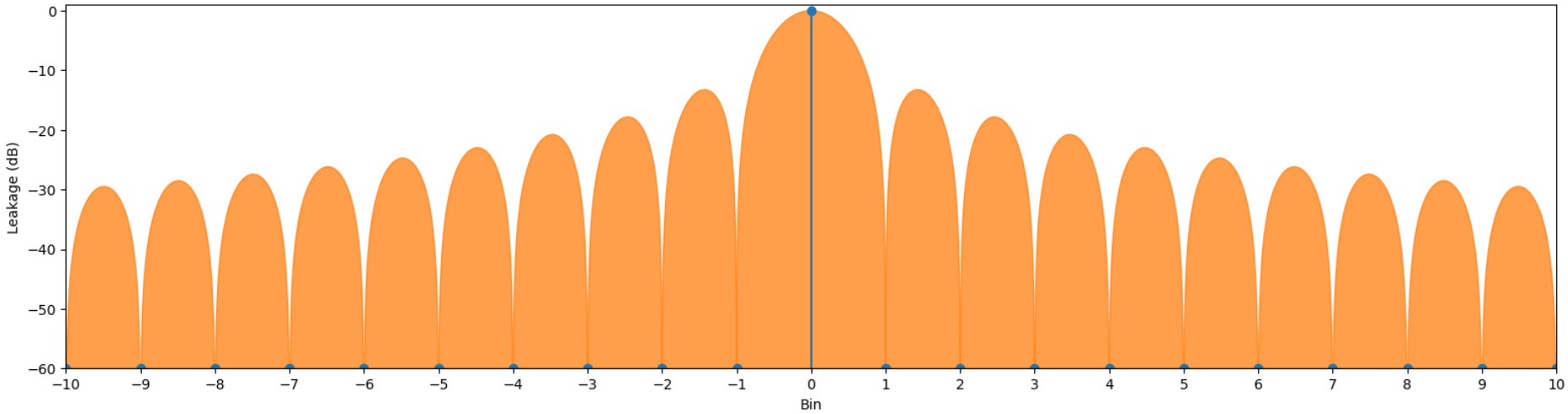
- Function of time $\rightarrow$ function of frequency
- Continuous/infinite $\notin$ computers $\rightarrow$ DFT

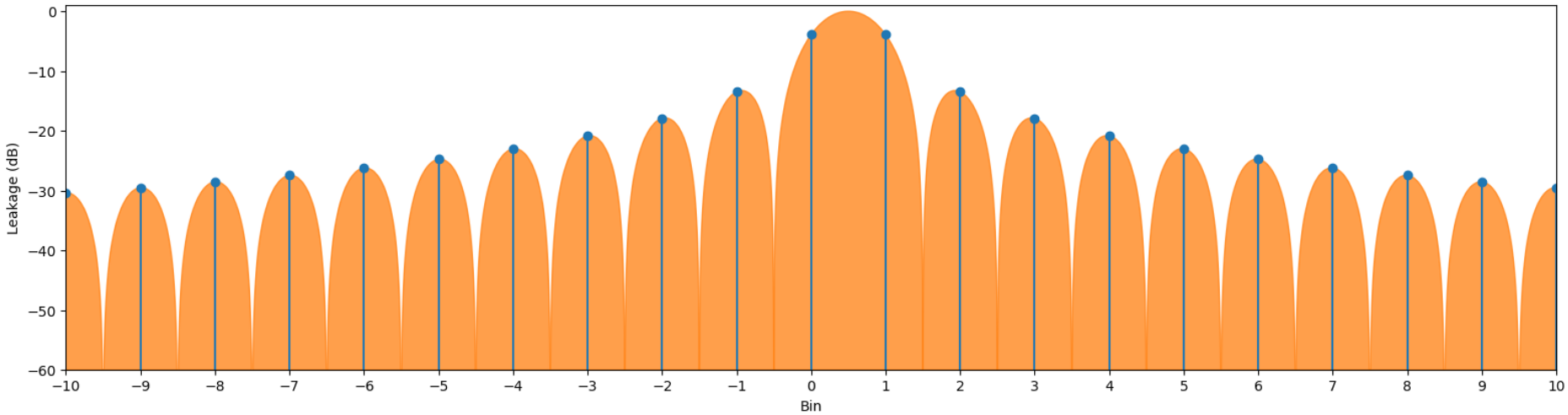
Discrete Fourier transform (DFT)

- Frame of samples $\rightarrow$ frequency bins
- Fourier resolution = sample rate / frame size = frame time $^{-1}$
- Nyquist frequency = sample rate / 2
- Each bin corresponds to one frequency
- Spectral leakage

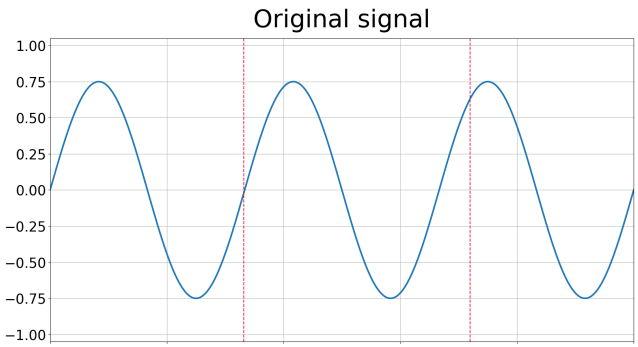
Universiteit Leiden
The Netherlands

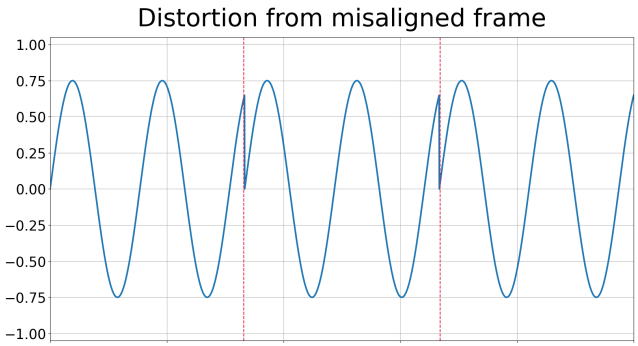
Preliminary knowledge

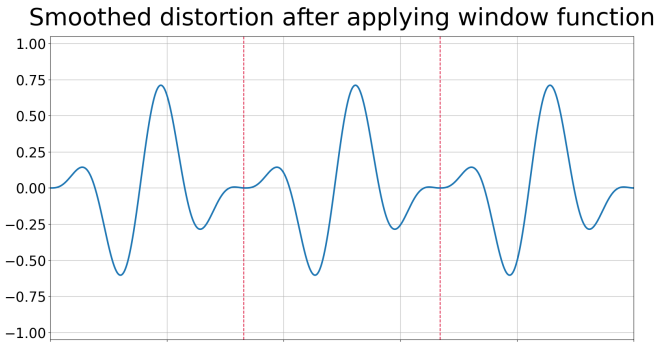


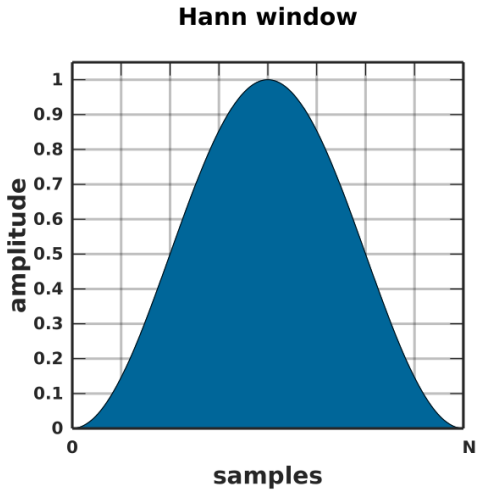


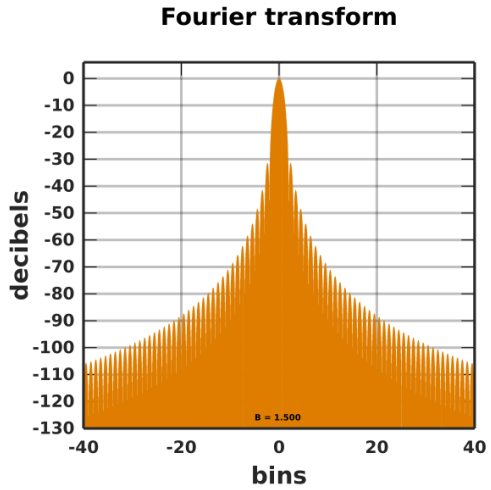
Preliminary knowledge

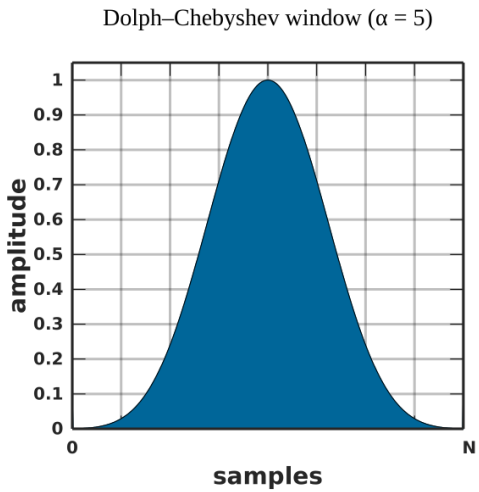


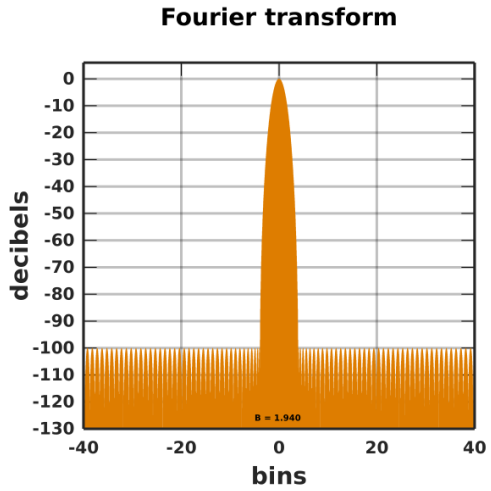












Preliminary knowledge

Music theory

- 12 tone equal temperament (12-TET)
- Tuning note ($A_4 = 440$ Hz)
- MIDI note numbers ($A_4 = 69$)

Physics of sound

- Fundamental and overtones
- Timbre
- Transients

Universiteit Leiden
The Netherlands

Basic estimator

Basic estimator algorithm

Given frame of samples F with sample rate f_{SR}

1. Apply window function
2. Fourier transform
3. Calculate amplitudes of bins
4. Find bin index with highest amplitude (i_{max})
5. Compute frequency of the bin ($f_b = i_{\text{max}} * \frac{f_{\text{SR}}}{|F|}$)
6. Compute MIDI note number corresponding to frequency

↳ $\underbrace{\left\lfloor 12 * {}^{2}\log \frac{f_b}{440} \right\rceil}_{\text{Distance from A}_4} + \underbrace{69}_{\text{A}_4\text{ MIDI number}}$

Universiteit Leiden
The Netherlands

Basic estimator

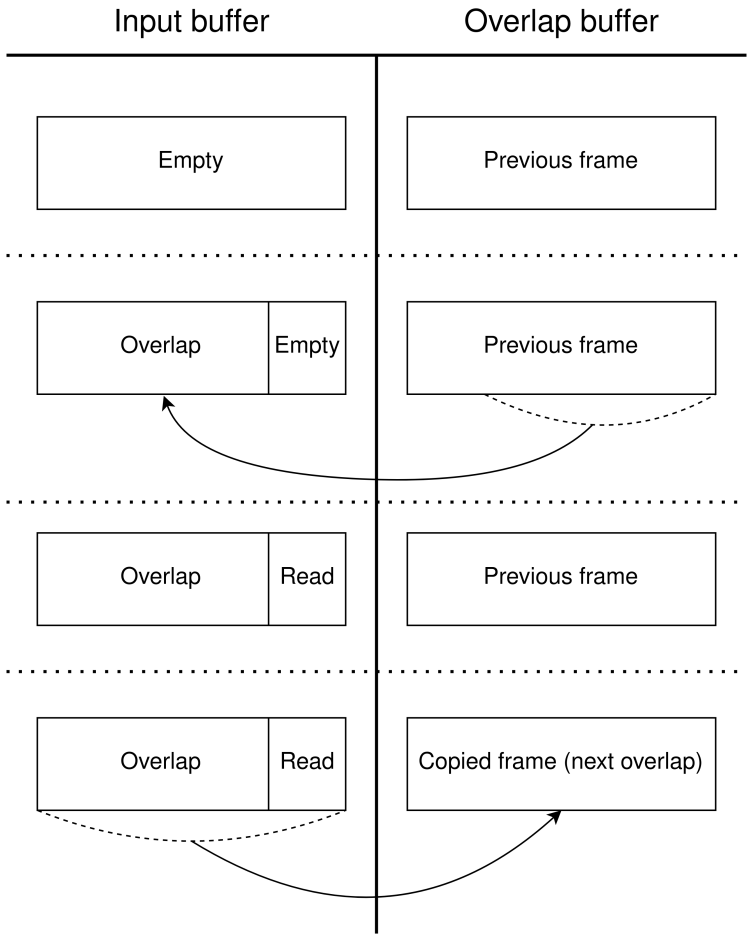
Shortcomings

- Latency
  - ↳ F_2 - $E_2 \approx 5$ Hz $\rightarrow$ 200 ms frame time
- Frame rate
  - ↳ Frame time $^{-1} \rightarrow 5$ FPS
- Estimation quantization
- Octave problem
- Never silent

Universiteit Leiden
The Netherlands

Low frame rate

- Decrease frame time $\rightarrow$ decrease Fourier resolution
- Overlap frames

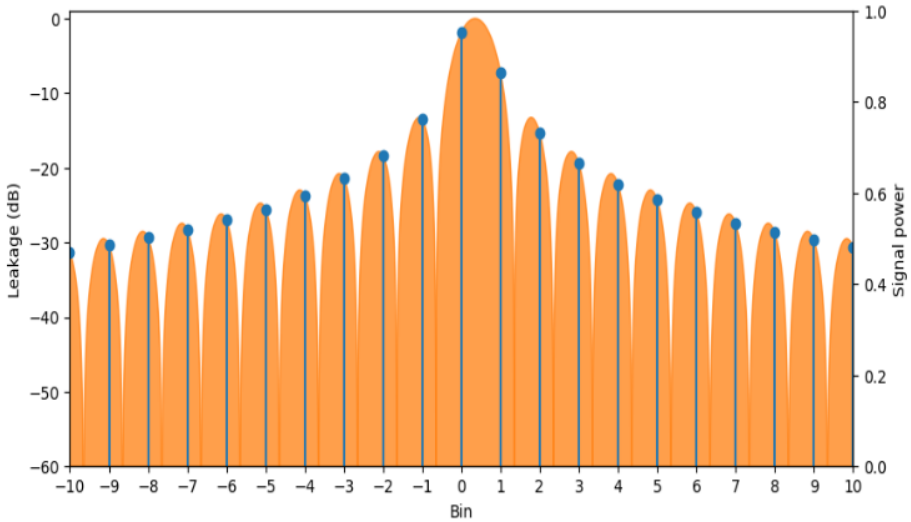


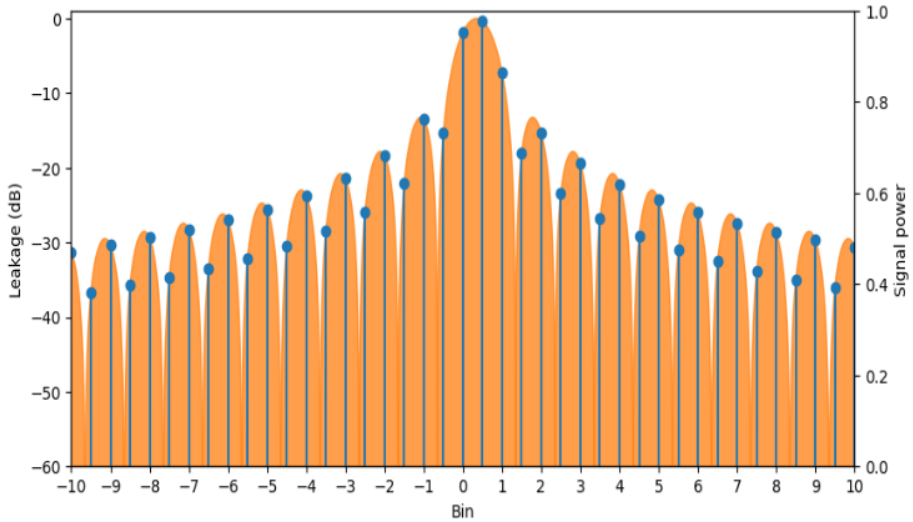
HighRes estimator

Low resolution $\rightarrow$ interpolation

Zero-padding

- Increase frame size $\rightarrow$ increase number of bins
- No resolution increase, only interpolation!

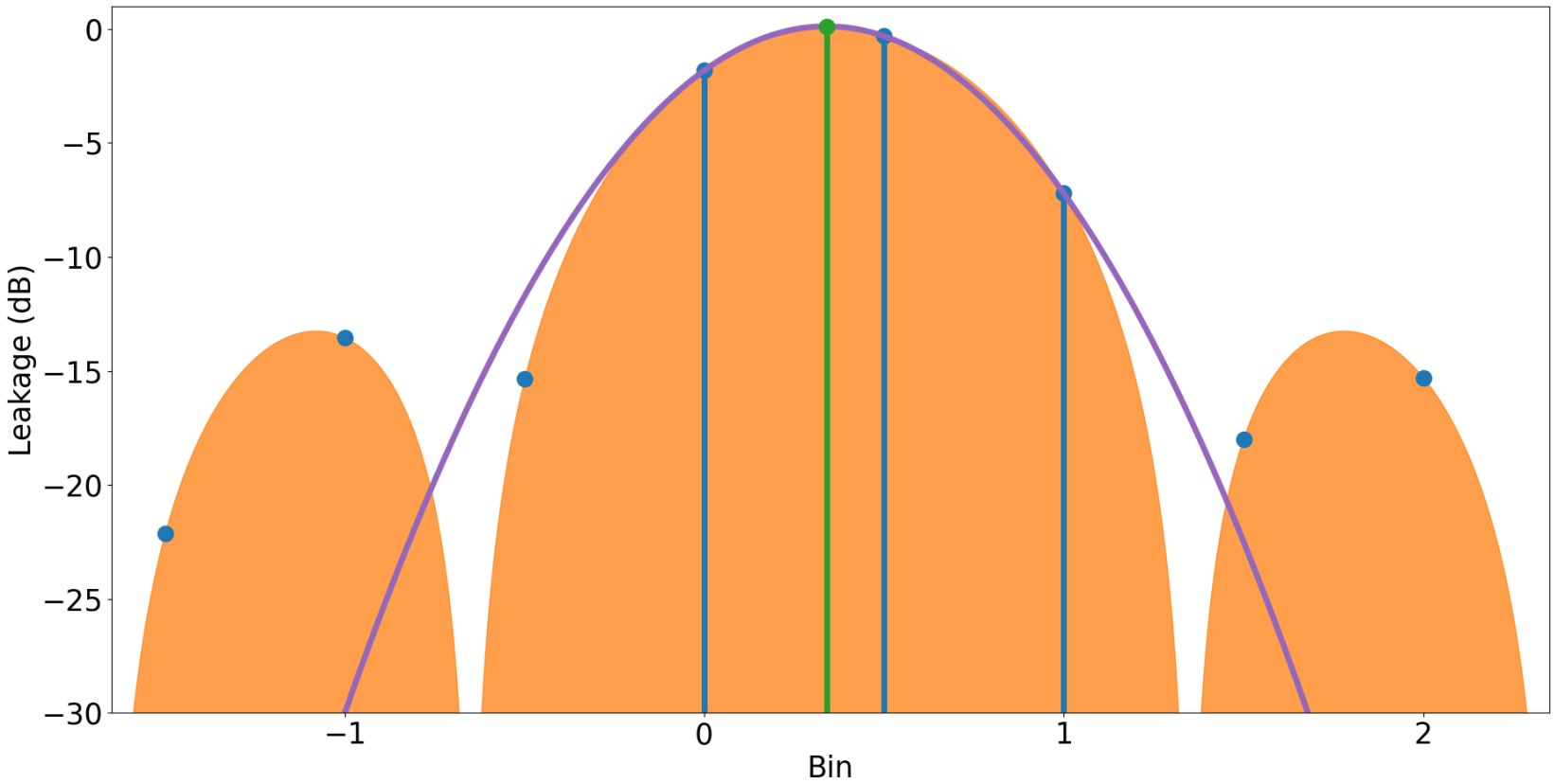




HighRes estimator

Quadratic interpolation (QIFFT)

- Fit Lagrange parabola through peak and neighbors
- Parabola peak is interpolated location
- Scaled magnitude spectrum



Peak picking and note selection

Basic estimator shortcomings:

- No silence
- Octave problem

↳ Solved by:

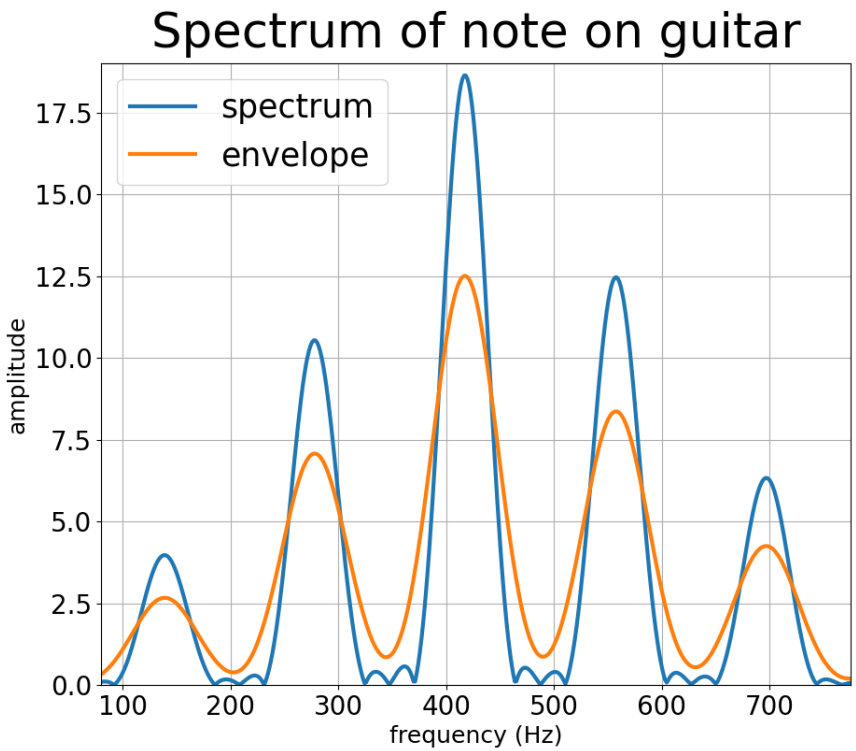
- Pick significant peaks
- Determine played note from picked peaks

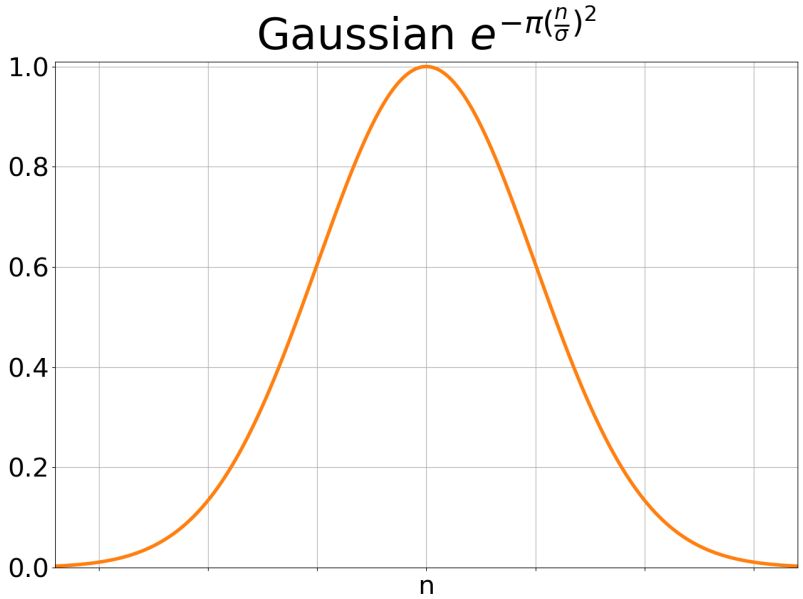
Universiteit Leiden
The Netherlands

HighRes estimator

Gaussian peak picking

- Gaussian envelope
- Moving average of spectrum
- Eliminates spectral leakage noise





Universiteit Leiden
The Netherlands

HighRes estimator

Note selection

- Set of peaks from peak picker
- QIFFT each peak
- Basic estimator: loudest peak
- Groups of overtones

Filtering

- Minimum peak/envelope/signal power
- Instrument range filter
- Signal-to-noise filter

Universiteit Leiden
The Netherlands

HighRes estimator

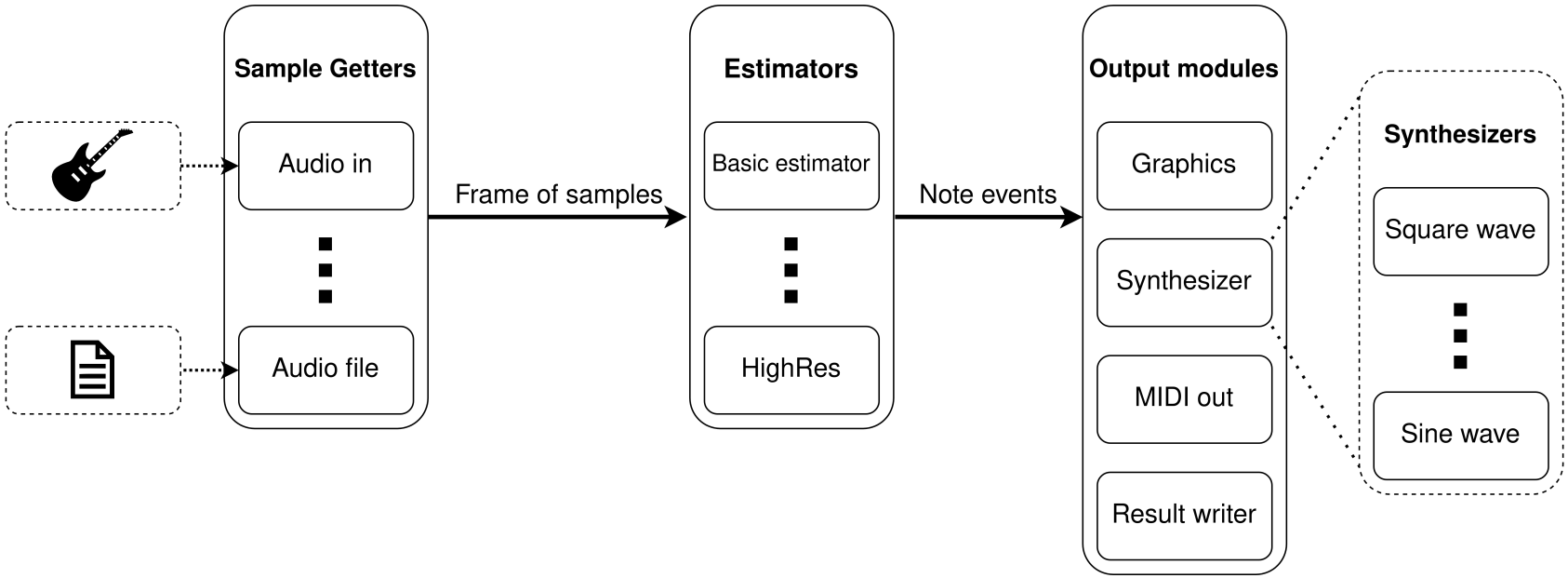
1. Window function
2. Fourier transform
3. Calculate amplitudes
4. Calculate Gaussian envelope
5. Pick peaks
6. Interpolate peaks
7. Select note

Note that the input signal is overlapped and zero-padded

Additional filtering at every stage

Universiteit Leiden
The Netherlands

Digistring



Universiteit Leiden
The Netherlands

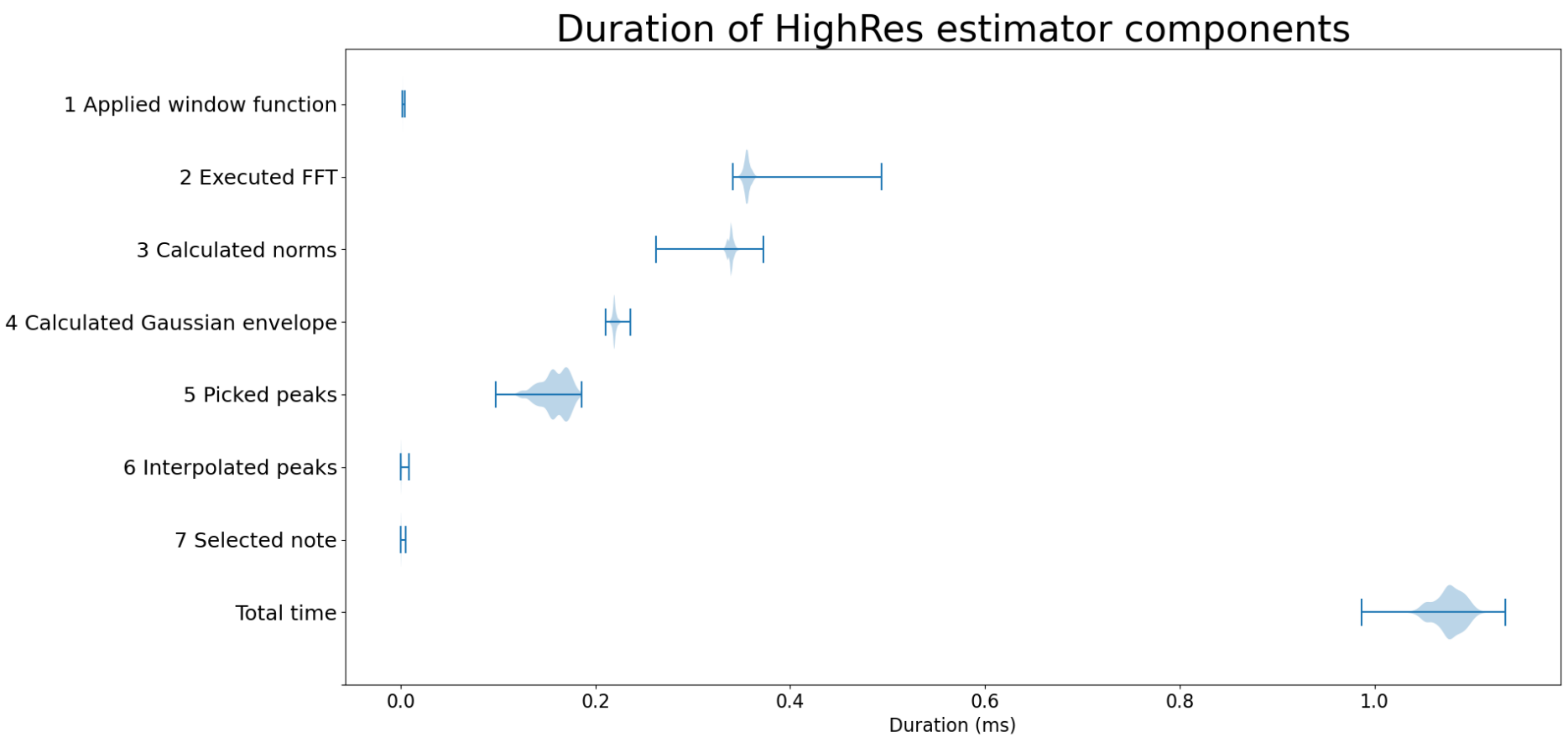
Experiments

Fourier size limit

- Resolution vs latency
- Lowest notes (E_2 and F_2)
- Pure sines: 5.33 ms
- Recordings: 30 ms
  - ↳ Dissonant
- Digistring: 42.67 ms

Universiteit Leiden
The Netherlands

Pitch estimation speed



Experiments

Pitch estimation accuracy

- Fraunhofer dataset
- Polyphonic $\rightarrow$ use a subset
- 4 subsets, 3 monophonic
- Challenging dataset

Universiteit Leiden
The Netherlands

Experiments

First subset

- Single note recordings
- All positions up to 12^{th} fret
- Every string
- 2 guitars
- 3 different pick-up combinations

version	correct	$t_{>0.9}$ correct	no interrupt
Fender	100 %	96 %	81 %
Ibanez N	100 %	96 %	76 %
Ibanez B	100 %	87 %	51 %
Ibanez N+B	100 %	92 %	67 %

Universiteit Leiden
The Netherlands

Experiments

Second subset

- 6 guitar licks
- 3 guitars
- Plectrum and finger style

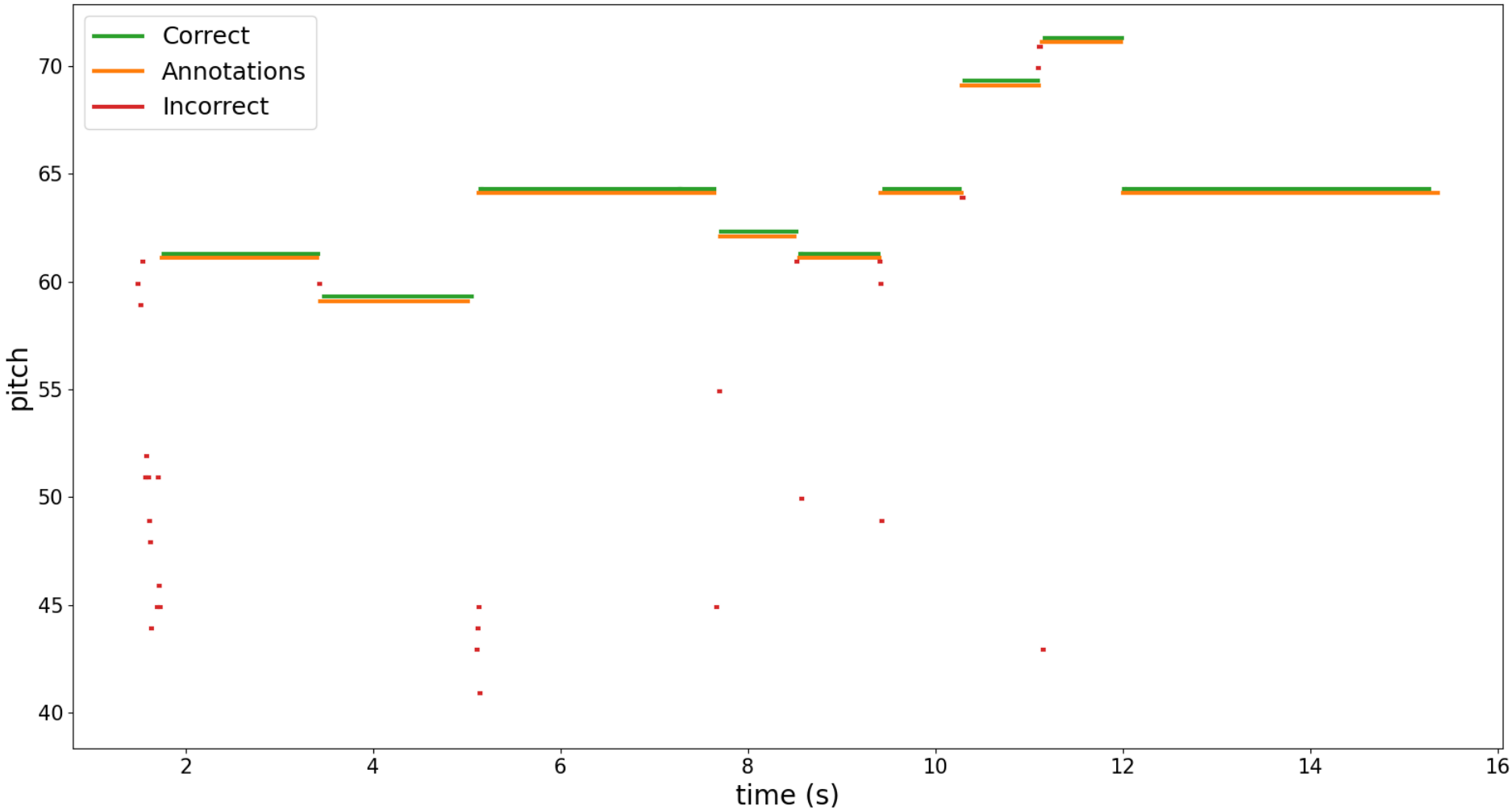
version	correct	$t_{>0.8}$ correct
Fender pick	100 %	83 %
Fender finger	100 %	58 %
Gibson pick	100 %	75 %
Gibson finger	100 %	58 %
Aristides pick	100 %	79 %
Aristides finger	100 %	63 %

Universiteit Leiden
The Netherlands

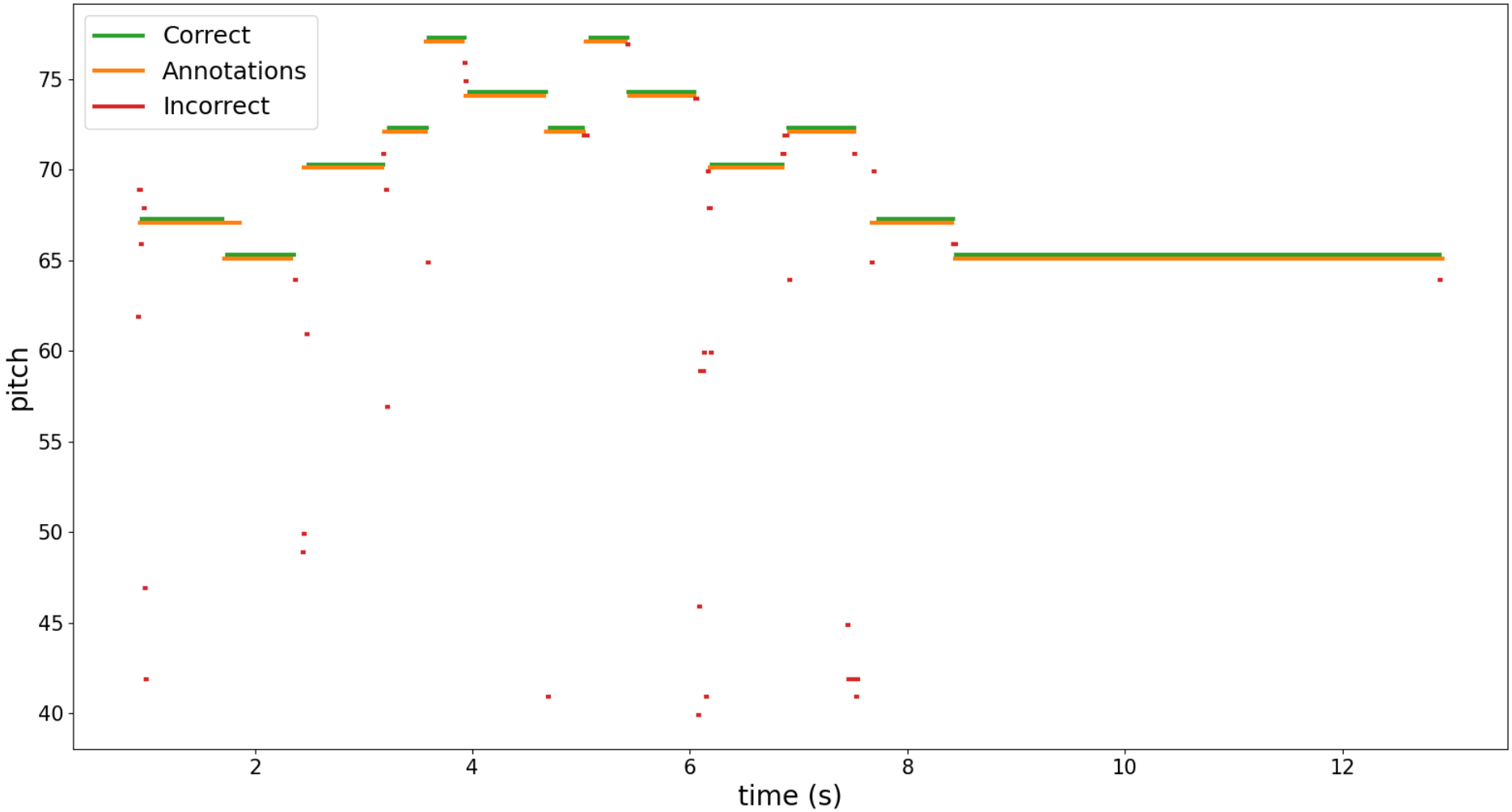
Experiments

Third subset

- 5 excerpts of classical pieces
  - ↳ Only 2 monophonic



Experiments



Conclusions

Contributions

- HighRes estimator
- Digistring

Conclusions

- Limited by Fourier resolution $\Leftrightarrow$ latency
- 200 ms frame $\rightarrow$ 43 ms frame
- Real-time constraint $\Leftrightarrow$ lowest note

Universiteit Leiden
The Netherlands

Digistring demo

Universiteit Leiden
The Netherlands