Mazurka Project

29 June 2006

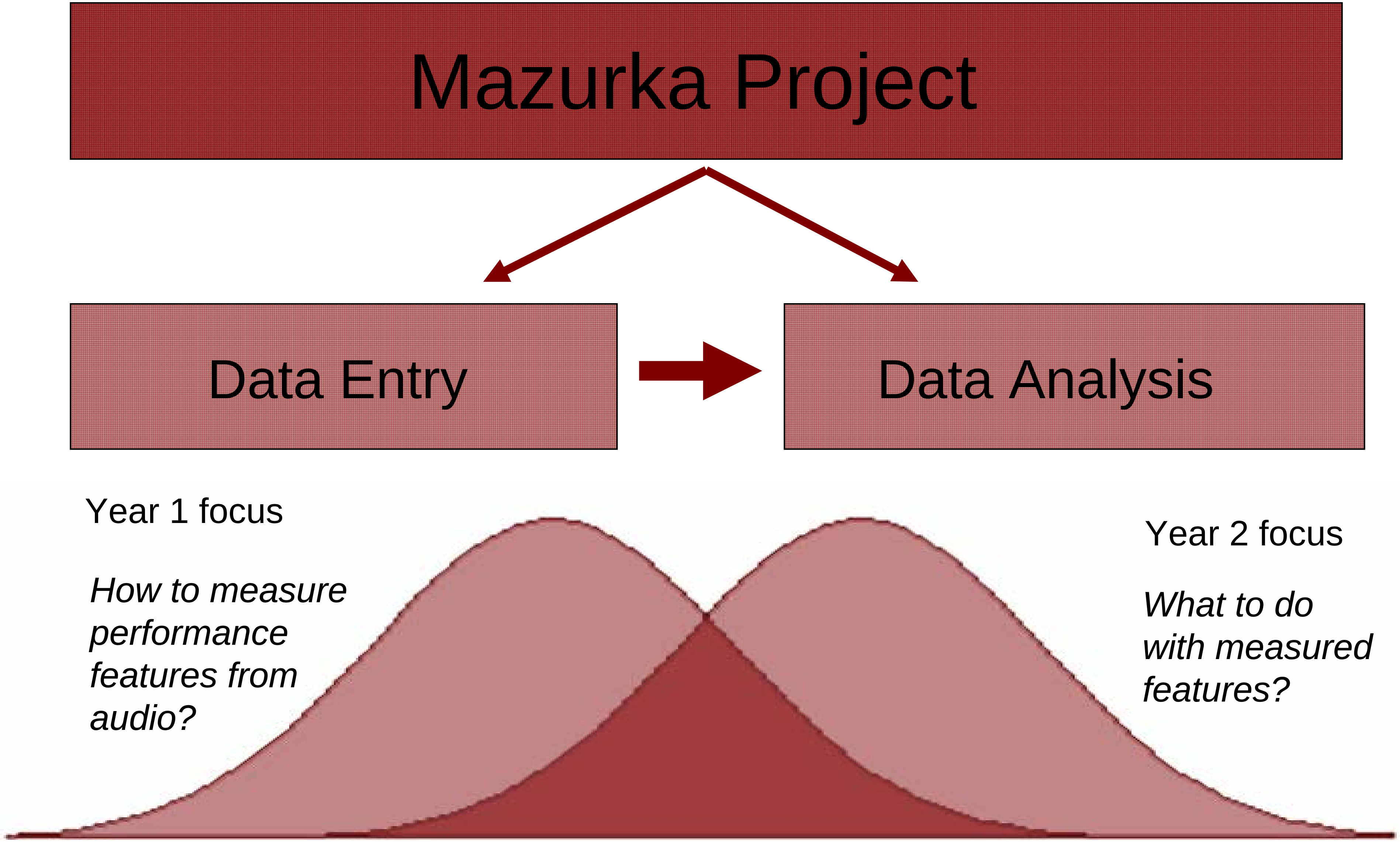
CHARM Symposium

Craig Stuart Sapp

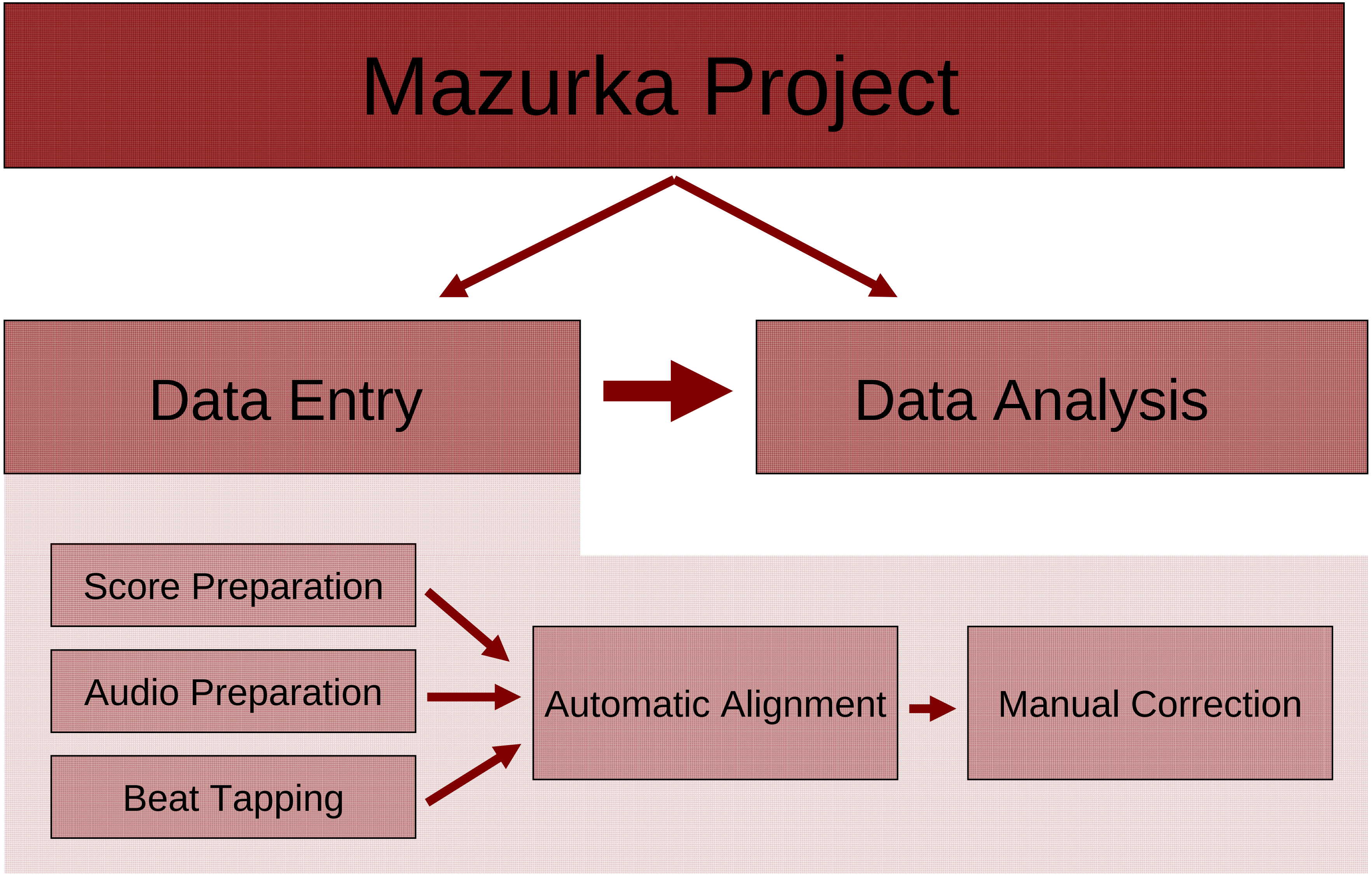
Sections

1. Overview
2. Power Measurements
3. Manual Correction
4. Automatic Alignment
5. Experiments
6. Performance Simulations
7. Initial Analysis

Top-Down Overview



Top-Down Overview



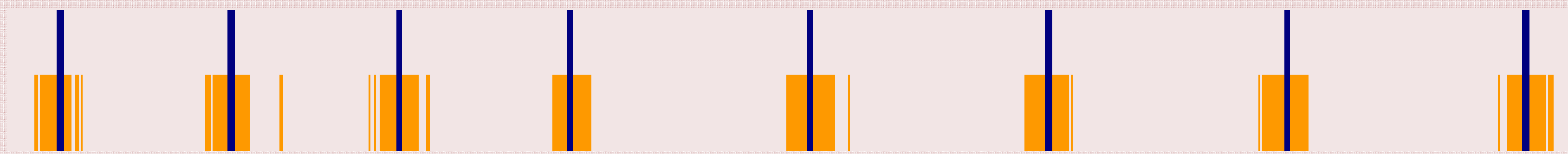
Data Entry Process

Beat Tapping → Automatic Alignment → Manual Correction

- Tapping to beats in audio
- Alignment of taps to audio times
- Previously done with non-graphical program
- Evaluating process in Sonic Visualiser

  - tapping alignment to audio done automatically
  - time resolution not as accurate? (maybe 10 ms instead of 5 ms)

- Tapping accuracy is about 50/60 ms away from beat on average for mauzurkas, or 25 ms away from beat for a steady tempo.

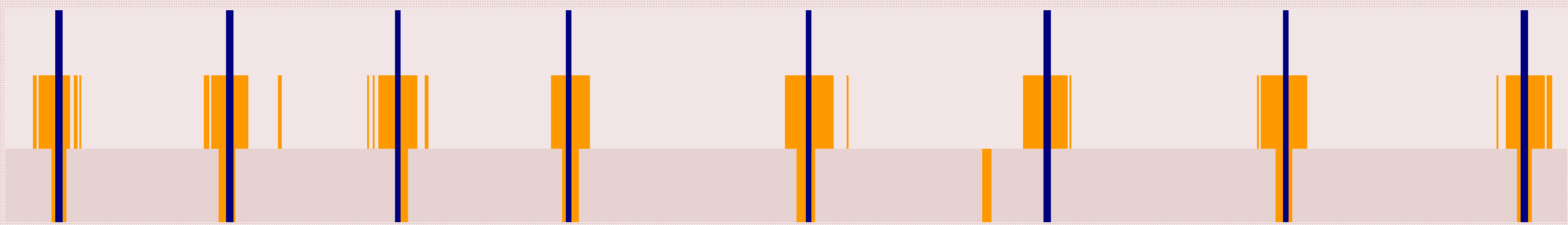


Data Entry Process

Beat Tapping → **Automatic Alignment** → Manual Correction

- Refines taps by searching for score notes in neighborhood
- Estimates and locates non-beat notes
- Measures event amplitudes (loudness)

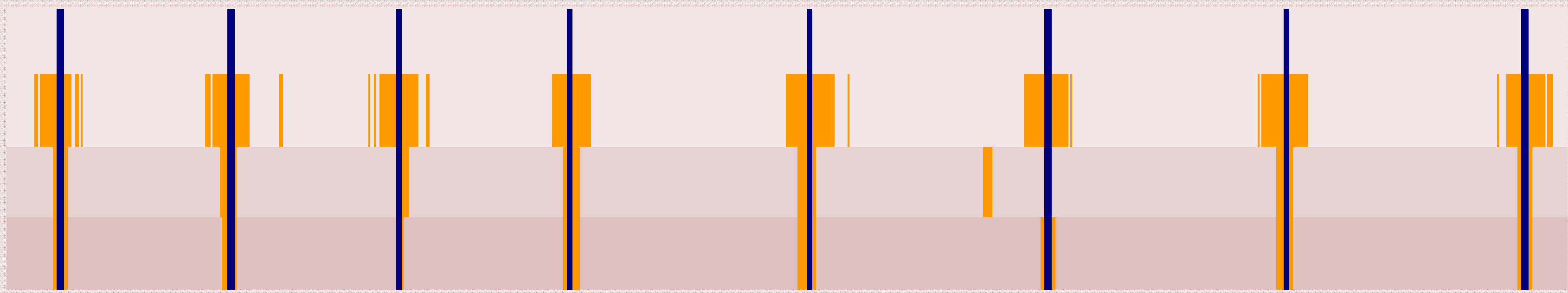
- Improves on tapping positions by 4-5x for modern recordings
- 2-3x improvement for historic (noisy) recordings



Data Entry Process

Beat Tapping → Automatic Alignment → **Manual Correction**

- Fix obvious errors in automatic alignment
- Verify automatic alignment by listening to clicks and music simultaneously



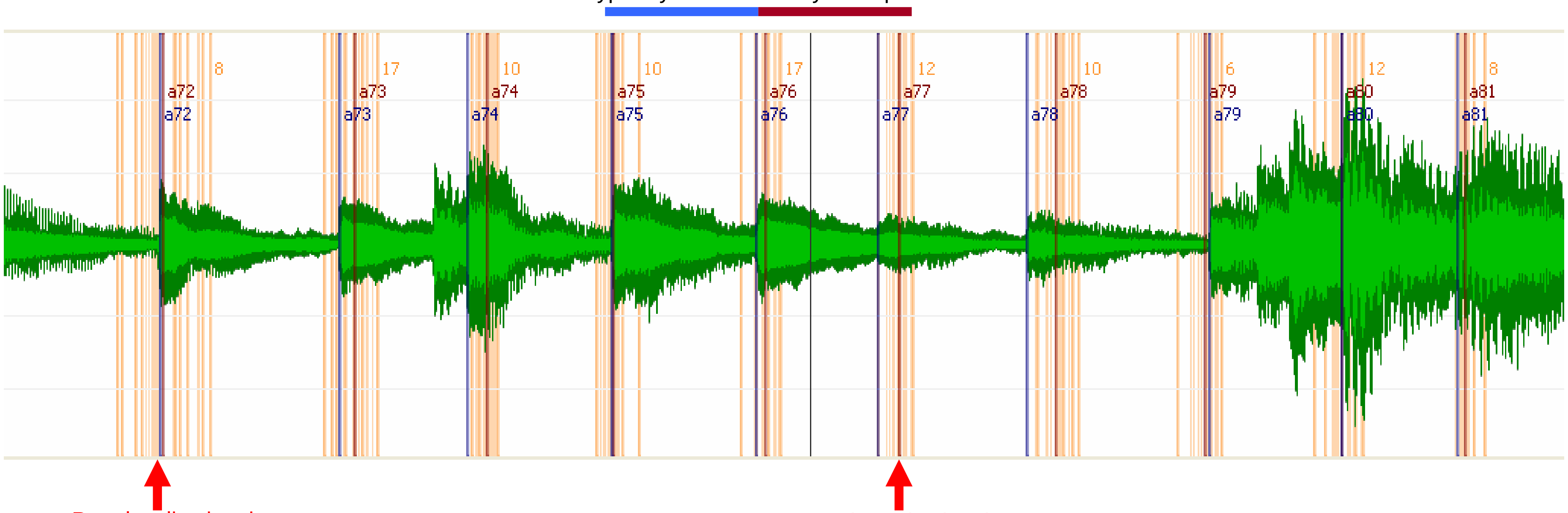
Reverse Conducting

- Orange = individual taps (multiple sessions) which create bands of time about 100 ms wide.

- Red = average time of individual taps for a particular beat

- Blue = actual beat onset times

Typically one beat delay in response



Poor localization, but accurate average

Good localization, but not accurate average

Power Measurements
(for manual corrections)

MzPowerCurve

- *Sonic Visualiser plugin to do various power measurements*

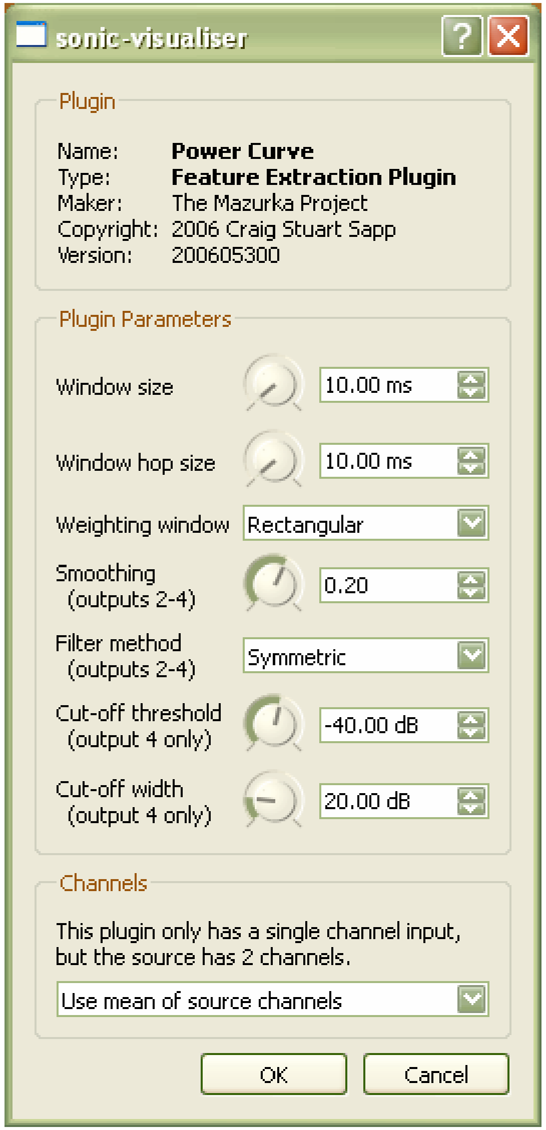
  http://sv.mazurka.org.uk/MzPowerCurve

- #1 raw power measurements – average and weighted

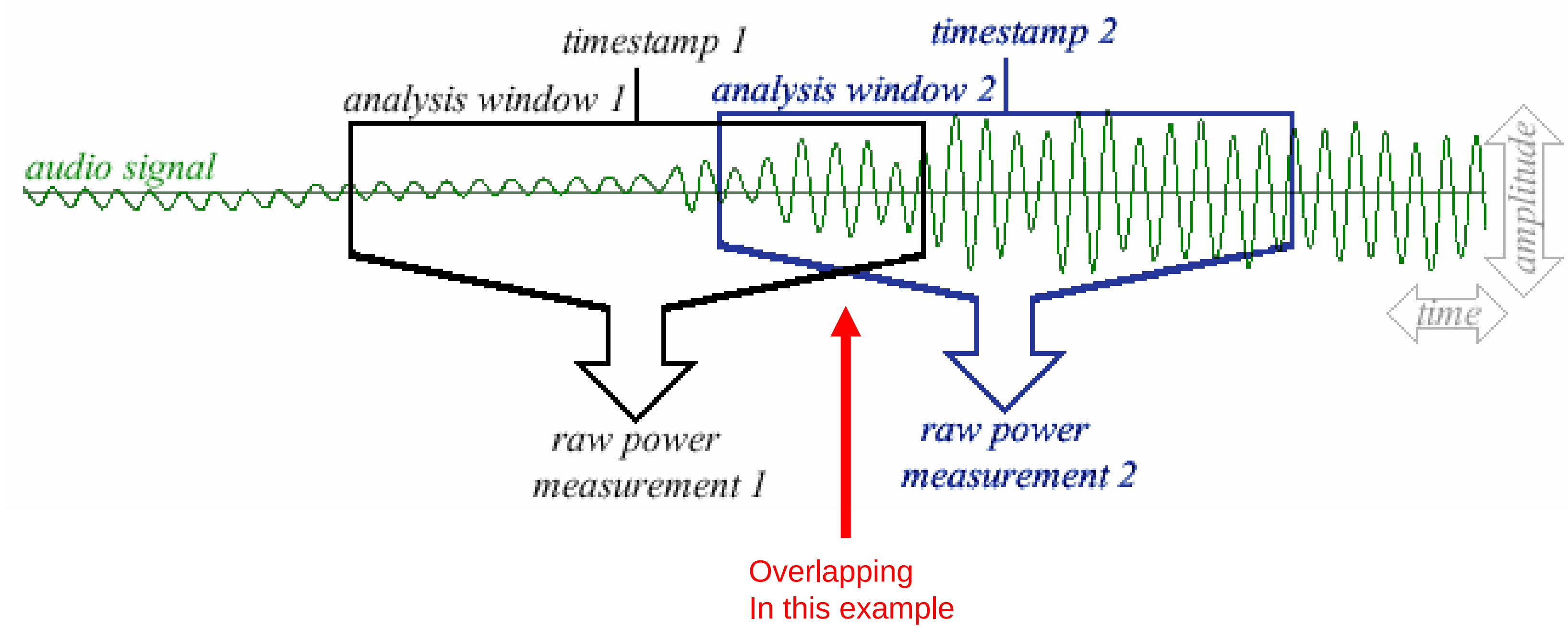
$$P_{\text{avg}} = 10 \log_{10} \left(\frac{1}{N} \sum_n x_n^2 \right) \qquad P_{\text{wavg}} = 10 \log_{10} \left(\frac{1}{N} \sum_n x_n^2 w_n \right)$$

- #2 smoothed power – useful for basic dynamics measurements

- #3,4 smoothed power slope – useful for manual corrections of note attacks (for percussive instruments such as piano).



Window and Hop



Raw and Smoothed power

(Measuring power every 10 ms, with a 10 ms window.)

- Smoothed power useful for getting basic dynamics levels.

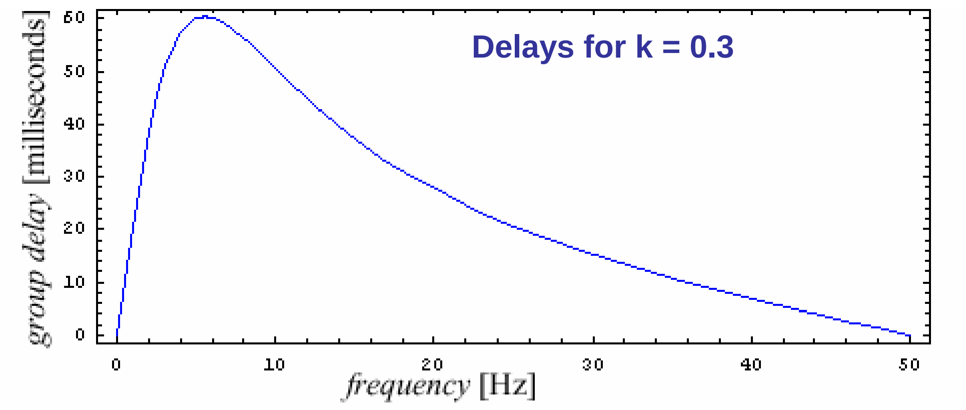
Smoothing Filter

- Using a filter called an *exponential smoother*:

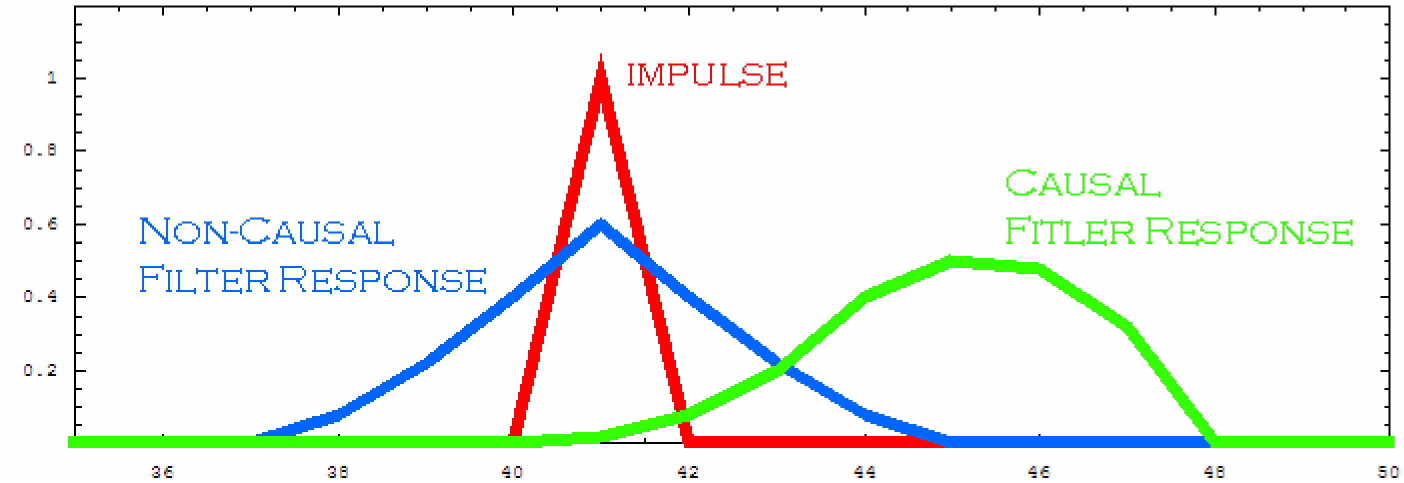
$$y[n] = k\ x[n] + (1-k)\ y[n-1]$$

Englishish: The current output equals the current input times the value *k*, plus the previous output times the value 1-*k*.

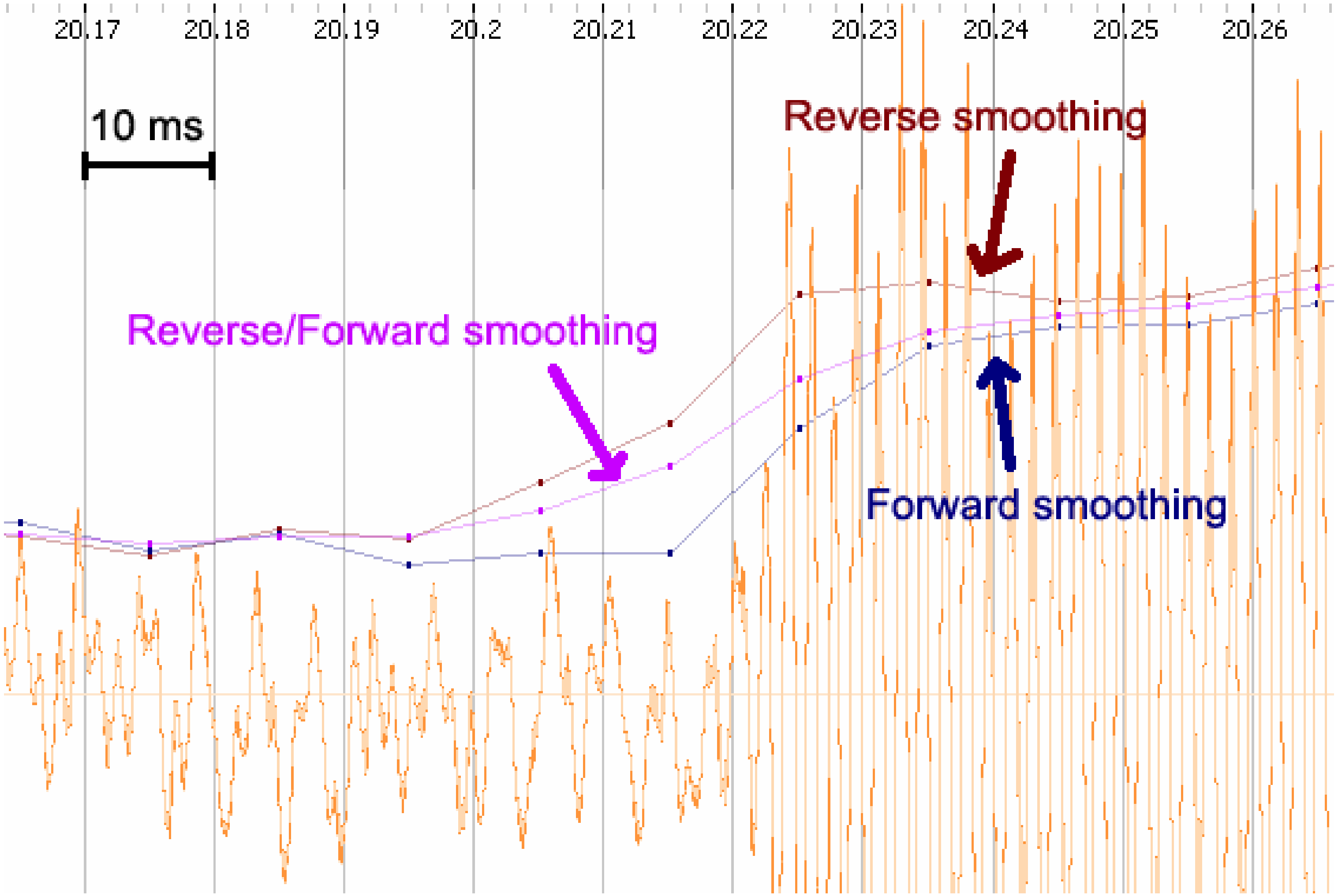
- All filters delay the input. Since this filter feeds back on itself, the filter will delay some frequencies more than others:



Symmetric Filtering



Filtering Direction



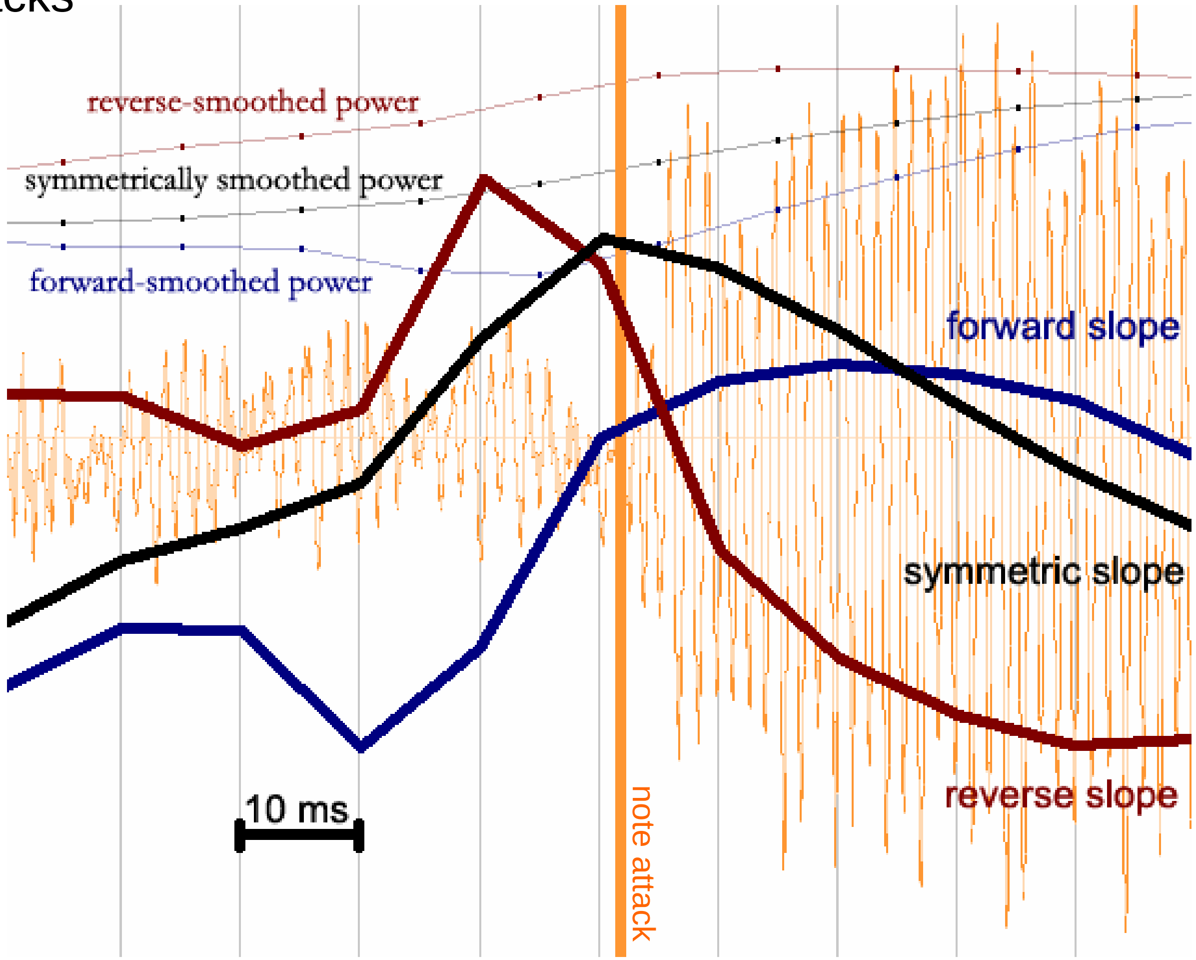
Smoothing Direction

• Avoid the funny delays by symmetric filtering

• Then slope of smoothed power aligns nicely with percussive note attacks

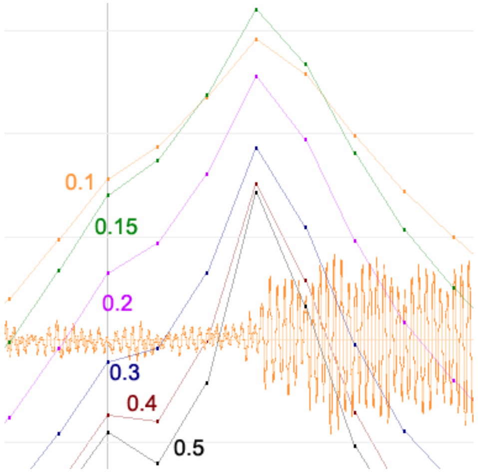
• symmetric filtering is best for localizing attacks

• reverse filtering is best for dynamics estimation

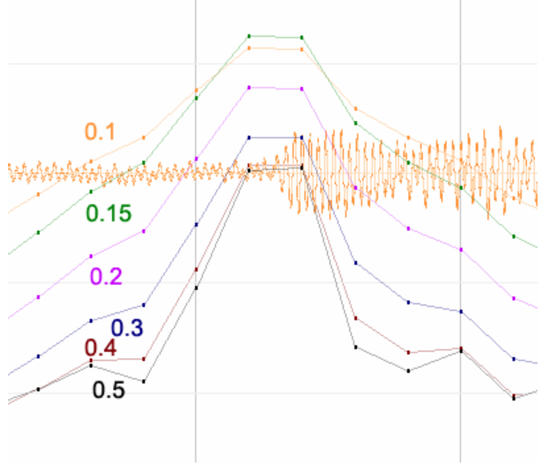


Smoothed Power Slope

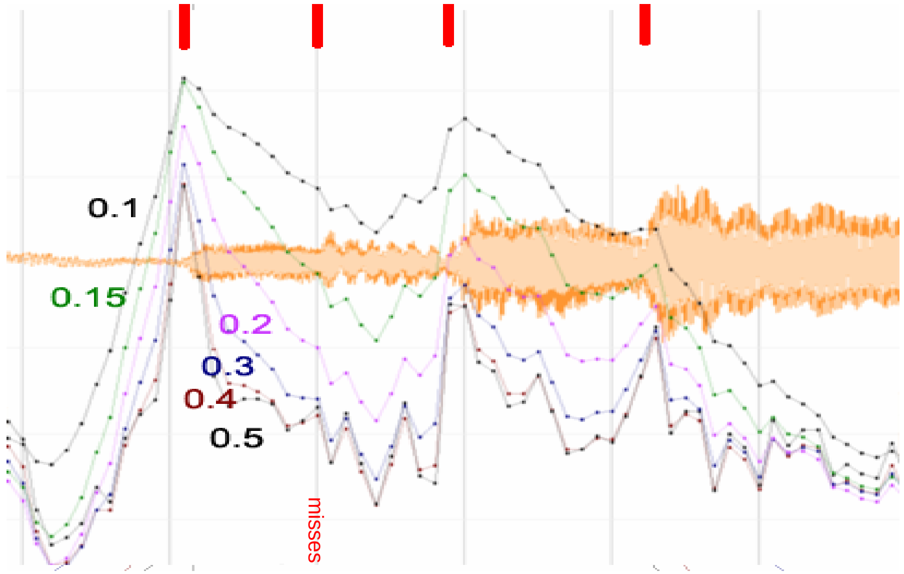
Attack on measurements



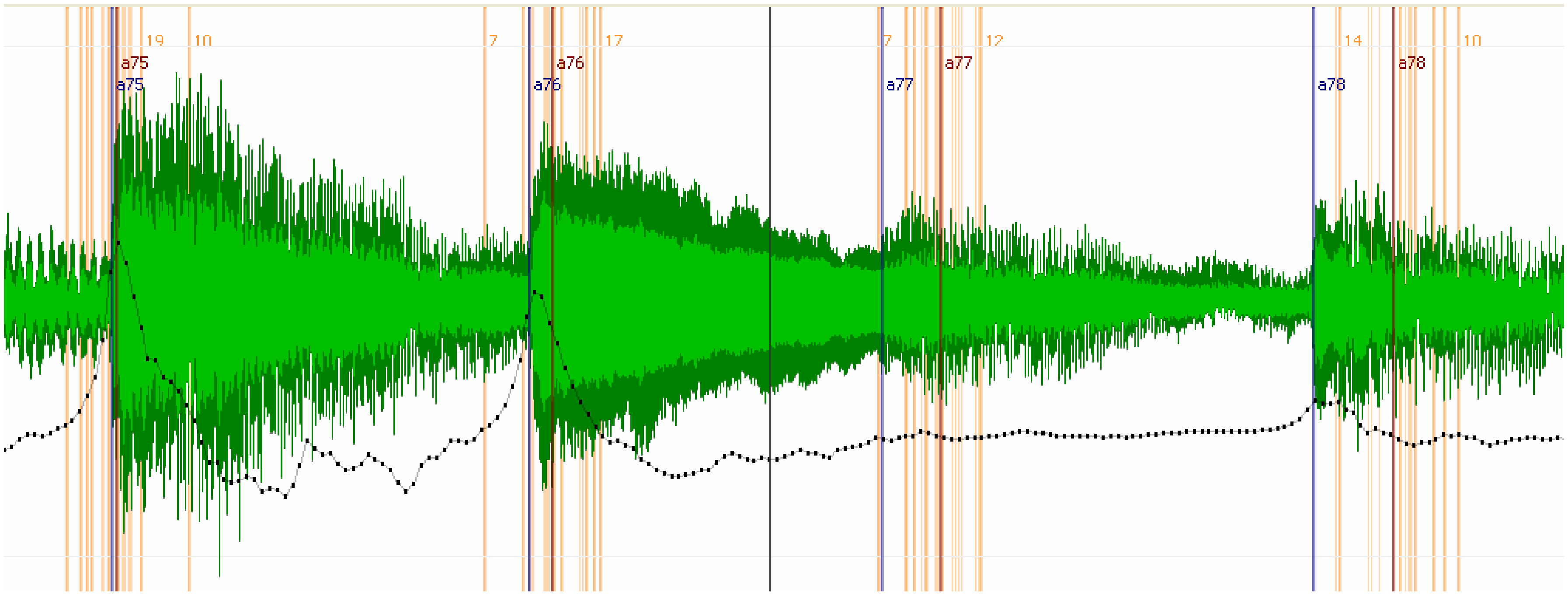
Attack ½ between measurements



• smoothing factor of about 0.2 gives best results over a variety of conditions



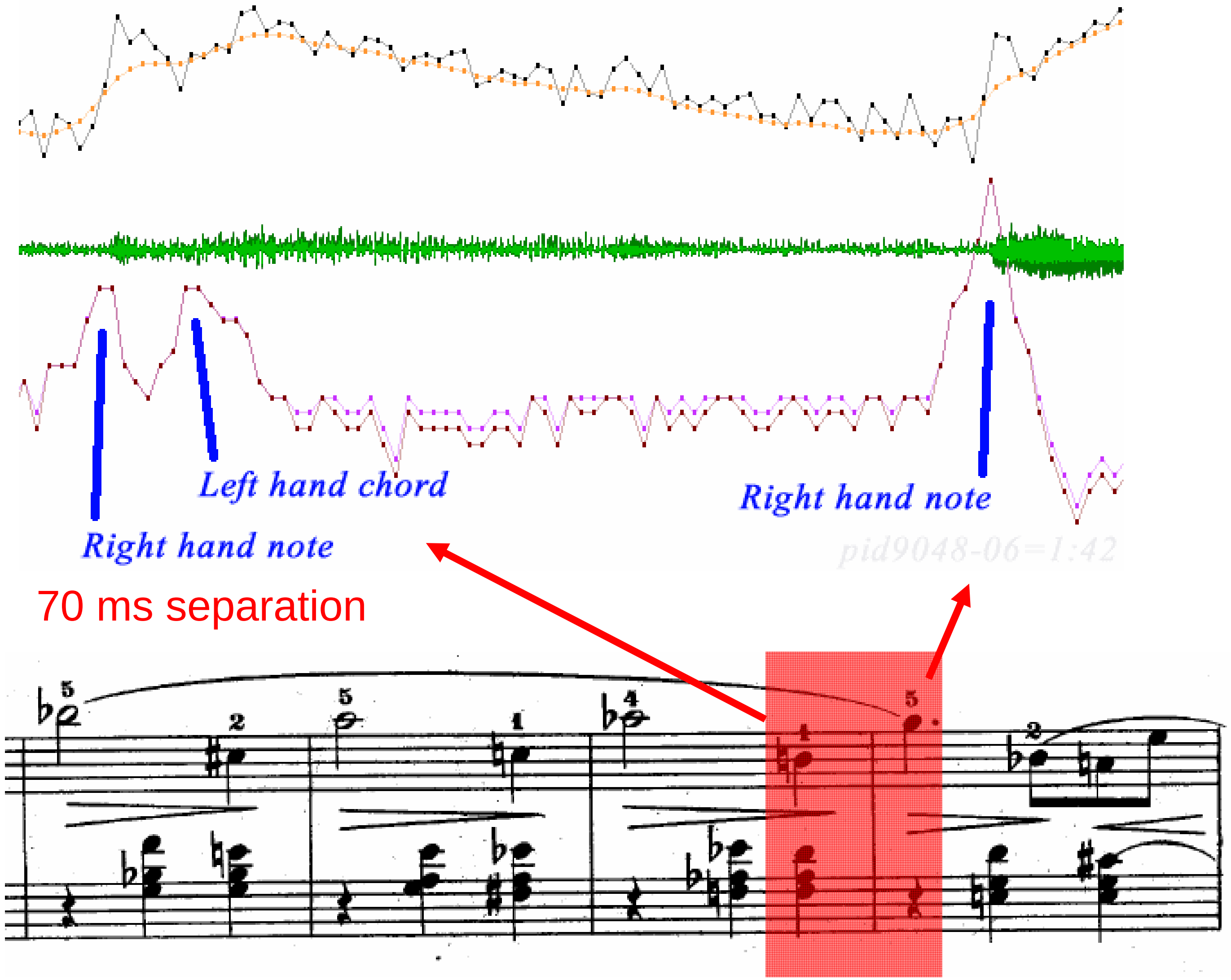
Power Slope for Correcting



Not helpful for this beat (no peak)

Non-Synchronous Hands

- *> 30 ms separation may be significant*



- Pianist is probably conveying a sense of relaxation at this point in music

Advantages/Disadvantages

• Time domain analysis, so localization can be better than for frequency analysis metrics (E.g. Earis & Bello methods)

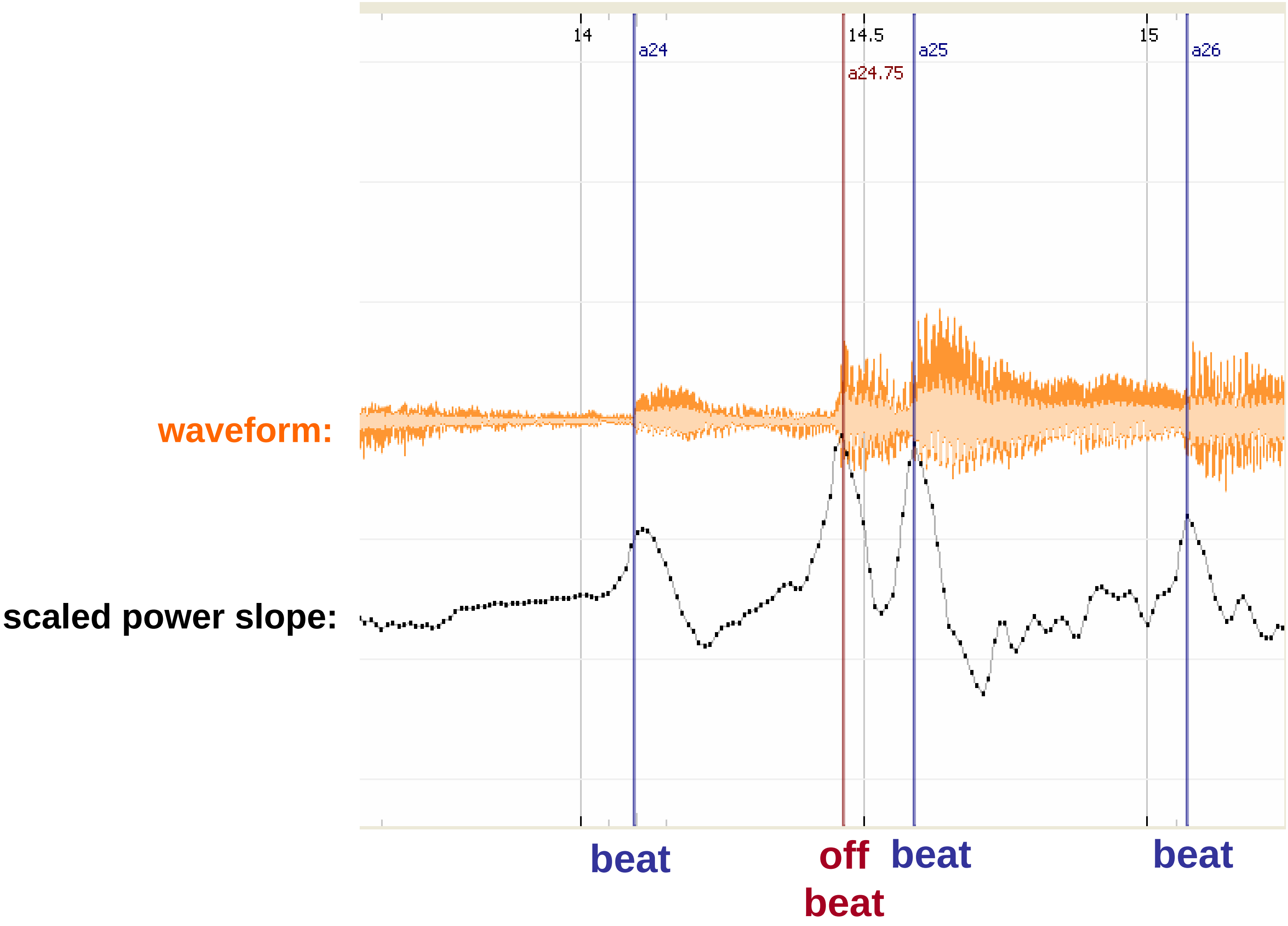
• Ignores frequency content, so not always or accurate.

•Good for instruments with percussive attacks (i.e. piano, drums)

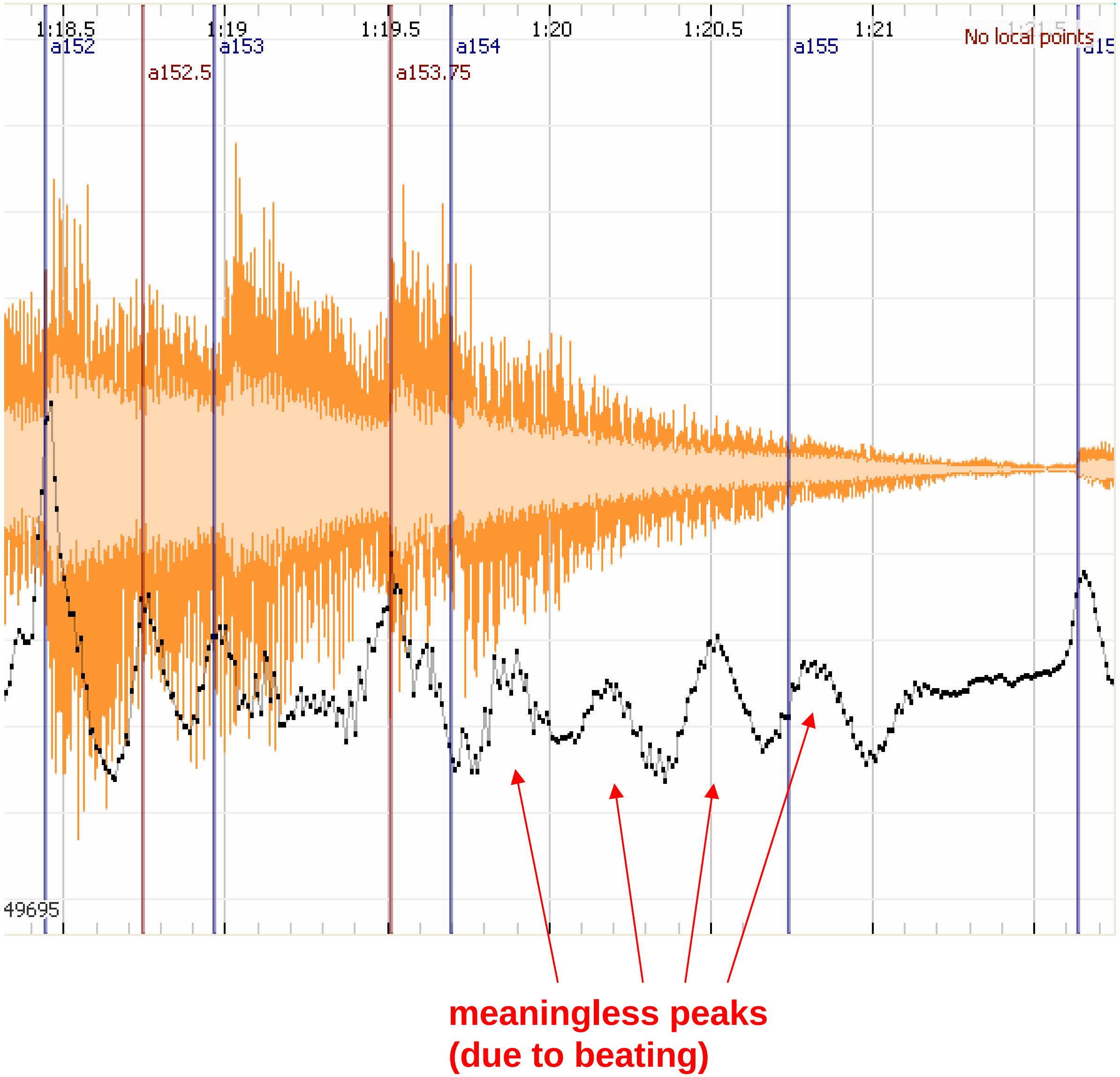
• Probably not good for non-percussive instruments: voice, violin, woodwinds, brass, etc.

Manual Correction

Beats + Offbeats

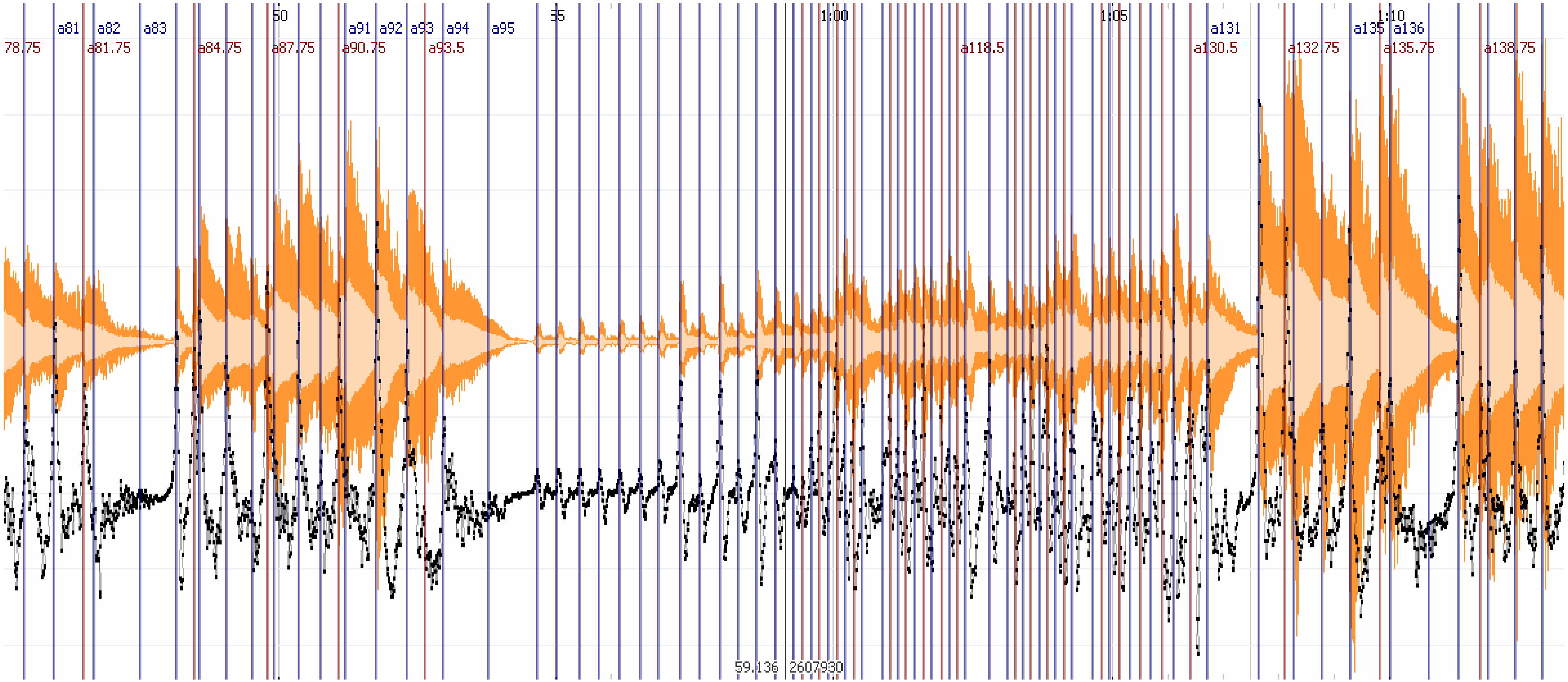


Advantage of Having the Score



- If you don't have a score, you are wasting your time.

Large-scale View of Beats/Offbeats



• Layer clicks can be played with different timbres/loudness.

Probable Entry Scenario

0. Become familiar with the performance. (Score already entered) **(15 min)**

- Tap to performance in Sonic Visualiser **(5 min)**
- Cursory check of beat positions with onset annotations **(10 min)**
- Interpolate off-beat positions based on score
- View/listen to audio with beats/off-beats and automatic annotations **(10 min)**
- Automatic adjustments of the onset times of beats/off-beats
- Careful manual proof listening/reading of the automatically adjusted times **(30 min)**
- Extract secondary performance features such as dynamics and non-simultaneous chord notes.

red: manual time estimates for a 5 minute piece

→ *about 2 hours for 5 minutes of music*

Automatic Alignment Evaluation

Summary

• Automatic alignment improves accuracy about 4-5x for modern recordings and 2-3x for historic recordings when compared to reverse conducting accuracy.

• **Earis system parameter search optimization**

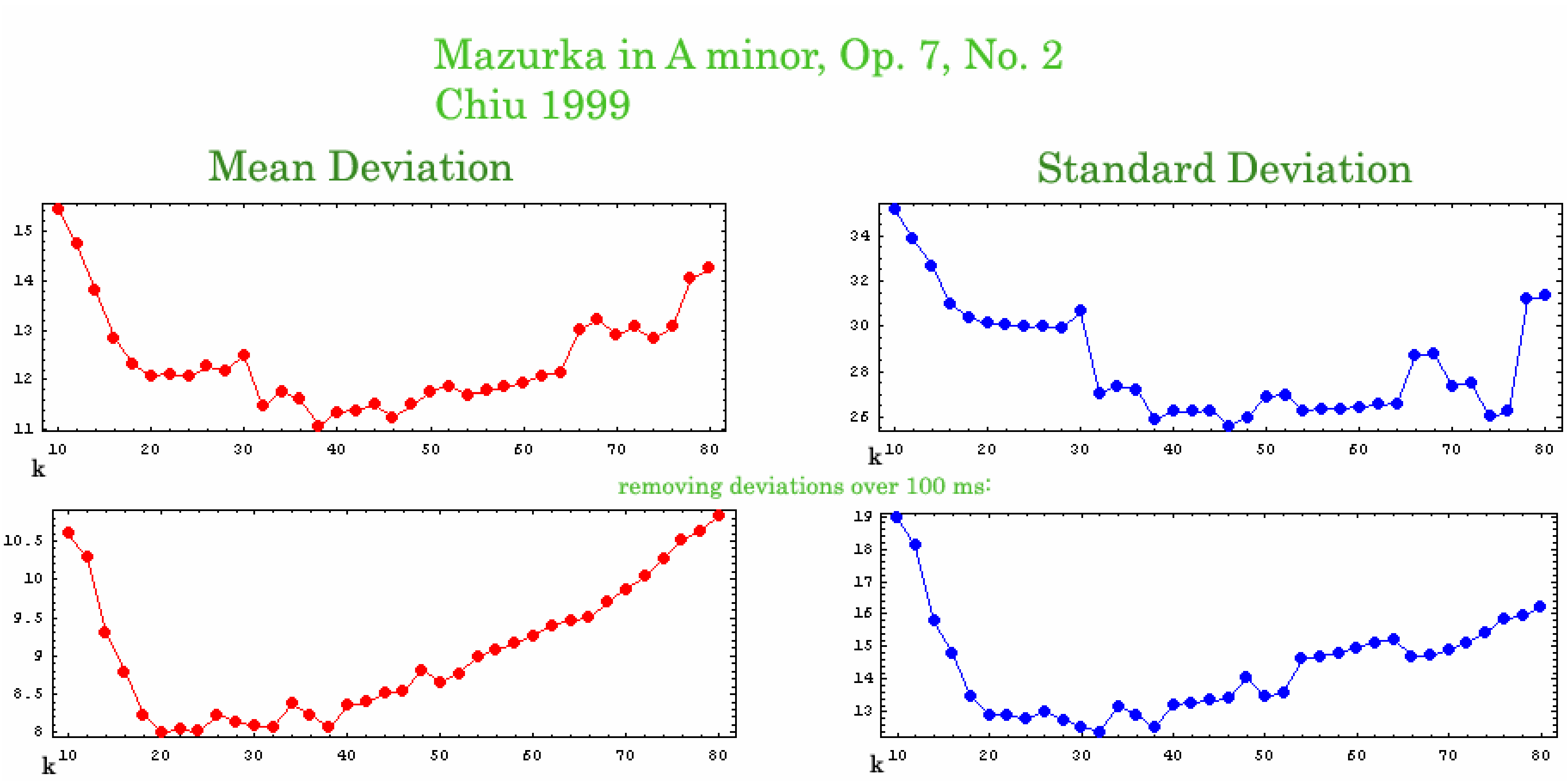
1. wavenumber (k)
2. low-pass filter order (LPF)
3. tuning factor

• **Other evaluation/exploration for the system:**

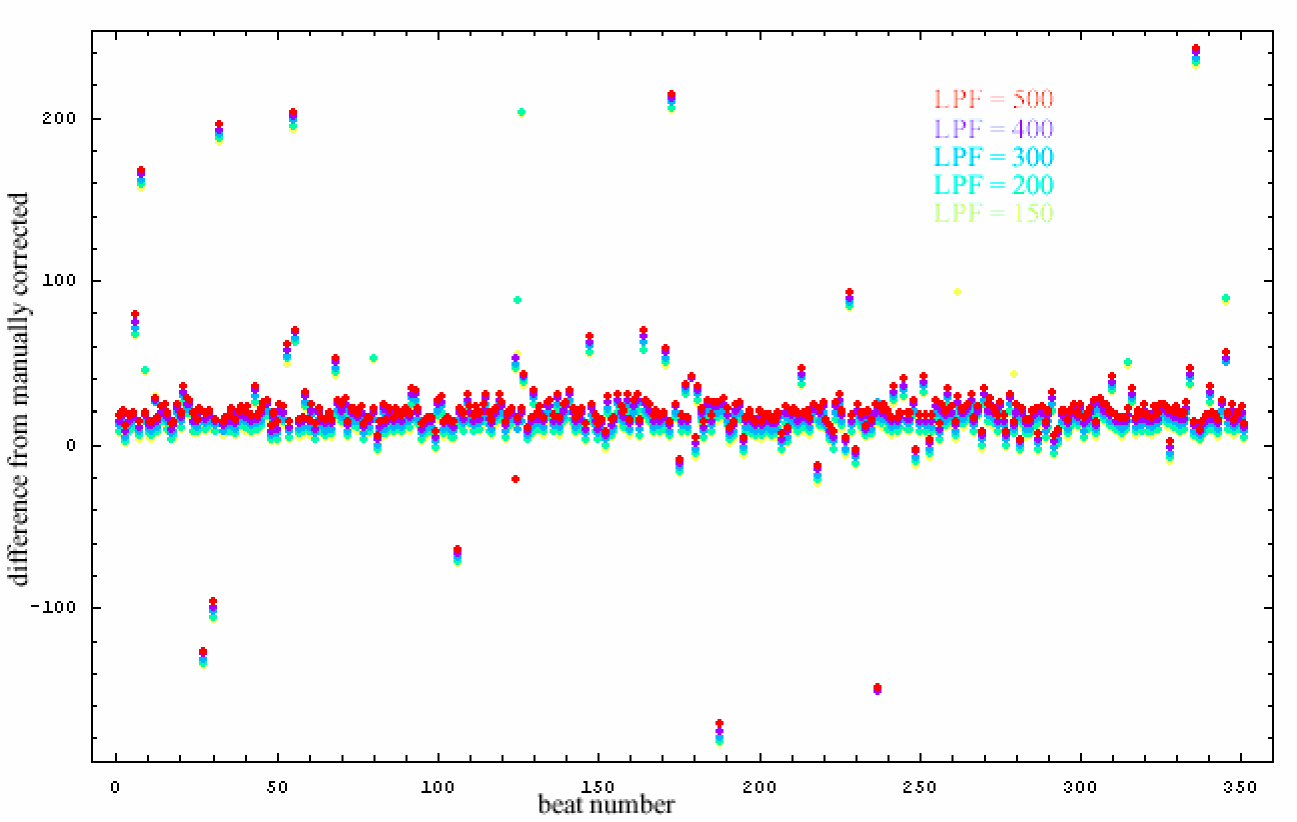
1. search window method
2. square/Gaussian window method
3. recursive processing
4. wanderer identification
5. removing harmonics of previous event
6. symmetric LPF filtering

k Parameter Sensitivity

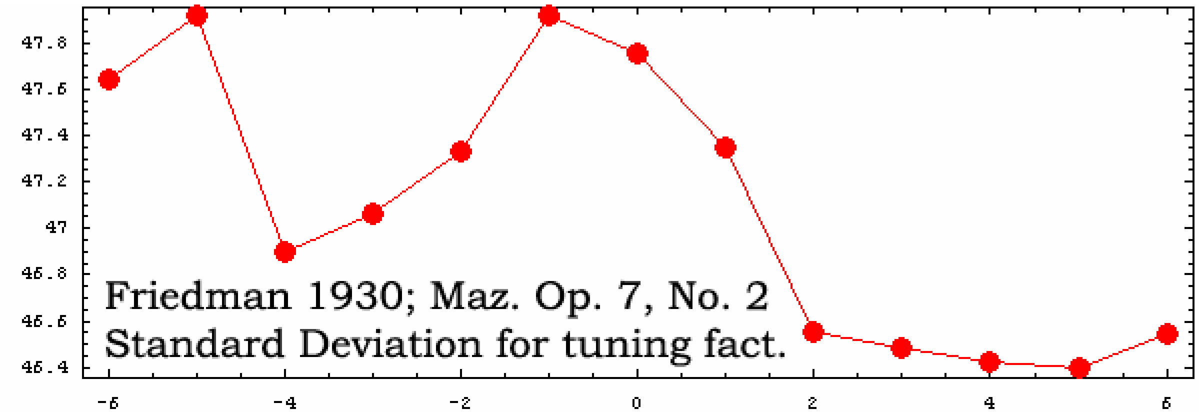
k = wave number (how many cycles of a sinewave) to analyze with

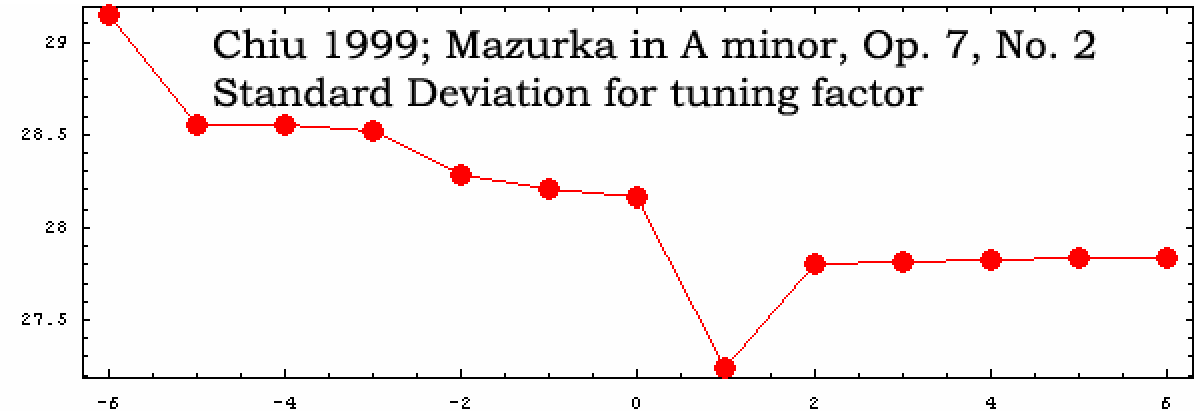


LPF Parameter

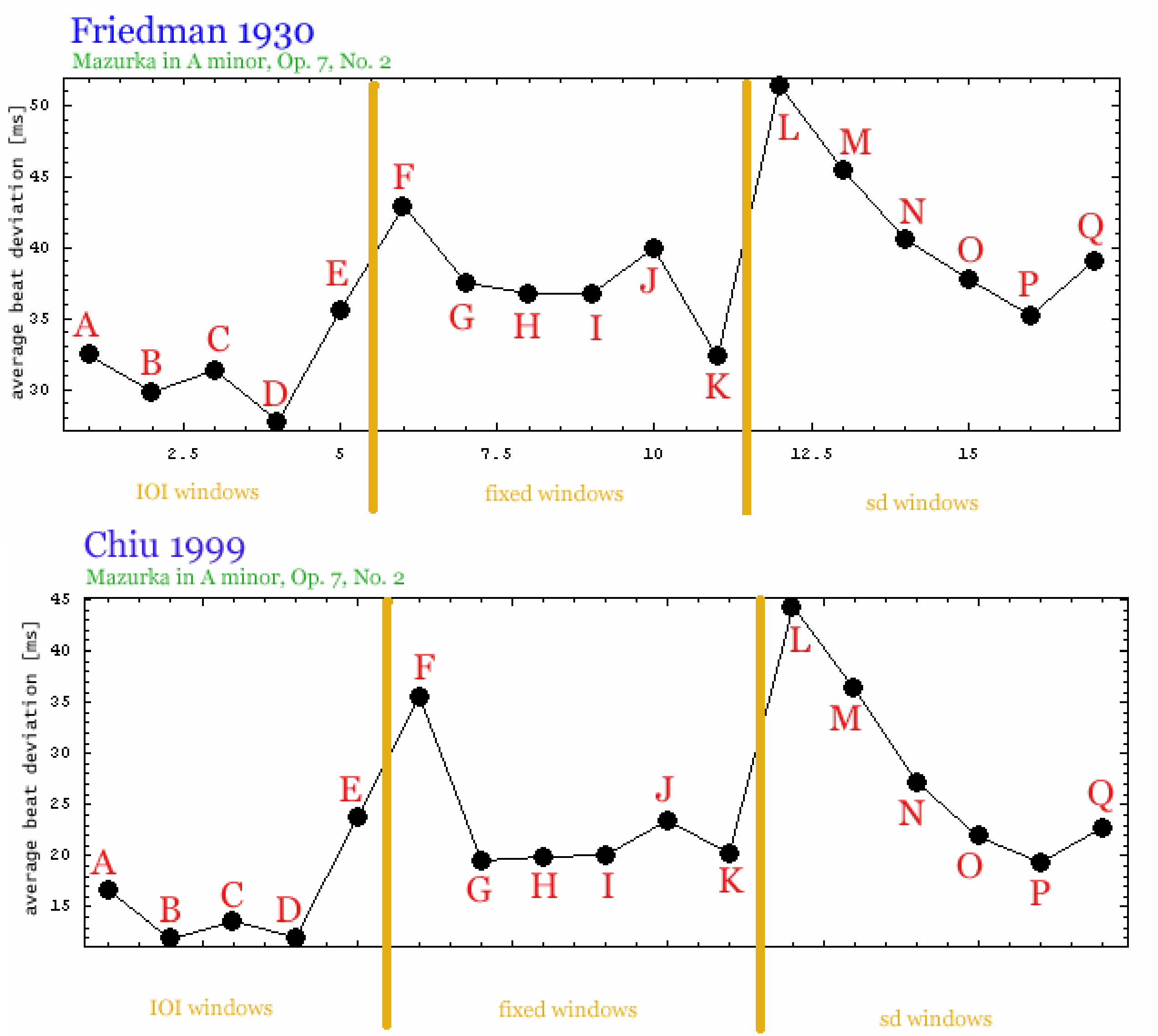


Tuning parameter





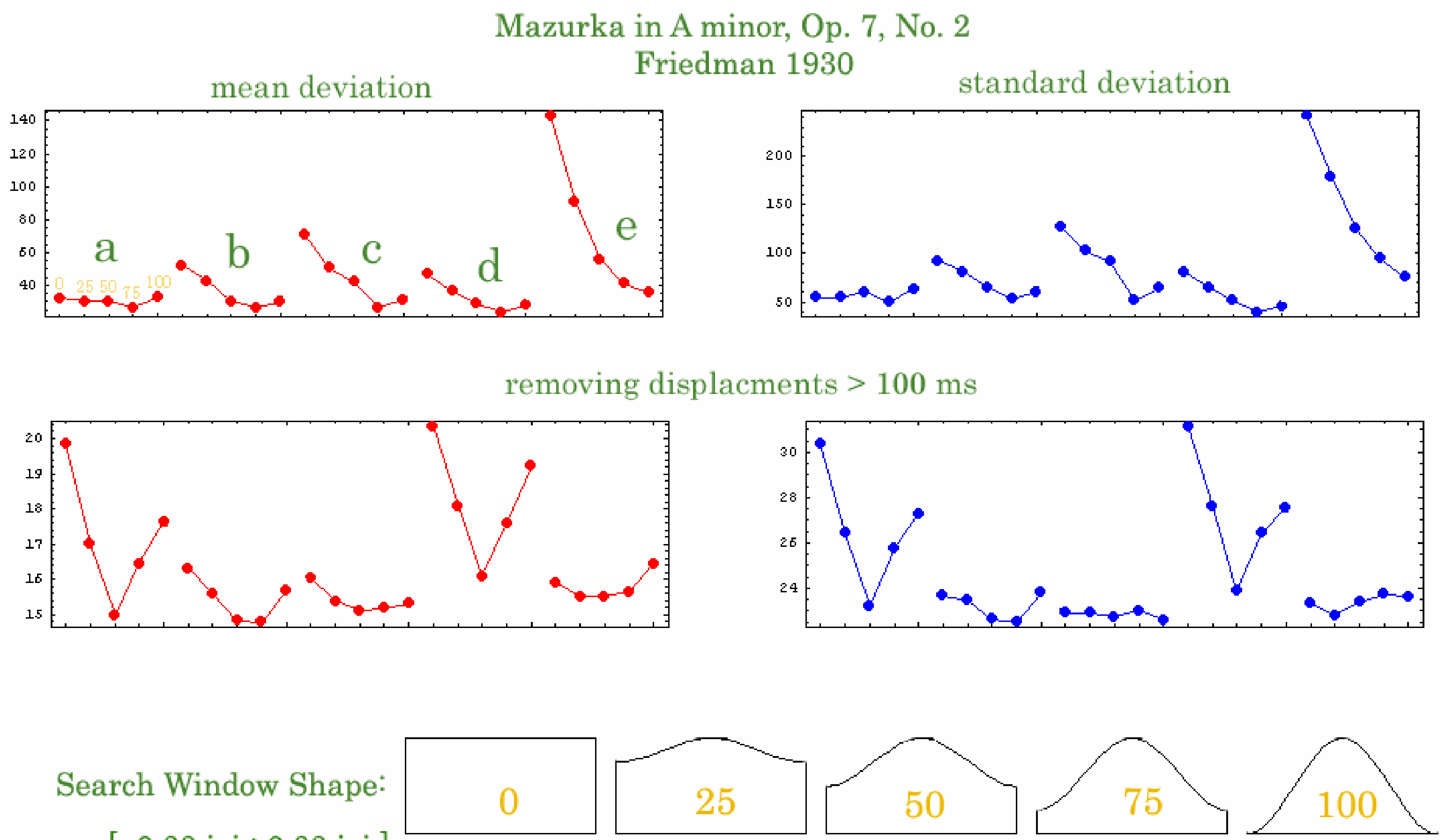
Windowing Methods



a = [-0.33 ioi : 0.33 ioi]
b = [-0.50 ioi : 0.50 ioi]
c = [-0.67 ioi : 0.67 ioi]
d = [-0.33 ioi : 0.67 ioi]
e = [-1.00 ioi : 0.67 ioi]
f = [-100 ms : 100 ms]
g = [-200 ms : 200 ms]
h = [-300 ms : 300 ms]
i = [-400 ms : 400 ms]
j = [-500 ms : 500 ms]
k = [-200 ms : 500 ms]
l = [-1 sd : 1 sd]
m = [-2 sd : 2 sd]
n = [-3 sd : 3 sd]
o = [-4 sd : 4 sd]
p = [-5 sd : 5 sd]
q = [-2 sd : 4 sd]

- IOI is best method

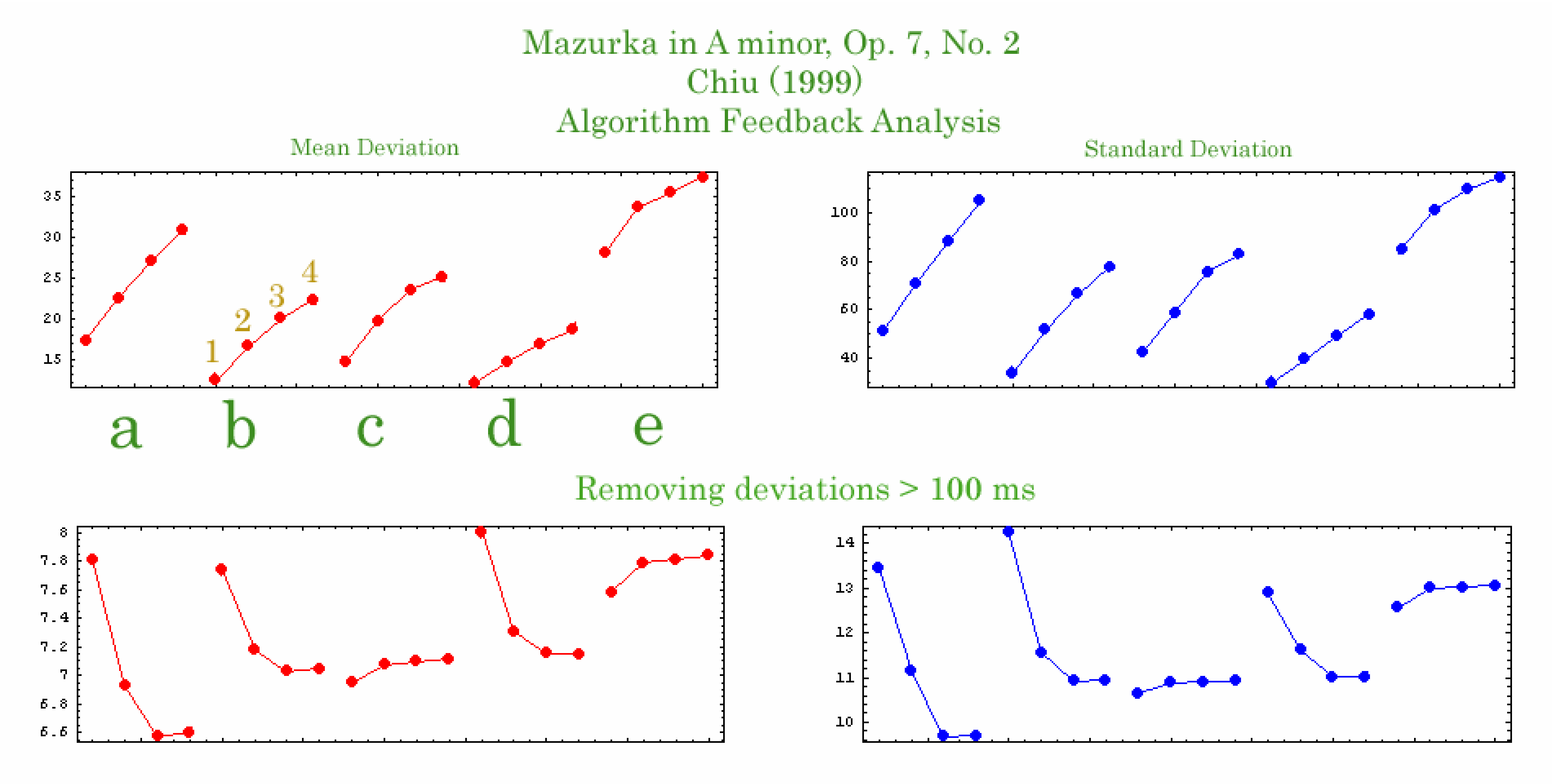
Hybrid Window Shape



a = [-0.33 ioi : 0.33 ioi]
b = [-0.5 ioi : 0.5 ioi]
c = [-0.67 ioi : 0.67 ioi]
d = [-0.33 ioi : 0.67 ioi]
e = [-1 ioi : 1 ioi]

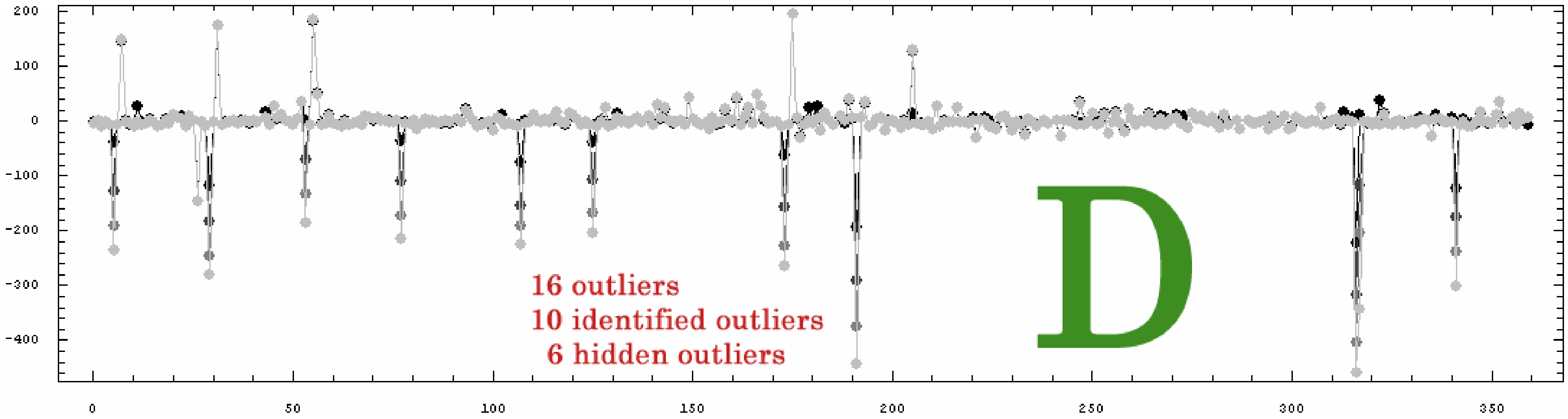
	all beats	good beats
Chiu 1999:	100%	60%
Friedman 1930:	75%	50%

Recursion effect on localization

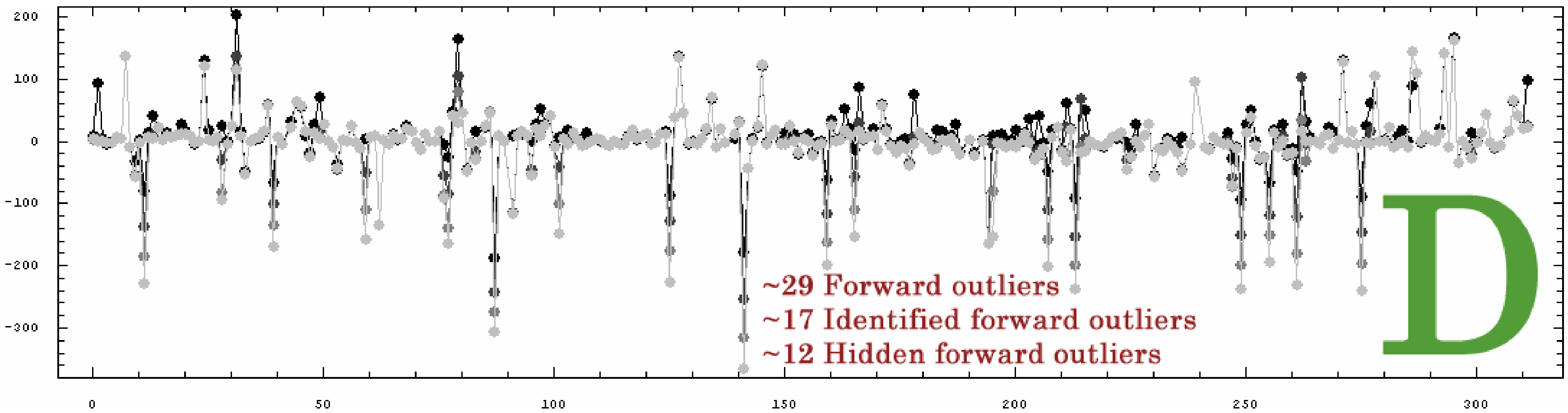


Recursion & Wandering

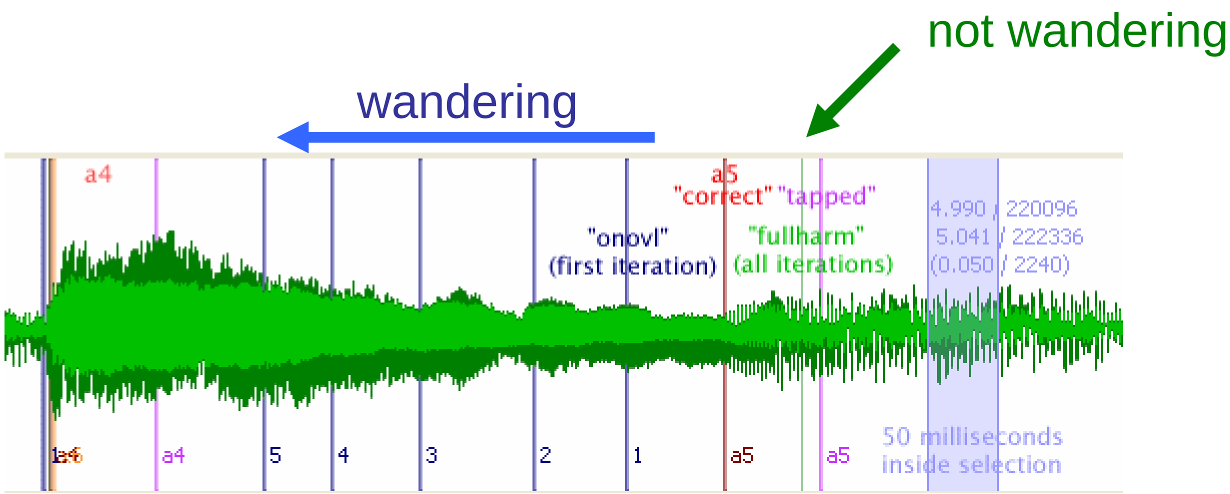
modern recording



historic recording

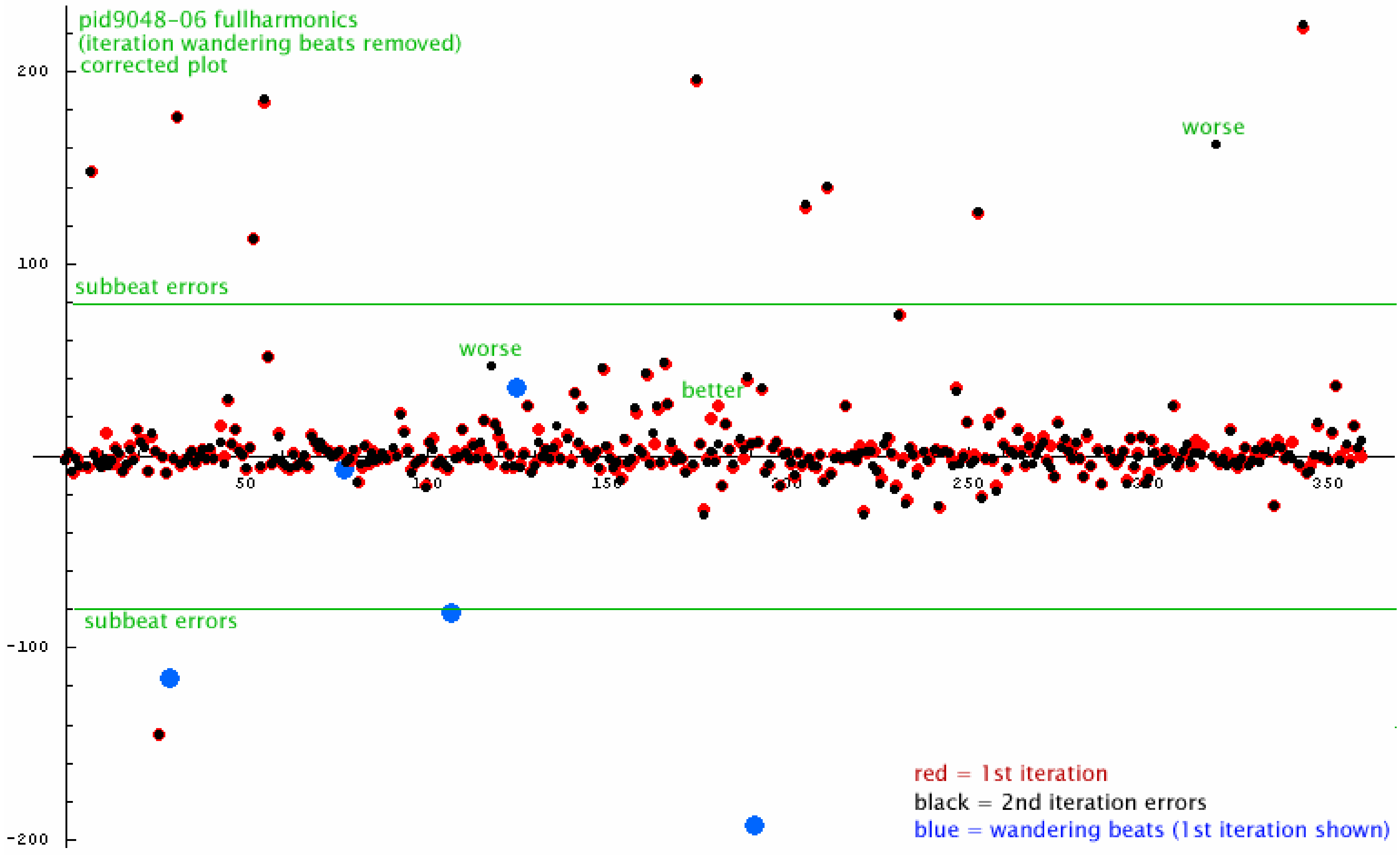


Wandering

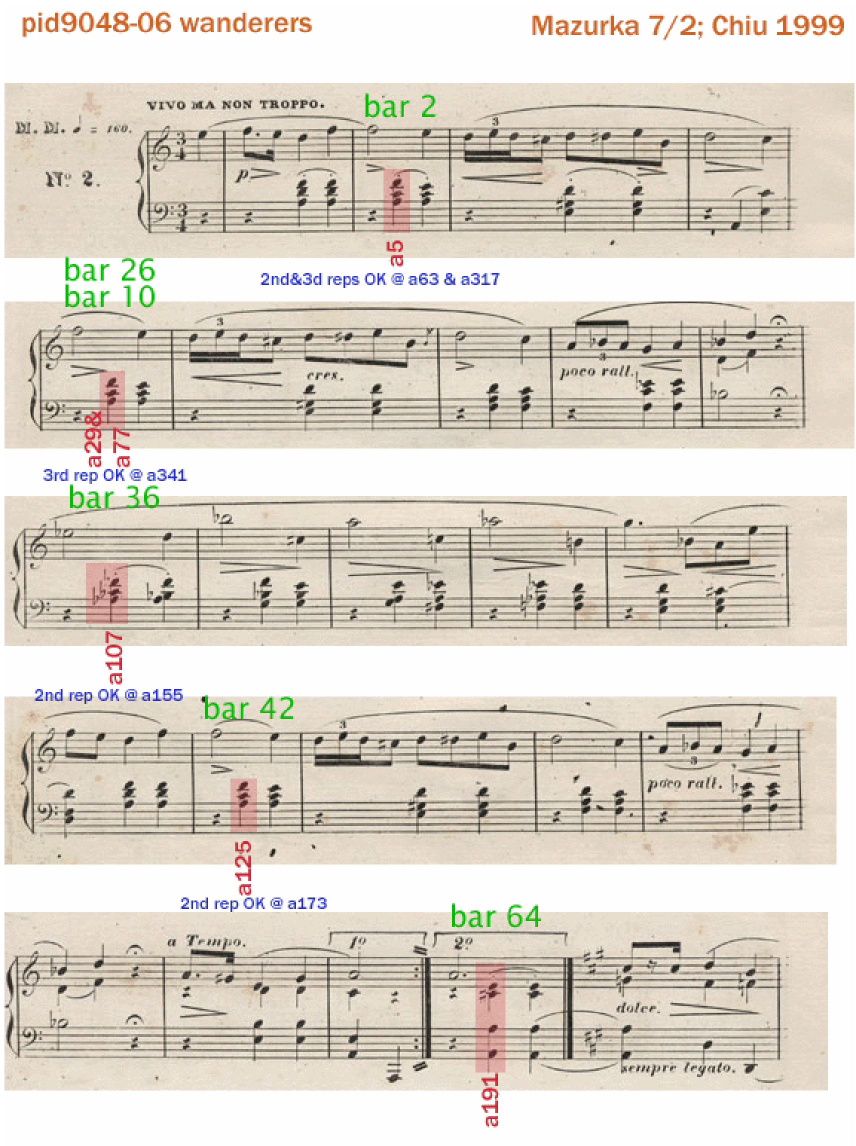


not wandering
wandering
a4
a5
"correct" "tapped"
"onovl"
(first iteration)
"fullharm"
(all iterations)
4.990 / 220096
5.041 / 222336
(0.050 / 2240)
a4
5
4
3
2
1
a5
a5
50 milliseconds
inside selection

Wanderers

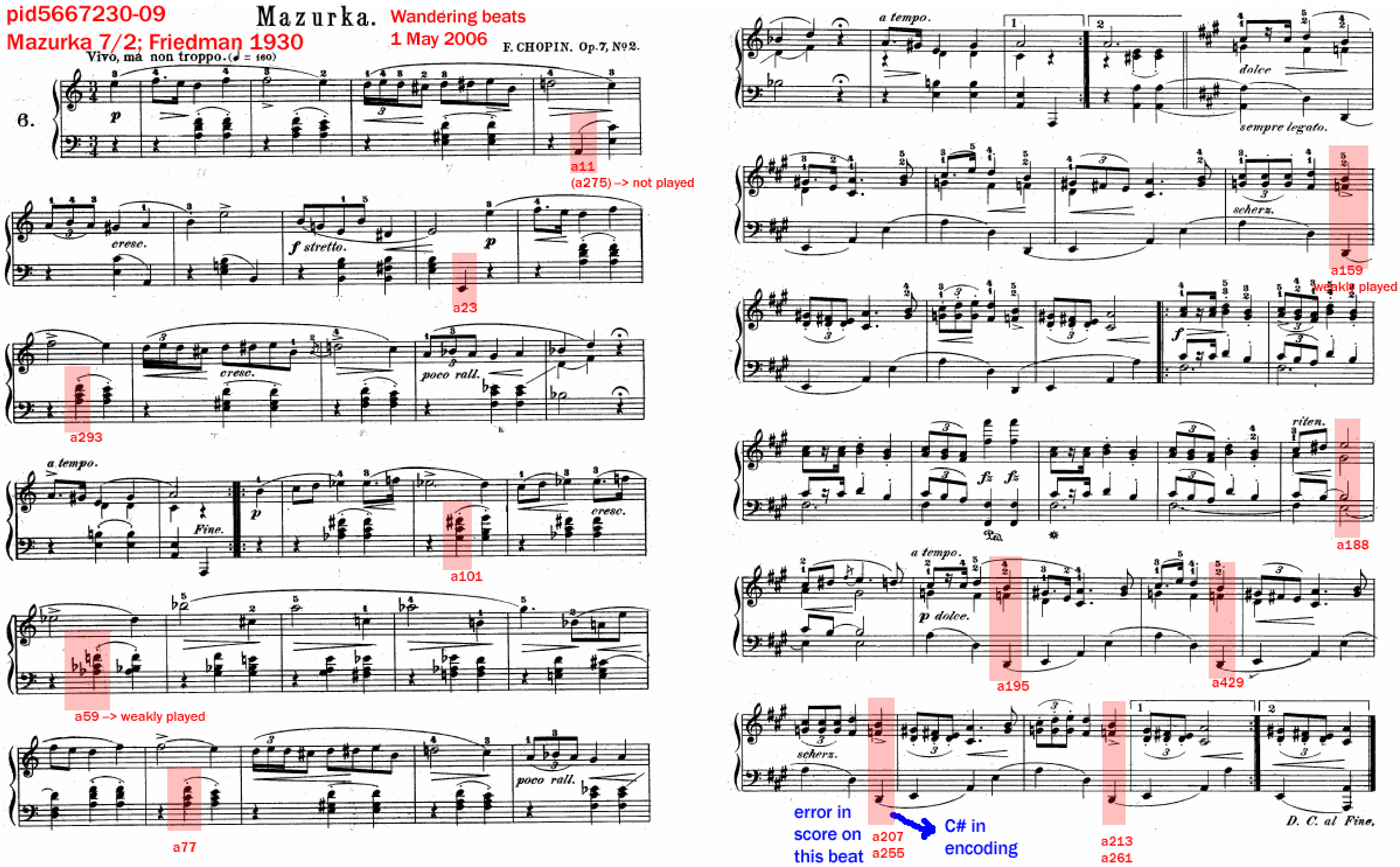

pid9048-06 fullharmonics
(iteration wandering beats removed)
corrected plot
200
100
-100
-200
50
100
150
200
250
300
350
subbeat errors
subbeat errors
worse
worse
better
red = 1st iteration
black = 2nd iteration errors
blue = wandering beats (1st iteration shown)

Wanderers (2)



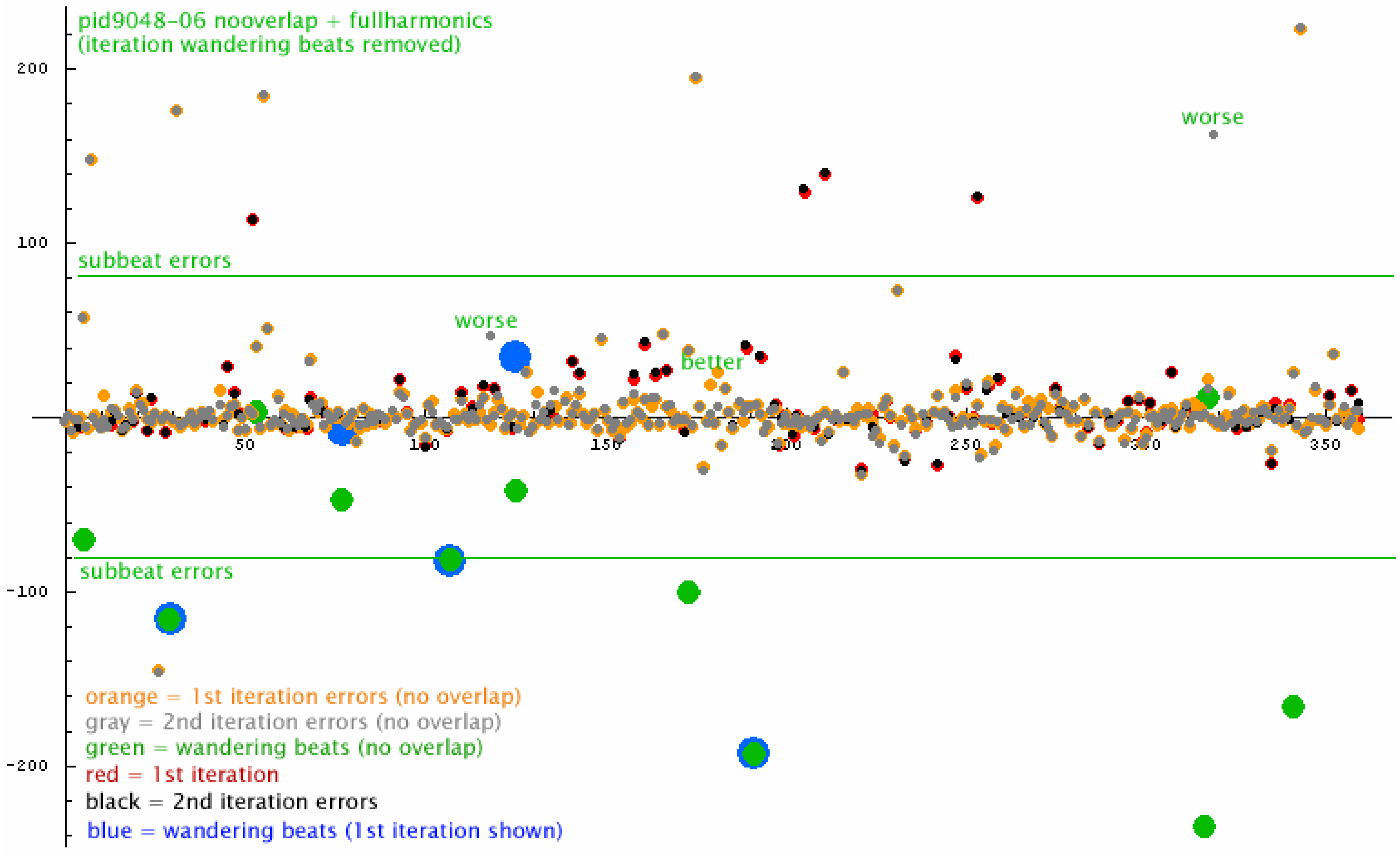
- events which the analysis method cannot "see".

Wandering (3)


pid5667230-09
Mazurka 7/2; Friedman 1930
Mazurka.
Wandering beats
1 May 2006
F. CHOPIN. Op.7, No2.
Vivo, ma non troppo.
6.
a11
(a275) –> not played
cresc.
f stretto.
a23
a293
cresc.
poco rall.
a tempo.
Fine.
a101
cresc.
a59 –> weakly played
a77
poco rall.
a tempo.
dolce
sempre legato.
scherz.
a159
weakly played
riten.
a188
a tempo.
p dolce.
a195
a429
scherz.
error in
score on
this beat
a207
a255
C# in
encoding
a213
a261
D. C. al Fine.

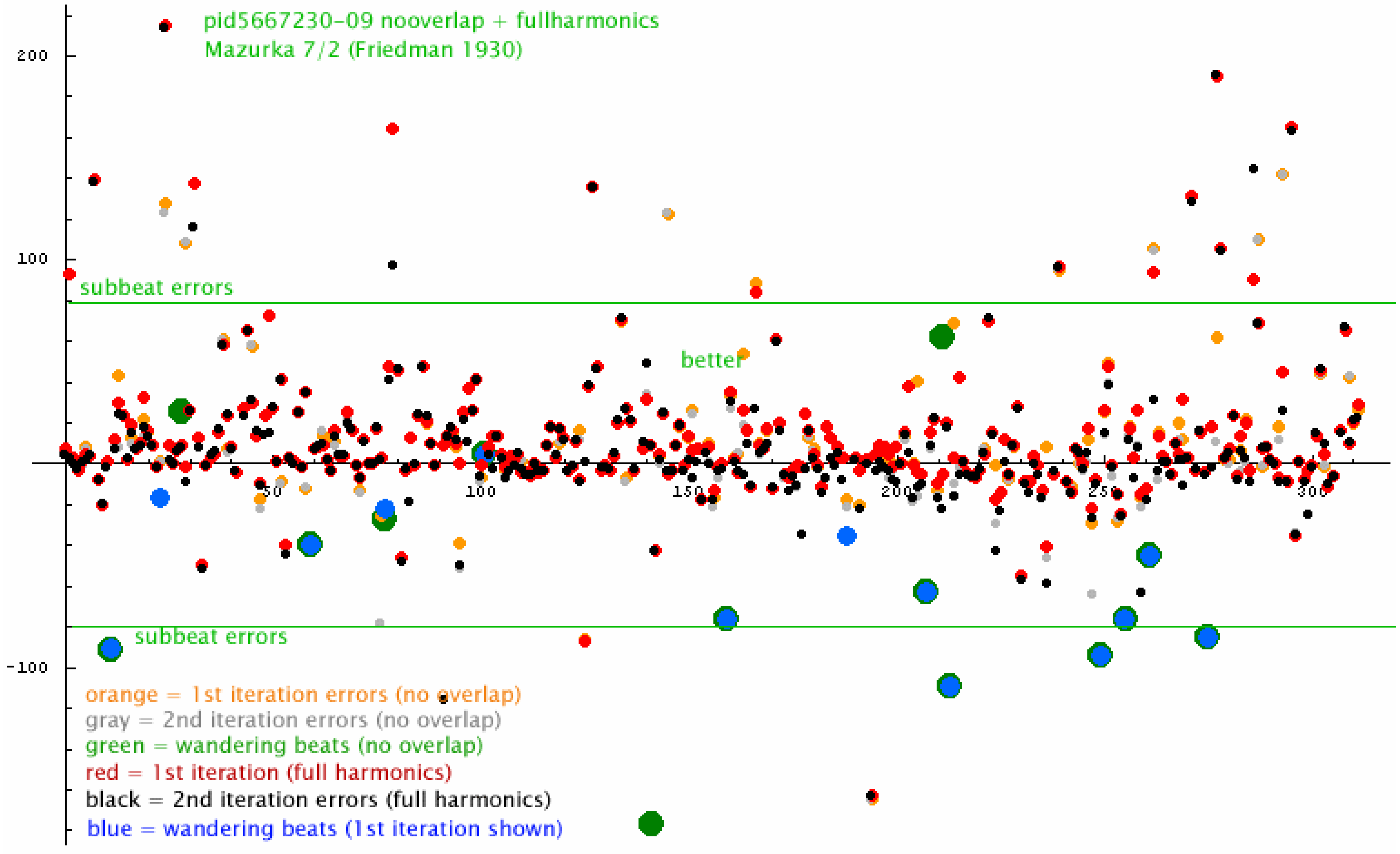
Selective Harmonics

• Remove shared harmonics with previous event to improve attack identification and remove potentially beating harmonics.



• Removing shared harmonics with previous events didn't help: more wanderers.

Selective Harmonics (2)

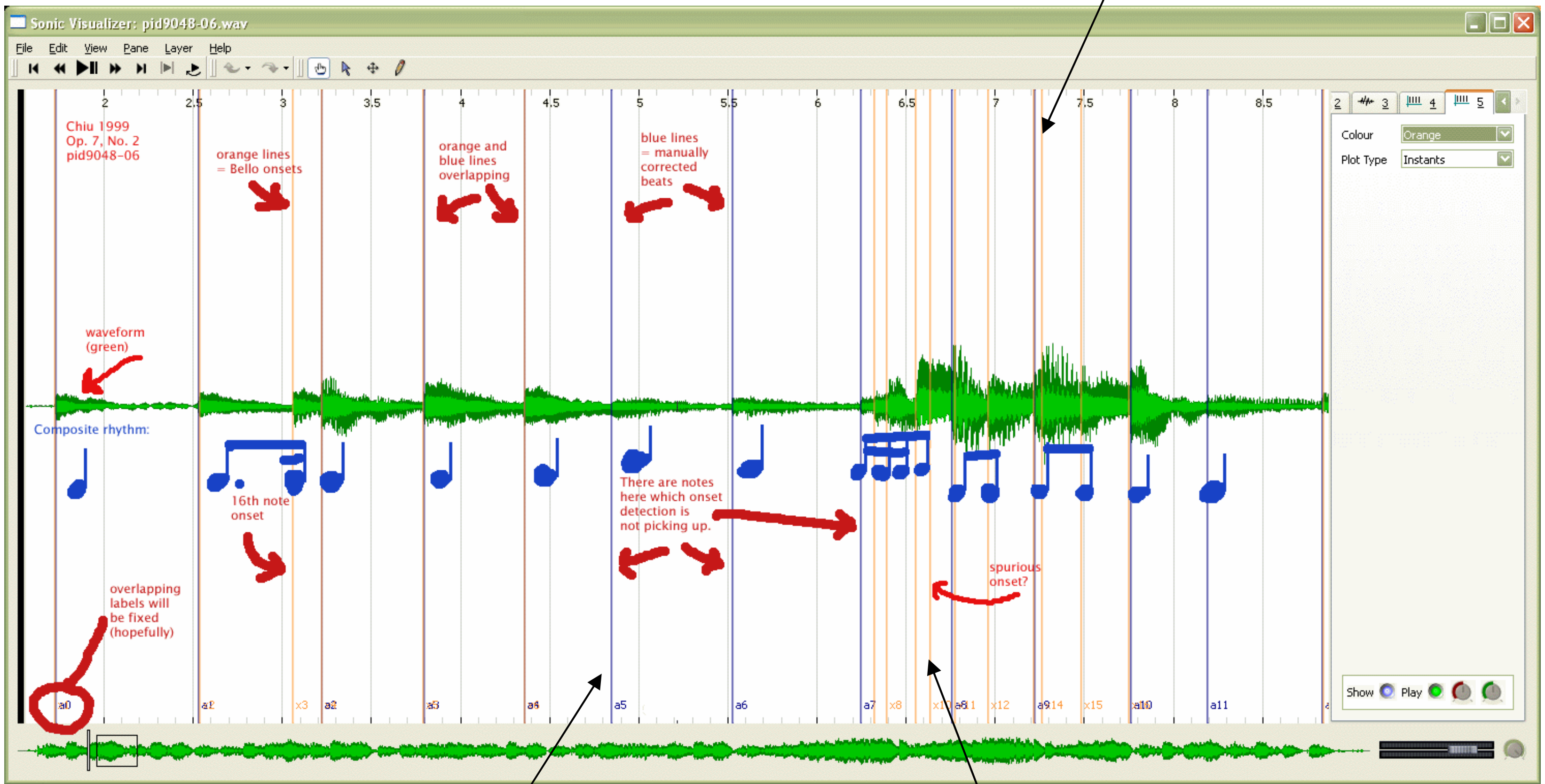


Bello Onsets

http://mazurka.org.uk/auto/onset

- Spectral measurements used to identify event onset locations
- Can give false positivies and false negatives
- Does not utilize a score

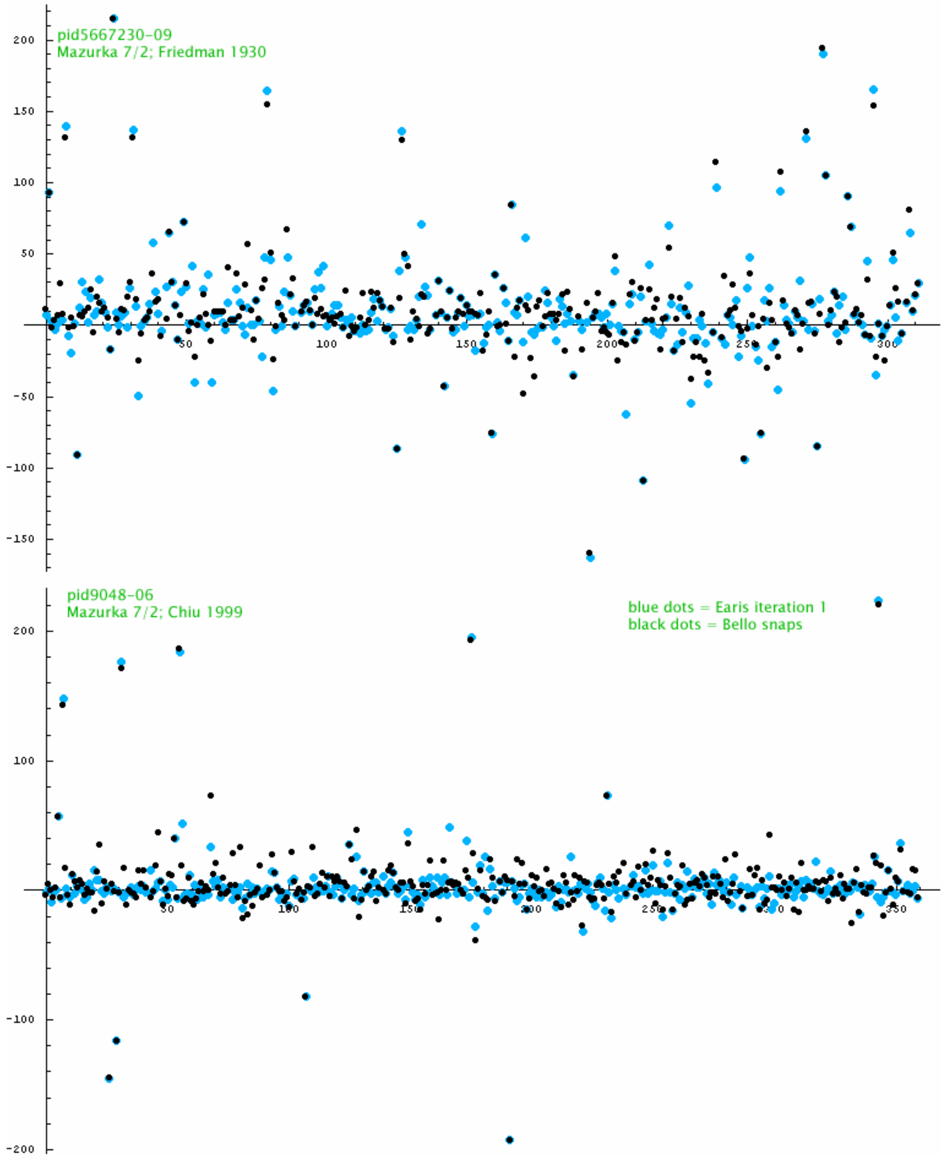
LH/RH?



false negative

false positive

Onset Snapping

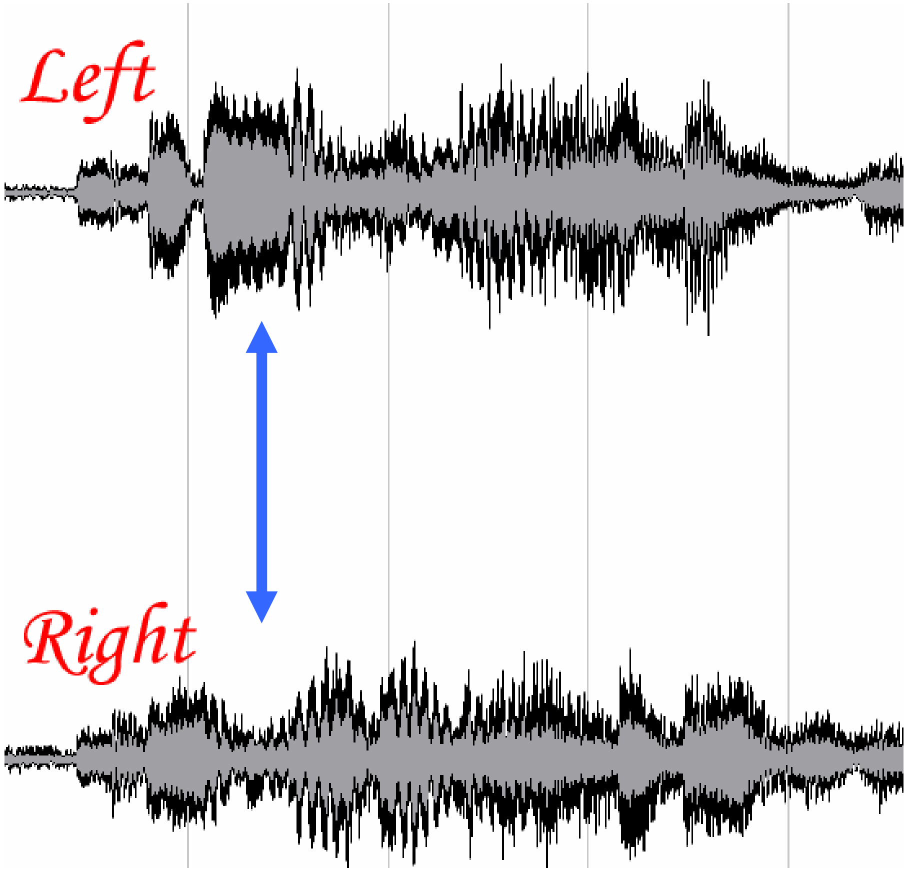


- Snap earis to bello if a bello onset is less than 50 ms away.

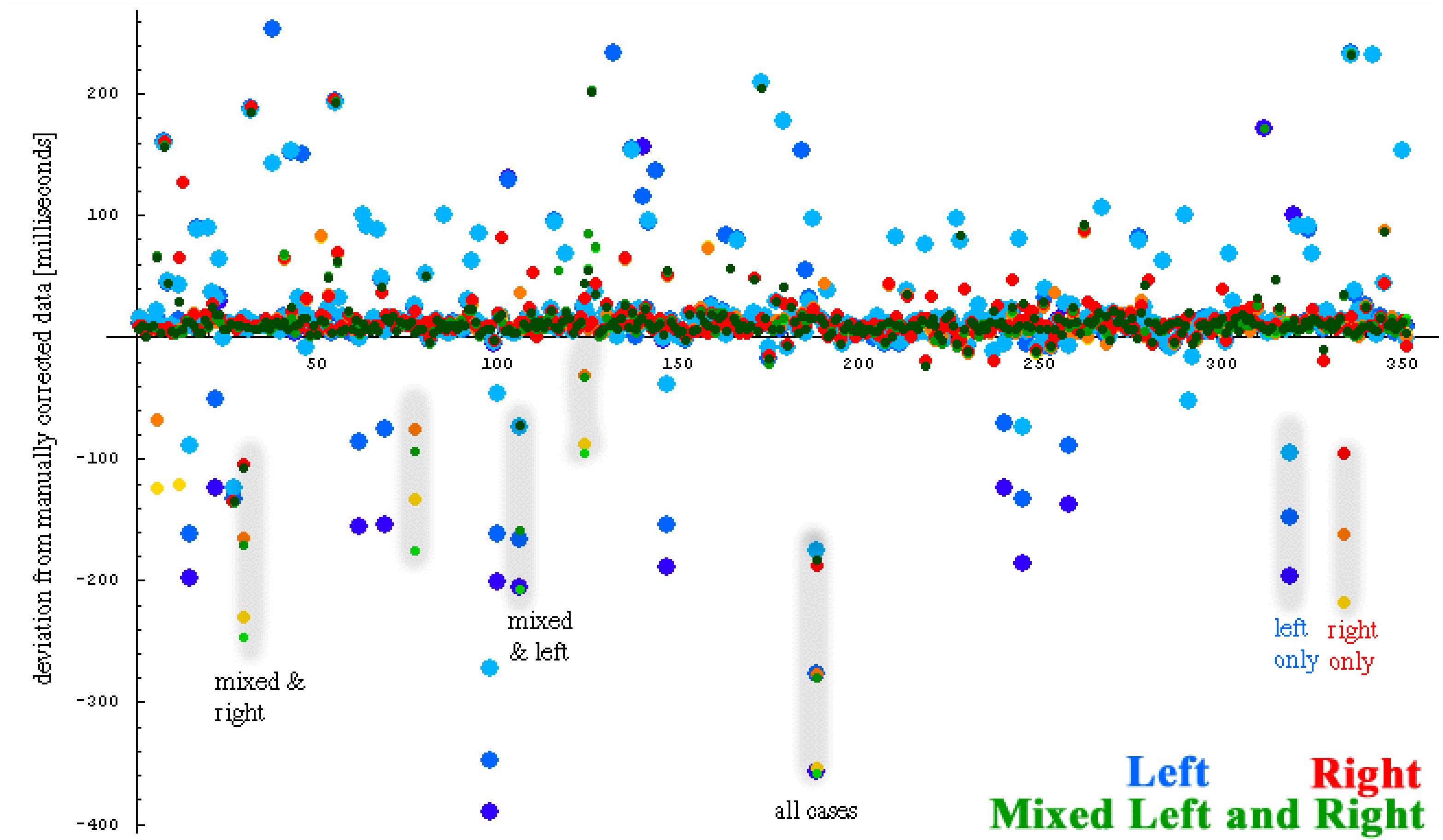
Friedman 1930, Mazurka 7/2:
iteration 1 mean deviation: 22.0 ms
bello snapping md: 22.1 ms

Chiu 1999, Mazurka 7/2:
iteration 1 mean deviation: 10.5 ms
bello snapping md: 13.4 ms

Stereo Differences

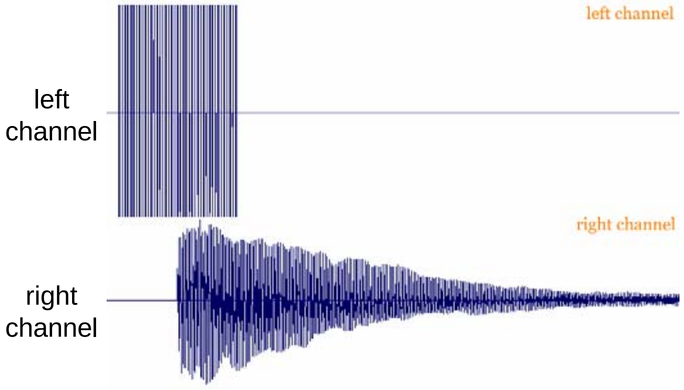


Stereo Comparison

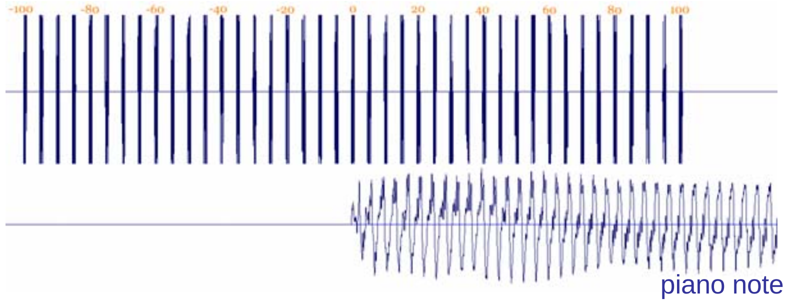


Experiments

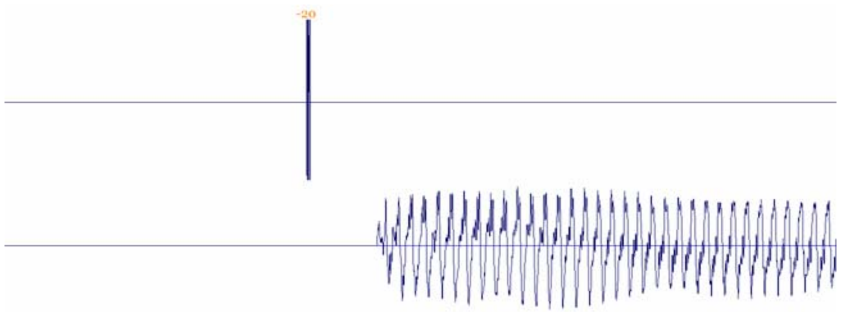
Note Onset Time Resolution



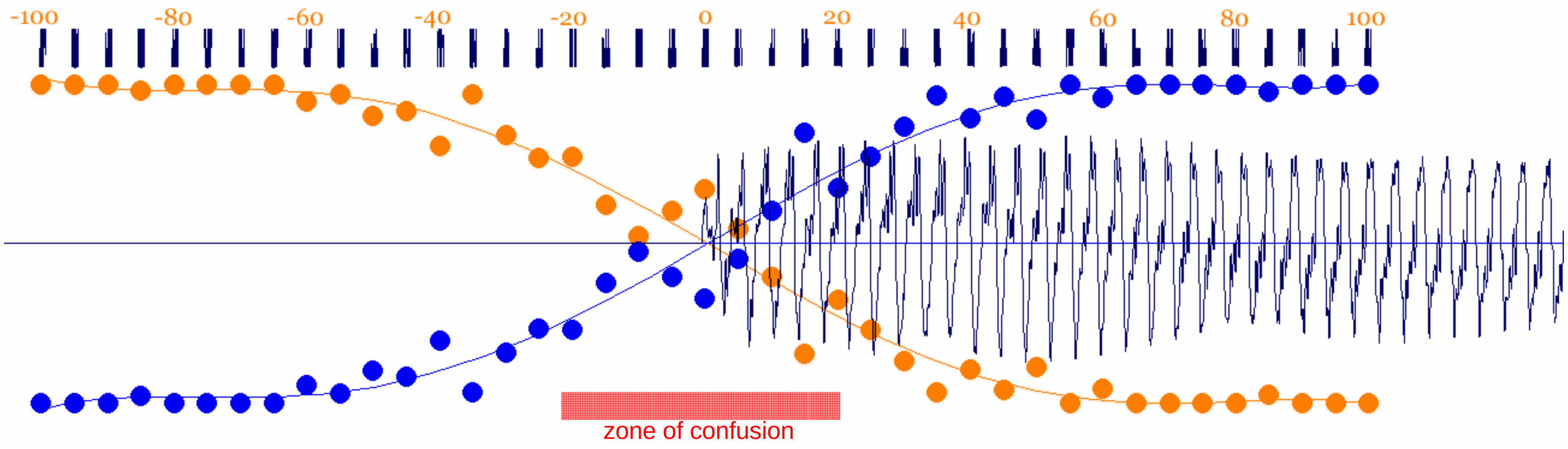
zoom in



- Play one of these clicks and ask listener: did it come before or after the start of the piano note?

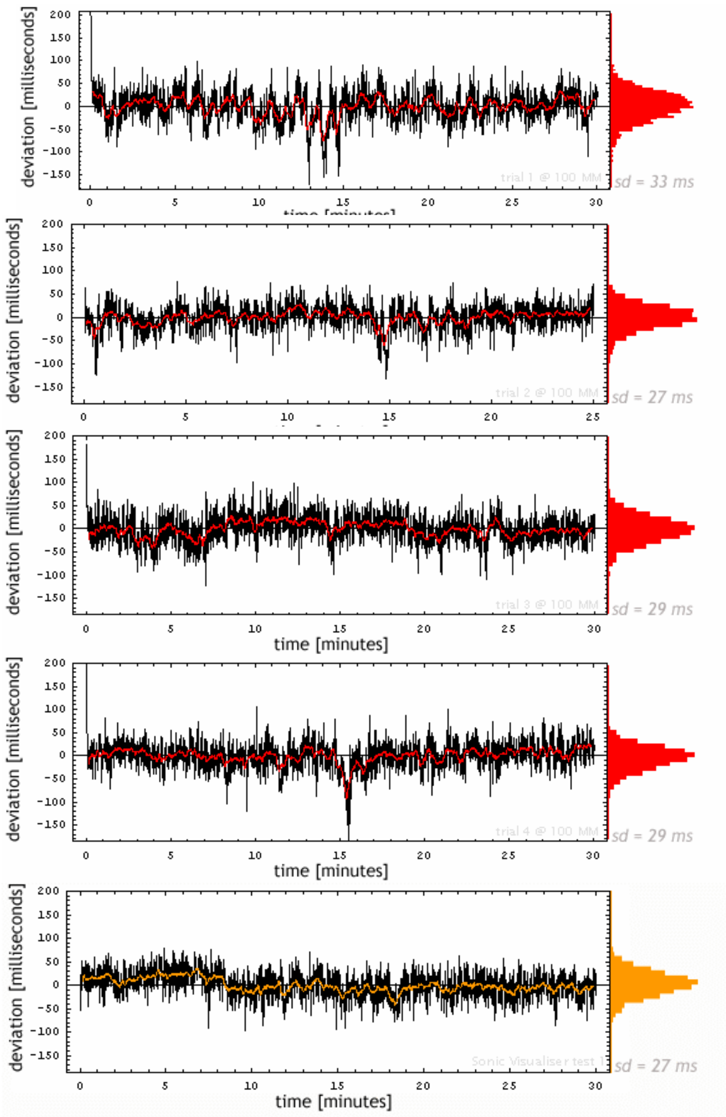


Note Onset Time Resolution



- 75% accuracy or better outside -21.2 to +22.0 ms range around note attack.
- Symmetric about the note onset.
- Very accurate to distinguish which came first when difference is > 60 ms.

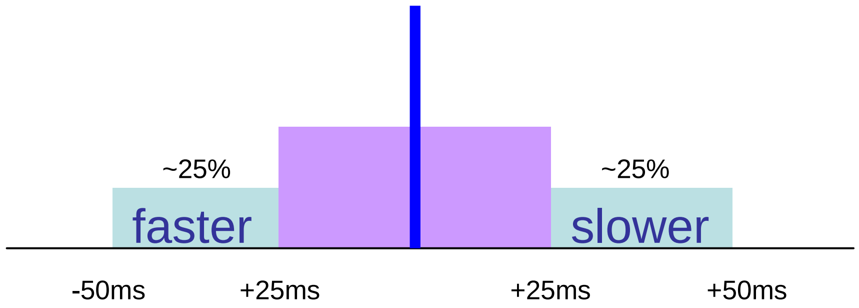
Tapping Accuracy



• Tap for 30 minutes to a constant tempo

• 50% of taps occur within +/- 25 ms of actual event

• 95% of taps occur within +/- 50 ms of actual event

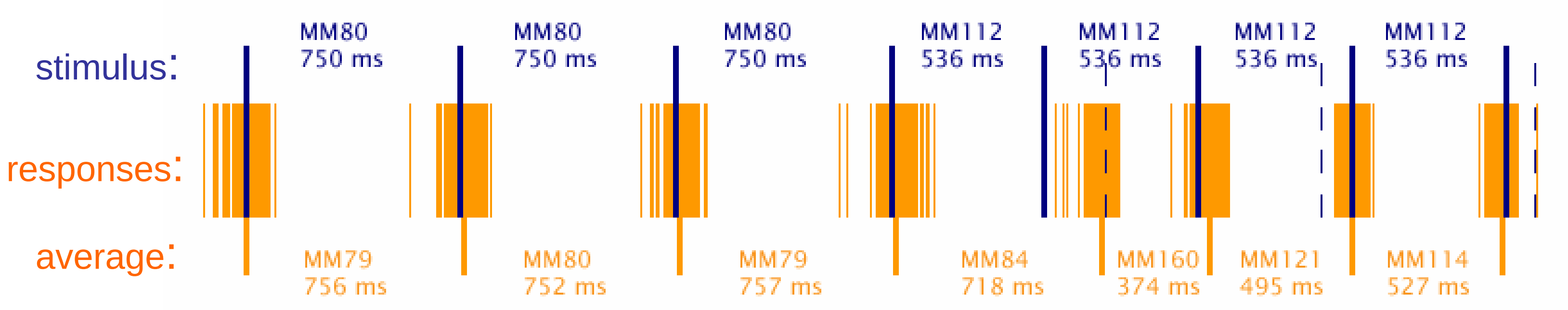


• For Mazurkas (significant tempo changes), accuracy is about twice as much (50% occur within 50 ms of actual event).

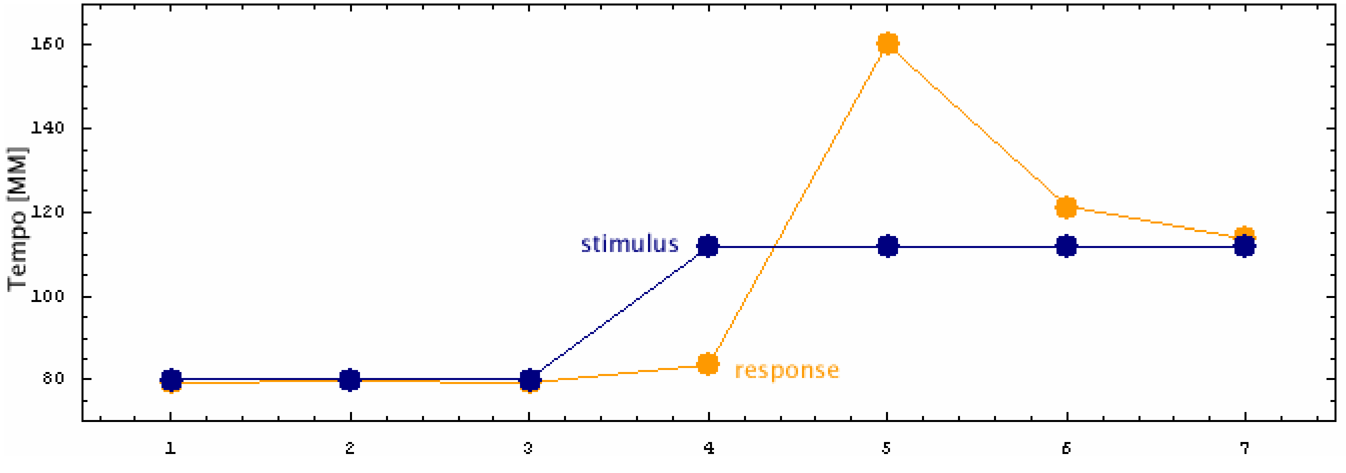
Unpredictable Tempo Changes

- **Tapping to an unknown sudden change in tempo**

- Suddenly faster:



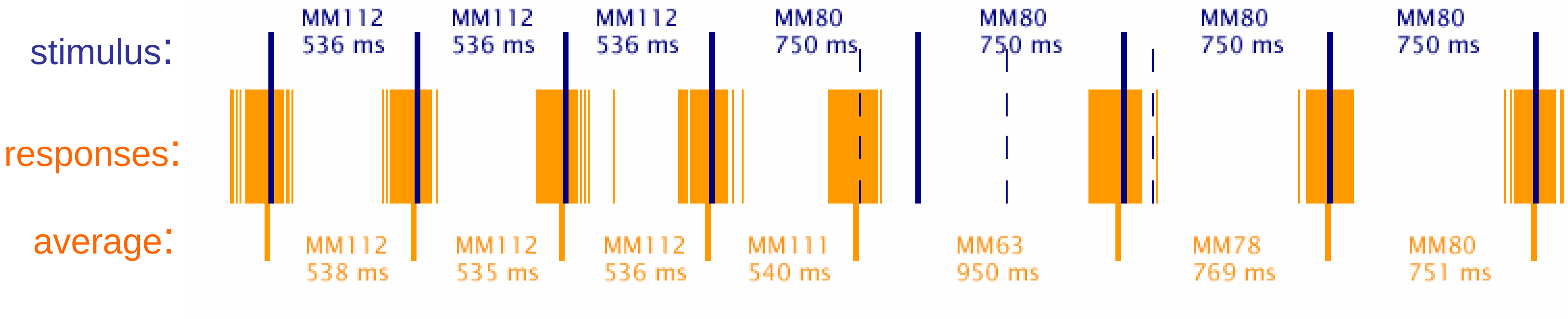
Same data as a tempo plot:



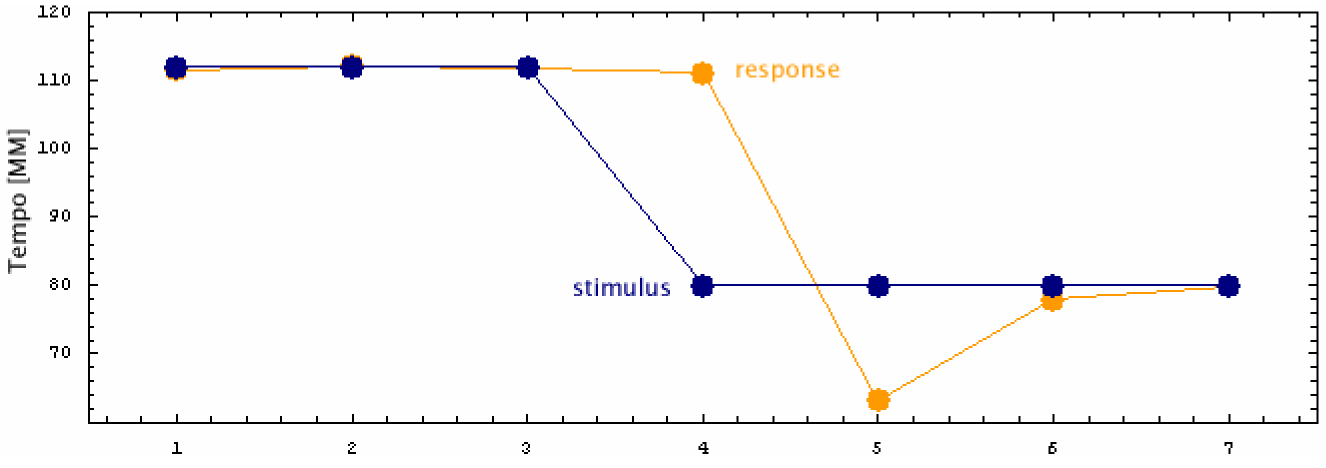
Unpredictable Tempo Changes (2)

- **Tapping to an unknown sudden change in tempo**

- Suddenly slower:

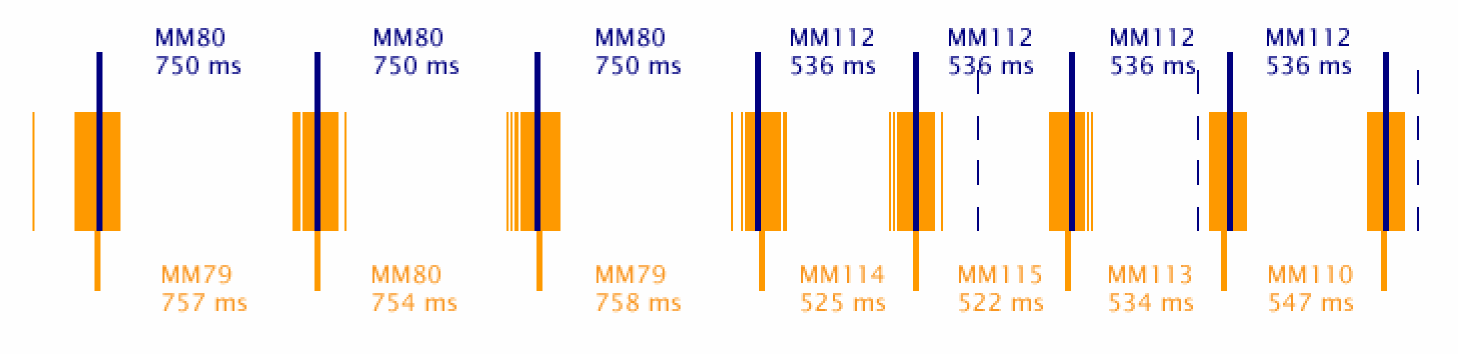


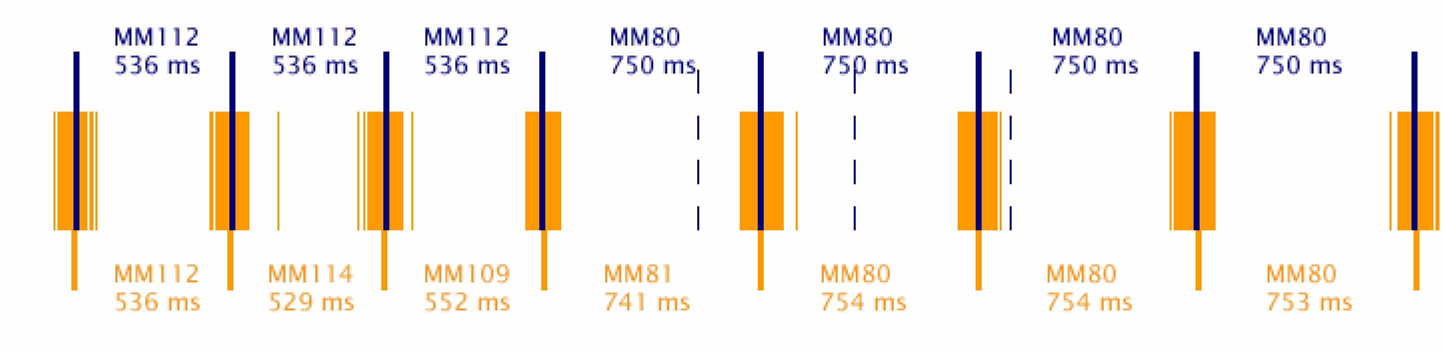
Same data as a tempo plot:



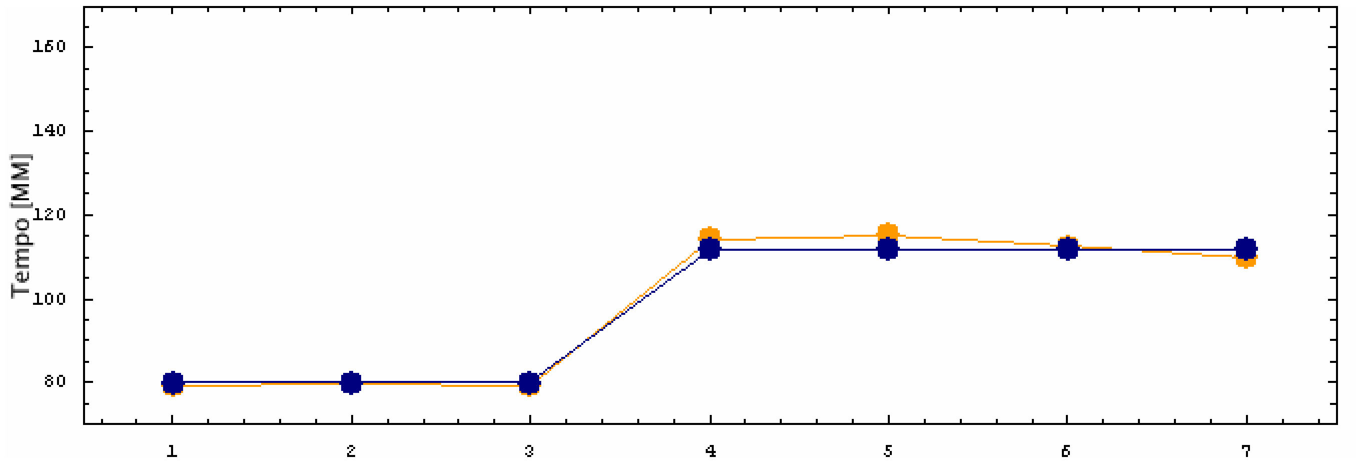
Predictable Tempo Changes

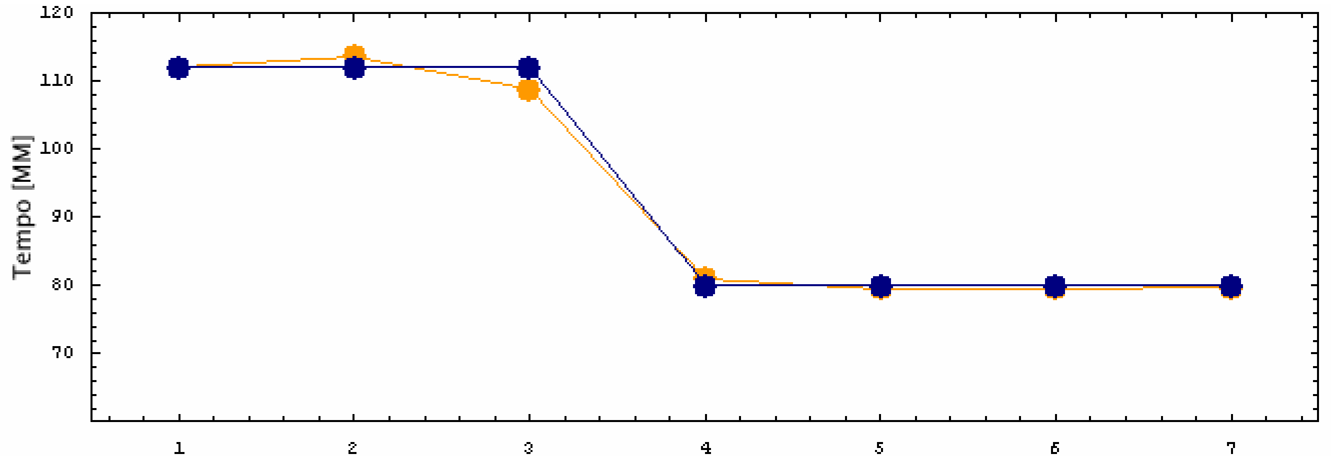
• Tapping to an **known** sudden change in tempo:





Predictable Tempo Changes





Tempo JND

- How little can the tempo change before it is noticed?

clicks:	3	4	5	10	20
JND:	0.0265	0.0115	0.0082	0.0040	0.0024
faster end delta:	-26.5	-34.4	-48.9	-142.7	-404.9
slower end delta:	26.5	34.6	49.5	145.3	416.0
faster end tempo:	63.3	62.1	62.0	62.2	62.8
slower end tempo:	56.9	57.0	58.0	57.9	57.3

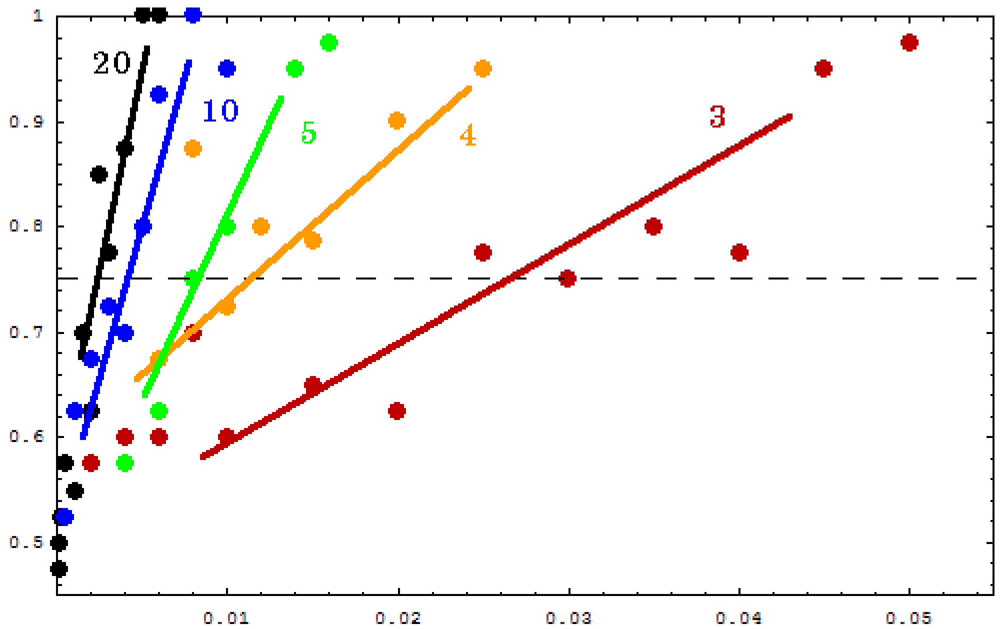
3: | 60.0 | 63.3 |

4: | 60.0 | 61.05 | 62.1 |

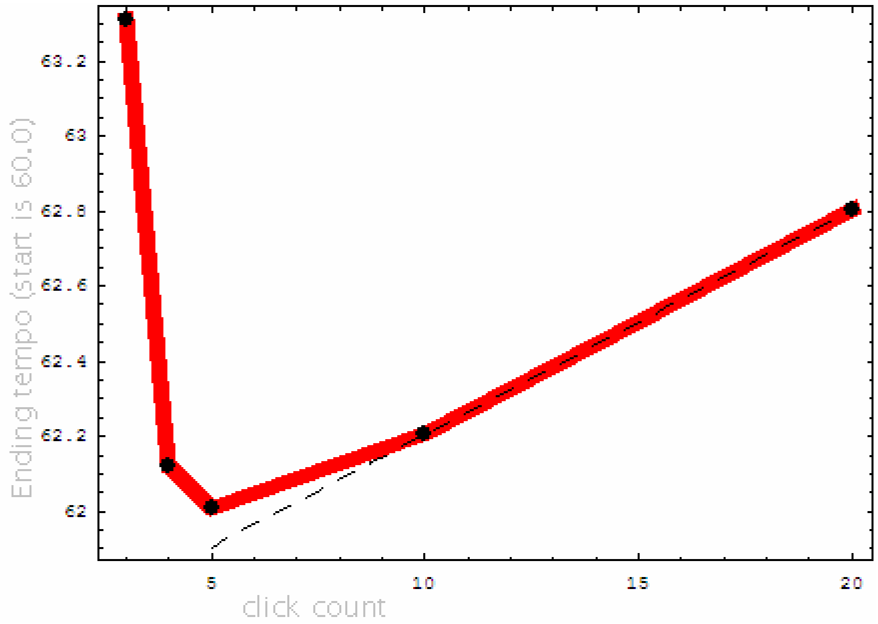
5: | 60.0 | 60.67 | 61.33 | 62.0 |

10: | 60.0 | 60.28 | 60.55 | 60.83 | 61.10 | 61.38 | 61.65 | 61.93 | 62.2 |

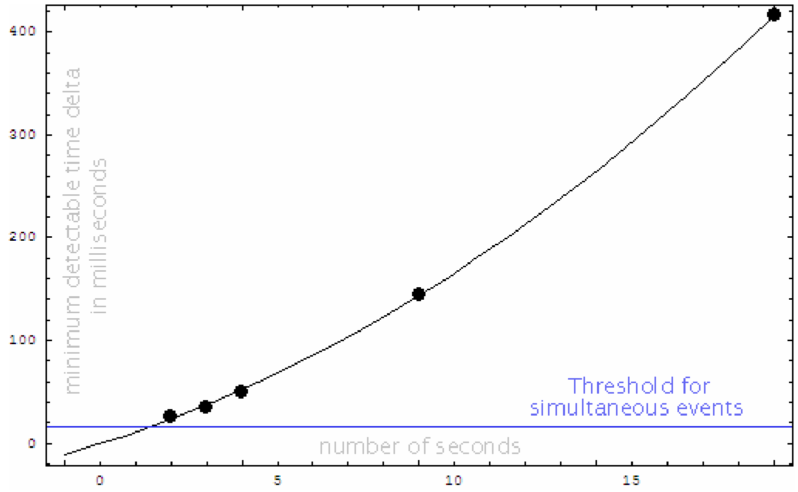
Tempo JND (2)



1
0.9
0.8
0.7
0.6
0.5
20
10
5
4
3
0.01
0.02
0.03
0.04
0.05



Ending tempo (start is 60.0)
63.2
63
62.8
62.6
62.4
62.2
62
5
10
15
20
click count



minimum detectable time delta
in milliseconds
400
300
200
100
0
Threshold for
simultaneous events
number of seconds
0
5
10
15

Performance Simulations

Performance Feature Layers

http://mazurka.org.uk/ana/midi

Tempo:		*Type 0*	*Type 1*
constant →	**Constant Tempo Performance Simulations**		
	• *Plain simulated performance* -- These MIDI files contain a constant tempo throughout the performance, and are basically literal performances of the written score. However, the performance tempo is set from the average tempo of the model performance. Thus, the duration of the MIDI files will match the duration of the original performance.	plainsd0 MM 102	plainsd MM 102
beat →	**Beat-Level Tempo Performance Simulations**		
	• *Beat-level tempo* -- These MIDI files contain a tempo which changes once per beat based on the duration between beats in the originating performance. Sub-beat level note times are estimated from a constant tempo throughout the beat. Score dynamics are used.	corbsd0	corbsd PerfViz match file
	• *Average tapped performance* -- These MIDI files contain a tempo which changes once per beat throughout the performance based on an average of several reverse conductions of the performance. Score dynamics are used.	avgtapsd0	avgtapsd PerfViz match file
event →	**Subbeat-Level Tempo Performance Simulations**		
notes →	**Note-Level Tempo Performance Simulations**		

Performance Components

Tempo/Timing

1. Average tempo (of entire piece)
2. Beat-to-beat tempo
3. Sub-beat timings (continuous tempo)
4. Non-simultaneous events (LH/RH, arpgggios)

Dynamics

1. Score dynamics
2. Composite loudness
3. LH/RH loudness
4. Individual note loudness

PerfViz

- 3D performance worm visualiser by Martin Gasser (Vienna)

MIDI file + **Match file**

```
info(matchFileVersion,2.0).
info(scoreFileName,'STDIN').
info(midiFileName,'pid9048-06-corbsdpv.mid').
info(midiClockUnits,480).
info(midiClockRate,500000).
info(keySignature,[an,minor]).
info(timeSignature,3/4).
info(approximateTempo,102.4).
snote(n1,[e,n],5,0:3,0,1,0,1,[])-note(1,[e,n],5,1656,2428,2428,43).
snote(n2,[f,n],5,1:1,0,3/16,1,1.75,[])-note(2,[f,n],5,2428,2925,2925,49).
snote(n3,[e,n],5,1:1,3/16,1/16,1.75,2,[])-note(3,[e,n],5,2925,3090,3090,41).
snote(n4,[a,n],3,1:2,0,1,2,3,[])-note(4,[a,n],3,3090,3366,3366,40).
snote(n5,[d,n],4,1:2,0,1,2,3,[])-note(5,[d,n],4,3090,3366,3366,40).
snote(n6,[f,n],4,1:2,0,1,2,3,[])-note(6,[f,n],4,3090,3366,3366,40).
snote(n7,[d,n],5,1:2,0,1,2,3,[])-note(7,[d,n],5,3090,3642,3642,40).
snote(n8,[a,n],3,1:3,0,1,3,4,[])-note(8,[a,n],3,3642,3912,3912,43).
snote(n9,[d,n],4,1:3,0,1,3,4,[])-note(9,[d,n],4,3642,3912,3912,43).
snote(n10,[f,n],4,1:3,0,1,3,4,[])-note(10,[f,n],4,3642,3912,3912,43).
snote(n11,[f,n],5,1:3,0,1,3,4,[])-note(11,[f,n],5,3642,4181,4181,39).
snote(n12,[f,n],5,2:1,0,2,4,6,[])-note(12,[f,n],5,4181,5301,5301,62).
snote(n13,[a,n],3,2:2,0,1,5,6,[])-note(13,[a,n],3,4649,4975,4975,39).
snote(n14,[c,n],4,2:2,0,1,5,6,[])-note(14,[c,n],4,4649,4975,4975,39).
```

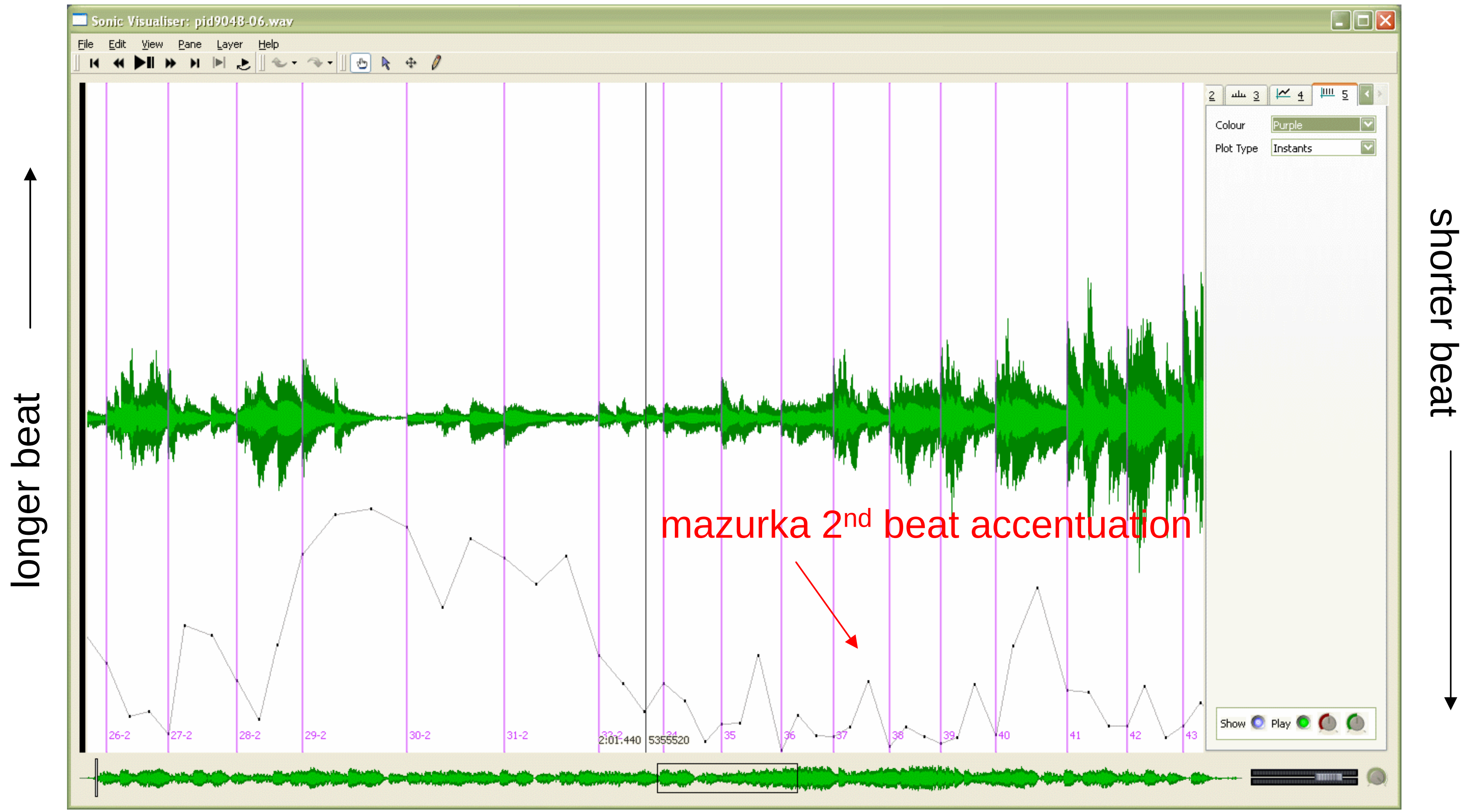
PerfViz (2)

3 Axes:

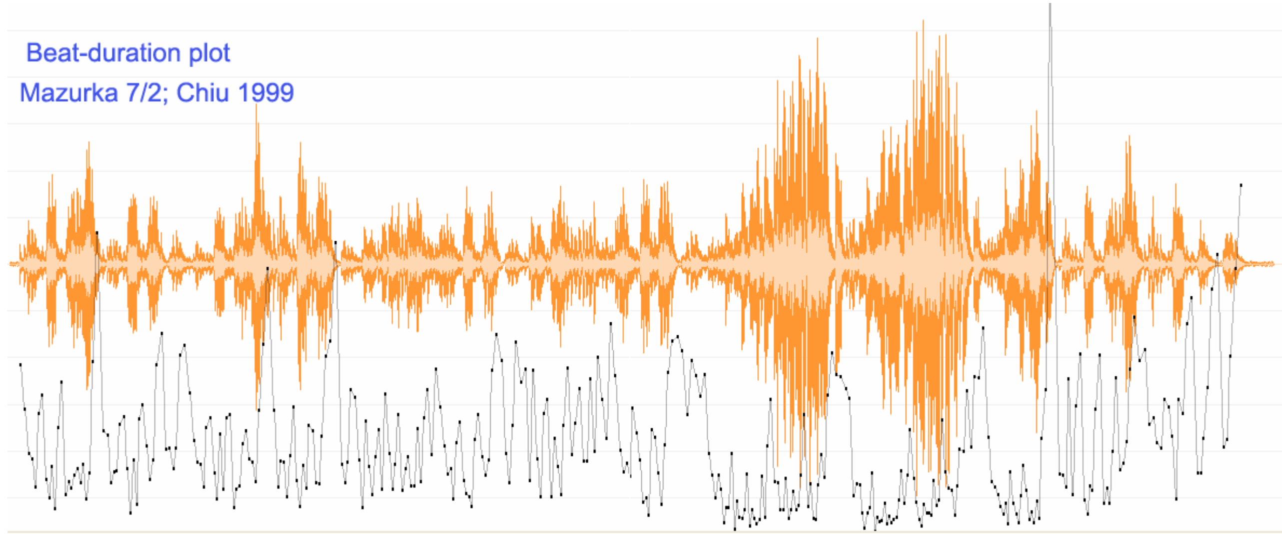
1. Time
2. Tempo
3. Loudness

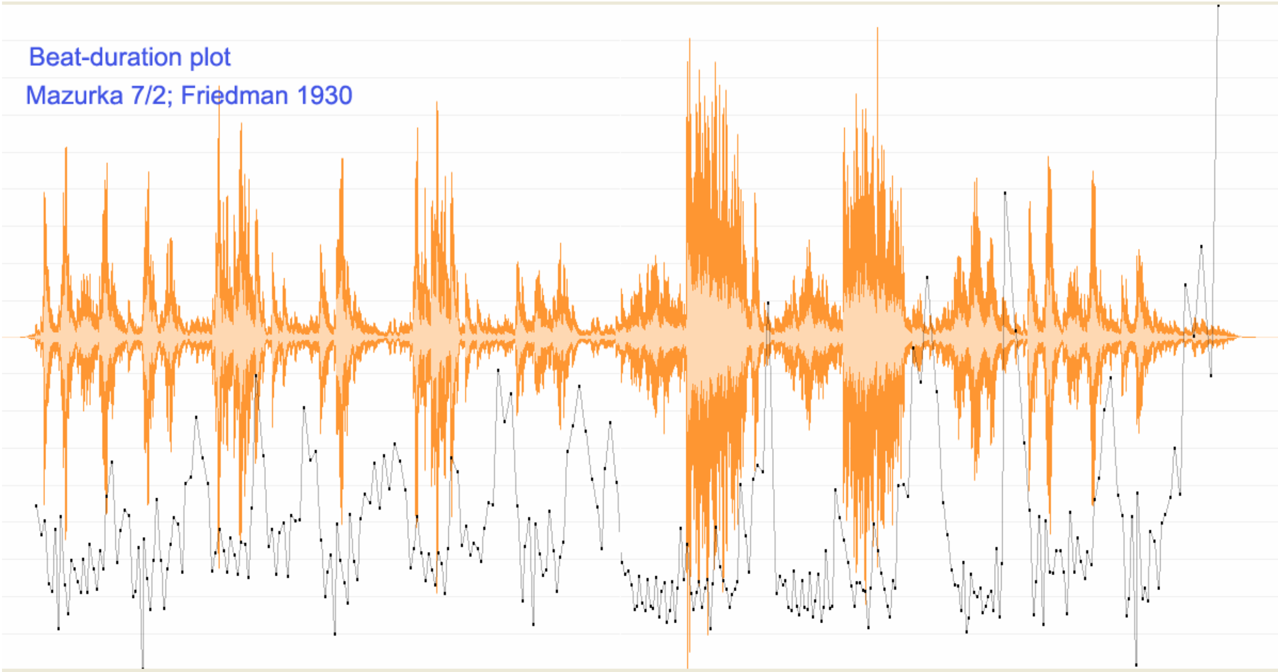
Initial Analysis

Beat Durations



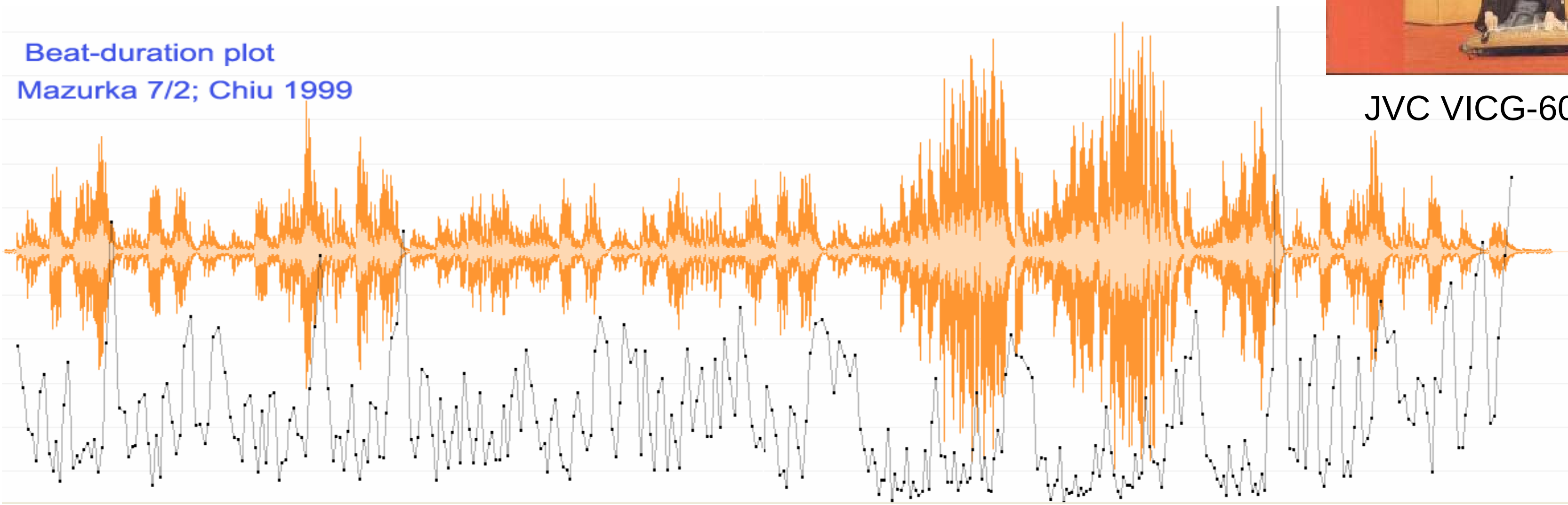
Beat Durations (2)





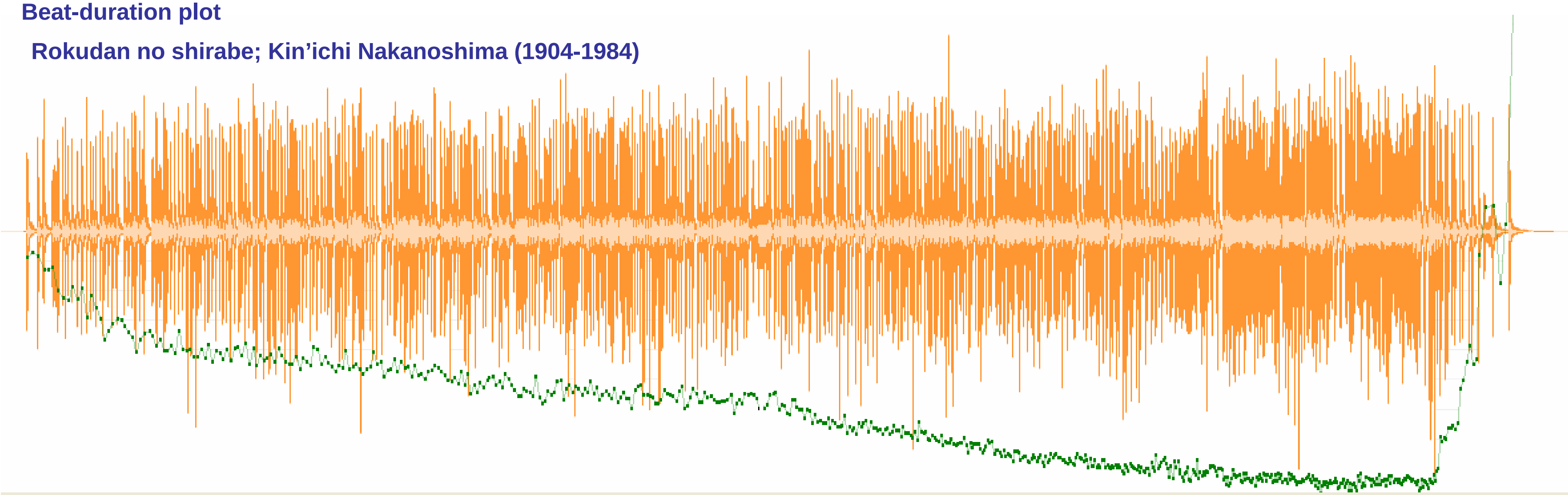
Tempo and Style

JVC VICG-60397

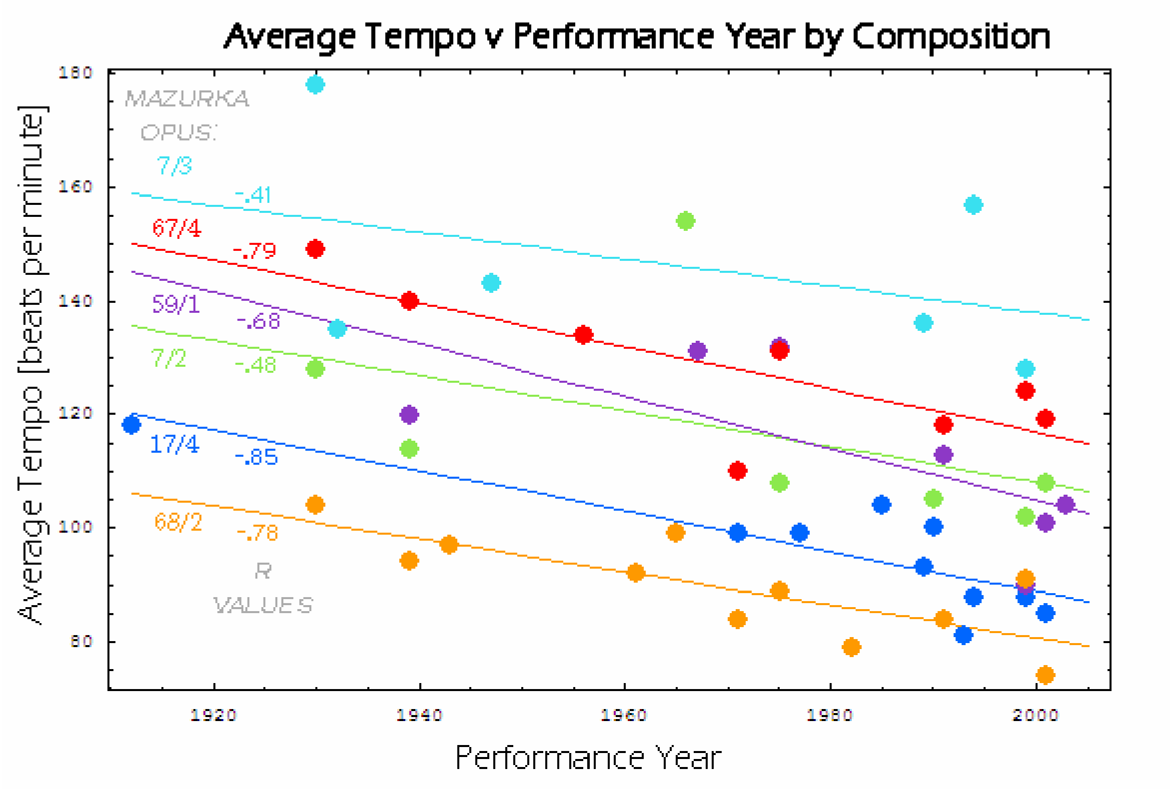


Beat-duration plot

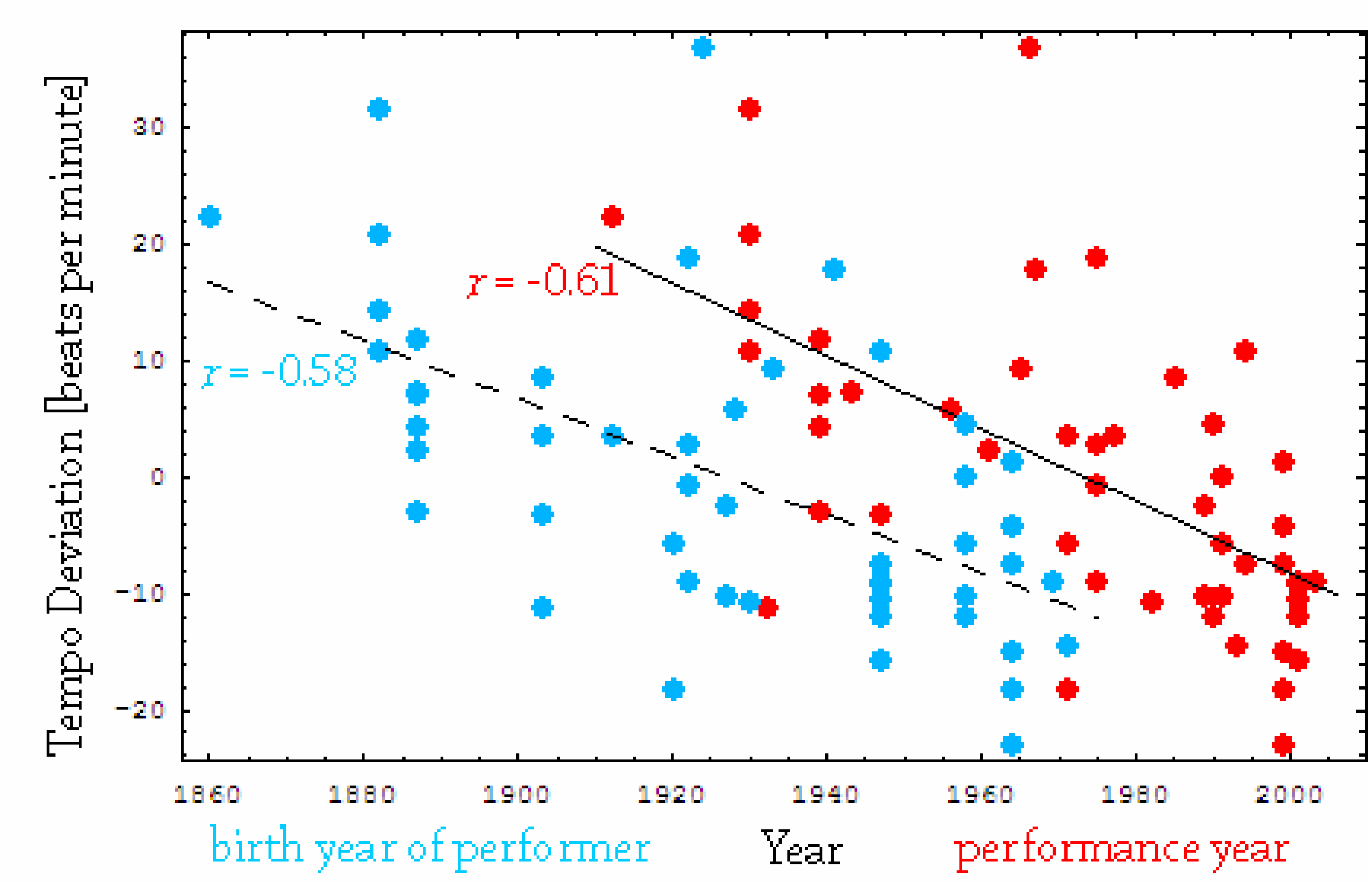
Rokudan no shirabe; Kin'ichi Nakanoshima (1904-1984)



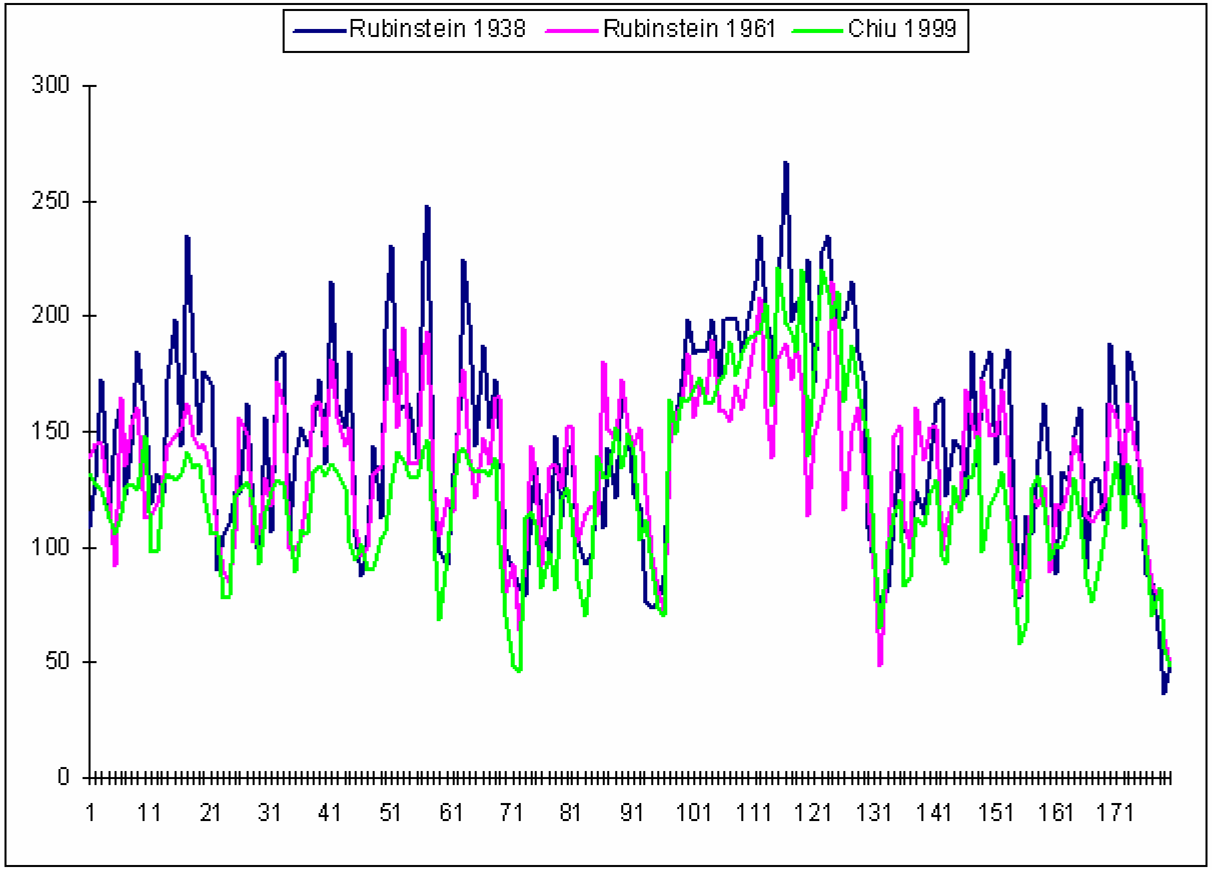
Average tempo over time



Average tempo over time (2)



Mazurka in F Major, Op. 68, No. 3



Rubinstein 1938

MM134

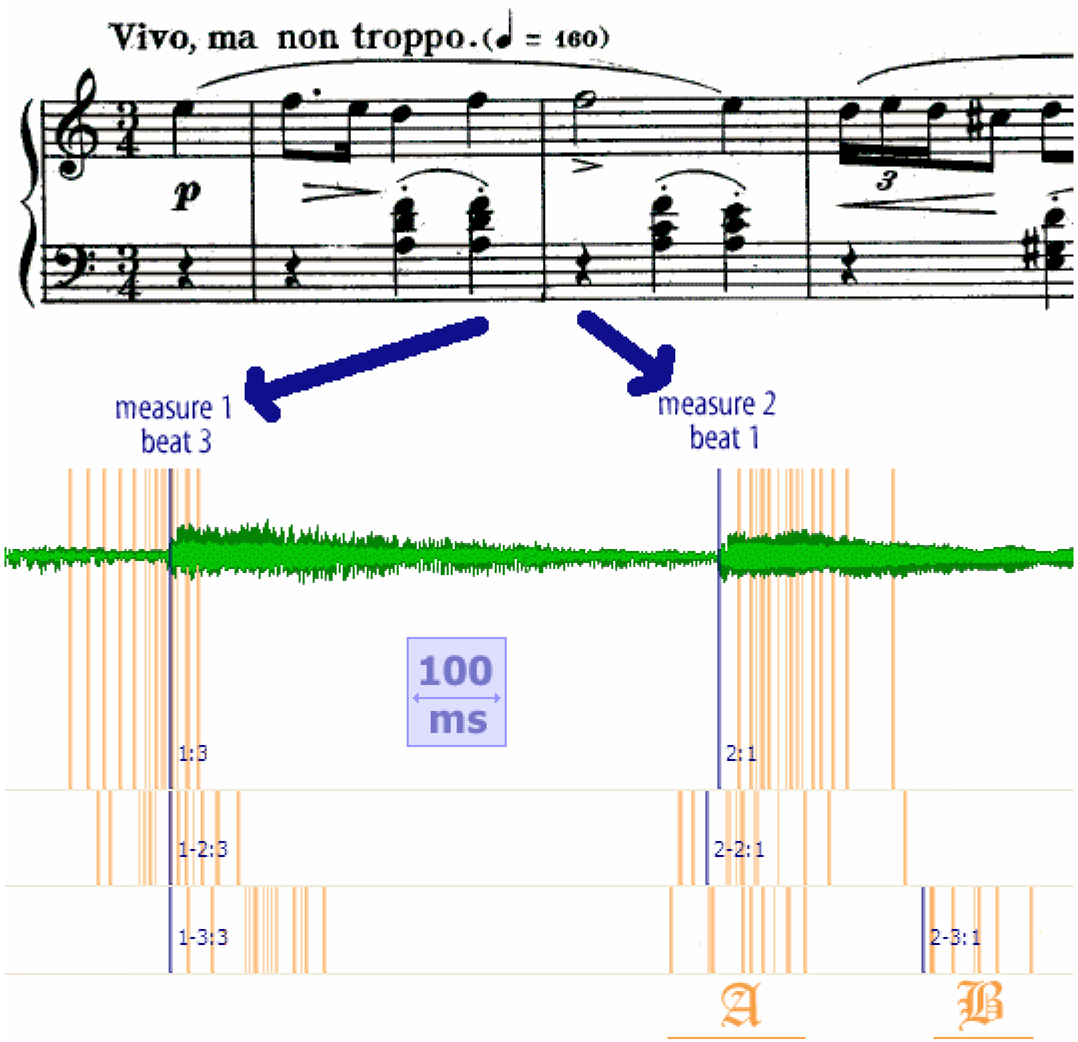
Rubinstein 1961

MM129

Chiu 1999

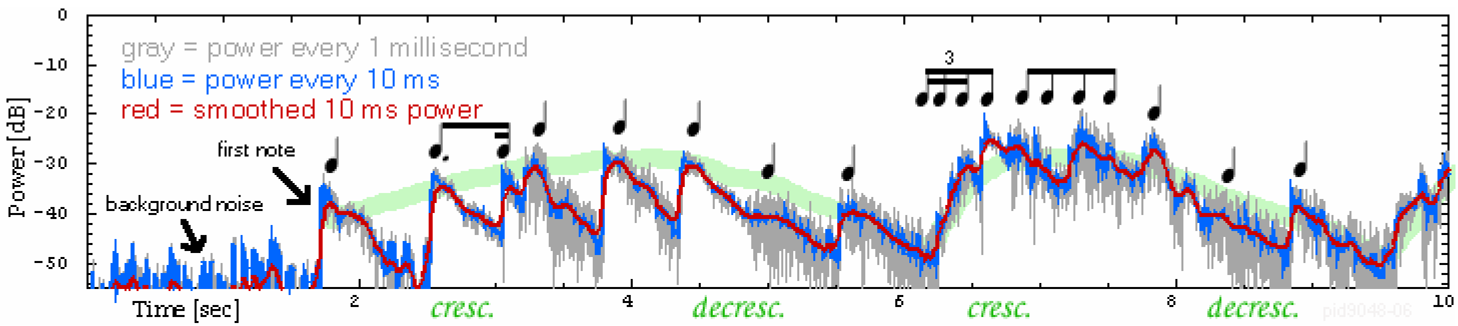
MM115

Repeats



Vivo, ma non troppo. (♩ = 160)
measure 1
beat 3
measure 2
beat 1
100
ms
1:3
2:1
1-2:3
2-2:1
1-3:3
2-3:1
A
B

Dynamics



gray = power every 1 millisecond
blue = power every 10 ms
red = smoothed 10 ms power
first note
background noise
Power [dB]
0
-10
-20
-30
-40
-50
Time [sec]
2
4
6
8
10
cresc.
decresc.
cresc.
decresc.
3