

Reverse Conducting Evaluation

Craig Stuart Sapp <craig@ccrma.stanford.edu>

15 Nov 2005

This is a generic *Mathematica* notebook used for examining the individual reverse conducting performances of Chopin Mazurkas compared to carefully corrected versions of the average beat tappings for the performance.

Mazurka in A Minor, Op. 7, No. 2 Chiu 1999

■ Loading the raw data

First load the absolute beat positions for the average beat positions and the corrected beat positions:

```
performance = "pid9048-06";
pvar = StringReplace[performance, "-" -> "x"];

Get[performance <> ".macorrected", Path -> "D:\mazurkas\eval"]
mbeats = ToExpression[pvar <> "corrected"];

Get[performance <> ".maaverage", Path -> "D:\mazurkas\eval"]
mabeats = ToExpression[pvar <> "average"];
```

Now measure the time difference between each average and corrected beat. A negative value means that the average beat occurs before the corrected beat time.

```
mdiffs = Transpose[mabeats][[2]] - Transpose[mbeats][[2]]

{187, 28, -45, -19, 76, 70, 20, 43, 47, 0, 55, 117, 103, -38, -5, -20, 11, 47, 73, 75, 53, 60, 84, 0,
-128, 15, -67, 22, 29, -25, 15, 75, 14, 50, -53, 32, 29, 62, -51, -25, -32, -78, 0, 204, 52, 14,
13, -41, 146, -20, -5, 0, 47, 53, 64, 100, 47, 46, -54, 76, 60, 49, 38, 32, 52, -38, -10, 31,
40, 35, -39, -36, 16, 58, 79, 6, 31, 79, 107, -21, -3, 33, -45, 5, 8, 27, 0, -17, -86, -109, 0,
-246, 47, 65, 61, 33, 118, 45, 9, -108, 18, 45, 105, 35, -7, -152, 8, 26, 83, -25, 38, 56, 27,
20, 59, -72, 36, -67, -105, 46, -45, 13, -33, 32, 38, 62, 62, 68, 21, 52, 50, 81, 42, 26, -44,
-14, 23, -108, 0, 64, 49, 53, -47, -89, 144, -133, -14, -183, 15, 41, -20, -14, 2, -16, 46, 47,
-23, -69, -20, -41, -77, 0, 66, -89, -6, -162, 24, 27, -97, 0, -9, 31, 70, 45, 0, 105, -7, 46,
40, 113, 83, 119, 0, 22, -32, -47, 0, -25, -218, -114, -102, -38, -8, -31, 27, -23, 20, -27, 0,
39, 0, 27, -98, 21, 0, 11, 0, 49, -36, 8, 31, 24, 29, 70, -37, 15, -66, -64, 43, 0, 78, 14, -25,
84, 64, 0, 72, -10, -13, 0, 92, 5, 80, 21, -8, -30, 67, 0, -157, 47, 110, 75, -85, 30, 7, 9, 69,
78, 52, 44, -16, 60, 58, 42, 22, 6, 88, 25, 46, 26, 15, 23, -7, 56, 74, 61, 29, 58, 96, 42, 50,
68, 54, 82, -27, 0, 54, -14, 0, 8, -16, 17, 0, -54, 62, 0, 77, 13, 0, -93, 11, 24, 20, 14, 37,
16, -4, 80, -37, -3, 54, 33, 0, -58, 0, 32, 54, 47, 0, 79, -49, -21, -121, 188, 78, 97, -85, 95,
-23, -83, -3, 35, 105, 81, 75, -60, 12, 13, 82, 9, -106, 126, 61, -58, -52, 0, 176, -69, -33,
-113, 65, 0, 40, 90, 0, 89, -134, -100, -92, -135, 0, 0, 0, 97, 0, -181, 73, 39, -69, -74, 0}
```

The average correction to each beat from the average tap times of all trials is typically around 48 milliseconds. This means that, on the average, each average tap beat position must be moved 48 milliseconds to align with the correct beat time.

```
<< Statistics`ConfidenceIntervals`
```

```
Mean[Abs[mdiffs]] // N
```

```
48.5568
```

■ Manual correction of average reverse conducting beat positions

This section displays the difference between the average reverse conducted beat positions and the values derived from manually correcting these values by ear using an soundfile editor.

This corrected beat duration data was measured by listening to the audio file and manually locating the beat positions in the soundfile. The initial positions of the average reverse conducting performance were used as a baseline, and this baseline was then adjusted so that it sounded as if all beats were occurring with the attacks of notes in the performance. There are probably a few errors in the corrected due to the tediousness of getting this data, and there are a few locations in the audio where the beat positions become vague, but overall, this data represents a highly accurate description of the pianists' actual beat-by-beat performance tempo, probably an accuracy within 10 milliseconds for all beats.

```
corrected = Transpose[mcbeats][[2]]
```

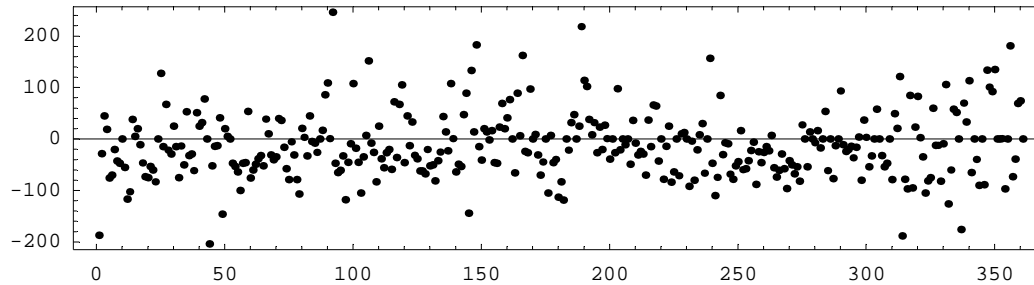
```
{1725, 2529, 3219, 3794, 4355, 4843, 5522, 6246, 6755, 7217, 7758, 8190, 8831, 9591, 10061, 10563,
11048, 11566, 12103, 12602, 13150, 13608, 14132, 14944, 16088, 16721, 17343, 17843, 18371,
18901, 19550, 20219, 20756, 21179, 21737, 22180, 22843, 23543, 24137, 24646, 25205, 26008,
26891, 27478, 28066, 28603, 29193, 30021, 30874, 31605, 32214, 32766, 33313, 33803, 34446,
35112, 35638, 36096, 36721, 37204, 37872, 38548, 38985, 39467, 39959, 40558, 41192, 41753,
42307, 42809, 43496, 44351, 45401, 46086, 46574, 47129, 47681, 48145, 48715, 49410, 49916,
50354, 50899, 51385, 52032, 52666, 53167, 53664, 54284, 55110, 55975, 57093, 57641, 58142,
58696, 59436, 60156, 60713, 61150, 61809, 62350, 62819, 63387, 64024, 64507, 65236, 65811,
66293, 66839, 67513, 67993, 68530, 69097, 69588, 70080, 70718, 71220, 71886, 72625, 73197,
73989, 74683, 75278, 75799, 76336, 76793, 77395, 78052, 78557, 79030, 79493, 79932, 80542,
81218, 81786, 82304, 82861, 83651, 84533, 85348, 85924, 86425, 87071, 87932, 88690, 89481,
89986, 90776, 91337, 91802, 92481, 93194, 93658, 94272, 94814, 95282, 95928, 96722, 97295,
97949, 98691, 99247, 99802, 100569, 101150, 101974, 102687, 103302, 104211, 104986, 105568,
106095, 106644, 107160, 107851, 108480, 109039, 109489, 109949, 110364, 111000, 111616, 112140,
112584, 113180, 113964, 114829, 115705, 116544, 117219, 118034, 118810, 119532, 120311, 120886,
121404, 121864, 122382, 122864, 123262, 123697, 124134, 124709, 125090, 125542, 125953, 126362,
126791, 127312, 127700, 128129, 128538, 128933, 129339, 129855, 130267, 130861, 131575,
132079, 132579, 133010, 133440, 133952, 134359, 134789, 135267, 135696, 136138, 136658,
137334, 137774, 138271, 138679, 139085, 139580, 140153, 140667, 141393, 142228, 143006,
143778, 144520, 145237, 145703, 146201, 146695, 147118, 147500, 147922, 148358, 148883,
149259, 149671, 150070, 150472, 150918, 151311, 151725, 152121, 152526, 152937, 153391,
153921, 154369, 154888, 155525, 156037, 156488, 156892, 157281, 157724, 158145, 158568,
158985, 159491, 159932, 160386, 161047, 161574, 162061, 162542, 162958, 163450, 164036,
164617, 165353, 165946, 166720, 167489, 168388, 169005, 169509, 169997, 170467, 170905,
171324, 171771, 172164, 172693, 173141, 173559, 173956, 174422, 174966, 175447, 175853,
176277, 176675, 177128, 177537, 178152, 178892, 180643, 181165, 181685, 182174, 182941,
183411, 184108, 184940, 185395, 185814, 186306, 186787, 187476, 188303, 188750, 189235,
189682, 190263, 191033, 191567, 192116, 192723, 193514, 194441, 195256, 196098, 196749,
197417, 198021, 198611, 199326, 200020, 200639, 201096, 201810, 202487, 203397, 204373,
204899, 205423, 206038, 206783, 207781, 208869, 209460, 210070, 210895, 211945, 213210}
```

```
avgtimes = Transpose[mabeats][[2]];
```

Here are the timing differences between the average reverse-conducting absolute times and the absolute times measured in a sound editor:

```
diffs = corrected - avgtimes;
```

```
ListPlot[diffs, Frame → True, AspectRatio → 1/4, PlotStyle → PointSize[0.008]];
```



The average beats are not quite centered on the corrected beats. The average beats are, on the average, 4.6 milliseconds after the corrected beats.

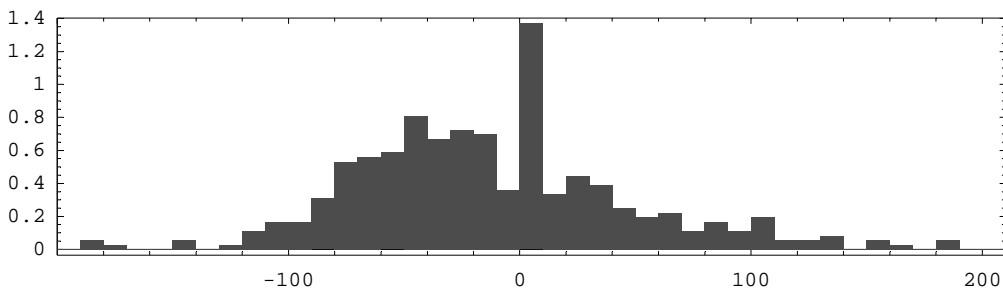
```
offseterror = Mean[diffs] // N
```

```
-12.3463
```

The histogram below shows the data from the plot above. It generally shows a nicely distributed range of corrections. The peak at 0 is due to the correction values being too small to change to the theoretically correct values about up to 10 milliseconds before their measured positions.

```
<< Graphics`Graphics`
```

```
Histogram[diffs, AspectRatio → 1/4, Frame → True, BarEdges → False,  
HistogramCategories → 50, HistogramScale → 100, HistogramRange → {-200, 200}];
```



■ Trial Quality Measurements

Define a logarithmic score for the amount of correction needed for a given beat. A score of 1 means 20 ms correction, 2 means 40 ms correction, 3 means 80 ms correction, 4, means 160 ms correction, 5 means 360 ms correction, *etc.* A score of 0 means there was less than 1 millisecond correction necessary, and a score of 0.5 means 10 milliseconds were needed to correct the beat location.

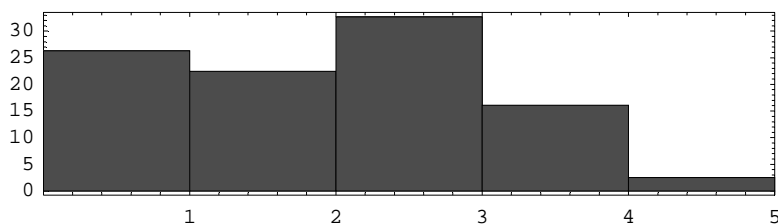
Qualitatively, a score of 0-1 is an "A", 1-2 is a "B", 2-3 is a "C", 3-4 is a "D", 4-5 is an "E", and 5-6 is an "F". A score of 6 means the correction was 640 milliseconds, or about 2/3 of a second.

```
QScore[seq_] := Module[{sign, scores},
  sign = Map[If[# < 0, -1, If[# > 0, 1, 0]] &, seq];
  scores = Log[2.0, Abs[seq] / 10] /. -Infinity -> 0;
  scores = Map[If[# < 1, (# + 1) / 2.0, #] &, scores];
  scores = Map[If[# < 0, 0, #] &, scores];
  scores * sign
]
```

Mostly the corrections for the average of the tapping trials are evenly distributed between A, B, C, and D quality, although "A" corrections were the most common. The worst case correction was between a value of 4 and 5. This occurs on the first beat when there has been no preparation of a previous event, so is most likely due to the reaction time of the reverse conductor. Other corrections in the range between 4 and 5 are also likely due to an unprepared change in tempo which was not expected beforehand.

```
<< Graphics`Graphics`
```

```
Histogram[Abs[QScore[mdiffs]], AspectRatio -> 1 / 4,
HistogramScale -> 100, Frame -> True, HistogramCategories -> 5];
```



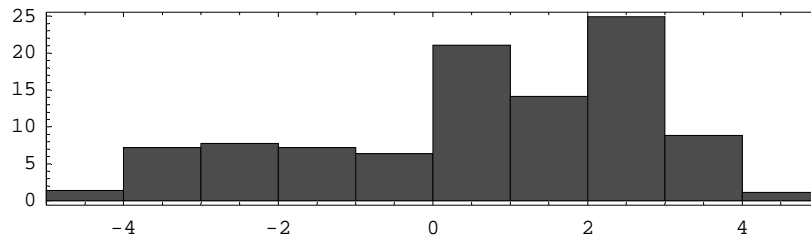
The range from 0-1 can be characterized as "no audible difference" (0-20 ms); 1-2 as "slight audible difference" (20-40 ms); 2-3 "minor audible difference" (40-80 ms); 3-4 as "noticeable audible difference" (80-160 ms), 4-5 as "very noticeable audible difference" (160-320 ms), and anything higher as "extremely noticeable audible difference" (> 320 ms).

```
Mean[Abs[QScore[mdiffs]]]
```

```
1.87813
```

Here is a histogram display of the necessary corrections with positive and negative adjustments separated. Negative values mean that the correct value occurs before the average reverse conducting time. Notice that most of the larger corrections occur in the negative range

```
Histogram[QScore[mdiffs], AspectRatio -> 1 / 4,
HistogramScale -> 100, Frame -> True, HistogramCategories -> 11];
```



■ Examine the learning curve

How does the quality of the reverse conducting improve over time? Does the conductor learn to follow the performance better after repeated listening? First, load the individual tapping trials which contain precalculated offset values into the audio file of the performance recording:

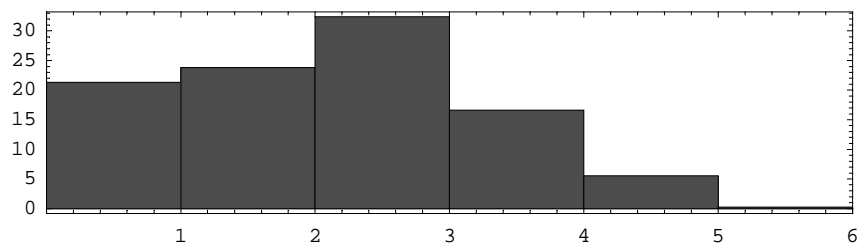
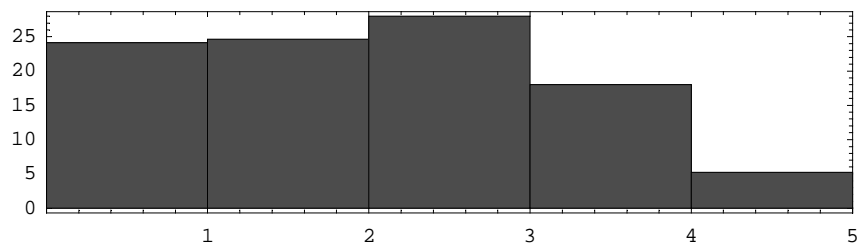
```
Get[performance <> "-abs.ma", Path -> "D:\mazurkas\eval"]
data = ToExpression[pvar <> "abs"];
```

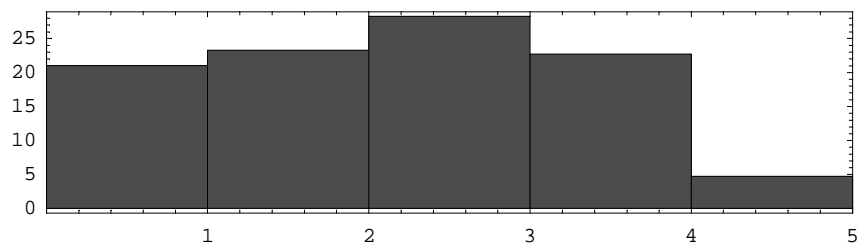
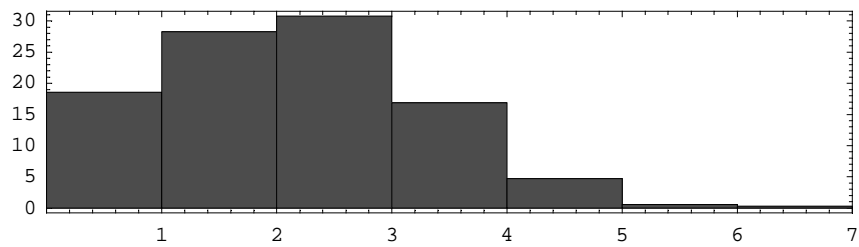
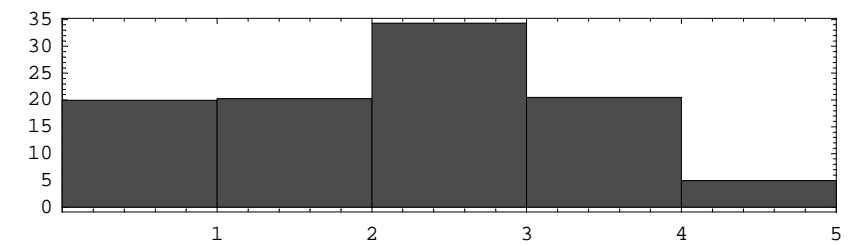
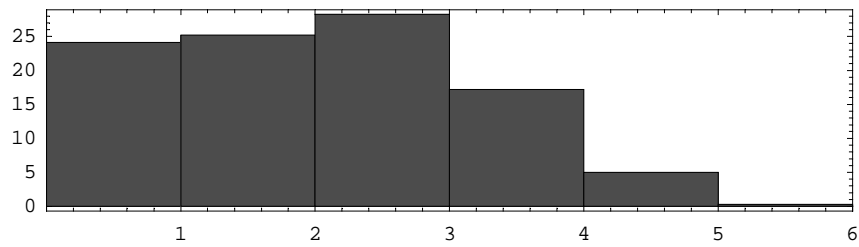
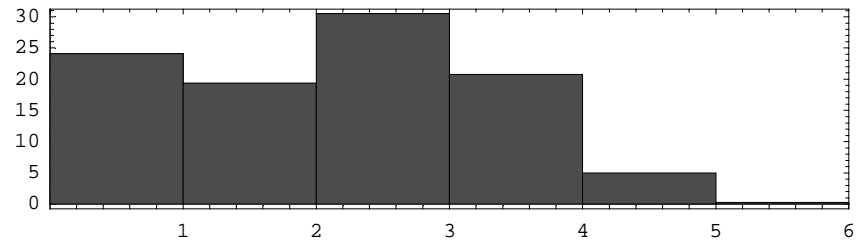
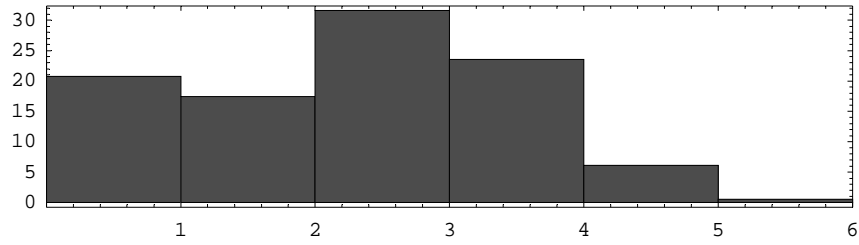
Now calculate the timing differences between the individual trials and the corrected beat times.

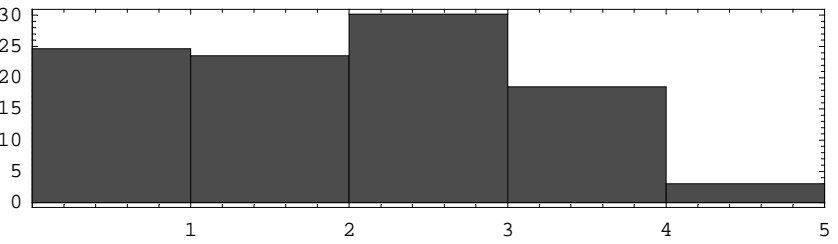
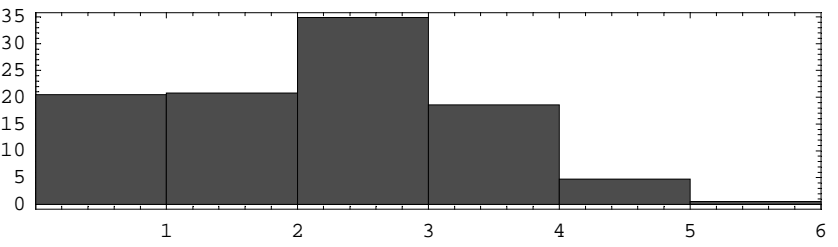
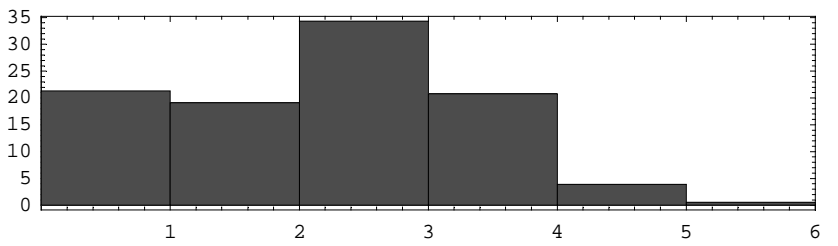
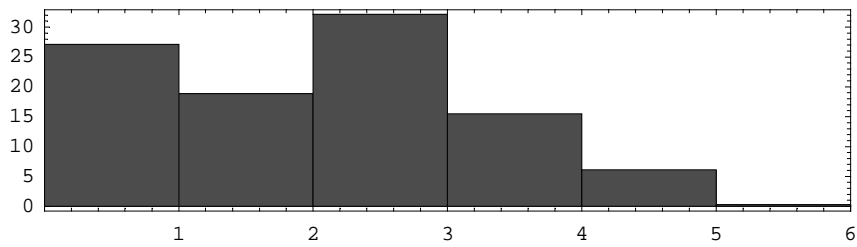
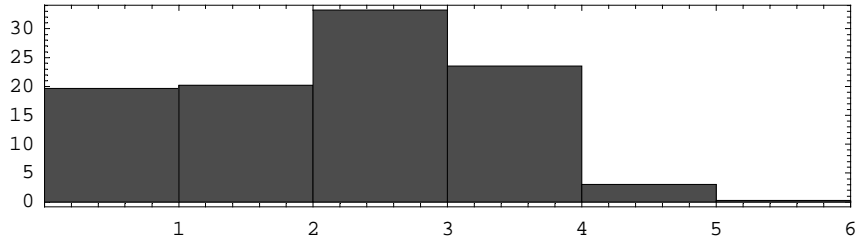
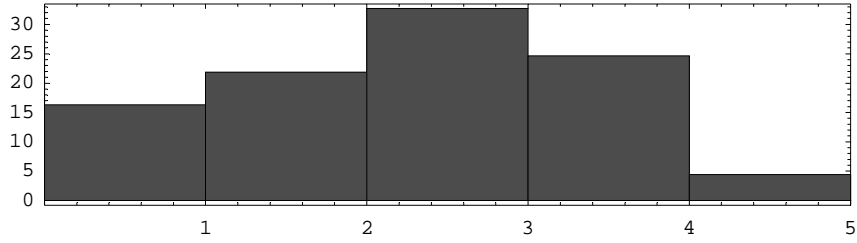
```
trialdiffs = Transpose[Transpose[data] - Transpose[mcbeats][[2]]];
Map[Mean[Abs[QScore[#]]] &, trialdiffs]
{2.04135, 2.06811, 2.24259, 2.14905, 2.00161, 2.18155,
 2.12978, 2.11026, 2.27989, 2.17168, 2.01347, 2.16474, 2.1607,
 1.99798, 2.09167, 2.21806, 2.20949, 2.12877, 2.27043, 2.17207}
```

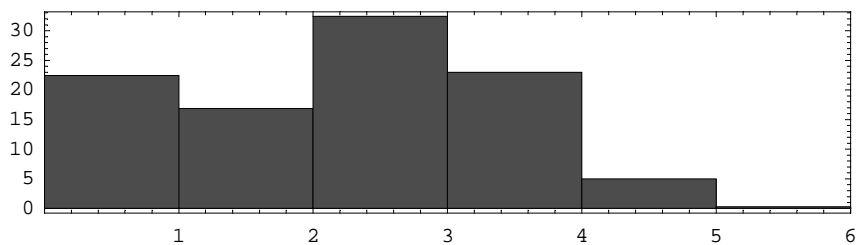
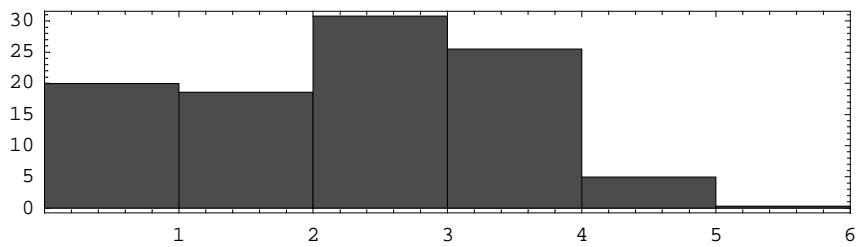
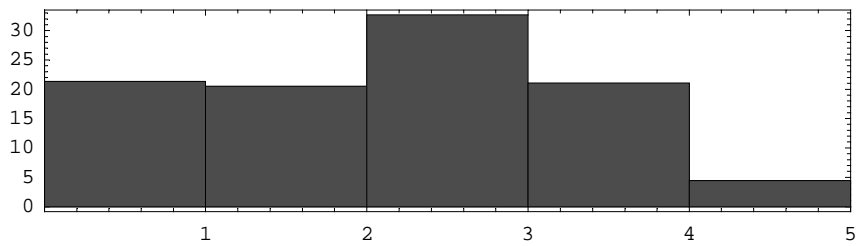
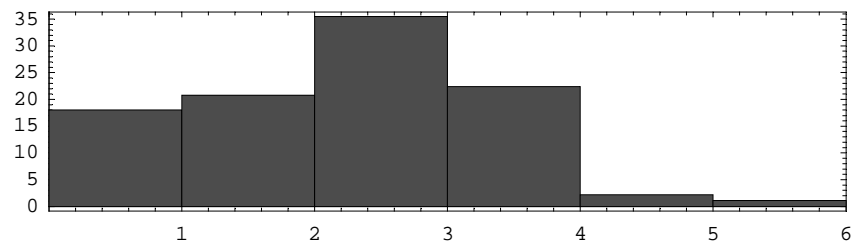
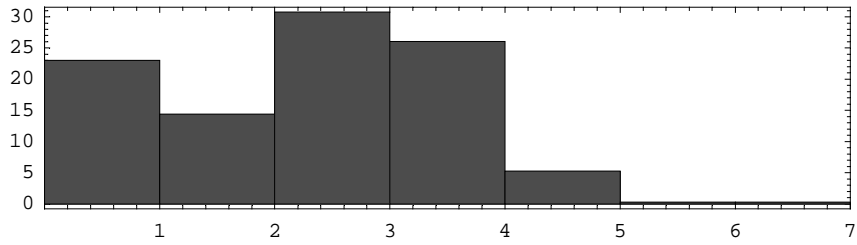
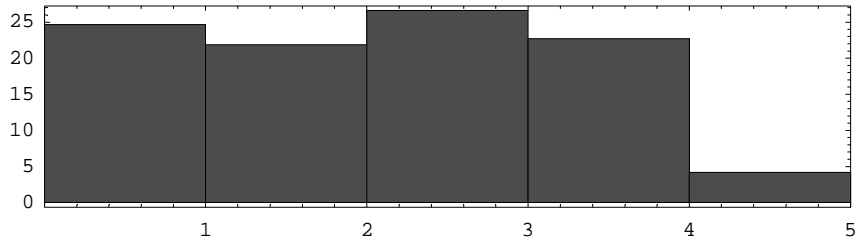
Here are quality score plots for each trial:

```
Table[Histogram[Abs[QScore[trialdiffs[[xz]]]], AspectRatio -> 1 / 4,
Frame -> True, Range -> {{0, 6} {0, 30}}, HistogramScale -> 100,
HistogramCategories -> 5], {xz, 1, Length[trialdiffs]}];
```









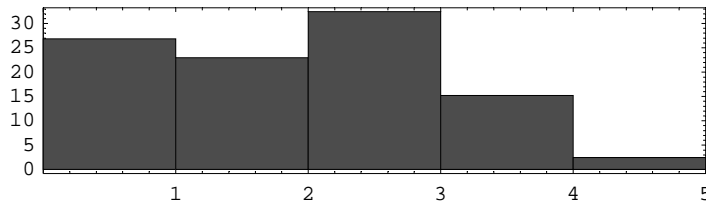
How does the accuracy of the first half of the trials compare to the second half of the trials?:


```

firsttrials = Drop[trialdiffs, -10];
lasttrials = Drop[trialdiffs, 10];
avgfirst = Map[Mean, Transpose[firsttrials]];
avglast = Map[Mean, Transpose[lasttrials]];

Histogram[Abs[QScore[avgfirst]], AspectRatio -> 1 / 4,
HistogramScale -> 100, Frame -> True, HistogramCategories -> 5];

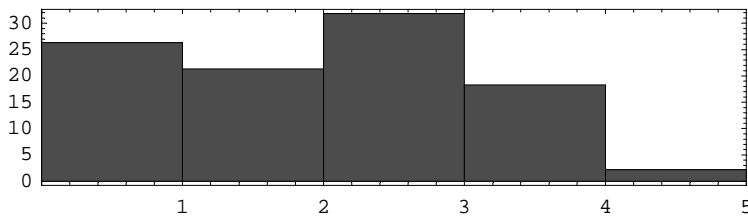
```



```

Histogram[Abs[QScore[avglast]], AspectRatio -> 1 / 4,
HistogramScale -> 100, Frame -> True, HistogramCategories -> 5];

```



```
Mean[Abs[QScore[avgfirst]]]
```

```
1.9081
```

```
Mean[Abs[QScore[avglast]]]
```

```
1.94312
```

On the average, beats in the second half of the trials are 6 milliseconds closer to the "true" beat locations:

```
firsterr = Mean[Abs[avgfirst]]
```

```
49.4903
```

```
lasterr = Mean[Abs[avglast]]
```

```
50.5104
```

```
firsterr - lasterr
```

```
-1.02016
```

```
allerr = N[Mean[Abs[mdiffs]]]
```

```
48.5568
```

```
allerr - lasterr
```

```
-1.95365
```

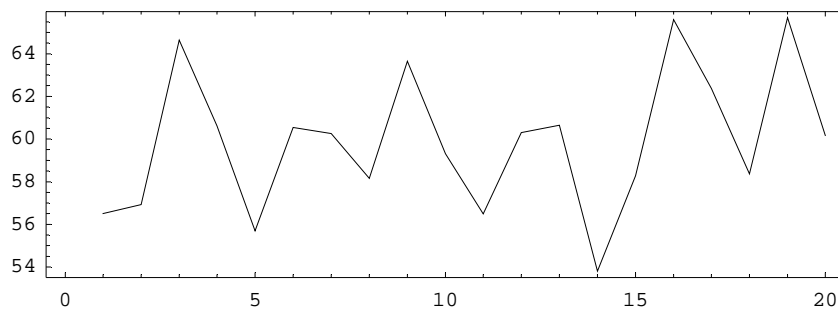
■ Plotting the average displacement error for each trial

This section plots the average displacement error for each trial which shows the gradual improvement in the reverse conducting over time.

```
avgerr = Table[Apply[Plus, Sqrt[ trialdiffs[[i]] * trialdiffs[[i]] ] ] /
  Length[trialdifs[[i]]] * 1.0 , {i, 1, Length[trialdifs]}]

{56.507, 56.9239, 64.6343, 60.6135, 55.7011, 60.5374,
  60.2579, 58.154, 63.6383, 59.3075, 56.4848, 60.3004, 60.6507,
  53.8017, 58.2758, 65.6013, 62.3704, 58.3653, 65.6864, 60.1634}

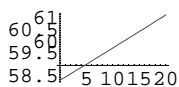
errrplot = ListPlot[avgerr, PlotJoined→True, Frame→True, AspectRatio→1/3];
```



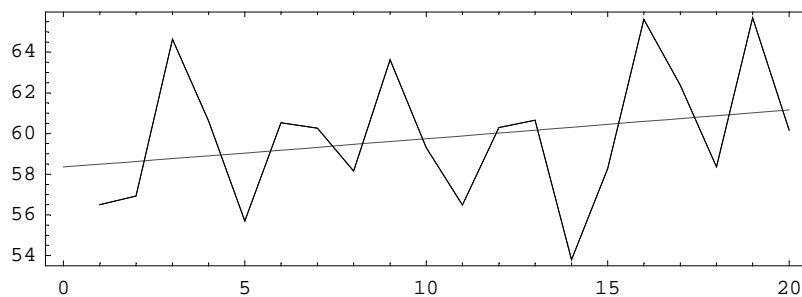
Now fit an exponentially decreasing line though the trial errors to model what the "learning curve" is for reverse conducting this performance.

```
logfit = Fit[Log[10, avgerr], {1, x}, x]
1.76607 + 0.00102197 x

fitplot = Plot[10^logfit, {x, 0, 20}, PlotStyle→Hue[0]];
```



```
Show[errrplot, fitplot, errrplot];
```



Predictions of accuracy can be derived from the learning curve over time:

```
err40 = 10^logfit /. x -> 40
```

```
64.113
```

```
lasterr - err40
```

```
-13.6026
```

```
err100 = 10^logfit /. x -> 100
```

```
73.8353
```

```
lasterr - err100
```

```
-23.3249
```

■ Effect of removing the "worst" trial

What happens if the "worst" trial is removed from the average tapping time for each beat?

```
maxindex = 1;
```

```
For[i = 1, i < Length[avgerr], i++,
```

```
  If[avgerr[[maxindex]] < avgerr[[i]], maxindex = i];
```

```
];
```

```
maxindex = maxindex
```

```
19
```

```
newerrset = RotateRight[Drop[RotateLeft[trialdiffs, maxindex], -1], maxindex - 1];
```

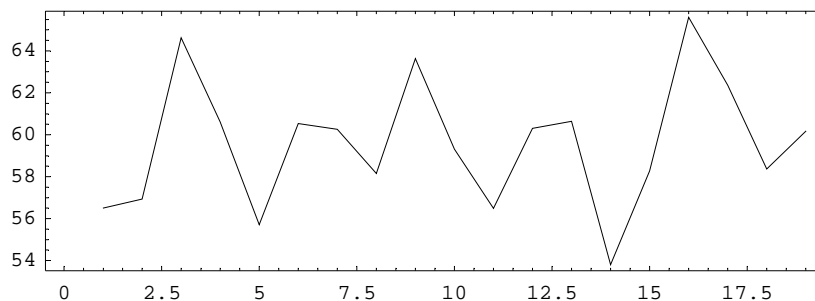
```
newavgerr = Table[
```

```
  Apply[Plus, Sqrt[newerrset[[i]] * newerrset[[i]]] / Length[newerrset[[i]]] * 1.0 ,
```

```
  {i, 1, Length[newerrset]}]
```

```
{56.507, 56.9239, 64.6343, 60.6135, 55.7011, 60.5374, 60.2579, 58.154, 63.6383, 59.3075,
 56.4848, 60.3004, 60.6507, 53.8017, 58.2758, 65.6013, 62.3704, 58.3653, 60.1634}
```

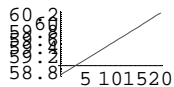
```
newplot = ListPlot[newavgerr, PlotJoined -> True, Frame -> True, AspectRatio -> 1 / 3];
```



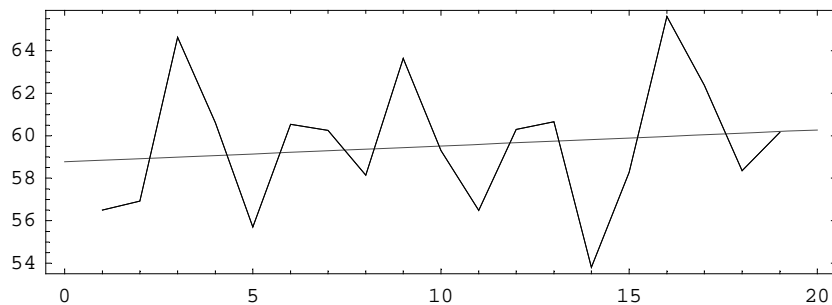
```
logfit = Fit[Log[10, newavgerr], {1, x}, x]
```

```
1.7692 + 0.00054558 x
```

```
newfitplot = Plot[10^logfit, {x, 0, 20}, PlotStyle -> Hue[0]];
```



```
Show[newplot, newfitplot, newplot];
```



```
newavg = Mean[Abs[Map[Mean, Transpose[newerrset]]]]
```

```
48.6299
```

```
allerr
```

```
48.5568
```

■ Trying to identifying "worst" performances from the average

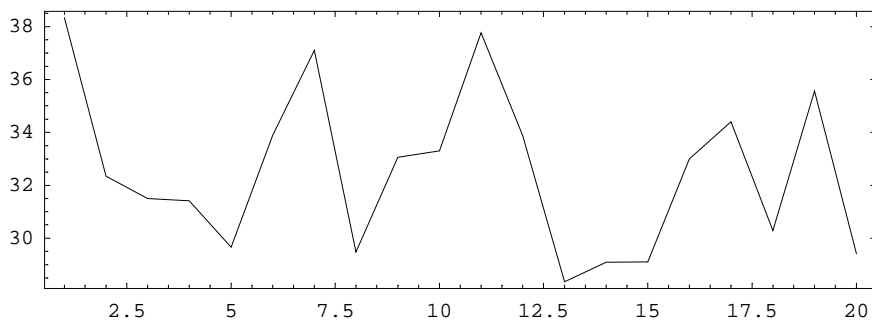
Instead of comparing the individual performances to the corrected data, compare them to the average of all trials.

```
avgtrialdiffs = Transpose[Transpose[data] - Transpose[mabeats][[2]]];
```

```
errfromavg = Map[Mean, Abs[avgtrialdiffs]] // N
```

```
{38.331, 32.3445, 31.4958, 31.422, 29.6624, 33.8864,  
37.0975, 29.4772, 33.0576, 33.3012, 37.7618, 33.8766, 28.3517,  
29.0929, 29.1071, 33.0052, 34.4049, 30.2876, 35.5696, 29.4201}
```

```
errminusone = ListPlot[errfromavg, PlotJoined -> True,  
Frame -> True, PlotRange -> All, Axes -> False, AspectRatio -> 1 / 3];
```



■ Dropping any single trial

Dropping the worst trial does not improve the accuracy of the average. Does dropping any other trial improve the average of all trials?

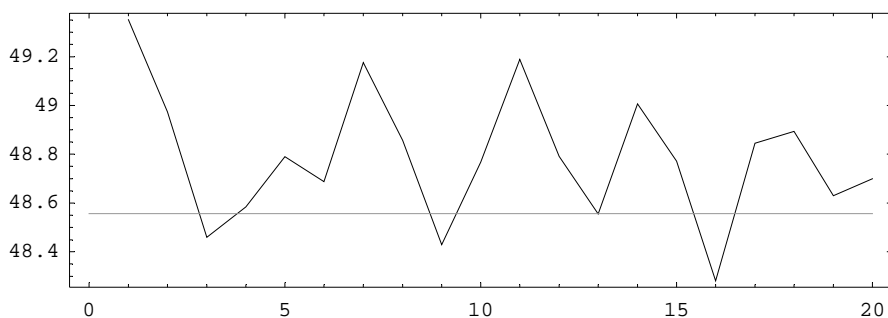
```
droppedsets = Table[RotateRight[Drop[RotateLeft[trialdiffs, i], -1], i], {i, 1, 20}];

droppedavg = Table[Mean[Abs[Map[Mean, Transpose[droppedsets[[i]] ]]]], {i, 1, 20}]

{49.3511, 48.9733, 48.46, 48.5845, 48.7905, 48.6879,
 49.1756, 48.857, 48.4298, 48.7671, 49.1889, 48.7916, 48.5553,
 49.0072, 48.7714, 48.2813, 48.8454, 48.8939, 48.6299, 48.6992}

plot = ListPlot[droppedavg, PlotJoined → True,
  Frame → True, Axes → False, DisplayFunction → Identity];

Show[plot, Graphics[{RGBColor[0, 1, 0], Line[{{0, allerr}, {20, allerr}}]}],
  DisplayFunction → $DisplayFunction, AspectRatio → 1 / 3];
```



■ Dropping a range of trials

Now examine how the displacement error changes as more and more of the earlier trials are removed from the average.

```
dsets = Table[Drop[trialdiffs, i], {i, 1, 19}];

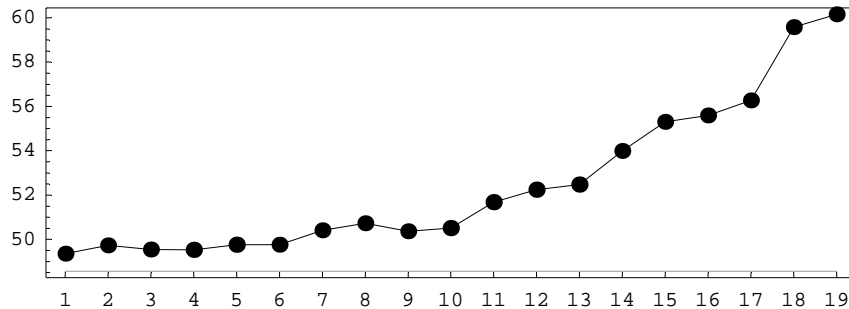
davg = Table[Mean[Abs[Map[Mean, Transpose[dsets[[i]] ]]]], {i, 1, 19}]

{49.3511, 49.7347, 49.5443, 49.5338, 49.7629, 49.7653, 50.3979, 50.7338, 50.3732,
 50.5104, 51.6882, 52.251, 52.4764, 53.9909, 55.2995, 55.5872, 56.2678, 59.5909, 60.1634}

dplot = ListPlot[davg, PlotJoined → True,
  Frame → True, Axes → False, DisplayFunction → Identity];

dplotp = ListPlot[davg, PlotStyle → PointSize[0.02],
  Frame → True, Axes → False, DisplayFunction → Identity];
```

```
dropstart =
  Show[dplot, dplotp, Graphics[{RGBColor[0, 1, 0], Line[{{1, allerr}, {19, allerr}}]}],
  FrameTicks -> {Range[19], Automatic, None, None},
  DisplayFunction -> $DisplayFunction, PlotRange -> All, AspectRatio -> 1/3];
```



Here is a plot created by dropping the later trials more and more:

```
rdsets = Table[Drop[trialdiffs, -i], {i, 1, 19}];

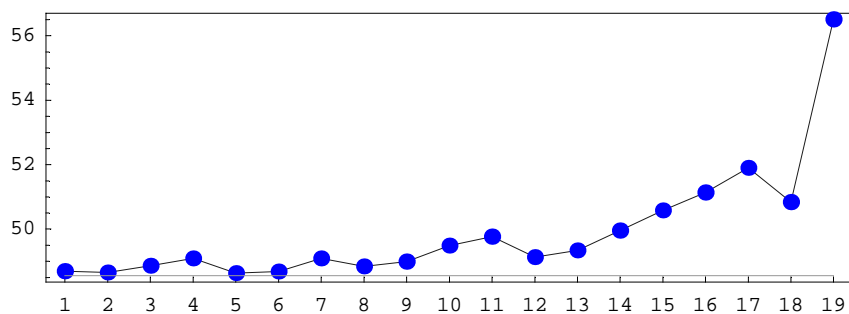
rdavg = Table[Mean[Abs[Map[Mean, Transpose[rdsets[[i]]]]]], {i, 1, 19}]

{48.6992, 48.6586, 48.8658, 49.0951, 48.6422, 48.6863, 49.089, 48.8473, 48.9915, 49.4903,
  49.7696, 49.1352, 49.3441, 49.9554, 50.5793, 51.1396, 51.8999, 50.837, 56.507}

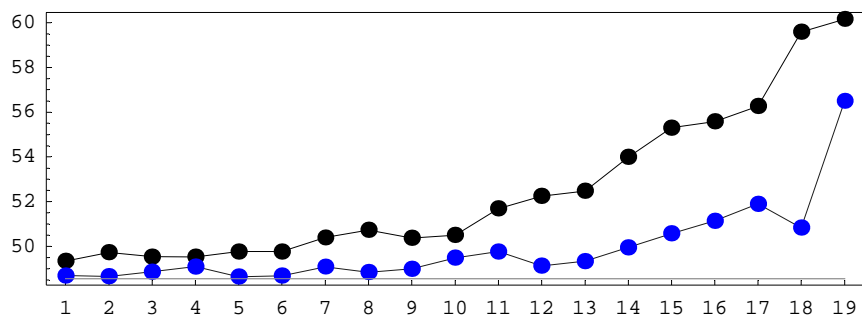
rdplotp = ListPlot[rdavg, PlotStyle -> {RGBColor[0, 0, 1], PointSize[0.02]},
  Frame -> True, Axes -> False, DisplayFunction -> Identity];

rdplot = ListPlot[rdavg, PlotJoined -> True, Frame -> True,
  Axes -> False, DisplayFunction -> Identity, PlotStyle -> RGBColor[0, 0, 1]];

dropend = Show[rdplot, rdplotp,
  Graphics[{RGBColor[0, 1, 0], Line[{{1, allerr}, {19, allerr}}]}],
  FrameTicks -> {Range[19], Automatic, None, None},
  DisplayFunction -> $DisplayFunction, PlotRange -> All, AspectRatio -> 1/3];
```



```
Show[dropstart, dropend];
```



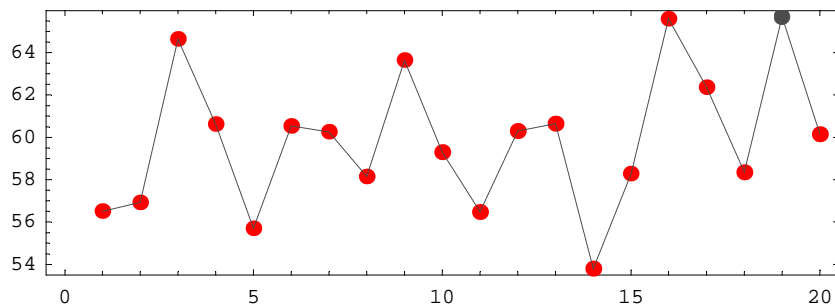
```
individ = Map[Mean[N[Abs[# - Transpose[mcbeats][[2]]]]] &, data]
```

```
{56.507, 56.9239, 64.6343, 60.6135, 55.7011, 60.5374,
 60.2579, 58.154, 63.6383, 59.3075, 56.4848, 60.3004, 60.6507,
 53.8017, 58.2758, 65.6013, 62.3704, 58.3653, 65.6864, 60.1634}
```

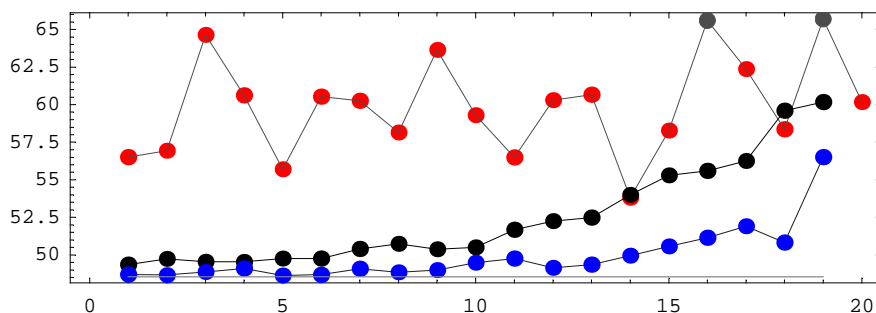
```
individplot = ListPlot[individ, PlotStyle -> {Hue[0.0], PointSize[0.02]},
  Frame -> True, Axes -> False, DisplayFunction -> Identity, AspectRatio -> 1/3];
```

```
individplotp = ListPlot[individ, PlotJoined -> True, Frame -> True,
  Axes -> False, PlotStyle -> Hue[0.0], DisplayFunction -> Identity];
```

```
Show[individplot, individplotp, DisplayFunction -> $DisplayFunction];
```



```
Show[individplot, individplotp,
  dropstart, dropend, DisplayFunction -> $DisplayFunction];
```



The red curve in the plot above shows the average displacement errors from the corrected times for each of the 20 individual trials. The blue curve represents dropping more and more of the later trials being dropped. The black curve represents dropping more and more of the earlier trials.

■ Offset sensitivity

Individual trials were aligned to the test case using a few points which seemed to be in a stable tempo region. How accurate is the offset calculated from these few points in the audio? Would the average of all trials improve if an offset is calculated for each trial based on the corrected data?

```

SquareDiff[x_, y_] := Apply[Plus, (x - y) * (x - y)]

AlignSequences[seq1_, seq2_] := Module[
  {diffs},
  diffs = Table[{x, SquareDiff[seq1, seq2 + x]}, {x, -10000, 10000, 100}];
  (x /. Solve[D[Fit[diffs, {1, x, x^2}, x], x] == 0, x])[1]
]

AlignSequences[{1, 2, 3}, {2, 3, 4}]

-1.

addoffsets =
  Table[AlignSequences[Transpose[mcbeats][[2]], data[[i]]], {i, 1, Length[data]}]
{1.92753, -5.1369, -18.4903, -5.94693, -4.13806, -25.374,
 4.06809, -7.46698, -20.5283, -20.474, 15.7202, -7.74283, -9.09105,
-9.50737, -17.8417, -24.4784, -21.2948, -19.4867, -30.9822, -20.8939}

newmabeats = Map[Mean, Transpose[data + addoffsets]];

extraoffset = Mean[addoffsets]

-12.3579

Min[addoffsets]
Max[addoffsets]

-30.9822

15.7202

omdiffs = Transpose[mcbeats][[2]] - newmabeats;

N[Mean[Abs[omdiffs]]]

46.9017

```

■ Comparing corrected beats to average reverse conducting beats

This section plots the average tempo range and compares it to the actual tempos measured from the audio file.

```

Get[performance <> ".ma", Path → "D:\mzurkas\eval"]
data = Transpose[ToExpression[pvar]];

```



```

eachbeatstats = Map[DispersionReport, data] // N;

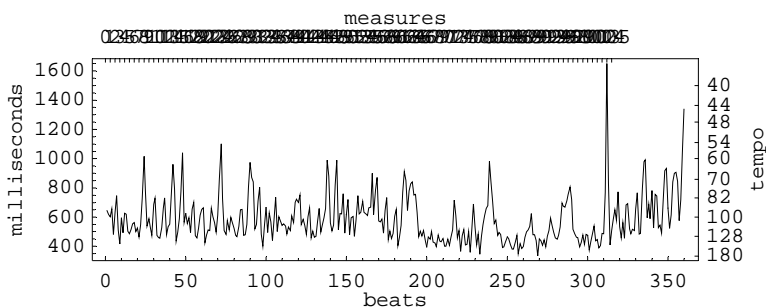
confidence =
  Map[MeanCI[#[[1]], KnownStandardDeviation → (KnownStandardDeviation /. #[[2]])] &,
    Transpose[{data, eachbeatstats}]];

durationmean = Map[Mean, data] // N;
maxdurs = Map[Max, data];
mindurs = Map[Min, data];

meanhighplot = ListPlot[Transpose[confidence][[1]], PlotJoined → True,
  DisplayFunction → Identity, PlotStyle → RGBColor[.5, .5, .5]];
meanlowplot = ListPlot[Transpose[confidence][[2]], PlotJoined → True,
  DisplayFunction → Identity, PlotStyle → RGBColor[.5, .5, .5]];
maxplot = ListPlot[maxdurs, DisplayFunction → Identity,
  PlotJoined → True, PlotStyle → RGBColor[0.8, 0.8, 0.8]];
minplot = ListPlot[mindurs, DisplayFunction → Identity,
  PlotJoined → True, PlotStyle → RGBColor[0.8, 0.8, 0.8]];

meanplot = ListPlot[durationmean,
  PlotJoined → True, PlotRange → All, AspectRatio → 1/3, Frame → True,
  FrameLabel → {"beats", "milliseconds", "measures", "tempo"},
  FrameTicks → {Automatic, Automatic, Table[{x, x/3}, {x, 0, 315, 3}],
    Map[{60000.0 / #, #] &, {30, 40, 44, 48, 54, 60, 70, 82, 100, 128, 180, 280}}]];

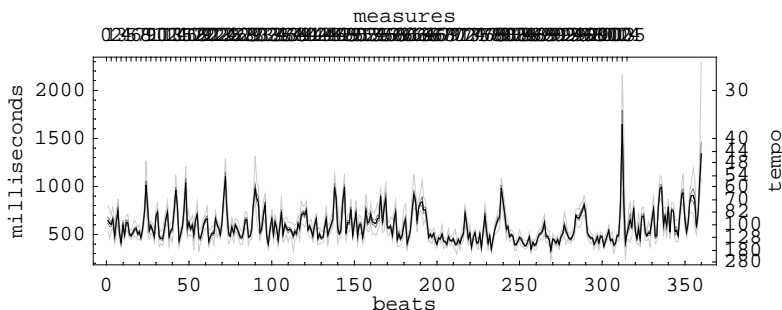
```



```

allplot =
  Show[meanplot, maxplot, minplot, meanhighplot, meanlowplot, meanplot, Axes → False];

```



```

correctdeltas = Drop[RotateLeft[corrected, 1] - corrected, -1];

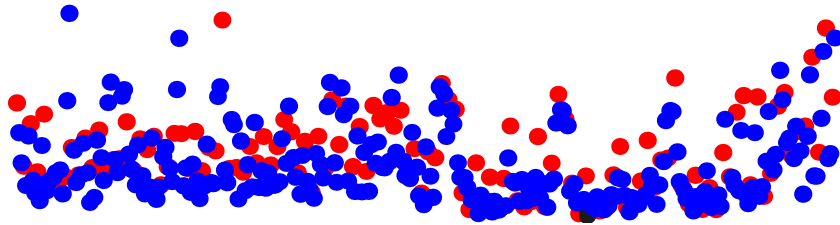
```

```

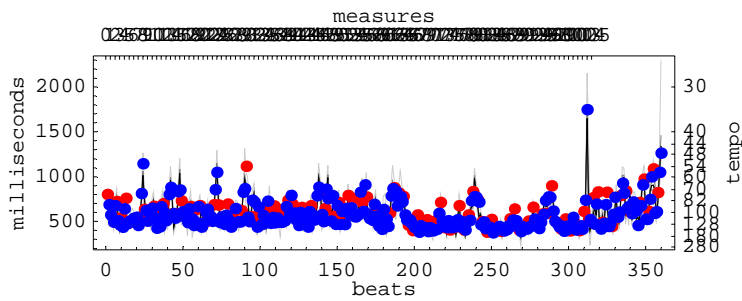
cordels = Transpose[
  Partition[Transpose[{Drop[Transpose[mcbeats][[1]] + 1, -1], correctdeltas}], 3]];

```

```
correctplot = Show[Graphics[
  {PointSize[0.02], RGBColor[1, 0, 0], Map[Point, cordels[[1]]], RGBColor[0, 0, 1],
  Map[Point, cordels[[2]]], Map[Point, cordels[[3]]]}], AspectRatio -> 1 / 4];
```

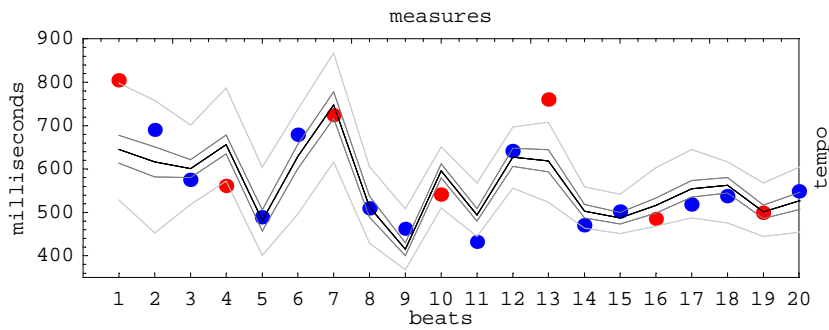


```
newallplot = Show[allplot, correctplot];
```



```
finalplot =
```

```
Show[allplot, correctplot, allplot, Frame -> True, PlotRange -> {{0, 20}, {350, 900}},
  FrameTicks -> {Range[20], Automatic, Automatic, None}];
```



```
ShowFrame[finalplot_, minrange_, maxrange_, len_, frame_, width_] := Module[
  {framecount, startpoint, endpoint, rmin, rmax, starti, endi, delta},
  framecount = len / width + 0.99 // Floor;
  startpoint = frame * width + 1;
  endpoint = (frame + 1) * width + 1;
  starti = (startpoint - 1) + 1;
  endi = (endpoint - 1) + 1;
  If[endi > len, endi = len];
  If[starti > len, starti = len];
  rmax = Max[Take[maxrange, {starti, endi}]];
  delta = rmax * 0.03;
  rmax = rmax + delta;
  rmax = 1500;
  rmin = Min[Take[minrange, {starti, endi}]];
  rmin = rmin - delta;
  rmin = 200;
  Show[finalplot, PlotRange -> {{startpoint, endpoint}, {rmin, rmax}},
  FrameLabel -> {"beats", "milliseconds", "measures", "tempo"},
  FrameTicks -> {Automatic, Automatic, Table[{x + 1, x / 3 + 1}, {x, 0, 2000, 3}],
  Map[{60000.0 / #, #} &, {30, 40, 44, 48, 54, 60, 70, 82, 100, 128, 180, 280}]},
  Axes -> False]
]
```

Here is a zoom in on the tempo for every 8 measures. The black line indicates the mean reverse-conducted durations for every beat. The dark gray lines surrounding the average duration line is the 95% confidence range for the true mean, and the light gray lines indicate the maximum and minimum durations for each beat from all reverse conducting trials.

The red and blue dots indicates the beat event durations from the audio file which are more accurate than the average reverse conducting durations, and can be assumed to be the "correct" answer. The red dots represent the first beat of a measure, and the blue dots represent the other beats in the measure.

```
Table[ShowFrame[newallplot, mindurs, maxdurs, Length[data], i, 24],
  {i, 0, Floor[Length[data] / 8.0 / 3.0 + .99] - 1}];
```

