SelfOtherEffect-ClarinetDuo-ICPA2011

Davi Alves Mota, Thiago de Almeida Magalhães Campolina, Maurício Alves Loureiro

Evidence for self-other effects and structure-related attentional mechanisms in the primo/secondo performances of clarinetists

Work presented at the International Conference of Perception and action 2011 by: Rafael Laboissière, Davi Alves Mota, Thiago de Almeida Magalhães Campolina, Hani Yehia, Maurício Alves Loureiro

Introduction

This study aims at understanding the mechanisms of on-line synchronization of instrument performance in musical ensembles. Musicians use a variety of strategies for manipulating acoustic material in order to encode their expressive intentions while performing a score. When playing with others, a performer must synchronize the results of his own actions with the actions of the others. When playing in a real situation like in an orchestra, performers take advantage of acoustical and visual information in order to improve the synchronization. However, performers are also capable of accompanying another performer solely from an acoustical recording. The great precision of temporal manipulation of acoustic events has been the object of many studies on musical expressiveness.

A series of studies in the literature have shown that musicians play better with themselves than with others [1,2,3]. This is known as the "self/other effect" in action recognition and simulation and has been demonstrated in other action-perception tasks, like the recognition of handwriting [4] and the outcomes of dart throwing [5]. In the present study, we analyzed the synchronization patterns of professional clarinetists.

[1] Keller PE, Knoblich G, & Repp BH (2007). Pianists duet better when they play with themselves: on the possible role of action simulation in synchronization. Conscious Cogn, 16(1):102–111.

[2] Keller PE & Appel M (2010). Individual differences, auditory imagery, and the coordination of body movements and sounds in musical ensembles. Music Percep 28:27–46.

[3] Keller PE (2001) Attentional resource allocation in musical ensemble performance. Psychol Music, 29:20–38.

[4] Knoblich G, Seigerschmidt E, Flach R, and Prinz W (2002). Authorship effects in the prediction of handwriting strokes: evidence for action simulation during action perception, Q J Exp Psychol A 55(3):1027–1046

[5] Knoblich G & Flach R (2001). Predicting the effects of actions: Interactions of perception and action, Psychol Sci 12:467–472.

Aim

Understanding the mechanisms of on-line synchronization of instrument performance in musical ensembles.

Materials and Methods

Participants

Five professional clarinet players and one student with previous experience in orchestra practice were recruited in Belo Horizonte, Minas Gerais, Brazil. All them knew each other. Two of them took clarinet lessons with the same teacher. Until the moment of the recordings, the student was a clarinet student of one of the other participants.

Music Material

In order to replicate as much as possible a real performance situation, we chose a short excerpt extracted form the *Dance of the Peasant and the Bear* of Igor Stravinsky's ballet *Petrushka*, where "the peasant plays the pipe and the bear walks on his hind feet". This is a *soli a due* passage in which first and second clarinets play in unison. The selected excerpt not only demands synchronization in every single, but also guarantees equal musical conditions for both co-performers, unlike in a regular duet situation, where *primo* and *second* play distinct parts. Besides that, the experiment presented a situation closer to a real performance situation, since professional clarinetists are all familiar with this unison condition.

Figure 1: Fig. 1: Soli a due

Experiment

The recordings were done in two sessions, separated by a period of three days:

1. In the first session, musicians were asked to play four times the Stravinsky excerpt as first clarinet (*primo*) with their preferred expressivive intentiosn. Nevertheless, before they started playing, three beats of metronome imposed the execution tempo, as in real situation where tempo is given by the conductor and orchestra accompaniment. Once the session was over, the musician was asked to choose one of the four recorded performancs, which later served as *primo* recording for the second session.
2. In the second session, musicians were instructed to play as the second clarinet (*secondo*), accompanying each of the *primo* recordings chosen in the first section, including those performed by themselves. The only instruction given was to accompany the *primo* in the best possible way. Three metronome beats were included at the beginning of each *primo* recording used in the second session. The tempo of each recording was estimated as the average time between the most prominent notes in the passage, the last three eighth notes at the end of the first measure. After hearing the *primo*'s execution once, the clarinetists played as *secondo* four times (*takes*), while listening to the *primo* through a headphone in his right ear. All *primo* recordings used in the second session were presented at random.

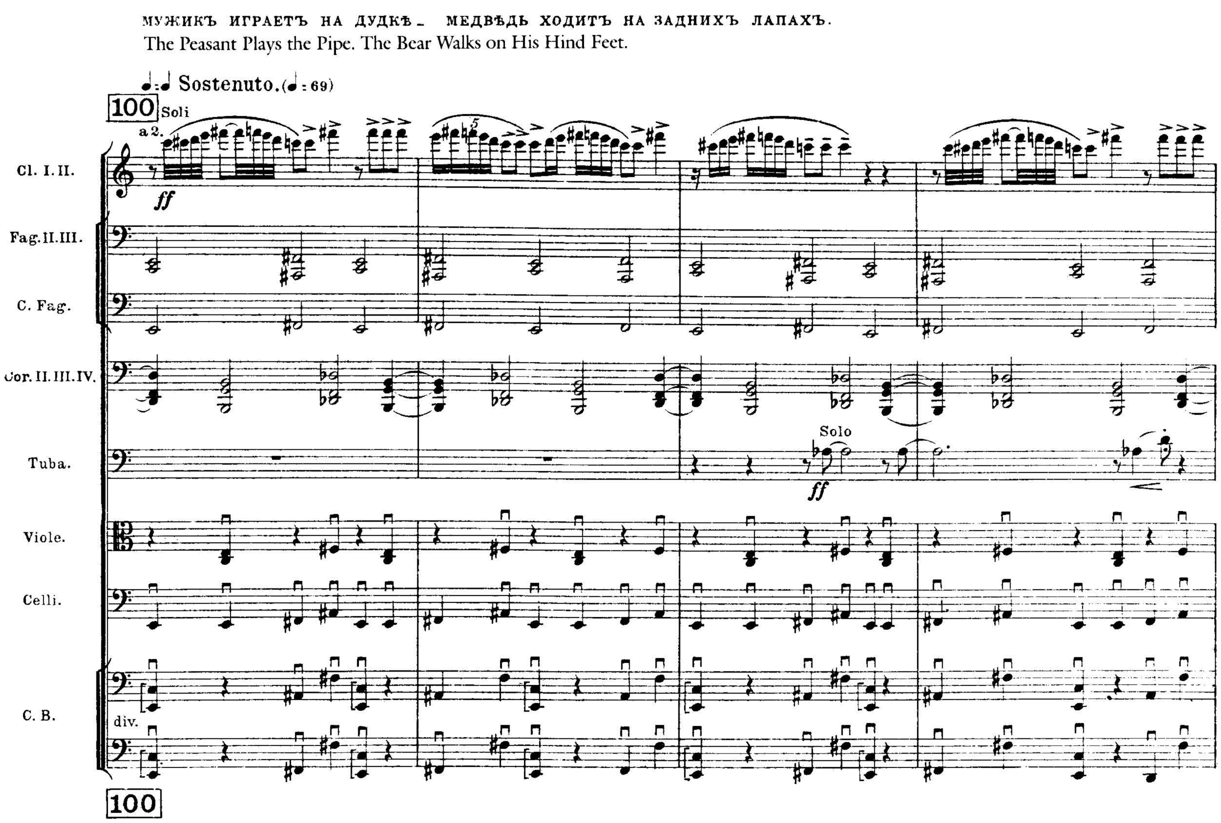


Figure 2: Fig. 2: Orchestral excerpt

Data acquisition and processing

The audio was captured using an M-Audio FireWire 1814 interface with a sampling frequency of 44,100 Hz, in just one channel, using an omnidirectional microphone in a room with basic acoustic treatment. The microphone was placed one meter away from the subjects, in order to avoid major changes in the amplitude of the signal, which could be caused by the movement of the clarinetists. Each clarinetist used their own instrument and materials during the recording sessions.

The onsets of each note were automatically detected using ***Expan***, a system tool specially developed in previous studies for empirical analysis of musical performance [6].

[6] Mauricio Alves Loureiro; Hani Camille Yehia; Hugo Bastos de Paula; Thiago de Almeida Magalhães Campolina; Davi Alves Mota.Content Analysis of Note Transitions in Music Performance. 6th Sound and Music Computing Conference (SMC 2009), 2009, Porto. Proceedings of the 6th Sound and Music Computing Conference (SMC 2009). Porto, Portugal: INESC Porto, 2009. p. 355-359. ISBN: 978-989-95577-6-5. Julho de 2009.

O código abaixo carrega os dados, nomeia as variáveis e calcula a assincronia absoluta (onset.secondo - onset.pirmo).
[author: Rafel Laboissiere (enviados por e-mail para o resumo do ICPA 2011].

```
# knitr::opts_chunk$set(echo = TRUE)
```

```
rm(list=ls())
library (car)
```

```
## Loading required package: carData
```

```
onset.data <- read.table ("raw_data.txt", header = FALSE)
names (onset.data) <- c ("primo", "secondo", "composer", "take","note", "primo.onset", "secondo.onset")
```

Total de 5904 observações = 4 takes x 41 notas x 36 duetos (6 clarinetistas em dueto com 6 clarinetistas, inclusive o próprio). Apenas 5330 observações foram capturadas. O código calcula missing takescomo primo e como secondo.

```
takes.pri <- tapply(onset.data$primo,onset.data$primo,length)/41
takes.sec <- tapply(onset.data$secondo,onset.data$secondo,length)/41
cbind(takes.pri, missing.takes.pri = 24-takes.pri, takes.sec, missing.takes.sec = 24-takes.sec)
```

```
##   takes.pri missing.takes.pri takes.sec missing.takes.sec
## A        23                 1        24                 0
## D        20                 4        22                 2
## F        17                 7        24                 0
## M        22                 2        24                 0
## N        24                 0        21                 3
## W        24                 0        15                 9
```

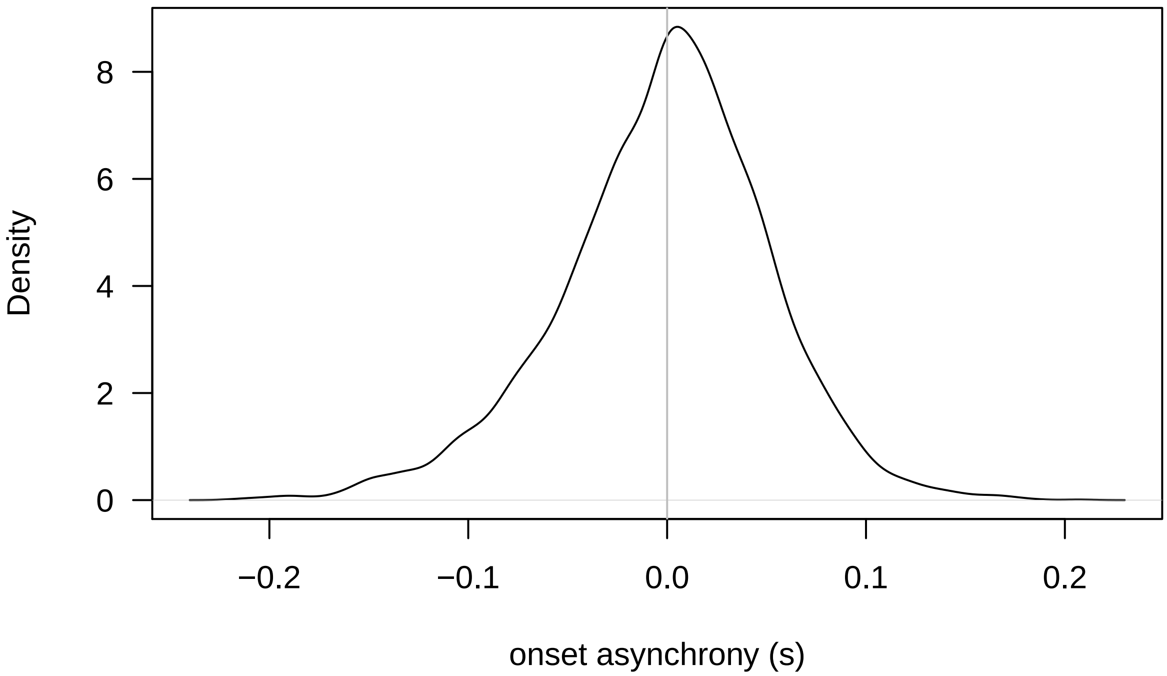
Asynchrony Measure

First, the temporal synchronization of each duo was analyzed, measuring the asynchrony between each note played by each clarinetist, defined as the time difference between the note onsets played by the clarinetist *secondo* and the clarinetist *primo* - positive asynchrony when *secondo* plays later then the *primo*. The observed distribution of signed values of asynchrony, thus defined, was symmetrical and showed mean and median close to zero (-1.4 ms), which would indicate that *secondos* tend to to antecipate the *primos*. However, the distribution peak lays around +5.6 ms, which would indicate delay in those executions, as shown in Fig. 3. The difference between mean and distribution peak happens when the distribution is not normal and highly skewed.

---

plota a distribuição da assincronia.
[author: Rafel Laboissier]

```
onset.data$asynchrony <- (onset.data$secondo.onset - onset.data$primo.onset) / 44100
d <- density(onset.data$asynchrony,na.rm=TRUE)
plot (d, las = 1, main = "", xlab = "onset asynchrony (s)")
abline (v = 0, col = "gray")
```




```
dev.copy2pdf (file = "figures/asynchrony-distribution.pdf")
```

```
## pdf
##   2
```

```
# calcula a média da assincronia e o instante onde é máxima
mean(onset.data$asynchrony,na.rm=T)*1000
```

```
## [1] -1.359458
```

```
d$x[which(d$y==max(d$y))]*1000
```

```
## [1] 5.604611
```

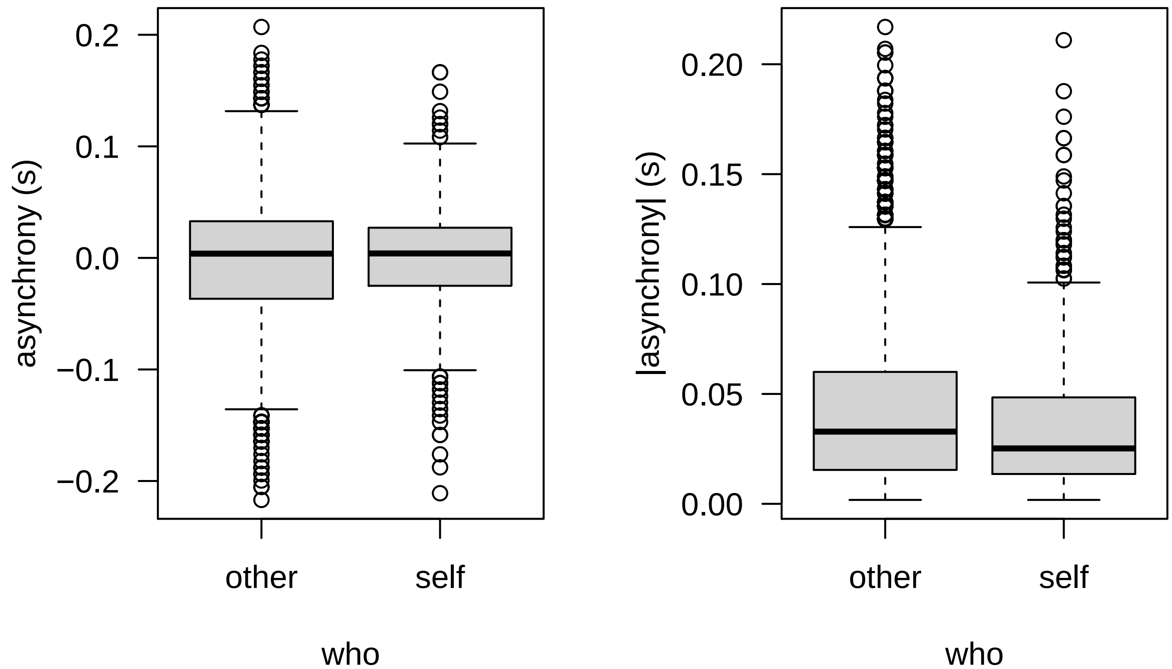
Fig. 3: Signed asynchrony distribution

The mean values of the signed asynchrony observed in each “self” and “other” situations, were both colse to zero with no significant difference, as shown in the left panel of ### Fig. 4.## The right panel shows the absolute asynchrony. The boxplots. Both asynchrony signed and absolute show grater variability for the *other* conditon.

---

boxplots das assincronias real e absoluta para cada condição self other.
[author: Rafel Laboissier]

```r
who <- rep ("other", nrow (onset.data))
idx <- which (onset.data$primo == onset.data$secondo)
who [idx] <- "self"
onset.data$who <- factor (who)

par(mfrow=c(1,2))
boxplot (asynchrony ~ who, data = onset.data, ylab = "asynchrony (s)", las = 1)
boxplot (abs (asynchrony) ~ who, data = onset.data, ylab = "|asynchrony| (s)", las = 1)
```




```r
par(mfrow=c(1,1))
dev.copy2pdf (file = "figures/signed-abs-asynchrony-who.pdf", width = 6, height = 7)
```

```
## pdf
##   2
```

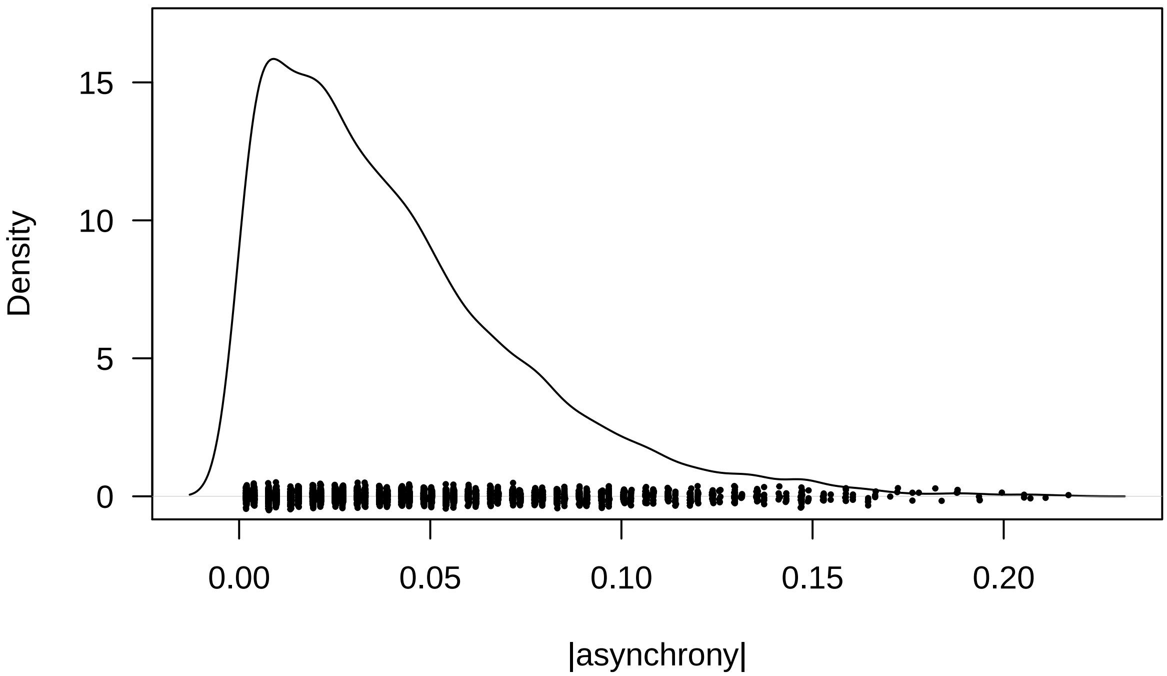
Fig. 4: Boxplots of signed and absolute asynchrony for *self* and *other* conditions.

Asynchrony was then defined as the absolute value of the time difference between the onset times of each note, resulting in an asymmetric distribution, as shown in Fig. 5.

---

distribuição da assincronia absoluta.
[author: Rafel Laboissier]

```r
onset.data$asynchrony <- abs(onset.data$asynchrony)
plot (density (onset.data$asynchrony, na.rm = TRUE), ylim = c(-0.15, 17), xlab = "|asynchrony|",
las = 1, main = "")
points (onset.data$asynchrony, rnorm (length (onset.data$asynchrony), sd = 0.15), pch = 19, cex = 0.3)
```




```
dev.copy2pdf (file = "figures/abs-asynchrony-distribution.pdf")
```

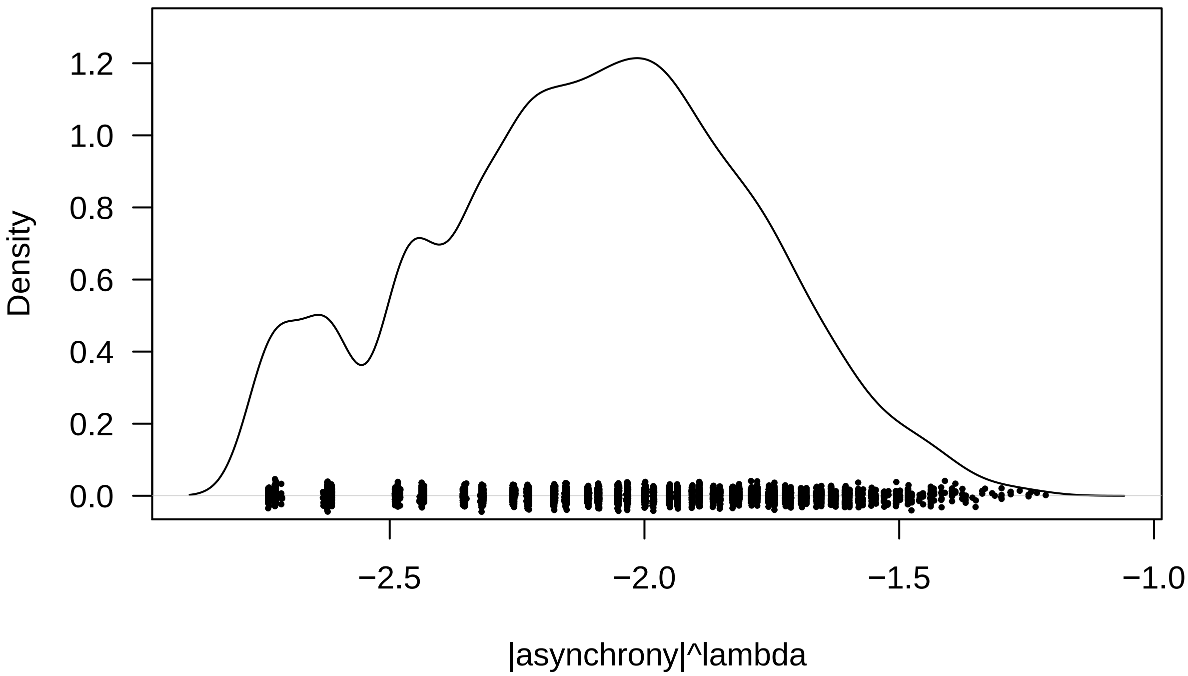
```
## pdf
##   2
```

Fig. 5: absolute asynchrony distribution with hihg skewness

The high skewness of the asynchrony distribution was compensated using a Box-Cox power transform [7].

Correção da assimetria: O bloco de códigos abaixo foi utilizado para a correção da assimetria das assincronias. A função =bcPower= realiza a transformação Box-Cox nos dados. A função =skewness.coeff= é utilizada para calcular a assimetria dos dados.

```
y <- bcPower (onset.data$asynchrony, powerTransform (onset.data$asynchrony)$lambda)
plot (density (y, na.rm = TRUE), ylim = c(-0.013, 1.3),
        xlab = "|asynchrony|^lambda", las = 1, main = "")
  points (y, rnorm (length (y), sd = 0.013), pch = 19, cex = 0.3)
```




```
dev.copy2pdf (file = "figures/corrected-abs-asynchrony-distribution.pdf")
```

```
## pdf
##   2
```

```
skewness.coeff <- function (x)
    mean ((x - mean (x, na.rm = TRUE))^3, na.rm = TRUE)/sd (x, na.rm = TRUE)^3

skewness.coeff (onset.data$asynchrony)
```

```
## [1] 1.341426
```

```
skewness.coeff (y)
```

```
## [1] -0.07330861
```

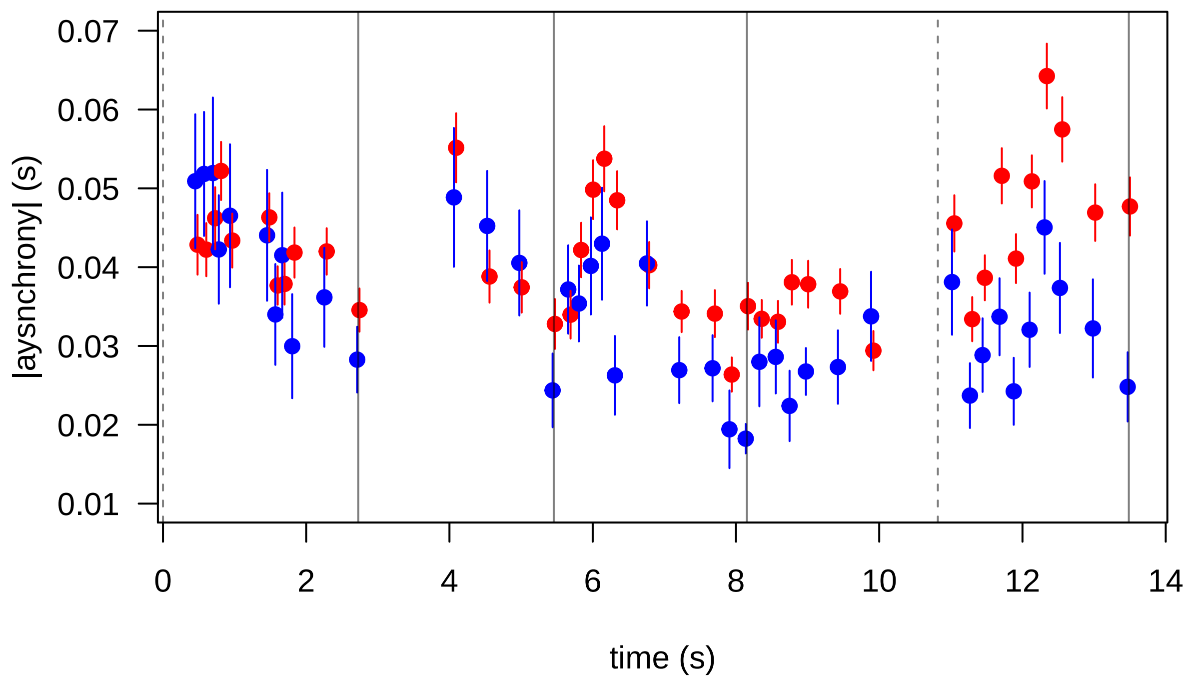
[7] Box GEP & Cox DR (1964). An analysis of transformations,J Roy Stat Soc B Met, 26(2):211–252.

Results

Asynchronies were significantly greater in the situation in which *secondos* accompanied other *primo* (situation *other*) compared to the situation in which *secondos* accompanied their own *primo* recordings (situation *self*). Fig. 5 shows the average values of asynchrony for each note on a time scale (situation *other* in red and *self* in blue). Low values of asynchrony were also observed in notes coinciding with strong measure pulses (the vertical lines correspond to the units of time in the measure).

```
## Copmpute note times from primo executions (for locating the notes along x axis)
onsets <- aggregate (primo.onset ~ note, data = onset.data, mean)$primo.onset / 44100
onsets <- onsets + onsets [15] - 2 * onsets [11]    # fixa o início da partitura = zero subtraindo a dur
# como não tem nota no tempo 1 do compasso 1, subtrai 2 metades do compasso 1 (2* nota 11)toma a duraçã
onset.data$time <- onsets [onset.data$note]

## Compute aggregate means and standard errors (for ploting on y axis)
onset.mean <- aggregate (asynchrony ~ who + time + note, data = onset.data, mean)
```

```
se <- function (x) { sd (x) / sqrt (length (x)) }
onset.se <- aggregate (asynchrony ~ who + time + note, data = onset.data, se)

## Plot results: points (time,asynchrony)
spacing <- 0.16 * c (0.1, -0.1)
colors <- c ("red", "blue")
plot (onset.mean$time + spacing , onset.mean$asynchrony, pch = 19,
      col = colors [onset.mean$who], las = 1, xlab = "time (s)",
      ylab = "|aysnchrony| (s)", ylim = c (0.01, 0.07))

## Plot results: standart errors lines upwards & downwards on points (time,asynchrony)
condition <- c ("other", "self")
for (i in 1 : 41) {
    for (j in 1 : 2) {
        idx <- which (onset.mean$note == i & onset.mean$who == condition [j])
        lines (rep (onsets [i], 2) + spacing [j],
               onset.se$asynchrony [idx] * c (-1, 1) + onset.mean$asynchrony[idx],
               col = colors [j])
    }
}

for (i in c (11, 15, 25, 41)) {
    abline (v = onsets [i], col = "#00000080")
}
abline (v = 0, col = "#00000080", lty = 2)
abline (v = mean (onsets [c (25, 41)]), col = "#00000080", lty = 2)
```




```
dev.copy2pdf (file = "figures/asynchrony-notes-who.pdf", width = 12, height = 5)
```

```
## pdf
##   2
```

Plot the same using skewness-corrected data of asynchrony

```r
# ## Compute aggregate means and standard errors (for ploting on y axis)
# onset.data$asynchrony <- y
# onset.mean <- aggregate (asynchrony ~ who + time + note, data = onset.data, mean)
# se <- function (x) { sd (x) / sqrt (length (x)) }
# onset.se <- aggregate (asynchrony ~ who + time + note, data = onset.data, se)
#
# ## Plot results: points (time,asynchrony)
# spacing <- 0.16 * c (0.1, -0.1)
# colors <- c ("red", "blue")
# plot (onset.mean$time + spacing , onset.mean$asynchrony, pch = 19,
#       col = colors [onset.mean$who], las = 1, xlab = "time (s)",
#       ylab = "|aysnchrony| (s)")
# #ylim = c (0.01, 0.07)
#
# ## Plot results: standart errors lines upwards & downwards on points (time,asynchrony)
# condition <- c ("other", "self")
# for (i in 1 : 41) {
#     for (j in 1 : 2) {
#         idx <- which (onset.mean$note == i & onset.mean$who == condition [j])
#         lines (rep (onsets [i], 2) + spacing [j],
#                onset.se$asynchrony [idx] * c (-1, 1) + onset.mean$asynchrony[idx],
#                col = colors [j])
#     }
# }
#
# for (i in c (11, 15, 25, 41)) {
#     abline (v = onsets [i], col = "#00000080")
# }
# abline (v = 0, col = "#00000080", lty = 2)
# abline (v = mean (onsets [c (25, 41)]), col = "#00000080", lty = 2)
#
# # return to uncorrected data
# onset.data$asynchrony <- abs((onset.data$secondo.onset - onset.data$primo.onset)) / 44100
#
# dev.copy2pdf (file = "figures/corrected-asynchrony-notes-who.pdf", width = 12, height = 5)
```

A generalized linear mixed model (GLMM) [8] was fitted to the transformed data using the 3 fixed factors:

- WHO - a discrete factor with levels “self” and “other”;
- TAKE - a continuous factor going from 1, the first take, to 4, the last take;
- CLASS - the class of each note with two levels: “first” being the notes at each strong beat, when present, and “nonfirst”, the other notes).

Three random factors where also considered:

- PRIMO - the influence of the primo performer;
- SECONDO - the influence of the secondo performer;
- TIME/SECONDO - the coefficient of variation of the asynchrony during the excerpt.

First, a simple statistical test using the whole duration of the execution was used to insure that the clarinetists were really trying to follow the primo execution and not simply playing from memory what they played as primo themselves.

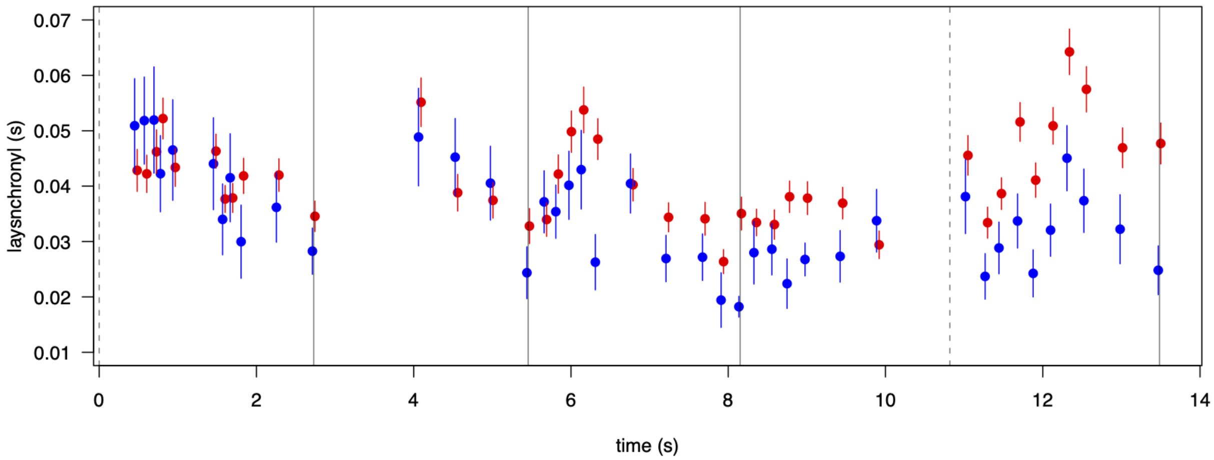


Figure 3: Fig. 5: Asynchrony values in the *other* situation (in red), compared to the *self* situation in (blue).

Dissecando a análise (author: Davi Mota)

O bloco inicial cria a estrutura de dados =onset.data= que será utilizada na função =lmer=. Também são criados os rótulos para as condições /self/ e /other/ utilizando a função =factor= que codifica o vetor nos níveis sugeridos.

```
library (lme4)
```

```
## Loading required package: Matrix

## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car
```

```
library (multcomp)
```

```
## Loading required package: mvtnorm

## Loading required package: survival

## Loading required package: TH.data

## Loading required package: MASS

##
## Attaching package: 'TH.data'

## The following object is masked from 'package:MASS':
##
##     geyser
```

```
onset.data <- read.table ("raw_data", header = FALSE)
names (onset.data) <- c ("primo", "secondo", "composer", "take", "note",
                         "primo.onset", "secondo.onset")
```

```r
onset.data$signed.asynchrony <- abs (onset.data$secondo.onset
                                   - onset.data$primo.onset) / 44100
who <- rep ("other", nrow (onset.data))
idx <- which (onset.data$primo == onset.data$secondo)
who [idx] <- "self"
onset.data$who <- factor (who)
```

O bloco seguinte aplica a função =lmer= aos dados.

```r
fm0 <- lmer(signed.asynchrony ~ who + take + (1 | primo) + (1 | secondo),
            data = onset.data)

K <- rbind (c (1, -1, 0), c (0, 0, 1))
(ci <- confint (glht (fm0, linfct = K)))
```

```
##
##   Simultaneous Confidence Intervals
##
## Fit: lmer(formula = signed.asynchrony ~ who + take + (1 | primo) +
##     (1 | secondo), data = onset.data)
##
## Quantile = 2.2323
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##        Estimate   lwr        upr
## 1 == 0  0.0544623  0.0446585  0.0642661
## 2 == 0 -0.0012025 -0.0020872 -0.0003179
```

Este modelo não apresenta resultados condizentes com os apresentados no ICPA, mesmo realizando a transformação Box-Cox dos dados.

Efeitos fixos

Acredito que na realidade varios modelos foram ajustados às diferentes questões abordadas. Ao analisarmos o código vemos que esta abordagem foi aplicada aos efeitos aleatórios. Isto é um procedimento documentado (e recomendado) em diversas publicações que tratam do tema. Acredito que esta abordagem foi também utilizada nos efeitos fixos. Para confirmar esta hipótese, ajustaremos um modelo para os subgrupos com /self/ e /other/.

```r
self.data <- onset.data[which(onset.data[,9]=="self"),]
fixef(lmer(signed.asynchrony*1000 ~ take + (1 | primo) + (1 | secondo),
           data = self.data))[2]
```

```
##       take
## -0.4060044
```

```r
other.data <- onset.data[which(onset.data[,9]=="other"),]
fixef(lmer(signed.asynchrony*1000 ~ take + (1 | primo) + (1 | secondo),
           data = other.data))[2]
```

```
##      take
## -1.397461
```

Ao analisarmos os resultados dos ajustes dos modelos acima, podemos ver que os /slopes/ são coerentes com os resultados apresentados no ICPA: decrescimo de =0.4060049 ms= para a condição /other/ e decrescimo de =1.397458 ms= para a condição /self/. É interessante notarmos aqui a similaridade entre os resultados dos fatores fíxos e os resultados de uma simples regressão linear:

```
lm(signed.asynchrony*1000 ~ take, data = self.data)
```

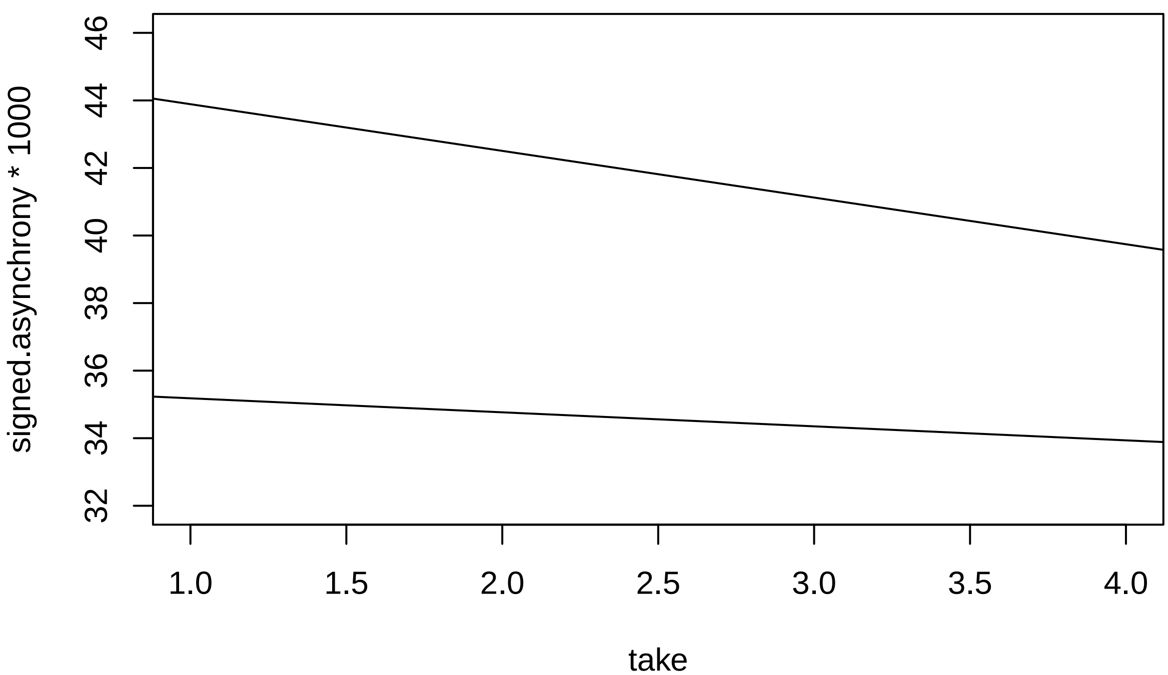
```
##
## Call:
## lm(formula = signed.asynchrony * 1000 ~ take, data = self.data)
##
## Coefficients:
## (Intercept)         take
##     35.5967      -0.4152
```

```
lm(signed.asynchrony*1000 ~ take, data = other.data)
```

```
##
## Call:
## lm(formula = signed.asynchrony * 1000 ~ take, data = other.data)
##
## Coefficients:
## (Intercept)         take
##      45.271       -1.382
```

O código acima nos apresenta /slopes/ de =-0.4152= para a condição /self/ e =-1.382= para a condição /other/.

```
plot(signed.asynchrony*1000 ~ take, type = "n", data = other.data, ylim = c(32,46))
abline(lm(signed.asynchrony*1000 ~ take, data=other.data))

# points(signed.asynchrony*1000 ~ take,data = self.data)
abline(lm(signed.asynchrony*1000 ~ take, data=self.data))
```



Efeitos aleatórios

A anlaise dos efeitos aleatórios segue o padrão proposto por grande parte dos analistas, que é confrontar modelos de diferentes complexidades para verificar a relevância de determinado termo para o ajuste. No bloco seguinte, os efeitos fixos são excluídos do modelo =fm=, restando apenas os efeitos aleatórios. Os modelos =fm1= contém apenas o efeito /primo/, e o modelo =fm2= contém e efeito /secondo/. ANOVA é utilizado para confrontar o modelo original mais complexo =fm0= (que contém todos os efeitos) e os modelos mais simples.

```r
fm <- lmer(signed.asynchrony ~ 1 + (1 | primo) + (1 | secondo), data = onset.data)
fm1 <- lmer(signed.asynchrony ~ 1 + (1 | secondo), data = onset.data)
(anova (fm0, fm1))
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: onset.data
## Models:
## fm1: signed.asynchrony ~ 1 + (1 | secondo)
## fm0: signed.asynchrony ~ who + take + (1 | primo) + (1 | secondo)
##     npar    AIC    BIC logLik deviance  Chisq Df Pr(>Chisq)
## fm1    3 -21248 -21228  10627   -21254
## fm0    6 -21339 -21300  10676   -21351 97.406  3  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
fm2 <- lmer(signed.asynchrony ~ 1 + (1 | primo), data = onset.data)
(anova (fm0, fm2))
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: onset.data
## Models:
## fm2: signed.asynchrony ~ 1 + (1 | primo)
## fm0: signed.asynchrony ~ who + take + (1 | primo) + (1 | secondo)
##     npar    AIC    BIC logLik deviance  Chisq Df Pr(>Chisq)
## fm2    3 -21044 -21024  10525   -21050
## fm0    6 -21339 -21300  10676   -21351 301.58  3  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Os resultados indicam que o modelo mais complexo possue um melhor ajuste, portanto os termos excluídos nos modelos mais simples são relevantes para o ajuste.

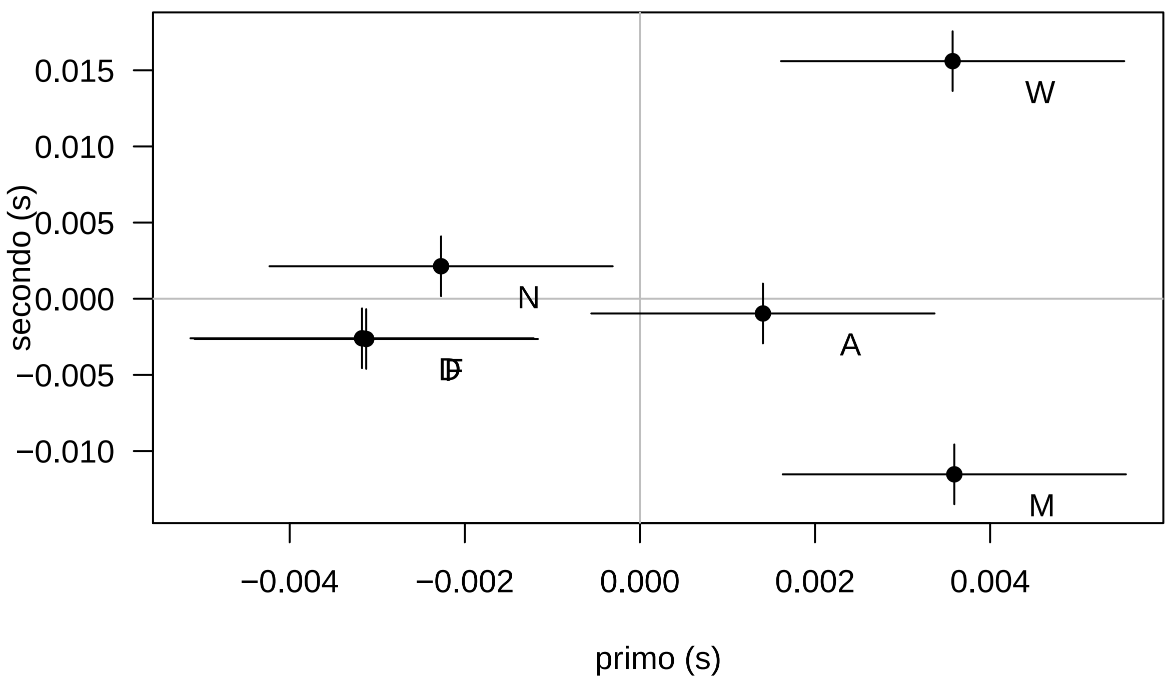
O resto é plot dos efeitos aleatórios …

```r
rand.eff <- ranef (fm, postVar = TRUE)
```

```
## Warning in ranef.merMod(fm, postVar = TRUE): 'postVar' is deprecated: please use
## 'condVar' instead
```

```r
rand.eff.primo <- rand.eff$primo [, 1]
## v <- sqrt (attributes (rand.eff.primo)$postVar [ , , 1 : 6])
v <- .001
rand.eff.primo.min <- rand.eff.primo - qnorm (0.975) * v
rand.eff.primo.max <- rand.eff.primo + qnorm (0.975) * v

rand.eff.secondo <- rand.eff$secondo [, 1]
## v <- sqrt (attributes (rand.eff.secondo)$postVar [ , , 1 : 6])
v <- .001
```

```
rand.eff.secondo.min <- rand.eff.secondo - qnorm (0.975) * v
rand.eff.secondo.max <- rand.eff.secondo + qnorm (0.975) * v

plot (c (rep (rand.eff.primo, 2), rand.eff.primo.min, rand.eff.primo.max),
      c (rand.eff.secondo.min, rand.eff.secondo.max, rep (rand.eff.secondo, 2)),
      type = "n", xlab = "primo (s)", ylab = "secondo (s)", las = 1)
abline (v = 0, col = "gray")
abline (h = 0, col = "gray")
points (rand.eff.primo, rand.eff.secondo, pch = 19)
for (i in 1 : 6) {
  lines (c (rand.eff.primo.min [i], rand.eff.primo.max [i]),
         rep (rand.eff.secondo [i], 2))
  lines (rep (rand.eff.primo [i], 2),
         c (rand.eff.secondo.min [i], rand.eff.secondo.max [i]))
}
text (rand.eff.primo + 1e-3, rand.eff.secondo - 2e-3,
      labels = row.names (rand.eff$primo))
```




```
dev.copy2pdf (file = "figures/random-effects-signed-asynchrony.pdf", width=5, height=5)
```

```
## pdf
##   2
```

The GLMM fit showed that the mean asynchrony is around 30 ms with a statistically significant drop of 6 ms when performers played with themselves. This is coherent with results found elsewhere. No significant TAKE effect was found, probably because the musicians could learn how to accompany from hearing the preliminary primo recording.

A zoomed plot for the fitted values is shown in figure [2], together with the standard error around the estimated values. The difference between the self and other asynchronies for take #1 is of 9.9 ms. This value is significantly different from zero (95% confidence interval: [6.2, 13.7], $p < 0.0001$). For the "other" condition, there is a significant change of the asynchrony with the take, with a decrease of 1.4 ms per take (95% confidence interval: [0.6, 2.3], $p < 0.01$). The change of asynchrony with take is not significant for the “self” condition, although there is a slight decrease of 0.4 ms per take.

All three random effects were significant, what demonstrates that, first, some performers are easier to follow than others, second, some performers are better followers than others and, third, performers behave differently as regards the adaptation of the synchronization during the excerpt.

The significance of the random effects were tested using a likelihood ratio test. Both the primo (Chisq [1] = 31.4, $p < 0.0001$) and the secondo (Chisq [1] = 258.5, $p < 0.0001$) random factors are significant. The fitted values for each of the six performs are shown in figure [3]. The horizontal axis show the primo effect and the vertical axis show the secondo effect. The individual values of the primo and second random effects have the following interpretation: a performer that has a positive value for the primo effect induces a increase of asynchrony on the other performers, i.e. it is difficult to follow him (or her). The performer who has a negative value for the secondo random effect is one who can better follow the primo.

Another interesting result is the fact that a significant CLASS effect was found. Asynchrony is about 5 ms lower for the notes at strong beats. This opens the avenue for analyzing our data under the prospects of the dynamics of attending, as well as for understanding how the synchronization patterns may arise from the musical structure of the excerpt [9,10].

References

[1] Keller PE, Knoblich G, & Repp BH (2007). Pianists duet better when they play with themselves: on the possible role of action simulation in synchronization. Conscious Cogn, 16(1):102–111.

[2] Keller PE & Appel M (2010). Individual differences, auditory imagery, and the coordination of body movements and sounds in musical ensembles. Music Percep 28:27–46.

[3] Keller PE (2001) Attentional resource allocation in musical ensemble performance. Psychol Music, 29:20–38.

[4] Knoblich G, Seigerschmidt E, Flach R, and Prinz W (2002). Authorship effects in the prediction of handwriting strokes: evidence for action simulation during action perception, Q J Exp Psychol A 55(3):1027–1046

[5] Knoblich G & Flach R (2001). Predicting the effects of actions: Interactions of perception and action, Psychol Sci 12:467–472.

[6] Mauricio Alves Loureiro; Hani Camille Yehia; Hugo Bastos de Paula; Thiago de Almeida Magalhães Campolina; Davi Alves Mota.Content Analysis of Note Transitions in Music Performance. 6th Sound and Music Computing Conference (SMC 2009), 2009, Porto. Proceedings of the 6th Sound and Music Computing Conference (SMC 2009). Porto, Portugal: INESC Porto, 2009. p. 355-359. ISBN: 978-989-95577-6-5. Julho de 2009.

[7] Box GEP & Cox DR (1964). An analysis of transformations,J Roy Stat Soc B Met, 26(2):211–252.

[8] Bates D & Maechler M (2010) lme4: Linear mixed-effects models using S4 classes, R package version 0.999375-37.

[9] Barbara Tillmann, Catherine Stevens, and Peter E. Keller. Learning of timing patterns and the development of temporal expectations. Psychol Res, 2010.

[10] Large EW & Jones MR (1999) The dynamics of attending: How people track time-varying events, PsycholRev, 106(1):119–159.