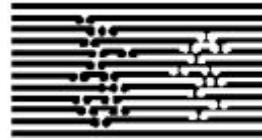




UNIVERSITY OF GENOA
Faculty of Engineering



Department of
Communication,
Computer and
System Sciences



InfoMus Lab
Laboratory of
Musical Informatics

Ph. D. DISSERTATION

On the recognition of expressive
intentions in music playing:
A computational approach with
experiments and applications

Roberto Dillon

Supervisor: Prof. Antonio Camurri

“Io la Musica son, ch'ai dolci accenti
So far tranquillo ogni turbato core,
Et or di nobil ira et or d'amore
Posso infiammar le più gelate menti”

Alessandro Striggio for the libretto of
Claudio Monteverdi “Orfeo” (Mantua, 1607)

Table of Contents

Introduction	3
Acknowledgements	3
<u>Part I: How to extract data from music playing</u>	
1.1: Expressivity in Music: how can we measure it?	5
1.2: Choosing and extracting the audio cues	6
1.3: Monophonic and polyphonic music	7
1.4: Audio cues in EyesWeb: time window and event triggered approach	8
<u>Part II: Recognition of Expressive Intentions</u>	
2.1: Real Time tracking of expressive intentions on recorder	13
2.2: Recognition of expressive intentions on violin	18
<u>Part III: Who is playing?</u>	
3.1: A slightly different problem: who is playing?	26
3.2: Glenn Gould, Maria Joao Pires and Director Musices play Mozart	26
<u>Part IV: Detection of Arousal</u>	
4.1: Another different problem: what is “arousal”?	34
4.2: Arousal in Bach Solo Violin Sonatas	34
<u>Part V: Conclusions and future developments</u>	
5.1: Possible uses of cues in actual music making	50
<u>Appendix</u>	
A: The EyesWeb Blocks in detail	51
B: Basic EyesWeb patches for interactive performances	58
<u>References</u>	62

Introduction

Can *perceived emotions* in music be somehow explained and objectively measured?

This thesis tries to shed some light on this difficult question by proposing a possible approach for understanding and quantifying *expressive intentions* in music playing by extracting meaningful information from note events. This will be done by looking only at the actual sound, without any data coming from other tools such as, for example, MIDI devices which, although useful in several aspects, can only approximate most of notes characteristics that are relevant to us.

The proposed approach was developed in the context of the European EU-IST Project MEGA (Multisensory Expressive Gesture Applications) no. IST-1999-20410 and the basic ideas were developed in close collaboration with Dr. Anders Friberg during my stays at the Royal Institute of Technology (KTH) in Stockholm as a guest researcher during 2001 and 2002.

A strong emphasis is given to *real-time* applications by developing and using a set of libraries integrated in the EyesWeb open platform (a software developed at the Laboratory of Musical Informatics of the University of Genoa) and the proposed system is tested in several experiments ranging from *expressive intentions* recognition to *playing style* recognition exploiting excerpts played by well known professional musicians on different instruments (recorder, violin, piano).

Acknowledgements

This thesis was made possible thanks to the fundamental help of my supervisor, Prof. Antonio Camurri (DIST), of Prof. Johan Sundberg, who was for me like a second tutor, and of Dr. Anders Friberg (KTH – Stockholm), who helped me in all the stages of the research with invaluable suggestions.

I'd like also to thank all the staff of the Speech, Music and Hearing department at KTH - Stockholm, for their help during my stays there, Dr. Paolo Coletta (DIST – Genova), for debugging the EyesWeb libraries, and all the members of the “Laboratorio di Informatica Musicale” for their constant feedback and fruitful comments.

I am highly indebted with the professional musicians who agreed to take part to my experiments, showing great patience, interest and kindness: Tanja Becker-Bender, Lorenzo Cavasanti and Fabrizio Ferrari.

Last but not least, I will be forever grateful to my parents who fully supported me during all these years of study.

Part I

How to extract data from music playing

1.1: Expressivity in Music: how can we measure it?

One of the main findings from the research aimed at studying music performance is that the actual performance of a piece of music never corresponds to the nominal values of the notes printed in the score.

Moreover the differences between a musical performance of the same piece by a world renowned soloist and a music student are striking to all of us: one is able to “move” us deeply, even to move us to tears or to produce shivers in our back, while the other has almost no effects on us.

A bad, although formally correct, performance like that of a good student is often referred to as “cold”, but what does this mean exactly? Can we somehow quantify what a “cold” performance is and compare it objectively to a memorable one?

This problem got a rising interest during the last 15-20 years in an interdisciplinary environment, made of psychologists, cognitive musicologists and computer engineers, since getting insights on this matter would have several important aspects: from having a better understanding on how the human mind perceives and elaborates external stimuli to the possibility of making new and more natural computer music systems.

Several studies, such as (Sundberg et al. 1991; Juslin 2000), concentrated on the study of sonological parameters involved in expressive performance and our research follows their footsteps.

The basic idea is to determine a set of parameters, that we will refer to as *audio cues*, that can somehow give a detailed description of the actual performance so as to quantify all “the small and large variations in timing, dynamics, timbre and pitch that form the microstructure of a performance and differentiate it from another performance of the same music” (Palmer 1997, p.118).

Once we can quantify these variations, we may have the key to understand where the “emotional impact” of the performance lies, as suggested by (Gabrielsson, 1995; Juslin, 1997).

1.2: Choosing and extracting the audio cues

Our approach starts from a very simple idea already proposed by Neil Todd at the beginning of the 90s (Todd 1992). This idea is to extract event information from a music input signal simply by low pass filtering it, exactly like if we were working with a PAM signal.

So, filtering the signal with filters having different cut off frequencies will provide us with profiles bearing different information: a cut off frequency of about 20Hz will extract the envelope of the signal (we will call this *note profile* since it shows fast events) while reducing the cut off at around 1 Hz or less will simply take the energy of the signal (we will call this *phrase profile* since it shows an average behaviour that changes slowly).

Then, if we compare the two profiles extracted in this way as shown in figure 1, we can divide the performance in several *events* that are detected when the envelope gets higher values than the energy.

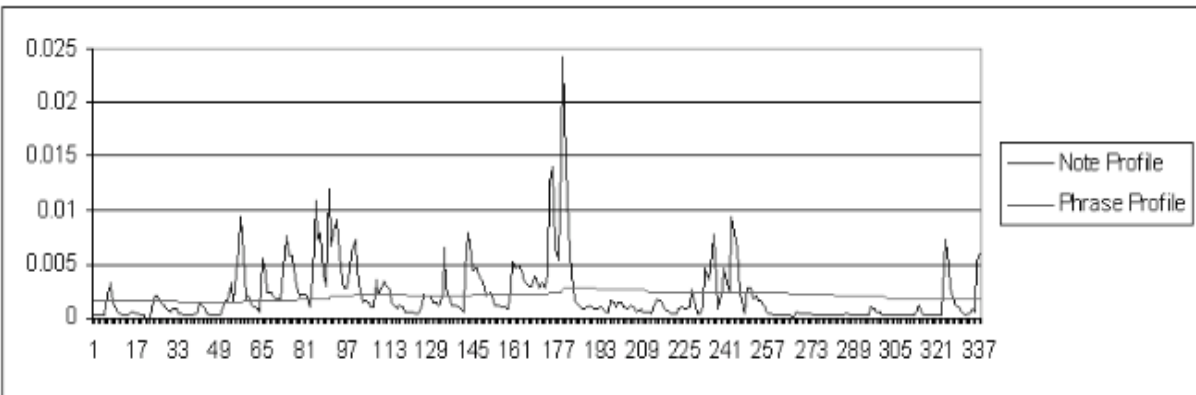


Fig.1: profiles obtained by low pass filtering a music file with different cut off frequencies (Y axis: amplitude, X axis: time)

From each of these events we can get insights on what is happening by analysing their shape and the time that occurs in between them so as to collect valuable information regarding several aspects:

- **Tempo** (how many events do we have per unit of time? How long are the detected events?)
- **Articulation** (the ratio between the event duration [*DR*] and the time that occurs between its onset and the one of the following event, [*IOI*])
- **Dynamics** (how loud are the detected events?)

By studying these aspects we will be able to extract several sets of audio cues, suitable for different experiments (as shown, for example, in Dillon 2001, Friberg et al. 2002), which will be explained in detail during the following chapters.

Anyway, before proceeding further, we should better understand the meaning of the two profiles in the case that monophonic or polyphonic music is being played and analysed.

1.3 Monophonic and polyphonic music

In the case a monophonic piece is running on, for example on instruments such as flute or violin, the interpretation of the profiles is quite straightforward: ideally, they should identify each note being played.

But what happens if the instrument is a piano, playing a complex polyphonic piece? What are we extracting the cues from?

To understand this, let's have a look at figures 2 and 3:

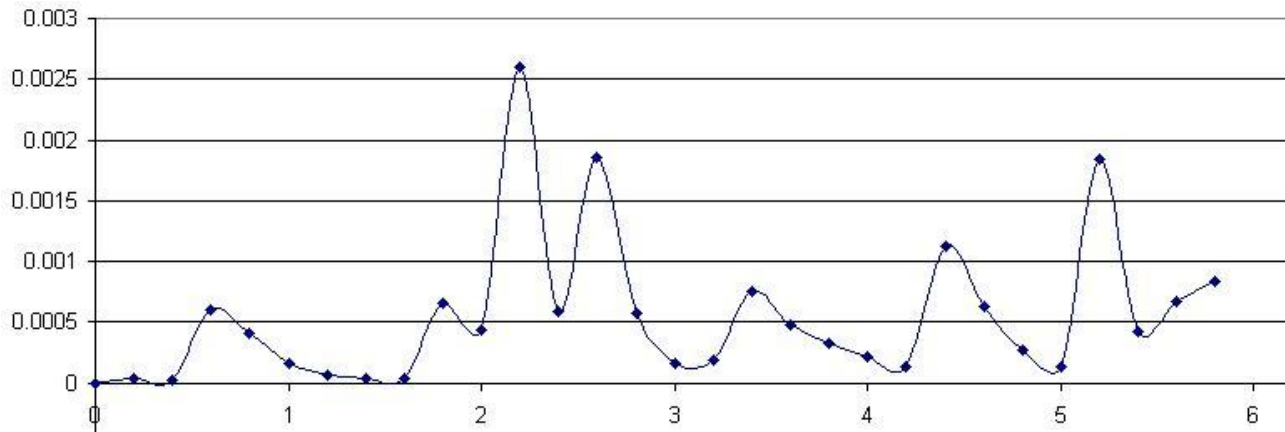


Fig.2: Piano music profile extracted by low pass filtering with a cut off frequency of 20 Hz

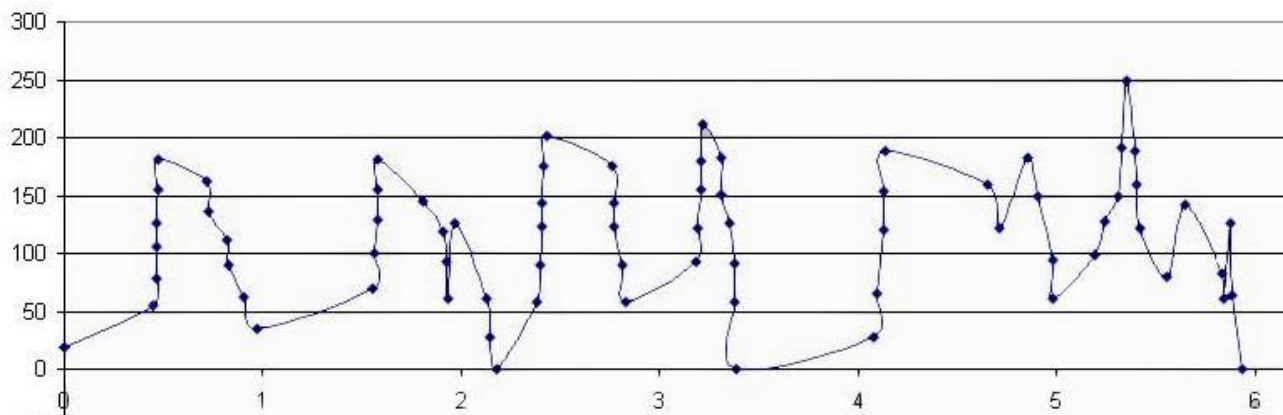


Fig.3: Same piano excerpt as in fig.2, obtained by adding the notes' key velocities of the corresponding MIDI performance

These figures show the same music excerpt, taken from Scriabin Etude Op.8 n.11 as used in (Casazza, Pertino 2003).

Figure 2 shows the extracted envelope after low pass filtering with a cut off filter of 20 Hz while figure 3 is a profile obtained from the corresponding MIDI file by summing all the key velocities of the notes being played at time t :

$$y(t) = \sum_{i=1}^{notes} KeyVelocity_i \quad (1)$$

As we can see, the two profiles are basically similar since they are showing the same events. It should be noted that, in this case, the envelope extracted by low pass filtering is not representing the single notes but full chords or notes overlapping on each other, nonetheless it is still a good picture, useful to point out the most important musical events on which we will concentrate our study.

1.4: Audio cues in EyesWeb: time window and event triggered approaches

To analyse input music performances in real time we developed a set of libraries, whose blocks will be explained in detail in Appendix A, running on the EyesWeb Open Platform.

EyesWeb is a software platform similar in conception to well known softwares like Max/MSP and PD and it was developed at the Laboratory of Musical Informatics of the University of Genoa. It allows the user to build patches made of simple blocks to perform complex operations and algorithms on audio and video signals in real time.

It can be freely downloaded from <http://infomus.dist.unige.it> and an official introduction can be found in (Camurri et al. 2000; Camurri et al. 2001).

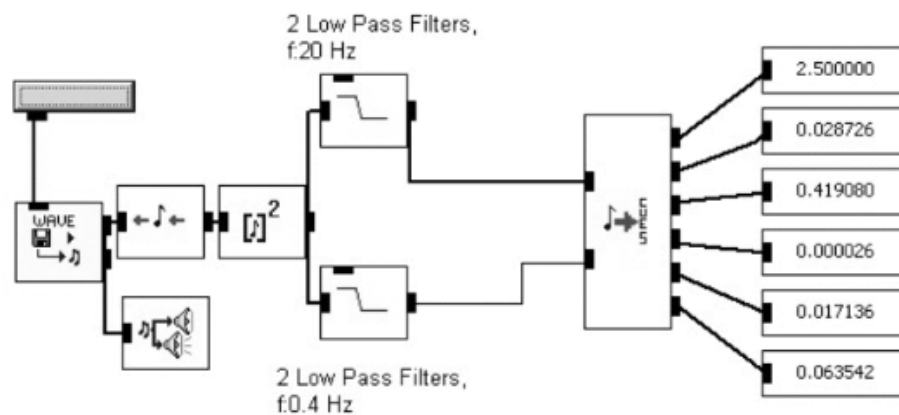


Fig. 4: A simple EyesWeb patch for extracting audio cues, time window approach

In figure 4 we have an example patch that extracts a set of cues.

It takes an input stream (it can be from a previously saved file or a live input from a microphone) and stores the last few seconds of music. This is a *time window* that is taken for analysis purposes and, in the upcoming experiments, this time interval was usually set to 4 seconds (unless otherwise stated) since it seemed a good compromise between the short time needed by the system to react in real time and the longer time required to understand what is going on from a musical point of view.

The time frame is updated n times per second (by the first block after the input wave reader in fig.4. In the following experiments, n will be set to 5, unless otherwise stated) so as to smoothly move it across the incoming performance. This overlapping between two following buffers allows an almost continuous analysis of the incoming signal, useful to appreciate any variations in the characteristics of the performance under analysis (such as following the development of a crescendo, etc.).

Once the time window is filled, the signal is squared and then low pass filtered as described earlier.

The filters used need to have zero ripple and, after experimenting with several ones, we decided for using a first order IIR filter, derived from those commonly used for “exponential averaging” of numerical series, so as to be able to enhance recent events and also to show long term behaviour. Its output at time n is:

$$Y_n = A_0 X_n + A_1 Y_{n-1} \quad (2)$$

Where

$$A_0 = 1 - A_1$$

$$A_1 = -\left(a - \frac{1}{a} + 1\right)$$

$$a = \tan\left(p \frac{f_0}{f_s}\right)$$

f_0 : *cutoff* – *frequency*

f_s : *sampling* – *frequency*

Then we can feed the two profiles to the cues extractor block. This block is able to extract several cues by analysing the signal framed by the time window. In particular we can get:

- **Tempo 1**, defined as the average Note Duration (DR), in seconds, of the notes in the buffer
- **Tempo 2**, defined as the number of events detected in the current buffer
- **Articulation**, defined as Actual Note Duration / Inter Onset Interval averaged across the events contained in the buffer
- **Standard Deviation of Articulation**, computed over the events contained in the buffer
- **Mean Sound Level** of the events contained in the buffer where *Sound Level* is the amplitude measured at the beginning of the event, i.e. at the intersection between the two profiles¹
- **Standard Deviation of Sound Level** of the events contained in the buffer where *Sound Level* is the amplitude measured at the beginning of the event, i.e. at the intersection between the two profiles
- **Mean Sound Level Difference**, where *Sound Level Difference* is defined as the difference between the current and the preceding event. This cue offers useful information for identifying local crescendo/decrecendo effects.
- **Mean Attack Velocity** of the events contained in the buffer. *Attack Velocity* is defined as the derivative of the note profile at the intersection with the phrase profile

¹ This value comes from the WAVE file in input and is always a number $|x| < 1$. If needed, it can be converted in dB by this simple formula: $y = 4.34 \ln(x)$

It should be noted that not all the cues here listed are as useful in all possible applications as expected and that some of them are more appropriate than others to face different problems, so in the experiments explained in the following chapters, we will use only those giving the best results.

The *time window* approach is particularly suitable for applications where a few seconds delay is acceptable and for studying problems where we want to conduct a statistical analysis of the data extracted.

In fact, by taking values referring to an average across numbers extracted in a buffer, we automatically satisfy the hypothesis of the Central Limit Theorem, regardless of the distribution of the original samples and hence we can simplify several assumptions by knowing that the distributions of the extracted data are Gaussians (for a comprehensive explanation of this fundamental theorem of probability theory see, for instance, (Papoulis 1991)).

Anyway, there can be applications where having a few seconds of delay can be unacceptable. In this case the system analysis should be *event triggered*.

A possible EyesWeb patch is shown in Fig.5. Here, after squaring and low pass filtering, the sound data are converted to ASCII values and then fed to the extractor block.

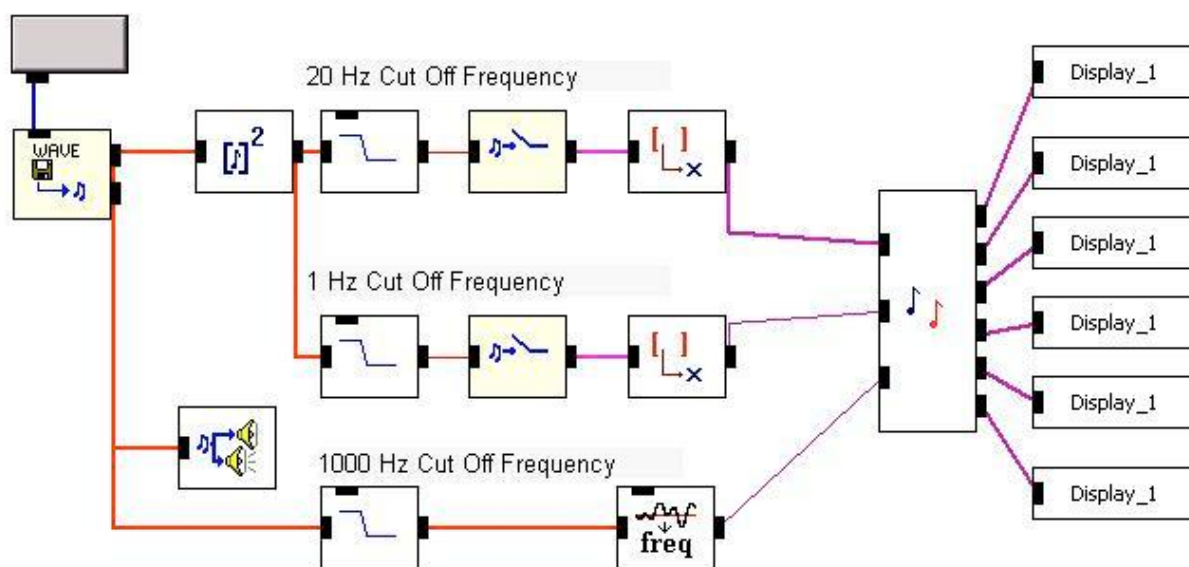


Fig.5: an EyesWeb patch for extracting cues each time a new event is detected

This approach has a couple of drawbacks, though: first, as long as the system doesn't detect a new event, it doesn't produce any output (for example, during a pause or a very long note this patch will be "frozen" while with the other approach we will always get some output that we could use for starting other processes). Second, the two-profiles comparison can skip some notes (especially if the performer plays with a high legato articulation) and, since here we are interested in every single event, this can be critical.

Anyway, to correct the latter problem, a further input to the cues extraction block can be added, as shown in figure 5: it takes in input the midi value of the signal being played (converted from a basic fundamental frequency extractor block, which works with a zero-crossing algorithm once the signal has been low pass filtered with a cut off frequency of, for example, 1000 Hz).

With this added information the cues block can detect musical events also if the musician is playing all legato.

In this case the extracted cues provided by the cue block are as follows:

- ***Tempo***, defined as the *Inter Onset Interval* between the current note and the preceeding one
- ***Onset Sound Level***, taken at the intersection of the two profiles or when MIDI value changes
- ***Max Sound Level***, the maximum value detected in the last event
- ***Articulation***, defined as the ratio between the DR of the last event and its IOI with the current one
- ***Attack Velocity***, the derivative of the note profile at the intersection with the phrase profile or when the MIDI value changes

Part II

Recognition of Expressive Intentions

2.1: Real Time tracking of expressive intentions on recorder²



Fig. 6: Excerpt from Arcangelo Corelli (1653 – 1713) Sonata Op.5 n.8

In the first of our experiments we will use the cues to track in real time the expressive intention as played on recorder by an international level soloist (maestro Lorenzo Cavasanti).

The artist was asked to perform the same music excerpt (shown in figure 6) trying to express a set of predefined emotional expressiveness. These were chosen from a palette of many possible expressive intentions (already used in experiments by several researchers, e.g. (Battel, Fimbianti 1998), (Canazza et al. 1998), (De Poli et al. 1998)) and, in particular, we selected the following so as to have an emotional space with a few well defined contrasting moods:

- Neutro (neutral)
- Passionale (passionate)
- Cupo (dark)
- Agitato (restless, hectic)
- Brillante (glittering)

Each expressive intention was recorded three times (using a sample frequency of 22050 Hz, 16 bit resolution) and then, for analysis purposes, we selected the best one accordingly with the player.

The analysis was carried out with the *time window* approach (window size: 4 seconds, updated 5 times per second) using the following cues:

- Tempo 2 (in the following table labeled simply as “tempo”)
- Mean of Articulation (“MArt”)
- Standard Deviation of Articulation (“SDArt”)
- Mean of Sound Level (“MSnd”)
- Standard Deviation of Sound Level (“SDSnd”)
- Mean of Attack Velocity (“MAttVel”)

² Preliminary results derived from previous research on this topic were presented in (Camurri, Dillon, Saron, 2000) and (Dillon, 2001)

By using these cues, each time frame of the performance is described by a 6 component vector and this allows us to study the different moods by looking at this new data set and hence to define mapping strategies so as to give a graphical idea of the perceived emotion.

In other papers, such as (Canazza et al. 1998) a graphical 2D space was organized by means of a factor analysis on the adjectives but here we wanted to experiment with a new simple algorithm whose results, despite its simplicity, shown very interesting analogies with more computationally complex approaches.

Our multidimensional scaling algorithm defines, starting from an N dimensional array, a bidimensional space where each expressive intention finds its place, determined in an off-line elaboration, so as to provide a set of predefined points. Then we can move between these points during a real time performance and see which is the intention we are getting closer to.

To define this plane, for each cue in each mood, the average value over the whole performance was computed so as to have a global set of identifying parameters, then they were scaled to get all the values in the same magnitude order (the results are shown in table 1):

	Tempo	Mart	SDArt	MSnd	SDSnd	MattVel
Agitato	190	48	63	43	12	49
Brillante	135	48	157	24	31	44
Cupo	85	62	93	33	96	24
Neutro	100	69	83	44	99	47
Passionale	123	61	71	58	107	59

Table 1: averages values for each cue over the whole performances

Now, the basic idea is to recursively reduce the dimensions of the emotional space by taking two axis at a time and reduce them to one as long as we don't reach the desired compression rate.

In particular we proceed as follows:

1. Select two cues and plot all the states as function of these parameters only
2. Plot the lines which join the origin with the two outer expressive intentions (i.e. those with smallest and largest angular coefficient) so as to define an angle α that spans all the plotted states
3. Plot the bisecting line η for the angle α ; η is at an angle ϕ with the X axis
4. Project all the expressive intentions on η : for each of them, the new value simply is the sum of old values multiplied by $\text{Cos } \phi$ and $\text{Sin } \phi$
5. Go back to step 1 until we get the desired compression rate.

In our case, we compressed all the tempo related cues (*tempo, mean of articulation, standard deviation of articulation*) on the X axis while those related to dynamics (*mean of sound level, standard deviation of sound level and mean of attack velocity*) were assigned to the Y axis, as shown by the schema presented in figure 7.

Figure 8, instead, shows an example of the compression algorithm for reducing the mean and standard deviation of articulation to one value.

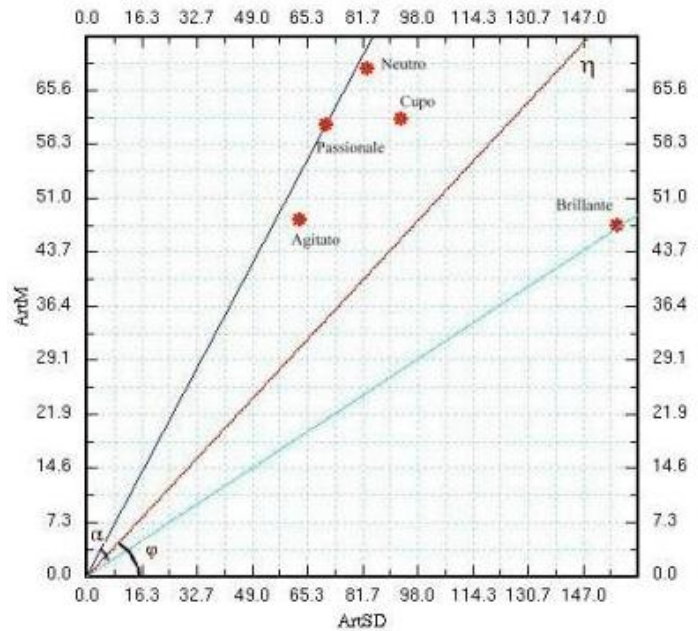
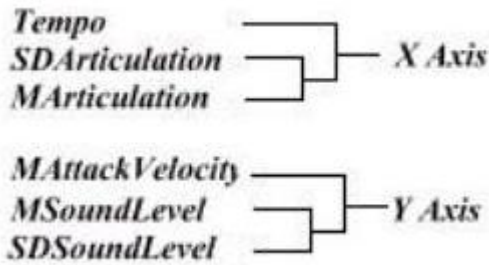


Fig.7: compressing the 6D space to 2D

Fig.8: an example of the compression procedure

Of course the choice of the compression order, in principle, may affect the final plot (it's like changing the weights in a sum) and this influence gets stronger as the number of original components increases. However, since in this experiment we had only a couple of compressions on each axis, the order didn't show to be relevant to the final graph which is showed in figure 9.

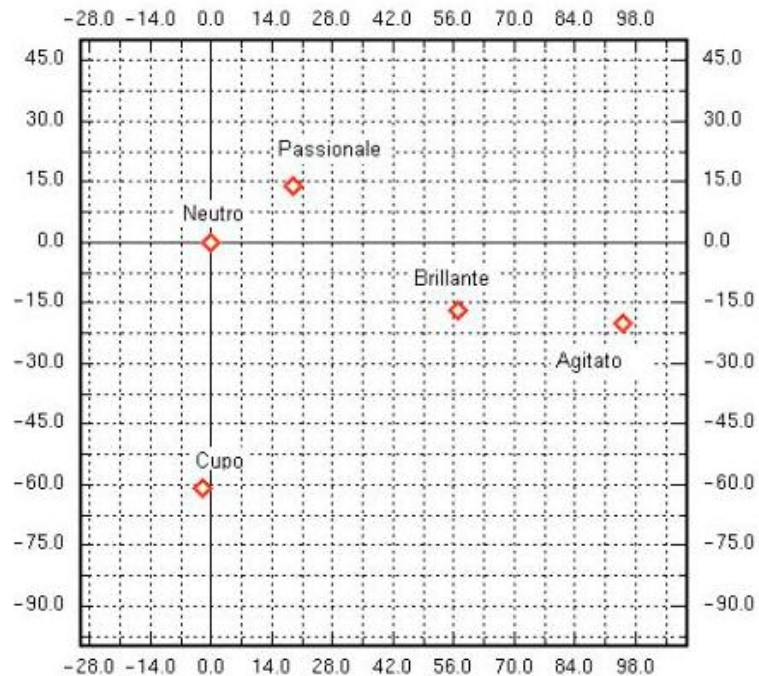


Fig.9: the 2D compressed emotional space

Now it is very interesting to compare this output with the well know Sammon multidimensional scaling algorithm. This algorithm (for a detailed description see (Sammon 1969)) tries to approximate the original distances between vectors in a N-dimensional space by finding a new set

of points in the Euclidean one. This is accomplished by optimizing a cost function (3) which indicates how much precisely the original distances should be approximated:

$$E_s = \sum_{k \neq l} \frac{[d(k,l) - d'(k,l)]^2}{d(k,l)} \quad (3)$$

Where $d(k,l)$ is the distance between vectors X_k and X_l in the original space while $d'(k,l)$ is the distance between vectors X'_k and X'_l in the compressed space.

Reducing the expressive space with this algorithm gives the output shown in figure 10:

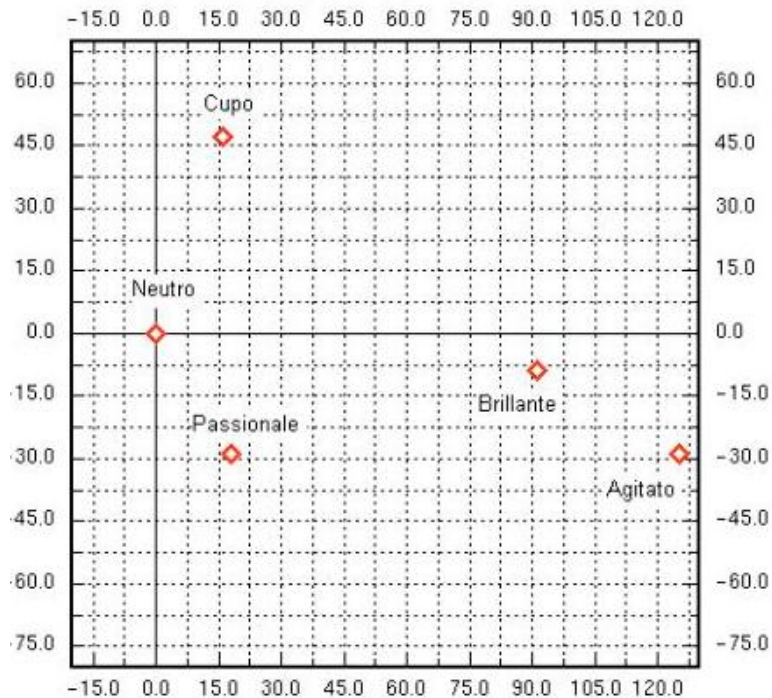


Fig.10: compressing the expressive space with Sammon algorithm

As we can see, the two outputs are strikingly similar: even if *Passionale* and *Cupo* are mirrored on the X axis, the important thing to point out is that the ratios between the distances among the various points are very close to each other in both plots.

This shows our algorithm to be an interesting tool to approximate the Sammon's. In fact, due to its computational complexity and lack of generability (adding a new point forces all the distances to be computed again) the Sammon algorithm is not suitable for real time applications while, with ours, it's very simple to add a new point into the space so as to track, for example, the expressive intention of a live performance.

This is exactly what the patch shown in figure 11 does while “listening” to the performances of our recorder player.

It should be pointed out, anyway, that the patch is tuned on the data of a particular player on a particular piece and so trying to track the same expressive intentions on different players or

2.2 Recognition of expressive intentions on violin³

Once shown we can track in real time a particular expressivity in an emotional space, it is now interesting to see whether it is possible to build a statistical model that can give us the probabilities assigned by the recognition system to each intention.

To accomplish this, we recorded two set of performances, still on the same excerpt shown in figure 6, played on violin by a first part of the Carlo Felice Theater Orchestra (maestro Fabrizio Ferrari). This time, for simplicity's sake, the required expressive intentions were limited only to the following three cases:

- Agitato
- Brillante
- Cupo

Like in the previous experiments, all performances were recorded with a sampling rate of 22050Hz, 16 bit resolution and then they were analysed by the EyesWeb patch like the one in figure 4 so as to extract the following set of audio cues (averaged by looking at the events detected in the time window which was set to a 4 seconds width and updated 5 times per second. For cues definitions, see §1.4):

- Tempo 1
- Articulation
- Standard Deviation of Articulation
- Sound Level
- Sound Level Difference
- Attack Velocity

One of the two recorded sets was chosen as reference for extracting representative data from the performances. By analyzing the various expressive intentions, the cues (whose distributions we know are Gaussian thanks to the Central Limit Theorem, as noted in §1.4) showed to have the following overall Mean (M) and Standard Deviation (SD) (Table 2):

Mean	Cupo	Brillante	Agitato
Tempo	0.20455	0.11263	0.14558
Articulation	0.59644	0.50132	0.55864
SD of Articulation	0.08214	0.05669	0.06693
Sound Level	0.05284	0.01156	0.03844
Sound Level Difference	-2.1226	-0.6427	-0.8611
Attack Velocity	0.08992	0.03093	0.07226

Standard Deviation	Cupo	Brillante	Agitato
Tempo	0.01402	0.00071	0.00203
Articulation	0.02154	0.01309	0.01556
SD of Articulation	0.00260	0.00092	0.00148
Sound Level	0.00165	0.00003	0.00021
Sound Level Difference	193.694	12.2506	16.1832
Attack Velocity	0.00775	0.00027	0.00100

Table 2: overall mean and standard deviation for the reference set of performances

³ Results from this experiment were presented in (Dillon 2003)

These data are also presented graphically in figure 12 so as to provide an easier evaluation of the differences among the ranges showed by the various cues.

It is interesting to notice how the *Agitato* expressive intention falls in between the *Cupo* and the *Brillante*, showing it has similar characteristics of both.

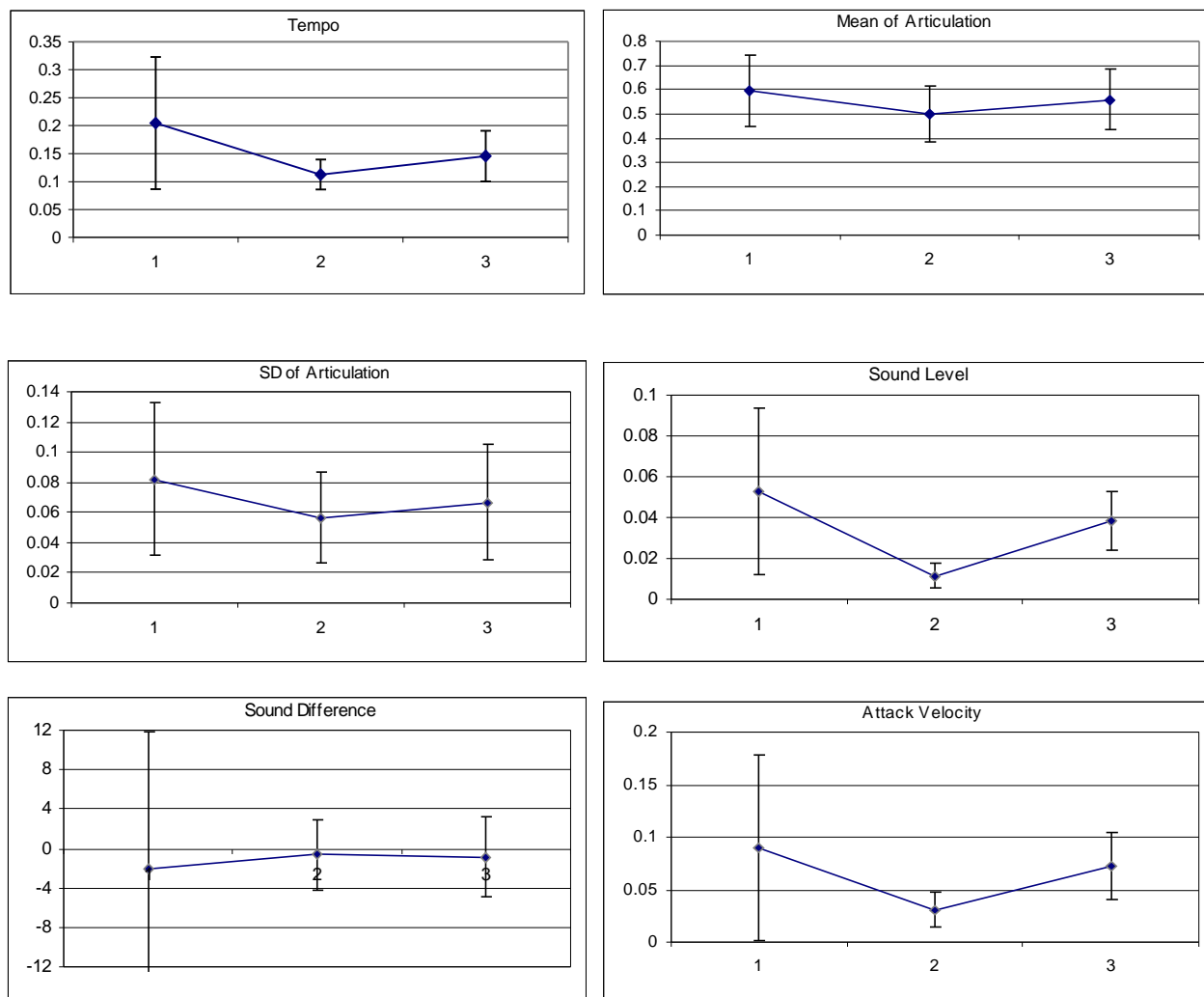


Fig.12: Mean and Standard Deviation of the cues (1 is for *Cupo*, 2 is for *Brillante*, 3 is for *Agitato*)

Now, to check whether the extracted cues are actually able to show statistically meaningful differences among the three performances, a single factor ANOVA⁴ analysis was carried out.

The computed F and p values ($\alpha = 0.05$. F critical value is 3.0191) are shown in Table 3:

Expressive Cue	Computed F	P-Value
Tempo	44.0845	5.69 E-18
Articulation	17.8136	3.98 E-08
SD of Articulation	12.8306	4.028 E-06
Sound Level	73.187	< E-20
Sound Level Difference	0.97481	0.3781891
Attack Velocity	31.6994	1.79 E-13

Table 3: ANOVA Results

⁴ For a reference on this and the following statistical analysis tests see, for example (Crow et al.1960)

As we see, the results are very good with the only exception of sound level difference. Actually the performer followed the same kind of crescendo/decrescendo patterns in all the performances, so this bad result is understandable.

To have a more accurate understanding of the results, also F and T tests were computed for every possible couple of performances for each expressive cue. In this way we can determine whether means and/or standard deviations of the Gaussian distributions are different enough to guarantee a good identification between different performances.

The results are presented in Table 4. The value shown is the probability of incurring in a Type-I error ($\alpha = 0.05$) i.e. rejecting the null hypothesis (samples are coming from the same distribution) while it is actually true.

Tempo	F-Test	T- Test
Agitato / Brillante	1.1 E-07	1.1 E-8
Agitato / Cupo	< E-10	< E-10
Brillante / Cupo	< E-10	< E-10
Mean of Articulation	F-Test	T- Test
Agitato / Brillante	0.3741	0.0017
Agitato / Cupo	0.1599	0.0039
Brillante / Cupo	0.0181	< E-10
Standard Deviation of Articulation	F-Test	T- Test
Agitato / Brillante	0.0158	0.0376
Agitato / Cupo	0.0243	0.0052
Brillante / Cupo	9.09 E-7	2.29 E-7
Mean of Sound Level	F-Test	T- Test
Agitato / Brillante	< E-10	< E-10
Agitato / Cupo	< E-10	2.96 E-5
Brillante / Cupo	< E-10	< E-10
Sound Level Difference	F-Test	T- Test
Agitato / Brillante	0.1518	0.6756
Agitato / Cupo	< E-10	0.2623
Brillante / Cupo	< E-10	0.1801
Mean of Attack Velocity	F-Test	T- Test
Agitato / Brillante	< E-10	< E-10
Agitato / Cupo	< E-10	0.0161
Brillante / Cupo	< E-10	< E-10

Table 4: F and T tests for every pair of performances

As these results show, the statistical analysis produced good results although with the relevant exception of the *Sound Level Difference* cue.

We see there are problems just in a few cases but when one test fails, the other test shows quite good results (like in the Agitato/Brillante for *Mean of Articulation* cue: the F test is very bad but the T test is good).

Hence the probability of wrongly classifying a performance is low, provided both mean and standard deviation of the distributions are known.

Now that we have proved the chosen cues to show statistically meaningful differences between the various performances, we can proceed to propose a system that attempts to recognize the particular expressive intention on the basis of the overall mean and standard deviation for each cue.

The proposed system is based on a Hidden Markov Model (for an introduction to HMM see, for instance, (Rabiner, Juang 1986) or (Ghahramani 2001)) that we implemented in MatLab.

The HMM yields the probability that the cues extracted by the EyesWeb patch at a certain time belong to a given intention and our aim is to test the system with the other set of performances previously recorded by the same violinist, so as to find out if we can correctly classify a cue vector from a particular time frame of a new performance by knowing only a set of global parameters, i.e. mean and standard deviation, gained from a different training set.

The Observable States of the HMM are the cue values while the Hidden States, i.e. those we want to guess by looking at the observable states, are the different expressive intentions, as shown in figure 13.

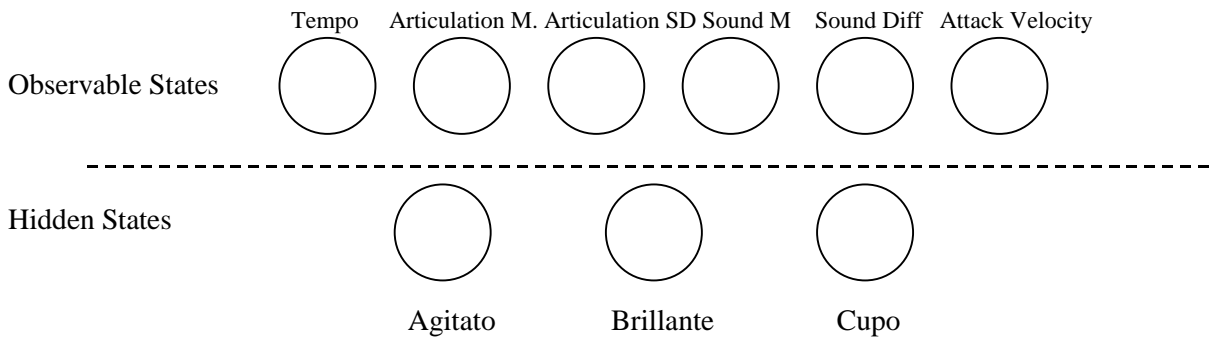


Fig.13: HMM structure

The HMM is defined by a *Transition Matrix* (TM), a *Confusion Matrix* (CM) and by a Π -vector representing the initial state, which is the hidden state we choose the system to start from.

It should be stressed that, in our case, the CM is not constant in time and is a 6x3 matrix defined so as each row yields, for each cue, the conditional probability (Pr.) that the current value belongs to one of the possible hidden states. Note that the sum of the elements in each row must be 1 since the incoming value must be from one of the possible hidden states.

$$\left(\begin{array}{ccc} \text{Pr } .Tempo | Agitato & \text{Pr } .Tempo | Brillante & \text{Pr } .Tempo | Cupo \\ \text{Pr } .Art.M | Agitato & \text{Pr } .Art.M | Brillante & \text{Pr } .Art.M | Cupo \\ \text{Pr } .Art.SD | Agitato & \text{Pr } .Art.SD | Brillante & \text{Pr } .Art.SD | Cupo \\ \text{Pr } .SoundM | Agitato & \text{Pr } .SoundM | Brillante & \text{Pr } .SoundM | Cupo \\ \text{Pr } .SoundDiff | Agitato & \text{Pr } .SoundDiff | Brillante & \text{Pr } .SoundDiff | Cupo \\ \text{Pr } .AttackVel | Agitato & \text{Pr } .AttackVel | Brillante & \text{Pr } .AttackVel | Cupo \end{array} \right)$$

Fig. 14: The Confusion Matrix

Where, for example:

$$\Pr .Tempo | Agitato = \int_A^B \frac{1}{\sqrt{2ps}} e^{-\frac{1}{2} \frac{(x-m)^2}{s^2}} dx \quad (4)$$

σ and μ are the overall standard deviation and mean previously calculated for the cue Tempo, respectively, in the agitato performance.

The integration interval [A,B] is centered around the current cue value and its amplitude D is equal to

$$D = (V_{max} - V_{min}) / N \quad (5)$$

Here V_{max} and V_{min} are the maximum and minimum values assumed by the cue, as calculated previously, and N is equal to $1+1.43\ln(n)$ where n is the overall number of samples (i.e. time frames).

The TM instead, which defines the probability of being in state X at time t , having been in state Y at time $t-1$, is constant in time and has been defined by optimizing the probability of correctly classifying the performances having the CM already defined and the original set of performances as input.

The initial state is assigned to the Agitato (Π -vector = [1 0 0]) since, as we have noted previously, this performance showed to be half way between the two others and so looks like a good starting point.

The MatLab application produces the results shown in figures 15-17: it takes as input a vector with the cues extracted by the EyesWeb patch plus the previous state. Then it computes the CM, normalizing the rows so that the sum of the elements is equal to 1.

Having the CM, we can now compute a 3-component vector V (where each component represents the probability of a hidden state given the current set of cues) by doing the product between a 6-component weighting row vector W and the CM columns (this is useful for emphasizing the effect of a particular cue over the others. In this particular case we slightly emphasized the tempo cue since the tests showed it to be a very meaningful and reliable one).

The probability vector V is normalized in such a way that the sum of its elements makes 1 and then its components (as a column vector) are multiplied with the respective elements of the Transition Matrix, having chosen the row according to the preceeding state value.

In this way we get a new vector P that, once normalized to 1, gives the final probability of having a particular hidden state.

In other words, the resulting j -th component is (st refers to the particular row, i.e. the preceeding state):

$$P_j = V_j * TM_{st,j} \quad (6)$$

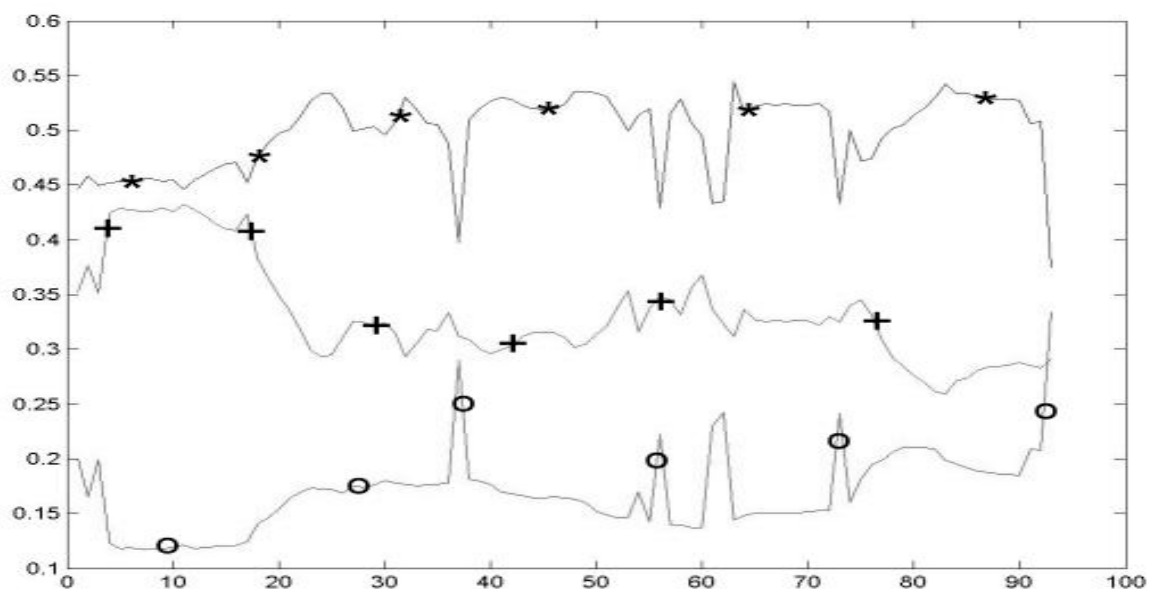


Figure 15: Listening to the ‘Agitato’ performance. The abscissa shows the sample number (there are 5 samples per second) while the ordinate is the probability assigned by the system to a particular expressive intention (P_j in formula n.6). The line marked with ‘*’ refers to Agitato, ‘+’ to Brillante, ‘o’ to Cupo.

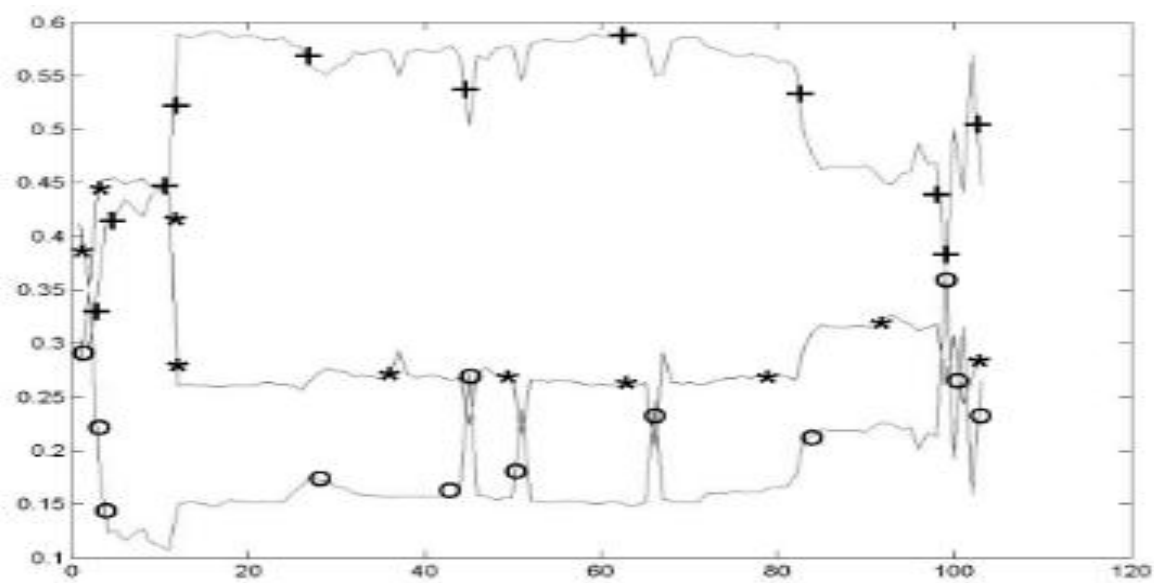


Figure 16: Listening to the ‘Brillante’ performance. Symbols as in fig.15

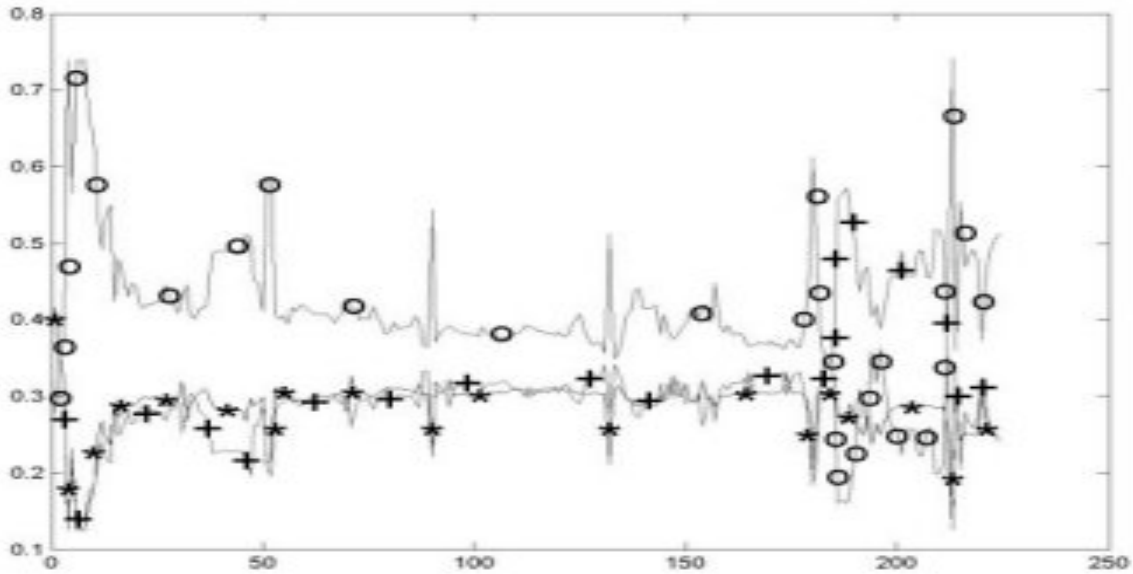


Figure 17: Listening to the ‘Cupo’ performance. Symbols as in fig.15

As we see from the figures, the system shows very good results: the ‘Agitato’ (fig.15) is correctly recognized during the whole performance without any problem, in fact its probability is the highest from the beginning to the end of the piece. The ‘Brillante’ is recognized very well too: the system switches on the correct state just in a couple of seconds. The ‘Cupo’ is recognized almost instantaneously and correctly identified over the whole performance with the exception of a frame, corresponding to the end of bar 7 and bar 8 as shown in figure 6, where it’s misunderstood for being ‘Brillante’.

Actually, in that point we have some repeated 8th notes patterns that were performed with the same kind of bowing between the different performances and this succeeded in “confusing” the system for a short while (it’s interesting to note also that in fig.16, at that point, the ‘Brillante’ and ‘Cupo’ probabilities get close to each other, although in that case the correct ‘Brillante’ state is maintained throughout the whole passage), anyway, during the cadenza ending on the dominant, which concludes the played excerpt, the performance is again recognized correctly, showing this could a reliable approach for the recognition of the expressive intentions of a player once the system has been trained with very basic data such as the mean and the standard deviation of the cues.

Part III

Who is playing?

3.1: A slightly different problem: who is playing?

The problem of recognizing the expressive intention of a musical performance is closely related to another, fascinating one: recognizing the artist who is playing, in other words, recognize his style characteristics.

Previous research on this topic (Widmer 2001; Stamatatos 2002; Stamatatos, Widmer 2002) showed promising results trying to classify several piano teachers and students, while recent developments (Zanon, Widmer 2003) are showing that it is possible to successfully studying performances of famous pianists by means of machine learning techniques.

To seriously study this topic, the acquisition of a very large set of feasible data is a required prerequisite and this is a problem in it's own right, so in this thesis we are simply carrying out a basic experiment (see Dillon 2003b) to check whether a basic knowledge of audio cues distributions is enough to correctly identify who is playing in a particular piece of music so as to understand whether this technique can be of some help in the very complex topic that is the study of style characteristics of different artists.

3.2: Glenn Gould, Maria Joao Pires and Director Musices play Mozart

For our experiment we got a recorded performance of the first 20 bars of the 2nd movement from Mozart Piano Sonata K332 by two very famous pianists: Glenn Gould (Sony Classical SM4K 52627, 1967) and Maria Joao Pires (DGG 431 761-2,1991) then, to make things more interesting, we selected also a computer rendered performance by the well know software Director Musices developed at KTH (Friberg et al. 2000).

The Diretor Musices performance was made without any attempts to simulate a particular expressive style and the following rules/values were used:

- Harmonic Charge: 2.0 (Amp: 1.0, Dur: 0.5, Vibfreq: 0)
- Score Staccato: 1.0
- Duration Contrast: -1.0
- Note Triplet Contrast: 1.0
- Punctuation: 1.0 (Dur: 1.0, Duroff: 1.0, Markphlev7 Nil)
- Phrase Arch: 0.5 (Phlevel: 7, Amp: 1, Turn: 0.5; Phlevel: 6, Amp: 1, Next:0.5, Turn: 0.5, Phlevel: 5, Amp: 1, Turn: 0.5)

Since this performance was originally rendered as a midi file, it was converted into a wave file to make it readable by the EyesWeb system exactly like the other performances (like the previous experiments, we are using a patch such as the one shown in figure 4). Moreover reverberation was added to make it sound more natural and similar to those by Gould and Pires.

Now we wonder whether, using the same set of cues and following the same approach we did in §2.2 for classifying the expressive intentions on violin, we can extract basic parameters, such as mean and standard deviation for each cue, that will be useful for correctly guessing at the particular player over the whole 20 bars and see whether and how the recognition changes across them.

By analyzing all cues over the whole 20 bars as performed by the two pianists and the Director Musices, the following overall Mean (M) and Standard Deviation (SD) values, listed in Table 5 and shown in figure 18, were obtained:

Mean:

	Tempo	M. Articulation	SD Articulation	M. Sound Level	Sound Level Diff.	Attack Velocity
Pires	0.197	0.585274	0.061273	0.001689	-0.085258	0.006014
Gould	0.189	0.542607	0.066712	0.002999	-0.080726	0.010789
Director Musices	0.193	0.571195	0.060982	0.011473	-0.194720	0.024611

Standard Deviation:

	Tempo	M. Articulation	SD Articulation	M. Sound Level	Sound Level Diff.	Attack Velocity
Pires	0.0719	0.124536	0.032434	13.011E-4	0.497080	0.005758
Gould	0.0674	0.091603	0.034942	22.859E-4	0.393085	0.008166
Director Musices	0.0594	0.099107	0.026003	47.366E-4	0.631001	0.009641

Table 5: Overall mean and standard deviation for each cue

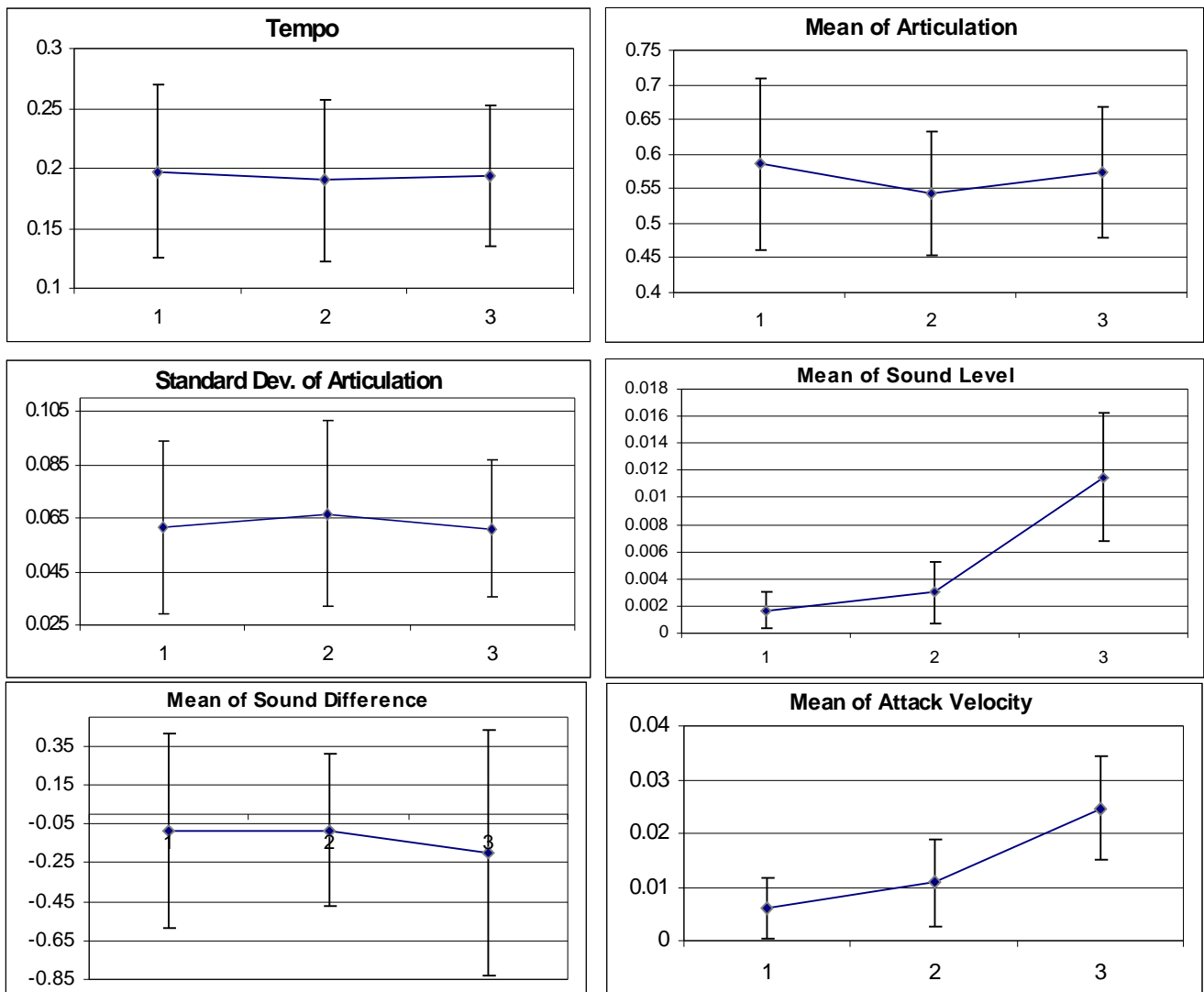


Fig.18: Mean and SD for each cue (1: Pires, 2: Gould, 3: Director Musices)

From these results, we can see that all the performances show very similar tempos and that the Director Musices performance shows much higher sound level values than the pianists.

It should be pointed out that no normalization of loudness was carried out, since the loudness differences between the performances appeared relevant from an expressive point of view. Thus Pires played softly while the others played louder producing higher tone amplitude peaks.

Now, as before, we should check whether this basic and overall description of the various performances is actually able to show statistically meaningful differences among the three performances. Therefore a single factor ANOVA analysis was carried out on all the different cues.

The computed F and p values ($\alpha = 0.05$. F critical value is 3.00) are shown in table 6

Expressive Cue	Computed F	P
Tempo	1.005354	0.366271
Articulation Mean	34.16078	< 0.00001
Articulation Std. Dev.	6.88839	0.001
Sound Level Mean	1874.934	< 0.00001
Sound Level Difference	5.527064	0.004
Attack Velocity Mean	777.0159	< 0.00001

Table 6: Results of the single factor ANOVA analysis for the three pianists

As we can see the results are, again, very good, showing very low p values for all the cues except for the Tempo. This is natural since we already noticed all the performances were similar in this aspect.

It should be noticed that the Director Musices performance was quite different regarding dynamics and loudness. This clearly influences ANOVA results. Therefore, to understand how much each performance differed from the others, F and T tests were computed, for every possible couple of performances for each expressive cue. In this way we can determine if means and/or standard deviations of the Gaussian distributions were different enough to guarantee a good identification between different performances, exactly like we did for the problem faced in §2.2.

The results are shown in Table 7 (here, again, the value shown is the probability of incurring in a Type-I error, i.e. rejecting the null hypothesis):

Tempo	F-Test	T- Test
Pires /Gould	0.12065	0.17359
Pires / Director Musices	0.00014	0.45811
Gould / Director Musices	0.00764	0.44361

Mean of Articulation	F-Test	T- Test
Pires /Gould	7.58942E-19	4.02423E-15
Pires / Director Musices	1.50146E-08	0.05476
Gould / Director Musices	0.20190	2.82246E-07

Standard Deviation of Articulation	F-Test	T- Test
Pires /Gould	0.02805	0.001663
Pires / Director Musices	1.14565E-06	0.871614
Gould / Director Musices	7.58327E-11	0.002139

Mean of Sound Level	F-Test	T- Test
Pires /Gould	1.64603E-54	4.40842E-42
Pires / Director Musices	1.0556E-195	4.87E-142
Gould / Director Musices	3.73603E-66	4.4638E-125

Sound Level Difference	F-Test	T- Test
Pires /Gould	2.23075E-05	0.89642
Pires / Director Musices	1.02221E-05	0.01009
Gould / Director Musices	6.04711E-18	0.003528

Mean of Attack Velocity	F-Test	T- Test
Pires /Gould	1.48838E-22	2.6873E-39
Pires / Director Musices	4.67813E-33	9.8332E-138
Gould / Director Musices	0.00023	2.1417E-93

Table 7: F and T Tests for every pair of performances for each expressive cue

As shown in Table 7, the statistical analysis produced once more good results, again with exception of the Tempo Cue, since all the performances were similar in overall tempo. There are problems just in a few cases but when one test fails, the other tests show quite good results so we can state again that the probability of wrongly classifying a performance is low, provided both mean and standard deviation of the distributions are known.

So we can proceed to build the HMM to classify the performances. In this case it will be like the one showed in figure 19:

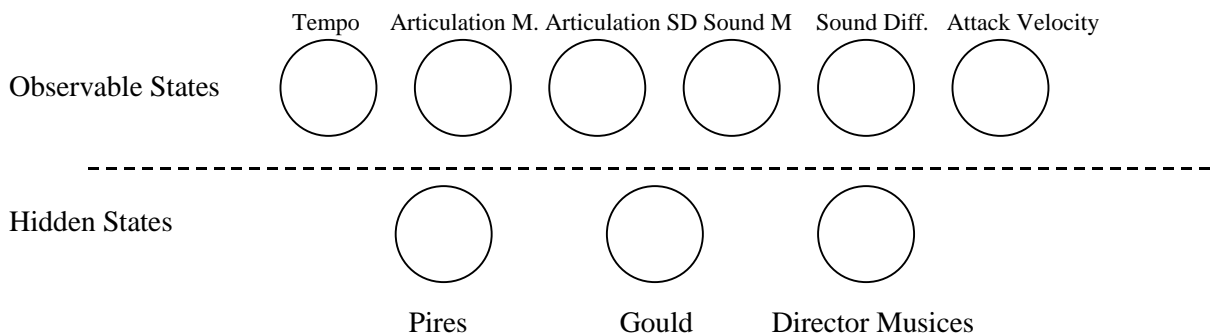


Fig.19: the HMM structure

In this case the Transition Matrix has been simply defined as shown below where rows refer to the previous state and columns to the current state. The rows/columns sequence is Pires, Gould, Director Musices. In each row the sum of the elements must be 1:

$$\begin{pmatrix} 0.45 & 0.35 & 0.2 \\ 0.35 & 0.45 & 0.2 \\ 0.3 & 0.3 & 0.4 \end{pmatrix}$$

As we see, the matrix is defined in such a way that it favours the mantaining of the current state but the mantaining probabilities, along the diagonal in the matrix shown above, are relatively low. This choice allows more freedom to the system since it can take wrong decisions with relative ease but can also favour possible subsequent corrections.

The initial state is assigned to the Director Musices (Π -vector = [0 0 1] i.e. selecting the third row of the TM). The reason for this choice was that the statistical analysis showed this performance to be considerably different from the other two in several aspects. Hence it should be recognized more easily. Accordingly, this allows us to set its mantaining value slightly lower than those assigned to the other states.

The changing state probabilities are equal for the Director Musices row while for Gould and Pires they were assigned to make switching between the professional performers easier than moving to Director Musices.

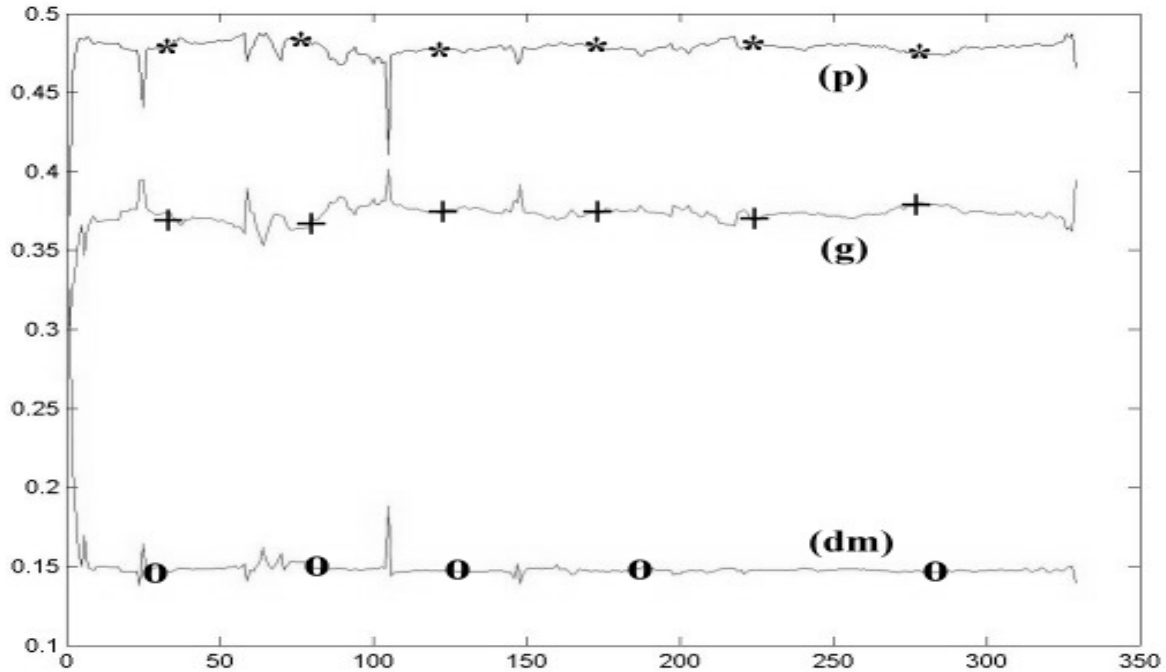
The Confusion Matrix (shown in figure 20) follows the same approach of the one in figure 14 and each element has the same meaning as explained by the formula (4).

$$\begin{pmatrix} \text{Pr.}Tempo | Pires & \text{Pr.}Tempo. | Gould & \text{Pr.}Tempo | DirectorMusices \\ \text{Pr.}ArticulationM | Pires & \text{Pr.}ArticulationM | Gould & \text{Pr.}ArticulationM | DirectorMusices \\ \text{Pr.}ArticulationSD | Pires & \text{Pr.}ArticulationSD | Gould & \text{Pr.}ArticulationSD | DirectorMusices \\ \text{Pr.}SoundM | Pires & \text{Pr.}SoundM | Gould & \text{Pr.}SoundM | DirectorMusices \\ \text{Pr.}SoundDiff | Pires & \text{Pr.}SoundDiff | Gould & \text{Pr.}SoundDiff | DirectorMusices \\ \text{Pr.}AttackVelocity | Pires & \text{Pr.}AttackVelocity | Gould & \text{Pr.}AttackVelocity | DirectorMusices \end{pmatrix}$$

Fig.20: The Confusion Matrix for classifying the Mozart performances by Pires, Gould and DM

To compute the final probabilities P_j (formula n.6) we proceeded like the previous chapter (see pag.22) but, in the present experiment, we emphasized the cues that showed the best statistical results (Articulation Mean, Sound Level Mean and Attack Velocity Mean) over the others.

The analysis of the different performances of the Mozart excerpt yielded the results presented in figures 21, 22 and 23



. Fig.21: Listening to Pires. The abscissa shows the sample number (there are 5 samples per second) while the ordinate is the probability assigned by the system to a particular performer (P_j). The line marked with ‘*’ refers to Pires (p), ‘+’ to Gould (g), ‘o’ to Director Musices (dm).

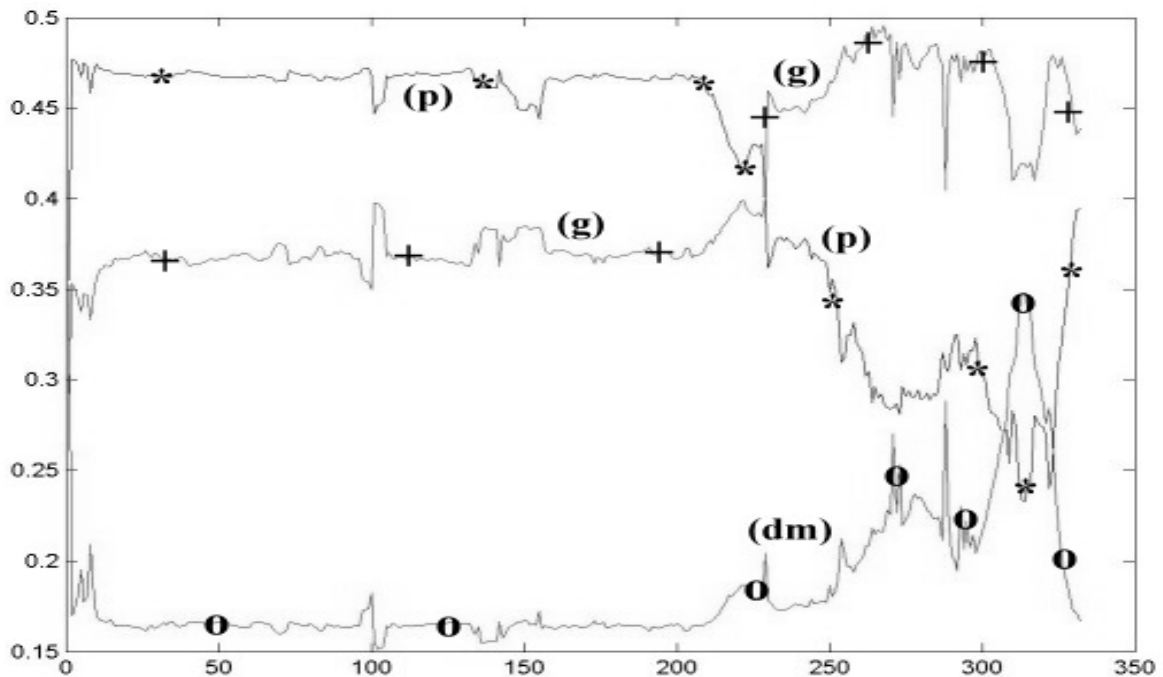


Fig.22: Listening to Gould. Symbols as in fig.21.

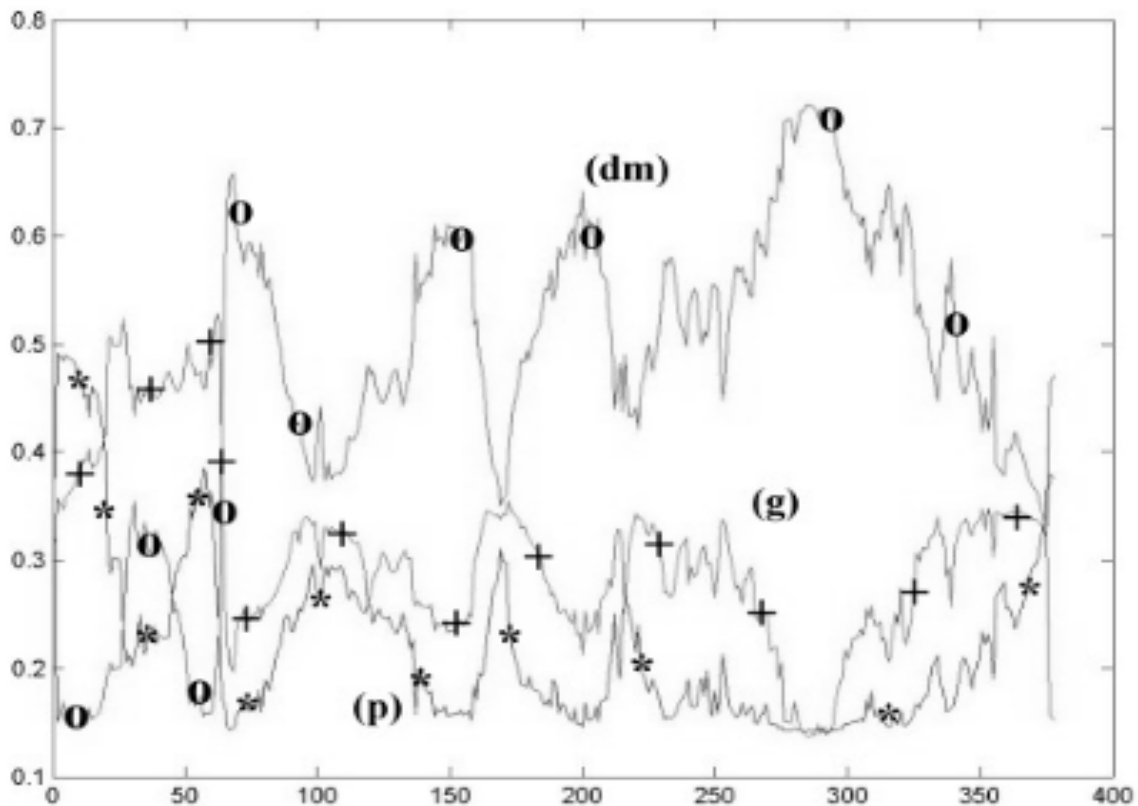


Fig.23: Listening to Director Musics. Symbols as in fig.21.

As we see from fig.21-23, the system shows good results: Pires (fig.21) is correctly recognized during the whole performance without any problem (her probability is the highest from the beginning to the end of the piece).

The Director Musics performance (fig.23) is correctly recognized too: it starts as Pires, moving to Gould soon afterwards, then it gets to the correct state at about sample #60 (i.e. after 12 seconds of music) and keeps it through the end of the piece. It is interesting to see that this performance is the one that received the highest probability value, higher than 70%. We can also notice that these curves are much more irregular than those plotted in the other figures but, like the *Cupo* expressive intentions studied in §2.2, this was expected since the statistical analysis showed several cues with high standard deviation values. Due to this reason, this performance looked more irregular.

The recognition process was least successful for Glenn Gould's performance (fig.22). His performance was mistaken for that of Pires for slightly more than half of the performance excerpt (up to sample #230) but correctly recognized afterwards. This was probably due to the fact that Gould starts very softly and then "builds up" in the second part of the performance so, at the beginning, he is actually "hiding" himself. Pires instead keeps an overall soft approach through the whole excerpt so, at the beginning, the two performances actually have similar characteristics.

In conclusion, the system displays good results and seems to be able to recognize the particular performer, in this particular piece, despite the rather simplistic tools used.

It would be then interesting to study large scale data and see whether it is possible to extract more general information regarding style characteristics of single artists or even of particular schools of playing.

Part IV

Detection of Arousal

4.1: Another different problem: what is “Arousal”?

Whether music is actually able to induce *emotions* in listeners or not is still a topic on which the scientific community frequently debates without being able to find a common view (Scherer 2003). Nonetheless the research on this topic is very lively all around the world and the studies that tried to correlate several aspects of musical structures to emotional reactions have produced fundamental works, from the pioneering (Cooke 1959) to (Sloboda 1991) and, more recently, (Gabrielsson, Lindstrom, 2001), while studies focusing on the analysis of musical performance aspects, such as tempo and articulation, are summarised in (Juslin 2001) .

Lately, also in our Musical Informatics Laboratory there have been some experiments aimed at measuring an *emotional engagement* of listeners and then tried to correlate the results with video and audio data streams (Gremo 2002, Casazza,ertino 2003, Timmers et al. 2003, Marolt et al. 2004), following the ideas and approach proposed in (Krumhansl, Schenck 1997), so as to quantify the *arousal of emotion* (i.e. a high emotional engagement) in people.

Personally, I believe music is able to produce strong and sudden emotions in listeners, as it was commonly believed in the past centuries where none would have discussed about its power to move people emotionally (as clearly shown by the introductory excerpt to this thesis taken from the beginning of Monteverdi’s Orfeo, 1607 or by XVIII century reports of performances of singers such as Farinelli, able to temporarily heal Felipe V from his depression, or Pacchiarotti, who forced a whole orchestra to stop during a performance since he moved all of them to sighs and tears (Barbier 1999), or even to XIX century histeric reports of Paganini’s concerts (Berri 1962, Guhr 1830)). So I think the biggest problem is not deciding whether music does produce emotions or not but how to scientifically measure this effect and how to produce it at will.

The measuring technique is a very difficult problem to solve in its own right: probably the best solution would be to design some systems that do not require a conscious feedback from the listener (such as electrodes and sensors measuring haert beat, skin conductivity, blood pressure etc.) but since such experiments are extremely difficult to be arranged properly, the most common tool used is a simple slider that should be moved by the listener when he/she feels the music is producing some effects on him/her.

This is obviously quite risky since it would be very easy for unexperienced listeners to simply track the volume of the performance running on and not any emotional effect such music is actually producing on them.

Due to this problem, we believe that, for having a meaningful experiment with this very basic tool not only the choice of the music pieces is critical but also the people who are chosen as subjects should be very well instructed on what to do and, whenever possible, they should also have had some previous experience in these kind of experiments so as to avoid fake measurements as much as possible.

4.2: Arousal in Bach Solo Violin Sonatas

Although being aware of the “*volume effect*” issue just explained, we decided to use a tracking slide in a EyewWeb patch (figure 24) and then to select a small group of people among researchers in the field who knew the risks involved in this approach and hence should have known how to avoid its pitfalls.

The used patch gives the listener control on a slider, ranging from 0 (no emotional involvement) to 127 (very high involvement) and then saves the slider position, along with a time stamp, to a text file for later analysis. The slide value is saved two times per second.

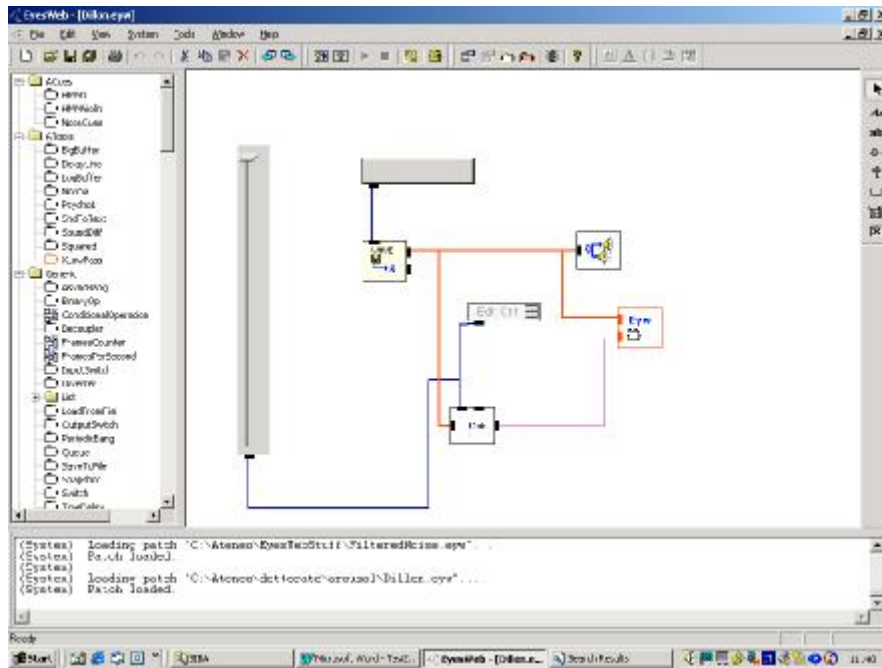


Fig.24: an EyesWeb patch for tracking emotional response while listening to music

For our experiment we selected two contrasting movements from J.S. Bach Sonatas for solo violin (the Presto from Sonata I, shown in figure 25, and the Largo from Sonata III, figure 26) so as to look for aspects that go beyond the character of the piece but that, nonetheless, could be responsible for rising emotional effects in listeners.



Fig.25: J.S. Bach: Sonata I BWV1001, Presto

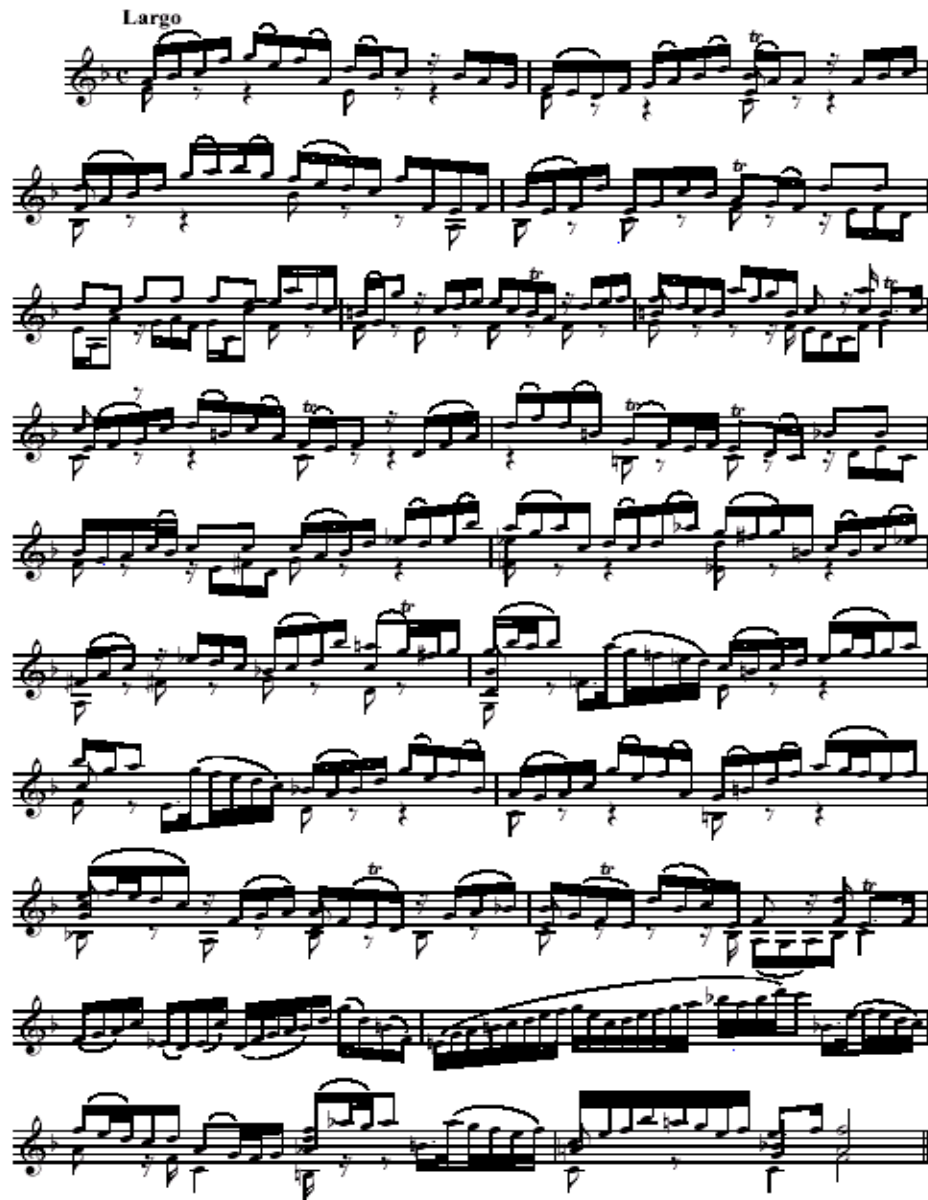


Fig.26: J.S. Bach: Sonata III BWV1005, Largo

The two movements were performed, on a priceless Guarneri del Gesù made in 1728, by Tanja Becker Bender, a young international level soloist winner of numerous prizes and awards, and professionally recorded in an historical setting (a XIV century Abbey located near Genoa) so as to make her feel as inspired as in a real performing environment.

For the experiment, three people were chosen and they had a chance to listen to the music first, if they were not already acquainted with these particular pieces, and then another listening session followed where they tracked their emotional engagement.

The results for each person and the overall average are shown in the following figures (27 – 34: Y axis shows arousal response, X axis shows sample number. There are 2 samples per second):

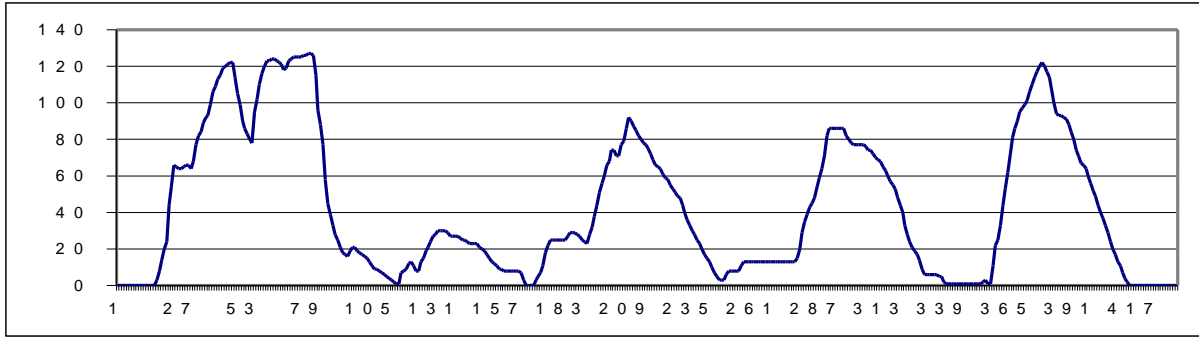


Fig.27: Arousal response to Presto BWV1001. First subject

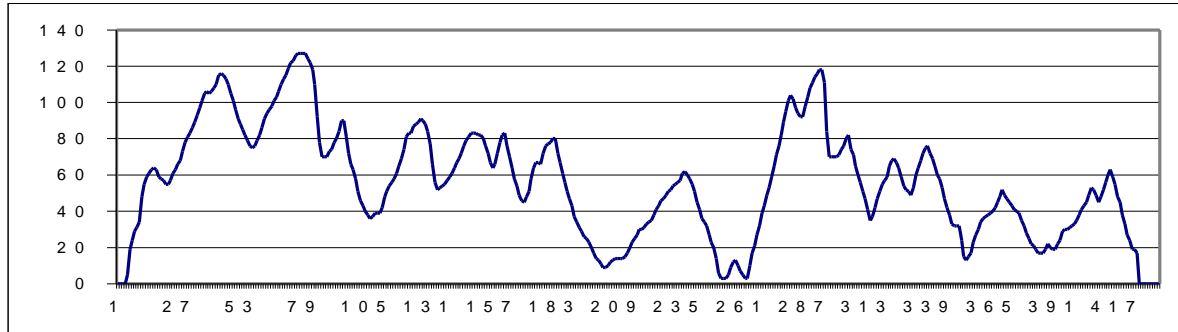


Fig.28: Arousal response to Presto BWV1001. Second subject

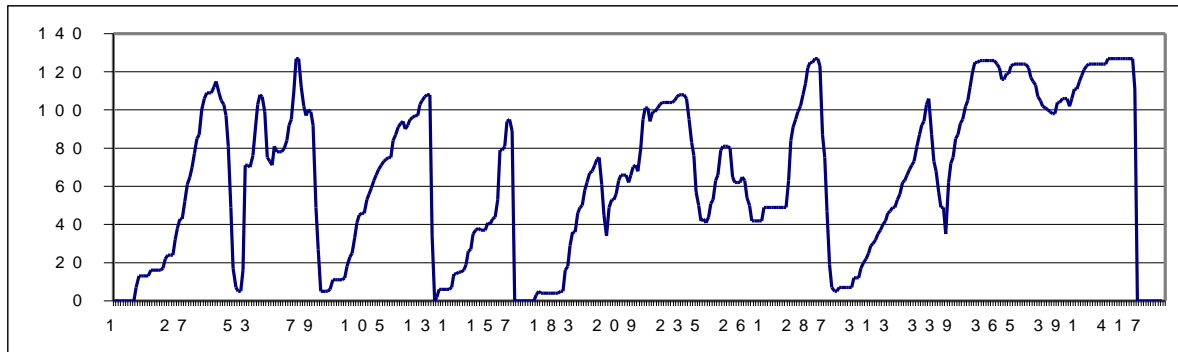


Fig.29: Arousal response to Presto BWV1001. Third subject

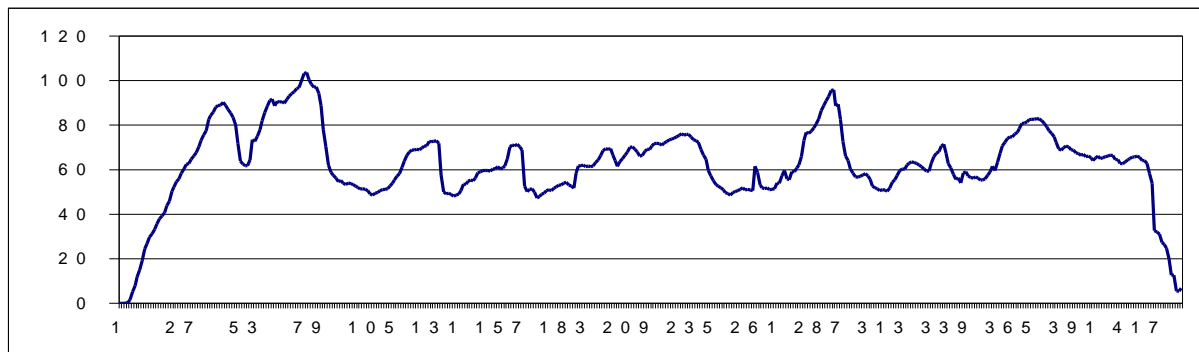


Fig.30: Arousal response to Presto BWV1001. Average

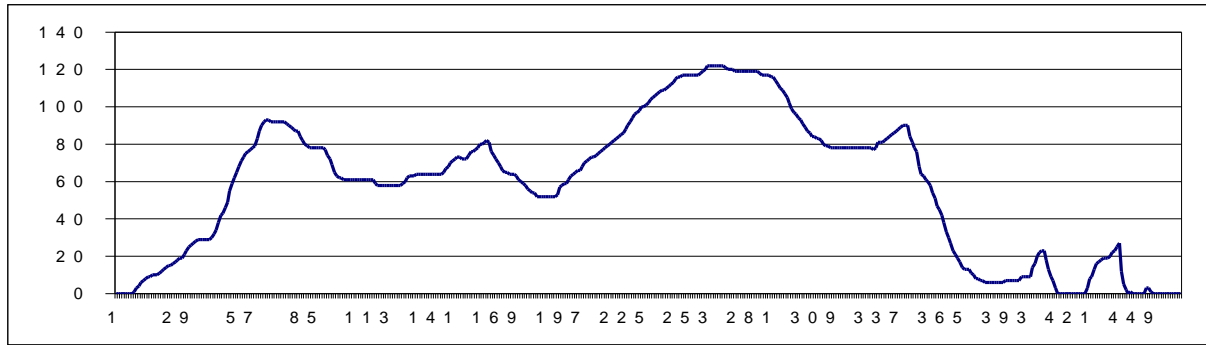


Fig.31: Arousal response to Largo BWV1005. First Subject

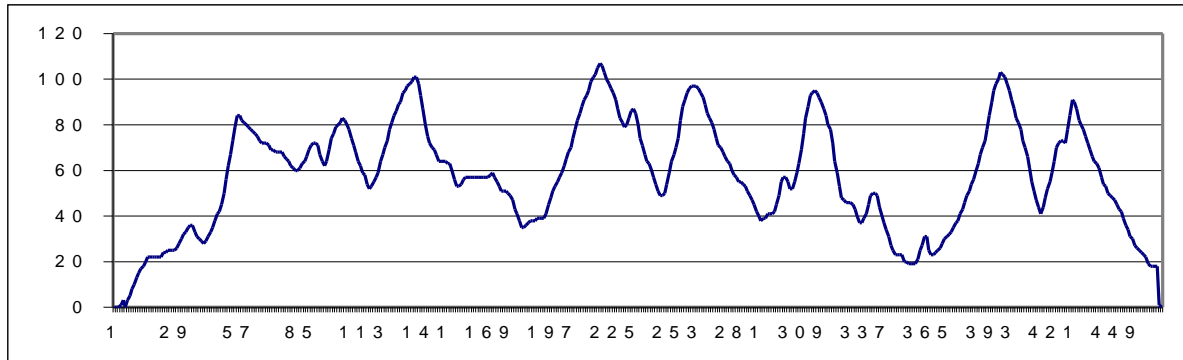


Fig.32: Arousal response to Largo BWV1005. Second Subject

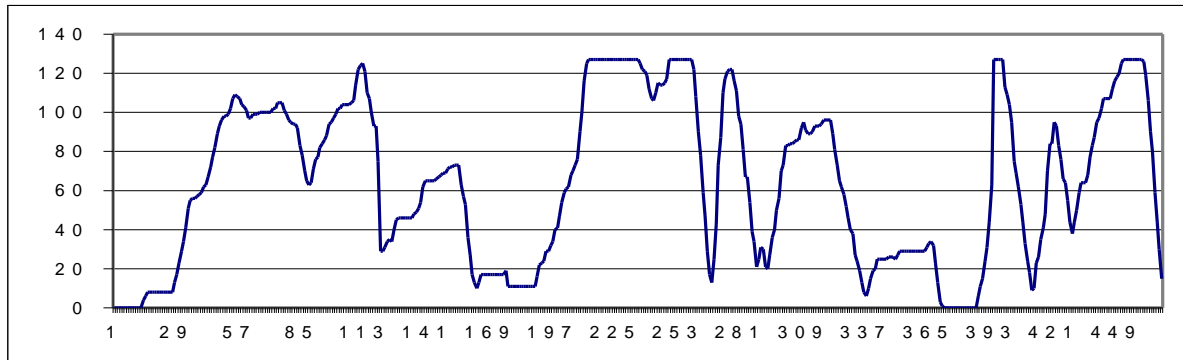


Fig.33: Arousal response to Largo BWV1005. Third Subject

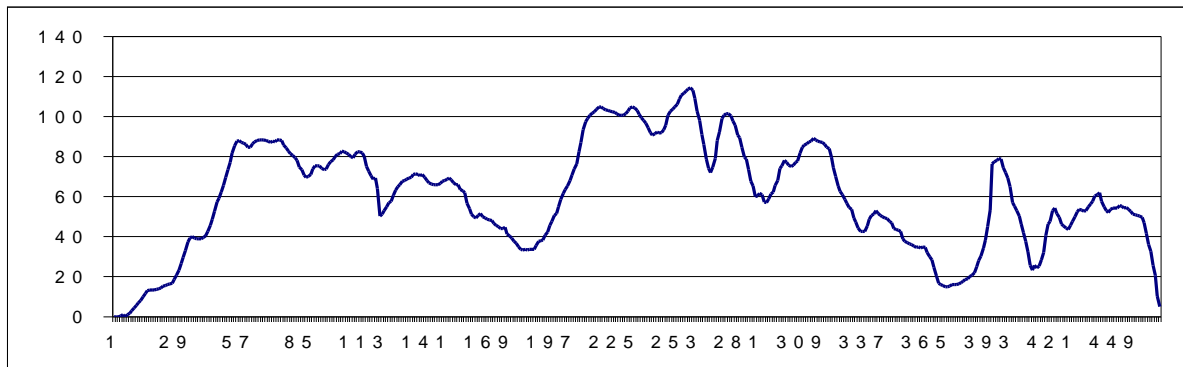


Fig.34: Arousal response to Largo BWV1005. Average

Interestingly, the responses of the various subjects show peaks in the same positions and these, underlined by the average profiles, are the points that interest us in this study.

It is also interesting to see that most of the peaks fall down more steeply than their rises, according to the idea that intense emotions can not last too long but only for a few seconds and then vanish quickly (Picard 1997).

Now we wonder which are the factors that determined the rise and fall of the emotional engagement and whether it is possible to extract a set of rules able to explain these and then to predict them.

First of all, let us split the average profiles dividing the rising parts from the falling ones, as shown in figures 35 and 36:

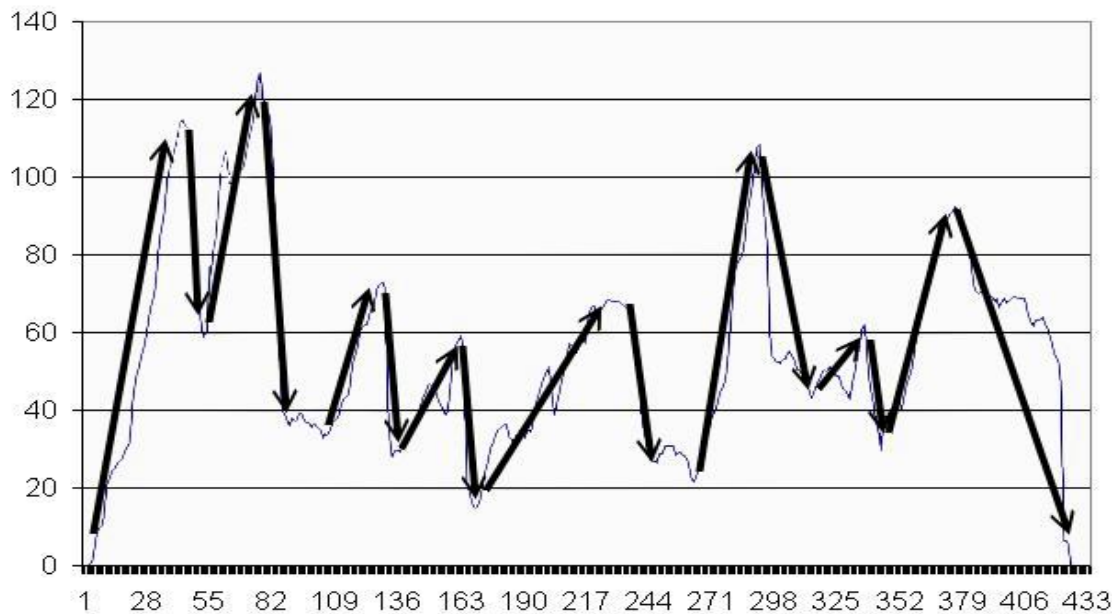


Fig.35: Presto BWV1001: average arousal underlining rising, falling and stable segments

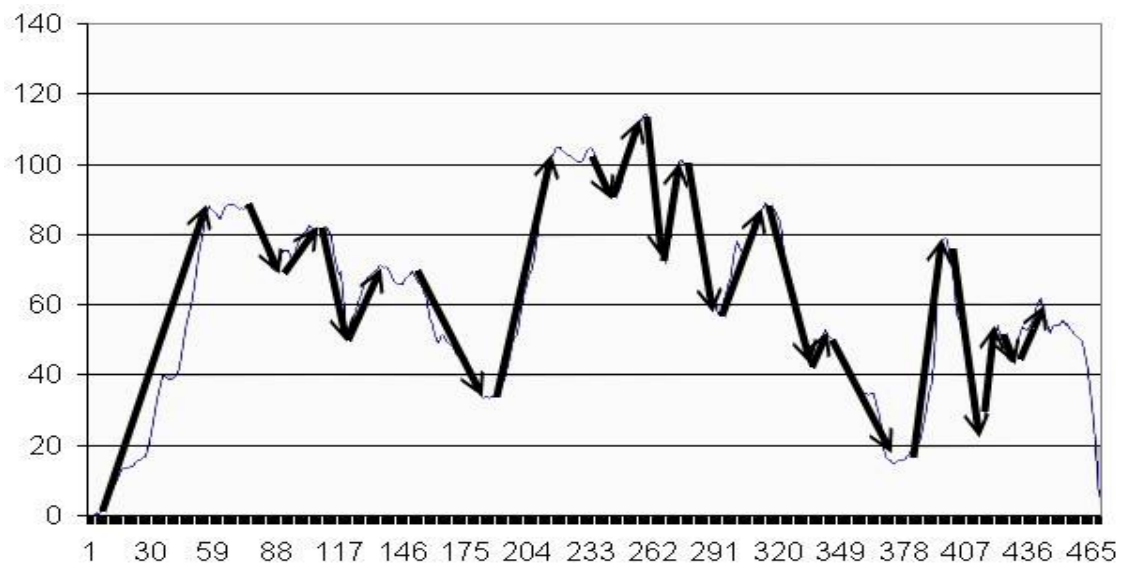


Fig.36: Largo BWV1005: average arousal underlining rising, falling and stable segments

The next step was to extract the cues from the performances and then to see how these correlate with the rising/falling arousal profiles.

Due to the particular task we are facing, this time we used an extended set of cues:

- Tempo 1 (Note DR)
- Tempo 2 (Notes per second)
- Articulation
- Standard Deviation of Articulation
- Sound Level
- Sound Level Difference
- Attack Velocity

Moreover we developed further blocks for extracting the overall energy of the signal and also the energy in different frequency bands so as to analyse whether some bands had more relevance in the listeners' responses (the Guarneri being used showed extremely strong harmonics) and we also looked at the pitch being played to see whether passages with rising/falling scales had any influence on the responses.

The available range was divided logarithmically so as to have a better resolution in the low frequency bands. In particular the analysed bands were:

- 172 – 334 Hz
- 344 – 689 Hz
- 689 – 1378 Hz
- 1378 – 2756 Hz
- 2756 – 5512 Hz
- 5512 – 11025 Hz

Both recorded performances were analysed using the time window approach.

The Presto had a time window width of two seconds while for the Largo the width was set to three seconds (we reduced the time frame width in respect to the others experiments since we didn't want to miss fast events that might have been responsible for changes in the arousal measurements). The time window was updated two times per second.

The following tables (8 through 15) show the correlation coefficients R between the measured arousal and the various cues. R values with higher absolute values than 0.40 are marked in **bold** and R range is from -1 (variables fully inversely correlated) to 1 (variables fully correlated):

Rises (start and ending samples)	MeanDR	Notes/s	ArtMean	ArtSD	SndLev	SoundDiff	AttVel	Av Energy	Pitch
1.44	-0.3769	0.4293	-0.2360	0.1682	0.0127	-0.0959	0.1997	-0.0228	0.4397
54.77	-0.7487	0.4186	-0.5953	-0.2744	0.8436	0.2337	0.6864	0.1837	-0.5525
108.131	0.4823	0.6648	-0.0824	-0.3672	0.2491	0.0078	0.3304	0.6589	0.5844
140.164	0.0211	-0.0322	-0.2231	0.2996	-0.4276	0.0139	-0.3210	-0.2836	-0.5498
172.237	-0.0488	0.1461	-0.0424	-0.1689	-0.7748	0.0879	-0.5161	-0.3935	0.2865
266.292	-0.2232	0.3065	0.1222	0.0932	0.2165	0.1195	0.2266	0.2315	0.0397
317.337	-0.2179	-0.3808	-0.1258	0.1070	-0.0324	-0.5850	0.1460	-0.3783	0.2776
345.379	-0.3466	-0.1674	-0.0029	-0.1311	0.6087	-0.1907	0.4162	0.1624	0.2863

Table 8: R values for basic cues (Presto BWV1001) tracking rises in arousal response

Rises (start and ending samples)	172-344 Hz	344/689Hz	689/1378Hz	1378/2756Hz	2756/5512Hz	5512/11025Hz
1.44	-0.3944	0.0305	0.0351	-0.0490	-0.0016	-0.0968
54.77	0.2015	0.2224	-0.2100	-0.1428	-0.3425	-0.3685
108.131	-0.0994	0.1113	0.6381	0.5963	0.5632	0.6652
140.164	0.1159	-0.3624	-0.1735	-0.1010	-0.2429	-0.3791
172.237	-0.2988	-0.3681	-0.1392	-0.1839	0.0292	-0.0417
266.292	0.0712	0.0372	0.4453	-0.0008	0.2902	0.2479
345.379	-0.2927	0.0599	0.1855	0.1999	-0.1630	-0.0205
317.337	-0.2105	-0.3592	-0.1946	-0.2504	-0.1973	-0.2228

Table 9: R values for energy band cues (Presto BWV1001) tracking rises in arousal response

Falls (start and ending samples)	MeanDR	Notes/s	ArtMean	ArtSD	SndLev	SoundDiff	AttVel	Av Energy	Pitch
46.54	0.8591	0.8565	0.6496	0.4792	0.9018	-0.5537	0.9036	0.5629	0.7375
76.92	0.2538	-0.2634	0.3679	0.7140	-0.3008	-0.0721	-0.6944	0.1448	-0.3476
131.137	0.8838	0.7792	-0.1595	0.5149	-0.6044	0.3122	-0.4703	0.7001	0.8931
164.172	0.1056	0.7804	0.5351	-0.5667	-0.8115	-0.3635	-0.7359	0.2035	-0.1393
237.251	0.6374	0.8911	-0.0302	-0.1005	-0.3339	0.0892	-0.3736	0.6584	0.6254
292.303	-0.9148	0.4703	0.2098	-0.1391	-0.5460	0.2005	0.1307	-0.6154	-0.3980
337.345	-0.5946	0.1086	-0.2943	0.5403	0.9852	-0.3288	0.7372	-0.3710	0.4285
379.401	-0.3763	-0.1775	-0.2744	-0.2414	0.1490	-0.5418	-0.0079	-0.0703	-0.0676

Table 10: R values for basic cues (Presto BWV1001) tracking falls in arousal response

Falls (start and ending samples)	172-344 Hz	344/689Hz	689/1378Hz	1378/2756Hz	2756/5512Hz	5512/11025Hz
46.54	0.0276	0.4019	0.7875	0.5109	0.6028	0.4708
76.92	0.1648	0.1466	0.1230	-0.0007	-0.3853	-0.4249
131.137	-0.6677	-0.0343	0.6681	0.6887	0.7042	0.9151
164.172	0.3286	0.1769	0.1783	0.3281	-0.4137	-0.3895
237.251	-0.3698	-0.4454	0.6985	0.5825	0.6997	0.6599
292.303	-0.2438	-0.3654	-0.3907	-0.2422	-0.5723	-0.4649
337.345	-0.2473	0.0095	-0.2743	-0.8711	-0.7635	-0.7594
379.401	-0.1480	-0.1494	0.0355	-0.0281	-0.2039	-0.1907

Table 11: R values for energy band cues (Presto BWV1001) tracking falls in arousal response

Rises (start and ending samples)	MeanDR	Notes/s	ArtMean	ArtSD	SndLev	SoundDiff	AttVel	Av Energy	Pitch
1.61	-0.4984	0.2645	-0.2649	0.0561	0.8676	0.1455	0.8334	-0.1700	0.3324
70.108	-0.2913	-0.1254	0.0752	-0.0450	0.0690	-0.2661	-0.1120	-0.3368	-0.1107
122.141	-0.3558	0.2800	-0.3236	0.4375	0.6838	-0.1214	0.7664	0.4368	0.4788
190.221	-0.0666	0.3624	-0.0143	-0.3642	0.8136	0.4348	0.5182	0.5438	0.7880
247.261	-0.8723	-0.4233	-0.6048	0.0130	0.7856	-0.2172	0.7713	0.0426	0.4659
269.278	-0.3859	0.6645	-0.2753	0.1566	-0.9628	0.2969	-0.8605	0.5298	-0.1459
294.318	0.3120	-0.4262	0.6200	0.4173	-0.3906	0.2692	-0.3362	-0.1851	-0.0029
337.345	-0.7628	0.01264	-0.8468	0.1714	0.8757	0.1834	0.5141	-0.7574	-0.3768
377.398	-0.2318	0.2168	-0.2318	0.4938	0.8095	-0.1467	0.6720	0.3749	0.2187
415.423	-0.2217	0.6668	-0.0567	-0.1360	-0.8021	0.1462	-0.4113	-0.3382	-0.4088
430.442	-0.0160	-0.6479	-0.5188	0.0924	-0.0330	-0.0232	0.6981	-0.3376	0.4616

Table 12: R values for basic cues (Largo BWV1005) tracking rises in arousal response

Rises (start and ending samples)	172-344 Hz	344/689Hz	689/1378Hz	1378/2756Hz	2756/5512Hz	5512/11025Hz
1.61	-0.0961	0.1419	0.4166	0.4409	-0.2491	0.0358
70.108	-0.2408	-0.2420	-0.2951	-0.2260	-0.2535	-0.1346
122.141	0.2052	0.0821	0.3584	0.3726	0.4364	0.4661
190.221	-0.1460	0.2412	0.6106	0.4667	0.5207	0.5951
247.261	0.3018	-0.5291	0.0752	0.0703	0.1984	0.4561
269.278	-0.3877	0.5041	-0.1079	0.1944	0.5331	0.1864
294.318	-0.4838	-0.2312	0.1811	-0.2021	-0.1489	0.1532
337.345	-0.3199	-0.4437	-0.9051	-0.3922	-0.7373	-0.8016
377.398	-0.1284	0.3894	0.5474	0.6762	0.1750	0.1031
415.423	0.0553	-0.1245	-0.2575	-0.4389	-0.3558	-0.1828
430.442	-0.5905	-0.6545	-0.2484	0.1112	-0.2626	-0.1307

Table 13: R values for energy band cues (Largo BWV1005) tracking rises in arousal response

Falls (start and ending samples)	MeanDR	Notes/s	ArtMean	ArtSD	SndLev	SoundDiff	AttVel	Av Energy	Pitch
76.89	-0.5486	0.4329	-0.2324	0.1130	0.2483	-0.7637	-0.2687	-0.5888	-0.2346
110.122	-0.3450	-0.1628	0.1869	-0.1208	0.8953	-0.0430	0.8420	-0.0111	0.4130
154.187	-0.1673	-0.3328	-0.0369	-0.0356	0.4420	-0.0641	0.1878	0.0287	-0.1127
234.246	0.3941	0.7818	0.2022	-0.4052	0.3087	0.8537	0.9556	0.8234	0.1505
260.269	-0.6003	0.4051	0.1393	-0.0524	-0.6743	-0.2971	-0.8605	0.2235	0.6168
278.291	0.0245	0.3117	0.2281	-0.3184	0.1652	0.6261	-0.5291	0.3099	0.5279
318.337	0.4243	0.7934	-0.1234	0.1351	0.5704	-0.1280	0.7159	0.7351	0.7902
344.375	0.0723	0.2748	-0.3534	-0.2941	-0.7079	0.2646	-0.7723	0.0445	0.4542
398.415	-0.2559	0.3618	-0.5106	0.4921	-0.9102	-0.2006	-0.8918	-0.2357	-0.5819
423.430	-0.8087	-0.4513	-0.6779	0.6924	-0.0530	-0.3975	0.6109	-0.5915	0.7880

Table 14: R values for basic cues (Largo BWV1005) tracking falls in arousal response

Falls (start and ending samples)	172-344 Hz	344/689Hz	689/1378Hz	1378/2756Hz	2756/5512Hz	5512/11025Hz
76.89	-0.4303	-0.6747	-0.3285	-0.2753	-0.4402	-0.1944
110.122	0.0068	-0.1936	0.1065	0.1401	0.2963	0.3914
154.187	0.1942	0.0307	0.1573	-0.0355	-0.3914	-0.2532
234.246	0.4767	0.5844	0.7395	0.7854	0.6557	0.7345
260.269	-0.2832	-0.5671	0.4198	0.4015	0.4459	0.6038
278.291	-0.5535	0.2307	0.3664	0.3400	0.4929	0.4603
318.337	0.1807	0.1757	0.6303	0.5381	0.5272	0.5410
344.375	-0.3309	-0.0249	0.0902	0.2419	0.0926	0.0519
398.415	0.4731	-0.0460	-0.2019	-0.2974	-0.1476	-0.1307
423.430	-0.3140	-0.5410	-0.2759	-0.6739	-0.5926	-0.6685

Table 15: R values for energy band cues (Largo BWV1005) tracking falls in arousal response

By looking at these tables we can try to understand which are the most relevant cues for underlining arousal effects and then try to use these for extracting rules able to predict arousal changes.

First of all, we should note that the *Sound Level Cue* is a very important one, as we would have expected, since it often gets high absolute values in the correlation coefficient. Nonetheless we see that we had rises marked both by positive values (volume gets louder) but also others marked by high negative ones (volume gets softer). This is true also for the falls, so the listeners have avoided the pitfalls of the *volume effect* problem (i.e. simply tracking volume changes).

In table 16 we see a summary of the previous tables where cues showing significantly high correlation value ($|R| > 0.40$) are written in bold:

Cue	Rises (BWV1001)	Falls (BWV1001)	Rises (BWV1005)	Falls (BWV1005)	Total
Note DR	2	5	3	4	14
Notes / s	3	5	5	5	18
Art. M.	1	2	4	2	9
Art SD	0	5	3	3	11
Sound Level	4	5	8	6	23
Sound Diff.	1	2	1	3	7
Att. Vel.	3	5	9	8	25
Av. Energy	1	4	4	4	13
Pitch	4	4	5	7	20
172-344 Hz	0	1	2	4	7
344-689 Hz	0	2	4	4	10
689-1378 Hz	2	3	4	3	12
1378-2756 Hz	1	4	4	4	13
2756-5512 Hz	1	6	4	6	17
5512-11025Hz	1	6	4	5	16

Table 16: number of times each cue gets a correlation value $|R| > 0.40$

From this table we see how the most relevant cues for identifying arousal changes are: Notes/s, Sound Level, Attack Velocity, Pitch and the energy in the 2756-5512Hz band (from now on we will call this “*Mid Harmonic Energy*”).

These cues data have then been used for generating classification/decision trees following the well known C4.5 generation and pruning algorithms proposed by Ross Quinlan (Quinlan 1993).

The trees were built by taking cues data during the rises and falls patters underlined previously (figs. 35 and 36) and our aim is to classify these two categories (for a similar experiment that tries to classify different kind of rises and falls, see (Marolt et al. 2004)). The overall data set was divided into two groups: a training one and test one, the latter containing 10% of the original data randomly chosen. In this way we had, for the Presto, a training set of 323 vectors (each containing the values of the five cues underlined in Table 16) and a test set of 36 vectors, while for the Largo the training set had 348 vectors and the test set 41.

A summary of the trees data and results is shown in the following tables:

Number of Training observations	323	Number of Predictors	5
Number of Test observations	36	Class Variable	Rise/Fall
Total Number of Nodes	106	% misclassified	
Number of Leaf Nodes	54	On Training Data	4.33%
Number of Levels	20	On Test Data	22.22%

Table 17: Classification Tree Model for Presto BWV 1001

Number of Training observations	348	Number of Predictors	5
Number of Test observations	41	Class Variable	Rise/Fall
Total Number of Nodes	150	% misclassified	
Number of Leaf Nodes	56	On Training Data	4.89%
Number of Levels	17	On Test Data	36.59%

Table 18: Classification Tree Model for Largo BWV 1005

While in figures 37 and 38 we see a little portion of the generated tree:

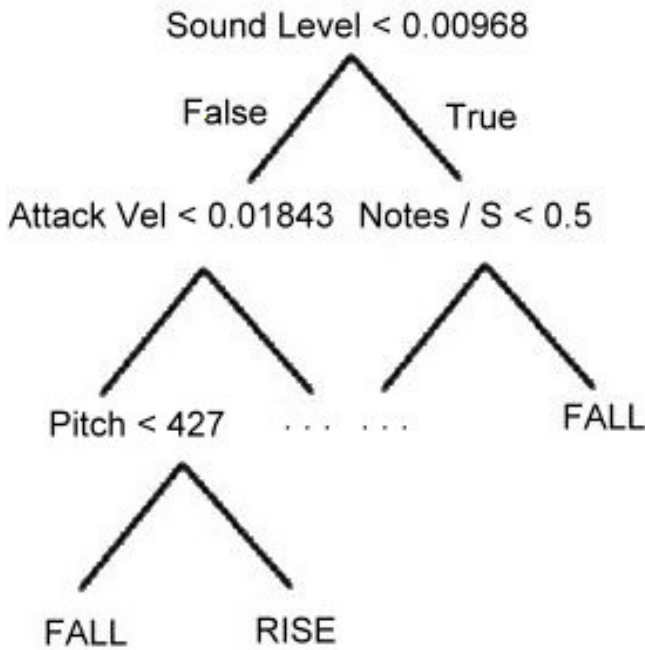


Fig. 37: Sample Tree for Presto

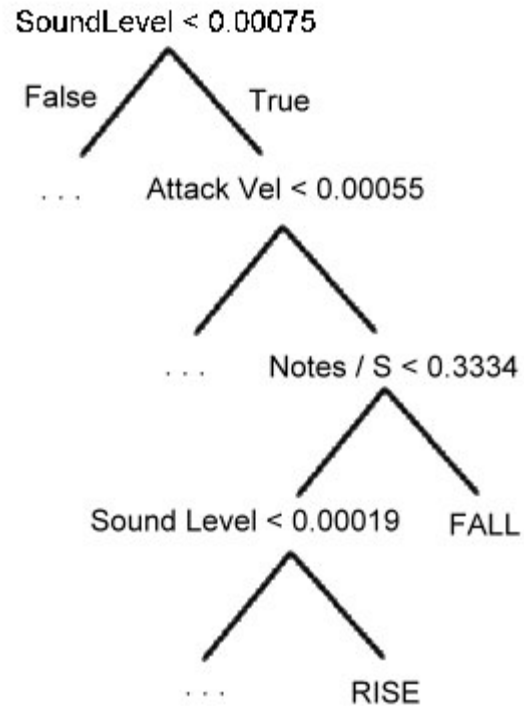


Fig. 38 Sample Tree for Largo

The generated rules try to underline some common aspects found while going through the trees and are sometimes overlapping and redundand. Among those generated by the system, the most interesting ones, evaluated by looking at the *support* (how much of the original data they can be applied to) and *confidence* (how much of the data they classify correctly) percentages are presented in Tables 19 and 20.

Rule	Support	Confidence
If Notes/s < 0.5 then R/F = F	1.5 %	100%
If Notes/s >= 0.5 then R/F = R	98.5%	73.8%
If Notes/s > 3.0 then R/F = R	38.3%	87.4%
If SoundLevel >= 0.00968 then R/F = F	12.1%	61.5%
If Sound Level < 0.01076 then R/F = R	91.3%	76.3%
If Sound Level >= 0.01076 then R/F = F	8.7%	60.7%
If Sound Level < 0.00827 then R/F = R	82.4%	78.2%
If Sound Level >= 0.00827 then R/F = F	17.6%	50.9%
If MidHarmonicsEnergy < 0.0003889 then R/F = R	54.2%	76.0%
If MidHarmonicsEnergy >= 0.0009881 then R/F = R	18.0%	72.4%
If AttackVel >= 0.01611 AND MidHarmonicsEnergy < 0.0003167 then R/F = F	2.2%	100%
If AttackVel >= 0.01843 then R/F = F	7.1%	79.6%
If AttackVel >= 0.01732 then R/F = F	8.0%	65.4%
If AttackVel >= 0.00818 then R/F = R	43.3%	67.1%
If AttackVel >= 0.00599 then R/F = R	61.3%	72.2%
If Pitch >= 553 then R/F = R	43.7%	79.4%
If Pitch >= 626 then R/F = R	26.5%	77.3%

Table 19: Generated Rules for Arousal prediction in Presto BWV1001 (R: Rise, F: Fall)

Rule	Support	Confidence
If AttackVel < 0.00014 then R/F = F	4.0%	64.3%
If AttackVel >= 0.00712 then R/F = R	22.4%	60.3%
If AttackVel >= 0.00055 AND SoundLevel < 0.00075 then R/F = F	6.9%	95.8%
If AttackVel > 0.00356 AND Pitch >= 601 AND SoundLevel >= 0.00131 then R/F = R	11.2%	87.2%
If SoundLevel >= 0.00371 then R/F = R	42.0%	61.6%
If Pitch >= 531 then R/F = R	64.9%	61.5%
If Pitch >= 712 then R/F = R	21.3%	64.9%
If Pitch < 495 then R/F = F	25.3%	61.4%
If Pitch < 357 then R/F = F	4.3%	80.0%
If Notes/s < 0.333 then R/F = F	1.2%	100%
If Notes/s < 0.666 then R/F = F	5.2%	61.1%
If Notes/s >= 0.666 then R/F = R	94.8%	54.4%
If Notes/s >= 1.666 then R/F = R	37.6%	61.1%
If MidHarmonicsEnergy >= 0.002352 then R/F = R	16.7%	62.1%
If MidHarmonicsEnergy < 0.00001 then R/F = F	10.1%	62.9%

Table 20: Generated Rules for Arousal prediction in Largo BWV1005 (R: Rise, F: Fall)

As we can see from the results showed in the previous pages, the pruned trees classify quite well most of the data but have some problems in test sets for correctly identifying the *falls* which were often misunderstood as *rises*, hence the relatively high values shown in tables 16 and 17.

Anyway, when classifying test data, the most important thing to look at is the resulting average behaviour and, since the training was carried out with only two classes (rise and fall, no “straight” lines) it is understandable that single vectors can be misclassified.

Regarding the generated rules, it is interesting to see whether they can actually predict the emotional response of a listener in a similar piece.

To verify this, we chose the *Preludio* from the the Partita III to test the rules generated previously and related to a fast piece (table 19), and the first half (repeat included) of the *Andante* from the Sonata II for the rules relating to a slow piece (table 20).

The rules were implemented in MatLab and weighted accordingly to their confidence percentage (so rules with higher confidence have stronger weights) then, by feeding the cues to the MatLab file, the system evaluates whether the current vector would provoke a rise or rather a fall in the emotional response of the listener. In this way it is able to propose a possible “arousal” profile that, for the *Preludio*, is shown in figure 39 (the graph is obtained by adding 1 when most of the rules predict a rise, subtracting 1 when the rules predict a fall, doing nothing when there is a tie).

Now we should compare this profile with that of an actual listener and see whether we have similarities.

Subject three of the previous experiment was hired again and had another session like that explained at the beginning of this chapter. His *arousal profile* is shown in figure 40.

Both profiles in figures 39 and 40 have been normalized to 1.

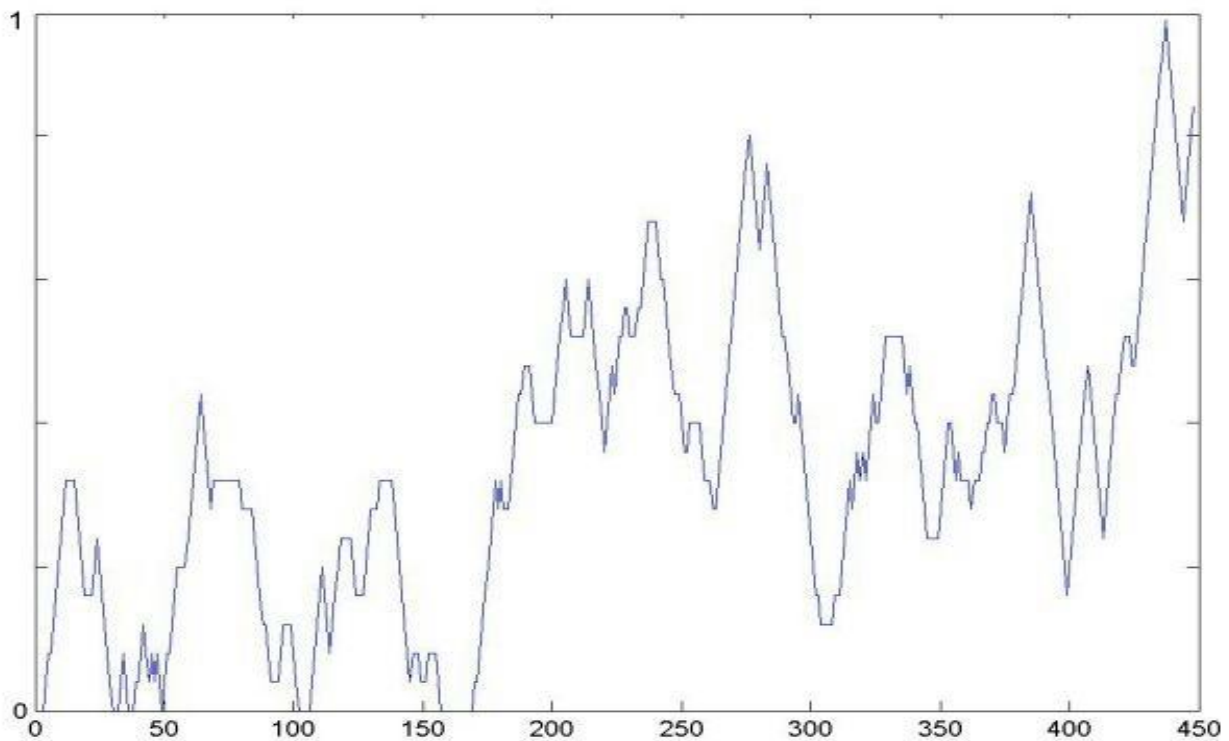


Fig. 39: Arousal profile predicted by the system for Preludio, Partita III BWV1006

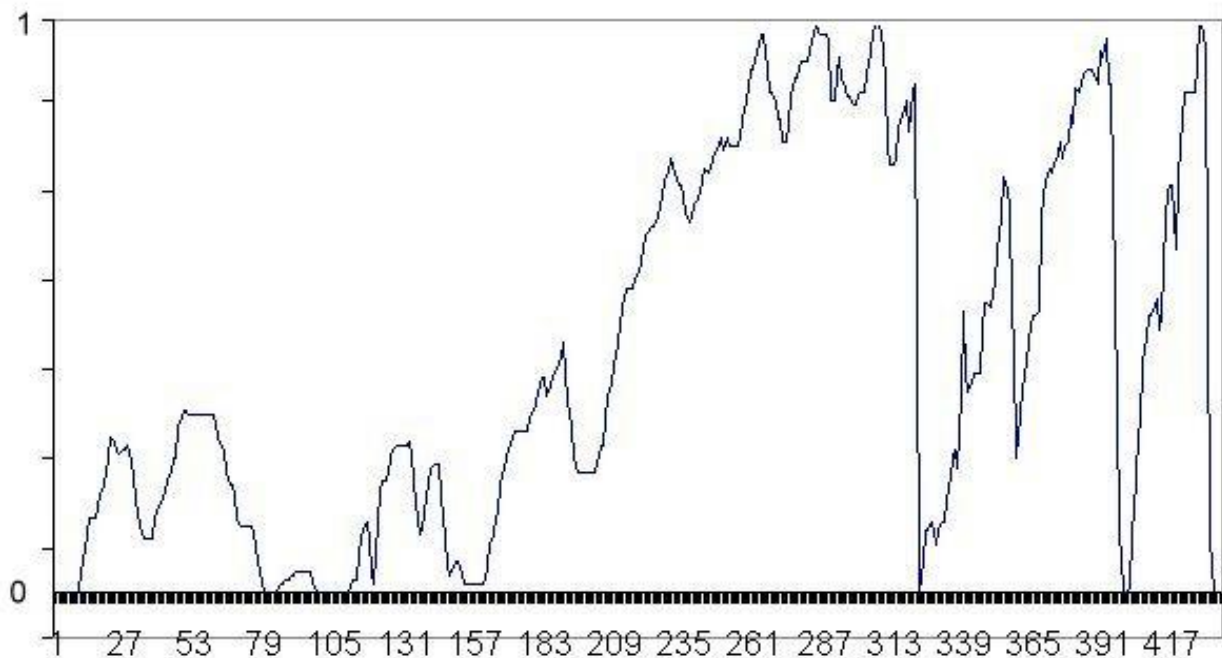


Fig. 40: Arousal profile as recorded by subject three for Preludio, Partita III BWV1006

As we can see, the two profiles in the above figures show strikingly similarities, with the second one slightly delayed due to the response time needed by the subject.

By analysing Ms.BeckerBender performance, we see it starts rather softly and then builds up with a crescendo following the rising pitch progressions in the fast passages of the score. These are the

characteristics that were identified by the system and that, probably, contributed to move the subject who was listening to the performance.

Now let us test the rules generated for the slow movements.

The predicted profile is shown in figure 41 while the subject response is shown in figure 42 (both have been normalized to 1).

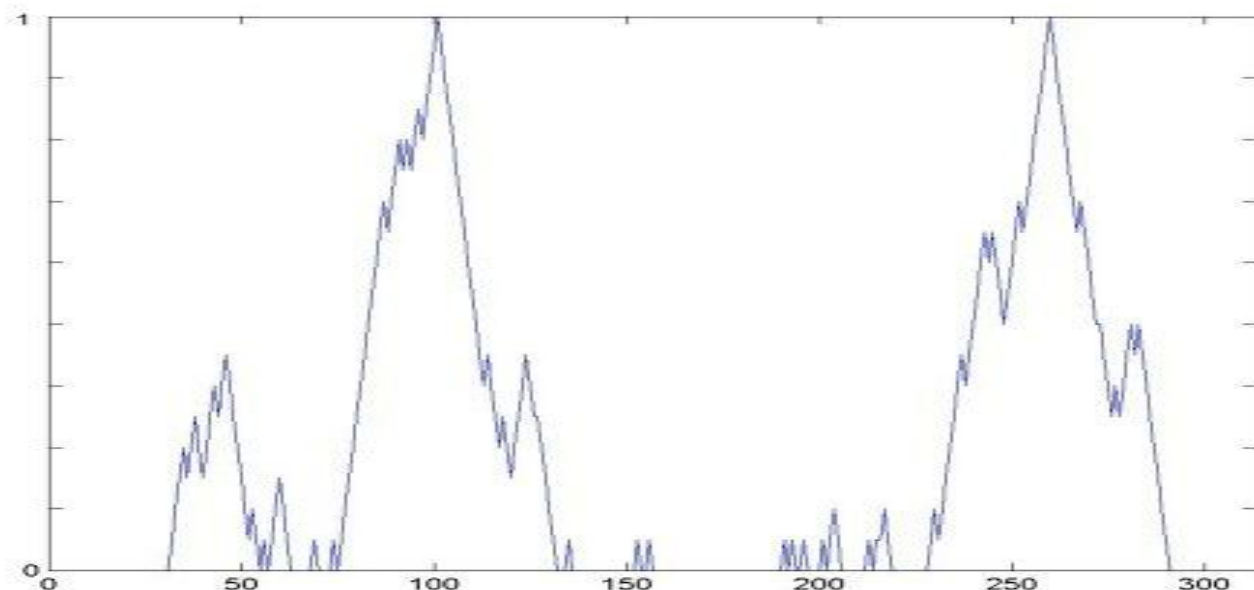


Fig. 41: Arousal profile predicted by the system for Andante (1st Half), Sonata II BWV1003

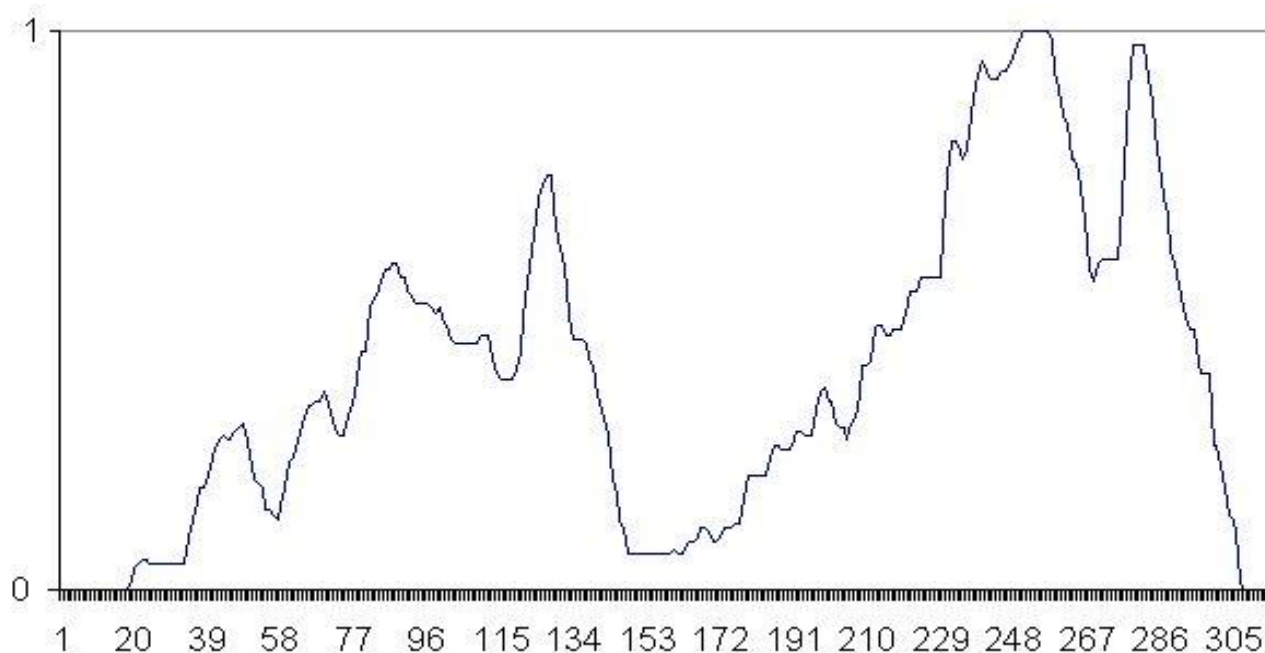


Fig.42: Arousal profile as recorded by subject three for Andante (1st Half), Sonata II BWV1003

Once more the similarities between the two profiles are striking as both of them have the arousal peaks in the same positions showing that the expert system, driven by the previously generated rules, is able to identify the points which are most likely to produce an arousal of emotions in listeners.

Looking at the graphs, it is interesting to note that now, having a slow piece, we do not have the slight delay between the predicted and the measured profiles we noticed earlier. The reason is simply that the fast piece evolves very quickly and the reaction time of the subject being measured is, in that case, enough long to be noticed. This doesn't happen while measuring the slow movement where the changes in the music, and hence in the response, have a much lower rate.

It is also very interesting to compare how the system and the listener faced the repeat of the movement which was performed by the artist emphasizing different aspects such as dynamics and articulation patterns.

At the beginning of the repeat the system identified only some little spots of possible *arousals*, but not enough to bring the emotion measures to a high level as it did during the first time. We should note that the system has no "memory" of what happened when the music was played for the first time. The listener, instead, had this knowledge and, as he noted commenting the results after the experiment, the emotional engagement he felt during the first time influenced and amplified his measurements during the second listening, adding more involvement and taking him to higher arousal levels from the very beginning of the repeat.

Part V

Conclusions and future developments

5.1 Possible uses of cues in actual music making

The experiments we discussed so far suggest several possible applications of the audio cues in real musical environments.

Most of these applications could also be of interest for commercial products: for example the arousal prediction system we talked about in the previous chapter could be used to develop a more complex and flexible system able to predict listeners' emotional engagements in particular pieces and hence it could be used to have predictions about the possible success, i.e. sales, a particular piece/song will have among a targeted audience (a piece that shows high predicted arousal and well defined peaks is more likely to be enjoyed and, hence, be successful).

Such approach could also be useful when querying databases: for example selecting pieces whose emotional profile looks close to that of a given piece, could be a reliable way for suggesting other possible items of interests in a shopping environment such as Amazon or others.

Experiments such as those presented in the first chapters are instead very interesting to show other application possibilities that range from interactive performances where a musician plays along with a computer, modifying the machine behaviour by changing his/her expressive intentions, to a new conception of teaching tools where a computer listens to the students and then is able to give comments on the particular performance regarding their style and, eventually, greet the performer with rewarding comments such as "Well done! You played this Sonata like Glenn Gould!".

Thanks to the continue advancing of the broad band internet technology, teaching tools of this kind could even be implemented in remote environments where the students are coached by a computer system running on a machine located anywhere. A scenario like this would surely bring many benefits to all music lovers who live far away from music schools and are not able to experience traditional teaching methods.

Moreover the possibilities that such cues recognition system offers in a very flexible environment such as the EyesWeb platform should not be forgotten.

EyesWeb, in fact, gives the artists the possibility of making their own patches where the cues could be used for identifying particular aspects of their playing to control a desired effect in real time. Some very basic patches, that could be used as a starting point for developing more complex tools by advanced users, are presented in Appendix B.

Of course, to achieve these kind of results, a lot of work is still needed to refine the research and make the systems more general and robust. Anyway the results obtained during the development of this research are very interesting and promising and show that the audio cues being selected are really meaningful to objectively describe and quantify several aspects of music playing that, so far, could only be explained in often ambiguous and subjective words such as those used for describing style characteristics or particular moods and "sound colours" in music playing.

Appendix A

The EyesWeb Blocks in detail

BigBuffer



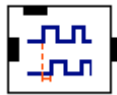
This block implements a FIFO structure for SoundBuffer datatypes.

Input: SoundBuffer

Output: SoundBuffer

Parameters: Buffer Length (in seconds), Buffer Mode (Normal, Overlapped)

DelayLine



This block delays an input SoundBuffer of the amount specified.

Input: SoundBuffer

Output: SoundBuffer

Parameters: Value (the amount we want the input soundbuffer be delayed of. Its type is specified in the following parameter), Type (Samples, Milleseconds)

LogBuffer



This block computes the logarithm (base 10) of an input SoundBuffer.

Input: SoundBuffer

Output: SoundBuffer

Parameters: none

Norma



This block normalizes an input SoundBuffer.

Input: SoundBuffer

Output: SoundBuffer

Parameters: Max (the value we want to normalize the input. Usually set to 1)

SoundDiff



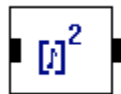
This block computes basic operations (+, -, *, /) between two input SoundBuffers.

Input: SoundBuffer1, SoundBuffer2

Output: SoundBuffer

Parameters: Operations (+, -, *, /)

Squared



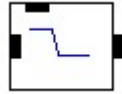
This block squares an input SoundBuffer.

Input: SoundBuffer

Output: SoundBuffer

Parameters: None

XLowPass



This block implements the 1st order IIR low pass filter as described in page 9 (equation 2).

Input: SoundBuffer

Output: SoundBuffer

Parameters: FC (Cut off frequency), N (number of filters to put in cascade)

AudioCues



This block extracts the set of audio cues, time window approach, as described in page 9.

Input: SoundBuffer1, SoundBuffer2

Output: Scalar1 (tempo), Scalar2 (standard deviation of articulation), Scalar3 (mean of articulation), Scalar4 (sound level difference), Scalar5 (mean of sound level), Scalar6 (mean of attack velocity), Scalar7 (tempo2, i.e. notes per second), Scalar8 (standard deviation of sound level)

Parameters: Filename (specifies a text file name where to save the cues. Writing “nofile”, no files will be written and data will not be saved)

NoteCues



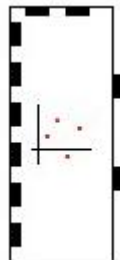
This block extracts a set of audio cues, event triggered approach, as described in page 10.

Input: SoundBuffer1, SoundBuffer2, Scalar

Output: Scalar1 (sound level), Scalar2 (sound max), Scalar3 (articulation), Scalar4 (IOI), Scalar5 (attack velocity), Scalar6 (flag for note detection)

Parameters: Filename (specifies a text file name where to save the cues. Writing “nofile”, no files will be written and data will not be saved), Step (time interval, in ms, for taking samples from the incoming soundbuffers), Mode (specifies if the block works having only the energy profiles as inputs or if it should use also the data from the scalar input for improving note detection)

ACues



This block implements the multidimensional scaling described in page 14.

Input: Scalar1, Scalar2, Scalar3, Scalar4, Scalar5, Scalar6

Output: Scalar1 (X coordinate), Scalar2 (Y coordinate)

Parameters: Style (specifies the angles to be used during the compression algorithm. So far only the recorder scheme is implemented but others can be added), Step (specifies the maximum amount, in pixels, for moving the cursor in the 2D space)

SampleToFreq_Midi



This block extract the fundamental frequency of an incoming sound by using a zero crossing algorithm.

Input: SoundBuffer

Output: Scalar

Parameters: Mode (specifies whether the output should be expressed as frequency in Hz or Midi value)

Energia



This block computes the energy of the incoming soundbuffer giving the user very high flexibility in defining the frequency bands where the energy will be computed

Input: SoundBuffer, Matrix (optional, specifies the frequency bands)

Output: Matrix (displays the list of the frequency bands and the energy computed in each of them)

Parameters: DimFFT (FFT Dimension), FileName (the file where the results will be saved), WriteFile (flag for specifying whether the file with the results should be saved or not), CompactLog (specifies whether the results should be saved in a short or verbose format), NumBands (number of frequency bands to split the spectrum in), BandType (specifies if the spectrum should be divided linearly, logarithmically or using an input matrix or a separate input text file), OutputBands (specifies the max number to be displayed in the output matrix)

Plucked_1_1



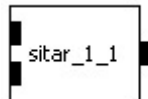
This block implements in EyesWeb a physical model of a stringed plucked instrument using the well known Karplus-Strong Algorithm (Karplus, Strong 1983) and based on the C++ classes proposed in the STK (Cook, Scavone 1999).

Input: Scalar1 (frequency), Scalar2 (amplitude)

Output: SoundBuffer

Parameters: LowestFrequency (specifies the lowest possible frequency in Hz)

Sitar_1_1



A variant of the Plucked_1_1 block, this implements in EyesWeb a physical model of a sitar instrument using the Karplus-Strong Algorithm (Karplus, Strong 1983) and based on the C++ classes proposed in the STK (Cook, Scavone 1999).

Input: Scalar1 (frequency), Scalar2 (amplitude)

Output: SoundBuffer

Parameters: LowestFrequency (specifies the lowest possible frequency in Hz)

Appendix B

Basic EyesWeb patches for interactive performances

This appendix shows some possible EyesWeb patches that use the blocks and the ideas explained in the thesis. These patches can be useful as a starting point to artists or advanced EyesWeb users who want to start developing their own patches using the cues for controlling particular audio, video or other special effects for interactive artistic performances.

For simplicity's sake, a cue extractor patch such as the one shown in figure 4 (page 8) is not included and the cues, or the output of any elaboration we want to make on them by means of HMMs, expert systems or other tools, will be directly connected to the blocks instead of the simple sliders presented in the following figures.

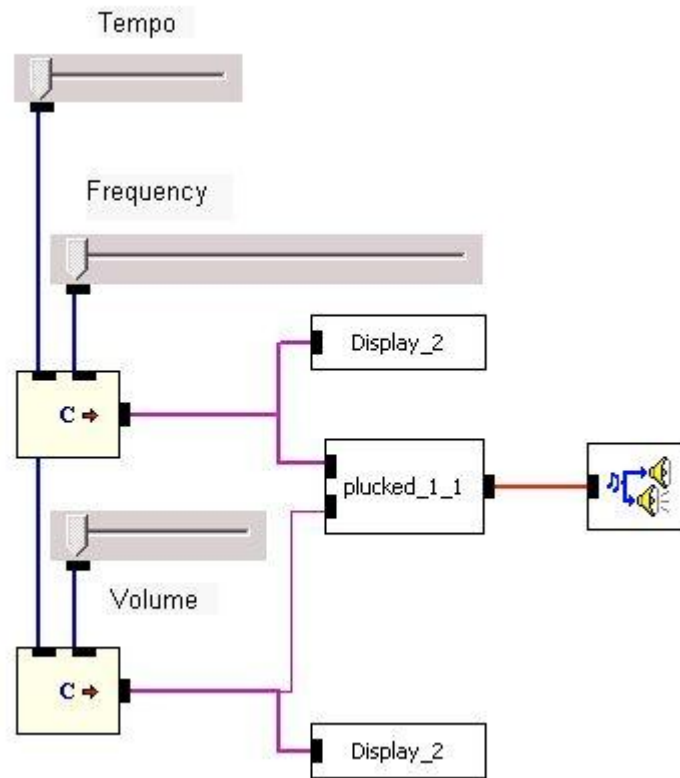


Fig. 43: a very simple patch for controlling in real time the Plucked_1_1 block

In Figure 43 we have a very basic patch that will be used as a starting block for those shown in figures 44 and 45. Very simply, it can control in real time the frequency, volume and tempo (i.e time between two following notes) of the note generated by a physical model of a plucked string instrument (see pag.59).

Figure 44 shows a patch that takes this module and builds a system for generating a random note among those specified in the scalar-generator blocks. These are inserted in a vector and then random number generator selects the particular entry. Volume and Tempo can be controlled in real time by particular cues so to react to particular effects.

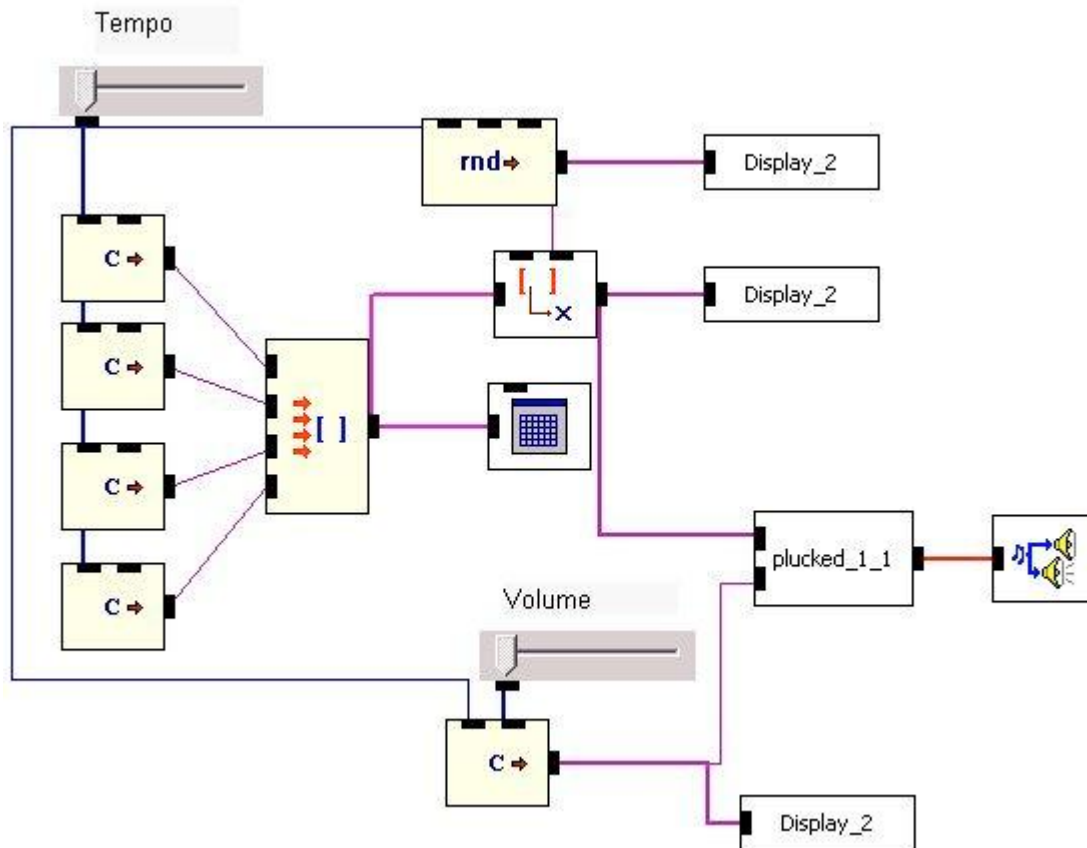


Fig.44: a patch that select a note at random among those defined in a vector.

Figure 45 develops the previous patch further: in this case the notes generated by the Plucked block are coming from a C major scale. The particular note is chosen at random and the cues can control the tempo, volume and shift the note's octave.

Figure 46 instead shows a patch where cues can control the low and high cut off frequencies to band pass a white noise signal (generated by the "rand" signal block). This can give the opportunity to a performer to play along and control in real time "wind like" effects during his/her performance.

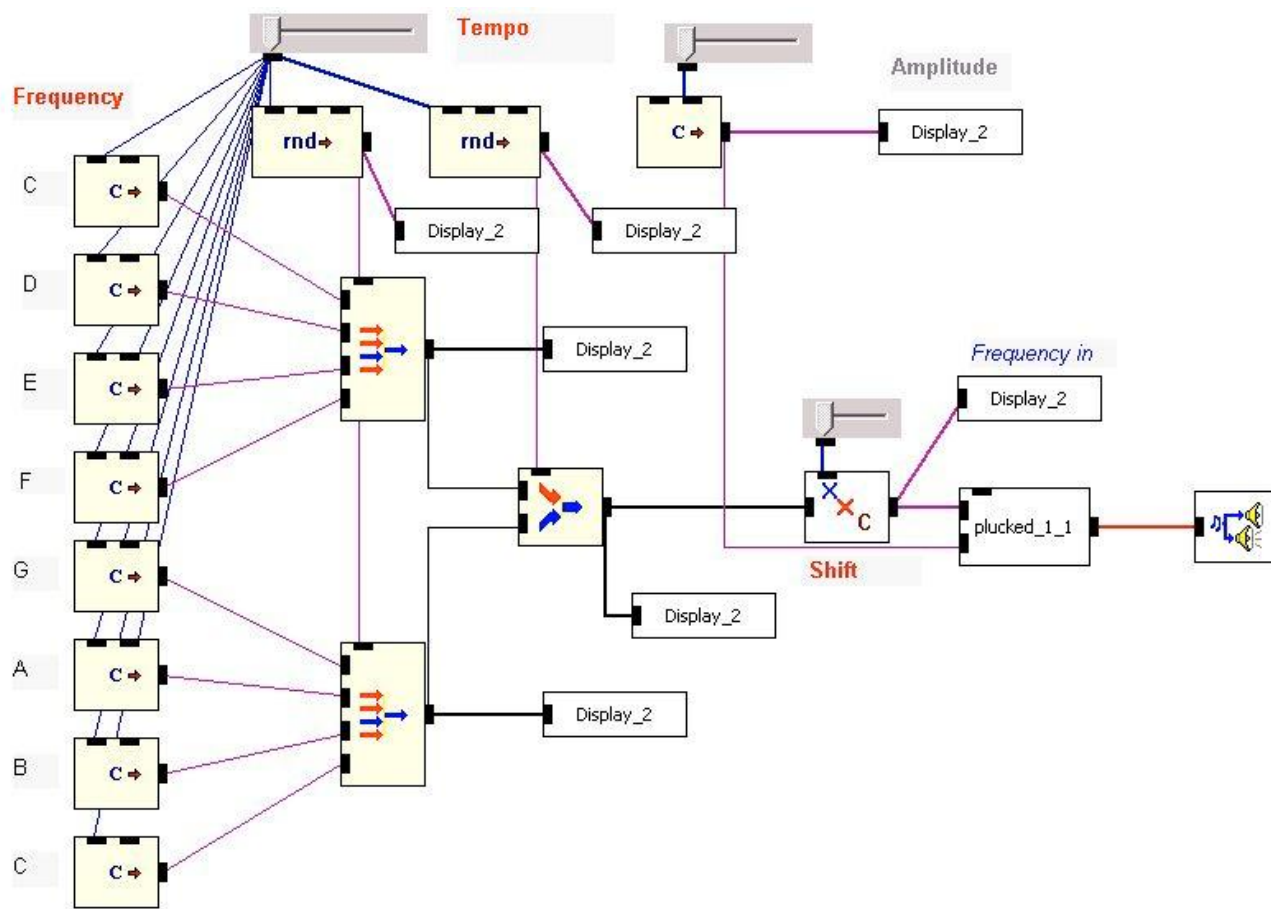


Fig.45: A patch generating notes taken from a C Major scale.

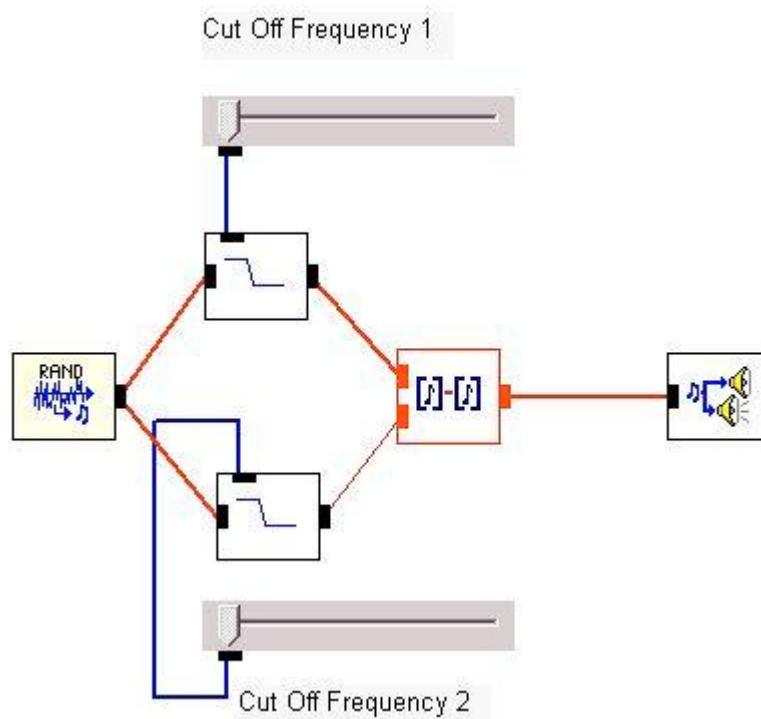


Fig.46: Filtering white noise for particular effects

References:

- Barbier P. (1999):** The world of the Castrati: the history of an extraordinary operatic phenomenon, *Souvenir Press*
- Battel G.U., Fimbianti R. (1998):** How communicate expressive intentions in piano performance. *Proceedings CIM 1998, Gorizia, Italy*, pp.67-70
- Berri P. (1962):** Paganini: Documenti e testimonianze, *Genova*
- Camurri A., De Poli G., Leman M. (2001):** MEGASE – a multisensory expressive gesture application system environment for artistic performances, *Proceedings CAST 01, GMD, St.Augustin-Bonn, Germany*, pp.59-62
- Camurri A., Dillon R., Saron A. (2000):** An experiment on analysis and synthesis of musical expressivity. In *Proceedings CIM 2000, L'Aquila, Italy*, pp. 87-91
- Camurri A., Hashimoto S., Ricchetti M., Suzuki K., Trocca R., Volpe G. (2000):** EyesWeb – Toward gesture and affect recognition in dance/music interactive systems, *Computer Music Journal*, 24 (1), MIT
- Canazza S., De Poli G., Di Sanzo G., Vidolin A. (1998):** Adding expressiveness to automatic musical performance. *Proceedings CIM 1998, Gorizia, Italy*, pp.71-74
- Casazza R., Pertino A. (2003):** Modelli per l'analisi del coinvolgimento emotivo di spettatori esposti a stimoli musicali, Master Thesis, DIST - University of Genoa
- Cook P., Scavone G. (1999):** The Synthesis Toolkit, *Proceedings ICMC'99*
- Cooke D. (1959):** The Language of Music, *Oxford University Press*
- Crow E., Davis F., Maxfield M. (1960):** Statistics Manual, *Dover Publications Inc., New York*
- De Poli G., Rodà A., Vidolin A. (1998):** Note-by-note analysis of the influence of expressive intentions and musical structure in violin performance. *Journal of New Music Research Vol.27 N.3, 1998*, pp.293-321
- Dillon R. (2001):** Extracting audio cues in real time to understand musical expressiveness. In *Proceedings "Current research directions in computer music", MOSART Workshop, Barcelona, Spain*, pp.41-44
- Dillon R. (2003):** A statistical approach to expressive intention recognition in violin performances. *Proceedings SMAC 03, Stockholm, Sweden*, pp. 529-532
- Dillon R. (2003b):** Classifying musical performance by statistical analysis of audio cues, *Journal of New Music Research, Vol. 32 n.3*, pp.327-332
- Friberg, A., Colombo, V., Frydén, L. & Sundberg, J. (2000):** Generating Musical Performances with Director Musices. *Computer Music Journal, vol. 24, no. 3.*, MIT Press pp. 23-29

- Friberg A., Schoonderwaldt E., Juslin P. & Bresin R. (2002):** Automatic Real-Time Extraction of Musical Expression. In *Proceedings ICMC 2002*, pp.365-367, Goteborg, Sweden
- Gabrielsson A. (1995):** Expressive intention and performance. *Music, Mind and Machine*, New York, Springer (R. Steiner, editor), pp.35-47
- Gabrielsson A., Lindstrom E. (2001):** The influence of musical structure on emotional expression. In Juslin P. & Sloboda J. (eds.): *Music and Emotion: theory and research*, Oxford University Press, London (pp.223-248)
- Ghahmarani Z. (2001):** An introduction to Hidden Markov Models and Bayesian Networks. *International Journal of Pattern Recognition and Artificial Intelligence*, Vol.15(1), pp.9-42
- Gremo F. (2002):** Metodi e strumenti software per l'analisi di spettatori soggetti a stimoli musicali, Master Thesis, DIST - University of Genoa
- Guhr W. (1830):** Über Paganini's Kunst, die Violine zu spielen, Mainz
- Juslin P. (1997):** Emotional Communication in music performance: a functionalist perspective and some data. *Music Perception*, Vol.14 pp.383-418
- Juslin P. (2000):** Cue utilization in communication of emotion in music performance: relating performance to perception. In *Journal of Experimental Psychology: Human perception and performance*, 26 (6), (pp.1797-1813)
- Juslin P. (2001):** Communicating emotion in music performance: a review and a theoretical framework. In Juslin P. & Sloboda J. (eds.): *Music and Emotion: theory and research*, Oxford University Press, London (pp.309-337)
- Karplus K., Strong A. (1983):** Digital Synthesis of plucked string and drum timbres, *Computer Music Journal*, vol. 7, no. 2, pp. 43-55
- Krumhansl C.L., Schenck D.L. (1997):** Can dance reflect the structural and expressive quality of music? A perceptual experiment on Balanchine's choreography of Mozart's divertimento No.15. *Musica Scientiae*, vol. I, pp.63-85
- Marolt M., Villon O., Camurri A. (2004):** On Modelling Emotional Engagement of Listeners with Performances of a Skrjabin Etude (*in press*)
- Palmer C. (1997):** Music Performance. *Annual Review of Psychology*. Vol. 48, pp.115-138
- Papoulis A. (1991):** Probability, Random Variables and Stochastic Processes. *McGrawHill*, 3rd Edition, pp.214-221
- Picard R. (1997):** Affective Computing, *MIT Press*
- Quinlan J.R. (1993):** C4.5: Programs for Machine Learning, *Morgan Kaufmann, San Mateo*
- Rabiner L.R., Juang B.H. (1986):** An introduction to Hidden Markov Models. *IEEE ASSP Mag.*, Jun 1986, pp.4-16

- Sammon J.W. (1969):** A nonlinear mapping for data structure analysis, *IEEE Transactions on Computer Science*, 18(5), pp.401-409
- Scherer K. R. (2003):** Why music does not produce basic emotions: pleading for a new approach to measuring the emotional effect of music. *Proceedings SMAC 03, Stockholm, Sweden*, pp. 25-28
- Sloboda J. (1991):** Music structure and emotional response: Some empirical findings. *Psychology of Music*, vol. 19, pp.110-120
- Stamatatos E. (2002):** Quantifying the differences between music performances: score vs. norm, *Proceedings ICMC 2002, Goteborg, Sweden*
- Stamatatos E., Widmer G. (2002):** Music performer recognition using an ensemble of simple classifiers. *Proceedings 15th ECAI, Lyon, France*
- Sundberg J., Friberg A., Fryden L. (1991):** Common secrets of musicians and listeners: an analysis-by-synthesis study of musical performance. In *Representing Musical structure*, Academic Press
- Timmers R., Camurri A., Volpe G. (2003):** Performance cues for listeners' emotional engagement. *Proceedings SMAC 03, Stockholm, Sweden*, pp. 569-572
- Todd N.(1992):** Music and Motion: a personal view. In *Proceedings 4th Workshop on Rhythm Perception, Bourges, France*
- Widmer G. (2001):** Using AI and machine learning to study expressive music performances: project survey and first report. *AI Communications*, 14(3), pp.149-162
- Zanon P., Widmer G. (2003):** Learning to recognize famous pianists with machine learning techniques. *Proceedings SMAC 03, Stockholm, Sweden*, pp.581-584