Expressive Processing of Audio and MIDI Performances in Real Time*

Sergio Canazza Giovanni De Poli Riccardo Di Federico Carlo Drioli
Antonio Rodá
Centro di Sonologia Computazionale (CSC-DEI), University of Padua, Italy
canazza@dei.unipd.it

Abstract

A framework for real-time expressive modification of audio and MIDI musical performances is presented. An expressiveness model computes the deviations of the musical parameters which are relevant in terms of control of the expressive intention. The modifications are then realized by the integration of the model with a sound processing engine and a MIDI synthesis device.

Introduction

In multimedia systems the musical media is essentially based on pre-recorded audio, which can be played by the listener without modification.

Recently, a growing interest has been demonstrated in the field of expressive signal processing [1],[4] and mathematical and perceptual models have been proposed which compute the desired deviations in order to add the desired expressive intentions to a neutral performance [2]. These models usually decides their action on the base of a symbolic description of the performance, such as the MIDI description, and give the result in term of deviations of parameters of the same description. In this work we propose to perform the expressive musical transformations on both MIDI and digitally recorded performances.

The rendering in the last case is performed by means of a sound processing framework integrated with the expressiveness model. In this way, the system performs expressive manipulations and high quality signal processing of sound in an organized way, interpreting the symbolic deviations related to expressiveness in term of sound transformations.

1 Architecture

The functional structure of the system is reproduced in Figure 1. The input of the expressiveness model is composed by a description of an neutral musical performance (played without any expressive intention), the nominal score of the performance, and a control of the expressive intention desired by the user. The neutral performance is usually made of a polyphonic accompaniment described by a MIDI file, and by a digitally recorded monophonic part. A symbolic (MIDI-like) description of the audio performance is derived by analysis of the recorded signal, containing all the musical parameters needed by the model (note onset and offset, time definitions of attack-sustain-release of notes, information on higher level attributes). The expressiveness model acts on the symbolic level, computing the deviations of all musical parameters involved in the transformation and finally driving the MIDI synthesizer section and the audio processing engine. In the last case, a re-synthesis procedure is involved in order to reproduce the sound from its sinusoidal representation. All the sound transformations addressed by the model are realized at re-synthesis time by means of a frame-rate parametric control of the audio effects.

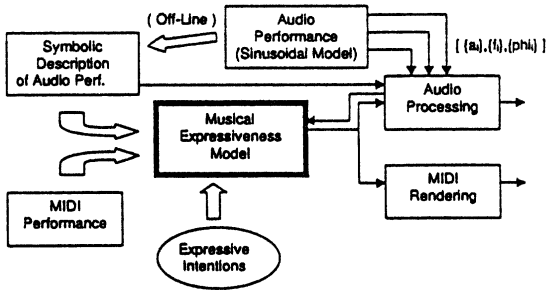

Figure 1: System architecture

2 The model

The expressiveness model was obtained starting from perceptual and acoustic analysis carried out on several recorded performances with different expressive intentions. Factor analysis on the listeners judgements showed that the performances were arranged in a bi-dimensional space, in which the axis are correlated mainly with the kinetics of music and with the energy of sound. When

* This research has been supported by Telecom Italia under the research contract *Cantieri Multimediali*

	N	Ha	S	He	L	B	D
δIOI		>			<<	>	>>
δL				>	<	<<	
δDRA		<<			>	<	<
δI		>>	<	>	<		<<
δEC			>>		<		>

Figure 2: Qualitative parameters changes for Neutral, Hard, Soft, Light, Bright and Dark performances. The deviations, referred to 1, are expressed by means of multiplicative factors, so that $\delta DRA = 1.2$ describes a 20% lengthening of the attack

the user moves in the space, the model computes the changes of a set of musical parameters, that are next summarized. With reference to Figure 4, IOI is the *Inter Onset Interval* between the current and the successive note, DR is the duration of the current note, DRA is the duration of the attack, the *Intensity* I is the mean envelope energy, the *Envelope Centroid* EC is the time location of the energy envelope center of mass. The definition of *Legato* is assumed to be $L = DR/IOI$. If a pause is written in the nominal score, the IOI of the note is estimated with the the formula $IOI = IOI' DR_n/(DR_n + DR_{n,p})$, where IOI' is the interval between the current note and the next note (including the pause), DR_n is the nominal duration of the current note and $DR_{n,p}$ is the nominal duration of the pause. With this definition, *Legato* between two notes gives $L > 1$, while different degrees of staccato gives $L < 1$ and are determined by the *micropause* between the two notes. Figure 2 summarizes the qualitative rules of the model (here, δ stands for a deviation of the corresponding musical parameter). More details on the research aspects and on the realization of the expressiveness model can be found in [2].

3 The sound analysis and processing framework

Audio processing is often aimed to change the acoustical or perceptual characteristics of the sounds. To modify the expressive intention of a recorded performance, however, an organized control of the audio effects is necessary which provides task specific high-level control. This approach is found, for example, in [1], where a case-based reasoning system is responsible for the evaluation of transformations realized in terms of basic audio effects.

Our expressiveness model is suited to produce the time-varying controls of the sound processing engine, focusing on a wide class of musical signals, namely monophonic and quasi-harmonic sounds such as wind instruments and solo string instruments.

The audio processing techniques involved in this stage are based on a *Sinusoidal* model of the input signal. First introduced by McAulay and Quatieri for speech coding [3], in the last decade it showed to be one of the most flexible tools for sound analysis and resynthesis. The estimation of frequencies, amplitudes and phases of sinusoidal components involves a Short Time Fourier Transform (STFT) analysis, followed by a peak picking algorithm. The result of this procedure is the identification of time evolving parameters.

Resynthesis of sound is performed by inverse FFT to synthesize overlapping output sound frames. This approach presents a much higher computational efficiency if compared to the classical additive synthesis, and therefore is preferred for real-time applications.

All the principal sound effects are obtained by control on the parameters of the sinusoidal representation, and are briefly summarized. *Time stretching* is obtained by changing the frame rate of resynthesis and by interpolating between the parameters of two frames in case of non-integer step. *Pitch shift* is obtained by scaling the frequencies of the harmonics and by preserving formants with spectral envelope interpolation. *Amplitude envelope* control is made by scaling of partial amplitudes, expressed in dB, with an additive constant b_{ampl}. *Brightness* control is performed by a spectral transformation function (see Figure 3), which emphasize (or de-emphasize) the spectrum. The reported function is added to the magnitude (in dB) of the original spectrum, and its shape is controlled with a scaling multiplicative factor a_{br}. The formula that resumes the control on amplitude and brightness is

$$H(f) = a_{br} G(f) + b_{ampl} \qquad (1)$$

where $G(f)$ is the function plotted in Figure 3.

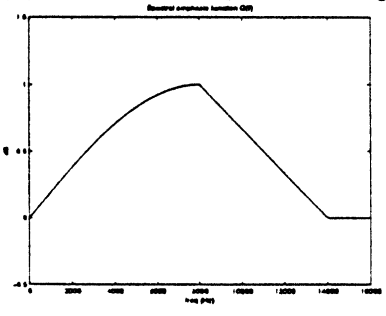

Figure 3: Function for spectral emphasis

Finally, some higher level sound processing, such as vibrato and tremolo control, is performed with specific algorithms, as detailed in [4]. This processing proved to be necessary whenever time stretching must be applied to sound segments originally characterized by vibrato or tremolo, in order to avoid the alteration of the vibrato and tremolo rate. This procedure relies on a first step which

identifies and subtract the feature on the sinusoidal representation, and on a second step which applies back the feature with the correct parameters on the stretched segment.

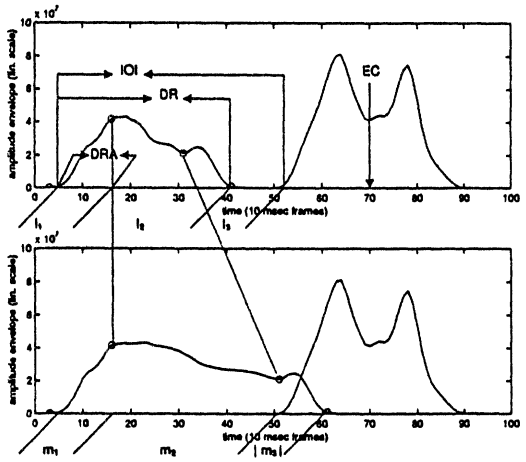

Figure 4: Musical parameters involved in the control of expressiveness. A modification of the *Legato* parameter is also shown

Among its properties, the sinusoidal representation of the sound offers a powerful analysis framework for the extraction of musical features at various level. A symbolic description of the performance is created with an off-line procedure, containing for each note all the musical parameters involved in the control of expressiveness.

The rendering of the deviations computed by the model may imply the use of just one of the basic sound effects seen above, or the combination of two or more of these effects, with the following general rules:

Local Tempo: time stretching is applied to each note. The analysis conducted on real performances with different expressive intentions, revealed that for strings and winds the duration of the attack is perceptually relevant for the characterization of the conveyed expressive intention. For this reason, a specific time stretching factor is computed for the attack segment and is directly related to the δDRA indicated by the model. The computation of the time stretch control on the note relies on the cumulative information given by the δDRA and δIOI factors, and on the δDR deviation induced by the *Legato* control considered in the next item.

Legato: this musical feature is recognized to have great importance in the expressive characterization of wind and string instruments performances. However, the processing of *Legato* is a critical task that would imply the reconstruction of a note release and a note attack if the notes are originally tied in a *Legato*, or the reconstruction of the transient if the notes are originally separated by a *micropause*. In both cases, a correct reconstruction requires a deep knowledge of the instrument dynamic behaviour, and a synthesis framework would be necessary. Our approach to this task is to approximate the reconstruction of transients by interpolation of amplitudes and frequency tracks.

The deviations of the *Legato* parameter are processed by means of two synchronized actions: the first effect of a *Legato* change is a change in the duration δDR of the note, while is $L' = L\ \delta L = (DR\ \delta DR)/(IOI\ \delta IOI)$ and $\delta L = \delta DR/\delta IOI$. This time stretching action must be added to the one considered for the *Local Tempo* variation, as we can see in detail. Three different time stretching zones are distinguished within each note (with reference to Figure 4): attack, sustain and release, micropause. The time-stretching deviations must satisfy the following relations:

$$m_1 = l_1\ \delta DRA \tag{2}$$

$$m_1 + m_2 = (l_1 + l_2)\ \delta DR \tag{3}$$

$$m_1 + m_2 + m_3 = (l_1 + l_2 + l_3)\ \delta IOI \tag{4}$$

and each region will be processed with a time stretch coefficient K computed from the above equations:

$$K_1 = \frac{m_1}{l_1} = \delta DRA \tag{5}$$

$$K_2 = \frac{m_2}{l_2} = \frac{(l_1 + l_2)\delta L\ \delta IOI - K_1 l_1}{l_2} \tag{6}$$

$$K_3 = m_3/l_3 = -(l_1+l_2)\delta L\ \delta IOI + (l_1+l_2+l_3)\delta IOI \tag{7}$$

Equation 7 can give a negative time stretch coefficient if an overlap occurs due to the lengthening of the actual note. In this case the second action involved is a spectral linear interpolation between the release of the actual note and attack of the next note over the intersection of the two (see figure 5). The overlap region length is determined by the *Legato* degree, and the interpolation within partial amplitude will be performed over the whole range.

The frequency tracks of the sinusoidal representation are prolonged to reach the pitch transition point. Here, a $10-15\ msec$ transition is generated by interpolating the tracks of the actual note with the ones of the successive. In this way, a transition without *glissando* is generated. Glissando effect can be controlled by varying the number of interpolated frames.

This procedure, used to reproduce the smooth transition when the stretched note overlaps with the following note, is a severe simplification of instruments transients, but is sufficiently general and efficient for real-time purposes.

Intensity: amplitude envelope control and brightness control are used together to reproduce

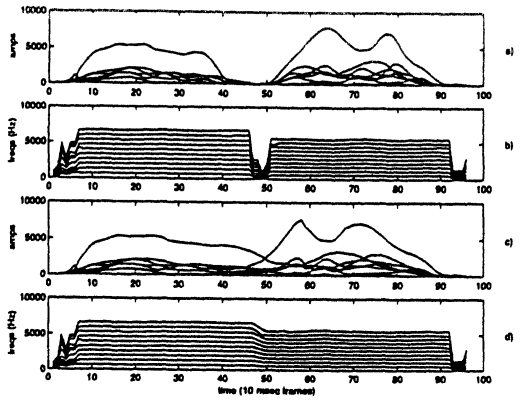

Figure 5: *Legato* of two notes originally separated by a *micropause* (only the first ten partials of the sinusoidal analysis are shown)

the natural relations between spectrum and intensity. The perceptual differences between expressive intensions like *bright* and *dark* have shown to be strictly related to the differences in the brightness of the sound. A natural way to control the brightness is to act on the spectral centroid (defined as the center of mass of the spectrum). For this purpose, equation 1 is used, where the emphasis and amplitude parameters a_{br} and b_{ampl} are controlled by a proportional low ($a_{br} = K_a\ \delta I$, $b_{ampl} = K_b\ \delta I$).

Envelope Shape: The center of mass of the amplitude envelope is related to the musical *accent* of the note, which is usually located on the attack for *Light* or *Heavy* intentions, or close to the end of note for *Soft* or *Dark* intentions. To change the position of the center of mass, a triangular-shaped function is applied to the energy envelope, where the apex of the triangle correspond to the new position of the accent.

4 Aspects related to real-time implementation

One of the objective of the system is to give the user the possibility to interact with the model of expressiveness by moving in the perceptual space and feel the changes rendered by the sound processing engine. This real-time approach requires the management of some aspects related to the information exchange and the synchronization between the model and the processing engine. The communication protocol between the model and the sound processing engine must be bidirectional, in order to let the model know the exact time-position of the processing engine, which is constrained to follow the time evolution of the underlying signal. The model can here be compared to a human director who gives his directions to the performing orchestra by listening to the current performance characteristics. This can produce delayed responses of the system to the user input.

A further time delay in the processing action, with respect to the user requests, is due to the fact that when the system is performing the note n, signal processing action may be already decided for the note $n+1$ (the *Staccato - Legato* processing is an example). The user control input will then be allowed to produce his effect starting from the note $n+2$. The delays in the sound processing actions are then reflected to the MIDI driving actions to preserve the synchronization.

A distributed architecture has been chosen for the system, in which the communication between the model and the sound processing is realized by a client-server paradigm. The model (client) relies on the services of the sound rendering engines (servers). With this architecture, when more digitally recorded instruments are to be processed, the high computational charge due to the sound processing algorithms can be distributed on different processing units. In our implementation, a socket-based communication solution has been addressed, leading to a network oriented distributed system. This solution performs well on fast local networks (LAN), and has interesting properties in terms of modularity and extensibility.

5 Conclusions

A system for real-time control of expressive intentions in musical performance has been presented. The system can manage musical performances in which MIDI and audio sections are synchronized. The main result of this research is the integration of an expressiveness model with a sound processing engine, for the control of expressiveness. Although simplified transformations on instrument transients where assumed for real time implementation, the system proved the feasibility of interactive transformations of expressiveness in digitally recorded performances.

References

[1] J. L. Arcos, R. L. de Mántaras, and Xavier Serra, "Saxex: A case-based reasoning system for generating expressive musical performances," *J. New Music Research*, 27(3), 194–210, 1998.

[2] S. Canazza and A. Rodá, "A parametric model of expressiveness in musical performance based on perceptual and acoustic analyses," *Proc. ICMC 99*, 1999.

[3] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 34(4), 744–754, 1986.

[4] R. Di Federico and C. Drioli, "An integrated system for analysis-modification-resynthesis of singing," *Proc. IEEE SMC 98 Conf.*, 1254–1259, 1998.