Playing with virtual musicians: the Continuator project

FRANÇOIS PACHET
Sony Computer Science Laboratories-Paris

Introduction

Playing with musicians, as if they were there, but without their physical presence, is an oldand strong fantasy. Indeed, wouldn't it be nice if you could play with your favourite – andinaccessible - Jazz musician? If you could play with him, or, rather, like him, but stillretaining your own musical intentions? Why not then play with a virtual orchestra of musicians? Going further (or closer), what about playing with virtual copies of yourself?

The Continuator project conducted at Sony CSL addresses these issues, by developing techniques that capture efficiently stylistic information, and by designing musical instruments that make these techniques easily useable, and seamlessly integrated in real world music playing contexts. The current Continuator prototype is able both to learn quickly arbitrary musical styles in real time, and to generate music consistent with these styles, while remaining intimately controllable. We describe here the main technical issues at stake, and the new playing modes that are made possible by the resulting system.

Learning musical styles automatically

The technical issue of learning automatically and in an agnostic manner a musical "style" has long been addressed by researchers in the community of artificial intelligence and information theory. Shannon introduced in his seminal paper in 1948 the concept of information based on probability of occurrence of certain messages. This notion was quickly used to model musical styles, for instance by (Brooks et al., 1957). These experiments showed that it was possible to create pieces of music that would *sound like* given styles, by simply computing and exploiting probabilities of note transitions. More generally the notion of Markov chains was applied to music in a number of ways. One of the most spectacular applications of Markov chains to music is probably (Cope, 1996), although his musical productions are not entirely produced automatically. A good survey of state-of-the-art of Markov based techniques for music can be found in Triviño-Rodriguez *et al.* (2001), including in particular variable-length Markov models, which capture more finely stylistic information.

In our context, we want to learn and imitate musical styles in a faithful and efficient manner. This raises a number of technical issues.

First, *efficiency*. The style learning and music generation mechanisms should be efficient enough so that the music produced by the system is seamlessly integrated in the playing mode of the musician. As an example, in the standard continuation mode, the system produces a continuation of the phrase played by the musician. The real time constraint is that the continuation be produced in less time than the average inter onset time of the input notes. To take a concrete example in Jazz, we analysed phrases played by John McLaughlin (see *e.g.* http://www.musicindustries.com/axon/archives/john.htm, said to be one of the fastest Jazz guitarists, and found a minimum inter onset time of about 60 milliseconds. This figure gives an approximate constraint for the computation time of our system: it should be able to learn and produce sequences in less than 30 milliseconds. We will not give the details of the implementation of our learning module here, but it was shown to learn and produce continuations for large corpuses of phrases in less than 5 milliseconds with a Java prototype running on a Pentium III laptop.

Second, the management of *imprecision*. Musical phrases in practice are far from "perfect". Academic works in musical Markov chains consider learning corpuses as sets of exact strings. The problem with real music is that musicians do not always play musical phrases exactly the same way. There are musical variations, as well as errors. These are not explicitly taken into account by Markov models. We have introduced a scheme that allows the system to learn simultaneously several Markov models with various degrees of precision. When the most precise model does not match an input phrase, the system looks for a continuation in a less refined model. For instance the most refined model will take into consideration the exact pitch, duration and velocity of each note making up the phrase. A less refined model will consider only pitch regions (i.e. C0 to E0 are the same item, then F0 to A0 are another item, etc.). This scheme makes it possible in practice to match input sentences that were never learnt before and to produce continuations which are optimally consistent.

There are other important technical issues to ensure that the learning and production mechanisms fit with the constraints of music (improvised music in particular), such as the management of polyphony and the segmentation of input phrases. These issues are not discussed here for reasons of space, and are addressed by Pachet (2002).

Turning a music automaton into a musical instrument

The learning scheme outlined above makes it possible to learn quickly and efficiently arbitrary musical styles. However, the goal is to build an actual musical instrument, useable in practice, in real world musical contexts. This context raises new technical and conceptual issues.

Musical issues related to real time

We have stressed above the importance of producing musical material *quickly*, so that the continuation of the system is undistinguishable from the phrase played by the musician. In practice, this raises the issue of detecting the end of a musical phrase as quickly as possible. In the current version of the system, we use a variable time threshold. Each time a note is played, we update the current average inter onset time of the input phrase. When no note is detected after 150% of this average time, we consider that the phrase is finished, and the system produces a continuation. This strategy works fine for most of the cases. More refined strategies can be envisaged, such as considering the relative importance of the last note (often significantly longer than the preceding notes). Additionally, we also estimate the tempo and beat phase of the input sequence, and produce a continuation which is synchronized with this beat, to respect the underlying beat, if any.

Musical issues related to context information

An important aspect of Markov models is that they generate locally consistent sequences, but are intrinsically unable to cope with external, unexpected parameters. In improvised music, however, the context changes all the time. For instance, in Jazz, the current harmony may be arbitrarily changed by one of the musician (e.g. the pianist), and other musicians have to adapt quickly to this change, for instance, by playing harmonically close notes in their solos. In our context it is unrealistic to provide these harmonic changes in advance to the system. We have therefore extended Markov generation to adapt to arbitrary, unexpected changes. The main idea is to substitute the usual probabilistic method for choosing continuations by a function that takes into account the context. The harmonic context can be defined as the notes played in the last 2 seconds by some musician (*e.g.* the pianist). These notes are provided as additional input to the system. We then introduce a harmonic function which computes the harmonic distance between a possible continuation and the harmonic context. The choice of a continuation is then determined by a compromise between the Markovian probability of a continuation, and this harmonic function. More precisely, the probability for choosing *X* as a continuation, given an input sequence *Seq* and a harmonic context *Ct* is now:

$$\text{Prob}(x) = S * \text{Markovian\_Probability}(X, Seq) + (1 - S) * \text{HarmonicDistance}(X, Ct),$$

where S is a parameter that can be adjusted by the user beforehand. When S = 1, the system behaves as a musical automata insensible to external context. When S = 0, the system is very reactive to the context, and probably looses any stylistic integrity. Interesting values are of course in the middle way, when the system retains some stylistic consistency, but still tries to follow the context. This is when an “interesting” dialog can take place.

Another musical issue is the management of rhythm. Experiments with Markov chains have mostly been performed with pitch information. Indeed, it has been long shown that musical pitches are somewhat easily recombinable. It is not the same with rhythm. Rhythmic information is conveyed both from local and global temporal relationships. The consequence is that rhythm information cannot easily be sliced up and recombined. We have introduced several modes for managing rhythm in the Continuator. More precisely, we separate rhythm from tempo, each of which may have different modes.

Rhythm:

- *Original rhythm*. The rhythm produced by the system is the rhythm as it was learnt. In this mode, each note carries with it its “local” rhythm, as defined by the note duration, the rest between the end of the note and the start time of its successor (possibly negative). As suggested above, this mode works well for styles having only locally consistent rhythms structures.
- *Input rhythm*. In this mode, the rhythm of the output is the rhythm of the input, possibly mapped is the output is longer than the input. This produces an effect of imitation.
- *Linear*. In this mode, the notes produced are streams of eighth notes. This mode is interesting for producing fast streams of notes, typically during be-bop style solos. - Metrical. This mode imposes a tempo, and the system learns the metrical structure along with the input sequences. This is particularly useful for “rhythmic” music in which the beat has a prominent importance.

The tempo can also be parameterised as follows:

- *Original tempo*. The tempo as it was played in the learnt corpus
- *Input tempo*. The tempo is the tempo of the input phrase. This mode is important when the musician wants the system to produce seamless continuations.
- *Fixed*. The tempo can be fixed to a given value, in particular in the Metrical Rhythm mode.

New playing modes

The system presented here can be used in many configurations, some of them actually unexpected. We list here the ones we have started to investigate and found most interesting.

Autarcy

In this mode, one musician is connected to the main input of the system. The system starts from scratch (no initial memory), and learns progressively the musical style of the musician. The system is set for instance to produce continuations of increasing size. At the beginning, only a few notes are generated, as continuations or answers to input phrases. The more the system learns, the more accurate the style representation is and the more notes the system plays. An interesting musical effect is that the musician gradually shifts from a “teaching”, active behavior, to a listening, passive mode: the system initially continues the phrases played by the musician, but eventually it is the musician who continues the phrases played by the system.

Virtual Duo

This mode is the same than Autarcy, except that the musician initially loads a pre-recorded memory. This memory is taken from a library of memories, built by simply letting various musicians play freely with the system. This allows a musician to play, virtually, with any body whose memory has been recorded previously.

Contextual Continuation

In this mode, one musician is connected to the main input of the system. Another musician (e.g. a pianist) is connected to the contextual input. The continuations produced by the system are built from the input phrases of the main user, but follow dynamically the contextual information provided by the second musician. This mode enhances the interaction between the two musicians, as the second input has clear, and parameterized, effect on the overall generated music.

Playing twice with oneself

This provocatively labeled mode consists in two phases. First a musician plays harmonically rich music to the system: chords, chord sequences of all kinds. In a second phase, the system produces an infinite stream from the learnt chord sequences. The musician then plays on top of this harmonic material a solo improvisation, which is fed in the harmonic context input of the system. The chord sequence played by the system tries then to “follow” the improvisation. This mode creates a striking impression to the musician (and the audience), as what happens in effect is that the musician virtually follows himself.

Swapping mode

In this mode, several musicians are connected, each one to a different version of the system. The respective memories are swapped. For instance, the guitarist will use the memory of the pianist. The result is that the continuations of the guitarist will use the patterns and style of pianist, and vice versa. This is a new collaborative music-playing mode with lots of potential, and which is still to be experimented further.

Experiments with these modes have been performed both in the laboratory and during various live concerts. Musician and composer Gÿorgy Kurtag Jr. has played extensively with the system and performed in *festival d’Uzeste* (August 2001) and festival *Sons d’hiver* (January 2002). Jazz pianist Bernard Lubat has played with the system and more collaboration is envisaged. An interactive musical composition is scheduled for the Vienna 2002 *Festwochen* as well as for the 2002 *Budapest festival*. Other experiments with Jazz musicians are being conducted with the goal of enriching our library of memories and collecting feedbacks on practical issues in using the system.

Additionally the system has applications for musical training. Indeed, preliminary experiments show that children are very sensitive to the imitative power of the system. Even when children are not musically trained, they developed instinctively personal playing modes. These modes can sound as “primitive”, but careful studies show that they are in fact rather differentiated. For instance, a child can hammer repeatedly one single note with one finger, or play chords with all his fingers, stick to the central region of the keyboard, or explore various regions, with notes, arpeggios, chords, etc. The use of the system in an imitative mode (autarcy) is believed to push the child to explore new playing modes. Current experiments are being performed in schools to test this hypothesis, and others, in the context of musical education.

Conclusion

We have described the technical issues at stake for designing an interactive system with a style learning facility. We illustrated the main new playing modes made possible by such a system. Examples can be found at our web site: http://www.cs.sony.fr/~pachet/Continuator. Current works focus on experimentations with musicians of various styles (contemporary, jazz, baroque) and children. Technically, a full audio version is being implemented, which should allow singers and more instruments to be taken into account.

Address for correspondence :
FRANÇOIS PACHET
Sony-CSL
6 rue Amyot
75005 Paris
France

E-mail: pachet@csl.sony.fr

References

Brooks, Hopkins, Neumann & Wright. "An experiment in musical composition." *IRE Transactions on Electronic Computers*, 6(1), 1957.

Cope, David. Experiments in Musical Intelligence. Madison, WI: A-R Editions, 1996.

Pachet, F. "Interacting with Style: the Continuator", submitted to *International Computer Music Conference*, 2002.

C. E. Shannon, ``A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379-423 and 623-656, July and October, 1948.

J. L. Triviño-Rodriguez; R. Morales-Bueno, Using Multiattribute Prediction Suffix Graphs to Predict and Generate Music, Computer Music Journal 25 (3) pp. 62-79, 2001.