Contents

1 Bebop Virtuosity Explained 1
François Pachet
1.1 Virtuosos as Exceptional Humans 1
1.1.1 Virtuosity in Art 1
1.1.2 The Cognitive Science Perspective on Virtuosity 3
1.1.3 Virtuosity as an Attraction Device 4
1.1.4 Virtuosos as Creators 4
1.2 The Case of Jazz 5
1.2.1 The Rules of the Game 7
1.2.2 Bebop Phrases 7
1.2.3 The Melodic/Harmonic Interplay 7
1.2.4 Playing Outside and Side-Slipping 9
1.2.5 Virtuosity is to Improvisation what Running is to Walking 11
1.2.6 Claims 12
1.3 Modelling Jazz Improvisation Generation 12
1.3.1 Non-Markovian Approaches 12
1.3.2 Markov Chain Approaches 14
1.4 A Note-Based Jazz Generator 15
1.4.1 Pitches for Representation, Beats for Generation 16
1.4.2 Handling Harmony 16
1.4.3 Chord Change Negotiation 18
1.4.4 An Example Training Set 20
1.5 Escaping Markovian Boredom 23
1.5.1 Side-Slips and Formal Transforms 23
1.5.2 The Control Issue 25
1.5.3 Reusing Intentional Scores 29
1.6 Virtuoso: A Virtuoso Enabling Interactive System 29
1.7 Discussion 30
References 31

Chapter 1
Bebop Virtuosity Explained

François Pachet

Abstract Virtuosos are human beings who exhibit exceptional performance in their field of activity. In particular, virtuosos are interesting for creativity studies because they are exceptional *problem-solvers*. However, virtuosity is an under-studied field of human behaviour. Little is known about the processes involved to become a virtuoso, and in how they distinguish themselves from normal performers. Virtuosos exist in virtually all domains of human activities, and we focus in this chapter on the specific case of virtuosity in jazz improvisation. We first introduce some facts about virtuosos coming from physiology, and then focus on the case of jazz. Automatic generation of improvisation has long been a subject of study for computer science, and many techniques have been proposed to generate music improvisation in various genres. The jazz style in particular abounds with programs that create improvisations of a reasonable level. However, no approach so far exhibits virtuoso-level performance. We describe an architecture for the generation of virtuoso bebop phrases which integrates novel music generation mechanisms in a principled way. We argue that modelling such outstanding phenomena can contribute substantially to the understanding of creativity in humans and machines.

1.1 Virtuosos as Exceptional Humans

1.1.1 Virtuosity in Art

There is no precise definition of virtuosity, but only a commonly accepted view that virtuosos are human beings that excel in their practice to the point of exhibiting exceptional performance. Virtuosity exists in virtually all forms of human activity. In painting, several artists use virtuosity as a means to attract the attention of their audience.

Sony CSL-Paris, 6, rue Amyot 75005 Paris, France, e-mail: `pachet@csl.sony.fr`

Felice Varini paints on urban spaces in such a way that there is a unique viewpoint from which a spectator sees the painting as a *perfect geometrical figure*. The effect is similar to looking at a city photograph on which the figure would have been added with a digital picture editor. Moving away from this precise viewpoint slightly distorts the figure; moving further away breaks it into fragmented shapes, thus breaking the illusion, which reveals the unsuspected virtuosity of these apparently simple creations.

Similarly, artist Liu Bolin paints himself so as to become almost invisible, when he stands exactly at specific locations (near a balustrade, in a cinema with red chairs, etc.). In both cases, what is at stake, from our viewpoint, is the production of simple objects (geometrical figures in the case of Varini, mundane backgrounds in the case of Liu Bolin), together with evidence of the *difficulty* inherent to their realisation.

Another example in the visual domain is the *Ryoanji* stone garden in Kyoto. This garden is well-known for the calm and serene atmosphere it creates and many studies have attempted to uncover the reasons of its attraction (see e.g., Tonder et al. (2002)). However, one reason stands out: wherever the watcher sits, only 14 out of the 15 stones are visible at a time (Figure 1.1). Such a property turns an apparently random configuration of stones into a fascinating, singular creation. We argue that a reason for this fascination may also be that the object to see is again both simple and understandably difficult to create.

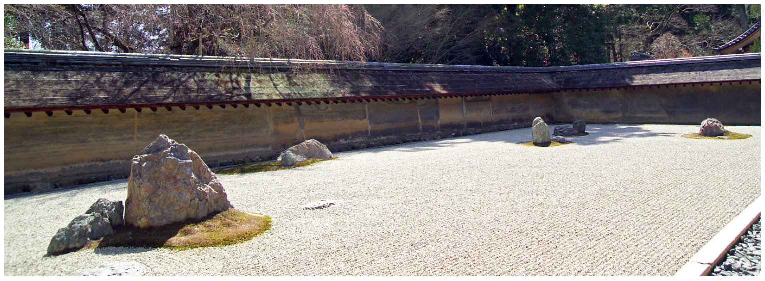

Fig. 1.1 The Ryoanji stone garden in Kyoto. It is said that all stones are visible except one, whatever the sitting position.

Virtuosity exists, or rather, occurs, also in time-related performance. People trained in performing fast mental computation compute operations several orders of magnitude faster than normal humans. Alexis Lemaire, world champion of the extraction of 13^{th} root of very large integers (200 digits), exhibits spectacular performance in all sorts of mental calculations. He calls this activity *hypercalculia* (Lemaire and Rousseaux; 2009). What he produces is simple, but almost no one else can do it.

Virtuosity (from the Italian word *virtuoso*) is an essential dimension of music performance. In the Western culture, virtuosity in performance is a controversial notion and is the subject of many debates. On one hand, virtuosity is considered the greatest possible achievement of the art of solo instrumental performance (Valéry;

1948; Penesco; 1997). On the other hand, virtuosity is often considered in opposition to expressivity (see e.g., O'Dea (2000)). But virtuosos are above all outstanding classical musicians (violinists in particular) who perform musical pieces known to be extremely difficult, *at the limit of human capacities*.

In the field of poetry, virtuosity manifests itself under the form of "satisfying difficult constraints." It was shown for instance that the adaptation of Latin rhetoric to old English poetry created complex constraints for the authors. Satisfying these constraints was the source of great inventiveness and creation (Steen; 2008). The association Oulipo (OuLiPo; 1988) pushed very far the idea that constraints, in particular difficult ones, could be the source of inventiveness in literature and poetry. Novels by Georges Perec such as "The void" (a novel without the vowel "e"), or its counterpart "Les Revenentes" (a novel with "e" as the only vowel) are spectacular achievements of this movement.

1.1.2 The Cognitive Science Perspective on Virtuosity

Despite these achievements, virtuosity has hardly been addressed by cognitive science. From the viewpoint of physiology, there are known limits to the motor systems and the sensory-perceptive abilities of humans that are relevant to the study of virtuosity (London; 2004, 2010). For instance, Fitt's law (Fitt; 1954) states that the time it takes to reach an object is a function of the distance to, and the size of, the target object(s). Consequently, tradeoffs have to be found between speed and accuracy, both ingredients being required for achieving virtuosity, e.g., in music. Another important law governing human interaction abilities is the Hick's law (Hick; 1952), which states that the time it takes to take a decision is a function of the number of possible answers:

$T = b \times log_{2(n+1)}$which generalizes to: $T = b \times H$, where H is the entropy of the system.

These two rules combined yield the interesting argument that virtuosity is somehow only possible at the cost of not thinking. As Justin London (2010) sharply phrases it: "Virtuosos can suppress the executive/monitoring functions of their brains when they perform; and thereby avoid the speed traps of their pre-frontal cortices".

The way to achieve this is by intense training. The *10,000 hour rule* (see e.g., Ericsson et al. (1993); Sloboda et al. (1996); Gladwell (2008)) states that about 10,000 hours of training are required to become a world expert in any domain. Most biographies of well-known musicians confirm the fact that music virtuosos (in classical music, jazz, and even pop) have spent most of their youth training (Mozart, Charlie Parker, John Coltrane, Bireli Lagrene, the Beatles).

1.1.3 Virtuosity as an Attraction Device

Bird songs are particularly interesting for virtuosity studies as they are a rare case in which the whole production and reception process has been studied in-depth, yielding breakthroughs and fascinating findings.

Researchers in animal behaviour have long been interested in the phenomenon of bird song production and its role in the mating process. In several bird species, male birds produce songs primarily to attract females. The issue of what makes a bird song more attractive than others has received particular attention in recent years. Various results have shown that specific features of songs can account for their popularity. For instance, great reed warbler females (*Acrocephalus arundinaceus*) exhibit a preference for long songs over short ones in the wild (Bensch andHasselquist; 1991).

More interestingly, the study by Draganoiu et al. (2002) focused on the case of the domesticated canary (*Serinus canaria*). Male canary songs have a specific phrase structure. Two features of these phrases were shown to significantly increase liking: frequency bandwidth and trill rate. However, it was also shown that these two features are somehow contradictory: similarly to Fitt's law, a tradeoff is observed in real phrases, due to the specific motor constraints of the birds vocal track.

The breakthrough experiment of Draganoiu et al. (2002) consisted of synthesising artificial phrases optimising these two features in an unrealistic way that is "beyond the limits of vocal production". The exposition of these artificial phrases to females birds showed unequivocally that females preferred these phrases to the natural ones (see Figure 1.2). An interesting interpretation for this preference is that the production of "difficult" phrases maximising both bandwidth and syllable rate may be a reliable indicator of male physical or behavioural qualities.

This evolutionary argument emphasises the role of virtuosity in music appreciation. In popular music, virtuosity is explicitly present in specific genres (e.g. so-called *shredding* in hard-rock, illustrated by guitarists such as Yngwie Malmsteen, or melodic-harmonic virtuosity in bebop), as we show below.

1.1.4 Virtuosos as Creators

In this chapter, we adopt a specific perspective on virtuosity. From the viewpoint of complexity and computer science, we envisage virtuosos as exceptional *problem solvers*. Virtuosity can be objectively measured, observed, and as such is, "as a concept, closer to the ground, than creativity" (Howard; 2008).

Indeed, the capacity to effortlessly navigate in large search space in real-time is not only a matter of physiological prowess. By transferring part of the decision processes to the body, a virtuoso naturally compiles his knowledge in a remarkable way that can teach us a lot about innovative problem-solving.

For instance, virtuosos in mental calculation invent and make extensive use of so-called mathematical *tricks*. As an example, squaring any number ending in 5 can

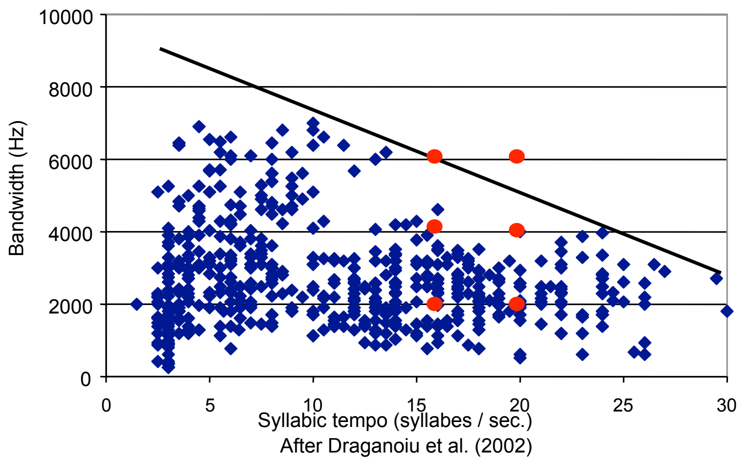

Fig. 1.2 The distribution of canary phrases, in a bandwidth/tempo space, representing the natural tradeoff between bandwidth and syllabic tempo. Red circles represent the phrases used for the experiment. The artificial top right phrases optimising the two features in unrealistic ways were the most successful.

be done easily using a simple formula (take the first digits except the last 5, multiply it by itself plus 1, and then concatenate 25 at the end). Some of these tricks are well-known, but many others are not, and probably ignored by their inventors: intense training may result in exceptional performance, not necessarily in clear explanations. In the following sections, we show how jazz virtuosos produce interesting inventions, and how modelling this virtuosity leads to interesting insights about the nature of invention and creativity.

1.2 The Case of Jazz

Much like language, the ability of humans to spontaneously improvise music in real time is considered by many as an extraordinary skill, a sort of magic. Most of this magic, again, comes from hard training. As Levine (1995) states in his introduction: “*A great jazz solo consists of 1% magic and 99% stuff that is explainable, analysable, categorisable and doable. This book is mostly about the 99% stuff.*“ This chapter is about putting the *99%* stuff in a machine, and making the remaining *1%* explicit. In particular, our aim is to separate clearly what can be reasonably automated – what virtuosos are able to do unconsciously – from what emanates from artistic, conscious decision-making.

Invented in the 1940s with Charlie Parker and Dizzie Gillespie, bebop is an idiom of jazz where a strong emphasis is put on melodic and harmonic dimensions. Virtually all instruments of the Classical orchestra have been used by bebop musicians. Nowadays, bebop musicians continue expanding the style. The case of jazz bebop improvisation is particularly interesting because the specific constraints of bebop are shared unambiguously and can be easily expressed using well-defined languages: In some sense, jazz improvisation is a special form of computing.

Scientists have long tried to debunk the magic of jazz improvisation, starting with the psychologist Philip Johnson-Laird. His work is not to be judged by the musical quality of his algorithmic productions, but by the seminal nature of his arguments. One of his main claims is that the ability to produce an improvisation does not require any "short-term memory" (Johnson-Laird; 1991, 2002). He demonstrated this idea by proposing memoryless automata that automatically generate rhythmic and melodic material. Since then, more powerful algorithmic techniques have been used to produce jazz improvisation (see §1.3), but it can be said that the problem of modelling "basic" bebop improvisation has been solved, notably by exhibiting improvisation generators satisfying the basic rules of the game (detailed in §1.2.3).

However, the improvisation problem has been only partially solved. Trained jazz musicians listening to the examples produced by these previous works rarely experience the feeling of hearing a machine outperforming humans.

In fact, professional bebop musicians are like Olympic sportsmen or chess champions, reaching a level of technicality which is far beyond the capacities of a beginner. They are usually sought after not so much because they exhibit a general "ability to improvise" – children can also improvise – but for their specific display of virtuosity. Contemporary jazz improvisers such as John McLaughlin, Al Di Meola, Bireli Lagrene (guitar), or Stefano di Battista (saxophone) exhibit a level of virtuosity that seems to reach beyond the limits of what most humans can do (the expression "not human" appears indeed often in commentaries about these performances on social Web sites). They play intricate phrases at such a speed that even the transcription of their solos from recording is a challenging task. Deciding which notes to play at that speed seems indeed impossible, so the virtuosity question can be rephrased as: How can one perform and execute these musical choices so accurately and so fast?

Of course, performance as well as timbral dimensions are undoubtedly important in music, and can themselves be the subject of virtuosity display (Bresin; 2000), but these are outside the scope of our study: Following the argument that "bebop is more about content than sounds" (Baker; 2000), we focus here on the melody generation task. We consider that virtuosity is not only an appealing facet of bebop, but one of its *essential features*. This situation bears some intriguing analogy with bird singing behaviour. Though bebop virtuosity is not only about speed as we will see below, this analogy suggests a primary role of speed in the attraction for specific melodic movements.

1.2.1 The Rules of the Game

In this section we define precisely the musical corpus we target: linear improvisation, which corresponds, roughly speaking, to virtuoso passages of bebop improvisations.

1.2.2 Bebop Phrases

Virtuoso phrases are played fast, typically $1/16^{th}$ notes at 120 *bpm* or more, which represent at least 8 notes per second. This speed implies a number of characteristics for these passages that we call “linear”. The term linear has been used in the jazz theory literature (e.g. Ricker (1997)) to describe phrases built from scales rather than from chords (i.e. arpeggios), thereby creating a sensation of melodic or horizontal consistency. More precisely we define linear improvisations as phrases which are (1) played fast (eighth-notes or faster), (2) monophonic, (3) without silences, and (4) rhythmically *regular*.

All these criteria are implied by speed: monophony because it is usually impossible to play fast a polyphonic instrument. Regular rhythm means that each beat in a measure is played with notes of the same durations (for the sake of simplicity, 1/4 notes, 1/8 notes, 1/16 notes, or triplets thereof, see Figure 1.3). Rhythmic regularity is also implied by speed as it is very difficult to change the rhythm when playing fast. Linear improvisation is pure melodic invention, other musical dimensions are secondary.

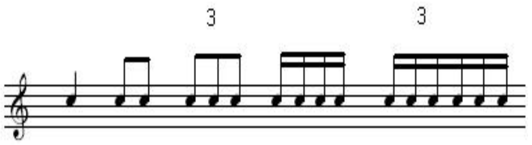

Fig. 1.3 Examples of various rhythms used during linear improvisation.

All virtuoso bebop musicians include linear passages in their choruses. Virtuoso improvisations are of course rarely entirely linear, but they are often at least locally so. As we hypothesise, linear passages correspond to a specific, intentional mode of musical production deliberately chosen (at some risk) by the musician.

1.2.3 The Melodic/Harmonic Interplay

Bebop improvisation is a particular form of tonal music in which harmony plays a central role. Given a chord sequence, usually taken from a shared repository such as the *Real Book* (Real; 1981), the game consists of producing a stream of notes that satisify two criteria simultaneously: *harmonic consistency* and *continuity*. This game, commonly referred to as "playing or negotiating the changes", can be considered the main technical challenge of bebop improvisation (it is also possibly a key ability in other domains, such as management, see Holbrook (2009)). Paradoxically, an extra recipe for producing interesting melodies consists of breaking the first rule through various *escape* mechanisms, as we will see below.

To the ears of a trained musician, virtuoso choruses rarely contain any mistakes with regards to these principles. This is a striking property of virtuosos in general, and jazz improvisers in particular: they produce *perfect melodies*, sounding as if they were proof-read before being delivered. We will now review these principles.

1.2.3.1 Harmonic Consistency

The generated melody must comply with the current chord in the sequence. Strictly speaking, this means that the melody should consist mostly of notes which belong to a scale appropriate for the chord. The identification of the correct scale requires, in principle, an analysis performed on the whole chord sequence. Chord sequence analysis was shown to be a non-trivial task in general (Steedman; 1984). In practice, however, simpler forms of analysis are used, which consist in using ready-made associations between chords and scales. Such associations are routinely available in harmonic treatises, e.g., in Aebersold (2000). For instance, on a *minor 7(5b)* chord, say, *D minor 7 (5b)*, one can use a *harmonic minor* scale one minor third above the root (here, *F harmonic minor*).

1.2.3.2 Continuity

Jazz beginners often improvise by playing arpeggios corresponding to each chord: this simple technique satisfies local harmonic satisfaction by definition, but produces obviously uninteresting, unmelodic phrases. Producing a "sense of melody" is difficult to define precisely. Continuity is a good approximation and is easier to define. We will see that low-order Markov processes exploiting carefully chosen scales guarantee a form of natural continuity.

Melodic continuity is a difficult challenge for a human when playing fast, as it requires the ability to find quickly short paths between the note currently being played and the next ones, which may be in a different scale. This ability is referred to as chord change negotiation, stressing its inherent problem-solving dimension.

Note that continuity does not necessarily imply *brownness*, in the sense of (Voss and Clarke; 1978), i.e. the sole use of small intervals. It rather implies that notes are glued together smoothly, and not made up of isolated elements or patterns, concatenated without care. For instance, the phrase in Figure 1.7 contains several large intervals but is perfectly continuous.

The One-Step-Max theorem

There is a factor that helps address the continuity challenge: the *one-step-max theorem*. The scales used in jazz (minor, major or diminished, in first approximation) contain an interval of maximum 3 semitones (in the harmonic minor scale). Consequently, any note is always within 1 semitone maximum (up or down) to a note of any possible scale, i.e. a “good” note. We will see below how this theorem can be used as a rescue mechanism when the basic generator fails to find a solution.

1.2.4 Playing Outside and Side-Slipping

The bebop language is deeply grounded in tonal harmony. However, like all languages, bebop evolves. One important development was caused by a paradoxical force that pushes musicians to escape the constraints of harmonic satisfaction, once they know how to satisfy them perfectly: *playing out* in jazz jargon. Playing out is not to be confused with *free jazz*, a radical way to escape the laws of tonal harmony, in which there are no more rules whatsoever. Playing out, in bebop, is a precisely defined musical device whose mastery necessitates perfect control of the instrument. The ability to play out “the right way” can be considered a sign of a complete mastery of the art.

The main way to play out is called *side-slipping* (or *side-stepping*). Shim (2007) dates the origin of side-slipping back to Art Tatum, an acknowledged piano virtuoso, who “displayed his mastery of chromatic harmony by effortlessly floating in and out of keys” (p. 183). A stepping stone of this evolution is probably the incursion of modal improvisation in the jazz repertoire; with the famous tune “So What” by Miles Davis, based on a long repetition of a D minor chord. To avoid a “tide of boredom” due to this harmonic monotony, various techniques for escaping tonality were invented, including side-slipping (Coker; 1984, p. 49).

Pedagogical definitions of side-slipping may be found in jazz theory books (Coker; 1984, 1997; Levine; 1995), with some variations. Side-slipping is a device that produces a short sensation of surprise, in a context deemed too predictable (Levine; 1995). The idea is to play out-of-key, with the goal of momentarily creating tension, and then come back to the right key, which can be different from the starting key. Most often, the out-of-key segment uses symmetry. For instance, it can be the same phrase transposed a semi-tone higher. The listening impression is described by Coker (1984) is as follows: “Like the stretching of a rubber band, the attuned listener seems to know that the player’s excursion into another key is very temporary and that he will snap back to the original key when the tension period is over. In the meantime, the listener has been taken on a brief trip that has broken the monotony of modality.” Side-slipping was intensively used by pianists like Lennie Tristano (Shim; 2007, p. 183), and many others (John Coltrane, Alan Holdsworth) and is now a classical ingredient of modern improvisation.

Figure 1.4 shows a typical side-slip, given by Coker (1997, p. 50). The mechanical dimension of the side-slip appears clearly: here a simple transposition of a 4-note pattern one semitone up, and then down. Figure 1.6 shows a more complex example of a side-slip produced backward in time, i.e. played before the non-transposed version, creating an even more surprising effect (shocking, then deliciously soothing). Note that such an effect would not work if played at low speed, as the time during which wrong notes are played would be too long, creating a risk for the listener to lose the sensation of tonality. As such side-slipping is not an ornamental device, but a central feature of linear improvisation.

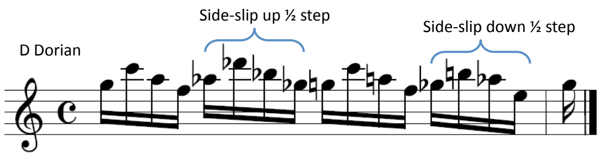

Fig. 1.4 Example of a side-slip, given by Coker (1997, p. 50). Note how the first side-slip smoothly continues in the "right key" (here, D minor).

There are many variants of side-slipping, notably concerning the device used to produce the phrase *out of key*, its length, metric structure, etc. (Coker; 1997). For instance, Figure 1.5 shows a diatonic side-slip invented by John Coltrane (and used extensively e.g., on his improvisations on *Giant Steps*). This is a nice "trick", or rather a small theorem of tonal music: when a motive in some major key (say, F) is transposed up 1 semitone, it is most of time in the initial key transposed 1 minor third up (here, Ab7).

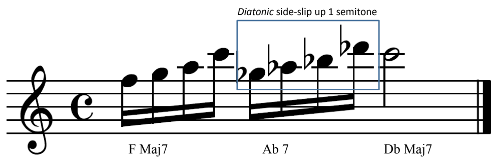

Fig. 1.5 A Diatonic side-slip invented by Coltrane. This particular side-slip is such that it actually does not create any harmonic tension, as the transposed motif (up 1 semitone) stays, miraculously, in the tonality (here, one minor third above).

The difficulty for improvisers is not only to produce the slide-slip, but to re-establish continuity during the *re-entrance* phase. This necessitates tricky planning, as the final notes of the transposed pattern are, usually, precisely out of key, so no natural continuation may be in the musician's hands.

We will see how our framework copes with side-slipping in a general way, by allowing side-slips to be inserted smoothly in the generated chorus while keeping continuity.

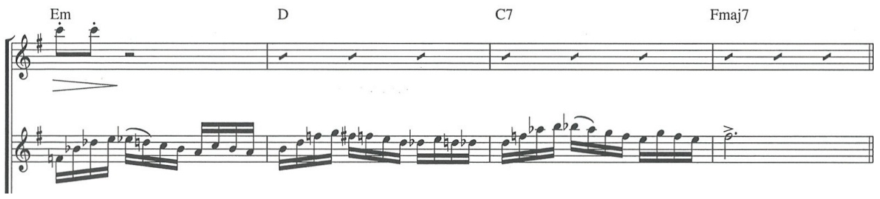

Fig. 1.6 A tricky example of a "reverse side-slip" by Al Di Meola in a chorus on Guardian Angel (GuitarTrio; 1977). The two first phrases are transpositions, (2 then 1) semitones lower, of the last one, which is in the right key, thereby creating a stunning resolution effect, only made possible by speed (here, 184 bpm).

1.2.5 *Virtuosity is to Improvisation what Running is to Walking*

Virtuosity is about speed, but not only speed. Beyond speed – innate for computers – virtuosity is the capacity to play *interesting* phrases fast, and make them appear as singular solutions to a difficult problem, much like a magician tirelessly extracts rabbits from a shallow hat. Like running is not walking faster (Cappellini et al.; 2006), playing virtuoso phrases calls up cognitive skills and motor mechanisms that differ from the ones used in standard improvisation, which consists basically of paraphrasing the original melody (Baggi; 2001). In this view, virtuosity is a specific improvisation *mode*, in which the musician deliberately chooses to enter and exit, during his solo. Often, virtuoso passages constitute the climaxes of the chorus. This is obvious in concert recordings such as the GuitarTrio (1977) in which virtuoso passages (see Figure 1.7) are followed by enthusiastic rounds of applauses.

An important apparent characteristic of virtuosity in bebop is that the musicians give an impression of precisely *controlling* their production, using some sort of high-level inner commands. Such an impression is obvious when listening to the effortless character of Art Tatum's improvisations, which allow him to flow in and out of harmonies with a total control on their high level structure. Indeed, the improviser's ultimate fantasy is probably not to *produce* but to *control* such a virtuoso flux (Sudnow; 1978), through high-level mental commands. In short, to be the director of one's inner orchestra. How is this possible?

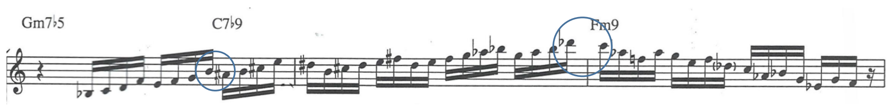

Fig. 1.7 A virtuoso passage (152 bpm) in a chorus by John McLaughlin on Frevo Rasgado (GuitarTrio; 1977). Note the "smooth" chord transitions.

1.2.6 Claims

In this chapter, we make a number of claims. The main one is that we present a system that generates virtuoso phrases of the same musical quality as the ones human virtuosos produce. The validity of this claim is left to the appreciation of a trained jazz listener, who can judge from the outputs (scores and videos) of our system, *Virtuoso*, available on the accompanying web site.

The second claim is that we propose an operational answer to the virtuosity question (how do they do that?), by introducing the notion of *intentional score*: the temporal series of high-level musical decisions taken by the virtuoso to generate a chorus. These decisions are arbitrary, and may be seen as the "*1% magic*" mentioned by Levine in his introduction (see §1.2). This intentional score is the backbone for automatically producing virtuoso phrases, and our system may be seen as an *interpreter* of this score, which generates a chorus that satisfies it, i.e. Levine's "*99% stuff*". We show through our various examples that this score suffices to generate virtuoso phrases of high quality. All the decisions in the intentional score are done at the *beat level* (and not at the note level), i.e. at a *low frequency*, thereby substantially reducing the cognitive load of rapid note-level decision making. This explains how the by-pass of high-level cognitive decision-making may be operated in practice (see §1.1.2).

Most importantly, we show how human jazz improvisers have contributed, at least in two respects to *inventing the bebop style* (and its extensions) *thanks to virtuosity*. The two features we focus on are only possible thanks to extreme virtuosity: (1) side-slips and (2) fine-grained control. We describe and interpret these two major contributions to style invention in the context of Markov-based music modeling.

After a review of the state-of-the art in jazz modelling, we describe a Markov-based model of jazz improvisation and show that it is well adapted to generate melodies that fit with arbitrary chord progressions. We then use this basic model to *state and solve* the two main issues of jazz generation: control and side-slips.

1.3 Modelling Jazz Improvisation Generation

Many studies have addressed music composition and improvisation, so we focus on those specifically addressing jazz. As is often the case in computer science, these studies follow the general algorithmic trends of the moment. We handle separately the case of Markov modelling as this is the core of our proposal.

1.3.1 Non-Markovian Approaches

Ulrich (1977) proposed an all-encompassing system that performs chord sequence analysis and chorus generation using a purely algorithmic approach, reaching a reasonable level of musicality. Walker (1997) and Thom (2000) built interesting systems emphasising the *dialog dimension* of improvisation rather than the musical quality. A more ambitious case-based reasoning approach was proposed by Ramalho and Ganascia (1994), emphasising the role of motivic components, and following the "knowledge level" paradigm. This approach proposes to explicitly reconstruct a cognitively plausible model of a working jazz memory, and was applied to the automatic generation of bass lines, yielding some interesting outputs, favourably compared to Ron Carter's samples (Ramalho; 1997). It relied on a manually entered set of cases, limiting its scope in practice. Genetic algorithms have been used for music generation by a number of researchers (Weinberg et al. (2008); Bäckman and Dahlstedt (2008); Papadopoulos and Wiggins (1998)), yielding real time systems used in concert, and producing interesting improvisation dialogs, like in the *GenJam* system of Biles (1994). These systems, again, apply a general paradigm (here, evolutionary algorithms) to chorus generation in a top-down approach, without concern for harmonic satisfaction and continuity. Their outputs, although sometimes spectacular, are still below the level of professional musicians, and do not display particular virtuosity. Interestingly, the system described by Grachten (2001) was used as a basis for studying jazz *expressivity* in saxophone solos (Ramirez et al.; 2008). The studies described in Hodgson (2006), also use genetic algorithm but focus on detailed characteristics of the Charlie Parker style. Notably, Hodgson shows the importance of dyadic (two-note) patterns in the elaboration of Charlie Parker's melodic repertoire. But the use of random generation, intrinsic to evolutionary algorithms, here also, gives results of varying quality, necessitating manual editing (the author describes the results as "partially correct"). Note that manual editing echoes the approach of Harold Cohen with his Aaron panting program (McCorduck; 1991), as well as that of David Cope (1996), who use partial manual editing to finish their compositions. We will see below how we substitute manual intervention by *intentional controls*, and the implication on our models.

Probabilistic grammars were used by Keller and Morrison (2007) to generate jazz improvisation. The outputs of their system "compare favourably with those played by college-level jazz students of at least an intermediate playing level, if not better." The grammar rules are manually encoded, and based on an explicit representation of note harmonic status (chord tone, passing tones, etc.). Note-level information is required when teaching improvisation, but we do not think they are necessary for generating improvisation. As we will see, we adopt an approach in which this information is not represented explicitly. However, our generated phrases do contain a natural blend of, e.g. chord tones and passing notes. Note-level characteristics naturally emerge from the generator, rather than being prescribed by the system.

Franklin (2006) showed that recurring neural networks could learn entire songs given a melody and the associated chord sequence, and produce new improvisations on these chord sequences. This system demonstrates that non-symbolic approaches can capture some of the knowledge of jazz musicians, but the results shown are also college-level.

Side-slipping is briefly mentioned as a possible composition operation in the *ImproVizor* system (Keller et al.; 2005), but the process and more generally the devices for playing out-of-key "the right way" have not yet been the subject of modelling in improvisation generation studies. Finally, it can be noted that commercially available software like *Band-in-a-box* (PG Music), or the *Korg Karma*$^{\text{TM}}$ family of synthesisers produce reasonable improvisations in real time, over arbitrary harmonic constraints, using algorithmic approaches. These systems may produce musically interesting outputs, but their analysis is difficult because of a lack of published technical information.

1.3.2 Markov Chain Approaches

Other approaches to jazz improvisation based on Markov chains have been explored recently, showing notable success. These systems follow a long tradition in computer music modelling, dating back to the works of Shannon on information theory (Hiller and Isaacson; 1958; Brooks and Wright; 1957). Markov processes are based on the "Markov hypothesis" which states that the future state of a sequence depends only on the last state, i.e.:

$$p(s_i|s_1,\dots,s_{i-1}) = p(s_i|s_{i-1}) \tag{1.1}$$

Extensions to higher orders as well as variable-orders (Ron and Tishby; 1996) do not change substantially the basic principle of Markov generation. The Markov hypothesis, in all its forms, can be seen as a concrete implementation of Longuet-Higgins's *memoryless assumption* (see §1.2).

The Markovian aspects of musical sequences have long been acknowledged, see e.g. Brooks and Wright (1957). Many attempts to model musical style have therefore exploited Markov chains in various ways (Nierhaus; 2009), notably for sequence generation.

Many experiments in musical Markov models have shown that there is indeed a strong Markovian dimension in musical surface in most genres of tonal music, including jazz (see e.g., Nierhaus (2009) for a survey). The *Continuator system* (Pachet; 2003) was the first to propose a real-time improvisation generation system based on Markov chains, producing sequences as continuations of input sequences played by humans. This system was shown to deliver striking results, even passing "jazz Turing tests" (Van Veenendaal; 2004).

Most Markov generators are based on a *random walk* process, exploiting a probabilistic model of the input phrases. The generation is performed step-by-step, in our case, note by note, using a random draw scheme, which takes into account the context, i.e. the phrase generated so far:

```
Iteration at step i:
$next = Random\_Draw(context_i);$
$context_{i+1} := Concatenate(context_i, next);$
```

In practice, the context is limited to a certain maximal *order*. Random choice is performed as a weighted random draw, using an efficient representation of all encountered suffixes computed from the training set, which yields a probability table. The generated event is then concatenated to the context, and the process is iterated.

It has been shown that this model enables the creation of realistic outputs in many musical styles, with professional musicians (Pachet; 2003; Assayag and Dubnov; 2004) as well as children (Addessi and Pachet; 2005). Like previous approaches, these systems use a general, agnostic algorithm, uniformly applied to music sequences of any style. Consequently, the qualities of its outputs are also independent of the style of its inputs, and uniformly good . . . or bad. However, it should be noted that these systems perform best in *musically unconstrained* contexts, such as free-form improvisation. No convincing results were obtained when used in a *bebop* setting with the constraints we have introduced in the preceding section.

Random walk approaches have shown limitations when used for generating *complete pieces* as this strategy does not always favour the most probable sequences in the long term (Conklin; 2003). This is not an issue in our case, as we will see how the generation can be controlled using higher-level *controls* that determine global characteristics of the generated sequences, taking precedence over the details of the basic generation algorithm. Indefinite memory length is the main claimed advantage of the system proposed by Assayag and Dubnov (2004). In our context, this problem is irrelevant, as our goal is not to reproduce similar pieces, but to use the training samples to generate novel melodies in a highly constrained context. We consider a variable-length generation model, but, following Hodgson (2006), we restrict our maximum length to 2, an intentionally short value, which ensures an optimal compromise between similarity and creativity.

Another problem is related to the case where no solution is found (*NSF* hereafter). This happens when the context has not been encountered in the training phase. This problem, known as the *zero-frequency* problem has been addressed by many researchers in Markov modeling (see e.g. Chordia et al. (2010)), with no general solution. Here again, we favour an approach based on the observation of bebop practice, and propose a bebop-specific, simpler solution, described below.

1.4 A Note-Based Jazz Generator

The basic engine in our proposal is a variable-order Markov chain generator, with a maximum order of 2. This generator, described in the preceding section, is able to yield the "next" note, given the last 2 notes (at most) already played. Our experience has shown that augmenting the memory length does not improve the quality of the generation.

1.4.1 Pitches for Representation, Beats for Generation

All major decisions for generation are taken at the beat level, and constitute *in fine* the *intentional score*, which is a temporal sequence of beat-level decisions. These decisions are the following.

At each beat, a rhythm is chosen (arbitrarily in the first approximation) within the 5 possibilities described in Figure 1.3. This rhythm in turn determines the number of notes to produce for the beat (in our case, 1, 2, 3, 4 or 6). Consequently, there is no need to use durations as training data, as these durations are entirely determined by this rhythm choice. The velocities of each note are the velocities played in the training corpus (see below). No harmonic information is used in the training phase either, as the model used for generation is chosen, for each beat, according to the current chord, as described below. Higher-level attributes such as pitch contour, chromaticity, etc. are handled yet at another level as described in §1.5.2. Consequently, the representation used for the Markov model is based solely on pitch, reducing this basic mechanism to a simple one.

The justification for this choice is based on a long experience with Markov models for jazz, which convinced us that pitch is the only dimension of music that is well captured. Although other dimensions can technically be represented as such, it does not make much musical sense. There are two main reasons for this: firstly, only *intrinsic* attributes, by definition, are well adapted to Markov modelling. Pitch is an intrinsic attribute, but not rhythm, which emerges from the relation between adjacent notes or events. Second, there is no concrete evidence that modelling higher-level dimensions (harmony, pitch contour, etc.) yields interesting musical material, as these dimensions are correlated to each other in intricate and complex ways, raising the "viewpoint problem" that inevitably leads to *ad hoc* solutions and compromises. In some sense, the situation is comparable to the *multiple inheritance* problem in object-oriented languages (Stein; 1992): it works well when there is no conflict, but all the solutions proposed to solve the problem in the general case failed and were progressively abandoned.

1.4.2 Handling Harmony

There are several ways to consider harmony in a Markovian context. One of them is to consider harmony as a specific musical dimension, and use it as a viewpoint. This approach is followed for instance by Conklin and Witten (1995) or Cont et al. (2007). As discussed above, simultaneously handling several viewpoints creates viewpoint interaction problems that do not have general musically meaningful solutions. Furthermore, it introduces unnecessary level of complexity in generation. In our case, we can observe that chord changes in bebop never occur within a beat (they usually occur at the measure of half-measure level, sometimes at the beat, never within a beat). Hence our solution is simply to use chord-specific training databases, which are selected at each beat according to the underlying chord sequence.

More precisely, we use a simple set of chord/scale association rules. Such rules can easily be found in jazz theory text books, e.g. Aebersold (2000). For each chord type appearing in a chord sequence, we select the Markov model which corresponds to a particular “scale”. Using various substitution rules, it is easy to reduce the number of needed scales to a much smaller number than the number of chords. A drastic reduction is proposed by Martino (1994) who uses only minor scales throughout all chord sequences, using clever chord substitutions (e.g. *C 7^{th}* chord uses the *G minor* scale, *C altered* uses the *G# minor*, *C maj7* uses *A minor*, etc.). Although the Martino case is easy to implement (and is available in our repertoire of styles) we follow here a more traditional approach, and propose five scales: major, minor, diminished, seventh and whole tone (for augmented chords). As a consequence, we only need training data for these five scales, in a single key (C). The databases for the other keys are simply transposed from the ones in C.

```
selectHarmonicDatabase (chord)
        if chord is major, major 7, major 6 then return MajorDB;
        if chord is minor, minor 7, minor 6 then return MinorDB;
        if chord is half diminished then return HalfDimDB;
        if chord is 7 altered then return SeventhAlteredDB;
        if chord is augmented 5 then return WholeToneDB;
```

Fig. 1.8 The selection of a harmonic database according to the current chord.

Many variations can be introduced at this level, such as chord substitutions (see, e.g., McLaughlin (2004)). These can be typically performed at random, or according to any other parameter (e.g. user controls), and belong naturally to the intentional score. An important aspect of this method is that it is independent of all other parameters of the system, and notably does not necessitate an explicit memory of the past.

Here again, our solution is analogous to the way humans improvisers practice and improvise, as illustrated by the huge literature proposing training scales and patterns.

Changing Markov databases at each beat also creates a potential problem with regards to continuity: how to ensure that phrases evolve continuously during chord changes? It turns out that there is again a simple solution to chord change negotiation, which does not necessitate modifying the generation algorithm, but consists of *carefully choosing the training corpus*. In cognitive terms, this amounts to train to the point that all possible chord changes have at least one solution.

Let us consider several cases in turn, to illustrate the Markov process. We start by a training sequence in the key of *A harmonic minor* consisting of a scale played up and down (Figure 1.10). Using our generator, we can produce phrases in all minor

```
GenerateBeat(context, i) // context = the last generated output
        RP := chooseRhythmPattern;
        H := selectHarmonicDatabase (i chord);
        segment := new empty segment;
        Repeat N times (N = number of notes in RP)
                next(H) = Random_Draw (H, context);
                segment := Concatenate (segment, next) ;
                context := Concatenate (context, next) ;
        return segment with rhythm
```

Fig. 1.9 The basic GenerateBeat function integrates all musical decisions. N is the number of notes per beat, H is the harmonic context, which determines the Markov model to be used. H is constant during the beat.

keys like the one illustrated in Figure 1.11 (still in *A minor*). Other keys are handled simply by transposing the input sequence.

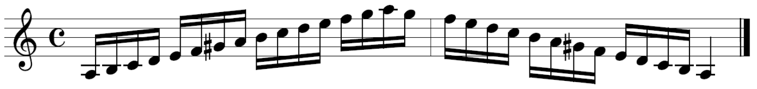

Fig. 1.10 A minor scale played up and down, used as the sole training phrase.

By definition, the generated phrases will all be Brownian, in the sense of Voss and Clarke (1978). This is caused by the fact that each pitch (except for the extremes) has only two possible continuations – one above and one below – in the diatonic scale used for training.

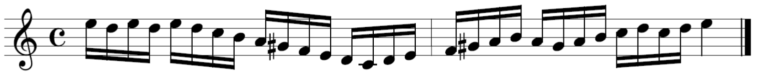

Fig. 1.11 A phrase generated by the Markov generator from the unique training phrase of Figure 1.10. Phrases generated by diatonic scales are all Brownian by definition.

1.4.3 Chord Change Negotiation

Let us consider now a chord sequence based on alternating between *A minor* and *A# minor*. We deliberately choose *A# minor* as this key is “far away” from *A minor*, and therefore harder to “negotiate”, because these two scales share only a few notes. Figure 1.12 shows an example of a phrase generated on this sequence. We notice that there are two NSF cases. They correspond to situations in which the last note of a phrase for a given chord does not exist in the training phrase for the next chord. Here, *C#* does not exist in the training base for *A minor* (first case), and *B* does not exist in the training base of *A# minor*, by definition of the harmonic minor scale.

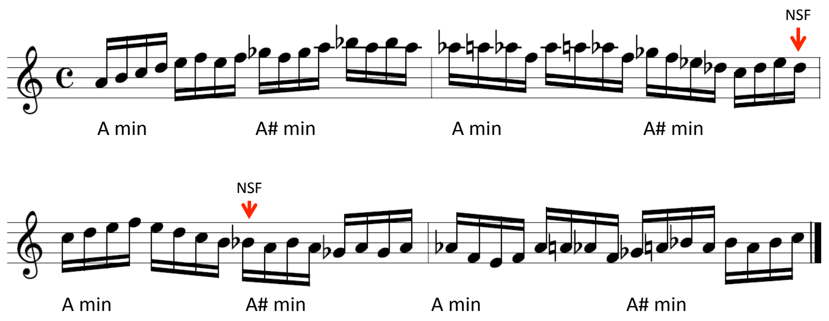

Fig. 1.12 A phrase generated on top of an alternating *A min / A# min* chord sequence, using the single ascending/descending A minor scale as training phrase. Note the two cases where no continuation is found to negotiate the chord changes (indicated by an arrow).

Contrarily to general approaches to the *zero-frequency problem*, we propose a musically justified solution with the two following arguments:

1. We reduce the number of NSF cases by carefully choosing the training corpus, as detailed in the next section. This step corresponds to *human training*,
2. In the remaining (rare) cases no solution is found, we use a simple heuristic based on the *one-step-max theorem* (see §1.2.3.2): since there is always a "good note" at a maximum pitch distance of one semitone, we try both and select the one that work, and we are guaranteed that there is always one.

This double solution turns out to work nicely. It can be seen in Figure 1.12 that in both NSF cases, the system chooses the right notes a semitone apart to fit with the harmony. The resulting phrase sounds smooth and continuous as if *nothing had happened*: it is virtually impossible to notice that the generated phrase is locally not Markovian. Furthermore, the system can easily produce a report after a series of improvisations, to suggest adding a training phrase containing the NSF cases encountered. In our case it could suggest the musician/system to practice/add phrases in *A minor* containing a *C#* or an *A#* (i.e. *B* in *A# minor*, once transposed in *A*): an interesting case indeed, which forces the use of chromaticisms or passing notes in a clever fashion. Figure 1.13 shows a more complete example on a succession of chromatically ascending minor chords.

It is interesting to note that this approach to the NSF problem corresponds to the pedagogical strategy proposed by Pat Martino (1994): learn only one scale (minor), but learn how to use it on any chord change to another minor chord. This implies practicing over 12 possible changes from one minor scale to another one, in all keys, so a total of "only" 132 cases from which any chord sequence can be smoothly negotiated (Pat Martino proposes a solution to substitute any chord by a minor chord, see §1.4.2).

Other strategies could be used, such as simply finishing the phrase. However, this kind of heuristic gets in the way of our modelling goal: the decision to stop or end a sentence is a "high-level" one that should not rely solely on such low-level technical considerations, but only on the musical intention of the musician.

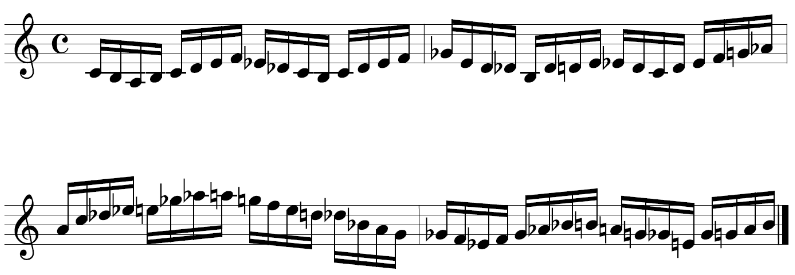

Fig. 1.13 A generation using only the harmonic scale as training base, on a succession of minor chords progressing one semitone up. NSF are indicated at chords #3, #4, and #7.

This mechanism produces phrases which satisfy local harmonic constraints, chord negotiation and continuity. However, the phrases wander up and down according to chance, and there is no direct means of controlling their structure. In some sense, this represents *technical virtuosity* (the ability to play fast), but not *controlled virtuosity* (the ability to play what you want). This most important issue is addressed in §1.5.2.

1.4.4 An Example Training Set

Obviously the choice of training phrases is crucial to the generation, as only these phrases are used to build the improvisation. Experiments using inputs entered in real time are problematic as errors cannot be corrected once learned by the system. Markov models have not been used, to our knowledge, in a *controlled setting* for jazz improvisation. Here again, the particular context pushes naturally to a careful selection of training patterns, like human improvisers do when they practice. But which phrases are the right phrases?

The example given above suggests a constraint on the training phrase: to ensure continuity (and avoid NSF cases), each Markov model should contain all pitches. This is a sufficient condition, by definition, but not a necessary one. Our repair strategy handles graciously the cases where no solution is found. Other more subtle constraints can be deduced from the desired nature of the improvisations to generate, dealing with longer patterns. For instance, the density of the Markov network determines the nature of the generated phrases: the more choice there is for a single note, the more possibilities there are for controlling the phrase. If a single note has only one possible continuation, there is a risk of producing repeated patterns when reaching this particular note. Note that this is a current situation with human players, who sometimes learn only a small number of escape solutions, when reaching particular notes or passages (on guitar, this is often true for notes played in the top of the neck). A static analysis of the Markov model can reveal such bottlenecks, and be used to suggest new phrases to learn to create new branching points.

To illustrate the generation of phrases from the training phrases, we describe a part of a Markov model, specifically designed to represent a "classical" bebop player, with no particular stylistic influence. We give here the complete set of phrases used in the *minor* scale. These phrases are played in the key of C, and then transposed in the 11 other keys. The interested reader can find the corresponding database for the other scales in C (major, diminished, seventh and whole tone) on the accompanying web site. These other databases are similarly transposed in the 12 keys.

The following six phrases were designed (by the author) to contain basic ingredients needed to produce interesting jazz melodies in C minor. Of course, they do not contain all the patterns of this style, as this would be an impossible task, but they can be enriched at will. As can be seen, not all pitches are present in the database (at least for all octaves). This is intentional to show how the mechanisms we present here interact with each other.

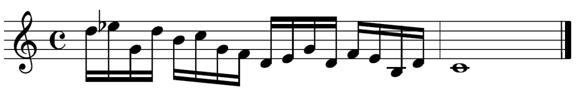

Fig. 1.14 Phrase #1 in C minor.

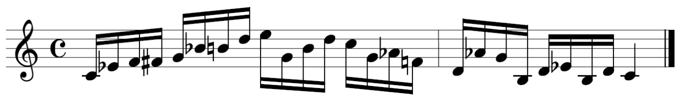

Fig. 1.15 Phrase #2 in C minor.

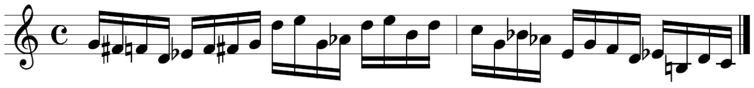

Fig. 1.16 Phrase #3 in C minor.

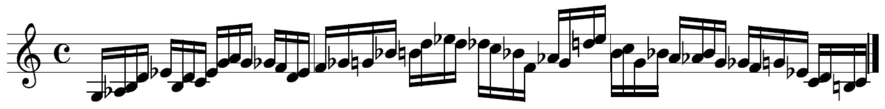
Fig. 1.17 Phrase #4 in C minor.

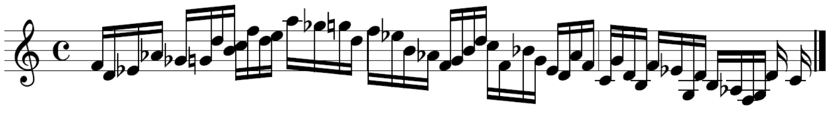
Fig. 1.18 Phrase #5 in C minor.

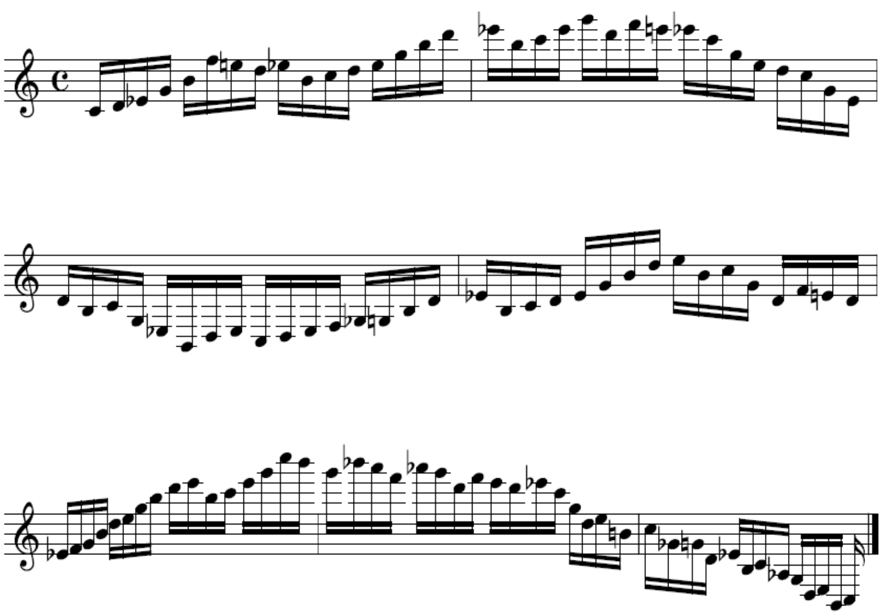
Fig. 1.19 Phrase #6 in C minor.

Figure 1.20 shows a phrase generated on a chord sequence consisting only of a C minor chord. The various segments of the training phrases are indicated, showing how Markov generation creates a new phrase by concatenating bits and segments taken from the training set.

Figure 1.21 shows a phrase generated on the chord sequence *C / B 7— E min / F# 7 — B maj7*, using the training phrases in *minor*, *major* and *seventh* in several keys. The NSF cases and the segments reused are indicated. The phrase produces a perfect sensation of continuity, and the harmony is locally satisfied. Other examples can be found on the accompanying web site.

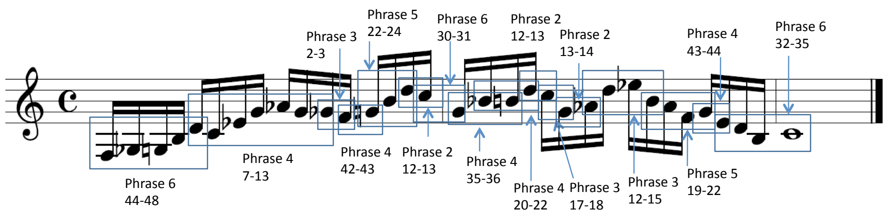

Fig. 1.20 A phrase generated on a C minor chord sequence. The compositions of the segments of the training phrases are indicated by overlapping boxes. Segments of length 2 to 7 are being used in this case. Training phrases #2 to #6 have been used. No NSF case is encountered.

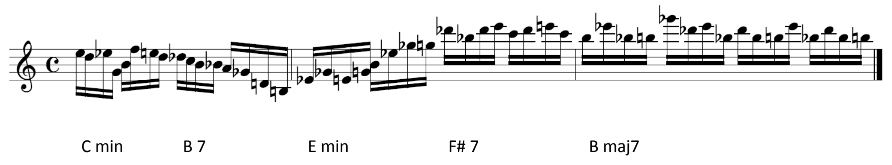

Fig. 1.21 A phrase generated on the sequence *C min / B 7 — E min / F# 7 — B maj7*. Two NSF cases were encountered, indicated by an arrow (and non-overlapping boxes): for the *C min* $\rightarrow$ *B7* transition, and for the *F#7* $\rightarrow$ *B maj7* one. They were addressed using the one-step-max theorem. The discontinuity is not musically perceptible by the author.

1.5 Escaping Markovian Boredom

Once we have established the basis for generating melodies that comply with the rules of the game, we can now describe how to model the two important innovations that bebop has introduced in jazz. These two innovations relate to boredom: producing phrases that satisfy the criteria we have described is pleasing, but may lead to boredom after repeated exposure. From a computer science perspective, we call “Markovian Boredom” the sensation that the phrases generated all come from the same set of rules, grammar, and that, eventually, there is no hope of hearing something new, striking, outstanding. From there on, boredom follows irrevocably.

As described in the introduction, two devices have been invented by jazz musicians to escape boredom. These devices have in turn contributed to changing the style itself. We describe here how these two devices can be modelled in our Markov framework, the issues they raise technically, and how they can be addressed.

1.5.1 Side-Slips and Formal Transforms

The model we have introduced so far generates notes streams on arbitrary chord sequences, which are continuous and satisfy local harmonic constraints. In the examples shown here, we use a limited training material (about six phrases for major, minor and seventh, three phrases for diminished and whole-tone, used in particular for augmented chords). More scales can be added easily (to deal with rarer chords like altered, or minor diminished 5th), but adding more scales or training sequence does not improve substantially the quality of the generation.

It turns out that *playing out* can be easily integrated in our framework. As we have seen, playing out or side-slips may be considered as an excursion from the tonality to another one, followed by a smooth landing to the right tonality. More generally, we can consider side-slips as specific *formal transforms*, operating on, e.g., the last generated phrase. Formally, side-slips can be introduced as yet another case in the *GenerateBeat()* method introduced in section 1.4.2:

```
GenerateBeatAsTransform(context, H, i):
        // context represents the last generated output
        return Transform(context, N)

where Transform is defined for each possible transform, e.g.:

Transform (phrase, N)
        return Transpose (phrase, N, 1) ;
```

The particular side-slip consisting in transposing the last phrase one semitone up, can simply be represented by a transform operation, taking a phrase as input and producing its transposition. Other reasonable bebop transforms include:

- Transposing a semitone, a minor third, a tritone or an octave up or down,
- Reversing then transposing a semitone up or down, as illustrated in Figure 1.22 (4th case).

Transforms can also be used to implement many *tricks* invented by bebop improvisers, such as transposing diatonically the phrase, by taking into account the harmony of the next beat (see the Coltrane or DiMeola examples in §1.2.4).

A most important aspect of formal transforms is the landing strategy: How to come back seamlessly to the original tonality? Our Markov framework provides the landing strategy for free: transforms may produce notes which are out-of-key, but the various strategies we proposed for negotiating chord changes can be used readily to ensure a smooth return to the key. As an example, Figure 1.22 shows a phrase generated on chord sequence composed only of *A minor* chords.

The decision to use a formal transform, again, belongs to the intentional score, i.e. is taken at the beat level. In the case of a purely automatic improvisation system, this decision can be determined by musical heuristics, such as the following:

- When a chord is used for a long time, e.g. more than 1 measure (the original reason for the introduction of side-slips in the first place),
- When a NSF case is encountered (thereby avoiding the use of a repair mechanism),
- When a *direction* is imposed (e.g., go up pitch wise) but no easy way to satisfy it is found (see §1.5.2 on control below).

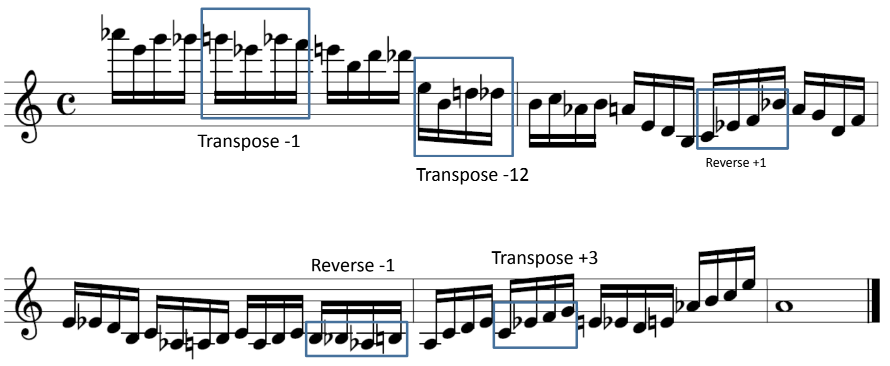

Fig. 1.22 A chorus generated on an *A minor* sequence. Formal transforms are triggered randomly during the chorus, as indicated. Note, and hear, the smooth landings on the A minor key following the transforms, in particular the striking effect produced by the third transform (reverse-1).

It is important to stress out that transforms are grammatical constructs, and as such cannot be learned effectively using any Markov mechanism. Using phrases such as the licks in 4 as training phrases for the corresponding scale (D minor) would blur the very notion of harmony, as notes from the side-slips would be considered as equivalent to notes from the original key. Furthermore, such an approach would require a tremendous amount of training data (for each possible pattern, and each possible side-slip). More importantly, it would then be impossible to trigger *intentionally* decisions to produce, or not, these side-slips.

1.5.2 The Control Issue

Above all, virtuosos can be seen as exceptional humans in the sense that they seem to exert *full control* on their production. The control issue is both the most difficult and the most interesting one to handle.

We can state the control problem as follows: how to generate a sequence that fits an arbitrary criteria, defined by target properties of the next generated phrase? In our context, such properties can be defined in terms of phrase features such as: pitch (go "higher" or "lower"), harmonic constraints ("more tonal notes"), intervals (chromatic), direction (ascending, descending), *arpeggiosity*, etc. Allowing such a level of control in our system is key to producing phrases with musically meaningful intentions.

The difficulty in our case comes from the fact that the generator operates on a note-by-note basis, whereas most criteria useful in practice operate on complete phrases. Let us suppose, for instance, that we want the next generated phrase to be "higher" in pitch than the current one. It does not suffice to simply choose, at the note level, the next note with the higher pitch. Such as policy has been proposed in (Pachet; 2003), and works well for simple harmonisation tasks, but not here, as we want the pitch criteria to hold on an entire next phrase. For instance, a good strategy could consist in first choosing a lower pitch and then playing an ascending arpeggio. So *longer-term planning* is needed to find satisfying, let alone optimal, phrases.

In a Markovian context, control raises a fundamentally difficult problem, because control goes in the way of the basic Markov assumption (see §1.3.2). Indeed, control consists precisely in establishing properties to be satisfied not on "the next item" to play, but on the next sequence of items. Unfortunately, the Markovian view of sequence generation is that the future in only determined by the current state (or the N last current states, depending on the chosen order).

We have addressed this problem from a fundamental perspective, and proposed in (Pachet and Roy; 2011) a general solution to generate Markov sequences satisfying arbitrary properties. This solution consists in reformulating Markov generation not as a greedy algorithm, but as a *constraint satisfaction problem* (Freuder; 1994). Constraint satisfaction is a powerful technique that enables the fast exploration of large search spaces. In our case, controlling a Markov sequence amounts to exploring the space of all possible sequences of length *N* (in our case, *N* is the number of notes per beat). This space can be huge as soon as the training set is large, or the length of the sequence to generate is high.

In this section we illustrate how controlled Markov generation can be used to influence the generation in real-time using meaningful musical criteria. Any criteria can be defined to control sequences, as long as they are computable. We present here a case in which the criteria are scalar values computed from a given sequence with simple *features*, but this scheme can be extended to more complex algorithms, classifiers in particular, as discussed below.

In a first phase, a set of melodic features are defined, such as:

- Mean pitch of a sequence,
- Mean interval of a sequence,
- Tonalness of a sequence.

Tonalness is a scalar value $\in [0,1]$ and gives an indication of how tonal is a melody with regards to the corresponding chord in the sequence. It can be computed using, e.g., a *pitch profile algorithm* (Krumhansl; 1990).

The next step is to substitute the generation of a beat by the corresponding constraint satisfaction problem, as described in (Pachet and Roy; 2011).

We illustrate the mechanism as follows. We start by a phrase played on an *A minor* chord (24). We then generate three beat continuations to fill up a four beat measure (still in *A minor* in our case; changing the harmony has no impact on the control issue). We select the ones maximising criteria we consider useful as *controls*: *higher/lower pitch*, *more/less chromatic*, and *less tonal*. Figures 1.24 to 1.28 show continuations which optimise these five criteria, as confirmed by the values of the features. These values are compared to the initial 4-note phrase values, i.e.:

- Mean pitch: 59.5;
- Mean interval: 2.33;
- Tonalness (in the key of *A minor*): 1.0 (all notes in *A* minor)

```
GenerateBeatMoreThan (context, H, I, BiasCriteria, startValue):
// context represents the last generated output
        State the generation problem as a CSP
        Compute one solution according to
        the time left that optimises the criteria
```

Fig. 1.23 The method for generating a beat according to a bias. We use the approach described in (Pachet and Roy; 2011). The criteria is optimised depending on the time available (using an anytime approach).

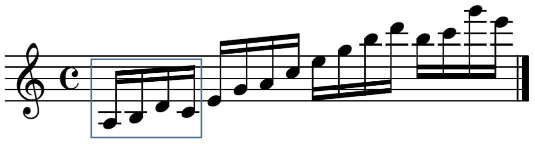

Fig. 1.24 The starting 4-note phrase is in the box. Here, a continuation with "higher pitch" (*mean pitch* = 78.5 $>$ 59.5).

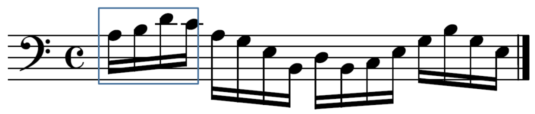

Fig. 1.25 A "lower pitch" continuation (*mean pitch* = 52.41 $<$ 59.5) (here in the bass clef).

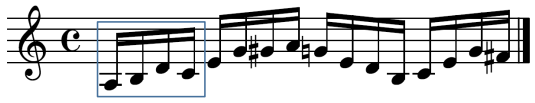

Fig. 1.26 A "more chromatic" continuation (*mean interval* = 0.666 $<$ 2.33).

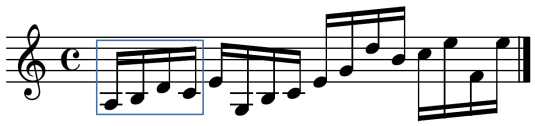

Fig. 1.27 A "less chromatic" continuation (*mean interval* = 2.33). Note the large intervals.

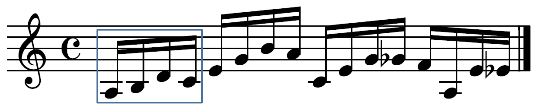

Fig. 1.28 A "less tonal" continuation (*tonalness* = 0.66 $<$ 1.0). Note the Gb and Eb.

It is important to note that this control mechanism is independent from the other mechanisms introduced here, notably formal transforms. Figures 1.29 and 1.30 show a combined use of transforms and controls on the chord sequence of Figure 1.7. It can be checked that indeed the generated phrase do satisfy all the criteria.

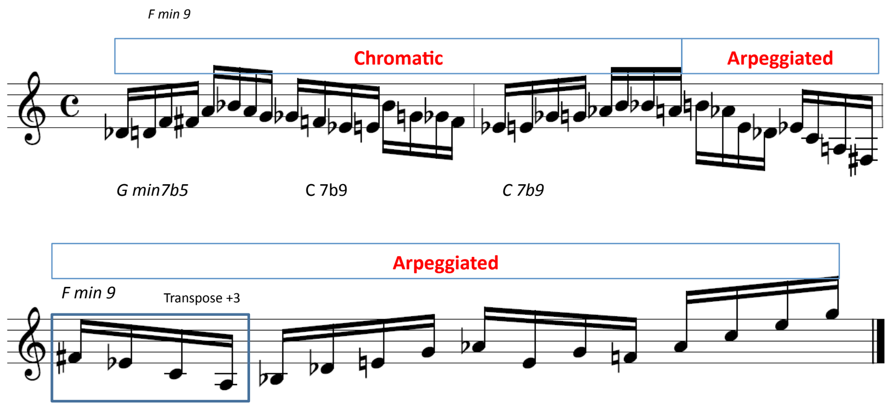

Fig. 1.29 A phrase generated with an intentional score consisting of "chromatic" for the first six beats, and "arpeggiated" for the next six on the same chord sequence, and one random transform. The melody generated fits almost perfectly the constraints.

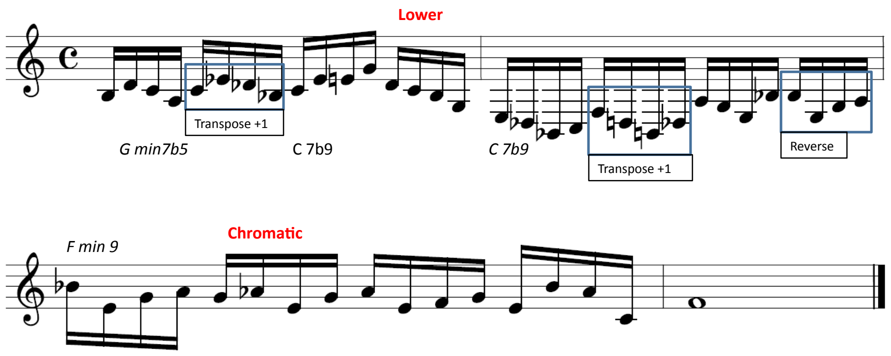

Fig. 1.30 A phrase generated on the chord sequence as Figure 1.7, with three intentionally chosen side-slips and three subjective biases.

1.5.3 Reusing Intentional Scores

The intentional score is the collection of all decisions taken for each beat. As we have seen above, these decisions concern (1) the choice of the rhythm pattern, (2) the choice of the scale (and hence, of the Markov database), (3) the decision to use and the selection of a formal transform, (4) the decision to control the generation with a specific criteria, and (5) the decision to start or stop the phrase. This score is a time line with commands at every beat. An improvisation can be seen as an *interpretation* of this score.

The intentional score represents the "arbitrary" portion of chorus generation, so it cannot be generated automatically. In practice, it can be set randomly, or using an interface, e.g. gestural to produce the various commands in real time, as described in the next section

An interesting application of of concept of intentional score is to induce such an intentional score from an *existing chorus*, to generate a new improvisation with the same *structure*. We illustrate this idea using the chorus shown in Figure 1.7. Of course, there is not a single way to infer the intentional score used by John McLaughlin. The score we consider uses solely "target pitch" subjective biases, extracted from the actual mean pitches of the various beats in John McLaughlin's phrase. It looks as Figure 1.31:

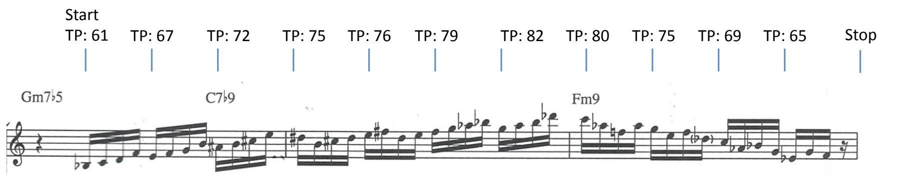

Fig. 1.31 A possible intentional score inferred from the phrase of Figure 1.7. TP denotes the mean MIDI pitch for each beat.

This score can be used to generate the phrase illustrated in Figure 1.32. It can be seen that the resulting phrase follows approximately the same pitch contour. The phrase is not the same, as it uses only the note patterns of our training set, but it gives an indication of how to exploit intentional scores in a creative way.

1.6 Virtuoso: A Virtuoso Enabling Interactive System

Virtuoso is an interactive musical system that implements the ideas presented here so that a user can experience the sensation of being a virtuoso, without having to be one himself. *Virtuoso* is a jazz chorus generator that is controllable in real-time using arbitrary input controllers. The main features we have introduced that account, from our point of view, for a substantial part for the virtuoso aspects of jazz (side-slips

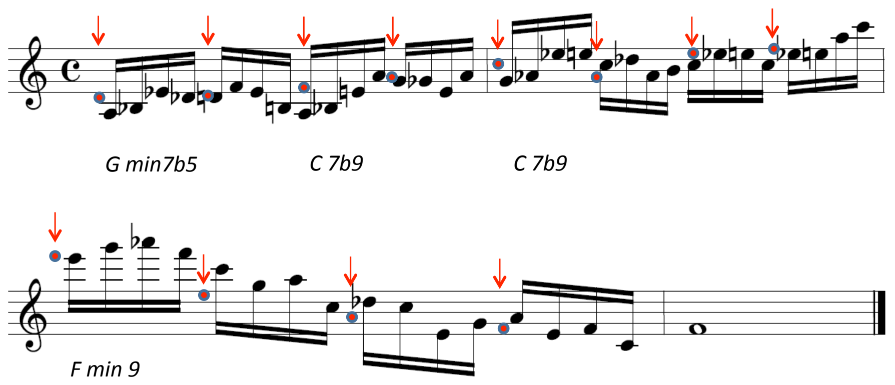

Fig. 1.32 A phrase generated on the same chord sequence as Figure 1.7, with the intentional score induced from John McLaughlin's chorus in Figure 1.31, consisting only of target pitches at every beat, as indicated.

and high-level control) are mapped to various gestural controls, including start, stop, speed (number of notes per beat), side-slips, as well as several criteria to control the generation as described in §1.5.2.

Several videos (Virtuoso; 2011) show the author using the system, as well as intentional scores deployed during the improvisation. A number of experiments were conducted with jazz pianist Mark d'Inverno. An *a posteriori* analysis of the session by the two musicians playing is provided. Although subjective, these analysis show that a sense of *playing together* was achieved, and the music generated by the system, controlled by a human, was of professional-level quality.

1.7 Discussion

The major claim of this study is that all important decisions concerning virtuoso performance in jazz can be taken at the beat level instead of note-level. This explains how virtuosos improvise melodies satisfying so many difficult and contradictory constraints at high speed. By delegating the choice of individual notes within a beat to a non-conscious, sensory-motor level, they have enough time to focus on high-level decisions, such as influencing pitch contour, chromaticity, tonality, etc. Concerning the *memoryless assumption* hypothesised by Longuet-Higgins (see §1.2.1), we invalidate it because of side-slips, which require the memory of the last phrase played. However, the cognitive requirements remain minimal. In some sense, most of the work is done by the fingers.

Conceptually, we do not consider Markov models as representations of musical *ideas*, but as a *texture* that can be controlled to produce meaningful streams of notes. The mechanisms we propose (transforms and controls) turns this texture into realistic, sometimes spectacular, virtuoso improvisations.

Concerning the relation of virtuosity studies to creativity studies, we have stressed the importance of two important dimensions of jazz improvisation (side-slips and fine-control) that are made possible only by *extreme virtuosity*. We have shown how to model these two aspects in a Markovian context. The first one (formal transforms) does not raise any difficult modelling issues. The second one (control) does, and induces a very difficult combinatorial problem. How human virtuosos solve this problem in real-time remains a mystery. It forms important future work for virtuosity studies.

Running is not the only locomotion mode of animals. Likewise, virtuosity is not the only mode of jazz improvisation. Our system is in fact a brittle virtuoso: it knows how to run, but not so well how to walk. Such brittleness was pointed out by Lenat and Feigenbaum (1991) in the context of expert-systems and attributed to a lack of *common sense* knowledge. A *musical* common sense is indeed lacking in most automatic systems, and much remains to be done to build a completely autonomous jazz improviser exhibiting the same level of *flexibility* as humans: a competence in virtuoso mode as well as in other modes, and the ability to intentionally switch between them. *Slow improvisation*, in particular, is a most challenging mode for cognitive science and musicology, as it involves dimensions other than melody and harmony, such as timbre and expressivity which are notoriously harder to model.

However, considering melodic virtuosity as a specific mode, we claim that these automatically generated choruses are the first ones to be produced at a professional level, i.e. that only a limited set of humans, if any, can produce. A claim we leave to the appreciation of the trained listener.

More generally, this chapter is an invitation to elevate virtuosity to a *field of study* for cognitive science and computer science. Its links to creativity have only been sketched here, but they are undoubtedly deeper and yet, unexplored. Understanding virtuosity is a key to understanding creativity, in humans and with machines.

References

Addessi, A.-R. and Pachet, F. (2005). Experiments with a musical machine: Musical style replication in 3/5 year old children, *British Journal of Music Education* 22(1): 21–46.

Aebersold, J. (2000). *The Jazz Handbook*, Aebersold Jazz Inc.
URL: *http://www.violistaz.com/wp-content/uploads/2009/01/ebook-guitar-the-jazz-handbook.pdf*

Assayag, G. and Dubnov, S. (2004). Using factor oracles for machine improvisation, *Soft Computing* 8(9).

Bäckman, K. and P. Dahlstedt (2008). A Generative Representation for the Evolution of Jazz Solos, *EvoWorkshops, vol. 4974 of Lecture Notes in Computer Science*, pp. 371-380, Springer, Berlin.

Baggi, D. (2001). *Capire il jazz, Le strutture dello swing*, Istituto CIM della Svizzera Italiana.

Baker, D. (2000). *Bebop Characteristics*, Aebersold Jazz Inc.

Bensch, S. and Hasselquist, D. (1991). Evidence for active female choice in a polygynous warbler. *Animal Behavior*, 44:301-311.

Biles, J. (1994). Genjam: A genetic algorithm for generating jazz solos, *Proc. of ICMC*, Aarhus, Denmark, ICMA.

Bresin, R. (2000). *Virtual Virtuosity, Studies in Automatic Music Performance*, PhD thesis, KTH, Stockholm, Sweden.

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

Cappellini, G., Ivanenko, Y. P., Poppele, R. E. and Lacquaniti, F. (2006). Motor patterns in human walking and running, *Journal of Neurophysiology* 95: 3426–3437.

Chordia, P., Sastry, A., Mallikarjuna, T. and Albin, A. (2010). Multiple viewpoints modeling of tabla sequences, *Proc. of Int. Symp. on Music Information Retrieval*, Utrecht, pp. 381–386.

Coker, J. (1984). *Jazz Keyboard for Pianists and Non-Pianists*, Alfred Publishing, Van Nuys, CA.

Coker, J. (1997). *Complete Method for Improvisation, revised edition*, Alfred Publishing, Van Nuys, CA.

Conklin, D. (2003). Music generation from statistical models, *Proceedings of Symposium on AI and Creativity in the Arts and Sciences*, pp. 30–35.

Conklin, D. and Witten, I. (1995). Multiple viewpoint systems for music prediction, *Journal of New Music Research* 24: 51–73.

Cont, A., Dubnov, S. and Assayag, G. (2007). Anticipatory model of musical style imitation using collaborative and competitive reinforcement learning, *Springer Verlag LNCS 4520* pp. 285–306.

Cope, D. (1996). *Experiments in Musical Intelligence*, A-R Editions, Madison, WI.

Draganoiu, T. I., Nagle, L. and Kreutzer, M. (2002). Directional female preference for an exaggerated male trait in canary (serinus canaria) song, *Proceedings of the Royal Society of London B*, Vol. 269, pp. 2525–2531.

Ericsson, K., Krampe, R. and C., T.-R. (1993). The role of deliberate practice in the acquisition of expert performance, *Psychological Review* 100: 363–406.

Fitt, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement, *J. of Experimental Psychology* 47(6): 381–391.

Franklin, J. A. (2006). Recurrent neural networks for music computation, *INFORMS Journal on Computing* 18(3): 321–338.

Freuder, E. (1994). *Constraint-based reasoning*, MIT Press.

Gladwell, M. (2008). *Outliers, the Story of Success*, Allen Lane.

Grachten, M. (2001). Jig: Jazz improvisation generator, *Workshop on Current Research Directions in Computer Music*, Audiovisual Institute-UPF, pp. 1–6.

GuitarTrio (1977). *Friday Night in San Francisco, choruses by Al Di Meola, John McLaughlin and Paco De Lucia*, Artist Transcriptions series, Hal Leonard, Milwaukee, USA.

Hick, W. E. (1952). On the rate of gain of information, *Quarterly Journal of Experimental Psychology* 4: 11–26.

Hiller, L. and Isaacson, L. (1958). Musical composition with a high-speed digital computer, *Journal of the Audio Engineering Society* .

Hodgson, P. W. (2006). Learning and the evolution of melodic complexity in virtuoso jazz improvisation, *Proc. of the Cognitive Science Society Conference*, Vancouver.

Holbrook, M. B. (2009). Playing the changes on the jazz metaphor: an expanded conceptualization of music, management and marketing related themes, *Foundations and Trends in Marketing* 2(3–4): 185–442.

Howard, V. A. (2008). *Charm and speed Virtuosity in the performing arts*, Peter Lang, New York.

Johnson-Laird, P. N. (1991). *Jazz Improvisation: A Theory at the Computational Level*, Academic, San Diego, Calif.

Johnson-Laird, P. N. (2002). How jazz musicians improvise, *Music Perception* 19(3): 415–442.

Keller, B., Jones, S., Thom, B. and A., W. (2005). An interactive tool for learning improvisation through composition, *Technical Report HMC- CS - 2005-02*, Harvey Mudd College.

Keller, R. M. and Morrison, D. R. (2007). A grammatical approach to automatic improvisation, *Proc. SMC 07*, Lefkada, Greece.

Krumhansl, C. (1990). *Cognitive Foundations of Musical Pitch*, Oxford University Press, New York.

Lemaire, A. and Rousseaux, F. (2009). Hypercalculia for the mind emulation, *AI Soc.* 24(2): 191–196.

Lenat, D. B. and Feigenbaum, E. A. (1991). On the thresholds of knowledge, *Artificial Intelligence* 47(1–3): 185–250.

Levine, M. (1995). *The Jazz Theory Book*, Sher Music Company, Petaluma, CA.

London, J. (2004). *Hearing in Time*, Oxford University Press, New York.

London, J. (2010). The rules of the game: Cognitive constraints on musical virtuosity and musical humor, *course at Interdisciplinary College (IK)*, Lake Mhne, Germany.

Martino, P. (1994). *Creative Force, Part II*, CPP Media/Belwin, Miami, Fl.

McCorduck, P. (1991). *Aarons Code*, W.H. Freeman & Co.

McLaughlin, J. (2004). This is the way i do it, *The Ultimate Guitar Workshop On Improvisation*, Mediastarz, Monaco. 3 DVD set.

Nierhaus, G. (2009). *Algorithmic Composition, Paradigms of Automated Music Generation*, Springer-Verlag.

O'Dea, J. (2000). *Virtue or Virtuosity?*, Greenwood Press, Wesport, USA.

OuLiPo (1988). *Atlas de littrature potentielle*, Folio/Essais, Gallimard, Paris.

Pachet, F. (2003). The continuator: Musical interaction with style, *Journal of New Music Research* 32(3): 333–341.

Pachet, F. and Roy, P. (2011). Markov constraints: steerable generation of Markov sequences, *Constraints* 16(2).

Papadopoulos, G. and Wiggins, G. (1998). A genetic algorithm for the generation of jazz melodies, *Proceedings of STeP'98*, Jyvaskyla, Finland.

Penesco, A. (1997). *Défense et illustration de la virtuosité*, Presses Universitaires de Lyon.

Ramalho, G. (1997). *Un agent rationnel jouant du Jazz*, PhD thesis, Ph. D, University of Paris 6. **URL:** *http://www.di.ufpe.br/ glr/Thesis/thesis-final.pdf*

Ramalho, G. and Ganascia, J.-G. (1994). Simulating creativity in jazz performance, *Proc. of the 12th National Conference on Artificial Intelligence, AAAI '94*, The AAAI Press, Seattle, pp. 108–13.

Ramirez, R., Hazan, A., Maestre, E. and Serra, X. (2008). A genetic rule-based model of expressive performance for jazz saxophone, *Computer Music Journal* 32(1): 38–50.

Real (1981). *The Real Book*, The Real Book Press.

Ricker, R. (1997). *New concepts in linear improvisation*, Warner Bros Publications, Miami, Fl.

Ron, D. Singer, Y. and Tishby, N. (1996). The power of amnesia: Learning probabilistic automata with variable memory length, *Machine Learning* 25(2–3): 117–149.

Shim, E. (2007). *Lennie Tristano, his life in music*, The University of Michigan Press. Cf. p. 183.

Sloboda, J., Davidson, J., Howe, M. and Moore, D. (1996). The role of practice in the development of performing musicians, *British Journal of Psychology* 87: 287–309.

Steedman, M. J. (1984). A generative grammar for jazz chord sequences, *Music Perception* 2(1): 52–77.

Steen, J. (2008). *Verse and virtuosity, The Adaptation of Latin Rhetoric in Old English Poetry*, U. of Toronto Press.

Stein, L. A. (1992). Resolving ambiguity in non monotonic inheritance hierarchies, *Artificial Intelligence* 55: 259–310.

Sudnow, D. (1978). *Ways of the hand*, Routledge & Kegan Paul Ltd.

Thom, B. (2000). Bob: an interactive improvisational music companion, *Proc. of the Fourth International Conference on Autonomous Agents*, ACM Press, Barcelona, Catalonia, Spain, pp. 309–316.

Tonder, V., J., G., Lyons, M. J. and Ejima, Y. (2002). Perception psychology: Visual structure of a japanese zen garden, *Nature* 419(6905): 359–360.

Ulrich, J. W. (1977). The analysis and synthesis of jazz by computer, *Proc. of IJCAI*, pp. 865–872.

Valéry, P. (1948). Esquisse d'un éloge de la virtuosité, *La Table Ronde* pp. 387–392.

Van Veenendaal, A. (2004). Continuator plays the improvisation turing test, http://www.csl.sony.fr/ pachet/Continuator/VPRO/VPRO.htm.

Virtuoso (2011). Accompanying website, www.csl.sony.fr/Virtuoso.

Voss, R. F. and Clarke, J. (1978). 1/f noise in music: Music from 1/f noise, *J. Acoust. Soc. Am.* 63(1): 258–261.

Walker, W. F. (1997). A computer participant in musical improvisation, *Proc. of ACM International Conference on Human Factors in Computing Systems*, Atlanta, Georgia, pp. 123–130.

Weinberg, G., Godfrey, M., Rae, A. and Rhoads, J. (2008). A real-time genetic algorithm in human-robot musical improvisation, *Proc. of 2007 Computer Music Modeling and Retrieval*, Kronland-Martinet, R. et al. Eds, LNCS 4969, pp. 351–359.