Interactive Free Improvisation Using Time-domain Extrapolation of Textural Features

Stéphan Schaub[1*], **Tiago Fernandes Tavares** [2†], **Adriano Claro Monteiro** [1‡]

[1]Núcleo Interdisciplinario de Comunicação Sonora –Universidade Estadual de Campinas
Rua da Reitoria, 165 – CEP 13083-872, Campinas, SP.

[2]Faculdade de Engenharia Elétrica e de Computação – Universidade Estadual de Campinas
Av. Albert Einstein, 400 – CEP 13083-852, Campinas, SP.

schaub@nics.unicamp.br, tavares@dca.fee.unicamp.br, monteiro.adc@gmail.com

Abstract. *In the following paper a simple interactive system for non-idiomatic improvisation is presented. The general approach assumes that a musician improvising in such a setting does not restrict his output to any fixed alphabet, avoids any pre-established grammar and concentrates on articulating the continuation of a musical flow through constant anticipation of its future developments. Using audio classification techniques we map each played phrase to a vector space representing textural features. The system then deduces possible continuations of the ongoing phrase sequence and re-injects past ones in accordance with (manually controlled) instructions as to "contrast" or to "follow" on its predictions. The system was tested with a professional saxophonist and proved a coherent and responsive environment with a wide range of possible extensions.*

Resumo. *No seguinte artigo, apresentamos um sistema interativo para improvisação musical não-idiomática. A abordagem geral assume que um músico improvisanado nesse contexto não restringe os elementos de sua linguagem musical a um alfabeto fixo, evita gramáticas pré-estabelecidas, e concentra-se na articulação e continuidade do fluxo musical através da constante antecipação de elementos futuros. Usando técnicas de classificação de áudio, mapeamos cada frase tocada e as transpomos em um espaço vetorial que representa características texturais. O sistema então prediz continuidades possíveis para a sequência em curso e re-injeta segmentos passados segundo instruções (controladas manualmente) para "contrastar" ou "seguir" as predições. O sistema foi testado com um saxofonista profissional e demostrou ser um ambiente de improvisação coerente e reativo, além de apontar para possibilidades de futuras ampliações.*

1. Introduction

An automatic interactive improviser is considered here to be a musical system capable of accompanying the playing of one or more live improvising musician(s). It is assumed that the output it provides does not reduce to pre-established sets of formulas but is elaborated "on the fly" from information extracted from the session of which it becomes an active participant. Though no *a priori* restriction applies to the form its outputs must take, a

[*] Supported by FAPESP.
[†] Supported by CNPq.
[‡] Supported by CAPES.

general stylistic and contextual coherence is one of the general aims of the development of such systems.

More often than not, automatic interactive improvisers are applied to free, or at least non-idiomatic, improvisational settings. This may be attributed to the facts that their mastery of truly idiomatic styles are still somewhat sketchy and that the exploration of new musical horizons has been, from the start, an important motivation behind their development. Free improvisation, however, does not solely - or even at times primarily - rely on the pitch dimension to develop its musical discourse. Microtonal inflections, “gestures”, alternative modes of playing, etc. might play a central role and bring the “sounding” or “textural” dimensions more or less explicitly into the foreground. Furthermore, as no pre-established grammar or *referent* (in Pressing’s sense [1]) is supposed, the principles from which continuity and general coherence of the musical flow are supposed to follow are difficult to pinpoint.

In the following contribution, we present a simple automatic interactive improviser for non-idiomatic improvisation the development of which was premised on the following two principles. First it had to avoid any reliance on a pre-existing alphabet to accommodate more “textural” types of features. Second, rather than to prolong a given *context*, it would base its responses on an anticipation of the musical session’s current developments into the immediate future.

In the approach proposed here the system captures the playing of an improvising musician, segments it into phrases and stores these alongside a (possibly extendable) set of “textural” features. Basing its anticipation on this latter information using linear prediction, it then re-injects past phrases into the present in accordance with a (manually controlled) instruction as to “contrast” or “follow” on its prediction. The resulting system was then tested for coherence and reactivity in improvisation sessions with a professional saxophonist. Though it is far from bringing the principles on which its development was premised to a close, it proved a flexible environment from which a wide range of extensions and applications can be envisioned.

After a brief overview of some related work (section 2), the proposed system will be presented in details (section 3). This will be followed by the description and analysis of the tests that have been conducted (section 4) followed by some general comments concluding remarks (section 5).

2. Related work

Since Lewis’s first experiments with Voyager in the 1980’s [2] a number of digital systems have been proposed that can be considered as falling into the definition proposed in our introduction. Some, however, consider that “the interactive nature of a system can be obscure, mysterious and opaque” [4] and have preferred to shift efforts away form attempting to produce plausible imitations of interactive behavior towards more general criteria and the designing of solutions that draw from a variety of techniques ([3], [5]).

Amongst the systems that are concerned with such imitation, an important family has taken on from research conducted in musical style modeling and imitation that goes back to the earliest days of computer applications to music and the work of Hiller and Isaacson [6]. The algorithms developed in this context have attained high level of sophistication as attested, for instance, by [7] and [8]. The transposition to interactive improvisation has drawn from a variety of algorithmic resources. Pachet’s Continuator [9], for instance, uses probabilistic decision trees, while OMax by Assayag *et al.* [10] uses factor oracles. All of these “learn” either from a corpus or from the past of the ongoing session to produce stylistically suitable continuations based on the most recent past of the improvisation (called the *context*).

Other types of approaches must also be mentioned. Some, such as Biles's GenJam [11], use genetic algorithm. In this particular case, phrases of fixed length are generated in response to the author's own playing. Neural networks have also been experimented with as exemplified by [12]. All these examples require lengthy, supervised, processes to train their underlying algorithms.

The notion of anticipation (or expectation) has integrated music theory and analysis through the pioneering work of Meyer [13] and known important developments ever since (see, for instance, [14]). It is also part of the psychology of improvisation behavior at least since Pressing introduced it into the field in [15]. More recently, it made its way into style analysis and imitation (see Conklin [7]) and into the development of automatic interactive improvisers, with a notable contribution provided by Cont *et al.* [17]. All the approaches involving the notion of anticipation the processes are based predominantly on structures extracted from pitch configurations.

The system described below can be considered as loosely inspired from the latter approach in that its responses are based on extrapolations of a current *context* into the future, rather than on its continuation *per se*. Though the implementation presented here does rely on pitch, it does not in principle depend on it as decisively. Indeed, it builds, for each incoming phrase, a vector containing features that describe its *overall* textural profile. The system extrapolates a possible continuation using a linear predictor (operating in a continuous space) and re-injects segments from the session's past based on this prediction. This system is now described in more detail.

3. Proposed system

Our system articulates a total of four steps. The first is responsible for recording the audio stream incoming from (for the time being) one live improviser and for segmenting it into successive phrases s_t, $t = \{1, 2, 3, ..., T\}$. The second calculates an acoustic feature vector $\boldsymbol{x}_t$ that maps s_t into a N-dimensional vector space $\mathbb{R}^N$. The third calculates a prediction of what the vector s_{T+1} might be. The fourth, finally, plays, at each trigger, one of the past phrases in accordance with instructions as to "contrast" or "follow" on its prediction. Each step is described in more details below.

3.1. Recording and Segmentation

Segments have been defined as audio slices having amplitude above a certain threshold and boundaries between two silent moments (i.e. with values below the threshold). This decision was based on the preliminary assumption that silences naturally articulate musical ideas and phrases. In order to capture points of segmentation, a simple threshold was set for a low-passed RMS function extracted from the incoming signal. The threshold value is determined to each performance by capturing the amplitude of the background noise from the recording environment and adding 6 to 9 dBs to it.

3.2. Mapping

The mapping module assigns a feature vector defining textural characteristics of the segments. To generate it, low-level features – pitch (with the octave divided into 24 microtones) and RMS – are extracted from the audio segments. These are calculated by using Sigmund, a standard pitch tracking algorithm in Pure Data [18]. For each segment, the chronologically ordered list of note durations is determined by considering the pitch curve as a sequence of time intervals between changes of pitch. Finally a higher-level feature vector $\boldsymbol{x}_t$ is generate that comprises:

1. The mean, standard deviation and slope (angular coefficient of the linear regression) estimated over time from RMS, pitch and duration curves;
2. The Pearson correlation between each two of the three feature curves;
3. The total duration of the audio segment;
4. A 24 chroma pitch-histogram, obtained by the pitch curve, defining the harmonic/intervallic characteristic of the segment.

3.3. Forecasting

The forecast module assumes that the feature vectors $\boldsymbol{x}_t$ form a time series $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T$. Standard forecasting procedures can be performed to estimate a value for the next vector $\boldsymbol{x}_{T+1}$. To obtain this we used a linear predictor, based on the equation:

$$\boldsymbol{x}_M = \sum_{k=1}^{K} \boldsymbol{a}_k \boldsymbol{x}_{M-k}, \tag{1}$$

where the prediction coefficient vectors $\boldsymbol{a}_k$ are obtained by minimizing the average estimation error over the training data and K corresponds to the predictor order, set *a priori*. Order $K = 5$, performed well while allowing the system to become operational from the onset of the improvisation. The predictor was obtained using standard pseudo-inversion techniques and is re-computed with each new incoming phrase.

In another possible setting, used mainly for testing, the predicted vector $\boldsymbol{x}_{T+1}$ is systematically taken to be the last completed phrase ($\boldsymbol{x}_T$). Transition between the two modes can be affected in real time if desired.

3.4. Output

At each (for the time being manually determined) trigger, the system re-injects a past phrase into the present of the improvisation. The selection process is based on a contrast factor γ. The system randomly selects a feature vector $\boldsymbol{x}_k$ among the $\lfloor \gamma(T-1)+1 \rfloor$ ones that are closest (in Euclidian terms) to the estimated $\boldsymbol{x}_{T+1}$. The past phrase re-injected is then s_k to which the feature vector s_k corresponds. In the context of the present work, the material was replayed literally or slightly transformed through timber distortion, but without impeding its recognition.

4. Preliminary testing

Once in place, the system was tested with a professional saxophone player in a series of improvisation sessions. The system was controlled by one of the authors of this paper. A total of six short sessions, as described in Table 1, were played and analyzed. The improviser was only informed on how the system works after the 3rd session was completed.

As hinted in Table 1, the system's parameters were kept fixed during each the first four sessions and varied freely during the last two. In the fourth session, the saxophonist was instructed to avoid providing the progression with a clear direction and to provide maximum contrast at all time. For the last two sessions no particular instruction were given. The first four and the last two are now considered in turn.

Piece	Description
1	Prediction is equated to last phrase, minimum contrast
2	Prediction based on forecast, maximum contrast
3	Prediction based on forecast, minimum contrast
Musician is informed on how the system works	
4	Prediction based on forecast, minimum contrast, saxophonist asked to intentionally avoid providing the progression with a clear direction
5	All parameters freely determined on-the-fly
6	All parameters freely determined on-the-fly

Table 1: Description of the pieces used for testing.

4.1. Leaving the parameters fixed

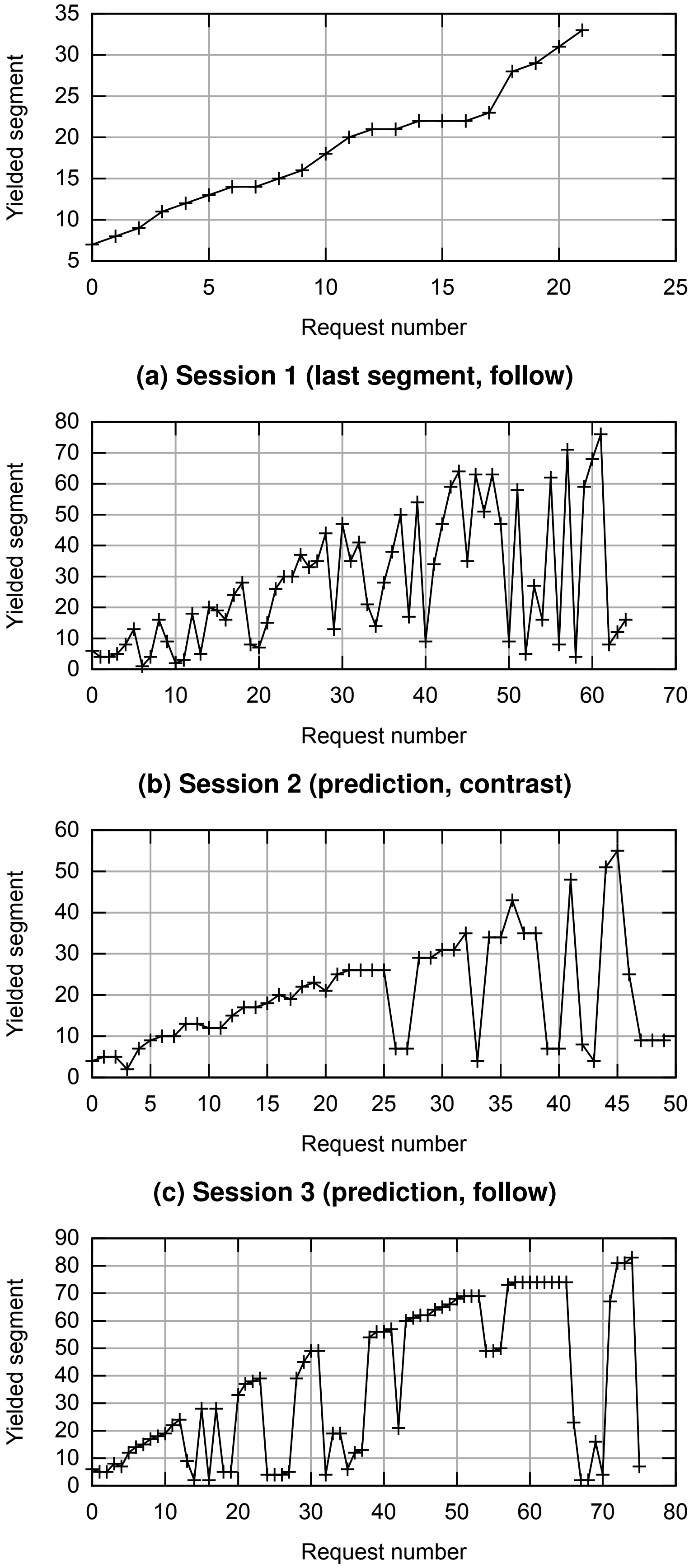
To observe the general behavior of the system the paths it took through past segments were plotted. The results are shown in Figure 1, whereby the y-axis represents to the (chronologically indexed) segments from the improviser and the x-axis represents to the (chronological) succession of triggers.

Figure 1a presents an entirely predictable behavior in which the last completed segment is systematically repeated. On the contrary, Figure 1b displays a highly erratic behavior, in which it is hard to detect a pattern. In both cases, the observed behaviors are precisely the ones that the settings used would have us expect.

When the system is set to find a close match to the continuation it extrapolates, its behavior is neither erratic nor entirely predictable, as shown in Figure 1c and 1d. In most cases, the system tends to present re-configurations of segments taken form the relatively recent past of the improvisation but without displaying the same predictability observed in the first session. This type of behavior is dominant in the first half of Figure 1c and alternates in Figure 1d. This behavior could be traced back to the fact that the improviser tended to project similar or gradually evolving musical ideas over time spans that included several consecutive segments.

Under this same configuration, the system also retrieved segments from a more distant past. It did so, however, less often than in the second session. Also, it was observed that these segments presented certain coherence with the current musical context. Figure 2 illustrates this by showing spectrograms of the saxophonist's and of the system's outputs (see Table 1). This example corresponds to segments 27 and 28 in Figure 1c.

Two visually distinguishable spectral textures can be seen in both images. The first is formed by sustained notes played in the higher register while the second is formed by notes in the lower register repeated in rapid successions. The later is produced simultaneously by the saxophonist and by the system. The one played by the system, however, had been retrieved from a more distant past and corresponds to the 7th phrase played by the musician.



(a) Session 1 (last segment, follow)

(b) Session 2 (prediction, contrast)

(c) Session 3 (prediction, follow)

(d) Session 4 (musician creates contrast)

Figure 1: Re-injected segment for each request in sessions 1 to 4.

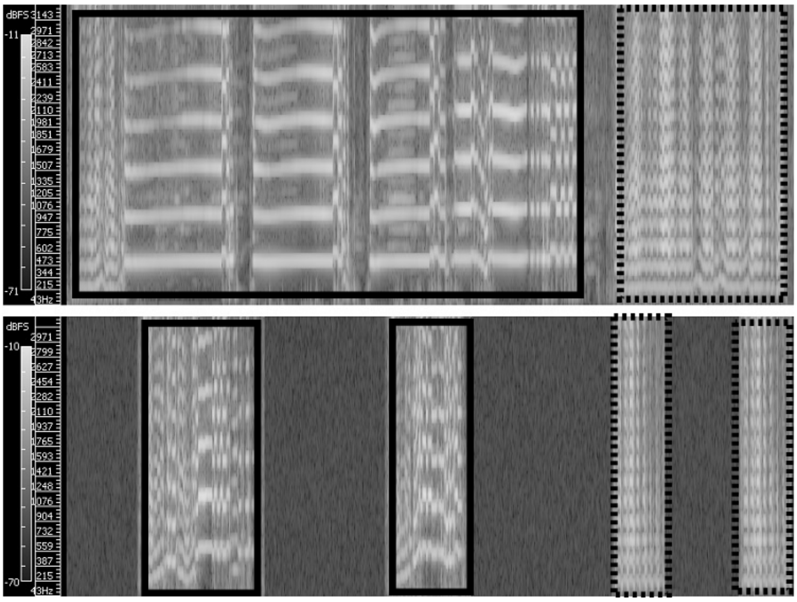
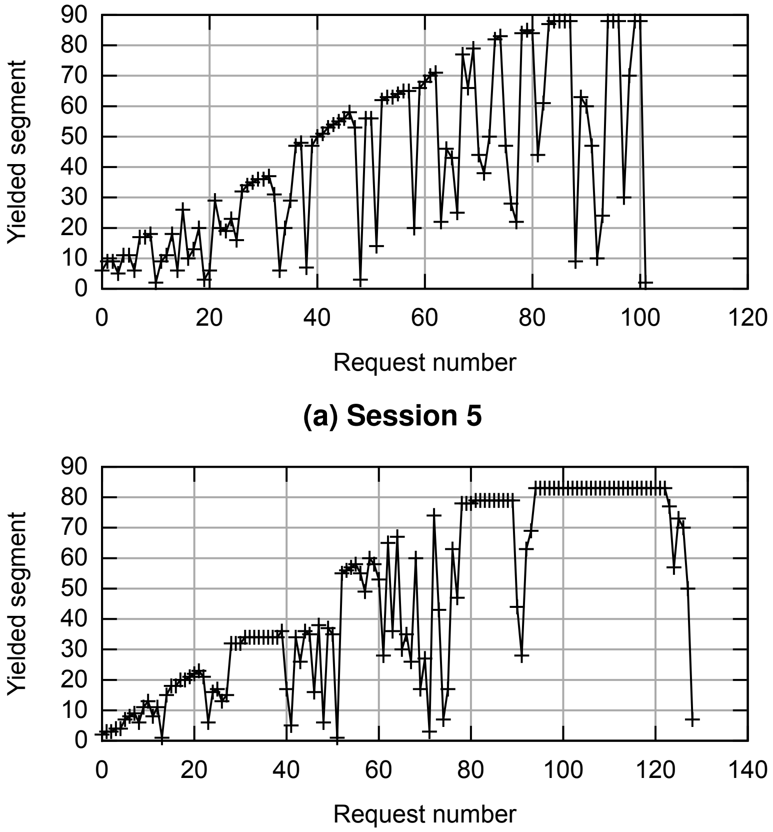


Figure 2: spectrogram

4.2. Changing parameters on-the-fly

Figure 3 illustrates the system's behavior in sessions 5 and 6, as described in Table 1. In these sessions, the system's parameters were changed on-the-fly by a human musician. It is possible to see some stable and unstable parts, combining the behaviors observed in Figure 1.



(a) Session 5

(b) Session 6

Figure 3: Re-injected segment for each request in free improvisation sessions.

In Figure 3a, the system's reactivity can be particularly clearly observed between requests 26 and 45. The system presented a very stable behavior until request 30, when it started behaving more erratically. Figure 4 depicts both the re-injected segments and the instrument parameters during this excerpt. As can be seen, the contrast parameter was null and, raised at request 30, caused the system to select segments that are more distant from the current one. After that, the contrast parameter varied, maintaining a similar behavior. After request 40, when the contrast was set back to null, the system returned to stability.

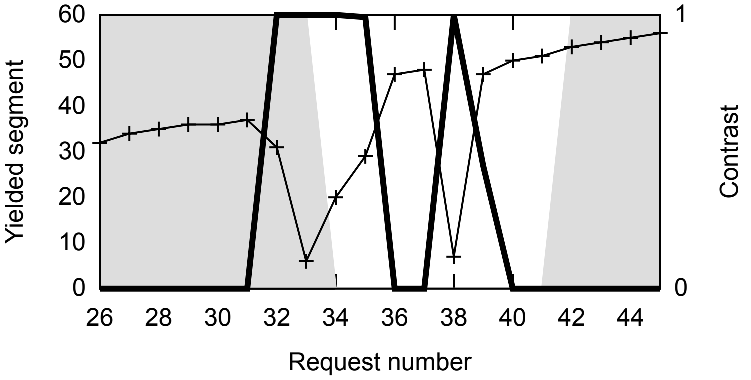


Figure 4: Re-injected segments (line with dots), contrast parameter (thick line) and reference for selection (gray background for prediction, white background for last sample).

A similar phenomenon can be observed in Figure 3b between requests 80 and 100. As shown in Figure 5, the unstable gap is created between two evidently stable parts by changing the value of the contrast parameter.

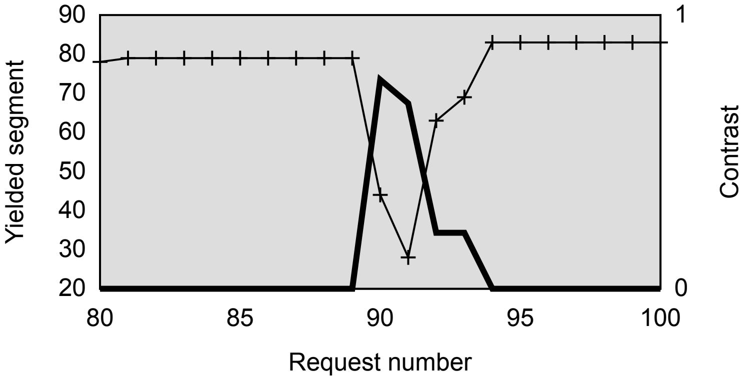


Figure 5: Re-injected segments (line with dots), contrast parameter (thick line) and reference for selection (gray background for prediction, white background for last sample).

The reactivity of the system can also be visualized in Figure 3b between requests 30 and 43. As shown in Figure 6, the user breaks the stability by raising the contrast level. During the stable part of this example, the parameters were set as in session 1.

These analyses show that the user can directly induce the system to respond in certain ways depending on his perception of the ongoing session. While the linear prediction seeks a coherent continuation and may reinforce a stable behavior for the interaction, the contrast factor can serve to re-inject musical ideas more removed from the present contexts, potentially inducing new directions to the musical progression.

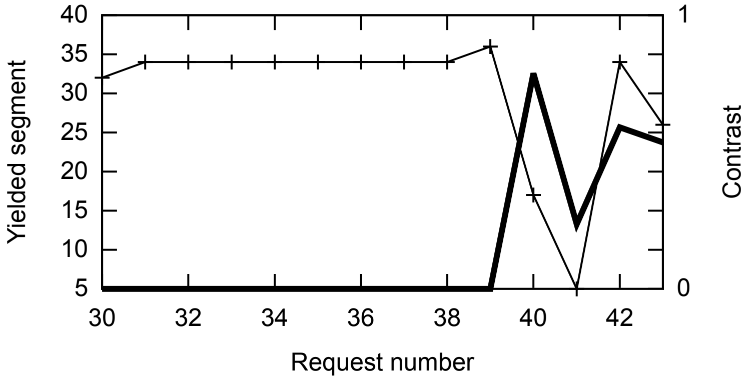


Figure 6: Re-injected segments (line with dots), contrast parameter (thick line) and reference for selection (gray background for prediction, white background for last sample).

5. Evaluation and Concluding remarks

The observations made during the tests just described showed the system to be both responsive and, in a certain narrow sense at least, coherent. An aspect that is much more difficult to translate into objective terms, but which was clearly present during these tests, is the playful quality the musical exchanges between the improviser and the person operating the system could be endowed with. Such observations were particularly encouraging when considered against the system's relative simplicity.

Concerning the two principles on which the development of the system was premised: the non-reliance on a fixed alphabet and the use of anticipation as a basis from which to determine outputs; both can be considered to having been met. The question remains open as to their true contribution to the quality of exchange just mentioned. As in any system that is supervised in real time, it is unclear to what extent success (or failure) is due to the human intervention or to the system's orderly behavior.

This problematic, whether answerable or not, directly suggests two directions in which future developments could be taken: that of a greater autonomy of the system and that of a more detailed and controlled anticipatory procedures. For the first, some principle determining the calling of new phrases (triggers) would have to be devised as well as criteria for the automatic selection of values for the contrast factor. For the second, the exploration of further sets of features, with the possibility to control their impact through weighting or through their partitioning into distinct sets could be envisioned. The same can be said of the mechanism underlying the anticipation procedure itself. Indeed, nothing impedes combining several, possibly concurrent, principle at the same time.

Each of these developments, from their prototyping to their eventual integration, can be undertaken in separate steps and accompanied by live testing.

Finally, yet another type of development can be considered. As the system's external controls are both simple and intuitive, setups can be envisioned in which the performer himself is in control through some form or another of augmented performance environment.

6. Acknowledgements

The authors would like to thank Prof. Rogério Costa, the saxophonist (and scholar of non-idiomatic improvisation) who has generous donated some of his time and lend his skills to the benefit of this project. Thanks also go to the institutions that have supported this research, FAPESP; CNPq and CAPES.

References

[1] Pressing, J. (1998) ”Psychological Constraints on Improvisational Expertise and Communication”, in *In the Course of Performance: Studies in the World of Musical Improvisation*. Ed. Nettl B. and Russel, M. University of Chicago Press, Chicago, p. 53-74.

[2] Lewis, G. (2000) ”Too Many Notes: Computers, Complexity and Culture in VoyagerÓ, *Leonardo Music Journal*, v.10,p. 33-39.

[3] Bown , O. (2011) ÒExperiments in Modular Design for the Creative Composition of Live AlgorithmsÓ. *Computer Music Journal*, 35:3, pp. 73-85.

[4] Bown , O., Eldriedge A., McCormack, J.(2009) ÒUnderstanding Interaction in Contemporary Digital Music from Instruments to Behavioural ObjectsÓ. *Organised Sound*. 14:2, pp. 188-196.

[5] McLean, A., Wiggins, G. A. (2010) ÒBricolage Programming in the Creative ArtsÓ In: http://yaxu.org/writing/ppig.pdf. Accessed 15/10/2013.

[6] Hiller, L.A, Isaacson. L. M. (1959) ÒExperimental Music: Composition with an Electronic ComputerÓ. McGraw-Hill Book Company, New York.

[7] Conklin, D. and Witten, I. H. (1995-2002) ”Multiple Viewpoint Systems for Music Prediction”, *Journal of New Music Research*, Vol 24/1,1995, p. 51-73, (revised version 2002).

[8] Cope, D. (2004) *Virtual Music: Computer Synthesis of Musical Style*, MIT Press.

[9] Pachet, F. (2002) ”The Continuator: Musical Interaction With Style”, In *Proceedings of the ICMC*.

[10] Assayag, G. and Bloch, G. and Chemillier, M. and Cont, A. and Dubnov, S. (2004), ”Omax Brothers: a Dynamic Topology of Agents for Improvization Learning”, In *Workshop on Audio and Music Computing for Multimedia*.

[11] Biles, J. A. (1994) ”GenJam: A Genetic Algorithm for Generating Jazz Solos”, In *Proceedings of the ICMC*.

[12] Franklin. J.A. (2004), ÒPredicting reinforcement of pitch sequences via lstm and tdÓ. In: *Proceedings of International Computer Music Conference* (ICMA), Miami.

[13] Meyer, Leonard B.(1956). ÒEmotion and Meaning in Music. Chicago University Press, Chicago.

[14] Huron, D. (2006) Sweet Anticipation: Music and the Psychologie of Expectation. MIT Press, Cambridge.

[15] Pressing, J. (1988) ”Improvisation: Methods and Models” In, Sloboda, J. *Generative Processes in Music*, Clarendon, Oxford, pp. 129-178.

[16] Conklin, D. (2003) ”Music Generation from Statistical Models”, In *Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, Aberystwyth, Wales,pp. 30-35.

[17] Cont, A., Dubnov, S. and Assayag, G., (2007) ”Anticipatory Model of Musical Style Imitation Using Collaborative and Competitive Reinforcement Learning”, In Butz, M. V., Sigaud, O, Pezzulo, G, and Baldassarre, G., Eds *Anticipatory Behavior in Adaptive Learning Systems*, Springer-Verlag, Berlin, Heidelberg, p. 285–306.

[18] Puckette, M. S. and Brown, J.C. (1998) ”Accuracy of frequency estimates using the phase vocoder”, In *Speech and Audio Processing, IEEE Transactions on*, v. 6:2,pp. 166-176.