

Reflexive Loopers for Solo Musical Improvisation

François Pachet

Sony CSL
6, rue Amyot
pachet@csl.sony.fr

Pierre Roy

Sony CSL
6, rue Amyot
roy@csl.sony.fr

Julian Moreira

Sony CSL
6, rue Amyot
moreira@csl.sony.fr

Mark d’Inverno

Goldsmiths &
Sony CSL
dinverno@gold.ac.uk

ABSTRACT

Loop pedals are real-time samplers that playback audio played previously by a musician. Such pedals are routinely used for music practice or outdoor “busking”. However, loop pedals always playback the same material, which can make performances monotonous and boring both to the musician and the audience, preventing their widespread uptake in professional concerts. In response, we propose a new approach to loop pedals that addresses this issue, which is based on an analytical multi-modal representation of the audio input. Instead of simply playing back pre-recorded audio, our system enables real-time generation of an *audio accompaniment* reacting to what is currently being performed by the musician. By combining different modes of performance – e.g. bass line, chords, solo - from the musician and system automatically, solo musicians can perform duets or trios with themselves, without engendering the so-called *canned (boringly repetitive and unresponsive) music* effect of loop pedals. We describe the technology, based on supervised classification and concatenative synthesis, and then illustrate our approach on solo performances of jazz standards by guitar. We claim this approach opens up new avenues for concert performance.

Author Keywords

Music interaction; loop pedals; classification; synthesis.

ACM Classification Keywords

H.5.5 Sound and Music Computing -- Signal analysis, synthesis, and processing – Systems. J.5 -- Performing arts (e.g., dance, music)

INTRODUCTION

Solo improvised performance is arguably the most challenging situation for a musician and especially so in jazz. The main reason is that in order to produce an interesting musical discourse, many dimensions of music have to be performed simultaneously, such as beat, harmony, bass and melody. A solo musician has to incarnate the different roles of a whole rhythm section as happens in standard jazz combos with piano, bass and drums. Not only do they have to maintain a rhythm section they also have play a solo such

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2013, April 27–May 2, 2013, Paris, France.

Copyright © 2013 ACM 978-1-4503-1899-0/13/04...\$15.00.

as a tune or improvisation on top of it. Technically this is possible only for few instruments like the piano, but even in that case it requires great virtuosity. For guitars, solo performance is even more challenging as the configuration of the instrument does not allow for multiple simultaneous music streams. In the 80s, virtuoso guitarist Stanley Jordan stunned the musical world by playing simultaneously bass, chords and melodies using a technique called “tapping” [5]. But such techniques are hard to master, and the resulting music, whilst at first exciting, is arguably stereotyped.

Several technologies have been invented to cope with the limitations of solo performances by aiming to extend their expressive possibilities. One of the most popular is the *loop pedal* [2]. Loop pedals are digital samplers that record music input during a certain time frame typically determined by the clicking of a foot pedal. Figure 1 shows a typical use. A first click activates the recording of the input. A subsequent click determines the length of the loop and starts the playback of the recorded loop (Figure 1).

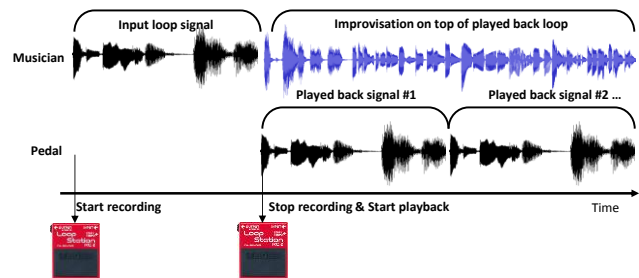


Figure 1. Typical loop pedal interaction.

With loop pedals the musician typically records a sequence of chords or bass line and then improvises on top of it. This scheme can be extended to stack up several layers, e.g., chords then bass using other interactive widgets, e.g., double clicking on the pedal. Loop pedals enable musicians to play several tracks of music in real-time. However, they produce a *canned music effect* due to the systematic repetition of the recorded loop without any variation whatsoever.

Another popular and inspiring device for enabling solo performance is the “minus-one” recording, such as the Aebersold series [1]. With these recordings, the musician can play a tune with a fully-fledged professional rhythm section. Though of a different nature, the *canned effect* is still present. First, the recording is unresponsive. Second, playing with a recording generates *stylistic mismatch*, as the style of the musician may not be compatible with the style of recorded musicians. Consequently, these devices are hardly

used in concerts or recordings, and their usage remains limited to practice or *busking* (outdoor playing).

Previous works have attempted to extend traditional instruments, such as the guitar, by using real-time signal analysis and synthesis. For example, [6] showed how to detect fine-grained playing modes from the analysis of the incoming guitar signal, and [9] proposed a rearranging loop pedal that detects and randomly reshuffles note events within a loop. In [4], a MIDI-based model of an improviser’s *personality* is proposed, to build a virtual trio system, but it is not clear how it can be used in realistic performance scenarios which require a predetermined harmony and tempo. An audio-based method for generating stylistically consistent phrases from a guitar or bass is proposed in [3]. But this technique can only be applied to monophonic melodies. Omax is a system for live improvisation that generates musical sequences built in real-time from a live source [7] using feature similarity and concatenative synthesis. Omax is suitable for free musical improvisation. In contrast, our system is intended for traditional jazz improvisation involving harmonic and temporal constraints and combining heterogeneous instruments and playing modes.

Observing real jazz combos gives clues to what a natural extension of a jazz instrument could be. In a jazz duo for instance, musicians typically alternate between comping (providing harmony with chords) and playing a solo line. Each musician also adapts in a mimetic way to the others, for instance in terms of energy or pitch. Based on these observations, we propose *Reflexive Loopers*, a novel approach to loop pedals that enables musicians to expand their musical competence as if they were playing in a duo or trio with themselves, but which avoids the canned music effect of pedals or minus-one recordings. This is achieved by 1) enforcing *stylistic consistency* (no pre-recorded material is used), while 2) allowing *natural interaction* between the human and the played-back material. Next, we describe our approach that uses *concatenative synthesis* and *feature-based similarity* illustrated by a solo guitar performance.

SYSTEM DESCRIPTION

Reflexive Loopers (RLRs) follow the same basic principle as standard loop pedals: they play back music material performed previously by the musician. RLRs differ in two aspects: firstly the playback material is determined not only according to the current position in the loop, but also to a predetermined *chord sequence*, and to the current playing of the musician through *feature-based similarity*. This ensures that any generated accompaniment actually follows the musician’s playing. Secondly, RLRs manage to differentiate between several *playing modes*, such as bass, harmony (chords) and solo. Depending on the mode the musician is playing at any point in time, the system will play differently, following the “*other members*” principle. For instance, if the musician plays a solo, the RLR will play bass and chords. If the musician plays chords, the RLR will play bass and solo and so on. This rule ensures the overall

performance is close to natural music combos where bass, chords and solo are always present but never doubled.

In our current implementation, input comes both as an audio and a MIDI stream. Audio is used for synthesis and MIDI for analysis and classification (see Figure 2).

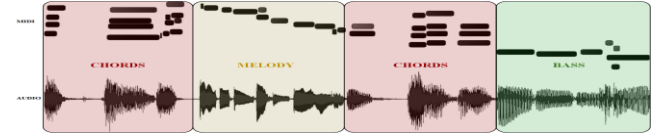


Figure 2. Classification of the musician’s input in different modes. MIDI is used for mode classification and audio for extracting interaction features and concatenative synthesis.

The Chord Grid

Like in many jazz accompaniment systems, a chord sequence (called a *chord grid*) is provided a priori (see Figure 3), as well as a tempo. The grid is used to label each played beat with the corresponding chord. A hard constraint imposed on RLRs is that each played-back audio segment should correspond to the correct chord in the grid. Because a grid often contains several occurrences of the same chord the system can reuse any recording for a chord at several points in the grid. This increases its ability to adapt to the musician’s playing as there is more content to choose from.

<i>C min</i>	<i>C min</i>	<i>G-7</i>	<i>C7</i>
<i>F maj7</i>	<i>F maj7</i>	<i>F-7</i>	<i>Bb7</i>
<i>Eb maj7</i>	<i>Eb-7 / Ab7</i>	<i>Db maj7</i>	<i>D-7 / G7</i>

Figure 3. A typical chord grid (The tune *Solar* by Miles Davis). Some chords are repeated (e.g. here, *C min* and *F maj7*), providing more choice for the system during generation.

Multi-modal classification

This subsection describes how the system automatically classifies the musician’s performance into (*playing*) *modes*. These modes map to different ways of playing a musical instrument such as a guitar or piano and in this experiment are categorized as **bass**, **chords**, and **melody**.

Table 1. Number of recorded bars in each mode

Tune	Number of chunks		
	Bass	Melody	Chords
Bluesette	106	103	106
Lady Bird	167	169	168
Nardis	221	220	219
Ornithology	226	227	226
Solar	145	146	147
Summer Samba	190	194	194
The Days of ...	141	143	143
Tune Up	126	129	128

Mode classification uses a *supervised learning* approach. We recorded the MIDI signal of a guitarist improvising with a Godin MIDI Guitar on eight standard jazz tunes in various tempos and rhythmic feels (*Bluesette*, *Lady Bird*, *Nardis*, *Ornithology*, *Solar*, *Summer Samba*, *The Days of Wine and Roses*, and *Tune up*). For each tune, we recorded three performances, one in each mode: bass, chords, and solo melodies. The MIDI input was segmented into non-

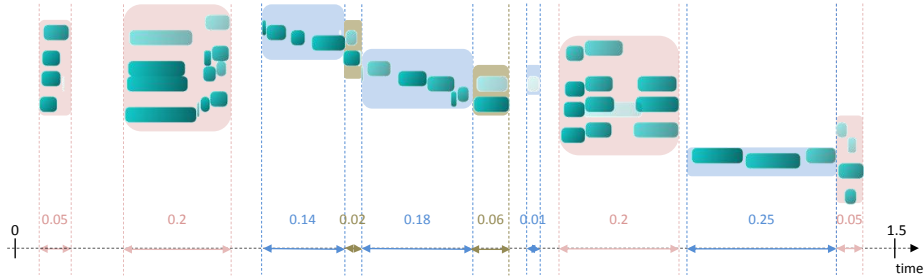


Figure 4. Computing the melody, chord, and interval duration: pink (≥ 3 notes playing together) = chords; blue (1 note) = melody; brown (2 notes) = interval. Here, the cumulated durations of chords, melodies, and intervals are 0.5 s, 0.68 s, and 0.08 s. Therefore chord-dur-ratio = 33%, melody-dur-ratio = 45%, interval-dur-ratio = 5% (the rest is silence).

overlapping one-bar ‘chunks’ at the given tempo. Chunks are *not* synchronized to the beat in order to ensure that the resulting classifier performs well on musical input that is *out of time*, a common technique used in jazz.

We considered an initial feature set with 18 MIDI features related to pitch, duration, velocity, and a specific *bar structure feature* (see Figure 4). The 315 bars of the first tune, Bluesette, were used for feature selection and to train the classifier. First, we used the Correlation-based Feature Subset Selection algorithm in its Weka implementation (*CfsSubsetEval* with the *BestFirst* search method). Eight features were selected: **full-cluster-ratio**, **mean-pitch**, **highest-pitch**, **lowest-pitch**, **mean-inter-onset**, **inter-onset-variance**, and **melody-dur-ratio**, and **chord-dur-ratio** (Table 2). Second, we trained a Support Vector Machine classifier on the training bars with these 8 features.

Table 2. The 8 selected features. A cluster is a group of at least two notes played together, i.e., overlapping by more than 75%. An event is a note or a cluster.

Name	Description
full-cluster-ratio	The number of clusters / number of MIDI events
mean-pitch	Avg. pitch of MIDI events. For clusters: the mean pitch of the notes in the cluster
highest-pitch	The highest MIDI pitch
lowest-pitch	The lowest MIDI pitch
mean-inter-onset	Duration between onsets of consecutive events.
inter-onset-variance	For clusters: onset of the first note of the cluster
melody-dur-ratio	Two dimensions computed by the feature described in Figure 4
chord-dur-ratio	

Eventually, the classifier was evaluated against the 3,669 bars recorded on the 7 other tunes. As indicated in Table 3, classification results are near perfect, ensuring robust mode identification during performance.

Table 3. The performance of an SVM classifier trained on the 315 bars of Bluesette and tested against the other 3,669 bars

Classified as \rightarrow	Bass	Melody	Chord	<i>F-measure</i>
Bass	1,314	6	2	0.98
Chord	24	1,294	13	0.97
Melody	1	12	1,318	0.99

In addition, we have developed an audio-based approach that uses the Yin pitch-detection algorithm [11]. Yin is used to extract the pitch and confidence indicator (CI) on a win-

dow of 50ms. On every bar, we extract three features: the average pitch, the average CI, and the variance of the CI on the 50ms windows constituting the bar. On the audio recordings in Table 1, the SVM classifier trained with these three features has an F-measure of 0.94. Although this is not as good a performance as the MIDI-based classifier, it is acceptable in practice.

Interactive Generation

During performance, the system generates audio streams using concatenative synthesis from material previously played. The generation follows two principles:

1) The “other members” principle

The currently played music is analyzed by the mode classifier, which determines the two other modes to generate (bass \Rightarrow chords & solo, chords \Rightarrow bass & solo, solo \Rightarrow bass & chords). In the case that there is no previously played content for the current bar the RLR outputs silence.

2) Feature-based Interaction

A crucial aspect of our system is that it does not simply play back a recorded sequence, but generates a ‘mimetic’ new one, adapted to the performance of the musician. This is achieved using *feature-based similarity*. The system extracts audio features from the user’s input: in practice, the user features are *RMS* (mean energy of the bar), *hit count* (number of peaks in the signal) and *spectral centroid*, though other MPEG7 features could be used (see [8]). The system finds and plays back recorded bars of the right modes (say, chords and bass if the user is playing solo), correct grid chord, that best match the chosen features. Feature matching is performed using Euclidean distance.

Note that the interaction we propose could hardly be obtained with physical controllers such as pedals, as it would require a large number of controls (three modes plus various feature values) to be activated at every beat, thereby eating up too much mental attention for the musician.

Audio generation

Audio generation is performed using *concatenative synthesis* [10]: audio beats are concatenated in the time domain and crossfaded to avoid audio clicks.

ILLUSTRATION

Guitar performance

We illustrate our approach with a solo guitar performance (by the first author) using the RLR on the tune “Solar” by Miles Davis (see accompanying annotated video). During this 2’50” performance, the 12-bar sequence is played 9 times. The musician played alternatively chords, solos, and bass, and the system reacted according to the two “other members” principle. Moreover, the RLR generated an accompaniment that matches the overall energy of the musician: soft passages are accompanied with low-intensity bass lines (i.e., bass lines with few notes as the *hit count* user feature is used), and with low-energy harmonic bars (i.e., with soft chords, as user *feature RMS* is used), and conversely. Figure 5 shows a time-line of one grid (#9) of the performance emphasizing mode generation and interplay, as well as the feature-based interaction used.

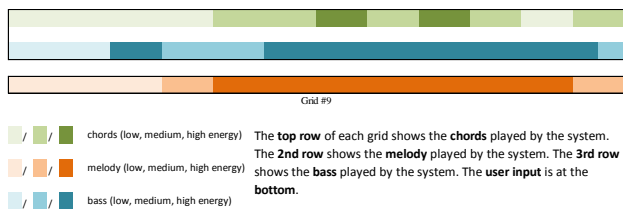


Figure 5. Extract of a performance of Solar (grid #9 in the video). The system does not play any melody (the “other member” principle). The system tries to match the musician’s energy when possible.

CONCLUSION

Our system addresses two critical problems of existing music extension devices: 1) *lack of adaptiveness* (loop pedals are too repetitive), and 2) *stylistic mismatch* (playing along with minus-one recordings generates stylistic inconsistency). Our system is based on a multi-modal analysis of solo performance that classifies every incoming bar automatically into one of a given set of modes (in this paper: bass, chords, melody). The RLR generates an audio accompaniment that 1) best matches what is being performed by the musician, using feature matching and mode identification so enabling adaptiveness and 2) consists only of bars the user has played previously during their performance, ensuring stylistic consistency. As a consequence, a solo musician can perform as a jazz trio, interacting with themselves on any chord grid, providing a strong sense of musical cohesion, and without creating a canned music effect.

The new kind of interaction described here was inspired by observations of, and participation in, real jazz bands. Many other scenarios have been investigated within our framework, including, for instance, an automatic mode in which the musician stops playing and simply controls the generated streams using gestural controllers, so as to let them focus on structure rather than on actual playing. Our system is

entirely automatic. However, we could reintroduce them to bring more control on the generated audio. A *freeze* pedal could allow the musician to play along in a preferred configuration without interfering with it. Another configuration would consist in playing a solo on top of a system generated one. In a final case, we could let the system play the three performance modes, and control each of them with dedicated controllers located on the instrument.

This experiment is part of an on-going study to investigate the premise that exploring our own style can boost creativity. Experiments conducted so far demonstrated that musicians enjoy playing with the system much more than with minus-ones or loop pedals. A series of solo performances with and without RLRs is scheduled to study the actual impact on the musician’s ability to develop new kinds of solo performances as well as on the audience’s response.

ACKNOWLEDGMENTS

This work was partially funded by the ERC *FlowMachines* grant. We thank Ed Jones, Jeff Suzda and Fiammetta Ghedin for their fruitful comments.

REFERENCES

1. Aebersold J., *How To Play Jazz & Improvise*, Book & CD Set, Vol. 1, 2000
2. Boss, RC-3 Loop Station Owner’s Manual, 2011
3. Cherla S. Automatic Phrase Continuation from Guitar and Bass-guitar Melodies, Master thesis, UPF, 2011
4. Hamanaka M., Goto M., Asoh H., and Otsu N., A learning-based jam session system that imitates a player’s personality model. *IJCAI*, pp. 51-58, 2003
5. Jordan S., *Magic Touch*, Blue Note Records 1985
6. Lähdeoja O., An approach to instrument augmentation: the electric guitar, Proc. New Interfaces for Musical Expression conference, NIME 2008
7. Lévy B., Bloch G., and Assayag G., *OMaxist Dialectics: Capturing, Visualizing and Expanding Improvisations*, Proc. NIME 2012, Ann Arbor, 2012
8. Peeters G., A large set of audio features for sound description (similarity and classification) in the CUIDADO project, *Ircam Report* 2000
9. Reboursière L., Frisson C., Lähdeoja O., Anderson J., Iii M., Picard C., and Todoroff T., *Multimodal Guitar: A Toolbox For Augmented Guitar Performances*, Proc. of NIME, 2010
10. Schwarz, D. Current research in Concatenative Sound Synthesis, Proc. Int. Computer Music Conf., 2005
11. Lachambre H., Andre-Obrecht R., and Pinquier J., Distinguishing monophonies from polyphonies using Weibull bivariate distributions, *IEEE Trans. Audio, Speech and Lang. Process.*, 19 (6), pp. 1837-1842, Aug. 2011