

# Intelligent real-time music accompaniment for constraint-free improvisation

Maximos A. Kaliakatsos-Papakostas  
Department of Mathematics  
University of Patras  
GR-26110 Patras, Greece  
Email: maxk@math.upatras.gr

Andreas Floros  
Department of Audio and Visual Arts  
Ionian University  
GR-49100 Corfu, Greece  
Email: floros@ionio.gr

Michael N. Vrahatis  
Department of Mathematics  
University of Patras  
GR-26110 Patras, Greece  
Email: vrahatis@math.upatras.gr

**Abstract**—Computational Intelligence encompasses tools that allow the fast convergence and adaptation to several problems, a fact that makes them eligible for real-time implementations. The paper at hand discusses the utilization of intelligent algorithms (i.e. Differential Evolution and Genetic Algorithms) for the creation of an adaptive system that is able to provide real-time automatic music accompaniment to a human improviser. The main goal of the presented system is to generate accompanying music based on the local human musician’s tonal, rhythmic and intensity playing style, incorporating no prior knowledge about the improvisers intentions. Compared to existing systems previously proposed, this work introduces a constraint-free improvisation environment where the most important musical characteristics are automatically adapted to the human performer’s playing style, without any prior information. This fact allows the improviser to have maximal control over the tonal, rhythmic and intensity improvisation directions.

## I. INTRODUCTION

The potentiality of automatic methods for music composition has been a subject of thorough research study. Various systems have emerged that utilize music theory, probabilities and evolutionary techniques among others, to compose music in an unsupervised or supervised manner. Unsupervised music composition allows the creation of music without any human intervention, except from the representation modeling per se. Supervised music composition on the other hand, modulates the model parameters towards a direction dictated by a fitness function, whether objective (e.g. set of musical features) or subjective (e.g. human evaluations). The utilization of intelligent adaptive techniques has proven to be an effective tool for the creation of consistent music that inherits characteristics specified by a fitness function. The paper at hand presents a system that employs intelligent music composition techniques for the accompaniment of a human improviser. Specifically, the system is able to listen to the improviser and respond in real-time with music accompaniment that fits the improviser’s playing style regarding tone, rhythm and intensity.

Several other systems have been proposed for automatic accompaniment which are reviewed in Section II, the majority of which incorporate prior knowledge about the playing material that the human improviser intends to execute. The presented system is oriented towards a different direction, where the human musician intends to improvise without constraints about

the tonal, rhythmic or intensity variations. Therefore, our system is able to decode the local<sup>1</sup> playing style of the human performer, without using any prior knowledge about the performer’s intentions. Afterwards, it employs intelligent algorithms that adapt to the human’s playing directions and generate melodic responses that are relevant but not identical to the melodies improvised by the human performer. Therefore, the human improviser is able to express hers/his instantaneous improvisational creativity and interact with the intelligent algorithms producing novel music.

The rest of the paper is organized as follows. Section II reviews and categorizes some of the automatic accompaniment systems that have already been presented and also reveals the motivation about the presented work. A thorough description of the architecture of the system and the methodologies that it utilizes is provided in Section III. The presentation of results is realized in Section IV, where the strengths and weaknesses of the proposed system are discussed. Section V provides some concluding remarks and references for future work and enhancements.

## II. AUTOMATIC ACCOMPANIMENT SYSTEMS CATEGORIZATION AND OUR MOTIVATION

Intense research effort on Automatic Accompaniment Systems (AASs) has resulted in the construction of significant systems that provide music accompaniment to a human performer. However, there is a great variability and a clear distinction in their perspective. These systems are suitable for different tasks and they are targeting at different musician groups. In [1] the interested reader may find a finer categorization of these systems according to several attributes regarding the type of their interaction with the musician (audio or MIDI), the representation of their knowledge (music theory or trained) and their timing consideration (continuous time with changing tempo or quantized time in fixed tempo). In this work we present a coarser and less detailed categorization of these systems in two categories that reflect their dependence on a pre-existing score. This distinction along with the potentiality of intelligent music composition algorithms illuminate our aims and motivation.

<sup>1</sup>The term *local* refers to the dimension of time and could be also referred to as *recent and upcoming*.

### A. Score-Constrained AASs

Score-Constrained systems utilize information provided by musical score to make decisions about their output. For example, the well-known Band in a Box software utilizes information provided by a *chord grid* and produces music accompaniments with orchestration, tonal and rhythmic characteristics that are defined by a library of musical styles. Similar educational oriented software have been introduced that are targeted to certain musical instruments (e.g. for guitar [2]). Score information may also be combined with musical input provided by the human musician. A distinct sub-category of score-constrained AASs is that of the *score followers*. These systems require information not only from the score, but also from the musician's location in the score. They aim to align the music performer with an existing score, resulting in a human-computer synchronization on a predefined musical stream given through a score. The early work of Vercoe [3] utilized optical sensors as a means of allocating the musician's position in the score. Recent attempts are focused on achieving optimal score following with spectral analysis of audio instrument signals, e.g. by an oboe [4]. While the aforementioned systems perform in real-time, an off-line score-constrained system has also been proposed for finding jazz-style arrangement of a melody and its chord sequence [5]. A real-time system that provides a jazz style bass line given a chord grid and the improvisation content of a musician has been presented in [6] and a system that performs the same task for guitar accompaniment has been presented in [7].

### B. Score-Free AASs

Score-free systems do not utilize score information and decide upon their reaction to human musical input defined by rule-based or acquired-by-training knowledge. These systems are mainly targeted in real-time performance, with a noticeable non-real-time exception of *MySong* software [8], which deduces music rules by trained Hidden Markov Models to create novel accompaniments of recorded singing voice melodies. Similar models, but focused on real-time interaction with a human musician, are the *Band out of a Box* [9] and the *Continuator* [10] software that aim to learn the personal style of a human improviser and create musical responses. Earlier, a system has been presented in [11] and extended in [12], that attempts to capture the musician's intension and predict a suitable accompanying context. Two systems that are closer to the presented approach where presented by Lewis' *Voyager* [13] and Rowe's *Cypher* [14], [15]. These systems aim to make *locally* consistent musical decision, based on recent playing characteristics of the human improviser, disregarding music-theoretic rules or machine learning.

### C. Our Motivation

Music improvisation is a combination of knowledge and creativity, expressed with the creation of musical passages that are both structured and spontaneous. Human improvisers create an aesthetic combination of these two characteristics, aiming to provide a novel musical dialogue (spontaneity) using

an understandable musical language (structure). Trained AASs are targeted towards capturing the human knowledge on music by applying machine learning techniques, a fact that embodies two hazards. Firstly, in these systems, the efficiency of the training process is case-dependent, i.e. different musical styles are represented by different tonal, rhythmical and expression rules. And secondly, the utilization of a trained system does not incorporate the element of surprise, depriving the human improviser the chance to *interact* with a machine.

The motivation of the work at hand is to experiment with the potentiality of computational intelligence in music improvisation and not in music learning. To this end, we propose the construction of a system that does not aim to mimic human musical behavior, nor to learn music rules. On the contrary, we explore the encapsulation of the minimal musical information required and allow the machine to create music background responses not according to music rules, but with the use of computational intelligence. Several noticeable systems have been created which aim to use computational intelligence as a means to create machine improvisers, but on contrast to the aim of this work, these systems are not suitable for the accompaniment of human improviser. For example, a responsive computer improviser has been presented by Biles [16] and a swarm intelligence improviser which was driven by a singer was proposed in [17]. As described later in detail, the proposed system utilizes Differential Evolution, FL-systems and Genetic Algorithms to produce accompanying music for a human improviser.

## III. THE PROPOSED INTELLIGENT IMPROVISATION ACCOMPANIMENT FRAMEWORK

In the presented work we consider three aspects of musical expression, *tones*, *rhythm* and *intensity*. Three modules have been constructed that implement different algorithms, one for each of the above musical expression aspects. Intelligent algorithms are utilized for the implementation of the tone and the rhythmic module, while a simple yet effective statistical model determines the intensities of the produced notes by all the intelligent instrumentalists. All three modules *listen* to the human performer and make some *locally coherent* decisions about the musical elements that they should incorporate. The musical knowledge that they adhere to is the most elementary and it is locally adjustable, i.e. there is no extensive music theoretic knowledge representation to predict the material of the human improvisation. Furthermore, the constantly incoming human improvisation data are analyzed according to some qualitative information criteria, disregarding their pure musical attributes.

Figure 1 illustrates a block diagram of the overall architecture of the proposed system. All its modules are constituted of two submodules, namely the *listener* and the *generator*, and are all described later in this Section. The system incorporates a steady tempo which is provided by the improviser and gathers information from the human performer in the form of MIDI data. The human improviser is able to choose the

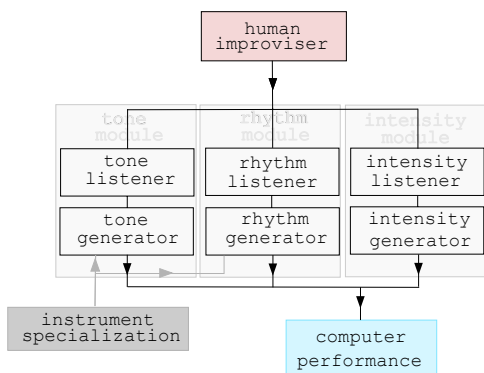


Fig. 1. Block diagram of the architecture of the proposed system, divided in three modules, the tonal, the rhythm and the intensity module. Each module is subdivided in the *listener* and the *generator* submodules.

accompanying instruments that are controlled by the computer, creating an artificial musical ensemble. Any MIDI input methodology is eligible, i.e. MIDI clavien or any efficient pitch to MIDI conversion mechanism. As described later, the tonal module is constructed in such a manner, so that polyphonic data can be manipulated and provide tonal information. The simulations described in Section IV were realized with an electric guitar using the Roland GI-20 pitch to MIDI converter, along with a Godin LGXT guitar with hexaphonic RMC piezo pick-ups. The described algorithms were implemented in Max/MSP, in Java and in Processing, using the Open Sound Control (OSC) protocol for their intercommunication.

#### A. The Tone Module

The tone module receives data from the human improviser and makes a decision about the tones that are suitable to be played as accompaniment by the artificial musicians. The decision about the list of suitable tones is made within the listener submodule, depicted in Figure 2 (a). This decision relies on minimal musical information in order to guarantee the preservation of substantial local tonal characteristics. No assumption is being made about the overall tonal constitution and the improviser’s intentions. Based on the list of suitable tones and the tonal range of each instrument in the intelligent ensemble, the generator submodule (depicted in Figure 2 (b)) provides a set of notes<sup>2</sup> for each instrument as an accompaniment for the human improviser. The decision about the number of voices, for polyphonic instruments, is made in the rhythm module described later, while the decision about the intensity of each note in the intensity module.

1) *The Tone Listener Submodule*: The tone listener submodule deduces a list of suitable tones, denoted as  $t_s$ , based on two auxiliary lists of tones, denoted as  $t_{\text{chord}}$  and  $t_{\text{PCP}}$ . The latter two lists, and consequently the former, are updated in sort time intervals (of 30ms) and track the Pitch Class Profile (PCP) of the human improvisation. The PCP [18] is a 12-dimensional vector that expresses the density of the respective

<sup>2</sup>We use the term “*tones*” to describe the Pitch Class Profile (PCP), while the term “*notes*” refers to specific notes in any octaves available for each instrument.

TABLE I  
THE CHORD CATEGORIES AND THE NON-ZERO VALUES OF THEIR BINARY PROFILES FROM POSITION 0.

chord	non-zero elements
maj	{0, 4, 7}
min	{0, 3, 7}
7	{0, 4, 7, 10}
maj7	{0, 4, 7, 11}
min7	{0, 3, 7, 10}
min7b5	{0, 3, 6, 10}
79	{0, 2, 4, 7, 10}
7b9	{0, 1, 4, 7, 10}
7#9	{0, 3, 4, 7, 10}
76	{0, 4, 7, 9, 10}
7+	{0, 4, 7, 8, 10}
maj69	{0, 2, 4, 7, 9}
min69	{0, 2, 3, 7, 9}
min79	{0, 2, 3, 7, 10}

pitch classes in a musical passage. We refer to the list  $t_{\text{chord}}$  as the “*chord list*” and to the  $t_{\text{PCP}}$  as the “*Pitch Class Profile (PCP) list*”. The chord list ( $t_{\text{chord}}$ ) is formulated when the improviser is emphasizing on the tones of a chord, in either of the two ways:

- 1) the improviser plays simultaneous notes that form a certain chord or
- 2) the accumulated notes within a time window (defined by the improviser) form a certain chord.

We consider that a certain chord is played if there is high correlation (above 0.8) between some predefined chord templates and the PCP in either a simultaneous chord stroke, or in an set of accumulated notes within a time window. The utilization of chord templates is among the most effective methodologies for chord recognition [19]. The chord categories and the non-zero values in their binary representation from position 0 that we use, are demonstrated in Table I.

When a chord is recognized, then the tones that constitute it are loaded on the list of suitable tones,  $t_s$ . This leaves the improviser the option to settle a steady tonal background when she/he wishes to, by emphasizing on the tones of a certain chord. On the other hand, when the improviser aims to create tonal instability, the second auxiliary list,  $t_{\text{PCP}}$  is activated and additional tones (different from the chord tones) are appended in  $t_s$ . This activation occurs automatically and in the rate that the improviser dictates, i.e. the more tonally unstable the improviser’s playing style, the more notes are added from the auxiliary  $t_{\text{PCP}}$  list. The occurrence of tonal instability is reflected by a great value in the Shannon Information Entropy (SIE) [20] measure of the PCP distribution, a feature that has proven to be a very informative about the tonal constitution of a musical piece [21]. Since the distribution of the PCP is a vector in  $\mathbb{R}^{12}$ , if additional different tones are played by the improviser, a higher SIE value is expected for the PCP distribution. Furthermore, there is a higher SIE value limit that a vector in  $\mathbb{R}^{12}$  can reach, given a specified number of non-zero elements. If the improviser at some point highlights the tones of a chord, but plays some additional off-chord tones then this is reflected in the SIE of the  $t_{\text{PCP}}$ .

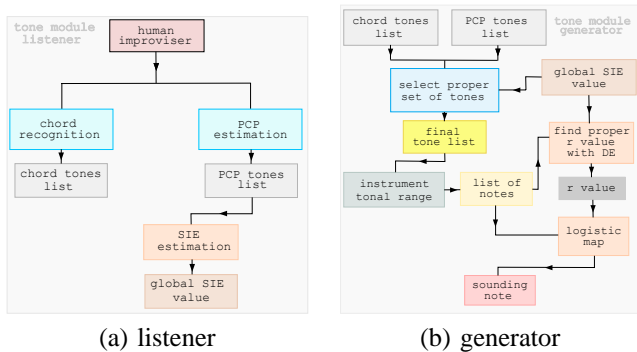


Fig. 2. The *listener* and the *generator* block diagrams of the tone module.

These off-chord tones are appended in  $t_s$ , which is the final tone list from which the tone generator selects tones with a procedure described later. Therefore, the number of tones that are eligible for reproduction by the tone generator module are selected according to the targeted SIE value, the list of notes and the PCP list.

2) *The Tone Generator Module*: The tone generator submodule provides notes within a predefined range (according to the target instrument) and with certain tonality which is set by the  $t_s$  list of the listener submodule. The resulting *note list* is denoted as  $n_s$  and it comprises of integers in ascending order that represent notes within the specified range and with tonality that complies with the  $t_s$  restrictions. The selection of notes from the  $n_s$  list is performed by a well-known iterative scheme called the *logistic map* which presents diverse dynamical behavior that abounds in sets of fixed points and chaotic instability. The logistic map is described by the following equation

$$x_{n+1} = rx_n(1 - x_n),$$

where  $r \in [0, 4]$  is a constant that determines the system's behavior and an initial value  $x_0 \in [0, 1]$  is considered. With this  $r$  and  $x_0$  setup, every iteration is guaranteed to remain within  $[0, 1]$ . Dynamical systems have exhibited interesting music compositional capabilities [22], [23], [24], a fact that motivated their utilization for the proposed tone generator submodule.

The note selection procedure is quite straightforward. The logistic map iterates with a random initial condition,  $x_0$ , and the value of the current  $x_i$  within  $[0, 1]$  is normalized to the range  $[1, N]$ , where  $N$  is the number of elements in  $n_s$ . The integer rounding<sup>3</sup> of the outcome of this normalization is the index of the  $n_s$  that provides the current note. The tonal characteristics of the melody that is produced by this procedure is dependent on the parameter  $r$ , since more complex dynamical behavior yields more complex melodies and vice versa. The melodic characteristics provided by the human improviser are encompassed in the SIE of the improviser's PCP and thus the system's tonal adaptation relies on finding the proper  $r$  value

<sup>3</sup>We consider as rounding of a real number  $x$  the following integer quantity,  $[x] = \lfloor x + 0.5 \rfloor$ .

so that the logistic map produces melodies with similar SIE values. The adaptation of the  $r$  value is performed through the Differential Evolution (DE) algorithm [25], [26]. The DE individuals carry a single value subject to optimization, a value for the  $r$  parameter, and their fitness evaluation is realized by measuring the distance of the target PCP SIE from the PCP SIE of the melodies they produce. For the experiments to be as independent as possible from the  $x_0$  value, we consider the mean distance of the SIE of the PCP of 50 melodies that are composed of 50 notes with random initial  $x_0$  for each individual.

## B. The Rhythm Module

The rhythm module works by giving onset commands (note on or note off events) in certain time instances, within a subdivision of time in equal time segments. The human improviser sets a tempo which is constant throughout the improvisation session. The subdivision of time is dependent on the time analysis of the measure, selected by the user, e.g. an analysis in 16ths in 4/4 measures were selected for the results presented in Section IV. Each measure is represented as a string of digits, with the length of this string depending on the measure time signature and the analysis. In the representation of the measure in the above mentioned example, a 4/4 measure and an analysis of 16ths, 16 digits are required to represent each measure, following the *quasi-binary* [27] representation. The quasi-binary representation incorporates the utilization of numeric digits for representing onset events. For example, the digit 0 may be used to represent a no-onset event and a positive integer value could represent an onset. In Section IV we refer thoroughly to some types of quasi-binary representations that we have experimented. The role of the rhythm module is to "listen" to the improviser's rhythm, and "generate" rhythmic sequences with similar characteristics. These rhythmic sequences are then "performed" by an intelligent instrument according to its quasi-binary representation. The rhythm strings are produced with the Finite L-systems (FL-systems) [27], and their adaptation according to the improviser's playing style is realized with the evolution of the FL-systems with genetic algorithms [28], as studied in [29].

1) *The rhythm listener module*: A block diagram of the rhythm listener submodule is depicted on the upper part of Figure 3. This module detects the location of the improviser's onsets within a sliding window of 4 measures. I.e., in every metronome beat the current onset event is calculated and a digit that describes it (1 if an onset occurs, otherwise 0) is added to the current position of the improviser's rhythm sequence, while at the same time the final digit of this sequence is discarded. At each metronome beat, several descriptive rhythm features are calculated from the improviser's rhythm sequence. These features are fed in the generator module as "target fitness values" and the FL-systems are genetically evolved to produce rhythmic sequences that comply with these features.

The aforementioned features are descriptive about the rhythm, in a sense that they describe general qualitative

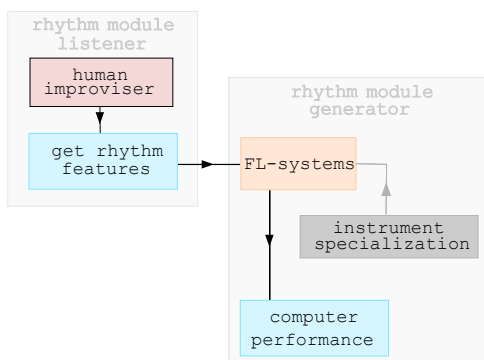


Fig. 3. The rhythm module block diagram with the *listener* and the *generator* submodules. Rhythmic features are collected in real-time by the human improviser, which are used as input to the FL-systems. The rhythmic sequence is created according to the intelligent musician’s instrument modeling.

characteristics of rhythmic sequences, but do not decode them explicitly. These features are the following:

- 1) **Density:** This indicator describes the number of events within a rhythmic sequence and it is expressed through the ratio of onsets events to total events.
- 2) **Syncopation:** The syncopation of a rhythms has been thoroughly studied with complex theoretic models [30] and cognitive subjective studies [31] among others. For the paper at hand we have utilized the rhythm syncopation measure described in [32].
- 3) **Symmetry:** The symmetry of a rhythm can be described as the repetitiveness of the distances of consecutive onset events and it is calculated through the *interval vector* representation [33] by dividing its standard deviation with its mean value.

2) *The rhythm generator module:* The rhythm generator submodule is illustrated at the lower part of Figure 3. This submodule generates rhythmic sequences using the FL-systems, with fitness evaluation value provided by the improviser’s rhythmic features collected in the listener submodule. At this point, it is important to remark that for the task at hand it is crucial to maintain a clearly descriptive set of features for rhythmic sequences. If we had a set features that describe a rhythmic sequence explicitly, then the produced sequence would be a more or less “exact copy” of the improviser’s sequence. This would violate the aim of this system, which is to provide independent yet consistent accompaniment to a human improviser, resulting in human replication. The rhythmic sequences are generated in accordance to the intelligent instrument’s modeling, i.e. the set of digits needed for the quasi-binary representation of rhythm. The results described in Section IV provide some examples of such modelings.

### C. The Intensity Module

The intensity module, on contrast to the tone and the rhythm modules, does not incorporate evolutionary algorithms. It adapts to the improvisers velocities with a simple combination of elementary statistics. The intensity “listener” submodule keeps track of the MIDI velocity mean and standard deviation

values of the active notes played by the improviser within a sliding time window. This window may be arbitrarily long, an issue examined in the results Section. The intensity “generator” submodule sets the current note with an intensity value in uniform distribution with the mean value provided by the listener, and a range given by the standard deviation. A more sophisticated adaptation scheme for intensities could be tested in a future work, but there are no evidences that it would produce more impressive results, since the intensity modeling is straightforward.

## IV. RESULTS

The results include statistics gathered from 3 improvisation sessions which are available online<sup>4</sup>, where one of the authors was playing the guitar and two artificial intelligent musicians were provided accompaniment with decisions made in real time. The improvisation sessions were recorded in such a manner, so that a wide range of the system’s capabilities can be examined, e.g. steep and gradual intensity changes were included, combined with rhythm and tonal diversity. The first two improvisations were modal, and the third was a standard blues form, with jazz playing feel. During the recording of these improvisations, a set of features was also recorded at every 16th of the measure, for every instrument, namely the human guitar player and the intelligent pianist and bassist. These features were the SIE of the PCP, rhythm density, syncopation, symmetry and onset intensity. By these procedure, 5 time series were created for each instrumentalist, one for each feature.

To examine the rhythmic potentialities, we followed two different rhythmic models for the intelligent instrumentalists, formalizing a polyphonic rhythmic decoding for the pianist and a monophonic for the bassist. As mentioned earlier, the FL-systems provide a rhythmic sequence with a quasi-binary representation, i.e. a string of integers that is decoded to rhythm. In the case of the bassist, the string is binary, meaning that we utilized the digit 1 to denote an onset and 0 to denote a no-onset event. The polyphonic model we follow utilized 5 digits, from 0 to 4. At every metronome beat, the rhythm sequence is scanned and a digit with a numeric value greater than or equal to 1 produces an onset event. The numeric value of this digit is translated into the number of notes that the target instrument (the piano in our case) will play simultaneously. The intensities follow the simple statistical model that is described in Section III-C.

Due to limitations in CPU power<sup>5</sup> the training of the tonal and rhythmic generator submodules, with DE and GA respectively, was not possible to be performed in small time intervals. The DE algorithm for the tone submodule was set to run every 5 seconds, and the GA for the FL-systems was set to run every 4 measures for the pianist, with a 3-second

<sup>4</sup><http://sites.google.com/site/maximoskp/Impro1gdom.mp3>,  
<http://sites.google.com/site/maximoskp/Impro2aDorian.mp3>,  
<http://sites.google.com/site/maximoskp/impro3aBlues.mp3>

<sup>5</sup>The system was simulated with a MacBook Pro laptop, using Open Sound Control for the intercommunication of the described submodules.

delay for the bassist. This fact in a manner violates the real-time adaptation of the system, nevertheless an overview of its capabilities and weaknesses can be realized. Moreover, the great CPU consumption by the application of the GA for the FL-systems does not allow us to perform the genetic adaptation with as many individuals as necessary, leading to evolution simulations that included 20 individuals in each generation for a total of 20 generations. The DE evolution was more CPU-friendly, allowing the incorporation of 100 individuals per generation, for 100 generations.

As mentioned earlier, five feature are monitored every 16th beat, creating five time series for each instrument in every improvisation. These time series for all instruments in the first improvisation are illustrated in Figure 4. The respective time series for the rest improvisations had similar characteristics to the presented ones. The human improviser controls the tonal, rhythmic and intensity parameters of the improvisation with hers/his playing style, leaving the system to adapt and respond with a similar music performance style. It is thus crucial to pinpoint the adaptation of the system to the improviser’s style in the tonal, rhythmic and intensity domain. This can be realized by studying the relations between the *leading* feature time series, i.e. the time series of the human improviser’s features, and the time series of the *followers’* features, i.e. the intelligent instrumentalists’ features.

Figure 4 reveals that some time series, for example the syncopation time series, present vague behavior in all improvisation sessions, with dense and sudden changes that do not explicitly follow any trend. This fact is indicative about the sensitivity of these features in small changes. In turn, this sensitivity imposes restrictions on the conclusion to safe results from these time series, since small diversification of the playing style between the human improviser and an intelligent musician could lead to large differences in the time series. To this end, we also consider the smoothed time series that is created by the Moving Average (MA) of each time series. The smoothing was performed within a window of four measures. Table II demonstrates the linear correlation of the human improviser’s features time series and the ones of the intelligent instrumentalists, considering both the initial and their MA time series. The time series’ values were documented in real time and they captured the instantaneous musicians’ (human and artificial) responses within a time window of 3 seconds for the tonal SIE features, 4 measures sliding window for the rhythmic, and a 3-seconds window for the intensity features.

On the column with the “no delay” indication, the time series are taken as they are, without considering the delay between feature capturing and training. The column denoted with the “delay” indication includes the correlation results with proper shifting of the piano and bass time series, so that the beginning of training coincides with the beginning of the collection period for the target features. The tonal SIE time series for both instruments is shifted for a 5-second interval (in accordance the tempo). For the piano rhythmic time series (density, syncopation and symmetry) a shift of 64 values occurred (16 beats for 4 measures) and for the bass an additional

shift of 3 seconds was realized. The intensity module does not incorporate any explicit time delay in training, since the intensity features are updated every 16th beat. Therefore, the consideration of the delay expresses the “*ideal*” adaptation of the system, while ignoring the delay indicates the “*pragmatic*” system’s response.

The correlations demonstrated in Table II reveal that some features in the playing style of the human improviser are reproduced quite accurately by the intelligent musicians. Specifically, the rhythmic density and the intensity follow are in all cases highly correlated. For the intensity feature, the high correlation value is expected because of its simple and straightforward statistical modeling. The tonal SIE and the rhythm syncopation MA features seem moderately correlated, especially with the delay adjustment. The rhythm symmetry feature is exhibited to be the less correlated feature, meaning that in terms of symmetry, the rhythms produced by the human improviser and the intelligent instrumentalists are not similar in terms of symmetry as we have defined it. These results indicate that the system is able at some extent to capture and reproduce the improviser’s instantaneous tonal, rhythmic and intensity playing style.

TABLE II  
CORRELATION OF THE GUITAR FEATURES WITH THE RESPECTIVE PIANO AND BASS FEATURES.

improvisation 1				
	no delay		delay	
	piano	bass	piano	bass
SIE	0.4280	<b>0.5571</b>	0.4240	<b>0.5735</b>
SIE (MA)	<b>0.6516</b>	<b>0.7700</b>	<b>0.6829</b>	<b>0.8320</b>
density	0.4659	<b>0.5321</b>	<b>0.7045</b>	<b>0.8557</b>
density (MA)	<b>0.5416</b>	<b>0.6053</b>	<b>0.7771</b>	<b>0.9064</b>
syncopation	0.1789	0.4802	0.2230	0.3418
syncopation (MA)	0.4542	<b>0.6417</b>	<b>0.6551</b>	<b>0.7188</b>
symmetry	-0.2060	-0.3752	-0.0050	0.4130
symmetry (MA)	0.1425	0.0314	0.4222	0.2274
intensity	<b>0.6696</b>	<b>0.6731</b>	—	—
improvisation 2				
	no delay		delay	
	piano	bass	piano	bass
SIE	0.1949	0.1236	0.2601	0.0551
SIE (MA)	<b>0.6189</b>	0.4954	<b>0.7198</b>	<b>0.5529</b>
density	<b>0.5229</b>	<b>0.6316</b>	<b>0.7907</b>	<b>0.7220</b>
density (MA)	<b>0.7090</b>	<b>0.7133</b>	<b>0.8941</b>	<b>0.8520</b>
syncopation	0.0326	0.2958	0.0977	0.2191
syncopation (MA)	<b>0.5455</b>	<b>0.5357</b>	<b>0.7203</b>	<b>0.6710</b>
symmetry	-0.1455	0.0474	0.2701	0.0616
symmetry (MA)	0.2125	0.3113	0.4305	0.4902
intensity	<b>0.5459</b>	<b>0.5745</b>	—	—
improvisation 3				
	no delay		delay	
	piano	bass	piano	bass
SIE	0.3686	0.2823	0.4133	0.2826
SIE (MA)	<b>0.6455</b>	0.4098	<b>0.7308</b>	0.4166
density	<b>0.6011</b>	<b>0.5285</b>	<b>0.8002</b>	<b>0.7011</b>
density (MA)	<b>0.7139</b>	<b>0.5964</b>	<b>0.8951</b>	<b>0.7687</b>
syncopation	0.0152	0.0764	-0.0298	0.1130
syncopation (MA)	0.4374	0.2382	<b>0.6900</b>	0.3164
symmetry	-0.1904	0.0927	0.1161	0.0235
symmetry (MA)	0.1348	0.2161	0.4947	0.2656
intensity	<b>0.6242</b>	<b>0.6345</b>	—	—

For all features, the MA time series present higher correla-

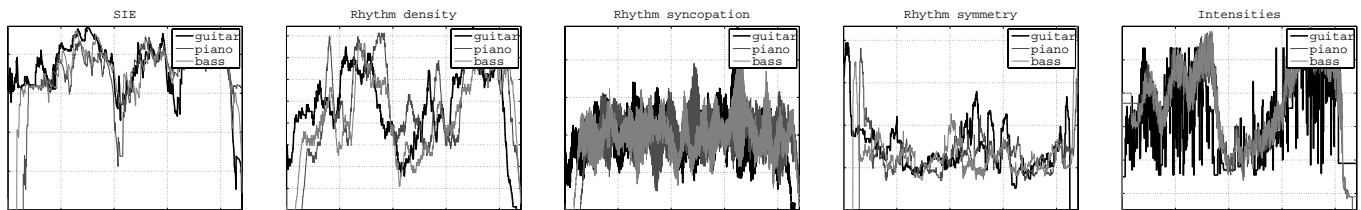


Fig. 4. Fluctuations of all the feature for all instruments in the first improvisation.

tion than the initial time series. This reveals the instability of the initial features, a fact that not only alters the assessment of safe results, but more importantly affects the training process and the adaptation potential of the system, i.e. the training target features are heavily dependent on the exact time that these features are captured. The instability that these features present results from their computation methodology, which is an open subject of computational music research. A possible solution to the feature instability problem would be to use the feature MA in stead of the features per se. Such an approach however, is expected to introduce additional latency to the responses of the system. Moreover, a greater correlation is presented for all the measurements that incorporate the training delay, exposing the “adaptation lag” of the system. As previously mentioned, due to CPU limitations the adaptation lag is high (5 seconds for tonal and approximately 7-10 seconds for rhythmic depending on the tempo). This problem could be overcome with additional code optimization, implementation with a compiled programming language (like C++) and distribution of several tasks to different computer systems.

Another interesting result is derived by the accumulative chroma profile of all the recorded improvisations, which is the summation of all the pitch class activations throughout the piece. This feature has proven to be evident about the composition key of a piece [34], thus it expresses the tonal identity of a music piece. The accumulative chroma profiles of all instrument in each improvisation are similar, a fact that is expressed with their strong correlation which is demonstrated in Table III. In this Table we observe that the accumulative chroma profiles of all instruments in the first two improvisations, which are modal, are correlated with a linear correlation above 0.9 and 0.95 for the first and second improvisation respectively. For the blues improvisation, the correlations drops to around 0.85, which is again strong. These strong correlations indicate that the tone module listens to the improviser accurately and reproduces the tonal environment effectively, so that the overall tonal impression created by the human improviser and the intelligent instrumentalists is consistent.

## V. CONCLUDING REMARKS

We have presented a system that provides intelligent automatic accompaniment to a human improviser without any prior musical considerations. This system adapts to the human improviser’s tonal, rhythmic and intensity performing style

and composes novel music that inherits descriptive qualitative characteristics from the human’s performance. With the proposed methodological approach, the reproduction or alteration of the human’s performance is avoided and the generation of novel music is accomplished with directions given by the human improviser. The tonal characteristics are based on the statistical confidence of the chord that the human performer accentuates, the Pitch Class Profile (PCP) and its Shannon Information Entropy (SIE). The rhythmic characteristics are described by three features (density, syncopation and symmetry) which are captured in real-time during performance. The intensity features are described by simple statistical quantities, the mean and standard deviation, and are captures during performance within a sliding window. These features are then used as fitness values for the generation of notes with the Differential Evolution (DE) algorithm, the generation of rhythmic sequences with Genetic Algorithms (GA) and Finite L-systems (FL-systems) and intensity variations by simple statistical distribution modeling.

Results are presented where 3 improvisation simulations with different play styles are performed with two intelligent accompanying instrumentalists, a polyphonic pianist and a monophonic bassist. These results indicate that the system is able to adapt to the improvisers playing style, producing musical responses that are qualitative similar but not identical. At the same time, the results expose some limitations of the system in its present infant form and provide indications about the improvements that need to be done. The first drawback that emerged is the instability of some features, which present steeply alternating behavior. The second is the introduction of “adaptation lag”, a fact that delays the system’s responses to the music stimuli presented by the human improviser.

Future improvements should primarily address the aforementioned two possible improvements. At first, its should be examined whether the training process could be based on the Moving Average (MA) time series of features. Second, a more efficient implementation of the system should be realized and parallel distribution of several tasks to multiple computer systems should be tested (a framework that is already considered by the existing one, since all the different module intercommunications are carried out through the Open Sound Control (OSC) protocol). Further adjustments could be investigated for the improvement of both the listener and the generator submodules of each module. For the listener submodules, the system could enhanced by the introduction of new features that incorporate further descriptive knowledge. For example, the

TABLE III  
CORRELATION MATRIX OF THE CHROMA PROFILES OF THE GUITAR, PIANO AND BASS.

	improvisation 1			improvisation 2			improvisation 3		
	guitar	piano	bass	guitar	piano	bass	guitar	piano	bass
guitar	1.0000	0.9208	0.9160	1.0000	0.9583	0.9806	1.0000	0.8418	0.8490
piano	0.9208	1.0000	0.9722	0.9583	1.0000	0.9754	0.8418	1.0000	0.9213
bass	0.9160	0.9722	1.0000	0.9806	0.9754	1.0000	0.8490	0.9213	1.0000

tone listener submodule could evaluate the note transition with the help of Markov transition tables and the rhythmic listener could be given the ability to allocate pauses, among many others. Additionally, the generator submodules of all the modules could be further enhanced by utilizing more sophisticated intelligent and adaptive techniques. All these improvements should be examined on more improvisation sessions conducted by musicians with a wide range of expertise levels and playing styles.

### REFERENCES

- [1] D. Martín, "Automatic accompaniment for improvised music," Master's thesis, University of Pompeu Fabra, Barcelona, Spain, 2009.
- [2] G. Cabral, I. Zanforlin, R. Lima, H. Santana, and G. Ramalho, "Playing along with D'Accord Guitar," in *Proceedings of the 8-th Brazilian Symposium on Computer Music*, 2001.
- [3] B. Vercoe, "The Synthetic Performer in the Context of Live Performance," in *Proc. of the International Computer Music Conference (ICMC)*, Paris, 1984, pp. 199–200.
- [4] C. Raphael, "Aligning music audio with symbolic scores using a hybrid graphical model," *Mach. Learn.*, vol. 65, no. 2-3, pp. 389–409, Dec. 2006.
- [5] N. Emura, M. Miura, and M. Yanagida, "A modular system generating jazz-style arrangement for a given set of a melody and its chord name sequence," *Acoustical Science and Technology*, vol. 29, no. 1, pp. 51–57, 2008.
- [6] G. L. Ramalho, P. Rolland, and J. Ganascia, "An artificially intelligent jazz performer," *Journal of New Music Research*, vol. 28, no. 2, pp. 105–129, 1999.
- [7] M. Dahia, H. Santana, E. Trajano, G. Ramalho, and C. Sandroni, "Using patterns to generate rhythmic accompaniment for guitar," in *Proceedings of the Sound and Music Computing (SMC) Conference*, Paris, 2004.
- [8] I. Simon, D. Morris, and S. Basu, "Mysong: automatic accompaniment generation for vocal melodies," in *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 725–734.
- [9] B. Thom, "Bob: an interactive improvisational music companion," in *Proceedings of the fourth international conference on Autonomous agents*, ser. AGENTS '00. New York, NY, USA: ACM, 2000, pp. 309–316.
- [10] F. Pachet, "Music interaction with style," in *SIGGRAPH 2003 Conference Abstracts and Applications*. San Diego, USA: ACM Press, August 2003.
- [11] I. Hidaka, M. Goto, and Y. Muraoka, "An automatic jazz accompaniment system reacting to solo," in *In Proc. of the International Computer Music Conference (ICMC)*, The Banff Center of the Arts, Banff, Canada, September, 3–7 1995, pp. 167–170.
- [12] M. Hamanaka, M. Goto, and O. N., "Learning-Based jam session system for a guitar trio," University Of Tsukuba, Tech. Rep., 2001.
- [13] G. E. Lewis, "Too many notes: Computers, complexity and culture in voyager," *Leonardo Music Journal*, vol. 10, no. 1, pp. 33–39, 2000.
- [14] R. J. Rowe, *Interactive Music Systems: Machine Listening and Composing*. The MIT Press, Oct. 1992.
- [15] —, "Machine composing and listening with cypher," *Computer Music Journal*, vol. 16, no. 1, pp. 43–63, 1992.
- [16] J. A. Biles, "Genjam: evolution of a jazz improviser," in *Creative evolutionary systems*, P. J. Bentley and D. W. Corne, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 165–187.
- [17] T. M. Blackwell and P. Bentley, "Improvised music with swarms," in *Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC '02*, vol. 2. IEEE, 2002, pp. 1462–1467.
- [18] T. Fujishima, "Realtime chord recognition of musical sound : a system using common lisp music," in *Proceedings of the International Computer Music Conference, (ICMC 1999)*, Beijing, China, October 22–27, 1999, pp. 464–467.
- [19] L. Oudre, C. Févotte, and Y. Grenier, "Probabilistic framework for template-based chord recognition," in *Multimedia Signal Processing (MMSp), 2010 IEEE International*, Saint-Malo, France, Oct. 2010, pp. 183–187.
- [20] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, pp. 3–55, January 2001.
- [21] M. A. Kaliakatsos-Papakostas, M. G. Eptropakis, and M. N. Vrahatis, "Feature extraction using pitch class profile information entropy," in *Mathematics and Computation in Music*, ser. Lecture Notes in Artificial Intelligence. Springer Berlin / Heidelberg, 2011, vol. 6726, pp. 354–357.
- [22] M. A. Kaliakatsos-Papakostas, A. Floros, and M. N. Vrahatis, "Music synthesis based on nonlinear dynamics," in *In Proceedings of Bridges 2012, Mathematics, Music, Art, Architecture, Culture*, Baltimore, USA, 25–29 July 2012, p. (To appear).
- [23] A. E. Coca, G. O. Tost, and L. Zhao, "Characterizing chaotic melodies in automatic music composition," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 20, no. 3, 2010.
- [24] J. Pressing, "Nonlinear maps as generators of musical design." *Comp. Music J.*, vol. 12, no. 2, pp. 35–46, 1988.
- [25] R. Storn and K. Price, "Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.
- [26] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.
- [27] M. A. Kaliakatsos-Papakostas, A. Floros, and M. N. Vrahatis, "Intelligent generation of rhythmic sequences using Finite L-systems," in *Proceedings of the Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2012)*, Piraeus, Athens, Greece, 18-20 July 2012, p. (To appear).
- [28] J. H. Holland, *Adaptation in natural and artificial systems*. Cambridge, MA, USA: MIT Press, 1992.
- [29] M. A. Kaliakatsos-Papakostas, A. Floros, M. N. Vrahatis, and N. Kanellopoulos, "Genetic evolution of L and FL-systems for the production of rhythmic sequences," in *In Proceedings of the 2nd Workshop in Evolutionary Music held during the 21st International Conference on Genetic Algorithms and the 17th Annual Genetic Programming Conference (GP) (GECCO 2012)*, Philadelphia, USA, 7-11 July 2012, p. (To appear).
- [30] W. T. Fitch and A. J. Rosenfeld, "Perception and Production of Syncopated Rhythms," *Music Perception*, vol. 25, no. 1, pp. 43–58, Sep. 2007.
- [31] P. E. Keller and E. Schubert, "Cognitive and affective judgements of syncopated musical themes," *Advances in Cognitive Psychology*, vol. 7, pp. 142–156, Dec. 2011.
- [32] G. Sioros and C. Guedes, "Complexity driven recombination of midi loops," in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*. Miami, USA: University of Miami, Oct. 2011, pp. 381–386.
- [33] G. Toussaint, "The geometry of musical rhythm," in *In Proc. Japan Conference on Discrete and Computational Geometry, LNCS 3742*. Springer-Verlag, 2004, pp. 198–212.
- [34] C. L. Krumhansl, *Cognitive Foundations of Musical Pitch*. Oxford University Press, USA, December 1990.