

# On Measuring Syncopation to Drive an Interactive Music System

**George Sioros**

Faculdade de Engenharia  
da Universidade do Porto  
gsioros@gmail.com

**André Holzapfel**

Music Technology Group,  
Universitat Pompeu Fabra  
hannover@csd.uoc.gr

**Carlos Guedes**

Faculdade de Engenharia  
da Universidade do Porto  
/ INESC Porto  
cguedes@fe.up.pt

## ABSTRACT

In this paper we address the problem of measuring syncopation in order to mediate a musically meaningful interaction between a live music performance and an automatically generated rhythm. To this end we present a simple, yet effective interactive music system we developed. We shed some light on the complex nature of syncopation by looking into MIDI data from drum loops and whole songs. We conclude that segregation into individual rhythmic layers is necessary in order to measure the syncopation of a music ensemble. This implies that measuring syncopation on polyphonic audio signals is not yet tractable using the current state-of-the-art in audio analysis.

## 1. INTRODUCTION

Rhythmic syncopation is an essential notion both in analyzing and characterizing music as well as in automatically generating musically interesting rhythmic performances. It is commonly related to rhythmic complexity and tension. Several operational and formal definitions of syncopation have been given (see [1], [2]), such as the one found in the *New Harvard Dictionary of Music* which describes syncopation as a temporary contradiction to the prevailing meter.

Various syncopation metrics have been reported (see [3], [4]), however, a reliable computational model that can measure syncopation directly on an actual music performance does not yet exist. Most metrics use binary representations as input and disregard information contained in the amplitudes of events. However, music performances are usually captured as audio signals or MIDI events, and in both cases the amplitudes of events play an important role in rhythm perception. A new syncopation measure was recently reported by Sioros and Guedes [5] (referred to as SG henceforth) that can be applied to a more detailed representation of rhythms as it also considers the amplitude of events. This kind of representation is closer to an actual music signal; it resembles a monophonic real time stream of MIDI events.

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.

© 2012 International Society for Music Information Retrieval

We aim to develop a system that uses syncopation to mediate the interaction between a musician performing live and an automatic rhythm generator. To this end, we explore the difficulties in measuring syncopation in a live music performance. The current study focuses on measuring syncopation in MIDI streams, from which we draw conclusions on how to measure syncopation in audio signals.

We examined the difficulties in measuring syncopation on rhythmic patterns derived from multichannel, multi-timbre MIDI streams by analyzing two datasets, one comprised of short drum-loops and the second of whole songs in various genres. We used the Longuet-Higgins and Lee's metric [6] (referred to as LHL) as it is well-known and shows good agreement with human judgments ([3], [7]). We conclude that the segregation of the instruments in the performance is needed to obtain meaningful syncopation measurements. A comparison between the SG and the LHL metrics was performed, which shows principal agreement of the two measures and deviations which can be attributed to processing the amplitude information in the SG metric.

Finally, we developed a software system that maps real time syncopation measurements to aspects of a rhythmic performance automatically generated by the kin.rhythmicator software [8]. The measurements are performed on either audio or MIDI inputs, as long as they are the result of a single instrument. The system serves as a tool for exploring, designing and creating interactive music performances.

In Section 2, we describe the two syncopation metrics used in the current study. In Section 3, a small study on syncopation follows, where the two MIDI datasets are examined and a comparison between the two metrics is made. In Section 4, we describe the interactive music system we have developed.

## 2. SYNCOPATION MEASUREMENTS

### 2.1 Binary representations of rhythms.

The computation of syncopation using the LHL algorithm [6] is based on a hierarchical metrical structure constructed by stratifying the given meter into metrical layers. The structure can be represented as a tree diagram with the whole bar at the top and the lower metrical levels under it. The exact form of the tree depends on the time

signature of the meter. For a 4/4 meter, the bar is subdivided first into half notes, then quarter notes, eighth notes etc. until a level is reached which represents the shortest note value to be considered. In this way the meter is subdivided into pulses that belong to different metrical levels, here the first pulse belongs to the whole bar level while the second belongs to the 8th note level. In the LHL measure, each pulse is assigned a metrical weight according to the metrical level it belongs to, starting with 0 for the whole bar level, -1 for the half note, -2 for the quarter note, and for each following level the weights are further lowered by 1. While in many studies, *e.g.*[7], the lowest level chosen is the 8<sup>th</sup> note, for applications involving actual music data at least the 16<sup>th</sup> note level is necessary. In the current study, all syncopation measurements are done on rhythmic patterns in 4/4 stratified to the 16<sup>th</sup> note level, however, in the examples given in this section we only show metrical levels down to the 8<sup>th</sup> note level for visual clarity.

Given a monophonic note sequence, we can compute a syncopation score based on the weights of the metrical positions of the notes and the rests. Note durations are ignored. A score is assigned to each rest that is the difference between the weight for that pulse minus the weight for the preceding note event. Summing all scores yields the syncopation value for the sequence. Some examples are given in Table 1. We placed the syncopation scores assigned to the rest-note combinations at the metrical positions of the rests. The depicted sequences are considered to wrap around as loops. In Example 2 we get a syncopation score of  $(0 - (-3)) + (-2 - (-3)) + (-2 - (-3)) = 3 + 1 + 1 = 5$ .

A closer look at these examples reveals, in certain cases, that the results of the algorithm contradict our musical intuition. Example 1 receives a value of 0 since it contains no rests. Example 3, however, also receives a syncopation of 0, against our experience that it is more syncopated than Example 1. This arises because negative scores compensate positive scores:  $(-3 - 0) + (-1 - (-3)) + (-2 - (-3)) = 0$ . We note that summing only positive scores in Example 3 would yield a positive syncopation value. The negative values computed by the LHL algorithm negatively correlate with what could be referred to as metrical strength: while the sequence of 8<sup>th</sup> notes in Example 1 has a neutral score, Example 4 supports the

beat more strongly, as indicated by the larger negative values. However, since we are mainly interested in reliably detecting the syncopation of a bar, we can sum only the positive scores (last column) in Table 1.


































In the present study, the results of the algorithm are normalized by the maximum possible score—one less than the total number of pulses in a bar—in order for the results to be independent of the number of pulses in the bar and the lowest stratification level chosen. The normalized syncopation will be referred to as NLHL-p.

As the LHL algorithm was proposed for individual monophonic sequences we need a method to compute syncopation when several layers of events take place simultaneously. This is the case of multiple instruments in a music ensemble, where a different rhythmic pattern might be performed on each instrument. The overall syncopation of the performance depends on how syncopated each one of the patterns is. We will explain the applied method by examining the four rhythmic patterns of Table 1 as if they were the four parts of a quartet. To obtain an overall syncopation measure for the polyphonic sequence we combine the maximum values for each metrical position and sum them. This results in a syncopation value of  $7/7=1$  for our example (last row of Table 1). Note that this polyphonic syncopation value can exceed the value of 1, which follows our musical intuition that the syncopation of a combination of instruments can be higher than their individual maximum values. This polyphonic syncopation will be referred to as POLYSYNC in the following sections.

## 2.2 Sequences of amplitude values.

We now provide an overview of the syncopation measure proposed by Sioros & Guedes in [5] (SG). This algorithm can be applied to a more complex representation of rhythmic patterns which, in addition to the metrical positions of the events, also includes their amplitudes. We also discuss the advantages of this kind of representation over a binary one with regard to measuring syncopation.

As in the case of the LHL algorithm described above, the SG syncopation measure is also based on a hierarchical model of musical meter. It compares a sequence of amplitude values to a metrical template similar to the one described in [9]. The algorithm is performed in two phas-

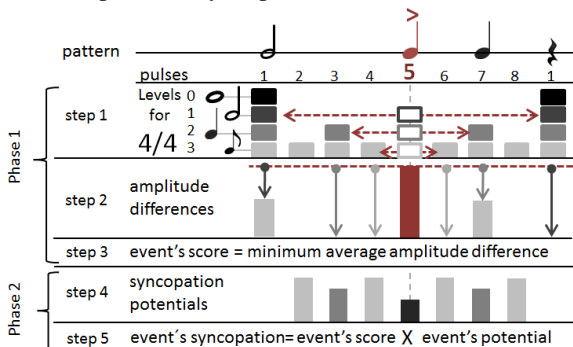
| 4/4 in  | pulses / sequences  |   |   |   |   |   |   |   | weights / scores |      |    |      |    |    |    |    | syncopation |       |
|--|---|---|---|---|---|---|---|---|------------------|------|----|------|----|----|----|----|-------------|-------|
|  | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 0                | -3   | -2 | -3   | -1 | -3 | -2 | -3 | LHL         | LHL-p |
| Example 1  |  |  |  |  |  |  |  |  |                  |      |    |      |    |    |    |    | 0           | 0     |
| Example 2  |  |  |  |  |  |  |  |  | 3                |      | 1  |      |    |    | 1  |    | 5           | 5     |
| Example 3  |  |  |  |  |  |  |  |  |                  | (-3) |    |      | 2  |    | 1  |    | 0           | 3     |
| Example 4  |  |  |  |  |  |  |  |  |                  | (-3) |    | (-1) |    |    |    |    | -4          | 0     |
| <b>combined</b>  |   |   |   |   |   |   |   | 3   |                  | 1    |    | 2    |    | 1  |    | 7  |             |       |

**Table 1:** Computation of the LHL syncopation metric on 4 example sequences. LEFT: the four sequences in binary form. RIGHT, the corresponding weights and scores of the pulses. The negative scores are shown in parentheses as they are ignored in our LHL-p measure.

es. First, it tries to identify loud events that do not occur regularly on the beat in any metrical level. These isolated events contribute to the overall syncopation feel of the pattern. The second phase of the algorithm scales this contribution according to the potential of each metrical position to produce syncopation. The algorithm is performed in five steps (Figure 1). The first phase includes steps 1 to 3 and the second phase, steps 4 and 5. We will demonstrate the algorithm by calculating step by step the syncopation of pulse 5 in Figure 1.

In the first phase, the events that occur regularly on some metrical level are eliminated. Step 1 consists of determining the metrical levels each pulse belongs to, according to the time signature of the meter. In the example of Figure 1, pulse 5 belongs to the half note metrical level (level 2), as well as to all lower ones, i.e. the quarter (3) and eighth (4) note levels. In step 2 the amplitude differences are taken from the neighboring events and are averaged in pairs for each metrical level. The corresponding amplitude differences and averages for pulse 5 would be: i) at the half note metrical level, pulses 5 – 1 in the current bar and 5 – 1 in the next bar; ii) at the quarter note level, pulses 5 – 3 and 5 – 7; and iii) at the eighth note level, pulses 5 – 4 and 5 – 6. In step 3, the lowest value of the calculated averages is taken as the syncopation score of the pulse. If the amplitudes of the events in pulses 1, 5 and 7 of the example are 0.5, 1.0 and 0.5, then the three averages are 0.75, 0.75 and 1. Taking the minimum, the syncopation score of pulse 5 is 0.75.

The second phase of the algorithm (steps 4 and 5) is needed to account for the fact that not all the metrical positions have equal potentials to contradict the prevailing meter: the higher the metrical level the lower its syncopation potential. In step 4 the syncopation potentials are calculated for each pulse as  $1-0.5^m$ , where  $m$  is the highest metrical level the pulse belongs to. In step 5, the syncopation score for each pulse is multiplied by the corresponding syncopation potential. For pulse 5 of the example  $m = 1$  and the final syncopation is  $0.75 \times (1-0.5^1) = 0.375$ . The final result is calculated as the sum of the syncopation of the individual events and is further normalized to the maximum possible syncopation for the same number of



**Figure 1:** Example of the SG algorithm. Calculation of the syncopation of the 2<sup>nd</sup> half note in a 4/4 meter (pulse 5).

events in the bar. This maximum corresponds to a pattern where all events are placed at the lowest metrical level and with amplitude equal to 100%.

The two syncopation measures used in this article have an important difference. The SG algorithm is applied to a more detailed representation of the rhythmic patterns that includes the amplitudes of the events. This makes it possible for the SG algorithm to measure syncopation in drum rolls or arpeggios where events are present in all metrical positions and the syncopation arises from accents in offbeat positions.

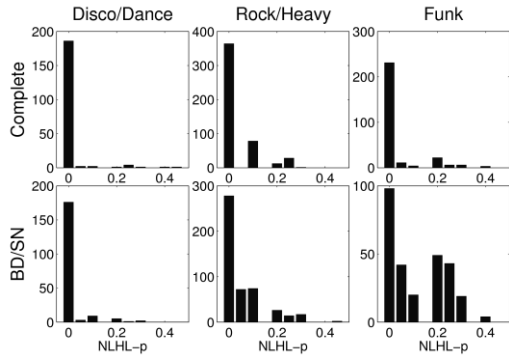
### 3. A SMALL STUDY ON SYNCOPATION

#### 3.1 Methodology

We applied the NLHL-p and the SG algorithm to two different kinds of MIDI datasets. The MIDI data was imported and quantized to the 16th note metrical grid. Syncopation measurements using the SG algorithm were obtained from sequences of amplitudes derived by the MIDI note velocities. When more than one note event was found at the same metrical position, the one with highest velocity was kept. The first dataset, which will be referred to as the Loops-dataset, consists of 602 drum loops from the following genres: Rock, Funk and Disco/Dance. The second dataset, which will be referred to as RWC36, consists of the first 36 songs from the RWC Music Genre<sup>1</sup> dataset that belong to genres of Western popular music. In contrast to the Loops-dataset, the RWC dataset contains whole songs, with each instrument part found in a different MIDI track. All loops and songs examined in here belong to the 4/4 meter, as comparing results between rhythms in different meters is a difficult and unexamined topic which is outside of the scope of this paper.

We used the algorithms to analyze the differences in the musical genres and instruments in terms of syncopation. This analysis reveals which of the examined genres make most use of rhythmic syncopation, as well as how this syncopation is distributed among the various instruments and sections of the songs. It serves as a first towards understanding how syncopation could be measured in audio signals. It must be noted that the results cannot be evaluated against a ground-truth, as there is no syncopation ground-truth available for any music dataset. Instead, we verified that the results are consistent with what is known about and expected for the syncopation in Western popular music. The same qualitative results were observed for both algorithms, so we restrict the representation of the results in sections 3.2 and 3.3 to the NLHL-p algorithm. In section 3.4 we make a more detailed comparison of the two measures.

<sup>1</sup> <http://staff.aist.go.jp/m.goto/RWC-MDB/>



**Figure 2:** Histograms of the number of bars each syncopation score was calculated by the NLHL-p algorithm, for the three most frequent styles in the Loops-dataset. Upper row: for the complete drum-set, lower row: only for the bass-drum and snare-drum events.

### 3.2 Loops-dataset

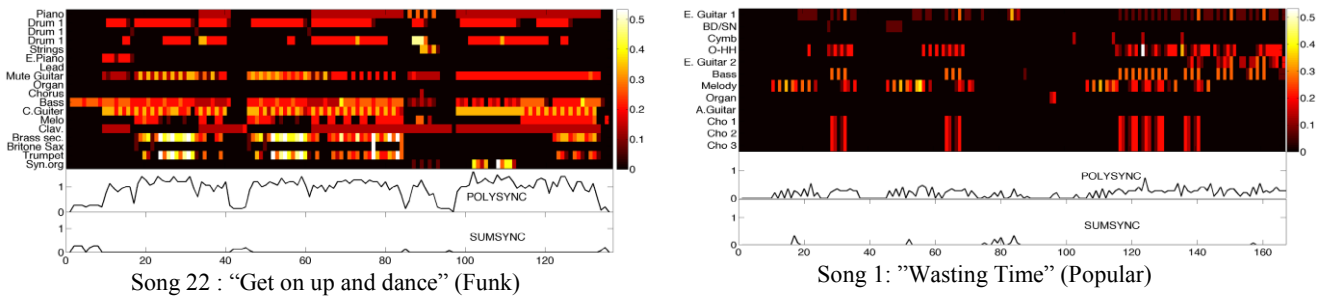
The Loops-dataset contains only short MIDI drum-loops of a few bars that use only the general MIDI drums set sounds. We measured the syncopation in every bar found in the MIDI files of the Loops-dataset, by applying NLHL-p algorithm to each bar separately, as if it constituted an independent loop. We were able to obtain a large number of syncopation measurements for three musical styles: Dance/Disco (198 bars), Funk (286 bars) and Rock/Heavy (484 bars). The histograms of the measured syncopation values are depicted in Figure 2. In the upper parts of the figure, the measurements were performed on the complete group of the general MIDI sounds, in effect ignoring MIDI note numbers. In this case, the Disco/Dance genre appears to be almost totally un-syncopated. While Rock and Funk appear slightly more syncopated, they still seem to contradict our expectations for higher syncopation. If we examine the rhythmic patterns of the bass-drum/snare-drum pair separately, ignoring all other drum sounds, we get more meaningful results as shown in the lower part of Figure 2. These histograms show an increasing percentage of syncopated bars from Disco/Dance to Rock/Heavy to Funk, as expected for these styles. This is a first indication towards the more general conclusion of this study that syncopation needs to be measured in the individual rhythmic patterns that com-

prise a musical performance, implying that at least a basic source/instrument separation is necessary.

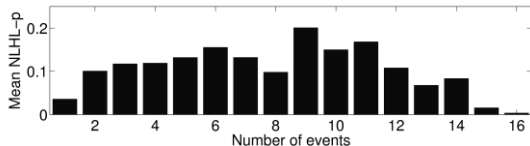
### 3.3 RWC-dataset

In contrast to the Loops-dataset, the RWC-dataset contains complete songs, with several instruments, each in its own MIDI track. We computed the NLHL-p syncopation measure for each track separately and combined the most syncopated events to compute the overall syncopation of the ensemble, using the POLYSYNC method described in Section 2.1. The drum tracks were separated into the following groups: Bass-Drum/Snare, Cymbals, and Openhiat. Such a separation was found to be appropriate from the analysis of the loops dataset. We also applied the same syncopation algorithm to the complete ensemble, considering all note events regardless of the tracks or instruments (SUMSYNC method). The results for two representative songs of the collection are shown Figure 3. The two methods clearly give different syncopation results. They only coincide when a single instrument is syncopating while the rest are silent or when all instrument play in unison. Computing the syncopation on the whole ensemble fails to capture the actual syncopation in the song, and only when we combined the syncopation measurements for each individual instrument the results reflected the actual performance, as can be seen from the POLYSYNC and SUMSYNC curves. Additionally, a much stronger syncopation is encountered in the Funk song, and with a wider distribution among the instruments and among the different sections of the song, as seen in the syncopation matrices of Figure 3.

The above conclusions are not limited to the two depicted examples but are quite common for all 36 songs of the collection. In fact, in less than 2% of the total bars that have syncopated events in some MIDI track, the two methods, the POLYSYNC and SUMSYNC, agree with each other. In contrast, almost 90% of the examined bars show detectable syncopation when using the POLYSYNC method. The syncopation measured in the rhythmic patterns derived from the complete ensemble show little to no syncopation and only when combining information of the various individual instruments can we get a realistic picture which agrees with our experience. This implies



**Figure 3:** Syncopation scores for two songs of the RWC collection. **Top:** individual instruments. **Bottom:** overall syncopation for the whole ensemble (SUMSYNC) and as the combination of the scores of the individual instruments (POLYSYNC).



**Figure 4:** Mean syncopation vs. density of events per bar.

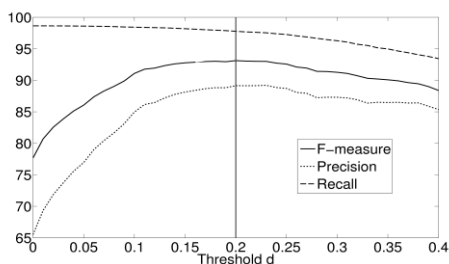
that detection of syncopation in audio signals is only possible after at least some basic instrument segregation.

Figure 4 shows how the measured syncopation is related to the density of events per metrical cycle. As expected for very high density values, the measured syncopation is close to zero, as all metrical positions are occupied by an event. Lower than average syncopation values are also obtained when only one event exists in a whole bar. Interestingly, a low mean NLHL-p value appears for bars with eight events. This is related to the fact that we only analyzed music in 4/4 where the most typical pattern with eight events would be a sequence of 8<sup>th</sup> notes that merely tend to keep the beat and therefore have no syncopation. Again, if we would consider the amplitudes of the events the average syncopation might increase.

Some conclusions about the different music genres in the RWC collection and their use of syncopation can also be made. They cannot, however, be generalized as the number of examined songs was very small. Rap and Funk songs are characterized by the highest syncopation values. In Rap songs syncopation is mainly encountered in the vocals, while in Funk it is always spread among several instruments. Notably, the Modern Jazz pieces were not characterized by high mean values, with the lead instrument in the trios always being more syncopated than the accompaniment.

### 3.4 Comparing NLHL-p and SG measure

We will now compare the two measures by considering the NLHL-p as ground truth, using all separate MIDI tracks of the RWC data. Bars were marked as syncopated when the NLHL-p measure showed syncopation non-zero values. Then we examined how well the SG measure detected those syncopated bars by applying a threshold  $d$  to the SG measurements, above which the bars were consid-



**Figure 5:** F-measure, Precision, and Recall for the detected syncopated bars by the SG algorithm with respect to the NLHL-p.

ered to be syncopated. The comparison was made in terms of F-measure, Precision and Recall (Figure 5).

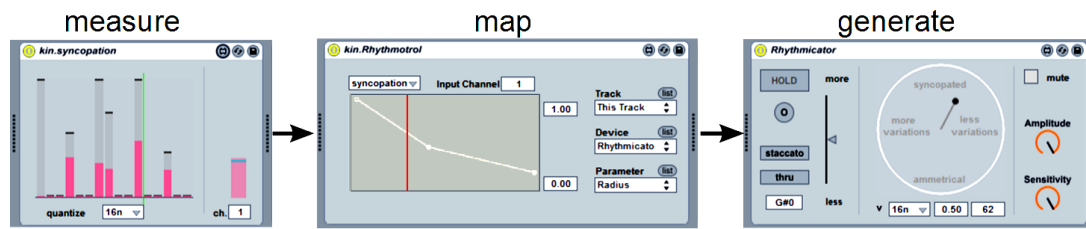
The comparison of the two measures shows a good agreement between them in detecting syncopation. The optimum threshold according to the F-measure is  $d=0.2$  (F-measure=93.11%). The two measures exhibit a different behavior at low threshold values, where the Precision (*i.e.* the ratio between number of correct detections and number of all detections) is lower. This is caused by the fact that the SG algorithm results in an almost continuous syncopation measurement that can distinguish between rhythmic patterns based on small differences in the amplitudes of events. In contrast, the LHL measure gives a syncopation ranking of 16 steps, as it depends only on the existence or not of an event in each of the 16 pulses of a bar.

In principle, it is possible to use both algorithms, the LHL and the SG, for measuring the syncopation in a music performance in real time. As shown here, both result in similar syncopation values for most cases, yet, the SG algorithm seems to be advantageous when syncopation originates from accenting certain notes in a sequence, e.g. in drum rolls. Thus, we chose the SG algorithm to develop our system that generates rhythms based on real-time syncopation measurements of user performances.

## 4. A SYNCOPATION DRIVEN INTERACTIVE MUSIC SYSTEM

We developed an interactive music system based on real-time syncopation measurements. The system comprises four Max4Live devices—Max/MSP based applications that have the form of plugins for the Ableton Live sequencer<sup>1</sup>. Two devices measure the syncopation and density of events in the input music signal, one maps those measurements to any parameter inside the Ableton Live environment and the kin.rhythmicator [8] device generates rhythmic patterns. The input music signal can either be MIDI note events directly grabbed from music instruments and MIDI clips, or it can be simple monophonic audio that is fed to the [bonk~] [10] object for onset detection. Both MIDI and audio signals should be monophonic, *i.e.* the result of a performance on a single instrument. Otherwise, the syncopation measurements will not reflect the syncopation of the input, as shown in the Section 3. The MIDI input or the detected onsets are converted into a sequence of amplitudes, suitable for measuring syncopation with the SG algorithm. The measurements are performed against a metrical template automatically generated according to the time signature of the Ableton Live Set. The implementation of the SG algorithm is similar to the one used in the kin.recombinator application described in [5] with the addition of the normalization described in section 2.2. In addition to the syncopation, the

<sup>1</sup> <http://www.ableton.com>; <http://www.cycling74.com>;



**Figure 6:** The Max4Live devices. The left most device receives the real time input and calculates its syncopation. The middle device receives the syncopation value and maps it to the radius parameter of the right most device, the rhythmicator. Finally, the rhythmicator generates a rhythm with the complexity being controlled by the syncopation measurements.

density of events per bar is also calculated. The measurements are then received by a second device that maps them to any parameter of any other device that the user chooses. The user also controls the exact form of the mapping.

A device like the kin.rhythmicator can be used to automatically generate rhythms. The kin.rhythmicator features a real time control over the complexity of the generated patterns, by controlling the amount of syncopation, variation and the strength of the metrical feel. It was chosen exactly for its explicit control of the syncopation. A possible “chain” of devices is shown in Figure 6. In this way, a user can “prepare” the rhythmicator to interact in real time with a musician, e.g. as the musician performs more complex and syncopated rhythms the automatically generated patterns are more steady and simple, while when the musician tends to perform simpler and less “dense” rhythms, the generated patterns becomes more complex, creating a more syncopated result.

Simple to complex mappings can be realized, involving several parameters in several devices and more than one performer. The described devices are meant as a way of creating direct links between musically meaningful qualities of a performance and an automatically generated output.

The Max4Live devices are available at our website: <http://smc.inescporto.pt/kinetic/>

## 5. CONCLUSIONS

In this paper we presented an interactive music system driven by syncopation measurements. In order to better understand and be able to reliably measure syncopation in an actual music performance, we analyzed two MIDI datasets, one consisting of drum loops, and one of whole songs using the NLHL-p and the SG syncopation metrics. We concluded that in any musical signal, whether it is a MIDI stream or an audio signal, it is important for syncopation measurements that it is first separated into the individual rhythmic layers or the instruments that comprise it. Our findings are of particular importance for our future research that focuses in computing syncopation in more complex music signals in order to drive a meaningful interaction between a musician and a rhythm that is being automatically generated.

## 6. ACKNOWLEDGMENTS

This work was partly supported by the Portuguese Foundation for Science and Technology, within project ref. SFRH / BPD / 51348 / 2011 and PTDC / EAT-MMU / 112255 / 2009.

## 7. REFERENCES

- [1] D. Huron and A. Ommen, “An Empirical Study of Syncopation in American Popular Music, 1890-1939,” *Music Theory Spectrum*, vol. 28, no. 2, pp. 211-231, 2006.
- [2] D. Temperley, “Syncopation in rock: a perceptual perspective,” *Popular Music*, vol. 18, no. 1, pp. 19-40, 1999.
- [3] F. Gómez, E. Thul, and G. Toussaint, “An experimental comparison of formal measures of rhythmic syncopation,” in *Proceedings of the International Computer Music Conference*, 2007, pp. 101-104.
- [4] I. Shmulevich and D.-J. Povel, “Measures of temporal pattern complexity,” *Journal of New Music Research*, vol. 29, no. 1, pp. 61-69, 2000.
- [5] G. Sioros and C. Guedes, “Complexity Driven Recombination of MIDI Loops,” in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, 2011, pp. 381-386.
- [6] H. C. Longuet-Higgins and C. S. Lee, “The rhythmic interpretation of monophonic music,” *Music Perception*, vol. 1, no. 4, pp. 424-441, 1984.
- [7] W. T. Fitch and A. J. Rosenfeld, “Perception and Production of Syncopated Rhythms,” *Music Perception*, vol. 25, no. 1, pp. 43-58, 2007.
- [8] G. Sioros and C. Guedes, “Automatic rhythmic performance in Max/MSP: the kin. rhythmicator,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2011, pp. 88-91.
- [9] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*, Cambridge: The MIT Press, 1996.
- [10] M. S. Puckette, T. Apel, and D. D. Zicarelli, “Real-time audio analysis tools for Pd and MSP analysis,” in *International Computer Music Conference*, 1998, vol. 74, pp. 109-112.