Bassline Pitch Prediction for Real-Time Performance Systems

Andrew Robertson
Centre for Digital Music
School of Electronic Engineering and Computer Science
Queen Mary University of London
a.robertson@qmul.ac.uk

ABSTRACT

This paper presents a methodology for predicting the pitch of bass notes by utilising their metrical position within the bar. Our system assumes two separate audio channels for drums and bass. We make use of onset detection and beat tracking algorithms to determine the onset times for each bass note and the beat locations. A monophonic bass track is analysed for repetitive structures relative to the beat grid, enabling the system to make a prediction of the pitch of each bass note prior to any pitch-related analysis. We present an analysis on a small collection of studio recordings.

1. INTRODUCTION

When humans participate in live music performance, an element of prediction is required. Where there is a musical score, performance requires each player to simultaneously predict the desired playing time for each specified musical event and enact the requisite motor actions ahead of time to schedule each note onset. Where music is improvised, the underlying structure of the music must be anticipated in order to compose phrases that will work in the context of the performance. Whilst signal processing can provide information about auditory events after the fact, Collins [1] regards the ability of human musicians predict and anticipate on a variety of time-scales as crucial to their skill in performing.

Vercoe and Puckette [2] proposed the *listen-perform-learn* model as an integral design in their score following system. Raphael [3] makes use of Bayesian graphical models to learn a performer's timing tendencies from a series of rehearsals. To participate in improvised music to a steady beat, such as blues and rock jams, a player is required to predict key and chord changes ahead of time and yet this is unproblematic for human musicians, partly on the basis that these tend to involve the repetition of a set sequence of chords [4].

Prediction plays an important role in the development of real-time beat trackers which tend to operate on the basis that musical events occur more often on the beat than not. Dixon and Gouyon [5] distinguish between *predictive* and *descriptive* beat tracking behaviour. The former anticipates the beat ahead of time, whereas the latter labels the beat having observed the audio. Real-time beat trackers generally function in a causal predictive manner. Ellis's [6] model uses dynamic programming to calculate the beat times and has been used in a real-time context [7] [8] in which beat predictions are continually updated on the basis of new observed information.

Computational modelling of composition has proved a difficult task [9]. Pearce and Wiggins [10] examine how Markov models or n-grams have been used to model the statistical expectation of melodic parts. Assayag and Dubnov [11] propose the use of the factor oracle, used in string matching tasks, to analyse harmonic and melodic sources within music improvisation. Stark and Plumbley [12] make predictions of the values of future beat-synchronous chroma vectors by matching recent observations in the sequence to similar occurrences in the past. Stowell and Plumbley [13] used a predictive schema for real-time classification of a human percussive vocal in the "beatbox style", whereby a fast reaction is made to classify audio using a provisional classification whilst a more reliable delayed decision is made afterwards. This system produces audio (such as a kick or snare sound) based on the initial classification at low latency after the onset is detected, but makes changes to the audio output where this initial classification appears erroneous.

In this paper, we investigate a related problem of real-time bass pitch prediction based on musical position. For live performance, onset detection methods [14] [15] can reliably indicate the presence of a new note event with a short latency. However, the determination of pitch requires the use of pitch tracking techniques, such as the *yin* algorithm [16], which would impose a larger latency than the threshold for successful networked performance, measured to be approximately 30 msec [17]. A predictive schema might be used to overcome such latency issues.

The predicted output could be used in the synthesis of audio parts or in augmenting the performance with lighting or visuals. Currently, synchronised computer accompaniment can be unreliable, particularly when reliant on beat trackers which are vulnerable to errors such as skipping ahead or behind by a beat [18]. By following the bassline, a beat tracker might be able to recognise when this occurs and autocorrect. The proposed schema of beat-synchronous bassline analysis makes it comparatively sim-

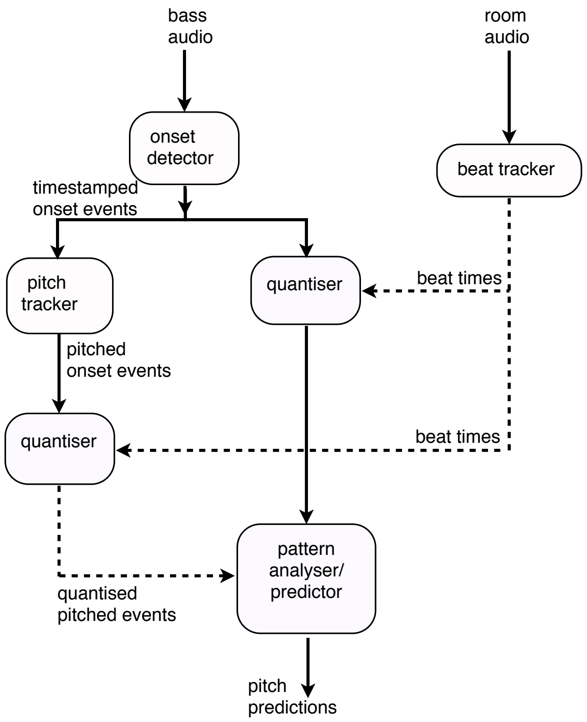

Figure 1. Overview of the algorithm. Pitched events are analysed whilst onset events are labelled according to predictions.

ple to follow the structural sections within a song, thereby bringing about an interactive system that is capable of following an expected performance.

2. METHODOLOGY

Our proposed system uses beat tracking to provide an approximation of the beat grid which can be used to transform event times into a quantised musical time. We assume that there is a dedicated bass stream, typically be through use of a D.I. box or an insert on the mixing desk, and also a general audio stream available suitable for beat tracking, such as from a mono mix of all channels or a room microphone on the drums. A selection of real-time beat tracking algorithms have been developed that might be suitable for such a task, including *B-Keeper* [19], a specialised drum tracking system, *IBT*∼ [20], a multi-agent system based on Dixon's *Beatroot* [21], and *btrack*∼ [7] and *beatseeker*∼ [8] which use autocorrelation. The proposed algorithm detects new bass events using an onset detector and finds the musical position relative to the beat grid. By estimating the optimal lag at which the bassline repeats, we can make a reasonable causal prediction for the pitch using previous pitch tracking estimates. A similar system has been proposed by Stark and Plumbley [12] for the prediction of beat-synchronous chroma sequences. An overview of the algorithm is shown in Figure 1.

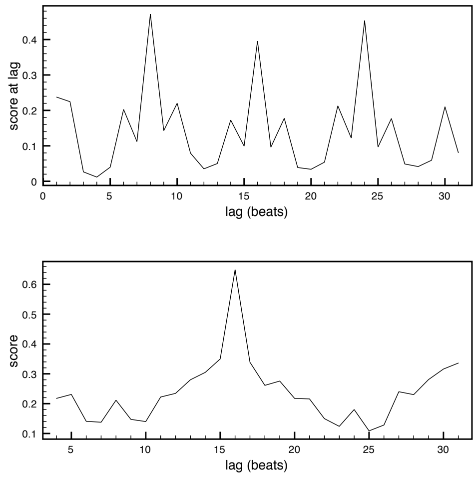

Figure 2. Scores of correlation of quantised pitched events at a range of lags for the song The Radio's Prayer I (top) and Idiots Dance (bottom)

2.1 Event Quantisation

From a standard onset detector, such as those proposed by Bello et al. [15], we obtain event times from a bass signal with low latency, typically under 10 msec. In practice, we compensate for latency by first detecting the onset using peak thresholding on the onset detection function, and then using a second stage that determines the precise point for which the energy change is greatest in the signal. This is done iteratively by dividing the recent audio buffer into segments, choosing the segment which has the highest energy, then repeating the process until the buffer segment contains just one sample and an exact sample position is chosen.

We then wish to specify the musical position of the event relative to the beat grid. Let our sequence of beat times be $\{b_0, b_1, b_2, b_3, ..\}$. Then for an event at time t, we find n such that $b_n <= t < b_{n+1}$. The musical position of the event, $\gamma(t)$, in beats from the start is given by

$$\gamma(t) = n + \frac{t - b_n}{b_{n+1} - b_n}. \quad (1)$$

If we wish to quantise this position to be on an eighth note, triplet, sixteenth or sixteenth note triplet relative to the beat times, we can divide the beat into twelve and instead of the fractional term, find the value m for which $n + \frac{m}{12}$ is closest to $\gamma(t)$ and meets the condition that 3 divides m or 4 divides m. For pitch detection, we use the *yin* algorithm [16] on the segment of audio that immediately follows the detected onset. Our implementation used a real-time version of the *pyin* algorithm by Mauch and Dixon [22] with a framesize of 8192 to provide sufficient resolution in the lower frequencies, although the probabilistic component of their approach was not required. This results in a frequency for each note event, that we then round to the closest MIDI pitch, $p(t)$.

2.2 Repeated Structure Detection

To detect repeated structures, we look for the optimal shift at which the same bass part is played. This is particularly common in rock, blues, pop and dance music, although relatively rare in free improvisation, classical and much jazz music. For a sequence of pitched events at quantised positions relative to the beat grid, we compare it to the same sequence shifted by the integer multiples of beats. At each possible lag, measured in beats, we calculate a score using a method similar to cross-correlation.

Let p_n be the midi pitch of the note at quantised musical position n, obtained from the onset time using Equation 1 and quantising. Our mean correlation value for each lag is given by

$$score(lag) = \frac{1}{N} \sum_{n} \delta_{p_n, p_{n-lag}} \tag{2}$$

where n is the musical position relative to the beats, p_n is the pitch found at beat position n if such an event exists, N is the total number of quantised events found at the given lag that could be compared and $\delta_{p_n, p_{n-lag}}$ is 1 when the pitches match and 0 otherwise. Figure 2 shows the scores across a ranger of lags for two different songs. Where the song's bassline repeats every eight beats (top), we see a series of peaks at multiples of this lower lag, whereas in the bottom plot, the bassline only repeats after sixteen beats. Our optimal lag, used in the causal prediction of bass pitches, is that with the highest score up to a maximum of 32 beats. A weighting system can be used to give preference to a longer or shorter number of beats .

2.3 Pattern Prediction

We made use of two simple methods for predicting the pitch of each bass note. The first, which we shall refer to as the naive method, involves looking back at the observed pitches at successive multiples of our optimal lag and choosing the pitch for the first observed note event found at a corresponding position. For example, if the optimal lag is 8 beats, then we first look if there was a bass note observed exactly 8 beats prior to the one under consideration. Where such an event is found, we predict our current bass note to have the same pitch as this previous observation. Where one is not, we look for a note 16 beats prior, and so on.

However, this does not take into account that there may be changes between song section and that bass line phrases might vary within the same section. To try to exploit our knowledge of transitions made within previous sections, we make an additional constraint that the note found in the same corresponding metrical position has an immediately preceding note that matches the most recently observed note in both pitch and corresponding metrical position. Thus we ensure that the first order Markov transition in pitch for our predicted note from the preceding one is identical to that observed for the note found at a corresponding position that has been used to make this prediction.

3. EVALUATION

To evaluate our two methods, we chose some studio multitrack recordings that allowed the loading of a room audio track and a direct bass track. We made use of the Vamp implementation of an offline beat tracker [1] based on the method of Davies et al. [23] and Ellis [6]. For each note, we counted whether the predicted pitch matched the observed pitch. We have also included the statistic for when the notes matched but differed by an octave. This octave difference is not likely to be due to the pitch detection method (which is reliable for a direct channel of monophonic bass), but rather due to differences in the bass part being played. The evaluation is scored on all bass notes for which the *yin* pitch detection algorithm outputs a result, regardless of metrical position.

Where there is a section change in a song, the naive method naturally results in poor scores, since the bass part no longer matches what was occurring at the previous lag. The results are shown in Table 1. The first order Markov method, which requires that the bass part used for the prediction has a matching previous note, fares a little better in general. On rock material, we tend to observe correct bass predictions approximately 50% to 70% of the time, with the result dependent upon structural changes in the song and the degree of improvisation in the bassline part.

4. DISCUSSION

These methods provide a simple and effective way to predict bass pitches in real-time, particularly when there is a repeating part or section. The main difficulty for these methods is where the piece consists of a sequence of transitions between higher level structures. In this case, we might exploit the failure of the algorithm to correctly predict the pitch for successive note events to recognise that a new section has begun, thereby generating section specific predictions.

Liang et al. [24] propose a methodology for scheduling synchronous events across a network where there is an unknown latency in the communication of messages. Assuming access to a shared clock, such as the atomic clock, each component of the computer performance system sends the system time for the musical event and the tempo as a triple (beat position, system time, tempo), thus enabling an accurate prediction of future beat times even when there is significant latency in the network. By contrast, a beat tracker or score follower that simply sends a message on each beat is rendered highly unreliable in such a scenario. Their method for communicating beat times could be employed here to allow the integration of bassline analysis and prediction within a networked performance.

Our method makes use of a drum mix and a separate bass channel, thereby allowing analysis of bass events at a higher semantic level, with metrical position and associated pitches. A potential application here is the learning of stylistic trends in bass line playing from a corpus of multitrack recordings, thereby making it suited to the computa-

[1] http://vamp-plugins.org/rdf/plugins/qm-vamp-plugins#qm-barbeattracker

		Naive method			First Order Markov		
Song	lag	Correct	Octave	Wrong	Correct	Octave	Wrong
Down The Line I	8	43.4	5.8	50.8	53.9	6.5	39.6
Down The Line II	16	49.2	3.3	47.5	65.0	5.6	29.5
Motorcade	16	45.4	5.6	49.1	61.1	2.8	36.1
The Radio's Prayer I	8	41.5	7.0	51.5	55.6	6.4	38.1
The Radio's Prayer II	16	54.1	4.5	41.5	70.1	4.5	25.5
Diamond White	16	64.1	12.3	23.6	62.6	8.2	29.2
Generic Rock Jam I	16	48.2	17.6	34.3	52.8	9.9	37.3
Generic Rock Jam II	16	54.6	15.9	29.5	46.9	14.4	38.7
Generic Rock Jam III	16	71.9	12.3	15.9	73.2	11.9	14.9
Upside Down Dues	16	69.3	14.6	16.1	74.1	11.7	14.3
Orange Crush	16	74.9	3.3	21.9	74.0	2.1	2.4

Table 1. Percentage results for the note predictions across several examples from a small corpus of live recordings.

tional creativity task of automatic generation of a bassline in a given style that fits a given chord progression.

The code for the evaluation study is available online at the Sound Software website [2] .

5. CONCLUSION

We have presented a method of beat-synchronous analysis of bass parts that enables us to make predictions about the likely pitch based on the note's position within the musical structure. A beat tracking algorithm is used to determine the metrical position of quantised bass onset events which are ascribed a pitch using standard pitch detection methods for monophonic audio. To predict bass pitches in a causal manner we find the optimal lag at which the bass part repeats. Two prediction methods are presented. The first chooses the most recent pitch observed at a corresponding metrical position according to this lag. The second method requires that the note at a corresponding metrical position has a previous note that matches the most recently observed note for metrical position and pitch, thereby preserving the first order Markov transition for the current predicted note event and the previous one.

The output of the prediction algorithm might be used to recognise section changes within a piece. The algorithm could then be improved through the recognition and learning of higher level structure for each song. If a rehearsal version of the song was available, further analyses might improve predictions made within each section. Our prediction is currently based solely on position within the most repetitive musical structure and this might be improved through the use of Markov techniques or string matching techniques to learn likely phrases.

One potential application is the use within synchronisation systems for rock performance. At present, such systems tend to rely solely on beat tracking to adjust their tempo to match that of live musicians. By monitoring the bass parts played, such systems be made more reliable and recover from errors where they get ahead or behind the correct beat. There is significant potential to develop further analysis techniques for multitrack audio channels of group improvisations, studio recordings and band rehearsals which might lead to new creative systems for interaction and composition.

[2] `https://code.soundsoftware.ac.uk/projects/bassline-prediction/repository`

Acknowledgments

Andrew Robertson is supported by a Royal Academy of Engineering and EPSRC Research Fellowship. Thanks to Matthew Davies for providing feedback on the paper.

6. REFERENCES

[1] N. Collins, "Towards autonomous agents for live computer music: Realtime machine listening and interactive music systems." Ph.D. dissertation, Centre for Science and Music, Faculty of Music, University of Cambridge, 2006.

[2] B. Vercoe and M. Puckette, "Synthetic Rehearsal, training the Synthetic Performer," in *Proceedings of the 1985 International Computer Music Conference*, 1985, pp. 275–278.

[3] C. Raphael, "Aligning music audio with symbolic scores using a hybrid graphical model," *Machine Learning*, vol. 21, no. 4, pp. 360–370, 2006.

[4] J. Pressing, *Improvisation: Methods and Models.* Oxford University Press., 1988, pp. 129–178.

[5] S. Dixon, "Match: A music alignment tool chest," in *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR-05)*, 2005, pp. 492–497.

[6] D. P. W. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.

[7] A. M. Stark, M. E. P. Davies, and M. D. Plumbley, "Real-time beat-synchronous analysis of musical audio," in *Proceedings of the 12th International Conference on Digital Audio Effects (DaFx-09), Italy*, 2009, pp. 299–304.

[8] A. Robertson, M. E. P. Davies, and A. M. Stark, "Real-time percussive beat tracking," in *Proceedings of the 53rd AES International Conference on Semantic Audio, London, United Kingdom*, 2014.

[9] D. Conklin, "Music generation from statistical models," in *Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, 2003, pp. 30–35.

[10] M. Pearce and G. Wiggins, "Improved methods for statistical modelling of monophonic music improved methods for statistical modelling of monophonic music," *Journal of New Music Research*, vol. 33, no. 4, pp. 367–385, 2004.

[11] G. Assayag and S. Dubnov, "Using factor oracles for machine improvisation," *Soft Computing*, vol. 8, no. 9, pp. 604–610, 2004.

[12] A. M. Stark and M. D. Plumbley, "Performance following: Real-time prediction of musical sequences without a score." *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 1, pp. 190–199, 2012.

[13] D. Stowell and M. D. Plumbley, "Delayed decision-making in real-time beatbox percussion classification." *Journal of New Music Research*, vol. 39, no. 9, pp. 203–213, 2010.

[14] P. M. Brossier, J. P. Bello, and M. D. Plumbley, "Fast labelling of notes in music signals," in *In Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004), Barcelona, Spain*, 2004, pp. 331–336.

[15] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions on Speech and Audio In Processing*, vol. 13, no. 5, Part 2, pp. 1035–1047, 2005.

[16] A. de Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[17] T. Mäki-Patola and P. Hämäläinen, "Latency tolerance for gesture controlled continuous sound instrument without tactile feedback," in *Proceedings of the 2004 International Computer Music Conference*, 2004, pp. 409–416.

[18] R. B. Dannenberg, "Toward automated holistic beat tracking, music analysis and understanding," in *Proceedings of the International Conference on Music Information Retrieval*, 2005, pp. 366–373.

[19] A. Robertson and M. D. Plumbley, "Synchronizing sequencing software to a live drummer," *Computer Music Journal*, vol. 37, no. 2, pp. 46–60, 2013.

[20] J. Oliveira, F. Gouyon, L. G. Martins, and L. P. Reis, "IBT: A real-time tempo and beat tracking system," in *International Society on Music Information Retrieval*, Utrecht, 2010, pp. 291–296.

[21] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Journal of New Music Research*, vol. 30, no. 1, pp. 39–58, 2001.

[22] M. Mauch and S. Dixon, "pYin: A fundamental frequency estimator using probabilistic threshold distributions," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, 2014, pp. 659–663.

[23] M. E. P. Davies and M. D. Plumbley., "Context-dependent beat tracking of musical audio." *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 1009–1020, 2007.

[24] D. Liang, G. Xia, and R. B. Dannenberg, "A framework for the coordination of media," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2011, pp. 167–172.