TEMPO TRACKING WITH A SINGLE OSCILLATOR

Bryan Pardo
Northwestern University Department of Computer Science
1890 Maple Avenue, Evanston, IL 60201-3150
+1-847–491–3500
pardo@northwestern.edu

ABSTRACT

I describe a simple on-line tempo tracker, based on phase and period locking a single oscillator to performance event timings. The tracker parameters are optimized on a corpus of solo piano performances by twelve musicians. The tracker is then tested on a second corpus of performances, played by the same twelve musicians. The performance of this tracker is compared to previously published results for a tempo tracker based on combining a tempogram and Kalman filter.

1. INTRODUCTION

Tempo tracking real music performances by machine has been the subject of much research. With some exceptions [1], recent systems divide into those using a bank of oscillators (such as the work of Large and Jones [2] and Goto [3]) and those based on probabilistic graphical models (such as the work of Raphael [4] and Cemgil and Kappan [5]).

I am interested in applying tempo tracking to accompaniment of a semi-improvisational style, such as Jazz or Blues. In such music, the exact sequence of notes to be played is unspecified (during an improvised solo, for example). Because of this, one must use approaches that work in real-time, with no score knowledge. Of the tempo-tracking approaches referred to in this section, only those based on oscillators and the tempo tracker described in the 2001 paper by Cemgil et al. [6] apply.

Cemgil, et al. modeled tempo as a stochastic dynamical system, using a Bayesian framework. Here, tempo is a hidden state variable of the system and its value over time is modelled by a Kalman filter. They improve tracking accuracy by using a wavelet-like multi-scale expansion of the performance and backtracking with smoothing. Excellent results are reported on a corpus of MIDI piano performances, and the authors made this corpus available to other researchers to allow comparison between systems on the same corpus.

Dixon [7] compared the Cemgil system to an off-line, two-pass tempo tracking system [8]. His results show the performance of the two-pass system to be statistically indistinguishable from the Cemgil et al. system.

Since Dixon's system was designed for off-line use, this leaves open the question of which approach, oscillator or Kalman filter, is more effective for on-line tempo tracking. As a first step towards answering this question, I built a simple, oscillator-based tempo tracker, and tested its performance on the corpus used by Cemgil et al**.** This paper describe the tracker, the test corpus used, the performance measures and parameter optimizations applied, and compares results to those in Cemgil et al.

2. THE TEMPO TRACKER

The tempo tracker created for this paper is based on a single oscillator, whose period and phase adjust to synchronize with a sequence of events.

The system treats a performance as a time series of weights, W, over T time steps (hereafter referred to as "ticks"). Here, w_t is the weight at tick t. For this paper, weight is defined as the number of note onsets occurring at tick t. If there are no note onsets at t, $w_t = 0$. This approach to time is natural in the world of MIDI, where all events occur on clock ticks. When dealing with audio, one must define a mapping from time-steps into time. In this case, the time step t would typically correspond to window of analysis t.

Before beginning, the initial beat onset and period must be selected. The method typically used by human musicians is to wait for a minimum of two events to occur. The time between the first two events is the initial estimate of the period and the onset time of the second event is taken as the start of a beat. The approach I take is similar.

The first beat, b_0, is defined as the first tick whose weight is non-zero. The second beat, b_1, is the second tick whose weight is non-zero, subject to the constraint that the time between them is at least the minimal allowed period for a beat, $p_{\min}$. The minimal allowed period prevents a sloppily played pair of notes, supposed to occur simultaneously, from setting the initial tempo estimate to a very fast value. For this paper, I fix $p_{\min} = 0.1$ seconds (600 beats per minute).

The initial estimates for the next beat onset, b_2, and beat period, p_1, work on the assumption the next beat period will be identical to the initial beat period. Once initialized, the tracker is updated at every clock tick, using the steps outlined in Equations 1 through 8.

Equation 1 finds the distance between the current tick, t, and the expected onset of the next beat, b_i. The value, d, is measured in units of the current estimated beat period, p_i. If $d > 0$, the current tick is after the expected beat onset. If $d > 0$, it is prior to the expected beat onset.

$$d = \frac{t - b_i}{p_i} \tag{1}$$

$$\text{If } (|d| < \varepsilon) \text{ and } (w_t > 0) \tag{2}$$

$$p_i = p_{avg}(1 + kd) \tag{3}$$

$$b_i = b_{i-1} + p_i \tag{4}$$

$$\text{Else if } (t > b_i) \wedge (w_t == 0) \tag{5}$$

$$i = i + 1 \tag{6}$$

$$p_i = p_{i-1} \tag{7}$$

$$b_i = b_{i-1} + p_i \tag{8}$$

Beat onset and period estimates are only affected by events that fall within a window range $[-\varepsilon, \varepsilon]$ of the expected beat onset. Window width is measured in periods, not ticks. Thus, as the estimated tempo slows, the window widens, since a period becomes longer.

If Equation 2 is true, Equation 3 updates the period estimate. This equation depends on p_{avg}, a weighted average of the last n beat periods (in this paper, n = 20), where the weight of each period is exponentially discounted by the memory parameter, m. Increasing m has the effect of smoothing the response of the tracker by increasing the weight applied to past periods. Equation 9 calculates the weighted average.

$$p_{avg} = \frac{\sum_{j=0}^{n-1} p_{i-j} m^j}{\sum_{j=0}^{n-1} m^j} \tag{9}$$

Once the average period is calculated, Equation 3 updates the current period estimate. Here the correction factor, k, determines how far the estimate is adjusted in response to an event. The larger k is, the farther the period and beat estimates are adjusted. Equation 4 updates the estimate of the next beat onset.

If Equation 5 evaluates to "true," the current tick, t, is past the window around b_i where the beat onset may be affected. In this case, beat and period estimates are updated by Equations 6 through 8. Thus, during a tacit passage, the beat tracker will continue to report beats using the current estimate of period and phase.

3. ERROR MEASUREMENT

When the tracker processes a performance, it produces a sequence of beat onsets, $B = \{b_0, b_1, \ldots, b_{n-1}\}$. Given a known-correct beat sequence for the performance, $C = \{c_0, c_1, \ldots, c_{m-1}\}$, the error in the sequence returned by the tracker can be measured. Define b_j as the element of the estimate sequence closest to the ith beat onset in the correct sequence. Equation 10 defines phase error for correct beat c_i.

$$e_i^{\phi} = \frac{|c_i - b_j|}{|c_{i+1} - c_i|} \tag{10}$$

Figure 1 shows an example calculation of phase error for the second correct beat. Here, the phase error between the second estimated beat and the closest correct beat is 0.3. The phase error for the third correct beat approaches 0, as the nearest estimated beat is quite close to the correct beat.

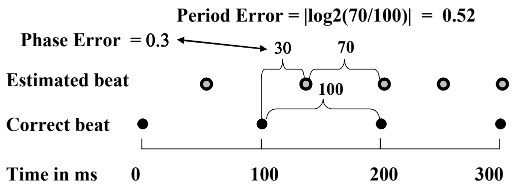

Figure 1. Examples of phase and period error

Equation 11 defines e_i^p, the period error for the ith correct beat. Here, p_j is the period estimate corresponding to estimate beat b_j. Figure 1 shows the period error estimate for the second correct beat.

$$e_i^p = \left| \log_2 \left(\frac{p_j}{c_{i+1} - c_i} \right) \right| \tag{11}$$

When $e_i^p = 0$, the period of the nearest estimated beat is exactly equal to the correct beat. If $e_i^p = 2$, the tracker has found a period either ¼ or 4 times that of the correct beat period. Note that phase error and period error are relatively independent, and it is possible to be in-phase with the correct beat, while having a period of twice the correct value.

$$\rho(B, C) = 100 \frac{\sum_i \max_j W(c_i - b_j)}{(n+m)/2} \tag{12}$$

Cemgil et al. created ρ, an error measure (the Cemgil score) that combines elements of period error and phase error. This is defined in Equation 12. Recall that m is the number of beats in the correct sequence, C, and n is the number of beats in the sequence returned by the tracker, B. Here, $W()$ is a Gaussian window function defined in [6].

It can bee seen that ρ grows from 0 to 100 as phase error and period error decrease to 0, and ρ = 100 only if the correct beat sequence, C, exactly equals the one returned by the tracker, B.

4. TRAINING CORPUS TESTING CORPORA

To compare results between systems, it is important to test both systems on the same testing corpus, and train on the same training corpus. Thus, I test and train on the same corpora used in Cemgil et al.

For the training corpus, a piano arrangement of *Michelle* (by the Beatles) was given to twelve piano players. Four were professional Jazz musicians, five were professional classical musicians, and three were amateurs. Each subject was asked to perform the piece at three tempi: "normal," "slow, but still musical," and "fast." It was left up to the player to determine what "normal," "slow," and "fast" meant. Three repetitions were recorded for each tempo and each performer. One amateur was unable to play Michelle, resulting in a corpus of 99 performances (11 subjects, 3 tempi per subject, 3 performances per tempo).

A testing corpus was created, using the same twelve subjects and protocol, with an arrangement of the Beatles' *Yesterday*. All twelve musicians were able to play the *Yesterday* arrangement, resulting 108 performances.

Performances were recorded as MIDI, using a Yamaha Disklavier C3 Pro grand piano connected to a Macintosh G3 computer running Opcode Vision DSP. Once recorded, the score time for each performance note was written into the file as a MIDI text message, to be used as an answer key.

Most performances in both the Michelle corpus and the Yesterday corpus vary in a 20 beat-per-minute range. The median tempo of the Michelle corpus is roughly 60 beats per minute, while that of the Yesterday corpus is 90 beats per minute. Space requirements preclude a more detailed discussion of the corpora. For more detail, please consult http://www.nici.kun.nl/mmm/, where both corpora are available for download.

5. OPTIMIZING PARAMETERS

The memory parameter, *m* and the correction rate parameter *k* adjust the responsiveness of the simple oscillator tracker to tempo variation. The window size parameter, ε, determines how far out of phase a stimulus may be and still affect the tracker's beat and period estimate.

To explore the space of possible parameter settings, I randomly selected 5,000 combinations of *k, m,* and ε, choosing values for all parameters from a uniform distribution over the interval (0,1). For each combination of parameter settings, I ran the tracker on all 99 performances in the Michelle training corpus, recording the mean values for phase error, e^{ϕ}.

The best set of parameters found for the Michelle corpus was $m = 0.65$, $\varepsilon = 0.36$, and $k = 0.43$, returning a mean $e^{\phi} = 0.0244$, or 0.024 seconds per beat, given an average tempo of 60 beats per minute.

6. TESTING

Using the parameters that minimize phase error on the Michelle corpus, I ran the tracker on all performances of Yesterday in the test corpus. Figure 2 shows a histogram of tracker performance on the Yesterday corpus for phase error, period error, and Cemgil score. The vertical dimension indicates the number of performances falling into a given bin. The horizontal dimension indicates the value of an error measure.

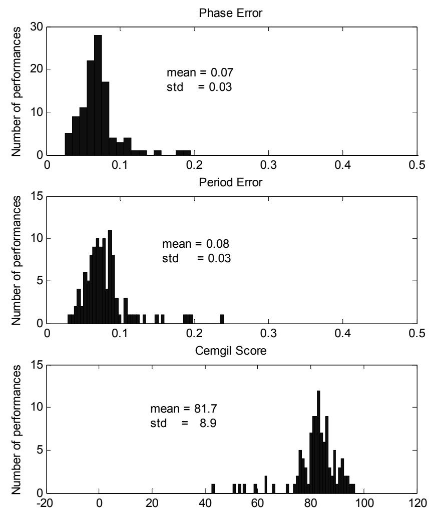

Figure 2. Phase error (e^{ϕ}), period error (e^{p}), and Cemgil score (ρ) on the Yesterday corpus

While the scores for period error would seem to indicate that the tracker correctly locked on to the quarter note level, the tracker actually locked on to the eighth note in every case. Since my scores are intended to be comparable to those published in Cemgil et al. [6], I have treated the eighth note as the beat level, as their scores were also adjusted to account for tracking at the eighth note level, rather than the quarter note level.

The phase error values in Figure **2** indicate the tracker was, on average, out-of phase by 7% of the value of an eighth-note on a typical performance. The mean tempo in the Yesterday corpus is 90 beats per minute, for an average beat onset error of 0.023 seconds per beat.

It is instructive to look at performances with high and low error scores. Figure 3 shows three individual tempo tracks from the Yesterday corpus. In this figure, the vertical dimension indicates the tempo in beats per minute. The horizontal dimension indicates the beat, from the beginning of the performance to the end. A solid line with black points shows the tempo from the answer key. The dashed line with hollow circles indicates the output of the tracker.

The upper panel of Figure 3 shows the performance on which the tracker performed the worst. In this case, Classical pianist 2 suddenly increased the tempo from 54 to 65 beats per minute, causing the tracker to lose the beat. The tracker never recovered. Appropriately, this performance had poor values on all error measures.

The middle panel shows a typical tempo track on a performance in the Yesterday corpus. This performance is also by Classical musician 2, and was selected for display because it gives a good idea of the typical performance of the tempo tracker. In this performance, the tracker looses the tempo in the third measure of Yesterday, then recovers by the fourth measure and continues to track the beat with success for the remainder of the performance.

The lowest panel shows an above-average tempo track, as indicated by all performance measures. Here, the system tracked with success, from beginning to end.

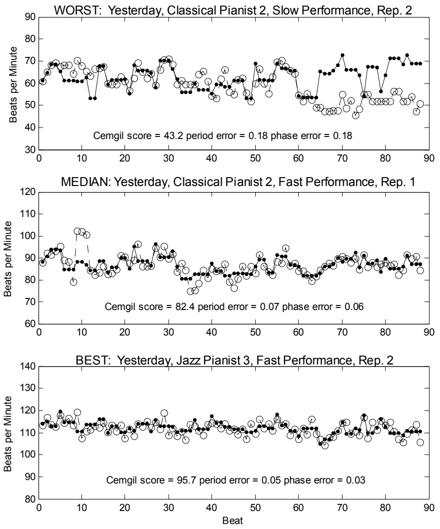

Figure 3. Performance on individual files

Table 1 compares the Cemgil scores for the tracker described in this paper to those of the tempogram tracker, and those of a tempogram plus a 10^{th} order Kalman filter, described in Cemgil et al [6]. All systems were trained on the Michelle corpus and tested on the Yesterday corpus. Values are rounded to the nearest whole number. Values in parentheses are standard deviations. The top row of this table shows mean values for all performances in the Yesterday corpus. The next three rows show mean values for all Amateur, Jazz, and Classical performers, respectively. The final three rows show values for the corpus, broken down by tempo.

Group	Single Oscillator	Tempogram	Tempogram + Kalman
All Perfs.	82 (8)	74 (12)	86 (9)
Amateur	82 (7)	74 (7)	88 (5)
Classical	76 (13)	66 (14)	82 (11)
Jazz	88 (5)	81 (7)	92 (4)
Fast	84 (12)	79 (9)	90 (6)
Normal	83 (5)	74 (9)	88 (6)
Slow	77 (12)	68 (9)	84 (10)

Table 1. Cemgil scores on the Yesterday corpus

Table 1 shows that the single oscillator tracker performs better than the tempogram, and somewhat worse than the tempogram plus 10^{th} order Kalman filter. The mean scores for the single oscillator fall within a single standard deviation of the scores returned by the tempogram plus 10^{th} order Kalman filter. This indicates that the performances of the two systems are, statistically speaking, very close.

7. CONCLUSIONS

I described a tempo tracker, based on a single oscillator, that is simple to implement for real-time tempo following, requires no score knowledge, and shows performance comparable to a more complex tracker based on a tempogram plus 10^{th} order Kalman filter.

While the results of this experiment do not resolve whether an oscillator approach is preferable to a Kalman filter, the average phase error for the simple oscillator tracker on a corpus of 108 performances of Yesterday played by 12 musicians was 23 milliseconds per beat. Average error this small indicates the tracker is good enough for many tasks requiring tempo estimation.

8. ACKNOWLEDGMENTS

Thanks to Richard Ashley, of Northwestern University, for collecting the training and testing corpora.

9. REFERENCES

[1] Rosenthal, D.F., Machine Rhythm: Computer Emulation of Human Rhythm Perception, in Media Arts and Science. 1992, MIT: Cambridge, MA. p. 139.

[2] Large, E.W. and M.R. Jones, The Dynamics of Attending: How People Track Time-Varying Events. Psychological Review, 1999. 106(1): p. 119-159.

[3] Goto, M. and Y. Muraoka, Real-time beat tracking for drumless audio signals: Chord change detection for musical decisions. Speech Communication, 1999. 27: p. 311-335.

[4] Raphael, C., A Hybrid Graphical Model for Rhythmic Parsing. Artificial Intelligence, 2002. 132(1-2): p. 217-238.

[5] Cemgil, A.T. and B. Kappan, Monte Carlo Methods for Tempo Tracking and Rhythm Quantization. Journal of Artificial Intelligence Research, 2003. 18: p. 45-81.

[6] Cemgil, A.T., et al., On tempo tracking: Tempogram representation and Kalman filtering. Journal of New Music Research, 2001. 28(4): p. 259-273.

[7] Dixon, S. An empirical comparison of tempo trackers. In 8th Brazilian Symposium on Computer Music. 2001. Fortaleza, Brazil.

[8] Dixon, S. and E. Cambouropoulos. Beat tracking with musical knowledge. in The 14th European Conference on Artificial Intelligence. 2000. Amsterdam, Netherlands.