MUSICAL TEMPO CURVES

Axel Berndt

Department of Simulation and Graphics
Otto-von-Guericke University, Magdeburg, Germany
aberndt@isg.cs.uni-magdeburg.de

ABSTRACT

Timing models are essential to a variety of music related applications. In the field of performance research they facilitate the analysis of music interpretations. For performance synthesis they are essential for creating expressive performances. This paper details our approach to modelling musical tempo. An important issue is the description and implementation of continuous tempo transitions. Therefore, we introduce two different approaches, *Explicit* and *Implicit Tempo Curves*, and an intuitive parameterization to control their characteristics. These are applied in a listener study to evaluate the human ability to distinguish different tempo transitions. The result provides clues for optimizations and simplifications.

1. INTRODUCTION: MUSICAL TIMING

Timing is one of the most prominent aspects of expressive music performance. Finding adequate timing models is central to performance analysis [4, 5, 7] and synthesis [1, 8, 9]. Timing defines the mapping of symbolic time (score position, MIDI ticks) onto physical time (milliseconds). Timing, however, is a complex aspect that combines several layers of macro and micro features. In our framework we distinguish tempo (macro timing), rubato (self-compensating micro deviations), asynchrony, and random imprecision (deviations that cannot be traced to systematic origins, yet) [3].

This paper details the tempo aspect. After a short introduction to the formalisms and timing conversion basics (section 2) we describe two approaches to model continuous tempo transitions, *Explicit Tempo Curves* (section 3) and *Implicit Tempo Curves* (section 4). Both have their advantages and drawbacks, as discussed in section 5. We complement this discussion by a listener study that systematically reveals the ability of human listeners to distinguish different tempo curves. This provides useful clues for optimization. Finally, section 6 gives a conclusion to this paper.

2. TEMPO

The tempo defines the basic meter, typically in the format "number of symbolic time units per physical time unit" (e.g., beats per minute, bpm). Over the course of a musical piece the tempo must not be constant. This would, in fact, create a quite mechanical and unmusical impression. Instead, human musicians apply discrete and continuous tempo variations to emphasize dramaturgical points of culmination, melodic bows, and musical form. The end of a musical section, for instance, is often marked with a local ritardando.

All tempo features can be formally represented by tempo instructions T that are organized in a sequential list, the so-called *tempomap* M_T. This formalism should be well known from modern sequencer software.

$$M_T = (T_0, T_1, ..., T_n)$$

The tempomap can also be seen as a sequence of curve segments that map symbolic time positions d to tempo values $Tempo(d)$. The MIDI standard implements only discrete tempo changes. Lots of small tempo steps are used to create seemingly continuous transitions. We want to extend this here. A tempo instruction is defined by the following quintuple.

$$T_m = (d_m,\ t_{1,m},\ t_{2,m},\ b_m,\ i_m) \quad \text{for} \quad m \in [0, n]$$

The MIDI tick date of the instruction is indicated by d_m. The instruction defines a continuous tempo transition from d_m to d_{m+1} (the date of the succeeding instruction or the end of the piece) that begins with tempo $t_{1,m}$ and ends up with $t_{2,m}$ (both in bpm). The length of one beat is given by b_m in the following format: quarter note $\rightarrow 0.25$, half note $\rightarrow 0.5$, and so on. The factor *PPQ* (pulses per quarter or ticks per quarter), that can be found in the header of each standard MIDI file, allows to convert this into MIDI ticks.

Continuous tempo transitions are rarely of linear shape but feature monotonous curves with varying shape. To capture this, we introduced the fifth attribute to the tempo instruction, $i_m \in (0, 1)$. In the case of a constant tempo ($t_{1,m} = t_{2,m}$) it can be ignored. Otherwise it indicates a relative position between d_m and d_{m+1}. At this position half of the tempo transition has to be processed.

$$Tempo((d_{m+1} - d_m)\, i_m + d_m) = 0.5\,(t_{1,m} + t_{2,m})$$

We call this the *mean tempo condition*. It is a metaphor that performers are used to working with when they say, for instance: "You have to be earlier faster." By $i_m = 0.5$ the most relaxed, in fact linear, characteristic is created. The more i_m differs from 0.5, the more the tempo transition comes early ($0 < i_m < 0.5$) or late ($0.5 < i_m < 1$).

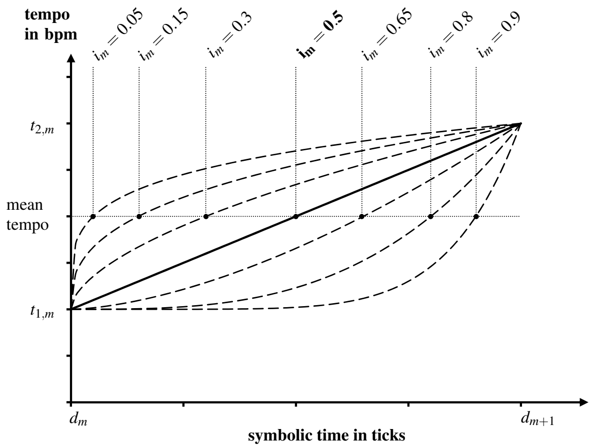

Figure 1. Explicit tempo curves by potential functions.

To convert a symbolic time position d (in ticks) into a milliseconds date $Ms(d)$ the integral of the inverse tempo function has to be computed. In the following notation of Ms this is done segment-wise, where one segment corresponds to one tempo instruction.

$$Ms(d) = \begin{cases} d + Ms(0) & : \quad d \leq d_0 \\ \mathrm{const}(d) + Ms(d_m) & : \quad t_{1,m} = t_{2,m} \\ \mathrm{tran}(d) + Ms(d_m) & : \quad \text{otherwise} \end{cases}$$

$Ms(0)$ is the time when the playback starts, d_0 the date of the first instruction, and m the index of the last instruction before d. For segments with constant tempo the conversion is done by function const.

$$\mathrm{const}(d) = \frac{60\,000\,(d - d_m)}{t_{1,m} \cdot 4 \cdot b_m \cdot PPQ}$$

Continuous tempo transitions are implemented by function tran. We will now describe two possible implementations of this function, *Explicit* and *Implicit Tempo Curves*.

3. EXPLICIT TEMPO CURVES

This approach constructs the tempo curve directly and performs the time conversion by numerical integration. For the tempo curve the potential function in the unity square is used and scaled to the actual measurements of the instruction (see figure 1).

$$Tempo(d) = \left(\frac{d - d_m}{d_{m+1} - d_m}\right)^{\mathrm{p}(i_m)} (t_{2,m} - t_{1,m}) + t_{1,m}$$

The exponent $\mathrm{p}(i_m)$ follows from the *mean tempo condition*.

$$\mathrm{p}(i_m) = \log_{i_m} 0.5 = \ln 0.5 / \ln i_m$$

The milliseconds date $\mathrm{tran}(d)$ is approximated by $\widetilde{\mathrm{tran}}(d)$ via numerical integration of the inverse tempo function (Simpson's rule with $N/2$ iterations and even N).

$$\widetilde{\mathrm{tran}}(d) = \frac{d - d_m}{3N} \left(\frac{1}{Tempo(x_0)} + \sum_{k=1}^{\frac{N}{2}-1} \frac{2}{Tempo(x_{2k})} + \sum_{k=1}^{N/2} \frac{4}{Tempo(x_{2k-1})} + \frac{1}{Tempo(x_N)} \right) \frac{60\,000}{4 \cdot b_m \cdot PPQ}$$

$$\text{with} \quad x_k = d_m + k \cdot \frac{d - d_m}{N}$$

The approximation accuracy can be controlled by N. Following the estimation of [6] the approximation error is given by

$$\left|\mathrm{tran}(d) - \widetilde{\mathrm{tran}}(d)\right| \leq \frac{(d - d_m)^5}{180 N^4} \max_{x \in [d_m, d]} \left| \left(\frac{1}{Tempo(x)} \right)^{(4)} \right|$$

But how much accuracy is necessary? The application is, of course, free to chose any setting. But the computation time increases together with N. Hence, too high a setting should be avoided in the context of realtime applications, like interactive and adaptive music [2]. The following question may be even more important: To what extent can the human listener distinguish different tempo curves? The study in section 5 traces this question and will give clues to find appropriate settings for N.

4. IMPLICIT TEMPO CURVES

In the previous approach, tempo curves were modelled explicitly and the time conversion was implemented by numerical integration. The *Implicit Tempo Curves* approach, goes the opposite way. The curve segment of Ms is directly constructed. This bypasses the (possibly expensive) time conversion and shifts the effort to the preprocessing. However, the challenge is to construct the timing curve so that the corresponding tempo curve features the desired characteristics. This means in particular that it has to hold to the *mean tempo condition* (see section 2).

We apply a quadratic Bézier curve to approximate the timing curve. It is spanned by the three control points P_0, P_1, and P_2. The coordinates of the first of them are given; they arise from the previous curve segment.

$$P_0 = (d_m,\, Ms(d_m))$$

The third control point can be computed using the *Explicit Tempo Curves* approach. As this runs in the preprocessing phase this can be computed with high fidelity (great N). In the context of performance analyses this can also be measurement data that shall be approximated.

$$P_2 = (d_{m+1},\, Ms(d_{m+1}))$$

Through $t_{1,m}$ and $t_{2,m}$ also the timing gradients, $g_{1,m}$ and $g_{2,m}$, at both points are given.

$$g_{z,m} = \frac{60\,000}{t_{z,m} \cdot 4 \cdot b_m \cdot PPQ} \quad \text{for} \quad z \in \{1, 2\}$$

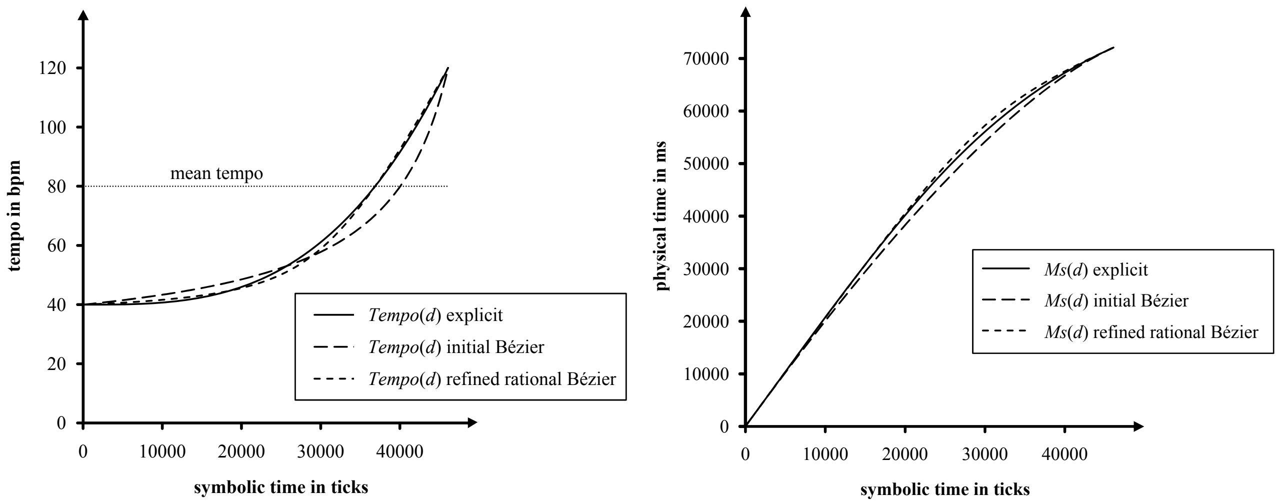

Figure 2. The explicit and implicit models' tempo and timing curve in comparison.

Thus follows the position of P_1.

$$P_1 = (d_{P_1},\ g_{1,m}(d_{P_1} - d_m) + Ms(d_m))$$

$$d_{P_1} = \frac{g_{1,m}d_m - g_{2,m}d_{m+1} - Ms(d_m) + Ms(d_{m+1})}{g_{1,m} - g_{2,m}}$$

When rendering these Bézier curves the corresponding tempo curves show similar characteristics to those of the explicit model but they are not identical (see figure 2)! They do not have to be exactly the same, either. The dispersion that we could observe in human performances [3] is too great to favor either of these curves. But the implicit model does not even hold to the *mean tempo condition*. The Bézier curve as it is constructed up to now is just an initialization and needs further adaptation. This can be done in two possible ways, as will be described in the following sections.

4.1. Shift the Terminal Point

P_1 derives directly from the other two control points. P_0 is fixed. But P_2, the end of the curve, can be shifted along the milliseconds axis. This will also change the position of P_1 and the course of the tempo transition. A simple approach to find an approximate solution is by bisection that checks for the *mean tempo condition*. The boundaries of the search space are:

$$Ms(d_{m+1}) \in \\ ((d_{m+1} - d_m)g_{1,m} + Ms(d_m),\ (d_{m+1} - d_m)g_{2,m} + Ms(d_m))$$

It is important to be aware of the fact that the timing of this approach differs from that of the explicit model (and of the initial Bézier curve) not just locally within $(d_m, d_{m+1}]$ but also beyond d_{m+1} as deviations are not compensated here. If it is desirable, for some reason, to keep synchrony with the explicit model or with some other timing data (like empirical measurements that shall be approximated), P_2 has to remain fixed at its initial position. Therefore, the following approach suits better.

4.2. Rational Bézier Curves

A rational Bézier curve applies weights w_0, w_1, w_2 to its control points. The rational quadratic Bézier curve is defined as follows.

$$P(t) = \frac{w_0(1-t)^2 P_0 + w_1 2t(1-t)P_1 + w_2 t^2 P_2}{w_0(1-t)^2 + w_1 2t(1-t) + w_2 t^2}$$

These weights can be used to distort the curve in a desired way and also to fulfill the *mean tempo condition*. In fact, it is only necessary to adapt the weight w_1 of point P_1 therefore (as done in figure 2). This still does not create a shape that is identical to the explicit model. If this or any other characteristic shall be approximated, both further weights, w_0 and w_2, can be edited, too. The effort, however, can be considerable and is probably not necessary in most application contexts, as the following discussion shows.

5. DISCUSSION AND STUDY

Timing conversions can be very expensive, especially for continuous tempo transitions. This may not be a problem as far as they can be done offline (before the playback). But if they have to be performed in realtime directly at playback, runtime issues become important. With two different approaches, *Explicit* and *Implicit Tempo Curves*, we try to tackle this issue. The explicit approach allows one to select a good tradeoff between accuracy of numerical integration and computation effort. The implicit approach shifts most effort to the preprocessing.

But both are not equivalent! Although they are all subject to the same *mean tempo condition* they create different tempo and timing curves. In measurements of human performed music we could observe a relatively large amount of dispersion [3], so that neither of the here presented models can be proven to be the better one. In certain application contexts one may be more handy than the other. The implicit model may, for instance, be better

type	i_m settings 1st	2nd	diff.	correct answers #	%	significance
acccel.	0.7	0.3	0.4	16	88.9	**0.001**
rit.	0.5	0.4	0.1	5	27.8	0.096
acccel.	0.6	0.8	0.2	9	50.0	1.0
rit.	0.4	0.4	0	8	44.4	0.815
acccel.	0.4	0.3	0.1	9	50.0	1.0
rit.	0.3	0.6	0.3	14	77.8	**0.031**

Table 1. The results of a listener study with 18 participants. Six pairs of continuous tempo transitions had to be classified as identical or different. They differed only with respect to the parameter i_m (see section 2).

suited to analyze and resynthesize empirical data. Nonetheless, both models produce equally valid approximations.

Therefore, the more important question may be: How precisely do they have to mimic each other to be indistinguishable for the human listener? The answer to this question implies also what integration accuracy is needed in the explicit model. Therefore, we conducted a listener study with 18 participants (7 musicians and 11 non-musicians). They listened to 6 pairs of tempo transitions and were asked to decide whether these were different or identical (indistinguishable). They were also given the sheet music for better orientation. The tempo transitions were all of the same length (16 measures with $4/4$ time signature) and were performed with the same music. All accelerandi began with 40 bpm and ended up with 120 bpm. All ritardandi began with 120 bpm and decelerated to 40 bpm. Only the parameter i_m, that controls the course of the tempo curve, was different. The music was rendered with the explicit model at $N = 512$ (that is a segmentation for each 16th note).

The outcome of the study is given in table 1. The participants were clearly able to recognize contrary curve characteristics, that is, "root shape" ($i_m < 0.5$) against "potential shape" ($i_m > 0.5$), that can be found in the first and sixth pair of tempo transitions. All other pairs with differing i_m up to 0.2 could not be distinguished significantly. In our music examples the 0.2 difference produced tempo differences of 10 up to 23 bpm lasting for 10 measures. The resulting timing difference was 15 seconds. Even the 0.1 difference caused tempo differences of 5 up to 8 bpm over 10 measures and timing differences of still 4 seconds.

These results indicate that differing curves that run within a certain narrowband of a tempo curve T_m can be applied alternatively. This narrowband is spanned by varying the parameter $i_m \pm 0.1$. For the listener it will make no difference, especially as the rational Bézier variant compensates its deviations up to the end. In the case of the explicit model even a rough numerical integration with $N \in \{2,4\}$ seems to suffice for most situations. This makes the explicit model unexpectedly suitable for realtime applications, even more than the implicit model. The narrowband furthermore gives a measure for comparison of any tempo models in the literature and which can be used alternatively.

6. CONCLUSION

This paper detailed the formalisms and implementation of our tempo model. We have introduced the *mean tempo condition* as a meaningful feature to describe and distinguish the characteristics of continuous tempo transitions. Based on this feature we conducted a listener study to evaluate the human ability to recognize different tempo transitions. The results indicate perspectives for optimization or simplification.

7. REFERENCES

[1] J. Beran and G. Mazzola, "Timing Microstructure in Schumann's Träumerei as an Expression of Harmony, Rhythm, and Motivic Structure in Music Performance," *Computers and Mathematics with Applications*, vol. 39, pp. 99–130, 2000.

[2] A. Berndt, "Decentralizing Music, Its Performance, and Processing," in *Proc. of the Int. Computer Music Conf. (ICMC)*, M. Schedel and D. Weymouth, Eds. New York, USA: International Computer Music Association, Stony Brook University, June 2010, pp. 381–388.

[3] A. Berndt and T. Hähnel, "Expressive Musical Timing," in *Audio Mostly 2009: 4th Conf. on Interaction with Sound—Sound and Emotion*. Glasgow, Scotland: Glasgow Caledonian University, Interactive Institute/Sonic Studio Piteå, Sept. 2009, pp. 9–16.

[4] A. Friberg and J. Sundberg, "Does Music Performance Allude to Locomotion? A Model of Final Ritardandi Derived from Measurements of Stopping Runners," *The Journal of the Acoustical Society of America*, vol. 105, no. 3, pp. 1469–1484, March 1999.

[5] H. Honing, "The Final Retard: On Music, Motion, and Kinematic Models," *Computer Music Journal*, vol. 27, no. 3, pp. 66–72, 2003.

[6] T. Huckle and S. Schneider, *Numerik für Informatiker*. Berlin, Heidelberg, Germany: Springer, 2002.

[7] R. Hudson, *Stolen Time. The History of Tempo Rubato*, reprinted ed. Oxford, UK: Clarendon Press/Oxford University Press, 2004.

[8] J. Sundberg, A. Friberg, and R. Bresin, "Attempts to Reproduce a Pianist's Expressive timing with Director Musices Performance Rules," *Journal of New Music Research*, vol. 32, no. 3, pp. 317–325, Sept. 2003.

[9] G. Widmer and A. Tobudic, "Playing Mozart by Analogy: Learning Multi-Level Timing and Dynamics Strategies," *Journal of New Music Research*, vol. 32, no. 3, pp. 259–268, Sept. 2003.