# Applied Estimation (EL2320) Project - Tempo-tracking using a switching Kalman Filter and Particle Filtering

Matthaios Stylianidis

December 2021

## Abstract

**In this paper, we evaluate a switching Kalman filter method on the task of estimating the tempo in drumming performances. The model treats the note onsets and the tempo period (inter-beat interval) as a hidden variable and uses the temporal difference between notes on the score sheet (score difference) as the switch variable. The score difference is estimated either by minimizing the residual error between the predicted and the observed note onset, or by using a particle filtering method. Our experiments demonstrate that the former method of computing the score difference, despite its simplicity in terms of number of parameters, achieves performance not far from that of complex state-of-the-art methods in the literature. Nonetheless, the particle filtering method performed very poorly in estimating the tempo, which we attribute to possible implementation faults.**

## I. Introduction

Beat and tempo are two related fundamental characteristics of rhythm in music. Intuitively, beat can be described as the pulse in music to which listeners tap their feet. More formally, songs are divided into blocks called measures, where each measure consists of a number of beats with fixed time intervals between them. The fixed time intervals define a beat frequency in the measure, also called *tempo*, which describes the pace of the song and is typically measured in beats per minute (BPM). In this paper, we deal with the problem of estimating the tempo in music. A more detailed introduction into the aforementioned terminology can be found in [24].

We can distinguish two approaches of estimating the tempo of a song in the literature. The first approach focuses on estimating the tempo globally, requiring the existence of a stable tempo throughout a song, which is often the case in genres such as pop, rock, etc. [26]. The obvious downside of this approach is that there are musical genres where the prescribed tempo can vary over the different parts of the song, but also that intentional or unintentional deviations might occur during human performances, without which music can sound dull [7]. Another downside is that global estimation techniques require the data of the complete song to be available in advance and are not applicable in real-time.

The second approach to tempo estimation is known as tempo tracking, which Cemgil et al. [9] define as "the task of following the tempo in a performance that contains expressive timing and tempo variations". Here the tempo is tracked locally, but one could also extract global features regarding the dominant tempos that exist in a complete song. In contrast to global tempo estimation, local tracking can also be used for real-time applications.

Tempo estimation has been a classical Music Information Retrieval (MIR) task, with tempo extraction being one of the challenges in the Music Information Retrieval Evaluation eXchange (MIREX) [22]. When rhythm is a predominant feature, tempo can be used for estimating music similarity [5], music classification [12, 2], song recommendations, as well as playlist generation [13].

Tempo estimation can also be applied for the rhythmic alignment of multiple instruments or channels (e.g. mixing) [2], as well as in tasks where automatic co-ordination with a beat is needed. More specifically, use-cases that can benefit from accurate tempo tracking are beat-driven computer graphics [2], robot applications (e.g. dancing robot) [15], and automatic music accompaniment [31, 7, 28]. Tempo tracking is also critical in automatic music transcription [7] and can improve the performance of other tasks such as chord recognition [30]. In fact, music transcription can suffer from low quality if we do not account for tempo fluctuations [7].

Another possible application is the one that inspired the present work, which to the best of our knowledge has not been mentioned in the literature. That is, the simple use of a real-time tempo tracker as a practice tool for the musician. During practice, instead of trying to match the beat of a metronome so as to keep a stable tempo, a musician could use a tempo tracking tool to evaluate how stable their tempo has been during practice. This better mimics a live performance where a metronome is not used, and can be especially valuable for musicians who have the role of a metronome and are responsible for keeping a stable tempo for the rest of the members of a band (e.g. percussionists).

Except for the prototypical cases of pop and electronic music, tempo estimation has been described as an unresolved problem [13]. In addition to the variability in human performance, another factor that usually degrades the performance of tempo estimation is syncopated rhythms [11, 7]. In such rhythms, the notes are absent or weak in the beat locations and are present or louder in the off-beat locations, causing models to lose track. In fact syncopated rhythms can make tempo difficult to track even for experienced human listeners [8]. Another factor that can degrade tracking accuracy is the perceived tempo ambiguity, as people can perceive different tempos for the same musical piece [22] (e.g. 60 instead of 120 BPM). Tempo estimation errors due to perceived tempo ambiguities are known as *octave errors* [25].

The main contribution of this work is the implementation and evaluation of a switching state Kalman Filter method for simultaneous tempo-tracking and rhythm quantization (i.e. determining the location and duration of each note in a score sheet) on a large corpus of drum audio recordings. We estimate the switch variable in each timestep analytically or track multiple hypotheses using a particle filter.

## II. Related work

We can distinguish tempo-tracking approaches in those that work with an input stream of discrete time events (e.g. MIDI notes) and those that use some continuous type of input. Note onset times are the features most frequently used in the literature for tracking the beat or tempo. However, even though they are such a frequently used feature, they might be insufficient if there is high complexity in the musical piece, and additional features can improve the performance. Dixon [11] achieved significantly better performance by adding a salience feature, calculated based on the note's pitch.

Other examples of features that can be used for tempo-trackers are the duration of notes or the relative amplitude of each note [18].

Go to and Muraoka [16] built a real-time multi-agent system where each agent detects onsets in different frequency bands of the input and uses those onsets to detect bass and snare drum sounds and extract drum patterns. The detected drum patterns are compared to pre-registered patterns to form a beat hypothesis in each agent, using a template-matching model. The individual agent results are then combined to infer the final beat locations. A downside of this system is that it assumes tempo to be nearly constant with time fluctuations present, so its performance might be poor in cases where this assumption does not hold.

In [23], a real time tempo-tracking method is used, where amplitude based features are extracted from 6 frequency bands of the audio signal and fed to a bank of comb filters. Each comb filter is tuned to a different period corresponding to a distinct tempo and the output of the filters from the different bands are eventually combined to determine the track's tempo.

A few authors have also employed the auto-correlation function to discover repeating patterns that relate to the beat period. In [6], the auto-correlation function is applied on the note onset and duration times acquired from MIDI formatted audio to perform beat tracking. In [10], two different onset detection functions are applied based on the spectral energy of the signal and the auto-correlation function is used on the detection functions' output to estimate the tempo in real time. In [2], the derivative of the frequency content with respect to time is used to perform detection in distinct frequency bands and the auto-correlation function is then used to detect periodicity in each band based on the spectral product at each point in time, finally combining the results from the different bands to obtain the general tempo.

In [1], a beam-search based method is used to track multiple hypotheses simultaneously in real-time and estimate the downbeat onset locations and tempo. To improve the stability of the method, a weighted average of the past tempos is used for estimating the current tempo, where an exponential decay function is used to weight the previous tempos. Moreover, different pruning methods are applied to decrease the magnitude of the beam search when expanding to a new set of possible states, where one of the methods prunes new states that rate low on a musicality, a metric calculated based on the type of notes and their temporal placement in the new state.

Another type of model that has been explored is oscillators. McAuley [20] used coupled adaptive oscillators to perform beat and tempo tracking where each oscillator predicts the next beat position based on the current beat position and period and then uses the closest observed feature to this prediction to update its period and phase. In this model each oscillator has a resting value for its period where it gradually returns in absence of input and a phase resetting mechanism which allows it to reset is phase when its total weighted input over time surpasses a certain threshold.

In [9], Cemgil et al. define an audio representation called the tempogram, which contains information about a collection of events rather than a single note onset, and then use the tempogram representation as a measurement to perform tempo tracking with a Kalman Filter. A simple oscillator model proposed by Pardo [21] achieved worse results, but statistically close to those acquired with the tempogram Kalman Filter.

Shiu and Kuo [27] proposed a modified Kalman Filtering approach for real-time tempo tracking, based on the tempogram Kalman filter model. For measurements, they calculate the cepstral difference between extracted MFCC audio frames, summing the squared differences in each timestep and using the frame with the maximum difference around a predicted beat as an observation. A search window size proportional to the tempo is used around the predicted beat so that the rate of change of the tempo is not based on a fixed value. Finally, they use a method called lock detection, which updates the measurement noise covariance values proportionally to the estimated tempo or sets them to 0 if the prediction error is below a certain threshold. The idea behind this is that the higher the tempo, the larger the possibility of having noisy measurements. In [28], Shiu and Kuo propose an enhanced probabilistic data association (EDPA) method for noisy measurement detection which considers information not only from onset intensities but also from prediction residuals in order to make tracking of beats with insignificant intensities possible.

In [7], Markov Chain Monte Carlo (MCMC) and sequential Monte Carlo methods are evaluated in simultaneous tempo-tracking and rhythm quantization, given a stream of discrete note onsets. A switching state space model is used where the switch variable corresponds to the onset location difference (score difference) and the hidden variable contains the tempo. A modified Viterbi algorithm is proposed on a Rao-Blackwellized version of a particle filter in order to deal with the high dimensionality of the hidden variable. Experiments showed that the particle filter method outperformed MCMC methods such as Gibbs sampling, simulated annealing, or iterative improvement.

Krebs et al. [19] suggested four different particle filter methods to overcome the problem of tracking multimodal probability distributions caused by ambiguities in tempo and phase. In their work, they used a variant of the spectral flux onset features [4], calculating the bin differences of consecutive extracted spectra and using their standardized sums over the low and high frequency bands as observation features [19]. Their proposed methods infer beat positions, tempo, as well as the time signature of the measure. To deal with particle deprivation, they employ systematic resampling selectively when the effective number of particles is above a certain threshold, but they also introduce the auxiliary particle filter (APF) and mixture particle filter (MPF) models. APF applies a monotonically increasing function to the weights before resampling to increase the survival probability, while MPF clusters the particles with a custom distance function and applied resampling separately in each cluster. Finally, an auxiliary mixture particle filter (AMPF) method is used, combining APF and MPF. Their experiments show that particle filtering is advantageous in terms of computational cost and achieves state-of-the-art accuracy, while the AMPF handles the deprivation problem better than the standard particle filter.

In [17], two different particle filter methods are proposed for beat and tempo tracking, loosely based on Cemgil's particle filter [7]. Observation features for strong energy sounds (e.g. drum bass) are extracted by taking the energy envelopes of three distinct frequency bands and calculating their gradients. For sound changes not associated with large energy changes (i.e. harmonic changes), a modified KL divergence measure is used to calculate the spectral change in the harmonic spectral region. The strong energy feature and harmonic change features are finally clustered and the clustering result is fed to the particle filter.

More recent beat and tempo tracking approaches have employed neural networks. In [25], a CNN directly estimates tempo from an MFCC spectrogram in a single step. The tempo estimation task is framed as a multi-class classification problem where the network classifies the input in one of 256 tempo classes. The authors mention that the CNN requires 11.9 seconds of audio as an input so it can be applied to real-time applications. However, it is not clear how good performance the network has in terms of execution time.

In [4] a bi-directional LSTM recurrent neural network is used to perform frame by frame beat classification of the signal, where spectral features of the audio are computed for each frame and an auto-correlation function is used to determine the pre-dominant tempo to eliminate erroneously tracked beats or add missing ones. In [3], the same RNN is used to learn an intermediate beat-level representation, which represents the probability of a frame being in a beat location, and is fed to a bank of resonating comb filters to determine the dominant tempo.

## III. Method

### A Datasets

### A.1 Groove MIDI Dataset

The Groove MIDI Dataset (GMD) is a corpus of 1.150 recordings corresponding to 13.6 hours of human performed, tempo-aligned expressive drumming [14]. For our experiments, we use the discrete note onset stream from the MIDI files as input to our models. In each recording, a drummer is playing along to a metronome set at a specific tempo with different performances having different tempos or time signatures. More than 80% of the data consists of recordings from hired professionals.

The dataset includes several short recordings corresponding to short drum fills and beats. We discard such recordings by filtering out those that are less than 5 seconds long. After this step, we are left with 517 recordings, from which 385 belong to the training set, 52 to the validation set, and 80 to the test set.

### B Evaluation method

To evaluate our method, we report values for the *Accuracy2* metric, as defined in [25]. The metric considers tempo predictions to be accurate compared to the ground truth using a 4% tolerance, while also considering correct any tempo predictions that are off by a factor of 2 or 3 to account for octave errors. Since our method performs local tempo tracking and makes multiple predictions for each recording, we evaluate its ability to converge to the correct tempo value by comparing the estimated tempo at the end of the recording to the ground truth, after rounding the tempo to the closest integer. We select the parameters that result in the best performance on the union of the training and validation set and evaluate the model's performance by reporting the average *Accuracy2* metric value on the test set.

### C Switching Kalman Filter

To track the tempo we use the switching Kalman Filter model proposed in [18]. In this approach the inter-beat period $\Delta_k$ and the onset times $r_k$ comprise the hidden state $x_k$ modelled by the Kalman Filter and the onset times provided by the MIDI recordings are used as observations. Figure 1 depicts a Bayesian network with the dependencies of the variables involved in the Kalman Filter modelling.

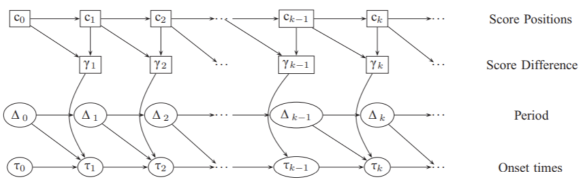

Figure 1: Bayesian network with variable dependencies involved in our Kalman Filter modelling. The square and the oval shapes denote discrete and continuous random variables, respectively. (taken from [18])

The score position of a note $c_k$ is defined as the discrete temporal position in a measure at which the note occurs, which resets to 0.0 after the end of each measure. From the score positions we can calculate the score difference, which is defined as $\gamma_k = c_k - c_{k-1}$. The score difference is defined as the switching variable, according to which we update our transition matrix $A$ at the arrival of each new observation. The transition matrix assumes a stable tempo and is used to predict the next state's onset and tempo (inter-beat) period as follows:

$$\begin{pmatrix} r_k \\ \Delta_k \end{pmatrix} = A \begin{pmatrix} r_{k-1} \\ \Delta_{k-1} \end{pmatrix} + w_k = \begin{pmatrix} 1 & \gamma_k \\ 0 & 1 \end{pmatrix} \begin{pmatrix} r_{k-1} \\ \Delta_{k-1} \end{pmatrix} + w_k \tag{1}$$

where $w_k \sim N(0, Q)$ and $Q$ is a spherical covariance matrix with the variance of the onset and tempo process noise on the diagonal. Thus, every time the switch variable $\gamma_k$ changes, we use a different linear dynamical model. In order to pick the right values for $Q$, we perform an exhaustive search over the hyperparameters on the training and validation set using the following possible values for both the onset and tempo: $[0.005, 0.01, 0.05, 0.1, 0.2]$.

The relationship between the hidden state $x_k$ and the note onsets $y_k$ in our model is formulated as follows:

$$y_k = Cx_k + u_k = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} r_k \\ \Delta_k \end{pmatrix} + u_k \tag{2}$$

where $C$ is the observation matrix and $u_k$ $N(0, R)$ a Gaussian white noise variable representing the noise in the MIDI measurements where we choose the best performing $R$ from the space $[0.01, 0.1, 0.2]$. As default values for $\Delta_k$ and $r_k$ we use 1 (i.e. 60 BPM) and 0.0.

Given the linear dynamical system described above, if the score of the performance is known in advance, then we can simply estimate the tempo using the standard Kalman Filter equations as described in [18], updating the switch variable after the arrival of each observation. Otherwise, we must employ a method for estimating the score difference.

### C.1 Ideal score difference

Given the above linear dynamical system, it can be shown that the ideal switch variable value that minimizes the error between the observed and the predicted onset is given by the following equation [18]:

$$\gamma_k = \frac{y_k - r_{k-1}}{\Delta_{k-1}} \tag{3}$$

Following [18], the estimated score difference is then rounded to the closest discrete value, given a set of possible values taken from a grid with a range from 0 up to 4 and a resolution of 0.25.

### C.2 Multiple hypotheses tracking

A $\gamma$ value can be highly probably in one timestep, but its likelihood can drop after a few new observations. In order to track multiple $\gamma$ values, we can treat the switch variable as a hidden variable and a particle filter can be employed to estimate its posterior probability. Again, we follow the same approach as in [18]. We track a state hypothesis with each particle and we calculate the ideal $\gamma$ value using Equation (3) at each timestep, expanding each particle to a set of $S$ new particles, where $S$ is the number of discrete values in the neighborhood of $\gamma$ on the discrete grid. After the $n$ particles are expanded to a new set of $n \times S$ particles, we prune the new set by keeping only the $n$ particles with the highest likelihood. For our experiments, we set $n$ to 25 and $S$ to 5. For the expansion of each particle to a new state and the update of each particle's state, we use the same Kalman Filtering equations we described earlier.

The likelihood of the particles after the expansion is computed by plugging the residual error between the predicted onset time and the observed time to a Gaussian distribution. Unfortunately, the approach for calculating the spread of this Gaussian described in [18] was not straightforward to understand and implement. Therefore, in our experiments we use a Gaussian with 0 mean and a variance equal to the measurement noise $Q$. For each timestep of tempo-tracking we consider the tempo estimated by the particle with the highest likelihood as the output of our model. Taking the weighted mean of all particles would prove problematic, especially since the true posterior is multi-modal due to the tempo ambiguities.

We also conducted preliminary experiments with an increased number of particles beyond 100 and an increased neighborhood size $S$. In addition, we implemented a particle filtering method loosely based on the one above. In that particle filtering method, the score difference is treated as part of the hidden state, and each of the $n$ particles with the highest likelihoods gets expanded to $S$ new states in each timestep, where $S$ is all the possible discrete values for the score difference. Without any further optimization, both aforementioend approaches resulted in prohibitive computational cost. Therefore, we will not be reporting performance values for either the configuration with an increased $n$ and $S$, or for our alternative particle filtering method.

## IV. Results

Table 1 depicts the test *Accuracy2* performance achieved on the GMD dataset with both of our tempo tracking methods. We observe that the single estimated value of the ideal score difference resulted in better performance than the particle filtering method. This is against our expectations of a higher performance for the particle filtering method, since the score difference values are sampled in the neighborhood of the ideal score difference. Due to time constraints, we could not identify the error behind the poor performance of the particle filter, and hence we hypothesize that it is due to potential faults in our implementation of the method. As a reference to compare our methods against, state-of-the-art bi-directional RNN and CNN models achieve a performance ranging from 86% to 95.4% on the full GMD dataset [29], without discarding any of the short recordings. These results suggest that even though our method is much simpler in terms of its total number of parameters, its performance is satisfactory when compared to complex neural network based approaches. To highlight the complexity of the state-of-the-art methods, the best performing neural network in [29] has 6.583.772 trainable parameters.

| **Estimation method** | **Test *Accuracy2*** |
|---|---|
| Ideal score difference | 85% |
| Particle filtering | 15% |

Table 1: *Accuracy2* performance achieved on the GMD test dataset.

In Table 2, we observe the performance of our best performing switching Kalman Filter method on the different musical genres that comprise the test set. We observe that our model struggled mostly with funk, jazz, and rock music, while achieving perfect accuracy for genres such as hip-hop, latin, or soul. These results are to a large extent in line with our expectations. Lower performance was expected in jazz and funk music where complex rhythms are more common. Moreover, rock music has often lower complexity in rhythm, but is not as simple as pop or hip-hop music. It is worth noting, however, that our test set includes only simple 4/4 time signatures, while the training and validation set includes a variety of more complex time signatures. Thereby, regardless of the genre type, the songs of the test set have less complicated rhythms than the training and validation set.

Figure 2 depicts the tempo tracker's output for an example test performance. We plot the original predicted tempo series as well as the series multiplied by a factor of 2 and 3, alongside the ground truth tempo. We observe that the first predicted tempo values are oscillating away from the ground truth tempo, but quickly converge close to the real value after 5 seconds.

| **Genre** | **Test *Accuracy2*** | $\mathbf{n_{samples}}$ |
|---|---|---|
| Afrobeat | 100% | 1 |
| Funk | 62.5% | 16 |
| Highlife | 100% | 1 |
| Hiphop | 100% | 7 |
| Jazz | 0.57% | 7 |
| Latin | 100% | 4 |
| Pop | 100% | 4 |
| Punk | 100% | 1 |
| Reggae | 100% | 1 |
| Rock | 85.7% | 21 |
| Soul | 100% | 18 |

Table 2: *Accuracy2* performance of our method, using the ideal score difference estimation approach, on different genres of the GMD test dataset.

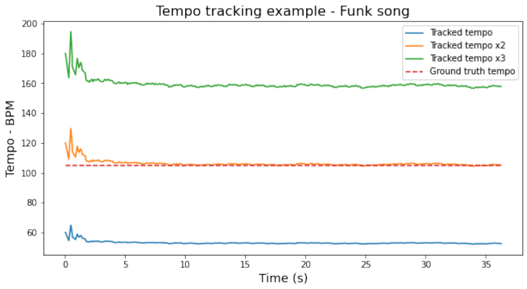

Figure 2: Tracking example on a test set sample: Blue line denotes the tracked tempo, orange line and green line denote the tempo multiplied by a factor of 2 and 3, and the dashed red line denotes the true tempo value.

## V. Conclusion and future work

In this paper we evaluated a switching Kalman filter model on the task of estimating the tempo of drum performances. The switch variable was computed either by minimizing the residual between the model's prediction and the observed note onset, or by using a particle filtering method. The particle filtering method was expected to increase performance by tracking multiple hypotheses, but instead performed very poorly compared to the first method. Since the particle filtering model estimates switch variable values around the ideal value that minimizes the residual, we hypothesize that the poor performance may be a result of an implementation fault. We also compare our performance to that achieved on the same dataset by other state-of-the-art methods in the literature. Our method is a simpler one than the state-of-the-art methods based on neural networks, yet the accuracy achieved is approximately 12% lower than the top performing neural network. Additional experiments that we performed with more particles or with an alternative particle filtering method where the switch variable is treated as a discrete hidden variable led to prohibitive computational cost and were aborted early on. Finally, when looking at our performance over the different musical genres, our results reinforce the prior belief that genres such as jazz or funk are difficult to perform tempo tracking on due to complex rhythms.

To extend our work, one could apply our methods in other available datasets to verify the performance difference between them and compare them to more available methods. Instead of datasets with recordings with a nearly constant tempo, datasets with variable tempo could be used to better evaluate our methods in the task of tempo tracking. To apply our method on datasets that consist of raw waveform signals, one would need to add an accurate onset detection function that extracts the note onset times from the signal. Moreover, other additional features could be used for tracking the tempo, such as the pitch or the duration of the notes.

## References

[1] Paul E Allen and Roger B Dannenberg. "Tracking musical beats in real time". In: *ICMC*. 1990.

[2] Miguel Alonso, Bertrand David, and Gaël Richard. "A study of tempo tracking algorithms from polyphonic music signals". In: *4th COST 276 Workshop*. Citeseer. 2003.

[3] Sebastian Böck, Florian Krebs, and Gerhard Widmer. "Accurate Tempo Estimation Based on Recurrent Neural Networks and Resonating Comb Filters." In: *ISMIR*. 2015, pp. 625–631.

[4] Sebastian Böck and Markus Schedl. "Enhanced beat tracking with context-aware neural networks". In: *Proc. Int. Conf. Digital Audio Effects*. 2011, pp. 135–139.

[5] Dmitry Bogdanov et al. "Unifying low-level and high-level music similarity measures". In: *IEEE Transactions on Multimedia* 13.4 (2011), pp. 687–701.

[6] Judith C Brown. "Determination of the meter of musical scores by autocorrelation". In: *The Journal of the Acoustical Society of America* 94.4 (1993), pp. 1953–1957.

[7] Ali Taylan Cemgil and Bert Kappen. "Monte Carlo methods for tempo tracking and rhythm quantization". In: *Journal of artificial intelligence research* 18 (2003), pp. 45–81.

[8] Ali Taylan Cemgil and Bert Kappen. "Tempo tracking and rhythm quantization by sequential monte carlo". In: *NIPS*. 2001, pp. 1361–1368.

[9] Ali Taylan Cemgil et al. "On tempo tracking: Tempogram representation and Kalman filtering". In: *Journal of New Music Research* 29.4 (2000), pp. 259–273.

[10] Matthew EP Davies and Mark D Plumbley. "Causal Tempo Tracking of Audio." In: *ISMIR*. 2004.

[11] Simon Dixon. "Automatic extraction of tempo and beat from expressive performances". In: *Journal of New Music Research* 30.1 (2001), pp. 39–58.

[12] Arndt Eppler et al. "Automatic style classification of jazz records with respect to rhythm, tempo, and tonality". In: *Proc. of the Conference on Interdisciplinary Musicology (CIM)*. 2014.

[13] Hadrien Foroughmand and Geoffroy Peeters. "Deep-rhythm for tempo estimation and rhythm pattern recognition". In: *International Society for Music Information Retrieval (ISMIR)*. 2019.

[14] Jon Gillick et al. "Learning to Groove with Inverse Sequence Transformations". In: *International Conference on Machine Learning (ICML)*. 2019.

[15] Aggelos Gkiokas and Vassilis Katsouros. "Convolutional Neural Networks for Real-Time Beat Tracking: A Dancing Robot Application." In: *ISMIR*. 2017, pp. 286–293.

[16] Masataka Goto and Yoichi Muraoka. "A real-time beat tracking system for audio signals". In: *ICMC*. 1995.

[17] Stephen W Hainsworth and Malcolm D Macleod. “Particle filtering applied to musical tempo tracking”. In: *EURASIP Journal on Advances in Signal Processing* 2004.15 (2004), pp. 1–11.

[18] Tim van Kasteren. “Realtime tempo tracking using Kalman filtering”. In: *so* (2006).

[19] Florian Krebs et al. “Inferring metrical structure in music using particle filters”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.5 (2015), pp. 817–827.

[20] J Devin McAuley. “Perception of time as phase: Toward an adaptive-oscillator model of rhythmic pattern processing”. PhD thesis. Indiana University, 1995.

[21] Bryan Pardo. “Tempo Tracking with a Single Oscillator.” In: *ISMIR*. 2004.

[22] Zbigniew W Ras and Alicja Wieczorkowska. *Advances in music information retrieval*. Vol. 274. Springer, 2010.

[23] Eric D Scheirer. “Tempo and beat analysis of acoustic musical signals”. In: *The Journal of the Acoustical Society of America* 103.1 (1998), pp. 588–601.

[24] Catherine Schmidt-Jones. “Understanding basic music theory”. In: (2013).

[25] Hendrik Schreiber and Meinard Müller. “A Single-Step Approach to Musical Tempo Estimation Using a Convolutional Neural Network.” In: *Ismir*. 2018, pp. 98–105.

[26] Hendrik Schreiber, Julián Urbano, and Meinard Müller. “Music Tempo Estimation: Are We Done Yet?” In: *Transactions of the International Society for Music Information Retrieval* 3.1 (2020).

[27] Yu Shiu and C-C Jay Kuo. “A modified Kalman filtering approach to on-line musical beat tracking”. In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP’07*. Vol. 2. IEEE. 2007, pp. II–765.

[28] Yu Shiu and C-C Jay Kuo. “Musical beat tracking via Kalman filtering and noisy measurements selection”. In: *2008 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2008, pp. 3250–3253.

[29] Mila Soares de Oliveira de Souza, Pedro Nuno de Souza Moura, and Jean-Pierre Briot. “Music Tempo Estimation via Neural Networks–A Comparative Analysis”. In: *arXiv preprint arXiv:2107.09208* (2021).

[30] Adam M Stark, Matthew EP Davies, and Mark D Plumbley. “Real-time beat-synchronous analysis of musical audio”. In: *Proceedings of the 12th Int. Conference on Digital Audio Effects, Como, Italy*. 2009, pp. 299–304.

[31] George Tzanetakis and Graham Percival. “An effective, simple tempo estimation method based on self-similarity and regularity”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2013, pp. 241–245.