EQUILIBRIA OF ADAPTIVE WAVETABLE OSCILLATORS WITH APPLICATIONS TO BEAT TRACKING

William A. Sethares and Raman Arora

University of Wisconsin
Department of Electrical and Computer Engineering
1415 Engineering Drive, Madison WI 53706

ABSTRACT

An adaptive oscillator is a system that can lock onto a time-varying input signal, synchronizing its output to both the frequency and phase of the input. A wavetable oscillator generates a periodic output by indexing into a lookup table that stores a single period of the waveform. An adaptive wavetable oscillator (AWO) combines these two ideas in a technique which separates the periodic output waveform from the parameters that control the adaptation of the frequency and phase of the waveform. The key issues in the design of AWOs are: the kind of oscillator to use, the class of admissible inputs, the shape of the wavetable, the control parameters, and the adaptive algorithm that adjusts the parameters. Wavetable oscillators can be applied to track the beat in MIDI signals, or, after an appropriate psycho-acoustical data reduction, to the tracking of audio signals. This paper examines these issues through analysis and simulation, focusing on conditions that achieve the desired entrainment between output and input. Sound examples demonstrate the application to beat tracking.

***Index Terms*—** adaptive systems, nonlinear oscillators, table lookup, oscillator stability

1. INTRODUCTION

Adaptive oscillators attempt to respond to external events by adjusting the frequency and phase of oscillation to achieve entrainment. They must generate a periodic waveform, admit an input, be capable of synchronizing to the period and phase of the input, and be robust to noisy periods, missing events, and unexpected events. In the beat-tracking problem, a locally generated waveform attempts to match phase and period to follow the foot-tapping "beat" of a musical passage ([7], [13]) which may originate as a MIDI signal or may be derived from an audio input. In typical oscillators, such as those of Van der Pol [15] and Fitzhugh-Nagumo [5], the state of the oscillator is inextricably tied to the period of oscillation; thus the "shape" of the output waveform changes with frequency. Adaptive wavetable oscillators, on the other hand, separate the detailed shape of the periodic waveform from the control signals that specify the phase and frequency of the oscillation. This separation allows standard techniques from the analysis of adaptive filters to be applied.

A number of researchers have investigated the application of oscillators to the beat tracking problem. These include work by Eck [4], McAuley [9], and Large and Kolen [7], each of whom apply particular osillator structures to the beat tracking problem. These draw from the tradition of Povel's [12] work which posits rhythmic activities as the result of an entrainment between an internally generated clock (the oscillator) and external events. Perhaps the most successful of these systems is Toiviainen's real time MIDI beat follower [14].

This paper introduces AWOs in Sect. 2 and then examines the behavior of the oscillators as the parameters adapt to follow various input signals in Sect. 3. The adaptation is analyzed in a variety of settings and the optimum (equilibrium) points of the algorithm are shown graphically and derived analytically. Examples in Sect. 4 demonstrate that the oscillators can achieve entrainment in a musical environment.

2. ADAPTIVE WAVETABLE OSCILLATORS

A wavetable oscillator consists of an array w containing N stored values of a waveform. The output of the oscillator at time k is

$$o[k] = w((s[k] + \beta) \bmod N)$$

where mod N is the remainder after division by N and where the indices into w are given by

$$s[k] = (s[k-1] + \alpha) \bmod N. \quad (1)$$

The index is initialized as $s[0] = 0$. The parameter α specifies the frequency of the oscillation while β defines the phase. The oscillator can be made adaptive by adjusting the parameters to align the oscillator with an external input. This can be accomplished in several ways. Suppose that the input to the oscillator is $i[k]$. One possibility is to use a correlation-style cost function

$$J(\beta) = \text{LPF}\{i[k]o[k]\} \quad (2)$$

which parallels the cost function used in a standard PLL [6]. The β that maximizes J provides the best fit between the input and the oscillator. It can be adapted using a gradient descent strategy

$$\begin{aligned}\beta[k+1] &= \beta[k] + \mu \frac{dJ}{d\beta} \\ &= \beta[k] + \mu \mathrm{LPF}\left\{ \; i[k] \left. \frac{dw}{d\beta} \right|_{\beta=\beta[k]} \right\}. \end{aligned} \tag{3}$$

Since w is defined by a table of values, $\frac{dw}{d\beta}$ is another table, the numerical derivative of w. Several candidate wavetables and their derivatives are shown in Fig. 1.

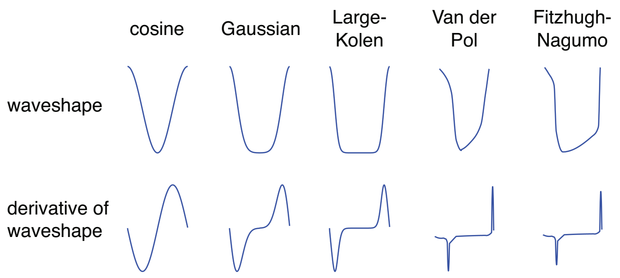

Fig. 1. Five common wavetables and their derivatives. The cosine wavetable is used in the PLL. The Gaussian shape is shifted so that the largest value occurs at the start of the table. The wavetable for the Large-Kolen oscillator is defined by $1 + \tanh(\gamma \cos(2\pi f t) - 1)$. The Van der Pol and Fitzhugh-Nagumo waveshapes are defined using waveforms culled from numerical simulations.

The frequency parameter can also be adapted in order to “learn” the frequency of the input and to continue at the new frequency even if the input stops. Perhaps the simplest technique is to use a gradient strategy that maximizes $J(\alpha) = \mathrm{LPF}\{i[k]o[k]\}$. This is:

$$\begin{aligned}\alpha[k+1] &= \alpha[k] + \mu_\alpha \frac{dJ}{d\alpha} \\ &= \alpha[k] + \mu_\alpha \mathrm{LPF}\{ \; i[k] \left. \frac{dw}{ds}\frac{ds}{d\alpha} \right|_{\alpha=\alpha[k]} \}. \end{aligned} \tag{4}$$

Since $s[k]$ is defined by the recursion (1), the derivative with respect to α cannot be expressed exactly. Nonetheless, when the stepsizes are small, it can be approximated by unity. Because the frequency parameter is more sensitive, its stepsize μ_α is usually chosen to be considerable smaller than the stepsize used to adapt the phase.

In adapting the β's and α's of the AWO, other cost functions may be used. Minimizing $J_{LS} = \mathrm{LPF}\{(i[k] - o[k])^2\}$ leads to an update that optimizes a least-squares criterion while maximizing $J_C = \mathrm{LPF}\{(i[k]o[k])^2\}$ leads to a method that parallels the “Costas loop” [6].

3. BEHAVIOR OF THE ADAPTATION

When attempting to locate the position of a train of spikes in time, oscillators that use pulses (such as Large and Kolen's or the Gaussian) are intuitively plausible. The pulse can be thought of as providing a window of time over which the oscillator “expects” another spike to occur. If the spike occurs at exactly the right time, the derivative is zero and there is no change. If the spike occurs slightly early, the derivative is positive and the phase increases. If the spike occurs late, the derivative is negative and the phase decreases. This process of adjustment actively aligns the oscillator with the spike train. Just as importantly, there is a zone between pulses where the value of the waveshape and its derivative are both small. In this region, the update term is small and the oscillator is insensitive to extraneous spikes and noisy data.

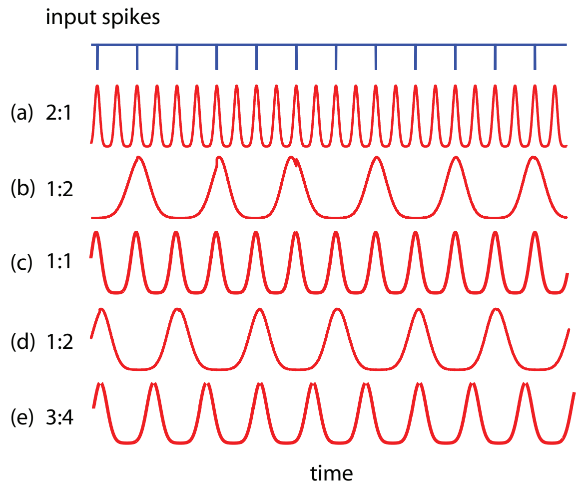

Fig. 2. The input spike train (5) excites the AWO. The initial value in (a) was $\alpha = 240$ ms and the oscillator synchronizes to a $2{:}1$ rate (two oscillator pulses occur for each input spike). (b) and (d) were initialized at $\alpha = 1050$ ms. The oscillator synchronizes to a $1{:}2$ rate (one oscillator output for every two input spikes). Depending on the initial value of β, the oscillator can lock onto either the odd or the even spikes. (c) was initialized at $\alpha = 550$ ms. Other synchronizations such as $3{:}2$ are also possible.

Fig. 2 shows how the AWO responds to an input

$$i(t) = \begin{cases} 1 & t = nT, \; n = 1, 2, \ldots, M \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

that is a regular train of spikes spaced $T = 500$ ms apart. The simulation uses a Gaussian pulse with phase and frequency parameters adapted according to (3) and (4). In (c), α was initialized with period 550 ms, corresponding to a 10% error. The phase and frequency converge within a few seconds and the pulses align with the spikes. The oscillator continues at the adapted frequency even after the input ceases. The same oscillator may synchronize in various ways to the same input depending on the initial values. The figure also shows 1:2, 2:1, and 3:2 entrainments where n:m means that n periods of the oscillator occur in the same time as m periods of the input. While such nonunity entrainments are common in the mode locking of oscillators, they are encouraged by the specifics of the waveshape: the dead (zero) region between pulses means that the adaptation is insensitive to spikes that occur far away from expected location. These simulations (using the Gaussian pulse shape) are effectively the same as when using a Large-Kolen oscillator or a cosine oscillator (as suggested in [8]), suggesting that the details of the waveshape are not particularly crucial to the ability of the adaptation to achieve synchronization.

Adaptive oscillators are often designed to synchronize to specific classes of input sequences such as spike trains (5). When the input is indeed of this form, it is reasonably straightforward to understand the convergence of the oscillator by plotting the cost function for all possible values of the parameters. The two-dimensional cost function for the AWO with Gaussian waveshape and correlation cost $J(\alpha, \beta)$ of (2) is shown in Fig. 3. The summits are the values to which the algorithm converges; when initialized at some α, β, the parameters adjust so that the cost increases at each time step. The parameter α is normalized so that unity corresponds to one period of the oscillator for each input spike. As β ranges between zero and α, it covers all the possible phases.

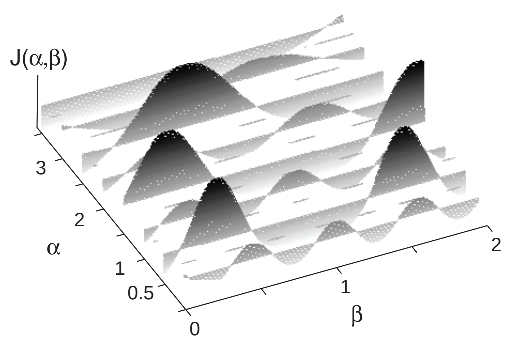

Fig. 3. The cost surface for the phase and frequency updates (3)-(4) of the AWO. Depending on the initial values, the period may converge to $\alpha = 1, 2, 3, \ldots$ or any of the other peaks at integer multiples. There are also stable regions of attraction surrounding $\alpha = 0.5, 1.5, 2.5,$ and 3.5, which correspond to various n:m synchronizations.

Observe that the oscillator may converge to different values depending on its initialization. If started near $\alpha = 1$, it inevitably synchronizes so that each period of the oscillator is aligned with an input spike. But other values are possible: α may converge to an integer multiple, or to a variety of n:m synchronizations. Some of the behaviors that are suggested by the cost surfaces are provable. For example, Appendix C in [1] analyzes the cost function for the correlation cost when the input and output are spike trains. As expected, the result is that the cost is maximized exactly when the frequency of the oscillator matches the rate of the input, or at some simple integer multiple (or submultiple).

Obviously, the wavetable $w(t)$ may assume a large variety of different shapes, as suggested in Fig. 1. But it cannot be arbitrary. Assume that the output of the AWO is

$$o(t) = \sum_{m=1}^{N} w(t - m\alpha - \beta) \tag{6}$$

and let the input be a pulse train with period α^* and phase β^*

$$i(t) = \sum_{n=-\infty}^{\infty} \delta(t - n\alpha^* - \beta^*). \tag{7}$$

The cost function is then

$$J(\alpha, \beta) = \sum_{m=1}^{N} \sum_{n=1}^{N} w(n\alpha^* + \beta^* - m\alpha - \beta). \tag{8}$$

If the wavetable (a) has a global maximum at $t = 0$, (b) has support $(-\frac{\alpha}{2}, \frac{\alpha}{2})$, (c) is monotonic increasing in $(-\frac{\alpha}{2}, 0)$, and (d) is symmetric about $t = 0$, then the cost function achieves its extremal points at one of the simple integer multiples n:m which correspond to the points at which the algorithm achieves synchronization. Appendix D of [1] provides the details and a proof.

4. APPLICATION TO MIDI BEAT TRACKING

An AWO is applied to a MIDI rendition of the Beatles' song *Michelle* drawn from the *Music, Mind, Machine* website [11] of expressive polyphonic piano performances which exhibit "considerable fluctuation in the tempo" [2]. First, the MIDI event list file is processed to turn the data into a sampled spike train suitable for input to an AWO. Suppose that the *i*th element of the MIDI event list occurs at time t_i. Let $a[i] = 1$ if this is a note-on event and zero otherwise, and quantize t_i to the nearest integer multiple of the sampling rate ($T_s = \frac{1}{1000}$ of a second is used in the simulations). The spike train is then the function

$$x(t) = \begin{cases} a[i] & \text{if } t = t_i \\ 0 & \text{otherwise} \end{cases}. \tag{9}$$

A wavetable is chosen (Gaussian is used in the simulations) along with initial (nominal) values for the period α and phase β (0.4 and 0). When all goes well, the oscillator synchronizes to the input $x(t)$ in such as way that the maximum value of the output (a Gaussian pulse train with period defined at time k by α_k) occurs at each beat location, or at some simple integer multiple (or divisor) of the beat, as suggested by the analysis of the previous section.

The beat tracking can be observed by superimposing a noise burst at each detected beat time. The results can be heard on our website at [10]. The noise bursts lock onto the beat rapidly and follow changes in the pulse. Careful listening reveals a glitch at around 29 sec, which is caused by a rapid succession of note events that momentarily increase α. By about 32 seconds, the pulse is regained. Because of the projection, the process is causal and can be implemented in real time (though the simulations reported here are implemented offline in MATLAB).

The cost surface for *Michelle* is plotted in Fig. 4. Overall, the surface is considerably rougher than the cost function for the idealized pulse train in Fig. 3. The string of large peaks represents the beat rate to which the algorithm converges. The various possibilities for β represent successive beats to which it might lock. There are stable points for both larger and smaller α (corresponding to both slower and faster tap rates), though the quarter-note pulse has the largest region of attraction. By initializing the oscillator at different values, it is possible to locate these other levels of the metric hierarchy.

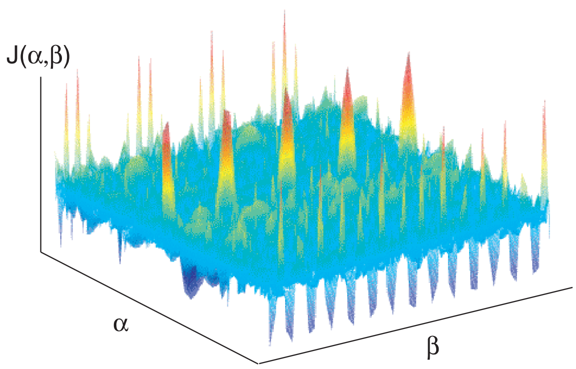

Fig. 4. The cost surface for the phase and frequency updates (3)-(4) of the AWO for the song *Michelle*.

This shows the overall averaged surface over which the algorithm evolves. In operation, the algorithm reacts to somewhat rougher time-varying instantaneous surfaces. One may posit, for example, that at the time of the glitch (at 29 seconds), the surface momentarily flattens or bends away from the steady underlying pulse.

5. CONCLUSIONS

The are many ways to accomplish the beat-tracking task. The use of oscillators that can adapt their parameters has several advantages: it is computationally simple, collections of oscillators can be combined to simultaneously locate several levels of a metric hierarchy, and the internal parameters (representing period and phase) are easy to understand and initialize. AWOs are a particularly appealing type of oscillator because they cleanly separate the dynamics of the state from the dynamics of the adaptation. The drawing of cost functions allows the behaviors of the algorithm to be easily visualized and some of the simpler behaviors are provable.

6. REFERENCES

[1] R. Arora and W. A. Sethares, “Adaptive wavetable oscillators,” to appear in *IEEE Trans. Signal Processing*.

[2] A. T. Cemgil, B. Kappen, P. Desain and H. Honing “On tempo tracking: tempogram representation and Kalman filtering,” *J. New Music Research*, Vol. 28, No. 4, 2001.

[3] S. Dixon, “Automatic extraction of tempo and beat from expressive performances,” *J. New Music Research*, Vol. 30, No. 1, 2001.

[4] D. Eck, “Finding downbeats with a relaxation oscillator,” *Psychological Research*, 66(1), 18-25, 2002.

[5] R. Fitzhugh, “Impulses and physiological states in theoretical models of nerve induction,” *Biophysical Journal*, 1, 455-466, 1961.

[6] C. R. Johnson, Jr. and W. A. Sethares, *Telecommunication Breakdown: concepts of communications transmitted via software-defined radio*, Prentice-Hall, 2004.

[7] E. W. Large and J. F. Kolen, “Resonance and the perception of musical meter,” *Connection Science* 6: 177-208, 1994.

[8] J. D. McAuley, Time as phase: a dynamic model of time perception,” *Proc. Sixteenth Conf. of Cognitive Science Society*, Lawrence Erlbaum, 607-612, 1994.

[9] J. D. McAuley, *Perception of time as phase: toward an adaptive-oscillator model of rhythmic pattern processing*, Ph.D Thesis, Indiana University, 1995.

[10] The sound example may be downloaded from http://eceserv0.ece.wisc.edu/∼sethares/beatstuff/Michelle-MIDIOsc.mp3

[11] *Music, Mind, Machine*, http://www.nici.kun.nl/mmm

[12] D. J. Povel, “A theoretical framework for rhythmic perception,” *Psychological Research* 45: 315-337, 1984.

[13] W. A. Sethares, *Rhythm and Transforms*, Springer, 2007.

[14] P. Toiviainen, “An interactive MIDI accompanist,” *Comp. Music J.*, 22(4):63-75, 1998.

[15] B. van der Pol and J. van der Mark, “The heartbeat considered as a relaxation oscillation and an electrical model of the heart,” *Phil. Mag. Suppl.* No. 6, 763-775, 1928.