Tone Generation with Polyphonic Cycles and Spline Modeling

Matt Klassen
DigiPen Institute of Technology
mklassen@digipen.edu

Greg Dixon
DigiPen Institute of Technology
greg.dixon@digipen.edu

ABSTRACT

In this paper we introduce polyphonic cycles for waveforms which can be used to produce tones with harmonic content. The cycles are first modeled with cubic splines, then mixed at the cycle level. Interpolation between cycles is also used to change timbre from instrument-based timbre to polyphonic content within the duration of one tone. We also introduce glissandos of polyphonic tones which are computed as sequences of cycles of increasing (or decreasing) length. These techniques, as well as melodic contour generation from spline cycles, are used in the accompanying composition "SplineKlang".

1. INTRODUCTION

Interplay between musical elements such as pitch, rhythm, harmony, and timbre, has been a tool and inspiration for music composition for centuries. In this paper we discuss methods and experiments with the generation of timbre which are directly related to harmonic content. This relationship is established at the very small time scale of one cycle. For example, a tone at the pitch A220 has one cycle of length $\frac{1}{220}$ second or roughly 5 milliseconds. Harmonic relations between pitches can be "baked into" a tone at the level of one cycle, using just intervals or frequency ratios. We refer to such cycles containing harmonic content "polyphonic cycles".

We model cycles of waveforms with polynomial splines, which we review briefly in section 2. We then define polyphonic cycles in section 3.1 and explore our main example of these using seventh chords in 3.2. In section 4 we summarize the technique of melodic contour generation from one cycle of a waveform. Finally, we give a construction of glissandos at the cycle level in section 5.

The techniques explored in this paper grew out of monophonic cycle modeling with splines, and turned in the direction of polyphonic cycles during the collaboration between mathematician and musician, which resulted in this paper. Introducing polyphony into the timbre of tones at first seemed out of line with the original intent, but quickly turned into a rich playground for generative sound.

2. SPLINE MODELING OF CYCLES

Previous use of splines in audio synthesis and f_0 model envelopes, can found in [1] and [2] respectively. We use the techniques described in [3] to model audio segments, having some discernable but approximate fundamental frequency f_0, using splines. One can choose a set of representative cycles, or segments which are approximately repeating, and use interpolation to fill in the intermediate cycles. This works well for instrument samples, where it is possible to construct accurate models of audio by using data from a small number of cycles. The local timbre of a waveform is preserved if the number of points in one cycle matched by the spline (called interpolation points) is approximately one third of the number of samples in the cycle. For more details on spline modeling, see [4].

An important point is that cycles are modeled over time intervals given as real numbers, or floats. This means that exact values of fundamental frequency f_0 correspond to exact cycle lengths $1/f_0$. Since we compute with sampled waveforms, it will be assumed that sampling can be treated as independent of f_0 and cycle length. The spline function used to model a cycle can be specified to match certain interpolation points coming from a real waveform. In this case we treat the sampled waveform as a continuous function by using linear interpolation between sample values, so that selecting say n evenly spaced points along a cycle can also be independent of the placement of samples. Once a spline model of a cycle is determined, as a set of B-spline coefficients, it can then be used on a cycle of any length, appropriately resampled.

Another important point is that even though cycles are initially thought of as representing waveforms of some constant f_0, this changes when we do cycle interpolation, or when we vary pitch as in glissandos. Whether we are modeling recorded sounds or synthesizing new sounds, or a mixture of these two, the cycle modeling and cycle interpolation methods thrive in the context of waveforms with constantly varying pitch and timbre.

3. POLYPHONIC CYCLES

3.1 Defining Polyphonic Cycles

Polyphonic cycles are meant to represent audio waveforms which have multiple fundamental frequencies, as opposed to a monophonic cycle having a single fundamental frequency. For the purposes of this paper, we define a polyphonic cycle as one which consists of a mix (or average of values) of several constituent waveform cycles, each of which is generated by a set of B-spline coefficients. If the duration (or length) of the polyphonic cycle is L, and two constituent waveform cycles have lengths L_1 and L_2, then we also require that L is a common multiple of L_1 and L_2. With these requirements it is possible to mix the polyphonic cycle with an exact integer number of cycles of each of the constituent lengths. If the polyphonic cycle is then repeated as a waveform, it contains not only prominent higher frequencies corresponding to cycles L_1 and L_2, but also frequency components (elements of timbre) corresponding to the constituent waveforms. So the data required to write a polyphonic cycle amounts to:

- time interval of length L (polyphonic cycle length)
- subintervals of length L_i, $i = 1, \ldots, m$ (constituent cycles)
- integers $k_i = L/L_i$
- B-spline coefficients $c_{i,j}$, $j = 1, \ldots, n$ for each constituent cycle

Associated to the cycle lengths in the above data are fundamental frequencies $f_0 = 1/L$ and constituent fundamental frequencies $f_{0,i} = 1/L_i$. A polyphonic cycle can then be realized as a buffer of audio samples over the time interval L. This set of audio samples can then also be resampled with a new spline model, giving a compressed representation of the polyphonic cycle as a vector of B-spline coefficients. The polyphonic cycle becomes more interesting if the values of $f_{0,i}$ have some harmonic relationships. We explore one example (among many possibilities) in the next section, in particular where the $f_{0,i}$, $i = 1, 2, 3, 4$, form the pitch sequence of a seventh chord.

3.2 Seventh Chords

We describe here certain types of seventh chords used to construct polyphonic cycles. These chords can be thought of as Just Intonation versions of traditional seventh chords and inversions of such chords having four pitches, or fundamental frequencies, contained within one octave. Such a chord is most simply represented by the well-known *harmonic seventh* which is a dominant seventh chord in root position with notes having fundamental frequencies $4f_0$, $5f_0$, $6f_0$, $7f_0$, or equivalently having ratios between pitches given as $5/4$ (major third), $6/5$ (minor third), and $7/6$ (minor third). The name, of course, comes from the fact that these frequencies occur naturally as harmonic partials of the fundamental f_0. The cent values of the intervals, or frequency ratios, are approximately 386.3, 315.6, and 266.9 (where 100 equals one equal-tempered semitone with frequency ratio $2^{1/12}$). Compared to Equal Temperament, these intervals are a flat major third and a sharp minor third, by about 15 cents each, followed by a very flat minor third, by about 33 cents. Such a harmonic seventh chord will produce virtually no audible beat patterns amongst the overtones or harmonic partials of the four tones, and can occur in barbershop quartet music as a preferred form of dominant seventh chord.

This type of seventh chord is also useful in generating polyphonic cycles since the ratios between frequencies correspond to simple ratios between cycle lengths. Then to create one polyphonic cycle of length $1/f_0$ we simply write into a cycle of length $1/50$ sec exactly 4 cycles of the first tone, 5 of the second, 6 of the third, and 7 of the fourth. Mixing these at the cycle level forms one cycle of the harmonic seventh chord. Note that this simple process is not possible with an equal-tempered seventh chord since the number of cycles for each tone will not be an integer value. Note also that we are not using sinusoids as the tones representing each fundamental frequency $f0_i$, rather we are using a full-spectrum tone with one predetermined cycle of some exact length generated from a spline model. It is possible to use different spline models for each of the four cycles which are derived from instrument tones or synthesized tones.

In table 1 summarize the harmonic seventh chord and its inversions. In each line the sequence of scalar values is obtained from the previous line by multiplying the lowest value by 2, having the effect of moving the lowest pitch up an octave. The last column is the *semitone separation* type, or S-type, which represents the 3 intervals between successive pitches in semitones.

f_0 scalars	Cent values	S-type
$4 \cdot 5 \cdot 6 \cdot 7$	$386.3 - 315.6 - 266.9$	$[4, 3, 3]$
$5 \cdot 6 \cdot 7 \cdot 8$	$315.6 - 266.9 - 231.2$	$[3, 3, 2]$
$6 \cdot 7 \cdot 8 \cdot 10$	$266.9 - 231.2 - 386.3$	$[3, 2, 4]$
$7 \cdot 8 \cdot 10 \cdot 12$	$231.2 - 386.3 - 315.6$	$[2, 4, 3]$

Table 1. Harmonic Seventh Chord Ratios and Intervals

One can find many other seventh chord representations using small integer sequences. For example, all of the 31 seventh chord types in the constraint-based system of seventh chords introduced in [5] can be done in this way. For example, in table 2 we use the similar sequence $5 \cdot 6 \cdot 7 \cdot 9$ to form the root position half-dimished seventh chord, and in table 3 we use the sequence $6 \cdot 7 \cdot 9 \cdot 11$ to form the root position of a minor seventh chord.

f_0 scalars	Cent values	S-type
$5 \cdot 6 \cdot 7 \cdot 9$	$315.6 - 266.9 - 435.1$	$[3, 3, 4]$
$6 \cdot 7 \cdot 9 \cdot 10$	$266.9 - 435.1 - 182.4$	$[3, 4, 2]$
$7 \cdot 9 \cdot 10 \cdot 12$	$435.1 - 182.4 - 315.6$	$[4, 2, 3]$
$9 \cdot 10 \cdot 12 \cdot 14$	$182.4 - 315.6 - 266.9$	$[2, 3, 3]$

Table 2. Half-Diminished Seventh Chord Ratios and Intervals

f_0 scalars	Cent values	S-type
$6 \cdot 7 \cdot 9 \cdot 11$	$266.9 - 435.1 - 347.4$	$[3, 4, 3]$
$7 \cdot 9 \cdot 11 \cdot 12$	$435.1 - 347.4 - 150.6$	$[4, 3, 2]$
$9 \cdot 11 \cdot 12 \cdot 14$	$347.4 - 150.6 - 266.9$	$[3, 2, 3]$
$11 \cdot 12 \cdot 14 \cdot 18$	$150.6 - 266.9 - 435.1$	$[2, 3, 4]$

Table 3. Minor Seventh Chord Ratios and Intervals

Finally, we also have a full-diminished seventh chord in table 4. In this case, each of the inversions is an approximation of an equal-tempered full-diminished seventh chord with same S-type $[3, 3, 3]$, but in this Just Intonation context the inversions are not the same.

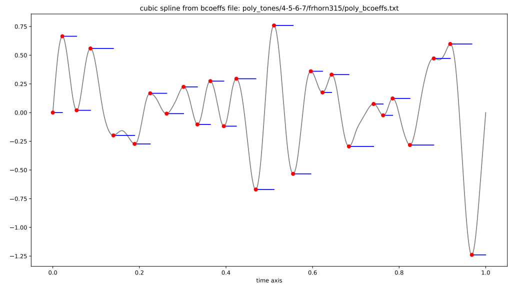
To attain other familiar seventh chords such as Major Seventh, inversions require the appearance of voices separated by one semitone, which requires larger consecutive integer values such as 15, 16.

It is interesting to also go in the opposite direction and consider sequences which do not represent good approx-

f_0 scalars	Cent values	S-type
$12 \cdot 14 \cdot 17 \cdot 20$	$266.9 - 336.1 - 281.4$	$[3, 3, 3]$
$14 \cdot 17 \cdot 20 \cdot 24$	$336.1 - 281.4 - 315.6$	$[3, 3, 3]$
$17 \cdot 20 \cdot 24 \cdot 28$	$281.4 - 315.6 - 266.9$	$[3, 3, 3]$
$10 \cdot 12 \cdot 14 \cdot 17$	$315.6 - 266.9 - 336.1$	$[3, 3, 3]$

Table 4. Full Diminished Seventh Chord Ratios and Intervals

Figure 1. polyphonic cycle



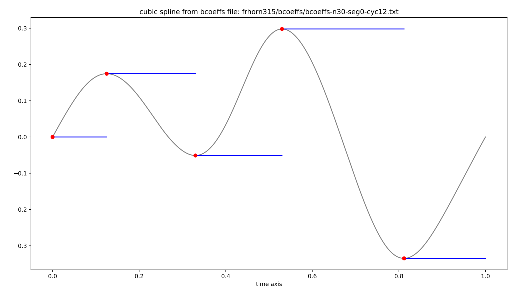
imations of any traditional seventh chords. For example, perhaps the simplest case is $6 \cdot 7 \cdot 8 \cdot 9$. This chord has spread from low to high tones given by a Just Perfect Fifth with ratio $3/2$, and has S-Type $[3, 2, 2]$ with cent values of successive intervals 266, 231, and 203, coming from the ratios $7/6$, $8/7$, and $9/8$ respectively. This could be described as a Just version of an eleventh chord.

4. MELODIC CONTOURS

Polyphonic cycles tend to have more variation in the sample values over one cycle than a monophonic cycle, say derived from an instrument sample with one fundamental frequency. If we use a polyphonic cycle as a melodic contour, we thus have a melody with more variation in pitch than one derived from a monophonic cycle. One way to measure variation is simply to count the number of zero crossings. For basic definitions and techniques of melodic contours derived from cycles of waveforms, see [6].

In figure 1 we show a polyphonic cycle generated from the monophonic cycle shown in figure 2. This monophonic cycle is modeled with a spline based on a recorded sample of french horn. It is used four times to generate each

Figure 2. monophonic cycle



of the constituent cycles at four distinct pitches. Those four pitches have the ratios of the harmonic seventh chord $4 \cdot 5 \cdot 6 \cdot 7$. So we write into one interval representing the polyphonic cycle first 4 cycles, then 5, then 6, then 7, then mix these into one polyphonic cycle waveform. The polyphonic cycle is then sampled again to obtain a spline model which is graphed in figure 1. We also plot stationary points on the graph to indicate one method of melodic contour generation. The melodic contour is sampled to get pitches as well as durations of notes. Each line segment on the graph indicates a pitch based on its y-coordinate, and a duration based on its length. The scale of x and y axes determines the pitch range and duration range. For example we may use $-1 \leq y \leq 1$ to determine pitches or fundamental frequencies $2^y f_0$ ranging an octave above or below some central chosen pitch f_0.

5. GLISSANDOS

In this section we present some algorithms to generate glissandos with exact integer numbers of cycles fitting into a prescribed time interval. These techniques were designed while working on polyphonic cycles, but they apply equally to any cycle type.

Glissandos are created typically using pitch-shifting algorithms, using FFT (Fast Fourier Transform) to work with the signal in the frequency domain. In the context of cycle modeling, we work primarily in the time domain but we keep track of fundamental frequency which is represented locally as the inverse of cycle length. This approach lends itself well to the problem of creating a glissando, or monotonic and continuous (increasing or decreasing) pitch change over a small time interval. We restate the problem using cycle length as follows:

Glissando Generation Problem: Given a time interval L, starting cycle length c_0 and ending cycle length c_1, with $c_0 + c_1 < L$, find a positive integer k and a sequence of cycle lengths $L_1, \ldots, L_k$ satisfying:

- the sequence $c_0, L_1, \ldots, L_k, c_1$ is strictly monotone
- $c_0 + \sum_{i=1}^{k} L_i + c_1 = L$.

Apart from a few cases which are not solvable (for instance $c_1 < c_2$ and $L - c_1 - c_2 < c_1$) we will assume that a solution exists and find several methods to find approximate and exact solutions. First, we assume that $L' = L - c_1 - c_2$ is reasonably large compared to c_1 and c_2, and a solution k exists. We will also let $L_0 = c_1$ and $L_{k+1} = c_1$. Finding an approximate solution using linear growth rate is a good first step. We can do this by first supposing that $L_1 = L_2 = \cdots = L_k = \frac{c_1 + c_2}{2} = a$, and that k is the floor (greatest integer) of the real number $k' = L'/a$. Clearly this is not a solution, since the sequence L_i is not monotone. Further, the sum is only approximate. To see this, let x be the fractional part of k', so that $k' = k + x$, and observe:

$$c_0 + \sum_{i=1}^{k} L_i + c_1 = c_0 + (k' - x)\frac{L'}{k'} + c_1 \qquad (1)$$
$$= L - ax$$

So the sum of all cycle lengths is $L - ax$. Next, we adjust the values of cycle lengths to linearly interpolate between c_0 and c_1. The transformation is most easily described by taking the k points $(\frac{i}{k+1}, a)$, $i = 1, \ldots, k$, and projecting them onto the line $(1-t)c_0 + tc_1$ which passes through $(0, c_0)$, $(1, c_1)$, and $(\frac{1}{2}, a)$. Call the new points $(\frac{i}{k+1}, y_i)$, where

$$y_i = \frac{k+1-i}{k+1}c_0 + \frac{i}{k+1}c_1.$$

By symmetry, these projected points have y-coordinates which still add up to ka, so we take the adjusted cycle lengths to be $L_i = y_i$ for $i = 1, \ldots, k$, $L_0 = c_0$, and $L_{k+1} = c_1$. These values are monotonic and have sum equal to $L - ax$.

In summary, the linear approximation to the above Glissando Generation Problem is given as

- $a = \frac{c_0+c_1}{2}$, $L' = L - c_0 - c_1$
- $k' = L'/a$
- $k = floor(k')$, $x = k' - k$
- $L_0 = c_0$, $L_{k+1} = c_1$
- $L_i = \frac{k+1-i}{k+1}c_0 + \frac{i}{k+1}c_1$, $i = 1, \ldots, k$
- $\sum_{i=0}^{k+1} L_i = L - ax$

Next, we construct a quadratic curve which gives an exact solution to the glissando problem. We begin with the same values a, k' and k, and the points $(\frac{i}{k+1}, a)$, $i = 1, \ldots, k$. But now we project these points onto the quadratic Bezier curve

$$q(t) = c_0(1-t)^2 + (a+\delta)2(1-t)t + c_1t^2.$$

This curve has the properties: $q(0) = c_0$, $q(1) = c_1$, and $q(\frac{1}{2}) = a + \frac{1}{2}\delta$. If $\delta = 0$ the curve contains the point $(\frac{1}{2}, a)$ and hence is equal to (a quadratic parametrization of) the line $(1-t)c_0 + tc_1$. If $\delta > 0$ then the curve will be above this line and we can use δ as a free parameter to force the sum of the $L_i = q(\frac{i}{k+1})$ to equal L.

After some simplification, we arrive at:

$$\delta = \left(\frac{L}{k+2} - a\right)\left(1 - \frac{2k+3}{3k+3}\right)^{-1}.$$

6. EXAMPLES

In our accompanying composition "SplineKlang" the first melodic fragment uses a melodic contour from one cycle and timbre from a mix of polyphonic and monophonic cycles. The mix of cycles are generated from instrument samples, such as clarinet, as well as polyphonic cycles. It is necessary to use the same spline dimension in each of these cycles in order to interpolate between them for the duration of a tone. This technique has been previously used primarily to generate tones which model instrument samples quite closely. In that case, the cycles are called *key cycles*, which serve a similar role as *key frames* in animation.

7. CONCLUSIONS AND FUTURE WORK

We have found that introducing polyphony into the timbre of tones on the very short time scale of one cycle opens up a wealth of possible timbres which can coexist with timbres based on instrument samples. The polpyphonic cycles constructed in this paper are based on seventh chords, which are justly tuned, but there are many other forms to explore. In addition to these forms, further experimentation in the mix of cycles used to construct the constituent cycles, such as coming from different instrument models, can also be considered. Variations on the cycle-based construction of glissandos are also under development. For example, one can use cubic and quartic polynomials to enable the growth rate of cycles to be close to zero at the ends and steeper in the middle, by setting the derivative to zero at the ends.

8. REFERENCES

[1] Collins, N.: *SplineSynth: An Interface to Low-Level Digital Audio*. Proceedings of the Diderot Forum on Mathematics and Music, Vienna, 1999, ISBN 3-85403-133-5, pp 49-61.

[2] Ardaillon, L., Degottex, G., Roebel, A.: *A multi-layer F0 model for singing voice synthesis using a B-spline representation with intuitive controls*. Interspeech 2015, Sep 2015, Dresden, Germany. hal-01251898v2.

[3] Klassen, M.: *Spline Modeling of Audio Signals and Cycle Interpolation*, Mathematics and Computation in Music, MCM 2022, Springer, LNAI 13267 (Lecture Notes in Artificial Intelligence), Montiel, M. et al, eds., pages 155-167.

[4] de Boor, C.: *A Practical Guide to Splines*, revised edition, Springer-Verlag, New York (1980)

[5] Klassen, M.: *Constraint-Based Systems of Triads and Seventh Chords, and Parsimonious Voice-Leading*, Mathematics and Computation in Music, MCM 2019, Springer, LNAI 11502 (Lecture Notes in Artificial Intelligence), Montiel, M. et al, eds., pages 185-197.

[6] Klassen, M., Lanthier, P.: *Melodic Contour Generation with Spline Models of Cycles*, Mathematics and Computation in Music, MCM 2024, Springer, LNCS 14639 (Lecture Notes in Computer Science), Noll, T. et al, eds., pages 210-222.