ESTIMATING PARAMETERS FROM AUDIO FOR AN EG+LFO MODEL OF PITCH ENVELOPES

Stephen J. Welburn, *

Centre for Digital Music,
Queen Mary University of London,
London, United Kingdom
stephen.welburn@elec.qmul.ac.uk

Mark D. Plumbley,†

Centre for Digital Music,
Queen Mary University of London,
London, United Kingdom
mark.plumbley@elec.qmul.ac.uk

ABSTRACT

Envelope generator (EG) and Low Frequency Oscillator (LFO) parameters give a compact representation of audio pitch envelopes. By estimating these parameters from audio per-note, they could be used as part of an audio coding scheme. Recordings of various instruments and articulations were examined, and pitch envelopes found. Using an evolutionary algorithm, EG and LFO parameters for the envelopes were estimated. The resulting estimated envelopes are compared to both the original envelope, and to a fixed-pitch estimate. Envelopes estimated using EG+LFO can closely represent the envelope from the original audio and provide a more accurate estimate than the mean pitch.

1. INTRODUCTION

Attempts to reproduce audio based on parameter estimation have a long history particularly regarding FM synthesis [1, 2] and wavetable synthesis [3, 4]. In general, these have looked at directly reproducing audio by matching spectral content.

We consider parameter estimation as part of an object-based coding of audio. Object coding of audio analyses a piece of audio to estimate parameters for synthesis objects. Driving the objects with the parameters allows an approximation of the original audio to be created. It is a form of analysis/synthesis encoding. Traditionally, it has been regarded as a technique for low bit-rate encoding and has been examined in relation to sum-of-sinusoids models[5] and instrument models [6]. We seek to produce a high quality representation of audio using similar techniques.

Rather than defining new standards for the encoding, we are looking to use MIDI encoded parameters for a Downloadable Sounds (DLS)[7] based synthesiser. DLS is a sample-based synthesis engine in which base samples are manipulated by modifying pitch, amplitude and timbre using modulators such as filters, envelope generators (EGs) and low frequency oscillators (LFOs) . DLS is one of the most prevalent synthesis standards being integrated in Microsoft Windows, Apple Mac OS X and mobile devices.

Using standard MIDI controllers, the MIDI data rate restricts the speed at which features can be directly modified, e.g. using expression or pitch bend. Additionally, a large number of individual changes must be stored. However, by specifying parameters before playing a note, DLS synthesis supports indirect modification of amplitude and pitch using EGs and LFOs. These modulators can be applied by the synthesis engine at the full sample rate and are defined using a small number of parameters. With few parameters *per-note*, we have a compact representation for pitch and amplitude envelopes for use in audio coding that is also compatible with existing synthesis techniques and standards.

In this paper, we propose to use de Cheviegné's YIN algorithm[8], to estimate the pitch envelope from a piece of audio. Individual notes' envelopes can then be modelled using the DLS parameters.

We look at representing the pitch envelope using the basic parameters available via DLS.

2. BACKGROUND

2.1. DLS Envelope Parameters

Using the default modulation routings, the DLS standard[7] provides an EG and a LFO for pitch modulation and a separate EG and LFO for amplitude modulation.

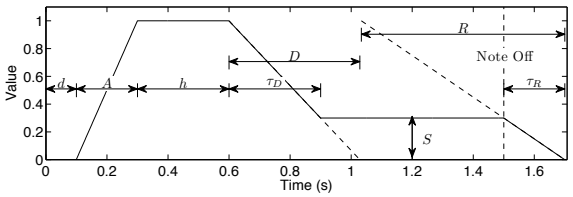

Figure 1: *6-stage Envelope Generator.*

The EGs consist of six stages (Figure 1) and are parameterized by:

- Delay time, d, during which time the EG output is zero;
- Attack time, A, the time taken to reach an output level of 1;
- Hold time, h, the time the EG stays at the peak level;
- Decay rate, D, the time it would take to decay from the peak level to a level of zero;
- Sustain level, S indicates the level (0 to 1) at which the sustain phase is held;
- Release rate, R, the time it would take to decay from 1 to a level of zero (the actual time spent in the release phase τ_R depends upon the sustain level).

Hence, such envelopes can be referred to as $dAhDSR$ envelopes. Additionally, an envelope depth d_{EG} will be associated with the envelope indicating the peak value. It is noted that the MIDI "note off" event occurs at the start of the release phase.

* Stephen J. Welburn is supported by an EPSRC Doctoral Training Account at Queen Mary, University of London.

† Mark D. Plumbley is supported by an EPSRC Leadership Fellowship.

The actual time spent in the decay phase, τ_D, depends upon the sustain level:

$$\tau_D = (1 - S) \times D \,. \tag{1}$$

Similarly, for the time spent in the release phase, τ_R:

$$\tau_R = S \times R \,. \tag{2}$$

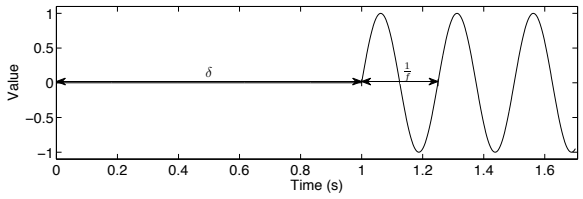

Figure 2: *Low Frequency Oscillator.*

Each low frequency oscillator (Figure 2) is parameterized by:

- Delay time, δ, during which time the LFO output is zero;
- Oscillator frequency, f.

Additionally, an envelope depth d_{LFO} will be associated with the envelope indicating the peak value.

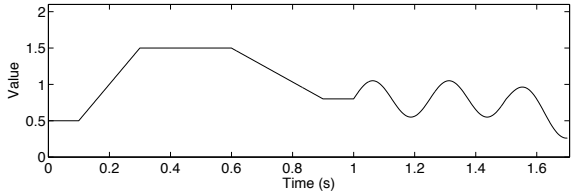

Figure 3: *Envelope.*

Combining the envelope generator, the LFO and a base value with the relevant depths, can then produce an overall envelope (Figure 3)

We seek to minimize the difference between a pitch envelope and the EG+LFO estimate. To do so, we specify an objective function that quantifies this difference.

2.2. Objective Function

The objective function used is based on the root-sum-square-error between the pitch envelope and the EG+LFO estimate. However, to allow comparisons between the errors for different envelopes, we normalise this error, giving as the objective function:

$$f_{est}(k) = \sqrt{\frac{\sum_i (e_i - g_i)^2}{\sum_i e_i^2}} \tag{3}$$

where $\mathbf{e} = (e_1, \ldots, e_n)$ is the envelope given by YIN and $\mathbf{g} = (g_1, \ldots, g_n)$ is the EG+LFO estimate. The use of a relative error function agrees with previous parameter estimation work [2].

We want to estimate the envelope parameters which minimize the objective function. The DLS specification defines the EG and LFO parameters in terms of bit-wise representations, we have therefore adopted a bit-wise optimization scheme to find a solution.

2.3. CHC

Estimating the pitch envelope parameters is a difficult problem, featuring local minima (e.g. each time the LFO delay is changed by the period of the LFO). Evolutionary algorithms are effective at solving such problems, hence we use Eshelman's CHC[1] algorithm [10] to estimate parameters.

CHC algorithm is based on a population of candidate solutions. The candidate solutions are randomly paired off and each pair of possible solutions is then compared. If their Hamming distance is large enough, then half of the differing bits in the parents are exchanged to produce a pair of children (*Half Uniform Crossover*, HUX) otherwise they produce no children. From the combined population of parents and children, the solutions that give the best values for the objective function are adopted as the new population of candidate solutions. If the entire population is too similar to produce any children for several generations, then a new population is generated by mutating the current best candidate solution (referred to as *Cataclysmic Mutation*). We note that the number of children produced each generation is not fixed.

Features of the CHC algorithm include:

- the CHC algorithm is *elitist*, always keeping the best result. Each generation is therefore guaranteed to be at least as good as the previous one;
- using HUX and Cataclysmic mutation, CHC preserves diversity even though it operates on a small population (usually 50 individuals);
- using a small population reduces the number of objective function calculations required each generation, hence improving performance;
- an algorithm including random mutation allows the entire search space to be examined.

The DLS standard includes data format definitions (pp. 32-34) specifying 32-bit representations for units of pitch, time, gain and frequency. These bit-wise representations form the basis of our encoding of EG and LFO parameters for use with CHC.

3. METHOD

59 files from the RWC Musical Instrument database [11] were used as test data for the parameter estimation. The files covered various instruments (piano, organ, violin, trumpet and clarinet), with multiple articulations and dynamics (see Table 1). Using YIN[8], pitch, power and aperiodicity were estimated for each file. Pitch envelopes, $\mathbf{e}$, were then extracted for 3,929 individual notes using $aperiodicity < 0.1$ to determine the pitched portions of the audio.

Splitting the $dAhDSR$ EG into two 4-stage segments ($dAhD$ and $(d{+}A{+}h)DSR$) allows the overall envelope to be represented as a linear combination of four components:

- a constant contribution of the base pitch value;
- a contribution from the $dAhD$ envelope, $eg1_i$;
- a contribution from the $(d + A + h)DSR$ envelope, $eg2_i$, i.e. the EG sustain depth;
- the LFO depth, l_i.

[1] CHC stands for "Cross generational elitist selection, Heterogeneous recombination and Cataclysmic mutation" according to Whitley[9]

i.e:

$$\begin{pmatrix} g_1 \\ \vdots \\ g_i \\ \vdots \\ g_n \end{pmatrix} = \begin{pmatrix} 1 & eg1_1 & eg2_1 & l_1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & eg1_i & eg2_i & l_i \\ \vdots & \vdots & \vdots & \vdots \\ 1 & eg1_n & eg2_n & l_n \end{pmatrix} \begin{pmatrix} b \\ d_{EG1} \\ d_{EG2} \\ d_{LFO} \end{pmatrix} \quad (4)$$

where $\mathbf{g} = (g_1, \ldots g_n)$ is the overall envelope estimate. The base pitch of the note is then b; the EG depth is $d_{EG} = d_{EG1}$; the sustain depth is d_{EG2}; the sustain level of the envelope, $S = \frac{d_{EG1}}{d_{EG2}}$; and the LFO depth is d_{LFO}.

Minimizing the objective function (Equation 3) minimizes the sum-square-error between the envelope estimate, $\mathbf{g}$, and the YIN data, $\mathbf{e}$. We can calculate values for b, $eg1_i$, $eg2_i$ and d_{LFO} that minimize this error from equation 4 - given the time-based parameters for the EG and LFO (i.e. $d, A, h, \tau_D, \tau_R, \delta, \frac{1}{f}$).

Within CHC, the time-based parameters were evolved, and the best-fit pitch parameters (base, EG depth, sustain level, LFO depth) were calculated for the time values. The objective function was then calculated based on the full set of time and pitch parameters. CHC was run for 10,000 iterations using the standard population size of 50 individuals. The initial population was random.

The resulting envelopes were then compared with the original data, and with the objective function value given using the mean pitch estimate from YIN.

4. RESULTS

Using the pitch estimate from YIN, we can calculate the mean pitch for note k, μ_k:

$$\mu_k = \frac{1}{n} \sum_{i=1\ldots n} e_i \quad (5)$$

where $\mathbf{e} = (e_1, \ldots, e_n)$ is the envelope given by YIN.

The accompanying graphs (Figures 4 to 10) plot the YIN difference $e_i - \mu_k$ (paler, background lines) and the difference between the mean YIN pitch and the generated envelope, $g_i - \mu_k$ (the bolder smoother lines) for several notes from each instrument, including both vibrato and non-vibrato trumpet and violin. Good approximations of the envelope shapes were found for many notes and vibrato phase and frequency were matched.

As an alternative to using the EG+LFO envelope estimate, a constant pitch estimate pitch p_c can be used. We can then calculate the objective function using this pitch:

$$f_{const}(p_c, k) = \sqrt{\frac{\sum_i (e_i - p_c)^2}{\sum_i e_i^2}} \quad (6)$$

where $\mathbf{e} = (e_1, \ldots, e_n)$ is the envelope given by YIN. The mean pitch value μ_k is the constant pitch value that minimizes this objective function. We let $f_{est} = f_{const}(\mu_k, k)$.

Figure 11 shows a histogram of $\frac{f_{est}}{f_{mean}}$ for the notes processed. A large number of notes are seen to have a small improvement (ratio 0.1 to 1) but there were also notes where the estimate was 10 to 100 times worse than using the flat mean (418 notes having ratio > 10).

Of the 3,929 notes processed, 2,731 (69.51%) had a better fit using EG+LFO than by simply using the mean value (see Table 1).

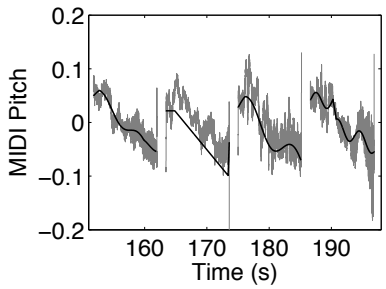

Figure 4: *Piano (with pedal) pitch estimation.* The paler plot is the YIN pitch envelope, the darker line the EG+LFO estimate. The piano pitch estimates show a similar decrease in pitch for each note. The EG+LFO estimate provides a smoothed approximation of the envelope bearing little resemblance to the "basic" EG+LFO envelope in Figure 3. It is notable that the YIN piano pitch variation is mainly in the ± 5 cent range (1 cent = 0.01 semitones).

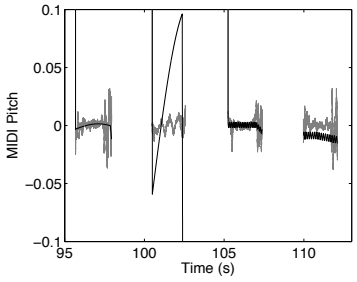

Figure 5: *Organ pitch estimation.* The YIN estimates of pitch of the pipe organ notes show less variation than the hammered-string sounds of the piano. Artifacts from the YIN output produce pitch drops at the start of each envelope, and, in matching this artifact, the second note has been unable to find a good approximation of the pitch envelope in the $10,000$ generations CHC processed.

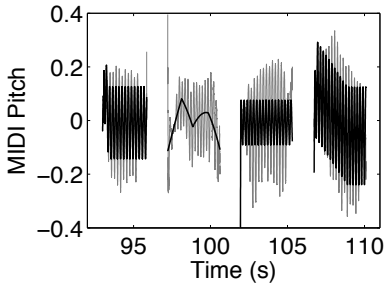

Figure 6: *Violin pitch estimation.* The standard violin articulation shows distinct vibrato. For three of the four notes, both the phase and the frequency of the vibrato has been closely matched. In addition, for these notes, the range of the vibrato is within the original envelope. For the second note, gross features of the envelope (the initial increase, the central dip, and the final fall) were reasonably matched.

Table 1: *Instrument Summary.*

Instrument	Number of files	Number of articulations	Dynamics	Number of notes, $\hat{n}$	$\bar{f}_{est}$	$\bar{f}_{mean}$	% notes with $f_{est} < f_{mean}$
Piano (011)	12	4	P/M/F	2080	0.0202	0.0200	73.4
Organ (061)	8	8	M	382	0.0165	0.0278	83.0
Violin (151)	2	2	M	128	0.0080	0.0070	68.0
Trumpet (211)	25	9	P/M/F	813	0.0186	0.0107	56.8
Clarinet (311)	12	4	P/M/F	526	0.0077	0.0038	64.2
Total	59	n/a	n/a	3929	0.0175	0.0163	69.5

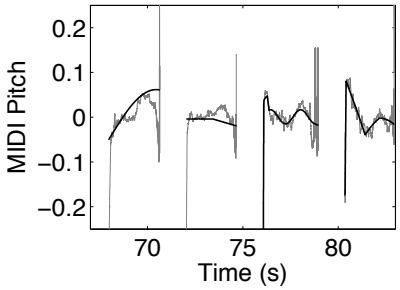

Figure 7: *Violin pitch estimation (no vibrato).* Violin notes were also processed from samples without vibrato (Figure 7). Similarly to the piano notes, a range of approximately ± 5 cents covers most of the variation. For the first two notes, little of the original envelope is captured, but for the other two notes a close match is seen between the YIN envelope and our estimate.

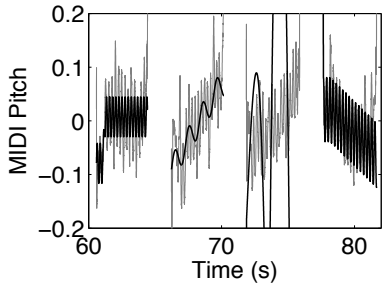

Figure 9: *Vibrato trumpet pitch estimation.* For vibrato trumpet notes (Figure 9) we again managed to match frequency and phase in some notes. Additionally, the major features of the envelopes were also represented (low then higher for the first note, decreasing for the fourth). However, for the second note the LFO was not matched, and for the third note CHC completely failed to find a reasonable estimate of the envelope in $10,000$ generations.

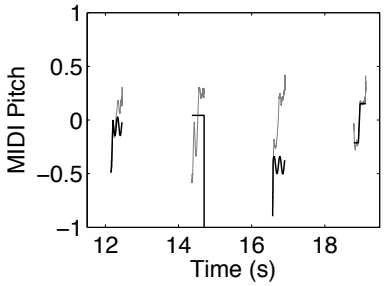

Figure 8: *Staccato trumpet pitch estimation.* Staccato trumpet notes (Figure 8) offer little data from which to estimate parameters. None of the four notes capture the rise in pitch thoughout the note, although the fourth example succesfully matches the two plateaux in the pitch. The first and third notes, however, apply the LFO to match the peaks at the start of and half-way through the envelope.

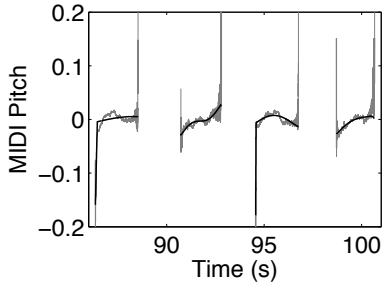

Figure 10: *Clarinet pitch estimation.* The final instrument processed was the clarinet (Figure 10). The pitch variation is less than 5 cents apart from at the note ends and the overall shape of the pitch envelopes is reasonably represented.

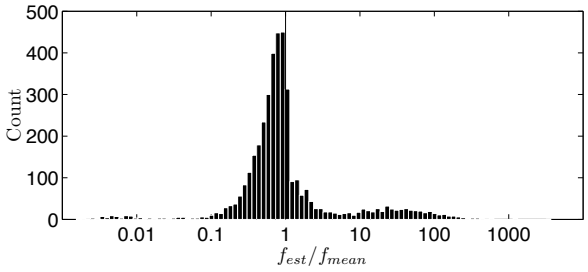

Figure 11: *Histogram of ratio of objective function values using EG+LFO estimate and flat mean.* If $\frac{f_{est}}{f_{mean}} < 1$ then the EG+LFO estimate is better than the flat mean

Allowing CHC to run for more generations will allow it to further explore the solution space and would increase the likelihood of finding good solutions.

Table 1 summarises the results by instrument, indicating the numbers of files processed and their breakdown according to articulation and dynamics. The total number of notes processed for each instrument, $\hat{n}$, is given as well as the mean f_{est}:

$$\bar{f}_{est} = \frac{1}{\hat{n}} \sum f_{est} \tag{7}$$

and mean f_{mean}:

$$\bar{f}_{mean} = \frac{1}{\hat{n}} \sum f_{mean} \ . \tag{8}$$

Although a better fit was found for most notes using EG+LFO (i.e. $f_{est} < f_{mean}$), the $\bar{f}_{est}$ is greater than the $\bar{f}_{mean}$ for all instruments except the organ. This agrees with Figure 11 having many notes for which EG+LFO gave a small improvement over using the mean, and a smaller number of notes with for which $f_{est} \gg f_{mean}$.

In more detail, only 19 of the 59 files processed had a $\bar{f}_{est}$ less than the $\bar{f}_{mean}$. However, in 10 files, the EG+LFO estimate was better for over 80% of notes. For some trumpet files, low proportions ($25\% - 40\%$) of notes had $f_{est} < f_{mean}$ i.e. for most notes using the mean value gave a better envelope estimate than EG+LFO. Further investigation is required as to how this relates to the note articulations and whether these results are a consequence of the YIN output or the envelope estimation procedure.

5. CONCLUSIONS

Using a EG+LFO estimate of the pitch envelope, we can produce similar envelopes to those estimated from the audio, using the LFO to represent both vibrato effects and more complex non-vibrato envelopes. This envelope can more closely approximate the original audio than simply applying a constant pitch and represents the envelope in a small number of parameters per note.

With vibrato examples, both frequency and phase can be matched (as in Figures 6 and 9). However, with more complex pitch variation, the minimum error found may use the LFO to represent coarser features than the vibrato and omit representing any vibrato that is present.

The rate at which the EG and LFO modulators operate is defined by the synthesiser used, allowing full sample-rate signal modulation based on parameter settings provided via MIDI. Per-note parameters are suitable when the outcome is known at the start of the note (e.g. for audio coding), but inappropriate for performance when the modulators need to be varied during the note. However, having initialised parameters per-note, they can be adjusted.

A similar procedure to that given for pitch can be followed for estimating amplitude envelopes using EG+LFO, e.g. based on the power figures from YIN. From the pitch and amplitude parameters found, it will be possible to use a DLS based synthesiser to create audio matching the original pitch and amplitude. Our next work will therefore involve estimation of amplitude envelopes; synthesis of audio from the parameters found; and listening tests to evaluate the subjective quality using the envelope estimates vs. the original sounds.

6. REFERENCES

[1] J. W. Beauchamp, "Synthesis by amplitude and "brightness" matching of analyzed musical instrument tones," in *69th Convention of the Audio Engineering Society*. AES, 1981.

[2] T.J. Mitchell and D.P. Creasey, "Evolutionary sound matching: A test methodology and comparative study," Dec. 2007, pp. 229–234.

[3] A. Horner, J. Beauchamp, and L. Haken, "Methods for multiple wavetable synthesis of musical instrument tones," *Journal of the Audio Engineering Society*, vol. 41, no. 5, pp. 336 – 356, May 1993.

[4] S. Wun and A. Horner, "Evaluation of iterative methods for wavetable matching," *Journal of the Audio Engineering Society*, vol. 53, no. 9, pp. 826–835, September 2005.

[5] E. Vincent and MD Plumbley, "A prototype system for object coding of musical audio," in *Proceedings of 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2005)*. IEEE, 2005, pp. 239–242.

[6] T. Tolonen, *Object-Based Sound Source Modeling*, Ph.D. thesis, Helsinki University of Technology, Espoo, Finland, 2000.

[7] MMA, *Downloadable Sounds Level 2.1*, MIDI Manufacturers Association, PO Box 3173, La Habra, CA 90632-3173. USA, April 2006.

[8] A. Cheveigne and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[9] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.

[10] L.J. Eshelman, "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination," *Foundations of Genetic Algorithms*, 1991.

[11] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Music genre database and musical instrument sound database," in *ISMIR 2003, the 4th International Conference on Music Information Retrieval*, 2003, pp. 229–230.