The evolution of drum modes with strike intensity: Analysis and synthesis using the Discrete Cosine Transform

Tim Kirby[1, a] and Mark Sandler[1]

Centre for Digital Music, Queen Mary University of London, London, UK

(Dated: 16 July 2024)

The synthesis of convincing acoustic drum sounds remains an open problem. In this paper, a method for analysing and synthesising pitch glide in drums is proposed, whereby the discrete cosine transform (DCT) of an unwindowed drum sound is modeled. This is an extension of the scheme initially proposed by [(Kirby and Sandler, 2020)], which was able to reproduce key components of drum sounds accurately enough, that they could not be distinguished from reference samples. Here, drum modes were analysed in greater detail, for a tom-tom struck at 67 different intensities, to investigate their evolution with strike velocity. A clear evolution was observed in the DCT features, and interpolation was used to synthesise modes of intermediate velocity. These synthesised modes were evaluated objectively through null testing, which showed that a continuous blending of strike velocities could be achieved, throughout the dataset. For perceptual evaluation, an AB test was performed with 20 participants. Exactly 50% percent accuracy was achieved overall, which demonstrates that the synthesised samples were deemed to sound as realistic as genuine samples. These results demonstrate that the DCT representation is a valuable framework for analysis and synthesis of drum sounds. It's also likely that this approach could be applied to other instruments.

[a]t.c.kirby@qmul.ac.uk

I. INTRODUCTION

(ISO 4020:2001) In this paper, we reexamine modal behaviour in drums, through a novel framework that incorporates the Discrete Cosine Transform and the Hilbert transform. This representation offers a new perspective on modal oscillation, allowing us to clearly track the evolution of modal oscillation with increasing strike velocity. This analysis is performed on the fundamental mode of a tom-tom. The concept generalises to all resonant modes, though there are diminishing returns in repeating this analysis for other modes, as they share the same form in the DCT.

We can use the knowledge of the DCT representation and its evolution to synthesise drum modes in a dynamic fashion. Interpolation based synthesis is used here, as one example of synthesis, to generate highly accurate simulations of modal behaviour at intermediate strike velocities, for the fundamental mode. The synthesised modes are then evaluated through objective and perceptual means, to validate the accuracy of the synthesised intermediate behaviour.

The interpolation based synthesis technique could be used to augment the number of static samples in a drum library, but more generally, we can work towards a physical model. For example, the DCT representation of the fundamental mode can be trivially translated or scaled to synthesise overtones, and it could even be modeled analytically to create a fully parameterised modal synthesis engine for drum sounds. Future work will investigate the synthesis of full drum sounds from the ground up.

A. Modeling Instruments

The proposed representation could be used to create or enhance Virtual instruments (VI's), which offer many advantages over physical instruments, being employed in both professional mixes and live performances (Collins, 2003), as well as being used by the wider public in educational and recreational settings (Brown, 2014). VI's can also inform research; Toontrack's Superior Drummer 3.0 (SD3) (Toontrack) is used here, and provides a detailed dataset of drum sounds.

Sample-based VI's such as SD3 are currently the most convincing way to replicate the sound of an acoustic drum kit, as they utilise genuine recordings. These libraries are limited, however, by the scope of the sampling process. One is limited to the use of the specific drums that were recorded, each with a finite number of velocity layers and articulations. These restrictions limit both the creative potential of the instrument, the level of expression that can be conveyed, and the resulting realism of the performance.

These limitations do not apply to modelled instruments, however. Modelled instruments can provide continuous control over key parameters for enhanced playability. Subtle variations can also be added to repeated midi notes, to avoid retriggering the exact same sound, which sounds unpleasantly artificial. This is particularly important for drum rolls, which naturally contain a lot of variation. Sample-based recreations of a drum roll can often resemble the monotonous sound of a machine gun; this dreaded phenomena is dubbed the "machine gun effect".

Furthermore, it is possible to model drums of arbitrary specification, providing the user complete control over the dimensions, materials, tunings, and gestures, so they can pursue their unique desired tone. It is even possible to model non-physical situations, such as a gigantic drum, or a membrane with an upwards pitch glide.

While these benefits are clear, synthesising realistic sounds remains an open problem. Many synthesis methods have been used to create drum sounds, with classic hardware synthesisers utilising additive, subtractive, and FM methods (Risset and Wessel, 1982). These sounds can't compete with sample-based drums in terms of realism, but they offer a pleasing alternative, nonetheless. They have been heavily used in contemporary music, and these methods have also carried over to the digital domain.

The most detailed attempt at additive synthesis used filter banks to decompose and model bass drum sounds (Fletcher and Bassett, 1975), but this method provides a fairly static representation of a drum. Drum sounds can also be decomposed into a source filter model, using Linear Predictive coding (LPC) (Sandler, 1990), offering somewhat improved flexibility.

Recently, Generative Adversarial Networks have been employed (Nistal *et al.*, 2020). This network was trained with electronic sounds, so it is hard to know what accuracy could be achieved with acoustic samples. Nonetheless, it does demonstrate the potential of machine learning techniques in this area. However, it is generally difficult to infer what machine learning networks have actually learned, so these techniques are likely to be less physically informative, with results that are less transferable, than techniques that use direct modeling.

Finite difference methods (Bilbao and Webb, 2013), modal synthesis (Avanzini and Marogna, 2010) and the functional transformation method (Marogna and Avanzini, 2009), (Trautmann *et al.*, 2001) have yielded more realistic results, such as those generated in the NESS project (Bilbao *et al.*, 2020), but none that are truly convincing.

Finite difference models have the potential to be further developed, which could help to close this gap, but their computational cost is so high that real-time performance is ruled out for the foreseeable future (Zappi *et al.*, 2017). In contrast, our paper presents the basis for a highly realistic method, with low computational cost, that could be used to create a real time synthesis engine.

B. The Discrete Cosine Transform

This research employs the Discrete Cosine Transform (DCT), which is a well-known signal processing technique (Ahmed *et al.*, 1974). In the context of acoustics, it is generally used for speech processing (Pastiadis and Papanikolaou, 2004) (Ramakrishnan *et al.*, 2015).

The DCT provides a frequency domain representation of a real-valued time domain signal, by expressing it as a sum of cosine functions. The Inverse Discrete Cosine Transform (IDCT) can be then be used to return to the time domain. The default variant (DCT-2) is defined (Rao and Yip, 2014) as:

$$X(k) = \sqrt{\frac{2}{N_s}} k_m \sum_{n_s=0}^{N_s-1} x(n_s) \cos\left(\frac{(2n_s+1)k\pi}{2N_s}\right), \quad k = 0,\ \ldots,\ N_s-1 \tag{1}$$

where $k_m = \frac{1}{\sqrt{2}}$ when $m = 0$ or N, else $k_m = 1$, n_s is the input signal sample number, ranging from 0 to $N_s - 1$, $x(n_s)$ is the the input signal, k is the frequency domain sample number, $X(k)$ is the spectrum of x , and δ_{k1} is the Kronecker delta.

This representation is a single, real-valued component, the DCT magnitude. This contrasts the complex representation generated by the Discrete Fourier Transform (DFT). Any phase information in the input signal is therefore encoded in the DCT magnitude representation. The DCT is equivalent to a DFT of roughly twice the length, operating on real data with even symmetry (Asmara *et al.*, 2017). This equivalence is important here, as the DCT is used instead of the real component of the DFT, which was used in (Kirby and Sandler, 2020), as explained in the following subsection.

C. Relationship to Inverse Fast Fourier Transform synthesis

This research makes use of the concept that underpins Inverse Fast Fourier Transform (IFFT) synthesis. If you can model the spectrum of a sound, you can synthesise it. IFFT synthesis introduced in 1980 (Chambelin, 1980), but has mainly been used as an efficient way of generating large ensembles of sinusoids for additive synthesis (Rodet and Depalle, 1992). These sinusoids have fixed amplitude and frequency within a given window, so the Fast Fourier Transform (FFT) representations that are modelled are still relatively simple.

It is, however, possible to transform an unwindowed audio signal of arbitrary length or complexity. The challenge is that it becomes harder to meaningfully interpret the frequency domain representation for more complicated signals, let alone edit or model them. If we are not dealing with a signal with a well-known Fourier transform, we can only compute the transform and investigate how the information is encoded. This paper demonstrates that entire drum samples transform in an informative manner and can be modelled in full, without the need for windowing.

One of the challenges of IFFT synthesis is that frequency domain representation is complex, so there are two components to interpret in tandem, whether these be the real and imaginary components themselves, or the magnitude and the phase of the signal. In usual procedure, both the real and imaginary components are required to reproduce a signal, so both components would need to be modelled.

However, as audio signals are real, there is degeneracy in this complex representation. In the use case of this research, it was therefore beneficial to instead use the Discrete Cosine Transform (DCT), for conceptual simplicity. This made it possible to simplify the synthesis problem, so that only a single, real-valued, frequency domain signal needs to be modeled.

This method could be referred to as IDCT synthesis. It is equivalent to that described in (Kirby and Sandler, 2020), which used the real component of the FFT, instead of the DCT. But that involved the non traditional discarding of the imaginary component, which was found to make explanations of the the method overly convoluted. It also jarred with peoples conventional understanding of the Fourier transform. Nonetheless, it was found that tom-tom modal oscillations have a common signature in the frequency domain, which encodes their entire time domain activity. This signature will be referred to as a “modal feature”. This paper builds on those results by analysing modal features in much greater detail, to investigate how these features evolve with increasing strike velocity. The key improvements are as follows:

The relevant concepts are discussed more thoroughly with full mathematical definitions. 67 different velocities are used instead of 13, to probe deep into dynamic modal behaviour, and to further validate the underlying concepts. The evolution of the amplitude function is now investigated. The evolution of the phase function is investigated in much more detail, using a more suitable unwrapping, and this is used to analyse how the pitch glide magnitude increases with strike velocity. This more complete analysis is used to inform a dynamic synthesis technique, synthesising modal behaviour at intermediate velocity, rather than the static technique used previously to simply replicate existing modal features. This synthesised audio is then evaluated much more thoroughly via objective evaluation and a listening test that was larger in scale. These improvements combine to make this paper more formalised and deeper in scope than the previous paper.

The paper is organised as follows: Section II provides an overview of the relevant Physics of a tom-tom, which underpins the method. Section III presents the method, detailing the data set of tom-tom samples that were analysed, a mathematical description of the proposed method, a description of the general form of a modal feature, and an explanation of how modal features can be decomposed. Section IV describes the initial analysis of the data set, detailing the evolution of modal features with strike velocity, and explaining how this clear evolution can form the basis of an interpolation based synthesis technique to synthesise modal behaviour at intermediate velocities. Section V describes an objective evaluation of this interpolation method that employs null testing. Section VI describes a listening test that was used to perceptually evaluate these synthesised modes. Section VII provides the results of this listening test, and Section VIII provides concluding remarks.

II. THEORETICAL BASIS

This is an overview of the relevant Physics of a tom-tom, which underpins the method. Tom-toms are composed of a hollow, cylindrical shell, typically between 6-18" in diameter, which can accommodate a membrane at either end. These membranes are tensioned via tuning lugs to a uniform level. One membrane (the batter head) is commonly struck with a stick, while the optional second membrane (the resonant head) vibrates sympathetically.

This creates a sound with two main components. The attack component is formed from the vibrations associated with the initial collision. A two-dimensional travelling wave then moves through the membrane, reflecting at the bearing edge, to form a standing wave, responsible for the sustain component of the drum sound. This contains normal modes (Fig. 1) which are solutions to the two-dimensional wave equation. Additional terms are necessary in the equation to fully describe observed behaviour, such as frequency dependent losses and nonlinear behaviour (eg. pitch glide); various forms of the 2-D wave equation with additional terms are discussed at length in (Torin, 2016), where it is explained that membrane vibration is a special case of plate vibration. A membrane being a thin plate, that is tensioned at its edge. The wave equation for an ideal membrane is as follows:

$$\frac{\partial^2 \Psi}{\partial t^2} = c^2 \Delta \Psi \tag{2}$$

where Ψ is the displacement of the membrane, Δ is the Laplacian operator, and t is time. Drum membranes are fixed at the bearing edge, leading to the boundary conditions known as "clamped conditions", where both the displacement and the gradient of displacement at the rim are zero.

An ideal circular membrane has Bessel-function solutions in the radial direction, and cosine function solutions in the azimuthal direction. As explained in (Errede), this leads to modes being classified by their number of nodal diameters, m, and their number of nodal circles, n. This is written as $(m,\ n)$, where the fundamental frequency is (0, 1). These solutions have the following form:

$$\Psi_{m,n}(r,\psi,t) = \alpha_{m,n} J_m(k_{m,n} r) \cos{(m\psi)} \cos{(\omega_{m,n} t)} \tag{3}$$

where $\Psi_{m,n}(r,\psi,t)$ is the modal displacement, at polar coordinate (r,ψ), at time t, for a membrane of radius R. $\alpha_{m,n}$ is the amplitude of modal oscillation at an antinode, $J_m(k_{m,n}r)$ is the m^{th}-order first kind Bessel function, $k_{m,n}$ is the wavenumber in m^{-1}, chosen so that $(k_{m,n}R)$ is the n^{th} non-trivial zero of the m^{th}-order first kind Bessel function, to satisfy the aforementioned boundary conditions at $r = R$.

Modal frequencies can therefore be calculated via $c = f\lambda$ as:

$$f_{m,n} = \frac{k_{m,n}}{2\pi}\sqrt{\frac{T}{\sigma}} \tag{4}$$

where σ is the surface density of the plate in $\text{kg}\,\text{m}^{-2}$, T is the surface tension in $\text{N}\,\text{m}^{-1}$, $\lambda_{m,n}$ is wavelength in m, $c = \sqrt{T/\sigma}$ is the wave speed in $\text{m}\,\text{s}^{-1}$, and $k_{m,n} = 2\pi/\lambda_{m,n}$. It should be noted that measured values can vary from their ideal values (Skrodzka *et al.*, 2006).

The amplitude of each modal oscillation is dependent on the amplitude of the modal surface at the strike position. It follows from equation (3) that central strikes excite circular modes (such as the fundamental) due to their central antinode. Off-center strikes will excite radial modes, causing the characteristic overtones to be heard. The presence of a second membrane also complicates the system, as the membranes resonate in a coupled fashion

(Bilbao, 2012). This creates additional modes and can suppress or accentuate existing modes.

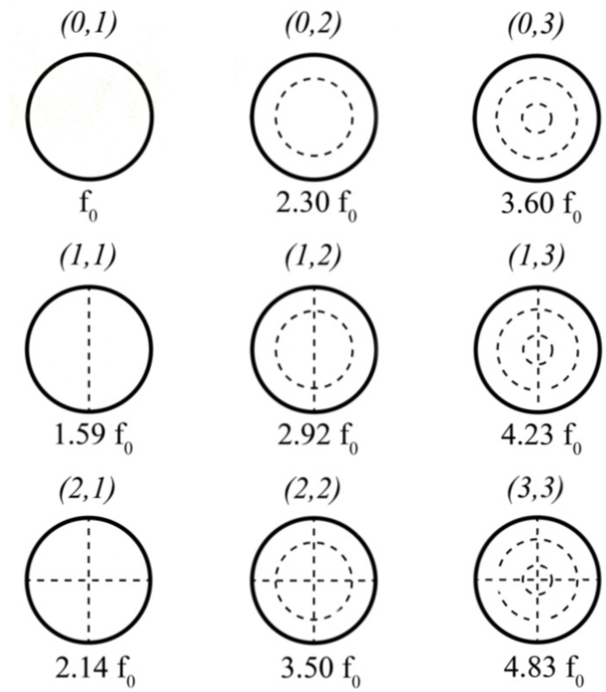


FIG. 1. Theoretical modes of a circular membrane, where (m,n) describes the number of nodal diameters, m, and the number of nodal circles, n.

When a drum is struck at high intensity, the local tension in the skin increases. This leads to the characteristic pitch glide found associated with drum sounds (Avanzini and Marogna, 2012). This is a downwards pitch glide, where the changing frequency of each mode, is proportional to that of the fundamental (Fletcher and Bassett, 1969). Finite Difference models have modeled this effect using a non-linear term from the von Kármán equations for thin plates (Torin and Newton, 2014).

Resonant modes are clearly identifiable in spectrograms as well defined curves (Fig. 2), with visible pitch glide. These make up the sustain component of the drum sound. The remaining energy is less well defined, and makes up the attack component that is produced by the initial collision. The sustain component dominates the bass and low mids, while the attack component dominates the higher frequencies. These components could be considered separately, for example, modeling them as deterministic and stochastic, respectively.

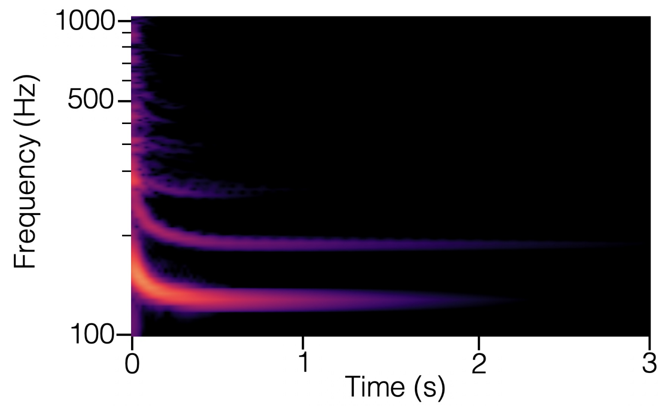


FIG. 2. Spectrogram of a 9x10” Yamaha Beech Custom tom-tom, struck centrally at maximum velocity, with visible pitch glide. Made using Sonic Visualiser (Cannam *et al.*, 2010). (Color Online).

III. METHOD

This section explains the overall method, starting with the dataset of drum samples (Section III A). Next, mathematical definitions are provided (Section III B), outlining the concept of Inverse Discrete Cosine Transform synthesis, as applied to drums here. This includes the definition of a modal feature, which is central to this research. Then, the chirp like form of a modal feature is explained (Section III C). Finally, the Hilbert transform is explained, along with how it is used to decompose modal features into simple instantaneous amplitude and phase functions (Section III D).

A. Data Set

Tom-tom samples were extracted from Superior Drummer 3 by Toontrack, by triggering every sample from 1-127 midi velocity, with all velocity layers loaded, and all hit variation features turned off. Each sample is 8s long, to ensure a full decay. The following analysis is based on the 9x10" Yamaha Beech Custom tom-tom, but is applicable to any tom-tom, and is likely to transfer to any drum that exhibits pitch glide.

67 unique samples of central strikes were obtained, and only one was deemed to be anomalous (off-centre). As the unique samples had been mapped to multiple velocities, each sample was now labelled with a single velocity (the lowest of the mapped values). The integrated loudness, as defined by (International Telecommunication Union, 2011), was also calculated for each drum sound, using the "integratedLoudness" command in MATLAB.

Fig. 3 depicts this dataset, and demonstrates how integrated loudness increases with midi velocity, as expected. There is some clustering around specific loudness values, which indicates the presence of "velocity layers". These velocity layers are a common feature of sample-based VI's, containing samples of similar loudness, which can be triggered consecutively to somewhat alleviate the "machine gun effect".

Both midi velocity and integrated loudness have their merits; midi velocity is a useful symbolic notation, which clearly indicates the maximum range of strike intensities that the sampling team deemed to be musically appropriate. It is, however, limited as an independent variable, being a discrete and relative measure. It is also worth noting that samples of identical loudness would be mapped to distinct, neighbouring velocities, because of the way velocity layers are programmed. Integrated loudness is a continuous and absolute measure, so was deemed to be a more suitable independent variable for the remainder of this study.

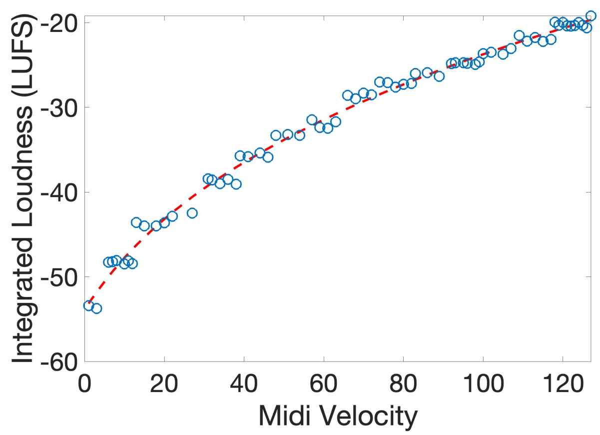


FIG. 3. Scatter plot of the mapping between midi velocity and integrated loudness, for the chosen data set (9x10" Yamaha Beech Custom tom-tom from Superior Drummer 3). There is a strong positive correlation, as expected, which has been fitted with a dual exponential. Note that the samples are clustered into velocity layers, of similar loudness. (Color online.)

B. Inverse Discrete Cosine Transform synthesis of drum sounds

The sustain component of a drum sound can be viewed as the sum of each mode's time domain oscillation:

$$x_{sus}(t) = \sum_{m=1}^{M} \sum_{n=1}^{N} x_{m,n}(t) \tag{5}$$

where $x_{sus}(t)$ is the sustain component's time domain signal, $x_{m,n}(t)$ is each individual mode's time domain signal, M is the maximum number of nodal diameters to be considered, and N is the maximum number of nodal circles. This yields a total of $N(M+1)$ modes.

Each mode's time domain signal could be modeled as:

$$x_{m,n}(t) = a_{m,n}(t) \cos\left(2\pi f_{m,n}(t)t + \theta_{m,n}\right) \tag{6}$$

where $a_{m,n}(t)$ is the instantaneous amplitude (envelope) of each mode's time domain signal, $f_{m,n}(t)$ is the frequency trajectory of each mode (incorporating any pitch glide and tending to the natural frequency), and $\theta_{m,n}$ is the phase constant of the mode's oscillation.

The DCT can be used on the entire sustain component, to obtain $X_{sus}(\omega)$, the full spectrum of resonant modes:

$$X_{sus}(\omega) = \mathrm{DCT}[x_{sus}(t)] \tag{7}$$

Or individually on a single mode, to obtain $X_{m,n}(\omega)$, an individual modal feature:

$$X_{m,n}(\omega) = \mathrm{DCT}[x_{m,n}(t)] \tag{8}$$

Due to the linearity of the Discrete Cosine Transform, addition in the frequency domain is equivalent to that in the time domain. It follows that the superposition of all modal features returns $X_{sus}(k)$:

$$X_{sus}(\omega) = \sum_{m=1}^{M}\sum_{n=1}^{N} X_{m,n}(\omega) \tag{9}$$

Each modal feature, $X_{m,n}(\omega)$, will be a sparse signal, that is non-zero only at frequencies close to each modes natural frequency, as one would expect in the spectrum of a single mode. Modal features encode $x_{m,n}(\omega)$, the entire time domain signal of a given mode, as defined in equation (6), and can be recovered individually:

$$x_{m,n}(t) = \mathrm{IDCT}[X_{m,n}(\omega)] \tag{10}$$

Or collectively:

$$x_{sus}(t) = \mathrm{IDCT}[X_{sus}(\omega)] \tag{11}$$

Equations (9) and (11) can therefore be used to synthesise the entire sustain component from modelled modal features. The attack component can also be synthesised by modeling its spectrum, as shown in (Kirby and Sandler, 2020). The two components can be superposed to create a full drum sound. The attack component requires a different model, however, and is not the focus of this paper.

C. DCT representation of fundamental mode

Similarly to the FFT, activity in the DCT magnitude representation corresponds to energy at a given frequency. Tom-tom samples contains chirp like modal features (Kirby and Sandler, 2020). Four modal features are shown in Fig. 4. Each modal feature, $X_{m,n}(\omega)$, encodes the entire time domain signal of the respective mode, including the characteristic envelope and pitch glide of the sinusoid.

The mapping between domains is best understood numerically:

1. Isolate the fundamental mode from a drum sample using a bandpass filter.

2. Calculate the DCT contribution from each successive time domain sample, and inspect the effect that each has on the DCT representation.

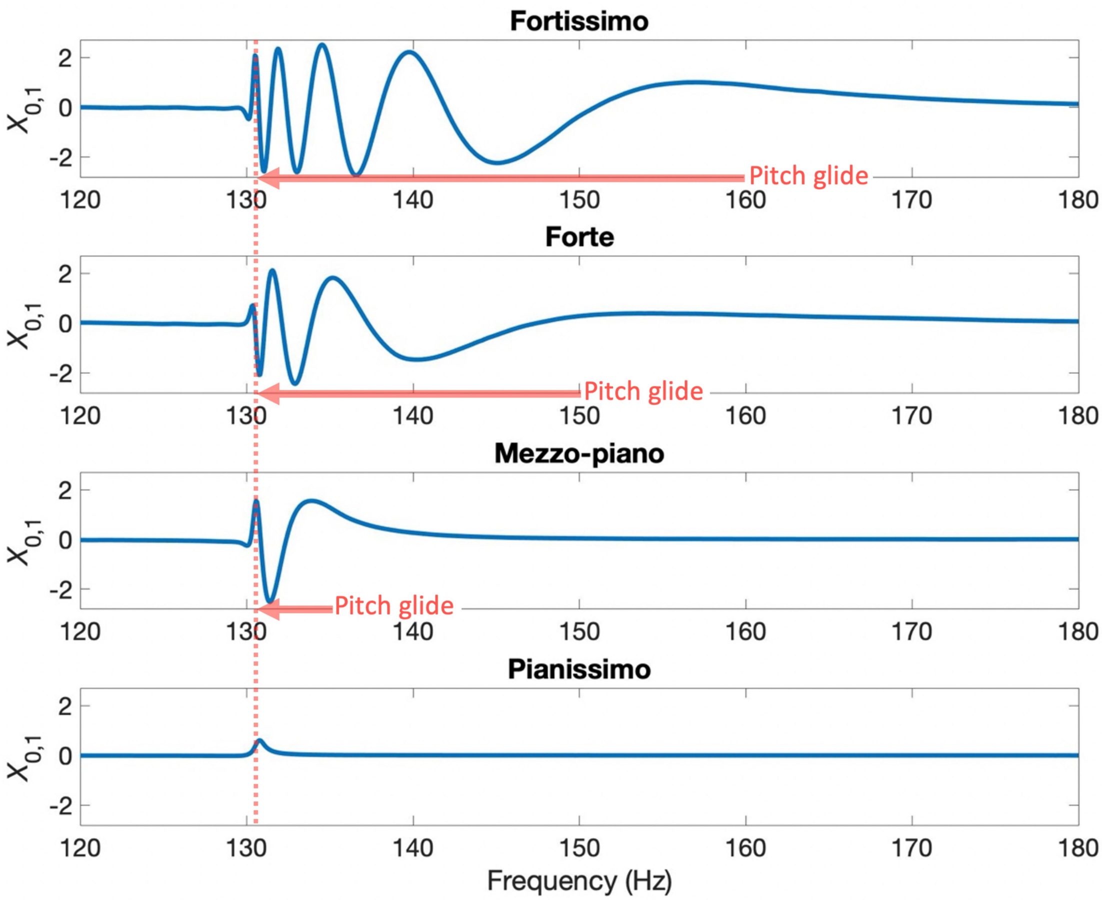


FIG. 4. DCT representation of the fundamental frequency of a 9x10" Yamaha Beech Custom tom-tom, struck at four different velocities. Four midi velocities were chosen throughout the full range of 1-127, to best illustrate the chirp like shape of modal features, and how this shape evolves with strike velocity. The dashed line indicates the unmodulated fundamental frequency of 131 Hz. Arrows indicate the direction and magnitude of the pitch glide, where appropriate. Midi velocities are 126, 105, 78, and 15 for fortissimo, forte, mezzo-piano, and pianissimo respectively. Modal features encode the full time domain signal of the mode; wider peaks correspond to earlier activity, and narrower peaks correspond to later activity.(Color online).

3. Notice that the initial samples correspond to a peak in the DCT located at the tension modulated frequency. When successive samples contributions are introduced, the DCT activity gradually shifts to lower frequencies, until the unmodulated frequency is reached, by which time a chirp signal is reliably obtained between these frequencies.

As modal features are only non-zero over a limited frequency range, they can be stored as sparse vectors, requiring far fewer samples than their corresponding time domain signal, $x_{m,n}(t)$ (of order 10^3 less in this research).

Each modal feature can be modeled in the DCT frequency domain as:

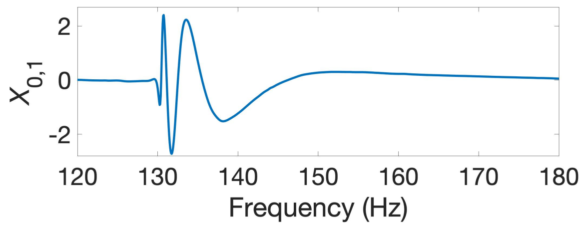
$$X_{m,n}(\omega) = A_{m,n}(\omega) \cos\left(\Phi_{m,n}(\omega)\right) \tag{12}$$

where $X_{m,n}(\omega)$ is the vertical axis that corresponds to DCT magnitude, $A_{m,n}(\omega)$ is the instantaneous amplitude (envelope) of the modal feature, $\Phi_{m,n}(\omega)$ is instantaneous phase of the modal feature, and $\omega = 2\pi f$, where f is the horizontal frequency domain axis in Hz.

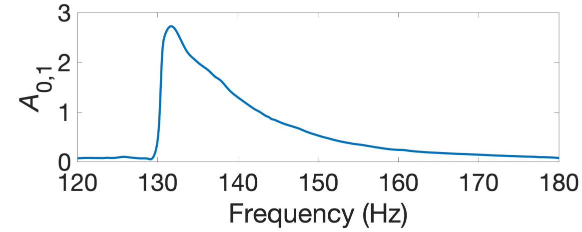
This is illustrated in Fig. 5, where a modal feature is plotted, along with its instantaneous amplitude and phase functions. The amplitude and phase functions are easier to model, so it is useful to decompose modal features in this manner. This method of decomposition is discussed in the following subsection.

D. Decomposing modal features via the Hilbert transform

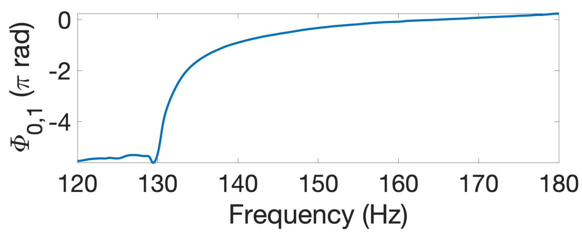
Modal features are decomposed into instantaneous amplitude and phase using the Hilbert transform. The Hilbert transform is related to the analytic signal (Rossi and Girolami, 2001). The analytic signal is a complex representation of real-valued signal. This real-valued signal



(a)



(b)



(c)

FIG. 5. (a) DCT representation of the fundamental frequency of a 9x10" Yamaha Beech Custom tom-tom, struck at the moderately high velocity of 102 out of 127. This is $X_{0,1}(\omega)$, a representative example of a chirp like modal feature, with an unmodulated fundamental frequency of 131 Hz. (b) Instantaneous amplitude (envelope) of the above modal feature, $A_{0,1}(\omega)$, extracted using the Hilbert transform. (c) Unwrapped instantaneous phase of the modal feature, $\phi_{0,1}(\omega)$, also extracted via the Hilbert transform. (Color online).

can be in any domain, such as the frequency domain where we will be using it, but first, it is best understood, as conventionally defined, in the time domain:

$$z(t) = u(t) + j\hat{u}(t) \tag{13}$$

where $z(t)$ is the analytic signal, $u(t)$ is the real-valued signal, and $\hat{u}(t)$ is the Hilbert transform of $u(t)$, denoted as H:

$$\hat{u}(t) = \mathrm{H}[u(t)] \tag{14}$$

The Hilbert transform introduces a phase shift of $\pm\pi/2$ on each frequency component, meaning that $u(t)$ and $\hat{u}(t)$ are in quadrature, and $z(t)$ has no negative frequency components.

The discrete-time Hilbert Transform is calculated through the following three-step algorithm (Marple, 1999) which returns the analytic signal:

1. Compute the Fast-Fourier Transform (FFT) of $u(t)$:

$$U(\omega) = \mathcal{F}[u(t)] \tag{15}$$

   where $u(t)$ is the real-valued signal of length Ω, $\mathcal{F}$ is the Ω-point FFT, and $U(\omega)$ is the FFT representation of $u(t)$.

2. Form the one-sided discrete-time analytic signal transform, Z(k), as follows:

$$Z(\omega) = \begin{cases} U(\omega), & \text{if } \omega = 1, \frac{\Omega}{2} + 1 \\ 2U(\omega), & \text{if } \omega = 2 : \frac{\Omega}{2} \\ 0, & \text{if } \omega = \frac{\Omega}{2} : \Omega \end{cases} \tag{16}$$

3. Compute the IFFT to return the analytic signal:

$$z(t) = \mathcal{F}^{-1}[Z(\omega)] \tag{17}$$

The analytic signal can be used to decompose $u(t)$ into instantaneous amplitude (envelope) and instantaneous phase functions:

$$a(t) = |z(t)| \tag{18}$$

where $a(t)$ is the instantaneous amplitude and $|z(t)|$ is the magnitude of $z(t)$.

$$\phi_{\text{wrapped}}(t) = \arg\left(z(t)\right) \tag{19}$$

where arg is the argument of $z(t)$, and $\phi_{wrapped}(t)$ is the wrapped instantaneous phase which is constrained to $-\pi \leq \phi \leq \pi$. $\phi_{wrapped}(t)$ can unwrapped to form $\phi_{unwrapped}(t)$, a function that does not contain the discontinuities associated with wrapping.

The instantaneous amplitude and instantaneous phase functions can be recombined to return the original signal:

$$u(t) = a(t)\cos\left(\phi(t)\right) \tag{20}$$

where $\phi(t)$ is interchangeably $\phi_{\text{wrapped}}(t)$ or $\phi_{\text{unwrapped}}(t)$.

In this subsection, the Hilbert transform was operating on a time domain signal. In this research, however, the transform is used in the frequency domain, on the modal feature $X_{m,n}(\omega)$, to determine the instantaneous amplitude, $A_{m,n}(\omega)$, and instantaneous phase, $\Phi_{m,n}(\omega)$, of the modal feature, as in equation (12). This gives us a representation of a modal oscillation that is much simpler to model than its corresponding time domain signal. These modal features are only active over a very narrow frequency range, so are often naturally isolated in the frequency domain. These isolated modal features give us a ”ground truth” representation to model.

The fundamental modal feature of a moderate velocity sample is shown in Fig. 5, along with its Hilbert transform decomposition. A moderate velocity was chosen to provide a a most representative example of a modal feature. The decomposed functions provide a simple, yet exact, representation of a drum mode oscillation.

IV. ANALYSIS OF DATA SET

In this section, the fundamental of each drum sample in the data set is analysed using the framework described in Section III. The instantaneous amplitude and phase are plotted, and their evolution with strike velocity is described (Section IV A). The clear evolution of these functions inspires the development of an interpolation based synthesis method, to synthesise modal behaviour at intermediate strike velocity (Section IV B).

A. Evolution of DCT representation with strike velocity

First, the fundamental modal features, $X_{0,1}(\omega)$, were extracted from the DCT representation of each complete drum sample. As these modal features were naturally isolated in the frequency domain, it was possible to simply specify a frequency range where the modal feature was non-zero, and then set the DCT magnitude to zero, for frequencies outside this range. This was done manually, with a typical frequency range of 76-186 Hz, centred around the mean unmodulated fundamental frequency of 131 Hz. The Hilbert transform was then used on the modal features, as explained in Section III D, to obtain the instantaneous amplitude, $A_{0,1}(\omega)$, and the instantaneous frequency, $\Phi_{0,1}(\omega)$ for each modal feature. The instantaneous phase functions were unwrapped so that their asymptotes were located at the same approximate phase value.

A very clear evolution was observed in the amplitude and phase functions, as strike velocity increases, as shown in Fig. 6. The amplitude functions are skewed bell curves. As strike velocity increases, the area under the curve increases, which corresponds to an increase in volume. This is to be expected, as increasing strike velocity drives larger amplitude oscillations. This is demonstrated in Fig. 7, which shows that the area scales smoothly with integrated loudness, and is well fit by a dual exponential.

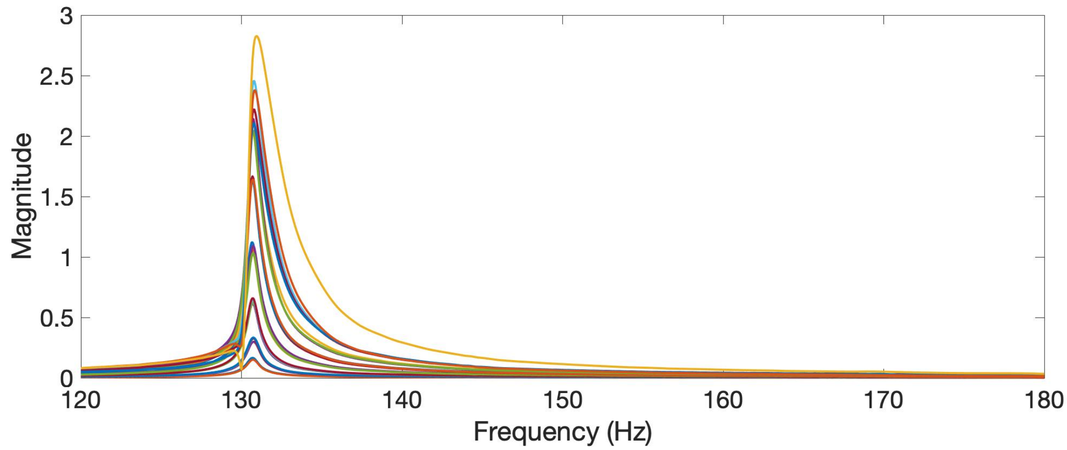
The amplitude function also becomes progressively more positively skewed. This is due to the increasing amount of pitch glide. The function is near symmetrical at low velocities, where there is negligible pitch glide. As the amount of pitch glide increases, more energy is introduced at frequencies above the unmodulated fundamental frequency, increasing the asymmetry of the modal feature. At the highest velocities, some slight distortion is also observed in the peak, such as that visible in Fig. 5b, though this is of relatively low perceptual importance, having no noticeable effect on the accuracy of the interpolation based synthesis in Sections V and VI.

The phase functions are smooth curves, with horizontal asymptotes. These curves create the chirp like shape of the modal feature, $X_{m,n}(t)$, encoding the general form of the modal oscillation, $x_{m,n}(t)$. The total change in phase increases with strike velocity, between the values of π and 13π radians, with each π radian adding another peak or trough to the overall modal feature. The frequency width of the active region also increases, which corresponds to the increasing magnitude of pitch glide, as demonstrated in Fig. 8. There is negligible pitch glide for loudness values below -36 LUFS, as the strike velocity is low, so the resulting oscillation is of low amplitude. The pitch glide becomes increasingly prominent beyond this threshold, due to the increase in tension that occurs in large amplitude oscillations.

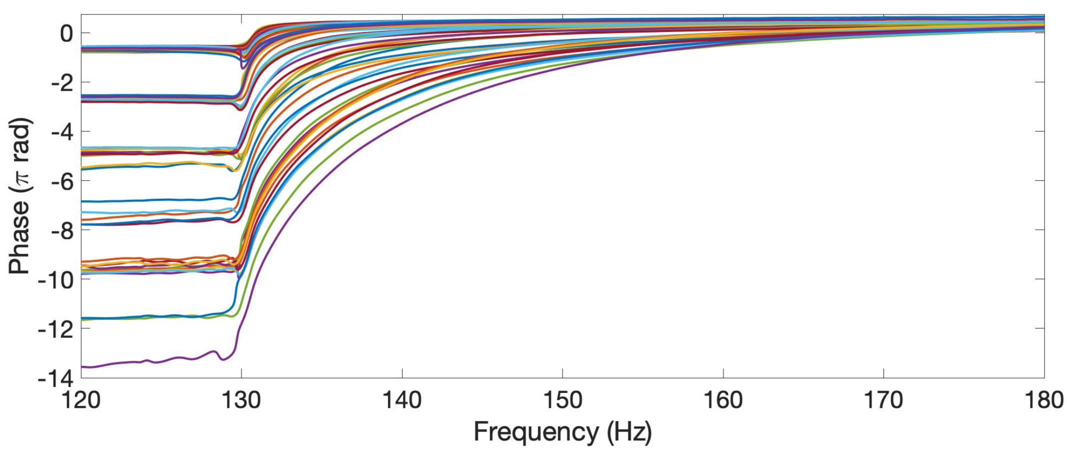
B. Interpolation of DCT representation

The clear manner in which modal features have been shown to evolve can inform modal synthesis. Synthesising a mode is simply a matter of modeling the amplitude and phase functions for a given modal feature, and then using equation (12) followed by equation (10). Not only is it possible to model reference modal features, we can use linear interpolation on both the amplitude functions and the phase functions of references modal features, to synthesise modal features of intermediate velocity, thereby creating new sounds. This means we can synthesise modes in a dynamic fashion. This interpolation based method is just one of many possible ways to model the evolution of modal features with strike velocity. It is used to demonstrate the value of the DCT framework in the context of dynamic modal synthesis.

This process is illustrated in Fig. 9. The interpolation based method could be used to synthesise additional samples, to supplement those recorded in a given library. For example, the 67 samples in this detailed library could be topped up to 127, or in fact, any chosen number, to create a performance that is as dynamic as the given control surface will allow, and to avoid the "machine gun effect" that can occur when note velocities are repeated. Equally, this technique could add some sorely needed detail to much more limited libraries.



(a)



(b)

FIG. 6. Evolution of the decomposed DCT representation with strike velocity, for the fundamental frequency of a 9x10" Yamaha Beech Custom tom-tom. (a) Evolution of the amplitude function. The amplitude functions of all unique samples between midi velocity 1-66 are overlaid, as these illustrate the general trend in a clear manner. The area under the amplitude function increases with strike velocity. (b) Evolution of the phase function. The phase functions of all unique sample in the data set are overlaid. The maximum magnitude of phase in the curved section increases with strike velocity, as does the frequency width. (Color online).

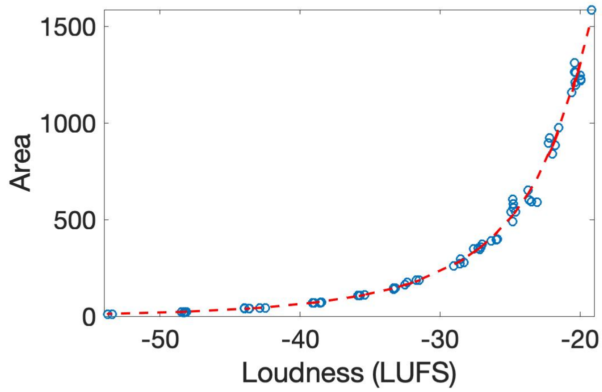


FIG. 7. Scatter plot of the tight correlation between the area under the amplitude function of the fundamental mode, and the integrated loudness of the drum sample. The entire data set is plotted, and fitted with a dual exponential. (Color online.)

Simply using this technique on the fundamental mode is enough to create a unique sounding sample, but this technique could also be used on any number of modes.

V. OBJECTIVE EVALUATION OF SYNTHESIS

In this section, we objectively evaluate the interpolation based synthesis method, to demonstrate that the proposed framework is valuable. This interpolation based approach is one of many possible methods in which modal features can be modeled. It serves as a proof of concept for modeling modal features in general, and also has a clear application in augmenting existing sample libraries. First, an intial test is described (Section V A), which augments the number of samples in the dataset, based on midi velocity, from 67 to 127, which is the maximum number of velocities supported by midi. Next, a more rigourous test

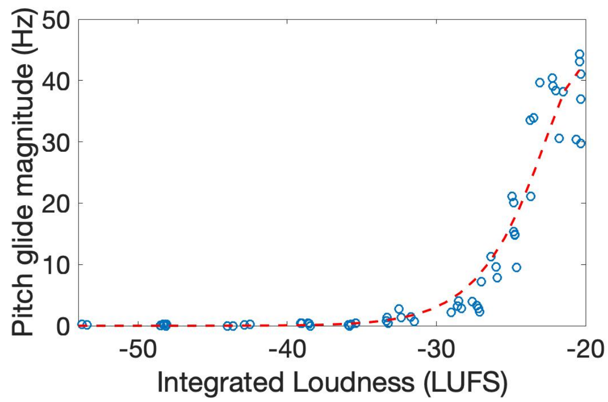


FIG. 8. Scatter plot of pitch glide magnitude against integrated loudness for the fundamental mode of each sample in the data set. The amount of pitch glide has been estimated from the frequency width of the phase function. There is negligible pitch glide for loudness values below -36 LUFS, as the strike velocity is low, so the resulting oscillation is of low amplitude. The pitch glide becomes increasingly prominent beyond this threshold, due to the increase in tension that occurs in large amplitude oscillations. A dual exponential has provided a reasonable fit, to guide the eye. (Color online.)

is performed (Section V B), that employs null testing for objective evaluation. Finally, the results of this null test are explained (Section V C).

A. Initial test

As an initial test, each of the 67 fundamental modal features were interpolated between, based on midi velocity, to create a total of 127 unique fundamental modes. Each extracted amplitude function was labelled with its respective midi velocity, and interpolation was used to generate amplitude functions for every velocity from 1-127. This was performed via the ”griddata” command in MATLAB, with the linear interpolation method selected. For example, there were unique samples assigned midi velocities of 102 and 105, so the amplitude functions for these velocities would be interpolated between, to create amplitude functions for velocities 103 and 104. This process was then repeated on the extracted phase functions.

Finally, each of the 127 pairs of amplitude and phase functions were inserted into equation (12), to generate 127 modal features in the frequency domain. The IDCT was then used on each modal feature (10), to synthesise the 127 fundamental modes in the time domain, where 67/127 were genuine fundamentals modes, and the remaining 60/127 were synthesised at intermediate velocities. A smooth blending of modal activity was achieved, with no output that appeared anomalous.

B. Null test: Method

Next, a null test was performed, to assess what loudness resolution was required for each synthesised fundamental to null (below -90 dBFS) with the consecutive sample, throughout the entire loudness range. This time, the interpolation was based on the integrated loudness values of each fundamental mode, rather than midi velocity, as this is a more suitable independent variable for rigorous investigation, as explained in Section III. The integrated loudness of each reference fundamental was first rounded to 1 decimal place, which resulted in a minimum loudness of -56.0 LUFS and a maximum loudness of -19.3 LUFS. These loudness values were then interpolated between, to synthesise behaviour at each unique loudness value within in this range, at this level of precision.

For example, there were fundamentals of similar integrated loudness with values of -28.7 and -28.3 LUFS, so the amplitude functions for these velocities would be interpolated between, to create amplitude functions for loudness values of -28.6, -28.5, and -28.4. This process was repeated for the phase functions, to create 367 modal features, which were again used to synthesise fundamental modes in the time domain, where 67/367 were genuine modes, and the remaining 300/367 were synthesised at intermediate loudness values.

A null test was then performed on every successive pair of fundamentals of increasing loudness (eg. the fundamental with loudness -56.0 LUFS was tested against the fundamental of loudness -55.9 LUFS, next the fundamental of loudness -55.9 LUFS was tested against the fundamental of loudness -55.8 LUFS, and so on).

The loudness resolution of 1 decimal place was insufficient for all pairs of fundamentals to null, so the loudness resolution was increased by rounding loudness values to one additional decimal place of accuracy. The experiment was then repeated at this increased precision. Increasing the precision by 1 decimal place will yield a factor of 10 increase in the number of modes to be null tested. Each successive pair will then be 10 times closer in loudness, and therefore more likely to null. The entire process was repeated until a precision was found where all sample pairs null below -90 dBFS.

C. Null test: Results

The results of the null test are shown in Table I. A loudness resolution of 1×10^{-7} LUFS was sufficient for all pairs to null below -90 dBFS. The main purpose of the null test was to demonstrate objectively that the continuous blending of phase and amplitude functions

TABLE I. Results of the null test

Loudness resolution (LUFS)	Number of pairs	Proportion that null (%)	Overall result
0.1	367	0.00	×
0.01	3679	29.1	×
0.001	36791	64.4	×
0.0001	367904	76.9	×
0.00001	3679042	96.7	×
0.000001	36790424	98.4	×
0.0000001	367904244	100.0	✓

results in the continuous blending of output audio. This confirmatory result demonstrates that the proposed framework captures the dynamics of modal behaviour. This framework can therefore be used to inform a synthesis technique that can accurately model the changes in volume, pitch glide, and decay time that occur with increasing strike velocity.

A loudness resolution of $1{\times}10^{-7}$ LUFS corresponds to $3.67{\times}10^{8}$ fundamentals being synthesised throughout the loudness range. This also provides an upper limit on the number of samples that would be required for this articulation in an ideal library. This is an overestimate, as a significant proportion of mode pairs nulled when lower levels of precision were used, as shown in Table I. This is because the required loudness resolution varied through the loudness range. Sample pairs of relatively low loudness were more likely to null at a given loudness resolution, than sample pairs of higher loudness.

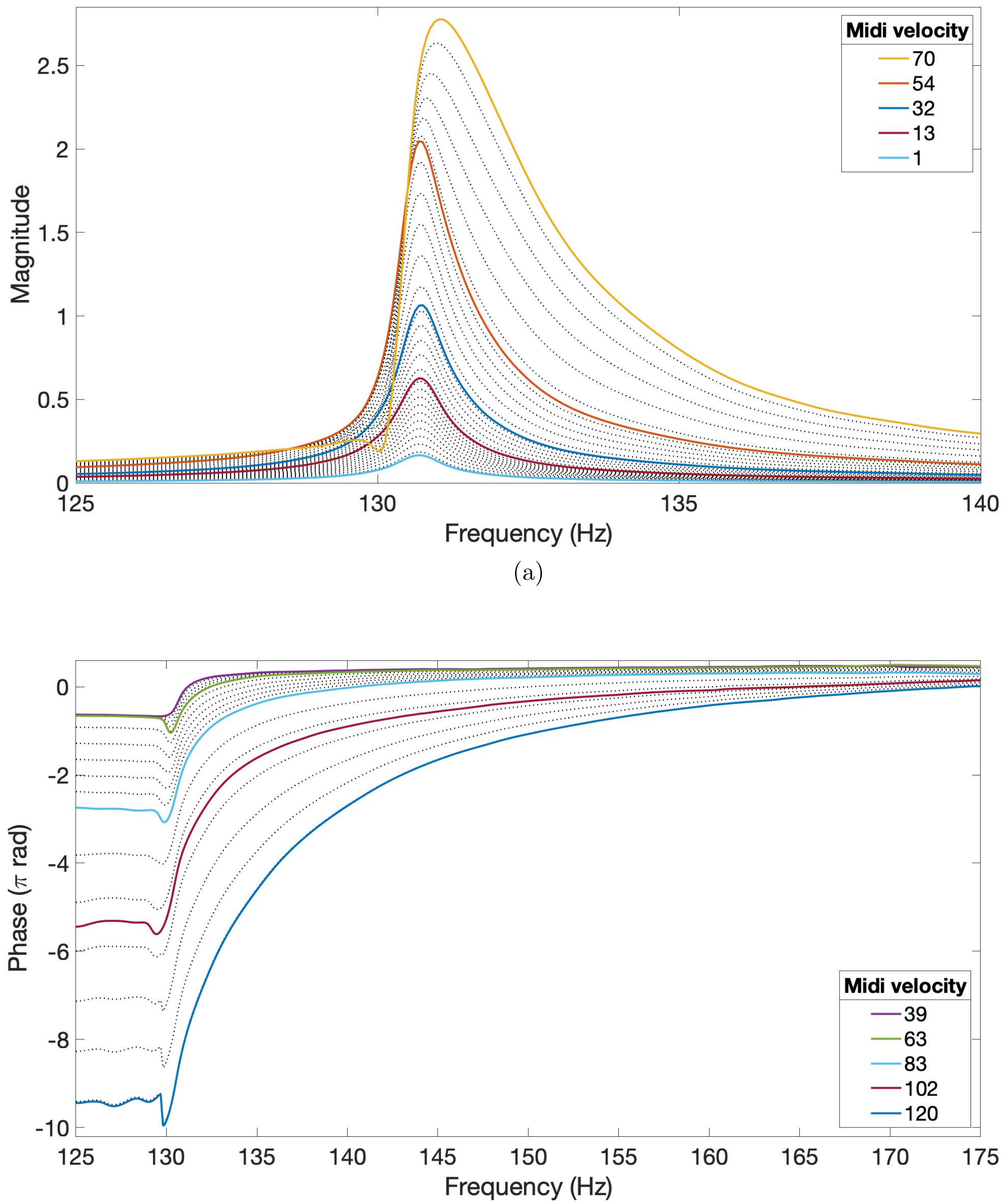
VI. PERCEPTUAL EVALUATION OF SYNTHESIS

Finally, a listening test was used to perceptually evaluate the realism of the modes that were synthesised using the interpolation based method. Section VI A describes the systematic sampling of the data set to select reference samples. Section VI B describes the interpolation based synthesis method used to generate synthesised data, for testing against the reference samples. Section VI C explains the experimental design of the listening test.

A. Systematic sampling

Amplitude and phase functions were once again synthesised using linear interpolation, in order to generate synthetic fundamental modes. This time, only a systematic sample of the 67 samples from the data set were interpolated between, leaving the remaining intermediate samples available as references. These references can be compared to the synthesised data, to judge the accuracy with which the linear interpolation method models modal behaviour at intermediate velocity.

In the following, we refer to samples that are interpolated between as training data, and the remaining reference samples as test data. To generate the training data, the drum samples were ordered by integrated loudness, and systematic samples were taken using the criteria shown in Table II. Three different samples were taken, with varying amounts of training data, to investigate the effects of varying interpolation distance. The hypothesis



(a)

(b)

FIG. 9. Interpolation of the amplitude and phase functions. Solid lines indicate genuine functions extracted from the dataset, dotted lines indicate synthesised functions. Parameters were chosen to best illustrate the concept. The midi velocity of each sample is shown in the legend, and the vertical order reflects that of the functions themselves. (a) Interpolation of amplitude function. (b) Interpolation of phase function. (Color online).

was that the interpolation based method would be most accurate for group A, as there were more training samples to be interpolated between, which meant the interpolation distance was smaller.

TABLE II. Sampling methods used to create the three training sets

	Sampling Interval	Training samples	Test samples	Total
Group A	2	34	33	67
Group B	8	9	58	67
Group C	16	5	62	67

B. Synthesis method

Interpolation was again performed based on integrated loudness. For each of the three groups, a fundamental mode was synthesised for each distinct loudness value within the full loudness range, by interpolating between the amplitude and phase functions that were present in each respective systematic sample. The remaining test samples in each group were then matched with a synthesised fundamental of equal loudness, for comparison in a listening test.

As the isolated fundamentals sound unusual when played out of context, both the genuine and synthesised fundamentals were played in context during the listening test. This was achieved by comparing the reference drum sound, to an edited version of the exact same drum sound, where the fundamental mode had been replaced with the synthetic one. This was achieved by replacing the genuine fundamental modal feature in the DCT representation, with the synthesised modal feature. The IDCT was then used synthesise this hybrid drum sound in the time domain, as in equation 11.

In this use case, it was deemed important that the synthesised samples sounded as realistic as the reference samples, rather than necessarily identical. This is because there can be subtle intrinsic variation between reference samples at a given loudness. Also, the matching process is not perfect, as the true amplitude and phase functions will not evolve in the strictly linear sense that the interpolation approximates. This discrepancy is most evident for Group C, which has the largest interpolation distance, meaning that the 62 remaining samples are all approximated from merely 5 training samples.

C. Experimental Design

Initial tests suggested that many pairs sounded identical, and while some had noticeable differences, they still sounded real. This was also the overwhelming qualitative feedback from participants. To this end, an AB test was chosen, with participants being asked to choose the sample that sounded the most realistic.

20 participants between the ages of 26 and 74 took part in the test, of which 13 were male and 7 were female. 13 were deemed to be critical listeners, based on their relevant experience as musicians, mixing engineers, and/or recording engineers. The test was performed using the Web Audio Evaluation Tool (Jillings *et al.*, 2015), with participants being instructed to wear high quality headphones (this was confirmed by survey). Drum sounds

were level matched to -23 LUFS, and participants were asked to leave their system volume at a constant, comfortable level.

VII. RESULTS AND DISCUSSION

The results from the listening test are summarised in Table III. The were no significant differences between the results for the full participant popuation, and those of the subset of critical listeners, so the results for the full sample are shown.

Binomial hypothesis testing was used to analyse the results. The test statistic, Y, which is the number of times the participants correctly selected the real drum sound was modelled as $Y \sim \mathrm{B}(n_{trials}, p_{correct})$, where n_{trials} is the number of trials, and $p_{correct}$ is the probability of a correct answer. This leads to $Y_{Group} \sim \mathrm{B}(400, 0.5)$ for each group and $Y_{Total} \sim \mathrm{B}(1200, 0.5)$ overall. The null hypothesis was that the synthesised modes sound realistic ($p_{correct} = 0.5$) and the alternative hypothesis was that there is an audible difference in realism $p_{correct} > 0.5$. A 10% level of significance was used for the test.

There is no evidence to reject the null hypothesis that synthesised samples sound realistic, for any of the 3 sets of data, or the combined total, at the 10% significance level. In fact, the overall participant accuracy was exactly equal to the expectation value of 50%. The result furthest from this expectation value was 47% for group B, but this indicated that the synthesised sounds outperformed the reference samples in terms of realism, which is a nonsensical result, if not for statistical fluctuation.

These results demonstrate that the participants could not distinguish the real samples from those that were synthesised. This tallies with the overwhelming qualitative feedback

TABLE III. Results of AB test for perceptual evaluation of synthesis method.

	Group A	Group B	Group C	Total
Training Samples	33	9	5	N/A
Correct	202/400	188/400	210/400	600/1200
Percentage	50.5%	47.0%	52.5%	50.0%
p-value	0.401	0.875	0.147	0.489

that many pairs sounded identical, and that on occasions where there were noticeable differences, the participants still couldn't tell which of the drum sounds was part synthesised.

While it is still possible that there could be very subtle perceptual artefacts that become apparent with familiarity, these results demonstrate that there is not a marked difference between the real and synthesised samples. This is an encouraging result, as most, if not all, synthesised drum samples sound clearly synthetic. Any differences would also be much harder to spot in the context of a mix, and even if some mix professionals could tell the difference, it would be doubtful that the general public could.

On the other hand, these samples were only part synthesised, and while the fundamental is clearly the most important component, further work needs to be done, to synthesise full samples. The current results strongly suggest that that this would be a worthwhile investigation. These fully synthesised sounds could then be perceptually evaluated against other synthesis methods.

VIII. CONCLUSION

In conclusion, it has been shown that drum modes are represented as chirp like signals in the DCT transform domain, providing a far simpler representation than the time domain signal itself. These chirps can be decomposed using the Hilbert Transform, to create an even simpler amplitude and phase function representation. A clear evolution with strike velocity was observed in both these signals. This evolution can be modeled, to synthesise drum sounds in a dynamic fashion, capturing not just a snapshot of a drum sound at a specific strike velocity, but entire modal behaviour throughout the entire loudness range. As an example of this kind of modeling, interpolation was used to synthesise modal behaviour at intermediate velocity.

The continuous blending of drum mode time domain signals that was achieved in the null test, is confirmation that the proposed DCT representation is a valuable framework for analysis and synthesis. The results of the listening test also support this conclusion, demonstrating that the synthesised intermediate modes sounded equally realistic to the participants as the references, with as few as 5 training samples. This strongly suggests that these synthesised intermediate modes are a good approximation of genuine intermediate behaviour, rather than merely the result of an unphysical warping between samples. This conclusion is also supported by the unambiguous evolution of modal features that is evident in the reference samples.

Overall, it has been proven that this technique can be used to analyse drum modes, and synthesise modes of intermediate velocity, with trivial computational cost. This could be used to create or enhance virtual instrument libraries. In addition, further research could lead to a full analytical model for both the amplitude and phase functions, which could be modelled with an appropriate probability density function and a linear rational function, respectively. An analytical model could then be combined with a modal model, to create a highly realistic, parameterised synthesis technique.

ACKNOWLEDGMENTS

This research was funded by Queen Mary University of London. The author would also like to thank Toontrack, and all those involved in creating the meticulously sampled library of drums sounds found in Superior Drummer 3, which has served as an invaluable dataset. Thanks are also given to all those who participated in the listening test.

Ahmed, N., Natarajan, T., and Rao, K. R. (**1974**). "Discrete cosine transform," IEEE Transactions on Computers **C-23**(1), 90–93, doi: 10.1109/T-C.1974.223784.

Asmara, R. A., Agustina, R., and Hidayatulloh (**2017**). "Comparison of discrete cosine transforms (dct), discrete fourier transforms (dft), and discrete wavelet transforms (dwt) in digital image watermarking," International Journal of Advanced Computer Science and Applications **8**, 245–249.

Avanzini, F., and Marogna, R. (**2010**). "A modular physically based approach to the sound synthesis of membrane percussion instruments," IEEE Transactions on Audio, Speech, and Language Processing **18**(4), 891–902.

Avanzini, F., and Marogna, R. (**2012**). “Efficient synthesis of tension modulation in strings and membranes based on energy estimation,” J. Acoust. Soc. Am. **131**, 897–906.

Bilbao, S. (**2012**). “Time domain simulation and sound synthesis for the snare drum,” J. Acoust. Soc. Am. **131**(1), 914–925.

Bilbao, S., Desvages, C., Ducceschi, M., Hamilton, B., Harrison-Harsley, R., Torin, A., and Webb, C. (**2020**). “Physical modeling, algorithms, and sound synthesis: The ness project,” Computer Music Journal **43**(2-3), 15–30.

Bilbao, S., and Webb, C. (**2013**). “Physical modeling of timpani drums in 3d on gpgpus,” J. Audio Eng. Soc **61**(10), 737–748.

Brown, A., ed. (**2014**). *Music Technology and Education* (Routledge, New York, USA).

Cannam, C., Landone, C., and Sandler, M. (**2010**). “Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files,” in *Proceedings of the ACM Multimedia 2010 International Conference*, Firenze, Italy, pp. 1467–1468.

Chambelin, H. (**1980**). *Musical Applications of Microproccessors*, 1st ed. (Haydens Books, New York, USA).

Collins, M., ed. (**2003**). *A Professional Guide to Audio Plug-ins and Virtual Instruments*, 1st ed. (Focal Press, New York, USA).

Errede, S. “Vibrations of ideal circular membranes (e.g. drums) and circular plates” Available at https://courses.physics.illinois.edu/phys406/sp2017/Lecture_Notes/P406POM_Lecture_Notes/P406POM_Lect4_Part2.pdf (Last viewed May 27, 2020).

Fletcher, H., and Bassett, I. G. (**1969**). “Quality of bass drum tones,” The Journal of the Acoustical Society of America **45**(1), 313–314.

Fletcher, H., and Bassett, I. G. (**1975**). "Analysis and synthesis of bass drum tones," The Journal of the Acoustical Society of America **58**(S1), S131–S131.

International Telecommunication Union (**2011**). *Algorithms to measure audio programme loudness and true-peak audio level* (Broadcasting Services series), iTU-R BS.1770-4.

ISO 4020:2001. "Road vehicles. Fuel filters for diesel engines. Test methods" (International Organization for Standardization, Geneva, Switzerland, 2001).

Jillings, N., Moffat, D., De Man, B., and Reiss, J. D. (**2015**). "Web Audio Evaluation Tool: A browser-based listening test environment," in *12th Sound and Music Computing Conference*.

Kirby, T., and Sandler, M. (**2020**). "Advanced fourier decomposition for realistic drum synthesis," in *Proceedings of the 23rd International Conference on Digital Audio Effects*, Vienna, Austria, pp. 155–162.

Marogna, R., and Avanzini, F. (**2009**). "Physically-based synthesis of nonlinear circular membranes," in *Proceedings of the 12th International Conference on Digital Audio Effects*, Como, Italy, pp. 373–379.

Marple, L. (**1999**). "Computing the discrete-time "analytic" signal via fft," IEEE Transactions on Signal Processing **47**(9), 2600–2603, doi: 10.1109/78.782222.

Nistal, J., Lattner, S., and Richard, G. (**2020**). "Drumgan: Synthesis of drum sounds with timbral feature conditioning using generative adversarial networks," in *Proceedings of the 21st ISMIR Conference, Montreal*, Vol. 1, pp. 590–597.

Pastiadis, C., and Papanikolaou, G. (**2004**). "Discrete cosine transform based de-noising of glottal pulses," Archives of Acoustics **29**(1).

Ramakrishnan, A., Abhiram, B., and Mahadeva Prasanna, S. (**2015**). "Voice source characterization using pitch synchronous discrete cosine transform for speaker identification," The Journal of the Acoustical Society of America **137**(6), EL469–EL475.

Rao, K. R., and Yip, P. (**2014**). *Discrete cosine transform: algorithms, advantages, applications* (Academic press).

Risset, J., and Wessel, D. (**1982**). *The Psychology of Music*, Chap. Exploration of timbre by analysis and synthesis, 37–39 (Academic Press, New York, USA).

Rodet, X., and Depalle, P. (**1992**). "Spectral envelopes and inverse fft synthesis," in *Audio Engineering Society Convention 93*, Paper 3393.

Rossi, L., and Girolami, G. (**2001**). "Instantaneous frequency and short term fourier transforms: Application to piano sounds," The Journal of the Acoustical Society of America **110**(5), 2412–2420.

Sandler, M. (**1990**). "Analysis and synthesis of atonal percussion using high order linear predictive coding," Applied Acoustics **30**(2), 247–264.

Skrodzka, E., Hojan, E., and Proksza, R. (**2006**). "Vibroacoustic investigation of a batter head of a snare drum," Archives of Acoustics **31**, 289–297.

Toontrack. "Superior drummer 3" Information available at https://www.toontrack.com/superior-line/ (Last viewed May 27, 2020).

Torin, A. (**2016**). "Percussion instrument modelling in 3d: Sound synthesis through time domain numerical simulation," Ph.D. thesis, University of Edinburgh.

Torin, A., and Newton, M. (**2014**). "Nonlinear effects in drum membranes," in *Proceedings of the International Symposium on Musical Acoustics*, pp. 107–112.

Trautmann, L., Petrausch, S., and Rabenstein, R. (**2001**). "Physical modeling of drums by transfer function methods," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*, Salt Lake City, USA, Vol. 5, pp. 3385–3388 vol.5.

Zappi, V., Allen, A., and Fells, S. (**2017**). "Shader-based physical modelling for the design of massive digital musical instruments.," in *NIME*, pp. 145–150.