LATENT FORCE MODELS FOR SOUND: LEARNING MODAL SYNTHESIS PARAMETERS AND EXCITATION FUNCTIONS FROM AUDIO RECORDINGS

William J. Wilkinson, Joshua D. Reiss and Dan Stowell

Centre for Digital Music
Queen Mary University of London
London, UK
w.j.wilkinson@qmul.ac.uk

ABSTRACT

Latent force models are a Bayesian learning technique that combine physical knowledge with dimensionality reduction — sets of coupled differential equations are modelled via shared dependence on a low-dimensional latent space. Analogously, modal sound synthesis is a technique that links physical knowledge about the vibration of objects to acoustic phenomena that can be observed in data. We apply latent force modelling to sinusoidal models of audio recordings, simultaneously inferring modal synthesis parameters (stiffness and damping) and the excitation or contact force required to reproduce the behaviour of the observed vibrational modes. Exposing this latent excitation function to the user constitutes a controllable synthesis method that runs in real time and enables sound morphing through interpolation of learnt parameters.

1. INTRODUCTION

Modal synthesis aims to reproduce the behaviour of the vibrational modes of a sounding object, through consideration of its physical properties [1]. If all the required physical properties are known, then the frequency and amplitude of the modes can be calculated. Alternatively, by taking the Fourier transform of a recording of the sounding object, we can observe these same features empirically. Hence we have a clear link between the physics of vibrating objects and observable acoustic behaviour. This has often been exploited to construct models for sound synthesis that provide users with both physical and phenomenological control [2, 3, 4].

In [4], modal synthesis parameters were learnt automatically from recordings of impact sounds by assuming the excitation force to be an impulse and inferring the modes' mass, stiffness and damping coefficients from data. Others have constructed detailed physical models for source-filter interaction, and set the filter parameters corresponding to observed peaks in the frequency spectrum [2, 5].

Recent work in the machine learning community, namely the development of latent force models (LFM), has shown that it is possible to build a model which incorporates physical knowledge and to fit it to data via an inference procedure [6]. We adopt this approach to formally use learnings from audio recordings to construct a simple mechanistic model for modal synthesis that is generalisable to a large class of sounds.

Our framework for synthesis utilises sinusoidal analysis [7, 8] to track modes over time, and makes assumptions about the behaviour of modes by representing their amplitude with first order ordinary differential equations (ODEs). The introduction of such ODEs into the model prior, a latent force model, allows us to infer both the system parameters and the excitation function required to reproduce the observed outputs. It does so by coupling the modes' amplitudes through consideration of their common dependence on a low-dimensional latent space, in this case the one-dimensional excitation function. The result is a real-time synthesis model that allows for user control and sound morphing. Interactive sound examples and MATLAB code for latent force modelling of sinusoidal amplitude data are provided.[†]

We formulate our problem in Section 2. In Section 3 we summarise the relevant literature relating to sound synthesis and latent force models. In Section 4 we present our approach to the application of latent force modelling to audio. Section 5 outlines how our approach can be utilised to perform real-time synthesis and sound morphing, and Section 6 presents empirical results and case studies.

2. PROBLEM FORMULATION

Consider M modes of vibration of a sounding object, for which we obtain observation data from sinusoidal analysis of an audio recording. We assume the frequencies f_i of the modes to be fixed and that the amplitudes $x_i(t)$ are modelled by exactly one excitation function $u(t)$ being fed through an idealised physical system:

$$\frac{dx_i(t)}{dt} + D_i g_x(x_i(t)) = S_i g_u(u(t)), \quad i = 1, ..., M, \quad (1)$$

where coefficients D_i and S_i relate to physical properties of the i^{th} mode, with g_x and g_u being potentially nonlinear functions of outputs x_i and input u respectively.

The task is to fit our data to this model in such a way that we can infer all the system parameters $\{S_i, D_i\}_{i=1}^{M}$ and predict the behaviour of $u(t)$. Doing so constitutes transformation of the data to a one-dimensional control space. With resynthesis in mind, we must encourage realistic parameters relating to stiffness and damping of the modes to be learnt, and require the predicted behaviour of $u(t)$ to be interpretable as physical energy driving the system.

After the model has been fit, the output audio signal Y can be synthesised through summation of sinusoids with the reconstructed amplitudes (and initial phase ϕ_i):

$$Y(t) = \sum_{i=1}^{M} x_i(t) sin(2\pi f_i t + \phi_i). \quad (2)$$

[†] http://c4dm.eecs.qmul.ac.uk/audioengineering/latent-force-synthesis

3. BACKGROUND

3.1. Sound Synthesis

Physics-based approaches to sound synthesis vary from detailed numerical simulation of the sound production mechanism represented by differential equations [9, 10], to standard digital filtering techniques informed by those same differential equations [5, 11]. These approaches require significant knowledge regarding the complex interactions that produce sound, and as such are limited to systems for which much of the pertinent physics are known.

Modal synthesis is a more generalisable, physically-inspired approach which typically represents the vibrational modes of a sounding object as a set of decoupled second-order differential equations, also known as mass-spring-damper systems [1, 2]. The forced mass-spring-damper corresponding to the i^{th} mode has coefficients relating to mass m_i, springiness (or stiffness) k_i and damping b_i:

$$m_i \frac{d^2 X_i(t)}{dt^2} + b_i \frac{dX_i(t)}{dt} + k_i X_i(t) = u(t), \tag{3}$$

where $u(t)$ is the forcing function that excites the system. The exact sound production mechanism is not modelled in full detail. Instead it is assumed that sound is produced through the vibration of an object or column of air, and that the frequency and relative amplitude of these vibrations can be predicted based on mass, stiffness and damping parameters determined by the physical properties of the object.

The solution to these mass-spring-damper systems is a bank of modes,

$$X_i(t) = x_i(t) sin(2\pi f_i t + \phi_i), \tag{4}$$

with time-varying amplitude $x_i(t)$, frequency f_i and initial phase ϕ_i, referred to as damped sinusoids, or oscillators. In traditional modal synthesis $u(t)$ is assumed to be an impulse, and we obtain the solution $x_i(t) = \alpha_i e^{-\beta_i t}$ where α_i and β_i are the amplitude and damping of the mode respectively. If we allow $u(t)$ to be unconstrained, then no analytical solution for the amplitude exists. In the present work we will constrain $u(t)$ by placing a Bayesian prior on its possible values (Section 3.2).

Sinusoidal modelling [7, 8] is an analysis-synthesis technique that compartmentalises a sound into its deterministic and stochastic components, and models the deterministic part as a sum of sinusoids such as those in equation (4). Energy is tracked through sequential frames of the Short Time Fourier Transform to create "partials" — sinusoids with frequency and amplitude that can vary over time.

Links between physical models and statistical behaviour have been exploited in the past to design hybrid synthesis frameworks that learn sound characteristics from data whilst enabling control through spectral transformation [4] or by learning a mapping between computed audio descriptors and a performed control space [12]. Our approach is to view sinusoidal data as the output of a series of digital filters representing the amplitudes $x_i(t)$ of the physical modes. This motivates the introduction of such filters (in ODE form) into the prior assumptions for a machine learning algorithm looking to infer knowledge from audio recordings.

3.2. Latent Force Models

Latent force models [6] are a probabilistic approach to modelling data which assumes that M observed output functions are produced by some $R < M$ unobserved (latent) functions being forced through a set of differential equations. If this set of differential equations represents some physical behaviour present in the system we are modelling, even if only in a simplistic manner, then such a technique can improve our ability to perform inference from data [13, 14]. This is achieved by placing a Gaussian process prior [15] over the R latent functions, calculating the cross-covariances by solving the ODEs, and performing regression.

Standard latent force modelling involves batch processing of data using prediction equations that involve inversion of large covariance matrices. This motivates the reformulation of the system into its state space representation which allows for inference on sequential time points [16]. This also gives us an intuitive form with which to perform resynthesis (Section 4.3).

The aim here is to construct a joint model which incorporates all of our ODE parameters and our assumptions about the input. From this point onwards we assume $R = 1$, since we are attempting to model a one-dimensional excitation force. The introduction of additional forces is straightforward, but not explored here.

Suppose we can describe the i^{th} output x_i by this linear first-order ODE:

$$\frac{dx_i(t)}{dt} + D_i x_i(t) = S_i u(t). \tag{5}$$

We must now assume that $u(t)$ can be modelled by a linear time invariant (LTI) stochastic differential equation (SDE) of the form

$$\frac{d^p u(t)}{dt^p} + a_{p-1}\frac{d^{p-1}u(t)}{dt^{p-1}} + ... + a_1 \frac{du(t)}{dt} + a_0 u(t) = w(t), \tag{6}$$

where p is the model order and $w(t)$ is a white noise process. If the covariance function chosen as part of the Gaussian process assumption cannot be written in this form with finite p, then approximations must be used. Here we choose $p = 3$, which is sufficient to represent the Matérn covariance function [15].

The joint state space model is constructed by inserting the coefficients of (5) and (6) into the transition matrix for a stable Markov process driven by $w(t)$:

$$\frac{d\boldsymbol{x}(t)}{dt} = F\boldsymbol{x}(t) + Lw(t), \tag{7}$$

where, if $\dot{u}$ represents the first differential of u w.r.t t,

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_M \\ u \\ \dot{u} \\ \ddot{u} \end{bmatrix}, F = \begin{bmatrix} -D_1 & 0 & 0 & S_1 & 0 & 0 \\ 0 & \ddots & 0 & \vdots & 0 & 0 \\ 0 & 0 & -D_M & S_M & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -a_0 & -a_1 & -a_2 \end{bmatrix},$$

$$L = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

This state space model includes all the necessary parameters, and we discretise it using standard techniques involving calculation of the matrix exponential. Its discrete form is

$$\boldsymbol{x}[t_k] = \hat{F}[\Delta t_k]\boldsymbol{x}[t_{k-1}] + \boldsymbol{q}[t_{k-1}], \quad \boldsymbol{q}[t_{k-1}] \sim N(\boldsymbol{0}, Q[\Delta t_k]), \tag{8}$$

where k is the time index, $\hat{F}$ is the transition matrix calculated using the matrix exponential of F, and Q is the process noise matrix

calculated using the spectral density of $w(t)$. Δt_k is the discrete time step size. Our output measurement model now becomes

$$\boldsymbol{y}[t_k] = H\boldsymbol{x}[t_k] + \epsilon[t_k], \quad \epsilon[t_k] \sim N(0, \sigma^2), \tag{9}$$

where H is the measurement matrix that simply selects the outputs from the joint model.

This form allows us to calculate the filtered (i.e. backwards-looking) posterior distribution $p(\boldsymbol{x}[t_k] \mid \boldsymbol{y}[t_{1:k}], \theta)$ of the state $\boldsymbol{x}[t_k]$ given observations $\boldsymbol{y}[t_{1:k}]$ and hyperparameters θ, for $k = 1, ..., T$, through application of Kalman filtering using the standard Kalman update equations [17]. Furthermore, we can also calculate the smoothing (i.e. backwards- and forwards-looking) posterior $p(\boldsymbol{x}[t_k] \mid \boldsymbol{y}[t_{1:T}], \theta)$ using the Rauch-Tung-Streibel smoother. The implementation of these combined sequential techniques is equivalent to Gaussian process regression [14, 18].

Kalman filtering therefore provides us with a method for sequentially estimating the state of the outputs *and* the latent inputs at each point in time given our data and the hyperparameters θ, which now include the ODE parameters. This sequential method provides a large efficiency gain over standard batch processing, and the Kalman filter equations also provide the necessary components to calculate the marginal data likelihood,

$$p(\boldsymbol{y}[t_{1:n}]|\theta) = \prod_{i=1}^{n} p(\boldsymbol{y}[t_i] \mid \boldsymbol{y}[t_{1:i-1}], \theta). \tag{10}$$

The usual approach to inference is to iteratively optimise θ by maximising this equation with gradient-based methods.

3.2.1. Nonlinear latent force models

During the prediction stage of Kalman filtering, we calculate the required cross-covariances between the outputs and the latent function by solving the necessary differential equations. However, these calculations are only tractable if our model is linear.

Consider the ODE presented in our problem formulation (1), in which nonlinear functions act on both x_i and $u(t)$. We can similarly construct the LTI SDE form of this model by again constructing a joint state vector $\boldsymbol{x}(t)$ such that

$$\frac{d\boldsymbol{x}(t)}{dt} = \boldsymbol{g}(\boldsymbol{x}(t), t) + L(\boldsymbol{x}(t), t)w(t). \tag{11}$$

However, exact calculation of the Kalman prediction equations in this case is not possible. Instead, the filtering and smoothing distributions are approximated with Gaussian distributions and numerically computed with cubature integration methods [19].

4. LATENT FORCE MODELS FOR SOUND

The Spear software [8] is used to obtain the sinusoidal partials from an audio recording. We then apply the above latent force modelling techniques to map the high-dimensional sinusoidal data to a controllable, one-dimensional latent function. In order for synthesis to be intuitively controllable, parameters must be physically meaningful and the learnt latent function must also be interpretable in a physical sense.

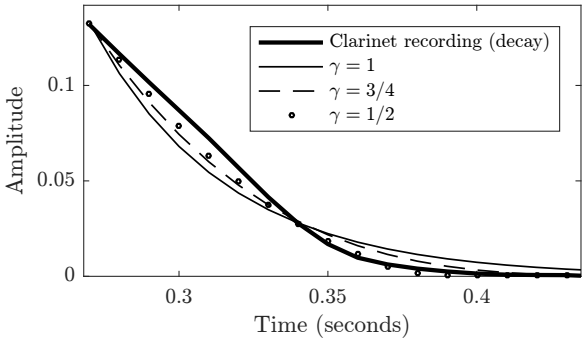

Figure 1: *Comparison of amplitude model choice: $\gamma = 1$ represents the standard model for the amplitude of a sinusoid. Selecting $\gamma < 1$ alters the decay behaviour to more closely represent the real data obtained from the decay section of the second harmonic of a recording of a clarinet.*

4.1. Modelling the Amplitude Data

Our approach is to consider M vibrational modes of a resonating object, modelled as in equation (4), assuming the modes have fixed frequencies. Given this assumption the problem becomes how to model the amplitude of the modes, $x_i(t), i = 1, ..., M$.

The analytical solution when $u(t)$ is an impulse is $x_i(t) = \alpha_i e^{-\beta_i t}$. This inverse exponential equation can be modelled with a linear first-order ODE obtained by removing the second-order term from the mass-spring-damper system (3). By doing so we obtain equation (5), where $D_i = k_i/b_i$ and $S_i = 1/b_i$ are physically relevant parameters related to damping and stiffness of the system.

In practice, when observing real amplitude data (for which $u(t)$ will never truly be an impulse), we found that partials tend to decrease in a more linear fashion than can be described by equation (5). Therefore we propose an alternative model containing a parameter γ which alters the "linearity" of the decay of the signal,

$$\frac{dx_i(t)}{dt} + D_i x_i^{\gamma}(t) = S_i u(t). \tag{12}$$

We found that a suitable range of values for representing real audio data was $\gamma \in [\frac{1}{2}, 1]$, where a reduction in γ increases the linearity of the decay. $\gamma < 1/2$ represents an almost straight line, whilst $\gamma > 1$ would mean the data may never reduce to zero. No formal method for selecting γ is presented here, instead we visually inspect the amplitude data and select an appropriate value based on the decay behaviour. Figure 1 shows the comparison between different choices of γ.

Since predicted values of x_i can go negative, raising our x_i term to the power of $\gamma < 1$ can give unwanted complex results. Therefore in practice we take the real part of the x_i term. This compromises the smoothness of the model, but inference is still possible with the nonlinear filtering approach outlined in Section 3.2.1, numerically approximating the solutions to these equations rather than solving them analytically.

We aim to learn meaningful parameters representing damped modes which reduce to zero in the absence of input. As such it is advantageous for us to enforce a positivity constraint on input $u(t)$ via a function g. This has two major benefits. Firstly, the new excitation force $g(u(t))$ becomes interpretable as a physical entity; positive energy driving the system. Secondly, it encourages

the optimiser to learn damping coefficients D_i that are more physically realistic (i.e. larger / more damped), since they must enable the system to reduce to zero when $g(u(t)) = 0$, whereas in the unconstrained case this could be achieved via negative inputs rather than damping.

A reliable positivity constraint that ensures smoothness is the "softplus" rectification function,

$$g(u(t)) = \log(1 + e^{u(t)}). \tag{13}$$

Introducing this nonlinearity gives us our final model for the amplitude of the i^{th} damped vibrational mode of a sounding object:

$$\frac{dx_i(t)}{dt} + D_i \text{Re}\left\{x_i^{\gamma}(t)\right\} = S_i g(u(t)), \tag{14}$$

which is the target system formulated in (1) with $g_x(x_i) = \text{Re}\left\{x_i^{\gamma}\right\}$ and $g_u(u) = g(u)$.

4.2. Selecting the Modes

Optimising our parameters in the latent force model framework is a high-dimensional problem, since we have parameters D_i and S_i (and the initial conditions) to estimate for all M outputs, in addition to the hyperparameters of the Gaussian process kernel for the latent input (we use the Matérn covariance function). As such it is common for optimisation to get stuck in local minima, and choice of initial parameter settings can significantly affect the optimality of our outcome.

Furthermore, we assume our outputs (the modes) to be strongly correlated, such that a mapping to a low-dimensional space that maintains much of their behaviour exists. The introduction of partials that don't represent vibrational modes could compromise this assumption, in turn compromising the model's ability to represent the system.

We must therefore identify which partials in the sinusoidal model are representative of the vibrational modes. If our analysis signal has strong harmonic content (musical instruments, for example), then picking the modes / harmonics is straightforward. For inharmonic sounds (such as a hammer striking a metal plate), energy is distributed across the sinusoidal model, and there may be a strong noise component. In this case, selecting the modes is not as simple as selecting the largest M partials. In Figure 2, we analyse the frequency spectrum of the signal, designing a filter based on the shape of the spectrum. We invert the filter to flatten the data, allowing us to pick the modes of vibration from the peaks of the filtered spectrum.

Once we have selected our M modes, we scale the observed amplitude data to normalise their weighting prior to inference. Note that it is possible to assign importance to particular modes by altering the observation noise assumptions for a particular dimension of the Kalman filter. We calculate the median frequency value for each partial, and treat their frequency as fixed from this point onwards. Inference on the amplitude data is now performed using the techniques outlined in Section 3.2 with the model in equation (14).

4.3. Resynthesis with the State Space Model

After inference is performed, we obtain an optimised set of parameters θ, and a posterior distribution over the outputs and the latent input. We apply an inverse scaling operation to obtain the original magnitude weightings. The posterior distribution provides us with

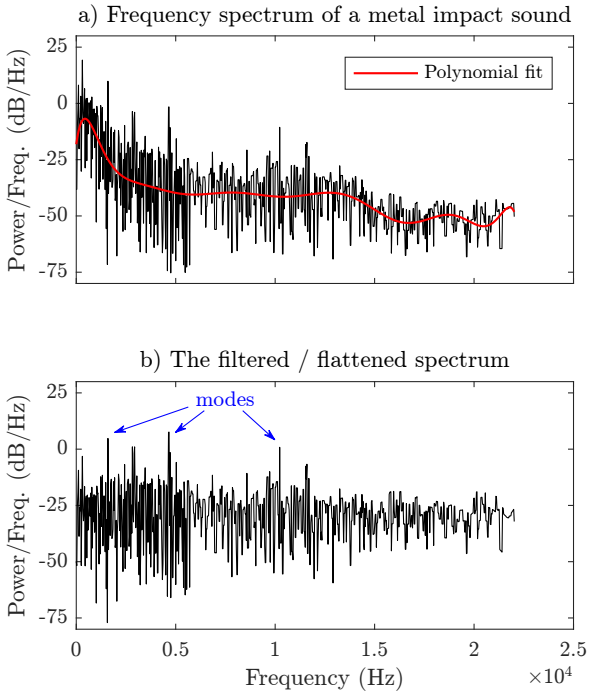

Figure 2: *A filter is designed by fitting a polynomial to the shape of the frequency spectrum. The filter is inverted and applied to the signal to flatten the spectrum. Peaks in the flattened spectrum are then used to pick the vibrational modes of the signal.*

information about the uncertainty of the prediction, and we can compare the posterior mean of the outputs to the analysis data to evaluate how much of the amplitude behaviour has been encoded.

Drawing samples from the distribution over the latent excitation function and passing them through the model constitutes resynthesis. Alternatively, to reproduce outputs faithful to the analysis data, we can pass the posterior mean through the model. To do so, we discretise equation (14) and restate it in state space form, solving it using the Euler method. The i^{th} output is therefore given by the discrete model

$$\begin{aligned}\begin{bmatrix} x_i[t_k] \\ \dot{x}_i[t_k] \end{bmatrix} &= \begin{bmatrix} 1 & \Delta t_k \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_i[t_{k-1}] \\ \dot{x}_i[t_{k-1}] \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 \\ -D_i \end{bmatrix} x_i^{\gamma}[t_{k-1}] + \begin{bmatrix} 0 \\ S_i \end{bmatrix} g[u[t_k]],\end{aligned} \tag{15}$$

where Δt_k is the time step size, chosen to be identical to the analysis step size in equation (8).

5. EXPRESSIVE REAL TIME SYNTHESIS AND SOUND MORPHING

An advantage of using a relatively simple state space model such as the one in equation (15) is its flexibility with regards to parameter control and time step size. We now illustrate how we can utilise these features to run our model in real time with user control, and to interpolate between parameter values to manipulate the sound timbre.

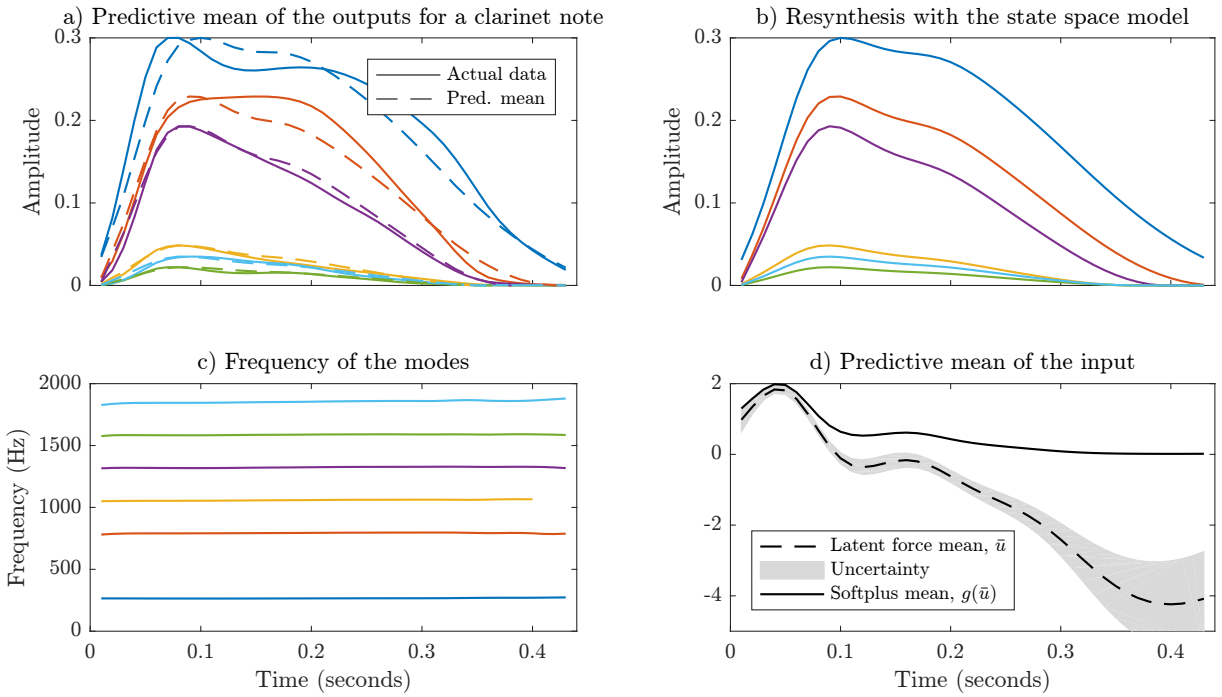

Figure 3: *Latent force modelling of a clarinet note. 6 modes are picked based on their amplitude, and the predictive mean of the output distribution is compared to the real data (top left). The frequency data (bottom left) shows the modes are, in order of magnitude, the 1st, 3rd, 5th, 4th, 7th and 6th harmonics. The mean, $\bar{u}$, and 95% confidence interval (uncertainty) of the latent input u is shown (bottom right). $g(\bar{u})$ is fed through the state space model to resynthesise the output (top right). Low uncertainty results in resynthesis very similar to the predictive mean.*

5.1. Real Time Synthesis

In the previous section Δt_k was fixed at the analysis time step size, corresponding to framewise modelling. During synthesis we can set the step size to be as large or small as required. Based on our desired sampling frequency, we modify Δt_k such that the model calculates sample-rate data and runs in real time.

This modification allows us to handle audio-rate input, which may be crucial for a synthesis model that requires expressive user control. As mentioned in Section 4.3, resynthesis can be performed by sampling from the posterior distribution over the latent excitation function and passing the sample through the model. However, with the aim of user-controllable synthesis in mind, and given that the excitation function is interpreted as physical energy forcing the system, it is possible to replace the mean of the latent distribution with a new function dependent on some user input.

We control the synthesis model with user input data corresponding to the pressure applied to a MIDI CC button or a force-sensing-resistor, scaling the data appropriately such that it has similar properties to the learnt latent input. Alternatively, we provide the user with a modifiable plot of the excitation function, which they can re-draw and modify to create new sounds.

5.2. Sound Morphing

Our linear time-invariant synthesis model has fixed stiffness and damping parameters corresponding to each mode. Adjusting these parameters has an impact on perceptual characteristics relating to timbre such as attack time, decay time and the modes' amplitudes relative to one another. Individual modification of these parameters is possible, but not desirable if we wish to maintain coherence across dimensions. Instead, we interpolate parameters between models to create new sound timbres not present in the original recordings.

Prior to parameter interpolation we match the modes between models by ranking them in order of frequency. We also normalise the magnitude of the excitation functions, adjusting the stiffness parameters accordingly. For sounds without definable harmonic structure, pairing the modes is straightforward and simply based on their rank position. For harmonic sounds we must be careful to match the n^{th} harmonic in model *A* to the n^{th} harmonic in model *B*. If we fail to do so, interpolation of the frequency value will compromise the harmonic structure of the sound.

Once modes have been paired we perform linear interpolation of physical parameters S_i, D_i and the initial conditions, and logarithmic interpolation of the frequency. Synthesis in this manner negates the need for time-domain modification (such as time-stretching) usually associated with morphing [20].

6. RESULTS

In order to show the versatility of our approach we consider two case studies: musical instruments, demonstrated here by a short clarinet note, and impact sounds, demonstrated by the sound of metal being struck by a solid object. We then measure the accuracy of our reconstructed data for a number of recordings, and show the output produced by morphing between two different sounds.

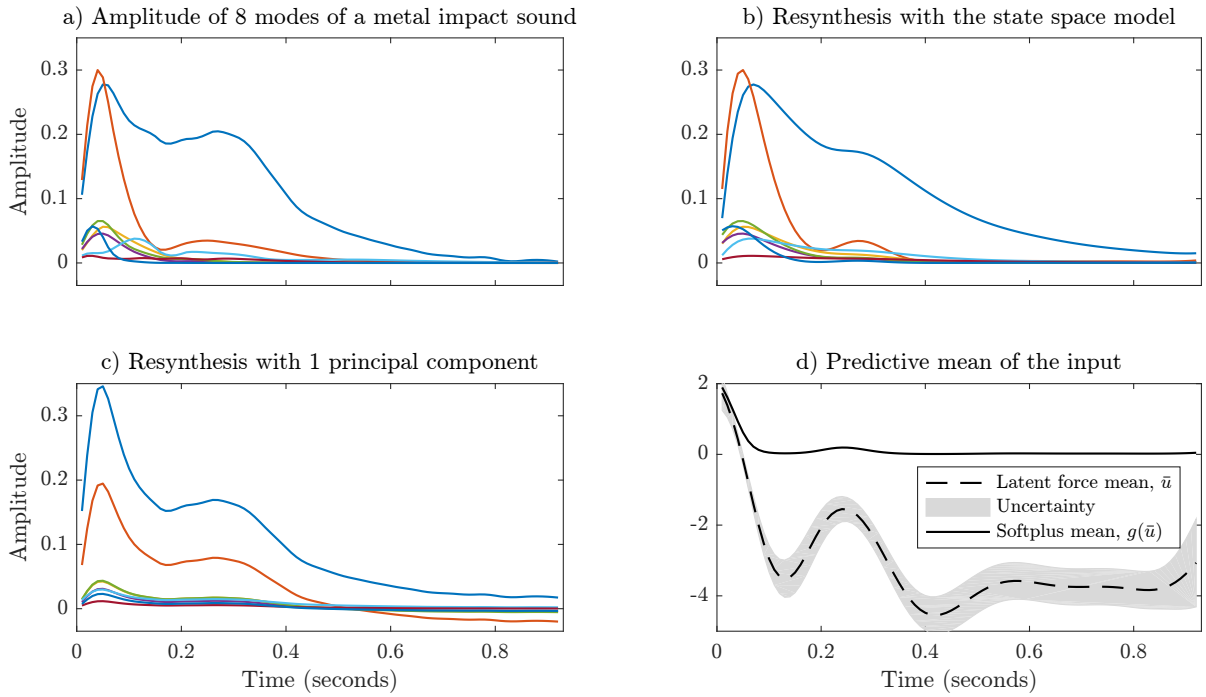

Figure 4: *Latent force modelling of a metal impact sound. The real analysis data shows some variation in behaviour between modes (top left). An increase in uncertainty in the posterior distribution after 0.1s reflects this fact (bottom right). The posterior mean of the latent distribution, $\bar{u}$, is fed through the state space model, and the result shows that much of the variable behaviour was captured (top right). PCA results are shown as a comparison, and we can see that the variable damping rates have not been reproduced (bottom left).*

6.1. Musical Instruments

Most musical instruments have strong harmonic structure, and the majority of the signal energy tends to be contained within relatively few sinusoids representing these harmonics. By inspecting the data and experimentally testing the results for various values for γ, we found that musical instruments tend to have a relatively linear decay, and a choice of $\gamma = 1/2$ fits the data best.

Figure 3 shows the results of latent force modelling of a clarinet note. The first 6 modes are considered, and on viewing the mean of the distribution of the outputs (Figure 3a), we can see that much of the behaviour has been captured in the model. The attack of the largest mode is partially altered to fit the shape of the other modes, since the simple mechanistic model struggles to encode peaks that are out of phase with each other. However, the variable damping rates have successfully been learnt, with the largest mode reducing to zero at a much slower rate than the smallest modes.

We plot the 95% confidence interval for the latent input (Figure 3d), and observe that the uncertainty in the learnt model increases towards the end of the signal, as some partials reduce to zero and their behaviour no longer correlates with the non-zero partials. The resynthesised outputs (Figure 3b) are almost identical to the predictive mean of the outputs when passing the mean of the latent input, $\bar{u}(t)$, through the model (14). This suggests that the observed degree of uncertainty is acceptable.

6.2. Impact Sounds

Impact sounds often lack clear harmonic structure, and energy is distributed across the frequency spectrum. In selecting just a small number of modes, we risk losing much of the audio content. However, our selected modes are capable of reproducing much of the deterministic character of the signal. The remainder is treated as a residual, and not addressed here. We found that for impact sounds a model choice of $\gamma = 3/4$ was more appropriate since the decay rate varies as the amplitude decreases (in Figure 4a, the partials' gradient flatten out over time).

Figure 4a shows that for a metal impact sound large variation of behaviour occurred between modes. To account for this, a large variation of stiffness and damping parameters were learnt, enabling much of the behaviour to be captured. Comparing the synthesised outputs for the two largest modes in Figure 4b, we see that whilst they have a similar attack, encoded by the stiffness or sensitivity measure S_i, they have a very different decay, encoded by the damping measure D_i.

Uncertainty in the metal impact model (Figure 4d) increased more quickly than in the clarinet model, reflecting the fact that behaviour is less consistent across these vibrational modes than across the harmonics of the clarinet. In particular we observe an increase in uncertainty after the initial attack, when the modes' behaviour begins to diverge from one another.

6.3. Model Accuracy and Comparison with PCA

To evaluate our results we calculated the root-mean-square (RMS) error between the actual data and our synthesised outputs. This gives us a measure of our ability to reproduce the analysed sinusoidal partials. Readers are also invited to listen to the sound examples provided.

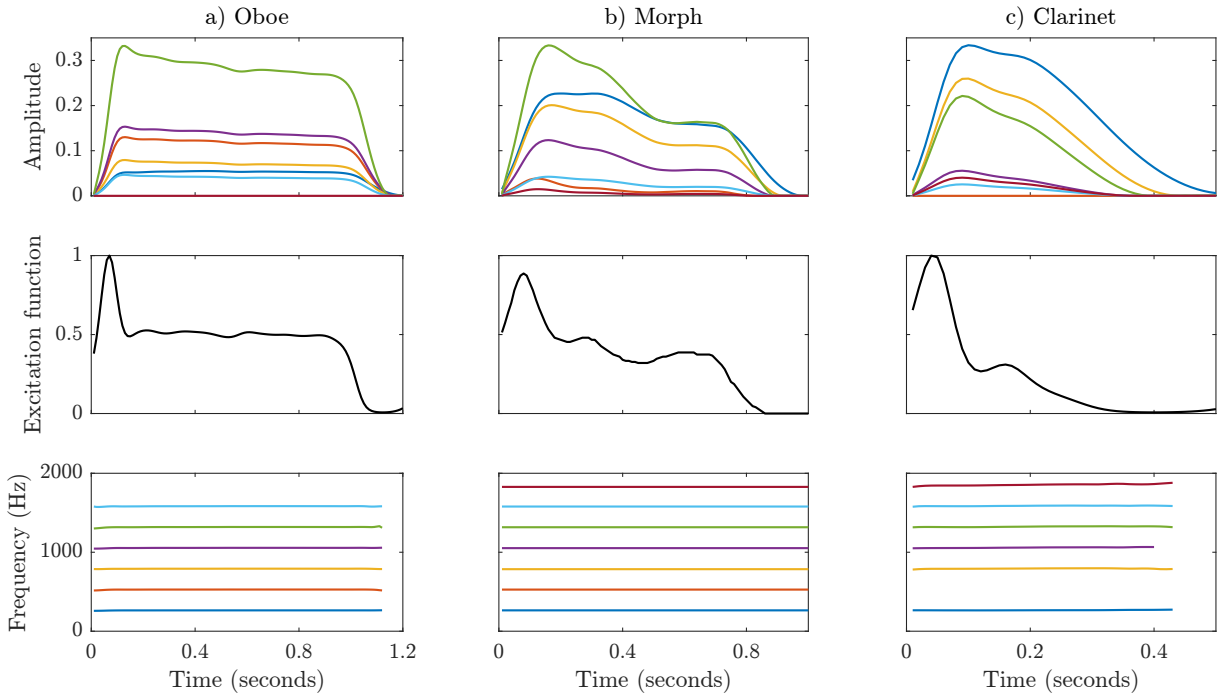

Figure 5: *Sound morphing between an oboe and a clarinet. The modes of an oboe (left) are matched with the modes of a clarinet (right) and colour-coded based on their pairings. Since the modes represent harmonics, it is important to maintain the harmonic structure, so the 2nd mode of the oboe does not have a match. Similarly, the 6th mode of the clarinet is not matched. Stiffness and damping parameters are interpolated, and a user-drawn excitation function of arbitrary length is used to produce the morphed output (middle).*

As a comparison, we run principal component analysis (PCA) on our amplitude data. PCA is another dimensionality reduction technique that similarly maps high-dimensional data to a lower-dimensional space through an input-output process (a simple scalar weighting), providing us with a set of orthogonal variables, called principal components, ranked in order of how much of the data's variance they describe.

Latent force modelling has many benefits over PCA, such as physical interpretability, model memory (PCA is an instantaneous mapping), the ability to introduce nonlinear mappings between inputs and outputs, and a probabilistic framework for calculating uncertainty and resampling new data (although probabilistic PCA techniques also exist). Regardless, PCA is a worthwhile comparison due to its simplicity and reliability.

Figure 4c shows the results of PCA on a metal impact sound. Using just one principal component to reproduce the 8-dimensional output fails to capture much of the behaviour, most notably the variable damping rates. With the one-dimensional LFM we are able to capture much more of the behaviour (Figure 4b). Note that it is possible to introduce more principal components, and also possible to run latent force modelling with more than one latent dimension, but this violates our assumption that the modes are produced by a common excitation function.

Table 1 compares the RMS error for latent force modelling and PCA for a number of audio recordings. The data is normalised to give equal weighting to each dimension of the model. When disparate behaviour occurs across dimensions, latent force modelling is more accurate than reconstruction with one principal component. For recordings in which the dimensions have high correlation, such as the oboe, even one principal component sometimes outperformed the latent force model. This poor performance of the LFM for the oboe could be due to the optimisation procedure converging on a sub-optimal local minimum, or due to the fact that the oboe partials reduce to zero at an almost linear rate, and their behaviour was not fully captured by our choice of $\gamma = 1/2$, i.e. a more optimal choice of γ exists.

RMS error		
Audio recording	LFM	PCA
Clarinet	**0.0325**	0.0593
Oboe	0.0189	**0.0156**
Piano	**0.0441**	0.0520
Metal impact	**0.0377**	0.0609
Wooden impact	**0.0139**	0.0291

Table 1: *Root-mean-square (RMS) error between modal amplitude data and outputs of latent force modelling (LFM) and principal component analysis (PCA). The LFM outperforms PCA when disparate behaviour across dimensions is observed.*

6.4. Morphing

Figure 5 shows the results of sound morphing between recordings of an oboe and a clarinet. A user-drawn excitation function is used as input to the morphed model (Figure 5b) and we observe the expected change in relative amplitudes. The modes of the oboe have

much faster decay times than the clarinet, and visual inspection of the morphed sound confirms that decay rates in between these two extremes are achieved.

7. CONCLUSIONS

The aim of this work was to demonstrate our ability to learn about the physical behaviour of sound from recordings. Such an approach will aid those looking to design and build synthesis models that are faithful to the real-world sounds we hear around us, whilst also providing opportunities for control and expression.

We utilised knowledge about the way in which objects vibrate to produce sound to construct a simple mechanistic model for the behaviour of sinusoidal modes. Although this model does not describe all the physical interactions that create sound, its simplicity enables the application of nonlinear latent force modelling techniques to infer physically relevant parameters from audio recordings, in addition to the excitation required to produce meaningful output.

After the learning process was complete, we demonstrated how to perform synthesis in this framework, adapting the model to run in real time with user control. We then provided a way to manipulate sound characteristics through parameter morphing. We showed how the model often outperforms PCA when attempting to map sinusoidal data to a one-dimensional control space, but noted how higher accuracy is not guaranteed since we rely on a high-dimensional optimisation procedure to find suitable parameter values.

As future work, the inference process would benefit greatly from intelligent selection of initial conditions to aid optimisation in finding appropriate solutions. Automatic identification of linearity measure γ, or inclusion of γ as a parameter to be optimised during inference, would also be highly beneficial. The introduction of additional latent functions would allow us to model more complex systems with multiple control inputs.

Subjective evaluation of our ability to reproduce the quality of a given audio recording was not presented here, but is necessary to further assess the suitability of our approach. Complex amplitude modulation is difficult to model if the modes' peaks are out of phase with each other, and a system that allows for variable frequency would greatly improve its applicability. Finally, consideration of the residual component of the signal is crucial for further development of these techniques.

8. REFERENCES

[1] Jean Marie Adrien and Eric Ducasse, "Dynamic modeling of vibrating structures for sound synthesis, modal synthesis," in *Audio Engineering Society 7th International Conference: Audio in Digital Times*, 1989.

[2] Perry R. Cook, "Physically informed sonic modeling (PhISM): Synthesis of percussive sounds," *Computer Music Journal*, vol. 21, no. 3, pp. 38–49, 1997.

[3] Gerhard Eckel, Francisco Iovino, and René Caussé, "Sound synthesis by physical modelling with Modalys," in *Proc. International Symposium on Musical Acoustics*, 1995, pp. 479–482.

[4] Zhimin Ren, Hengchin Yeh, and Ming C. Lin, "Example-guided physically based modal sound synthesis," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 1, pp. 1, 2013.

[5] Julius O. Smith, *Physical audio signal processing: For virtual musical instruments and audio effects*, W3K Publishing, 2010.

[6] Mauricio A. Alvarez, David Luengo, and Neil D Lawrence, "Latent force models.," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009, vol. 12, pp. 9–16.

[7] Robert McAulay and Thomas Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.

[8] Michael Klingbeil, "Software for spectral analysis, editing, and synthesis.," in *International Computer Music Conference (ICMC)*, 2005.

[9] Julien Bensa, Stefan Bilbao, Richard Kronland-Martinet, and Julius O. Smith, "The simulation of piano string vibration: From physical models to finite difference schemes and digital waveguides," *The Journal of the Acoustical Society of America*, vol. 114, no. 2, pp. 1095–1107, 2003.

[10] Lutz Trautmann and Rudolf Rabenstein, "Digital sound synthesis based on transfer function models," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 1999, pp. 83–86.

[11] Perry R. Cook, *Real sound synthesis for interactive applications*, CRC Press, 2002.

[12] Alfonso Pérez Carrillo, Jordi Bonada, Esteban Maestre, Enric Guaus, and Merlijn Blaauw, "Performance control driven violin timbre model based on neural networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 1007–1021, 2012.

[13] Mauricio A. Alvarez, David Luengo, and Neil D. Lawrence, "Linear latent force models using gaussian processes," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2693–2705, 2013.

[14] Jouni Hartikainen and Simo Särkkä, "Sequential inference for latent force models," in *Twenty-Seventh Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, 2011, pp. 311–318.

[15] Carl Edward Rasmussen and Christopher K. I. Williams, *Gaussian processes for machine learning*, MIT Press, 2006.

[16] Jouni Hartikainen and Simo Särkkä, "Kalman filtering and smoothing solutions to temporal gaussian process regression models," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2010, pp. 379–384.

[17] Mohinder S. Grewal, *Kalman filtering*, Springer, 2011.

[18] Simo Särkkä, *Bayesian filtering and smoothing*, vol. 3, Cambridge University Press, 2013.

[19] Jouni Hartikainen, Mari Seppanen, and Simo Sarkka, "State-space inference for non-linear latent force models with application to satellite orbit prediction," in *29th International Conference on Machine Learning (ICML)*, 2012, pp. 903–910.

[20] Marcelo Caetano and Xavier Rodet, "Musical instrument sound morphing guided by perceptually motivated features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1666–1675, 2013.