School of Electronic Engineering and Computer Science
Queen Mary University of London

Gaussian Process Modelling for Audio Signals

William J. Wilkinson

PhD thesis

Submitted in partial fulfilment of the requirements for the Degree of
Doctor of Philosophy in Computer Science
July 2019

Statement of Originality

I, William J. Wilkinson, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged and my contribution indicated. Previously published material is also acknowledged herein.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature:

Date:

Abstract

Audio signals are characterised and perceived based on how their spectral make-up changes with time. Uncovering the behaviour of latent spectral components is at the heart of many real-world applications involving sound, but is a highly ill-posed task given the infinite number of ways any signal can be decomposed. This motivates the use of prior knowledge and a probabilistic modelling paradigm that can characterise uncertainty.

This thesis studies the application of Gaussian processes to audio, which offer a principled non-parametric way to specify probability distributions over functions whilst also encoding prior knowledge. Along the way we consider what prior knowledge we have about sound, the way it behaves, and the way it is perceived, and write down these assumptions in the form of probabilistic models.

We show how Bayesian time-frequency analysis can be reformulated as a spectral mixture Gaussian process, and utilise modern day inference methods to carry out joint time-frequency analysis and nonnegative matrix factorisation. Our reformulation results in increased modelling flexibility, allowing more sophisticated prior knowledge to be encoded, which improves performance on a missing data synthesis task. We demonstrate the generality of this paradigm by showing how the joint model can additionally be applied to both denoising and source separation tasks without modification.

We propose a hybrid statistical-physical model for audio spectrograms based on observations about the way amplitude envelopes decay over time, as well as a nonlinear model based on deep Gaussian processes. We examine the benefits of these methods, all of which are generative in the sense that novel signals can be sampled from the underlying models, allowing us to consider the extent to which they encode the important perceptual characteristics of sound.

Acknowledgements

The completion of this thesis marks the end of a three and a half year project that has been at times rewarding, inspiring and social, whilst at other times overwhelming and isolating. During the ups and downs of this very independent journey, I have continually found refuge in the friendship and guidance of others, some of whom I will take the time to thank here.

Josh Reiss introduced me to the world of research, welcoming me into his group and supporting every path I chose, even during times when it lead away from his own interests. When I began my PhD studies, I asked Dan Stowell to be my co-supervisor, and I'm extremely grateful that he agreed. His ability to ask difficult questions drove me to be a better researcher, and his ability to remain patient as I delivered the wrong answers was seriously impressive.

There are uncountably many other people to thank from the Centre for Digital Music at Queen Mary. I've enjoyed being part of a research environment that is outward looking, creative, diverse and fun. Thanks to all those that I've worked and socialised with. A special shout out to Delia Fano Yela and Marco Martínez Ramírez, whose friendship has been so important to me in the difficult final year of the PhD.

I would like to thank Aki Vehtari for allowing me to visit his excellent research group at Aalto University, and for introducing me to my collaborators Arno Solin and Michael Riis Andersen, without whom much of the most interesting work in this thesis would not exist. Working with them has been a revelation for me, and I'm excited to see where our collaboration will lead as I start my postdoc in Arno's group.

I was fortunate enough to have this thesis examined by Magnus Rattray and Hamed Haddadi, who were thorough in their questioning, but also generous in their praise.

From deciding to study for my MSc, to quitting a stable job to begin a PhD, to moving to Helsinki to start a postdoc, the unwavering support of my family has made me feel secure even in the face of overwhelming uncertainty. Thanks to Mum, Dad, Haze and Elliot.

Lastly, and most importantly, thank you to Julia. For more than could be written in these pages.

‘The thing about working with time, instead of against it, … is that it is not wasted. Even pain counts.’

Ursula K. Le Guin, The Dispossessed

‘To be, in a word, unborable … It is the key to modern life. If you are immune to boredom, there is literally nothing you cannot accomplish.’

David Foster Wallace, The Pale King

Contents

Chapter 1

Introduction

No two instances of natural sound are identical in terms of their time-domain waveform. Yet the human auditory system is able to perceive and classify sounds, likely based on statistical representations (Turner, 2010; McDermott et al., 2013), and as human listeners we are implicitly aware that the sounds we hear around us are realisations of a physical process. Hence studying natural sound requires consideration of a unique blend of physical, stochastic, and statistical information. This thesis is concerned with techniques for audio analysis that incorporate these qualities.

The task of uncovering the hidden structure in an audio waveform, or estimating the sound production mechanism from its time-frequency representation, is highly ill-posed. For example, there are uncountably many ways in which an audio signal can be decomposed into a sum of time-varying periodic components (Cohen, 1995). Such unidentifiability motivates a Bayesian perspective on audio analysis, which marries the quantification of uncertainty and stochasticity with the use of prior information. We propose prior knowledge based on statistical features inspired by human audition and signal analysis, and physical properties based on our knowledge about how natural sound behaves.

The probabilistic approach is a natural fit for the difficult task of representing signals with complex latent structure. However, despite some promising initial results, these methods are not yet widely used for audio analysis. Their links to traditional signal processing are not fully understood, and they come at a large computational cost given that audio signals necessarily contain a large number of temporal data points. Here we utilise the Gaussian process paradigm to formulate our probabilistic models, and to draw links between seemingly disparate methods in the hope that this will inspire further research surrounding probabilistic treatment of sound.

Gaussian processes (Rasmussen and Williams, 2006) are an extension of the multivariate Gaussian distribution to infinite dimensions, allowing us to specify distributions over functions. Their popularity is growing in the field of machine learning due to their principled treatment of uncertainty, applicable in many regression and classification tasks. It has been shown that Gaussian processes have strong connections to stochastic differential equations and dynamical systems (Hartikainen and Särkkä, 2010), and we exploit this fact to deploy them as a tool for modelling audio signals.

Connections between the fields of machine learning and signal processing are explored. In chapter 3 we show that a state of the art probabilistic time-frequency analysis method is in fact identical to a Gaussian process whose kernel is a sum of quasi-periodic components (a spectral mixture Gaussian process, Wilson and Adams, 2013), allowing us to exploit the benefits of both perspectives in terms of inference methods and modelling flexibility.

In chapter 4 we go on to show how a joint model for time-frequency analysis and nonnegative matrix factorisation can be viewed as a nonstationary version of the spectral mixture Gaussian process, again utilising a signal processing perspective to ease the computational overhead, but showing how state of the art statistical inference methods, namely power expectation propagation (Minka, 2004), are crucial in such a complex setting. These models are very general, and we show how they can be applied to many practical tasks without modification, such as audio inpainting, denoising and source separation.

Drawing further connections between different modelling perspectives in chapter 5, we propose a hybrid statistical-physical model for audio magnitude spectrograms, a latent force model (Alvarez et al., 2009), which allows us to interpret learnt latent functions as physical forces driving a dynamical system to produce sound. Finally in chapter 6 we study how a multi-layer approach to analysis, deep Gaussian processes (Damianou and Lawrence, 2013), can be viewed as a nonlinear generalisation of temporal nonnegative matrix factorisation, and we consider the benefits of such a model in terms of missing data synthesis.

1.1 Publications

Publication I **"Unifying Probabilistic Models for Time-Frequency Analysis"** William J. Wilkinson, Michael Riis Andersen, Joshua D. Reiss, Dan Stowell, and Arno Solin *in Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019

chapter 3 is an extension of Publication I, containing a more detailed derivation of the key result as well as additional discussion of the problem domain and insights provided by the empirical results.

WJW carried out most of the theoretical development, designed all the experiments and wrote the majority of the paper. AS provided key theoretical insights involving the stochastic differential equation form of the quasi-periodic covariance function. MRA contributed to theoretical discussion and writing.

Publication II **"End-to-End Probabilistic Inference for Nonstationary Audio Analysis"** William J. Wilkinson, Michael Riis Andersen, Joshua D. Reiss, Dan Stowell, and Arno Solin *in Int. Conference on Machine Learning (ICML)*, 2019

Publication II forms the basis of chapter 4, but extra discussion and background are provided, particularly around the infinite-horizon approximation. We also improve the presentation of the state space model derivation and the connections to nonstationary spectral mixture models.

Theoretical contribution and paper writing was shared amongst WJW, MRA and AS. WJW took responsibility for practical implementations and designing and running all experiments. MRA assisted in formulating and implementing the expectation propagation algorithm. AS implemented the extended Kalman filter baseline algorithm.

Publication III **"Latent Force Models for Sound: Learning Modal Synthesis Parameters and Excitation Functions from Audio Recordings"** William J. Wilkinson, Joshua D. Reiss and Dan Stowell *in Int. Conference on Digital Audio Effects (DAFx)*, 2017

Publication III forms the basis of section 5.1 and section 5.2, proposing the latent force model paradigm for audio envelopes and drawing connections to modal synthesis.

WJW contributed the majority of work towards this publication, with guidance from JDR and DS.

Publication IV "A Generative Model for Natural Sounds Based on Latent Force Modelling" William J. Wilkinson, Joshua D. Reiss and Dan Stowell *in Int. Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2018

Publication IV forms the basis of section 5.3 and section 5.4, extending and generalising the latent force model approach to a wider class of sounds and evaluating the results via a listening test with human participants.

WJW contributed the majority of work towards this publication, with guidance from JDR and DS.

Chapter 6 "Deep Gaussian Processes as a Nonlinear Model for Audio Spectrograms"

chapter 6 is unpublished work. Concept and theoretical development carried out by WJW, as well as experimental design and most of the implementation, with guidance from DS and JR. MRA contributed to implementation of the new likelihood function and many theoretical discussions. Sachith Pai helped with part of the implementation of monotonic deep Gaussian processes.

Chapter 2

Background

Here we provide an overview of the necessary background required to motivate a probabilistic approach to audio signal processing, and we formally introduce Gaussian processes for time series modelling.

Throughout this document we notate time-domain observations of an audio signal, sampled at time instances $k = 1, \dots, T$, as $\mathbf{y} = \begin{pmatrix} y_1 & \dots & y_T \end{pmatrix}^\top \in \mathbb{R}^T$. Assume that $\mathbf{y}$ comprises D (quasi-)periodic subband components $\mathbf{s}_d = \begin{pmatrix} s_{d,1} & \dots & s_{d,T} \end{pmatrix}^\top \in \mathbb{R}^T$, $d = 1, \dots, D$, which can be summed to produce the signal. Further assume that the $\mathbf{s}_d$ can themselves be modelled as the product of a (quasi-)periodic carrier signal $\mathbf{z}_d = \begin{pmatrix} z_{d,1} & \dots & z_{d,T} \end{pmatrix}^\top \in \mathbb{R}^T$ and a nonnegative amplitude $\mathbf{a}_d = \begin{pmatrix} a_{d,1} & \dots & a_{d,T} \end{pmatrix}^\top \in \mathbb{R}^T$,

$$y_k = \sum_{d=1}^{D} s_{d,k} = \sum_{d=1}^{D} a_{d,k} z_{d,k}. \tag{2.1}$$

The majority of research in this thesis is concerned with identifying or modelling these unobserved (latent) components $\mathbf{a}_d$, $\mathbf{z}_d$ or $\mathbf{s}_d$, when we only have access to observations of the audio signal $\mathbf{y}$.

A typical approach to decomposing the signal is by passing $\mathbf{y}$ through a fixed set of arbitrary filters with coefficients $\boldsymbol{\theta}_{\text{filt}}$,

$$s_{d,k} = \text{filter}_d(\mathbf{y}_{1:k}, \mathbf{s}_{d,1:k-1}, \boldsymbol{\theta}_{\text{filt}}^{(d)}). \tag{2.2}$$

Any such decomposition requires parametric choices to be made in determining $\boldsymbol{\theta}_{\text{filt}}$, such as filter centre frequencies and bandwidths, and uncertainty in prediction of the latent components is not typically considered.

2.1 From Deterministic to Probabilistic: Bayesian Inference

Statistical inference (Gelman et al., 2013) is the practice of drawing conclusions about unobserved quantities, say $\mathbf{s}_d$, from numerical data, say $\mathbf{y}$. In the deterministic paradigm laid out in Eq. (2.1) and Eq. (2.2), the data tell us nothing about our choice of parameters $\boldsymbol{\theta}_{\text{filt}}$. That is to say, our choice of filters cannot be assessed or updated based on the decomposition they provide. Reversing this logic, it is also the case that the filters themselves (or their parameters) provide no information about the characteristics of the signal.

The *Bayesian* approach to statistical inference involves first writing down our model in terms of probability statements. In the general case, we specify a joint probability distribution over data $\mathbf{y}$ and model parameters $\boldsymbol{\theta}$, which can be written as a product of the *prior distribution* $p(\boldsymbol{\theta})$ and the *likelihood function* $p(\mathbf{y} \,|\, \boldsymbol{\theta})$,

$$p(\boldsymbol{\theta}, \mathbf{y}) = p(\boldsymbol{\theta}) p(\mathbf{y} \,|\, \boldsymbol{\theta}). \tag{2.3}$$

The prior distribution characterises our assumptions about the possible forms of the model, whilst the likelihood describes the data observation mechanism. Once these two components are defined, it is possible to draw conclusions about $\boldsymbol{\theta}$ via application of Bayes' rule, resulting in a *posterior distribution*

$$p(\boldsymbol{\theta} \,|\, \mathbf{y}) = \frac{p(\boldsymbol{\theta}, \mathbf{y})}{p(\mathbf{y})} = \frac{p(\boldsymbol{\theta}) p(\mathbf{y} \,|\, \boldsymbol{\theta})}{p(\mathbf{y})}, \tag{2.4}$$

where the normalisation term, the *marginal likelihood*, for continuous $\boldsymbol{\theta}$ is

$$p(\mathbf{y}) = \int p(\boldsymbol{\theta}) p(\mathbf{y} \,|\, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}. \tag{2.5}$$

Considering our example in Eq. (2.2), the prior $p(\boldsymbol{\theta}_{\text{filt}})$ should specify how probable the various choices of filter coefficients $\boldsymbol{\theta}_{\text{filt}}$ are, based on our knowledge about auditory filters.

The most common form of the likelihood model used in this thesis is a Gaussian distribution, which states that the latent components are observed through Gaussian noise. Continuing with our model for an audio signal, we can express this statement mathematically as

$$p(\mathbf{y} \,|\, \boldsymbol{\theta}) = \mathrm{N}\left(\sum_{d=1}^{D} \mathbf{s}_d, \sigma_y^2 \mathbf{I}_T\right) \tag{2.6a}$$

$$\overset{\text{i.i.d.}}{=} \prod_{k=1}^{T} \mathrm{N}\left(\sum_{d=1}^{D} s_{d,k}, \sigma_y^2\right), \tag{2.6b}$$

for T-dimensional identity matrix $\mathbf{I}_T$. The likelihood factorises over the time steps to give Eq. (2.6b) since the Gaussian noise is independently and identically distributed (i.i.d.). Equivalently we can add a noise term to Eq. (2.1),

$$y_k = \sum_{d=1}^{D} s_{d,k} + \sigma_y \varepsilon_k, \tag{2.7}$$

where $\varepsilon_k \sim \mathrm{N}(0, 1)$ is unit-variance Gaussian distributed. The entire model is now parameterised by $\boldsymbol{\theta} = \{\sigma_y,\ \boldsymbol{\theta}_{\text{filt}}\}$.

A crucial advantage of the Bayesian paradigm as laid out above is its capacity for model selection and prediction. The marginal likelihood can be interpreted as the evidence provided for $\boldsymbol{\theta}$ by the signal $\mathbf{y}$. In other words, it measures the agreement between the model and the data. Hence optimising Eq. (2.5) via gradient-based methods provides a way to tune the parameters to fit the data.

Given a set of tuned parameters, we can make predictions about the value of an unknown data point y_* using the *posterior predictive distribution*

$$p(y_* \,|\, \mathbf{y}) = \int p(y_* \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta} \,|\, \mathbf{y}) \, \mathrm{d}\boldsymbol{\theta}. \tag{2.8}$$

As we will see, in terms of temporal data where $y_* = y_{T+1}$, prediction amounts to synthesis of the next data point in the sequence. For our proposed model of an audio signal, tuning $\boldsymbol{\theta}_{\text{filt}}$ would mean that the model now represents the most probable set of filters characterising the signal (given our choice of $p(\boldsymbol{\theta})$ and $p(\mathbf{y} \,|\, \boldsymbol{\theta})$).

In the case of auditory filters however, as shown in section 2.2 and chapter 3, it can be more beneficial to treat the components $\mathbf{s}_d$ as latent variables and specify a probabilistic model for them directly, i.e. with priors $p(\mathbf{s}_d)$, potentially parameterised by a further set of hyperparameters $\boldsymbol{\theta}$. This results in the joint model,

$$p(\mathbf{s}_d, \mathbf{y} \,|\, \boldsymbol{\theta}) = p(\mathbf{s}_d \,|\, \boldsymbol{\theta}) p(\mathbf{y} \,|\, \mathbf{s}_d, \boldsymbol{\theta}). \tag{2.9}$$

The posterior in this case, $p(\mathbf{s}_d \,|\, \mathbf{y}, \boldsymbol{\theta})$, is now a probability statement about the latents $\mathbf{s}_d$. We now write the marginal likelihood as $p(\mathbf{y} \,|\, \boldsymbol{\theta}) = \int p(\mathbf{s}_d \,|\, \boldsymbol{\theta}) p(\mathbf{y} \,|\, \mathbf{s}_d, \boldsymbol{\theta}) \, \mathrm{d}\mathbf{s}_d$, which can be maximised to find the optimal $\boldsymbol{\theta}$.

2.2 Statistical Models for Sound and for the Perception of Sound

The idea that perceptual representations of sound should be statistical is supported by two recent advances in analysis of *sound textures*. This class of sounds encapsulates composite (multi-source) signals who's characteristics do not vary over time, such as running water, or a crackling fireplace.

Firstly, McDermott et al. (2009) showed that perception of these composite sounds is likely based on a set of summary statistics of subband modulation and energy distribution. An optimisation procedure was designed in which the statistics of a synthetic signal, initially instantiated as noise, were iteratively updated via gradient descent until they match those of a target signal. Such a process results in generation of novel stimuli that are recognisable as the same sound type as the target, and their realism is evidence supporting the claim that these statistics are in fact the ones utilised during human audition.

Consider the case where the subbands, $\mathbf{s}_d$, in Eq. (2.1) are generated by a deterministic set of cochlear filters: $D \approx 30$ band pass filters that logarithmically span an audible frequency range ($52 - 8844$ Hz) with equal rectangular bandwidth (ERB) spacing in a loose analogy with the processing performed in the human cochlea (Glasberg and Moore, 1990). Much of the perceptual information present in a sound, whether it be environmental sound or speech, is contained within the amplitude envelopes of these filter outputs (Shannon et al., 1995). For this reason many of the statistics, which we list below, are based on the spectrographic information $\mathbf{a}_d$ (McDermott and Simoncelli, 2011).

Cochlear envelope marginal statistics The first four normalised moments (mean, variance, skew and kurtosis) of the amplitude envelopes $\mathbf{a}_d$ multiplied by a windowing function.

Cross-band envelope correlation The correlation between $\mathbf{a}_d$ and $\mathbf{a}_j$ for $d - j \in [1, 2, 3, 5, 8, 11, 16, 21]$, eight neighbouring envelopes which are sufficient to reproduce the full covariance structure.

Modulation power The envelopes $\mathbf{a}_d$ are passed through a second set of 20 filters, spanning $0.5 - 200$ Hz, intended to measure amplitude modulation rates. The variance of the outputs of these filters represents the modulation power.

Modulation correlation The correlation between the modulation filter outputs, both within acoustic frequency channels d (across modulation bands) and across acoustic frequency channels.

As we can see from the list above, the perceptual statistics are concerned with modulation and co-modulation of amplitude envelopes. The success of this statistical approach has been followed by more work on synthesis of audio textures, as well as musical notes, most commonly based on convolutional neural networks applied to either the spectrogram (Antognini et al., 2018) or directly on the audio waveform (van den Oord et al., 2016; Engel et al., 2017). Whilst these models don't explicitly target perceptual characteristics, Kell et al. (2018) showed that a neural network architecture inspired by the human auditory cortex performed similarly to human listeners on speech and music recognition tasks, and perhaps even more significantly, made similar errors to human listeners. Likewise, rather than explicitly calculating perceptual statistics, Turner (2010) proposed to write down a probabilistic model which implicitly encodes many of the same characteristics. In this case, samples drawn from the model were also shown to be recognisable as the same sound type as the recording from which their parameters were learnt.

The likelihood model proposed by Turner (2010) is similar to Eq. (2.7) where the subbands are decomposed as a product of slowly-varying positive amplitudes $\mathbf{a}_d$ and fast-varying carriers $\mathbf{z}_d$ plus Gaussian noise,

$$y_k = \sum_{d=1}^{D} a_{d,k} z_{d,k} + \sigma_y \varepsilon_k, \tag{2.10}$$

but in this case the amplitudes are modelled as a linear mixture of $N < D$ processes $\mathbf{g}_n \in \mathbb{R}^T$, $n = 1, \ldots, N$, projected through a nonlinear mapping to enforce positivity (the softplus function $\phi(g) = \log(1 + \mathrm{e}^{g})$),

$$a_{d,k} = \sum_{n=1}^{N} W_{d,n}\phi(g_{n,k}). \tag{2.11}$$

$W_{d,n}$ is the mixture weight specifying how the n^{th} process affects the d^{th} envelope. This low-dimensional mapping ensures that cross-band amplitude correlation is captured, one of the important perceptual statistics listed by McDermott and Simoncelli (2011). The prior over $\mathbf{g}_n$ is Gaussian,

$$p(\mathbf{g}_n) = \mathrm{N}\left(\mathbf{0}, \mathbf{K}^{(n)}\right), \tag{2.12}$$

where $\mathbf{K}^{(n)} \in \mathbb{R}^{T\times T}$ is the covariance matrix constructed via the *exponentiated quadratic* covariance function, $\mathbf{K}^{(n)}_{i,j} = C_n(t_i, t_j)$ (see Rasmussen and Williams (2006) and section 2.5), which encodes the prior assumption that the amplitudes are smooth and vary slowly over time.

The prior over the latent carriers $\mathbf{z}_d$ is a second-order Gaussian autoregressive process,

$$\begin{aligned} p(z_{d,k}) &= \mathrm{N}\left(\lambda_{d,1}z_{d,k-1} + \lambda_{d,2}z_{d,k-2},\ \sigma_d^2\right) && \text{(2.13a)} \\ &= \lambda_{d,1}z_{d,k-1} + \lambda_{d,2}z_{d,k-2} + \sigma_d\varepsilon_k. && \text{(2.13b)} \end{aligned}$$

The model is parameterised by $\boldsymbol{\theta} = \{\{\lambda_{d,1}, \lambda_{d,2}, \sigma_d\}_{d=1}^{D}, \{\sigma_n, \ell_n\}_{n=1}^{N}, \sigma_y\}$, where σ_n, ℓ_n are the hyperparameters of the kernel C_n. Notice that this represents significantly fewer parameters than in the summary statistics model above whilst still implicitly encoding similar features, and they have been used to compare the perceptual response of normal-hearing and cochlear-implant human listeners to changes in behaviour of stochastic envelopes $\mathbf{a}_d$ (Gomersall et al., 2016).

A slight modification to this model in which the prior over $\mathbf{z}_d$ contains a periodic component results in a joint Gaussian time-frequency analysis and nonnegative matrix factorisation model (GTF-NMF) (Turner and Sahani, 2014), which we provide a new interpretation of in chapter 4.

Nonnegative matrix factorisation NMF (Lee and Seung, 1999) decomposes a high-dimensional matrix $\mathbf{A} = \begin{pmatrix}\mathbf{a}_1 & \dots & \mathbf{a}_D\end{pmatrix}^\top \in \mathbb{R}^{D\times T}$, such as the magnitude spectrogram of an audio signal, into a product of two lower-rank nonnegative matrices: a temporal dictionary $\mathbf{G}$, and a spectral dictionary $\mathbf{W}$,

$$\mathbf{A} \simeq \mathbf{W}\mathbf{G}. \tag{2.14}$$

Typically $\mathbf{W} \in \mathbb{R}^{D \times N}$ and $\mathbf{G} \in \mathbb{R}^{N \times T}$ are learnt by minimising the divergence between the left and right hand sides of Eq. (2.14), whereas in the above model, a Gaussian prior, $p(\mathbf{g}_n)$, is placed over the rows of $\mathbf{G} = \begin{pmatrix} \mathbf{g}_1 & \dots & \mathbf{g}_N \end{pmatrix}^\top$ and the elements of $\mathbf{W}$ are treated as free parameters of the probabilistic model. If we were to disregard the subband carrier signals, such a prior over the temporal components would result in a probabilistic extension to NMF called temporal NMF (tNMF, Bertin et al., 2010; Turner and Sahani, 2014).

Inference in the GTF-NMF model is not straightforward. The posterior can no longer be calculated in closed form since the likelihood model now contains a nonlinear mixture of the latents $\mathbf{g}_n$ and $\mathbf{z}_d$, which makes the integral for the marginal likelihood, Eq. (2.5), intractable. For this reason, Turner and Sahani (2014) use a two-stage inference method that separates out the amplitude model from the carrier model, iteratively updating one whilst fixing the other. Calculation of the posterior, tuning of the parameters, and prediction in the GTF-NMF is carried out via Kalman filtering (Kalman, 1960, see section 2.3). We provide a new way to perform joint inference on the full model in chapter 4.

Bayesian time-frequency analysis The application of statistical inference in time-frequency analysis has been addressed in a number of ways, most notably Bayesian Spectrum Estimation (BSE, Qi et al., 2002) and the probabilistic phase vocoder (PPV, Cemgil and Godsill, 2005). For an overview see Turner and Sahani (2014), where it is also shown that the PPV and BSE are equivalent up to a shift in frequency. The PPV version of this adaptive time-frequency analysis approach is

$$s_{d,k} = \psi_d \mathrm{e}^{i\omega_d} s_{d,k-1} + \rho_d \zeta_{d,k}, \tag{2.15a}$$

$$y_k = \sum_{d=1}^{D} \mathrm{Re}[s_{d,k}] + \sigma_{y_k} \varepsilon_k, \tag{2.15b}$$

where $s_{d,k} \in \mathbb{C}$ is now a complex phasor representing the latent subband signal in frequency channel d. $\zeta_{d,k} \sim \mathrm{CN}(0,1)$ is i.i.d. complex Gaussian noise and now the likelihood sums the real parts of $\mathbf{s}_d$ plus noise to produce the signal $\mathbf{y}$. Parameters ψ_d and ρ_d represent the process and noise variances respectively, whilst ω_d is the instantaneous angular frequency.

Whilst it was known that the above PPV, BSE and GTF-NMF approaches fall under the paradigm of *Gaussian processes* (see section 2.4), it was not clear until now how these models relate to the usual set of Gaussian process modelling techniques developed in the machine learning community. This is because they were partly conceived from the perspective of discrete-time autoregressive filters, rather than through design of covariance functions encoding our prior knowledge. We address this issue in chapter 3. We now introduce the Gaussian process framework, arriving at it from the perspective of signal processing, dynamical systems and stochastic differential equations.

2.3 Covariance Through Time in Stochastic Differential Equations

Any audio recording of natural sound comes about as a result of a physical process. As such, one view of our signal $\mathbf{y}$ is as a realisation of a dynamical system: a process which evolves over time based on some known physical model. We can choose to approximate such a system with a linear time-invariant (LTI) stochastic differential equation (SDE), which can be written in a general continuous state space form as (Särkkä and Solin, 2019):

$$\frac{\mathrm{d}\tilde{\mathbf{f}}(t)}{\mathrm{d}t} = \mathbf{F}\tilde{\mathbf{f}}(t) + \mathbf{L}\mathbf{w}(t), \tag{2.16a}$$

$$y_k = \mathbf{H}\tilde{\mathbf{f}}(t_k) + \sigma_y \varepsilon_k, \tag{2.16b}$$

for state vector $\tilde{\mathbf{f}}(t) = \begin{pmatrix} f(t) & \mathrm{d}/\mathrm{d}t f(t) & \dots & \mathrm{d}^{M-1}/\mathrm{d}t^{M-1} f(t) \end{pmatrix}^\top \in \mathbb{R}^M$, driven by white noise $\mathbf{w}(t) \in \mathbb{R}^S$ with spectral density $\mathbf{Q}_c \in \mathbb{R}^{S\times S}$. The prior, Eq. (2.16a), is characterised by feedback matrix $\mathbf{F} \in \mathbb{R}^{M\times M}$ and noise effect matrix $\mathbf{L} \in \mathbb{R}^{M\times S}$, and without loss of generality is centred about zero. We have assumed that there exists a single function of interest, f, which is observed through Gaussian noise in the measurement model, Eq. (2.16b), via measurement matrix $\mathbf{H} \in \mathbb{R}^{1\times M}$, and under this assumption typically $\mathbf{H} = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}$ such that $\mathbf{H}\tilde{\mathbf{f}}(t_k) = f(t_k)$.

The white noise term represents stochasticity in the system, and is the means by which we characterise uncertainty. Inferring the state of the continuous prior, Eq. (2.16a), boils down to calculating the mean $\mathbf{m}(t) = \mathbb{E}[\tilde{\mathbf{f}}(t)]$ and covariance $\mathbf{P}(t) = \mathrm{Var}[\tilde{\mathbf{f}}(t)] = \mathbb{E}[(\tilde{\mathbf{f}}(t) - \mathbf{m}(t))(\tilde{\mathbf{f}}(t) - \mathbf{m}(t))^\top]$ at each time step. To do so, we calculate their time derivative, which in the LTI case is (Särkkä and Solin, 2019),

$$\frac{\mathrm{d}\mathbf{m}(t)}{\mathrm{d}t} = \mathbf{F}\mathbf{m}(t), \tag{2.17a}$$

$$\frac{\mathrm{d}\mathbf{P}(t)}{\mathrm{d}t} = \mathbf{F}\mathbf{P}(t) + \mathbf{P}(t)\mathbf{F}^\top + \mathbf{L}\mathbf{Q}_c\mathbf{L}^\top. \tag{2.17b}$$

It is now beneficial to consider the steady state solution to the SDE, which occurs as $t \to \infty$, or rather as initial time step $t_1 \to -\infty$ such that the system is in a steady state at current time step t. In our linear model, the derivatives of the steady state mean $\mathbf{m}_\infty$ and covariance $\mathbf{P}_\infty$ should be zero. Eq. (2.17a) implies that $\mathbf{m}_\infty = \mathbf{0}$, whilst setting Eq. (2.17b) to zero requires us to solve the Lyapunov equation:

$$\mathbf{F}\mathbf{P}_\infty + \mathbf{P}_\infty\mathbf{F}^\top + \mathbf{L}\mathbf{Q}_c\mathbf{L}^\top = \mathbf{0}. \tag{2.18}$$

In the interest of capturing our desired perceptual statistics from section 2.2, including modulation rates, smoothness and correlations, we seek to calculate the covariance of the system in Eq. (2.16) across time steps, rather than at a single time instance t. Stationary covariance functions act on the time difference between two steps, $\tau = t - t'$, and in the LTI case are

$$C(\tau) = \begin{cases} \mathbf{H}\mathbf{P}_\infty \exp(\tau\mathbf{F})^\top\mathbf{H}^\top, & \text{if } \tau > 0 \\ \mathbf{H}\exp(-\tau\mathbf{F})\mathbf{P}_\infty\mathbf{H}^\top, & \text{if } \tau \leq 0 \end{cases} \tag{2.19}$$

where $\exp(\tau\mathbf{F})$ is the matrix exponential of the feedback matrix.

Filtering and smoothing solutions to SDEs Eq. (2.16) represents a continuous-discrete system: a continuous process from which we observe noisy samples at discrete time steps k. In the Gaussian likelihood case, its filtering and smoothing problems have closed form solutions. The filtering problem, to determine $p(\tilde{\mathbf{f}}(t_k) \,|\, \mathbf{y}_{1:k}) = \mathrm{N}(\mathbf{m}_k^{(f)}, \mathbf{P}_k^{(f)})$, is solved via Kalman filtering (Kalman, 1960). First we use the matrix exponential to discretise the model, giving

$$\tilde{\mathbf{f}}(t_{k+1}) = \mathbf{A}\tilde{\mathbf{f}}(t_k) + \mathbf{q}_k, \tag{2.20a}$$

$$y_k = \mathbf{H}\tilde{\mathbf{f}}(t_k) + \sigma_y \varepsilon_k, \tag{2.20b}$$

where $\mathbf{q}_k \sim \mathrm{N}(0, \mathbf{Q})$, $\mathbf{Q} = \mathbf{P}_\infty - \mathbf{A}\mathbf{P}_\infty\mathbf{A}^\top$ and $\mathbf{A} = \exp(\mathbf{F}\Delta t)$ for time step size $\Delta t = t_k - t_{k-1}$, which we assume to be constant since audio signals are typically sampled at a constant rate.

Now the Kalman filter is applied as follows, where the initial state is distributed $p(\tilde{\mathbf{f}}(t_0)) = \mathrm{N}(\mathbf{m}_0, \mathbf{P}_0)$ and typically $\mathbf{m}_0 = \mathbf{m}_\infty$, $\mathbf{P}_0 = \mathbf{P}_\infty$:
prediction step:

$$\begin{aligned} \mathbf{m}_k^- &= \mathbf{A}\mathbf{m}_{k-1}^{(f)}, \\ \mathbf{P}_k^- &= \mathbf{A}\mathbf{P}_k^{(f)}\mathbf{A}^\top + \mathbf{Q}. \end{aligned} \tag{2.21}$$

update step:

$$\begin{aligned} \mathbf{v}_k &= y_k - \mathbf{H}\mathbf{m}_k^-, \\ \mathbf{S}_k &= \mathbf{H}\mathbf{P}_k^-\mathbf{H}^\top + \sigma_y^2, \\ \mathbf{K}_k &= \mathbf{P}_k^-\mathbf{H}^\top\mathbf{S}_k^{-1}, \\ \mathbf{m}_k^{(f)} &= \mathbf{m}_k^- + \mathbf{K}_k\mathbf{v}_k, \\ \mathbf{P}_k^{(f)} &= \mathbf{P}_k^- - \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^\top. \end{aligned} \tag{2.22}$$

Similarly, the smoothing problem, $p(\tilde{\mathbf{f}}(t_k) \,|\, \mathbf{y}_{1:T}) = \mathrm{N}(\mathbf{m}_k^{(s)}, \mathbf{P}_k^{(s)})$, is solved via the Rauch–Tung–Striebel (RTS) smoother (Särkkä, 2013):
RTS smoother:

$$\begin{aligned} \mathbf{m}_{k-1}^- &= \mathbf{A}\mathbf{m}_k^{(s)}, \\ \mathbf{P}_{k-1}^- &= \mathbf{A}\mathbf{P}_k^{(s)}\mathbf{A}^\top + \mathbf{Q}, \\ \mathbf{G}_k &= \mathbf{P}_k^{(s)}\mathbf{A}^\top(\mathbf{P}_{k+1}^-)^{-1}, \\ \mathbf{m}_k^{(s)} &= \mathbf{m}_k^{(f)} + \mathbf{G}_k(\mathbf{m}_{k+1}^{(s)} - \mathbf{m}_{k+1}^-), \\ \mathbf{P}_k^{(s)} &= \mathbf{P}_k^{(f)} + \mathbf{G}_k(\mathbf{P}_{k+1}^{(s)} - \mathbf{P}_{k+1}^-)\mathbf{G}_k^\top. \end{aligned} \tag{2.23}$$

2.4 Gaussian Processes for Time Series Modelling

Given these tools for calculating covariance in dynamical systems, we now consider as an example the physical system represented by the Ornstein-Uhlenbeck process (Uhlenbeck and Ornstein, 1930):

$$\frac{\mathrm{d}f(t)}{\mathrm{d}t} = -\lambda f(t) + w(t), \tag{2.24}$$

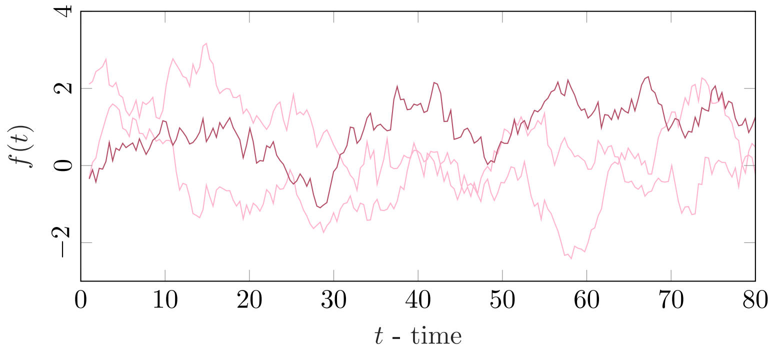


Figure 2.1: Samples from the Ornstein-Uhlenbeck process prior with $q = 0.2$, $\lambda = 0.1$.

which is a first-order LTI SDE driven by one-dimensional Gaussian noise $w(t)$ (the time-derivative of Brownian motion). Solving the Lyapunov equation in Eq. (2.18) for $\mathbf{F} = -\lambda$, $\mathbf{L} = 1$ and $\mathbf{Q}_c = q$ we get

$$-2\lambda \mathbf{P}_\infty + q = 0, \tag{2.25}$$

so that $\mathbf{P}_\infty = \frac{q}{2\lambda}$ and

$$C(t, t') = \frac{q}{2\lambda} \exp(-\lambda |t - t'|). \tag{2.26}$$

Our information regarding the system in Eq. (2.24) can be summarised as follows: the function values $f(t)$ and $f(t')$ are jointly Gaussian, since the Gaussianity of $w(t)$ is preserved under linear operations, with steady state mean of zero, $\mathbf{m}_\infty(t) = 0$, and whose stationary covariance between time steps decays exponentially with the time gap according to the function $C(t, t')$.

This information completely characterises a *Gaussian process* (GP, Rasmussen and Williams, 2006), and shows how the Ornstein-Uhlenbeck process is a particular form of the GP models that are commonly used for regression and classification tasks in the machine learning community. A similar equivalence can also be shown for many more SDE models (Solin, 2016).

Canonical form of a Gaussian process GPs are a generalisation of the Gaussian distribution to infinite dimensions, and hence they can be interpreted as a way to specify a distribution over functions (infinite-length vectors). A full GP model in its standard formulation with one-dimensional input and output is written as follows:

$$\begin{aligned} f(t) &\sim \mathrm{GP}(\mu(t), C(t,t')) && \text{(2.27a)} \\ \mathbf{y} \,|\, \mathbf{f} &\sim p(\mathbf{y} \,|\, \mathbf{f}) && \text{(2.27b)} \\ &\overset{\text{iid}}{=} \prod_{k=1}^{T} p(y_k \,|\, f(t_k)) && \text{(2.27c)} \end{aligned}$$

Eq. (2.27a) is the GP prior, which states that any finite collection of function values has a joint multivariate Gaussian distribution $\mathbf{f} = \begin{pmatrix} f(t_0) & \dots & f(t_T) \end{pmatrix}^\top \sim \mathrm{N}(\mathbf{m}, \mathbf{K})$ where $\mathbf{m}_i = \mu(t_i)$ and $\mathbf{K}_{i,j} = C(t_i, t_j)$:

$$\begin{pmatrix} f(t_1) \\ \vdots \\ f(t_T) \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} \mu(t_1) \\ \vdots \\ \mu(t_T) \end{pmatrix}, \begin{pmatrix} C(t_1,t_1) & \dots & C(t_1,t_T) \\ \vdots & \ddots & \vdots \\ C(t_T,t_1) & \dots & C(t_T,t_T) \end{pmatrix} \right). \tag{2.28}$$

Our choice of mean function $\mu(t)$ and covariance function $C(t,t')$ should be determined by our prior assumptions about the latent process $f(t)$. That is, we encode our assumptions about f by defining how its function values co-vary when evaluated at different points in time. Eq. (2.27b) is the likelihood model, which factorises across data points in the i.i.d. case, and the data $\boldsymbol{\mathcal{D}} = \{(t_k, y_k)\}_{k=1}^T$ consist of input–output pairs.

Inference in Gaussian process models Following the Bayesian approach, we aim to infer a posterior distribution that tells us about the form of $\mathbf{f} \,|\, \mathbf{y}$. Assume again that the likelihood is Gaussian, $p(\mathbf{y} \,|\, \mathbf{f}) = \prod_{k=1}^T \mathrm{N}(f(t_k), \sigma_y^2)$. Now for $\mathbf{t}_* = \begin{pmatrix} t_{*_1} & t_{*_2} & \dots \end{pmatrix}^\top$, a new set of input locations which could correspond to the next set of points in the sequence or a missing segment in the signal, the joint distribution is also Gaussian by the definition of a GP:

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}_* \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} \mathbf{m} \\ \mathbf{m}_* \end{pmatrix}, \begin{pmatrix} \mathbf{K} + \sigma_y^2 \mathbf{I}_T & \mathbf{K}_* \\ \mathbf{K}_*^\top & \mathbf{K}_{**} \end{pmatrix} \right) \tag{2.29}$$

where $\mathbf{f}_*$ and $\mathbf{m}_*$ represent f and μ evaluated at $\mathbf{t}_*$ respectively. $\mathbf{K}_{**i,j} = C(t_{*_i}, t_{*_j})$ is the covariance matrix evaluated on $\mathbf{t}_*$, whilst $\mathbf{K}_{*i,j} = C(t_i, t_{*_j})$ is the cross-covariance at $\mathbf{t} = \begin{pmatrix} t_1 & \dots & t_T \end{pmatrix}^\top$ and $\mathbf{t}_*$. The observation noise σ_y^2 is included to obtain the covariance for the observed signal $\mathbf{y}$.

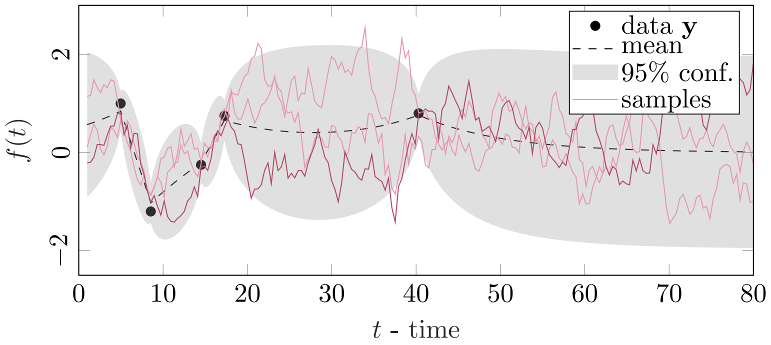


Figure 2.2: Samples from the posterior distribution of an Ornstein-Uhlenbeck process with $q = 0.2$, $\lambda = 0.1$ where five data points have been observed with Gaussian measurement noise $\sigma_y^2 = 0.05$.

Using the properties of the multivariate Gaussian distribution, we can now calculate the posterior over f evaluated at all time steps $\mathbf{t}_{\text{joint}} = (\mathbf{t}, \mathbf{t}_*)^\top$, with $\mathbf{m}_{\text{joint}} = (\mathbf{m}, \mathbf{m}_*)^\top$, as

$$\begin{aligned}\mathbf{f}_{\text{joint}} \,|\, \mathbf{y} \sim \mathrm{N}\Big(&\mathbf{m}_{\text{joint}} + \mathbf{K}_* \left(\mathbf{K} + \sigma_y^2 \mathbf{I}_T\right)^{-1} (\mathbf{y} - \mathbf{m}), \\ &\mathbf{K}_{**} - \mathbf{K}_* \left(\mathbf{K} + \sigma_y^2 \mathbf{I}_T\right)^{-1} \mathbf{K}_*^\top\Big).\end{aligned} \tag{2.30}$$

Non-Gaussian likelihood models If the likelihood model is non-Gaussian, then the posterior no longer has a closed form Gaussian solution. In such a case, many methods for approximation of the posterior have been developed. These are generally based on variations of expectation propagation (EP) (Minka, 2001), the Laplace approximation (LA) (Williams and Barber, 1998), or variational bounds (VB) (Gibbs and MacKay, 2000). Bui et al. (2017) demonstrates how these methods can be considered members of a single unified paradigm for approximating non-Gaussian distributions with Gaussians, and give an excellent overview of these methods.

In the SDE representation of a GP model, nonlinear filtering and smoothing methods developed in the signal processing field can also be used to perform inference and parameter estimation in more complex models whose likelihood goes beyond the linear Gaussian form (Nickisch et al., 2018). The most common signal processing methods are the extended Kalman filter (EKF, Jazwinski, 1970; Bar-Shalom et al., 2001) and unscented Kalman filter (UKF, Wan and Van Der Merwe, 2000). The EKF linearises the model about the mean at each time step before applying the filtering equations, while the UKF uses sigma-point methods (see McNamee and Stenger, 1967; Kokkala et al., 2016) to approximate the intractable integrals required for calculating the filter and smoothing distributions in the nonlinear case. However, Nickisch et al. (2018) showed that the approximate inference methods listed above (single-sweep EP, LA, VB) can also be implemented in the filter and smoother setting for GP models, and we demonstrate fully iterated EP in the GTF-NMF setting in chapter 4.

System identification via hyperparameter learning As stated in section 2.2, we can tune the hyperparameters $\boldsymbol{\theta}$ by maximising the (log) marginal likelihood, $\log p(\mathbf{y} \,|\, \boldsymbol{\theta})$. In the canonical form of GPs, assuming a Gaussian likelihood, this can be calculated analytically as (Rasmussen and Williams, 2006)

$$\log p(\mathbf{y} \,|\, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^\top(\mathbf{K}+\sigma_y^2\mathbf{I}_T)^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}+\sigma_y^2\mathbf{I}_T| - \frac{T}{2}\log 2\pi. \tag{2.31}$$

The approximate inference methods required for non-Gaussian likelihoods also typically provide ways to calculate or approximate the marginal likelihood.

In the SDE form of GPs, the Kalman filter equations provide us with the means by which to calculate the energy function $\upsilon(\boldsymbol{\theta})$ (Särkkä, 2013), which is equivalent to the negative log marginal likelihood,

$$\upsilon(\boldsymbol{\theta}) = \sum_{k=1}^{T}\frac{1}{2}\log|2\pi\mathbf{S}_k| + \frac{1}{2}\sum_{k=1}^{T}(y_k - \mathbf{H}\mathbf{m}_k^-)^\top\mathbf{S}_k^{-1}(y_k - \mathbf{H}\mathbf{m}_k^-). \tag{2.32}$$

In the non-Gaussian case we still have access to the required model components during filtering ($\mathbf{S}_k$, $\mathbf{H}$, $\mathbf{m}_k^-$), and so hyperparameter optimisation proceeds in the same manner (Mbalawata et al., 2013; Nickisch et al., 2018).

Computational issues with GP inference A major issue with the canonical GP approach for time series data is that the posterior requires calculation of a matrix inverse which scales cubicly with the number of data points, $\mathcal{O}(T^3)$. Perhaps even more fundamentally, simply calculating and sampling from a covariance matrix at many thousands of input locations is impractical both in terms of compute and memory.

For example, a 5 second audio recording sampled conservatively at 16kHz contains 80000 data points, and hence naive calculation of all the elements of $\mathbf{K}$ requires $80000^2 = 6.4$ billion calculations.

Again, many methods for alleviating these computational issues have been considered. Perhaps the most popular is the inducing point method (Quiñonero-Candela and Rasmussen, 2005; Snelson and Ghahramani, 2006), which leverages sparsity in the covariance to choose a set of $\bar{T} < T$ *pseudo-points* producing function values $\mathbf{f}_{\bar{T}}$ such that the joint model can be augmented: $p(\mathbf{y}, \mathbf{f}, \mathbf{f}_{\bar{T}}) = p(\mathbf{y} \,|\, \mathbf{f}) p(\mathbf{f} \,|\, \mathbf{f}_{\bar{T}}) p(\mathbf{f}_{\bar{T}})$, giving a new form of the posterior predictive distribution:

$$p(y_* \,|\, \mathbf{y}) = \int \int p(y_* \,|\, \mathbf{f}_{\bar{T}}, \mathbf{f}) p(\mathbf{f} \,|\, \mathbf{f}_{\bar{T}}, \mathbf{y}) p(\mathbf{f}_{\bar{T}} \,|\, \mathbf{y}) \, \mathrm{d}\mathbf{f} \, \mathrm{d}\mathbf{f}_{\bar{T}} \tag{2.33a}$$

$$= \int p(y_* \,|\, \mathbf{f}_{\bar{T}}) p(\mathbf{f}_{\bar{T}} \,|\, \mathbf{y}) \, \mathrm{d}\mathbf{f}_{\bar{T}}, \tag{2.33b}$$

where the second line is arrived at via the assumption that $\mathbf{f}_{\bar{T}}$ is a sufficient statistic for $\mathbf{f}$, forcing an independence between y_* and $\mathbf{f}$. It can be shown that, via a further approximation that involves defining a *lower bound* on the marginal likelihood, the posterior of the augmented model can be calculated with computational scaling $\mathcal{O}(\bar{T}^2 T)$, and that the inducing points can be treated as additional hyperparameters of the model (Titsias, 2009).

Whilst the inducing point method has proven useful in many applications, it is deficient for temporal data. In the case of missing data prediction in signal processing, the inducing points will not lie sufficiently close to the required input locations for anything but the shortest of gaps. Synthesis of future time steps will also necessarily go beyond the region covered by $\mathbf{f}_{\bar{T}}$. Both situations require us to define new inducing points by hand. Furthermore, as the signal grows in time, $\bar{T}$ must also grow to maintain the required level of sparsity, and as such the complexity still scales cubicly with time.

One approach to alleviating these issues further is via framing (Liutkus et al., 2011; Alvarado et al., 2019), which performs inference on short time segments before ultimately recombining them via an overlap-add procedure, however this still requires new inducing points to be defined whenever new data regions are considered. Other methods for speeding up inference, including interpolation approaches (Wilson and Nickisch, 2015), stochastic methods (Hensman et al., 2013; Krauth et al., 2017), basis function approximations (Lázaro-Gredilla et al., 2010; Hensman et al., 2018; Solin and Särkkä, 2014a) and band-structured or Toeplitz methods (Saatçi, 2012), also scale poorly for long and unbounded time series with potential for missing segments.

Fortunately, the dual formulation of the Gaussian process model as a dynamical system, Eq. (2.16), alleviates these issues since the smoothing solution $p(\tilde{\mathbf{f}}(t_k) \,|\, \mathbf{y}_{1:T})$ to the state space model is exactly equivalent to the posterior predictive distribution in Eq. (2.30) (Hartikainen and Särkkä, 2010). This approach scales linearly in the number of time steps and cubicly in the state dimensionality, which is independent of time, $\mathcal{O}(M^3T)$. Additionally, it only requires calculation and storage of T covariance matrices $\mathbf{P}(t_k) \in \mathbb{R}^{M \times M}$ ($T \times M^2$ calculations), rather than the full covariance $\mathbf{K}$. The majority of the commonly used covariance functions for GPs are compatible with the LTI SDE form, either exactly or approximately. We discuss some of these below, and refer the reader to Solin (2016) for a more detailed review.

Despite this improvement in scalability, practical issues still remain when processing audio signals with the Kalman filter methods, since the state dimensionality M is likely to be large (often in the hundreds) and because the memory requirements, $\mathcal{O}(M^2T)$, can still become infeasible. The infinite-horizon GP (IHGP) framework proposed by Solin et al. (2018) addresses both these issues by calculating a posterior steady state approximation to each GP such that propagation of the covariance terms in the filter and smoother can be simplified, leading to computational scaling of $\mathcal{O}(M^2T)$ and memory scaling $\mathcal{O}(TM)$. This approximation is accurate in the Gaussian likelihood case, but as we show in chapter 4, in the GTF-NMF model the computational benefits come at the cost of reduced performance on tasks such as audio inpainting and denoising.

Even taking into account all of the above, there still exists a computational barrier preventing wide-scale use of these methods for long temporal data. The IHGP method involves an overhead cost of calculating the stationary solutions, and the nested for-loops involved in learning the hyperparameters in the Kalman filter setting are inefficient, despite scaling linearly in time. Potential solutions for future work not considered in this thesis are online methods (to deal with remaining memory issues, e.g. Csató and Opper (2002); Nguyen-Tuong et al. (2009)) and the use of banded matrix operators to make GP models amenable to automatic differentiation (Durrande et al., 2019), significantly reducing the computation involved in each time step.

2.5 Useful Covariance Functions in SDE Form

Here we list the covariance functions (kernels) used in this thesis, in their canonical and SDE forms. For a more exhaustive list see Solin (2016).

Exponential By considering the Ornstein-Uhlenbeck process, Eq. (2.24), we derived the exponential kernel, Eq. (2.26), which we write again here for completeness:

$$C_{\exp}(t, t') = \sigma^2 \exp\left(-\frac{|t - t'|}{\ell}\right) \tag{2.34}$$

The SDE analogue has model matrices $\mathbf{F} = -\lambda$, $\mathbf{L} = 1$, $\mathbf{Q}_c = q$, $\mathbf{P}_\infty = q/2\lambda$, and equivalence is obtained when we set $\sigma^2 = q/2\lambda$ and $\ell = 1/\lambda$. We consider σ^2 to be the magnitude parameter, and ℓ the characteristic lengthscale.

Matérn The Matérn class (Matérn, 1960) has general form:

$$C_{\mathrm{Mat}}(t, t') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}|t - t'|}{\ell}\right)^{\nu} B_\nu\left(\frac{\sqrt{2\nu}|t - t'|}{\ell}\right) \tag{2.35}$$

where $\Gamma(\nu)$ is the Gamma function and $B_\nu(\cdot)$ is the modified Bessel function of the second kind (Abramowitz and Stegun, 1965). The Matérn class is additionally parameterised by ν, and the resulting process is n-times differentiable if $n < \nu$, and hence ν controls the stiffness, or smoothness, of the function.

Eq. (2.35) simplifies for half-integer ν, and has an exact SDE representation (Hartikainen and Särkkä, 2010). Letting $\nu = 3/2$, the covariance function is:

$$C_{\mathrm{Mat3/2}}(t, t') = \sigma^2 \left(1 + \frac{\sqrt{3}|t - t'|}{\ell}\right) \exp\left(-\frac{\sqrt{3}|t - t'|}{\ell}\right) \tag{2.36}$$

In this case, the SDE model components are:

$$\mathbf{F} = \begin{pmatrix} 0 & 1 \\ -\lambda^2 & -2\lambda \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \mathbf{P}_\infty = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \lambda^2\sigma^2 \end{pmatrix}, \tag{2.37}$$

for $\lambda = \sqrt{3}/\ell$. The noise process in the SDE model has spectral density $\mathbf{Q}_c = 4\lambda^3\sigma^2$. The measurement matrix takes the form $\mathbf{H} = (1 \quad 0)$.

Exponentiated Quadratic Also known as the radial basis function (RBF) or squared exponential, this kernel is infinitely differentiable, and is an extension of the Matérn kernel as $\nu \to \infty$. Its spectral density takes a Gaussian form by design, and it is defined as:

$$C_{\mathrm{eq}}(t, t') = \sigma^2 \exp\left(-\frac{(t-t')^2}{2\ell^2}\right) \tag{2.38}$$

The infinite differentiability of C_{eq} means that its SDE form would require us to construct a state vector $\tilde{\mathbf{f}}$ of infinite height, however approximations can be made based on Taylor series expansions (Hartikainen and Särkkä, 2010) or Padé approximations (Särkkä and Piché, 2014).

Quasi-Periodic Solin and Särkkä (2014b) demonstrated how periodic kernels can also be written in state space form. Quasi-periodic processes can be constructed via the product of a periodic kernel and another arbitrary kernel. We consider an example comprising the cosine and exponential kernels:

$$C_{\mathrm{q-per}}(t, t') = \sigma^2 \cos(\omega(t-t')) \exp\left(-\frac{|t-t'|}{\ell}\right) \tag{2.39}$$

We can construct the SDE form of the feedback matrix for a product of kernels by taking the Kronecker sum of the component matrices:

$$\begin{aligned} \mathbf{F} &= \mathbf{F}_{\mathrm{cos}} \oplus \mathbf{F}_{\mathrm{exp}} \\ &= \begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix} \otimes \mathbf{I}_1 + \mathbf{I}_2 \otimes -\lambda = \begin{pmatrix} -\lambda & -\omega \\ \omega & -\lambda \end{pmatrix}, \end{aligned} \tag{2.40}$$

and the other model matrices are calculated via the Kronecker product of the corresponding components such that $\mathbf{L} = \mathbf{I}_2$, $\mathbf{Q}_c = 2\lambda\sigma^2\mathbf{I}_2$, $\mathbf{P}_\infty = \sigma^2\mathbf{I}_2$.

Spectral Mixture A sum of (quasi-)periodic kernels is called a spectral mixture kernel (Wilson and Adams, 2013). This type of kernel was originally conceived to be a sum of cosine-modulated exponentiated quadratic functions, in order to generate a process whose spectral density is a sum of Gaussians. For one-dimensional input this is written:

$$C_{\mathrm{sm}}(t,t') = \sum_{d=1}^{D} \cos(\omega_d(t-t'))C_{\mathrm{eq}}^{(d)}(t,t') \tag{2.41}$$

It has also been proposed to construct a Matérn spectral mixture, a sum of multiple $C_{\mathrm{q-per}}(t,t')$ functions, which results in a kernel with Cauchy-Lorentz spectral density (Alvarado and Stowell, 2017). We discuss the state space form of these kernels, as well as their equivalence to the PPV model, Eq. (2.15), in chapter 3.

2.6 Latent Force Models

The models proposed so far are generally motivated by observations (i.e. prior knowledge) regarding the statistical behaviour of signals. An alternative to this approach is to write down the deterministic physical process by which the observed signal was generated. See Smith (2010) for a detailed discussion of the various physical modelling approaches for audio signal processing.

A hybrid statistical-physical method that incorporates physical assumptions into the Bayesian inference paradigm is called a latent force model (LFM, Alvarez et al., 2009). We consider the physical system in which D observed output functions, $a_d(t)$, $d = 1, \dots, D$, are produced by some $N < D$ latent functions, $f_n(t)$, $n = 1, \dots, N$, being forced through a set of first-order differential equations:

$$\frac{\mathrm{d}a_d(t)}{\mathrm{d}t} + U_d a_d(t) = \sum_{n=1}^{N} V_{d,n} f_n(t), \tag{2.42a}$$

$$f_n(t) \sim \mathrm{GP}(0, C_{\mathrm{Mat}3/2}^{(n)}(t,t')). \tag{2.42b}$$

U_d can be interpreted as the *damping* parameter of output a_d, and $V_{d,n}$ as the *sensitivity* of a_d in response to f_n. We have assumed a Matérn-3/2 GP prior over f_n.

If this set of differential equations represents some physical behaviour in the system we are modelling, even if only in a simplistic manner, then modifying our covariance function to incorporate Eq. (2.42a) can improve our ability to perform inference (Alvarez et al., 2013). In this example the covariance of the outputs a_i and a_j can be calculated analytically, as can the cross-covariance between a force f_n and an output a_d (see Alvarez et al. (2009) for the full formulation).

Hartikainen et al. (2012) showed that LFMs can be reformulated in SDE form, Eq. (2.16). For the first-order ODE above, letting $\dot{f}(t)$ be the time derivative of $f(t)$ and still assuming a Matérn-3/2 kernel, the state vector and model matrices are

$$
\begin{aligned}
\tilde{\mathbf{f}}(t) &= \begin{pmatrix} a_1(t) \\ \vdots \\ a_D(t) \\ f_1(t) \\ \dot{f}_1(t) \\ \vdots \\ f_N(t) \\ \dot{f}_N(t) \end{pmatrix}, \; \mathbf{L} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 1 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \; \mathbf{H} = \begin{pmatrix} 1 & \dots & 1 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}, \\
\mathbf{F} &= \begin{pmatrix}
-U_1 & & & V_{1,1} & & & V_{1,N} & \\
& \ddots & & \vdots & & & \vdots & \\
& & -U_D & V_{D,1} & & & V_{D,N} & \\
& & & 0 & 1 & & & \\
& & & -\lambda_1^2 & -2\lambda_1 & & & \\
& & & & & \ddots & & \\
& & & & & & 0 & 1 \\
& & & & & & -\lambda_N^2 & -2\lambda_N
\end{pmatrix}, \\
\mathbf{P}_\infty &= \begin{pmatrix}
0 & & & & & & & \\
& \ddots & & & & & & \\
& & 0 & & & & & \\
& & & \sigma_1^2 & 0 & & & \\
& & & 0 & \lambda_1^2\sigma_1^2 & & & \\
& & & & & \ddots & & \\
& & & & & & \sigma_N^2 & 0 \\
& & & & & & 0 & \lambda_N^2\sigma_N^2
\end{pmatrix}.
\end{aligned}
\tag{2.43}
$$

The bottom right block-partitions of $\mathbf{F}$ and $\mathbf{P}_\infty$ represent the feedback and stationary covariance matrices for the individual Matérn-$^3/_2$ GPs as outlined in Eq. (2.37). Therefore a different choice of covariance function requires modification of these sub-matrices. Inference in the above LTI SDE proceeds with Kalman filtering and RTS smoothing as usual.

Nonlinear extensions of the LFM, which arise when we wish to impose positivity on the latent functions, $f_n(t)$, can also be handled. Hartikainen et al. (2012) use sigma-point methods to approximate the intractable integrals required during nonlinear filtering. In chapter 5 we follow the same approach to model the amplitude envelopes of the vibrating modes of natural sound events.

2.7 Deep Gaussian Processes

All the models proposed above take known input (time) and project it through a non-parametric GP mapping to produce the observed data, often via a separate parametric mapping, such as the ODE in the latent force model, Eq. (2.42a), the sum operation in the Bayesian time-frequency analysis model, Eq. (2.15b), or the NMF and softplus mappings in the GTF-NMF model, Eq. (2.11). Deep Gaussian processes (Damianou and Lawrence, 2013) remove the need to specify these parametric mappings by replacing them with another GP. Their form as described by Bui et al. (2016), in which L layers of GPs are stacked such that the output of one GP is treated as the input to the GP in the next layer, is

$$\begin{aligned} f_l(h_{l-1,\cdot}) &\sim \mathrm{GP}(\mathbf{m}_l, \mathbf{K}_l), \quad l = 1\dots, L \\ \mathbf{h}_l \,|\, f_l &\sim \prod_{k=1}^{T} \mathrm{N}(f_l(h_{l-1,k}), \sigma_l^2), \quad h_{0,k} = t_k \\ \mathbf{a}_{1:D} &\sim \prod_{k=1}^{T} p(\mathbf{a}_{1:D,k} \,|\, f_L(h_{l-1,k})), \end{aligned} \tag{2.44}$$

for $l = 1, \dots, L$, where we have assumed Gaussian noise between the layers, and the input to the first layer is time, $\mathbf{h}_0 = \mathbf{t}$. We show a version with one-dimensional GPs here, but a more general model with multi-output kernels (Alvarez et al., 2012) is common, particularly since our observed data, $\mathbf{a}_{1:D}$, is D-dimensional.

Inference in this multi-layer model is notoriously difficult, however recent advances have shown them to be applicable to large datasets, outperforming deep (Bayesian) neural networks on many tasks (Salimbeni and Deisenroth, 2017). All approaches to date employ the inducing point method, with inference carried out via power expectation propagation (Bui et al., 2016), stochastic variational inference (Salimbeni and Deisenroth, 2017) and most recently stochastic gradient Hamiltonian Monte Carlo (Havasi et al., 2018), which has the advantage of being able to represent non-Gaussian posterior distributions.

2.8 Conclusion

Now that we have defined our terminology and laid out the relevant background relating to audio analysis and Gaussian processes, we are ready to present our investigation into the connection between GPs and traditional signal processing tools. We begin in chapter 3 with time-frequency analysis.

Chapter 3

Gaussian Models for Time-Frequency Analysis

In chapter 2 we outlined the theoretical ideas surrounding statistical inference and Gaussian process modelling. In this chapter we focus these tools on *time-frequency analysis*: the task of uncovering the time-varying spectral components of a one-dimensional time-domain signal. So far, the introduction of the SDE form of GPs was motivated by computational issues, but here we utilise the dual formulation to draw explicit links between the use of GPs as a tool for machine learning and their use in Bayesian signal processing. These links lead to a better understanding of all the modelling assumptions implicit in probabilistic time-frequency analysis. But in a more practical sense they allow for greater flexibility in both modelling and inference.

3.1 Probabilistic Time-Frequency Analysis

Traditional time-frequency (TF) analysis, a ubiquitous part of the signal processing chain for many real-world applications, requires various choices to be made regarding windowing functions, filter transfer functions, or wavelet functions, depending on the representation being used (Cohen, 1995). There is no consensus on how best to make these choices, for example the filter coefficients $\boldsymbol{\theta}_{\text{filt}}$ in Eq. (2.2), and their implications on downstream tasks is unclear. This drawback is particularly noticeable when TF analysis is used as a pre-processing tool for machine learning applications such as classification or source separation, where training times are long but the input data can vary significantly. In this scenario, a suboptimal one-size-fits-all TF representation is common, since

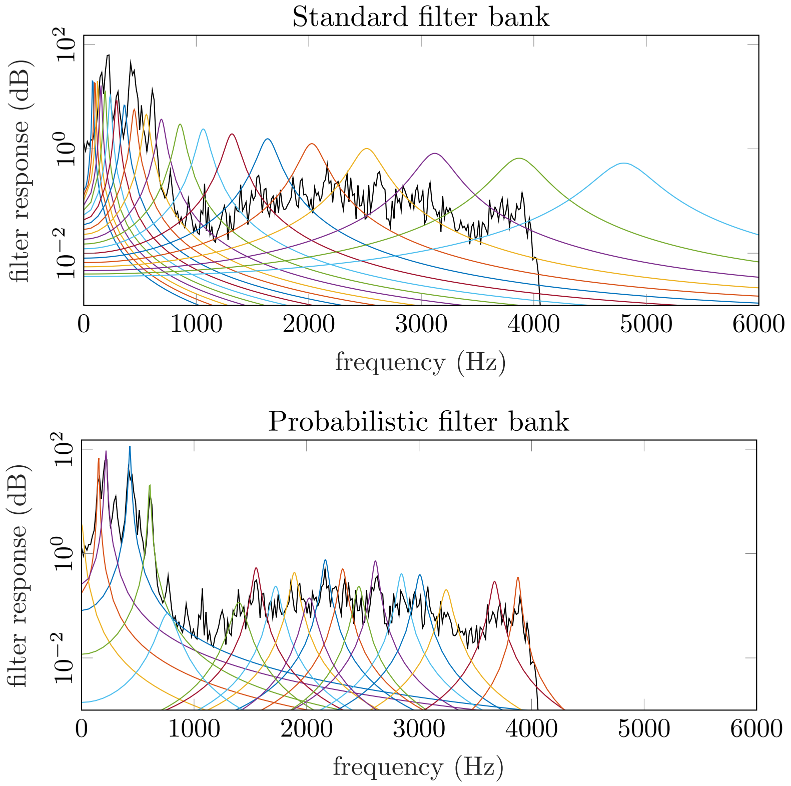


Figure 3.1: A comparison between a standard filter bank typically used in TF analysis (**top**), and a probabilistic filter bank (**bottom**), whose centre frequencies, magnitudes and bandwidths have been optimised to fit to the spectrum of a speech signal (shown in black).

manual testing of all possible parameter settings for all input signals is completely impractical.

Probabilistic TF analysis promises to remove the need for these difficult decisions by adapting to the incoming signal (Qi et al., 2002; Cemgil and Godsill, 2005; Sejdić et al., 2009; Zhong and Huang, 2010) and by propagating uncertainty information to downstream applications (Gillespie and Atlas, 2001; Turner and Sahani, 2014). By specifying a probabilistic model characterised by hyperparameters $\boldsymbol{\theta}$, a posterior distribution over the frequency components given the data, $p(\mathbf{s}_d \mid \mathbf{y}, \boldsymbol{\theta})$, can be found. Different modelling choices can be compared in a principled manner by evaluating the model likelihood given the parameters, $p(\mathbf{y} \mid \boldsymbol{\theta})$, which allows for parameter tuning in order to find the statistically optimal TF representation for a given signal. Figure 3.1 demonstrates the difference between filter banks used in the traditional TF analysis approach vs. those used in probabilistic TF analysis, namely that the probabilistic version adapts to the signal.

As outlined in section 2.2, existing state-of-the-art methods for probabilistic TF analysis can be viewed as modifications of the probabilistic phase vocoder (PPV, Cemgil and Godsill, 2005), which we rewrite here,

$$s_{d,k} = \psi_d \mathrm{e}^{i\omega_d} s_{d,k-1} + \rho_d \zeta_{d,k}, \tag{3.1a}$$

$$y_k = \sum_{d=1}^{D} \mathrm{Re}[s_{d,k}] + \sigma_{y_k} \varepsilon_k, \tag{3.1b}$$

where $s_{d,k} \in \mathbb{C}$ is a complex phasor, $\zeta_{d,k} \sim \mathrm{CN}(0,1)$ is i.i.d. complex Gaussian noise and the likelihood sums the real parts of $s_{d,k}$ plus noise to produce the observation y_k. Parameters ψ_d and ρ_d represent the process and noise standard deviations respectively, whilst ω_d is the instantaneous angular frequency and σ_{y_k} is the observation noise standard deviation.

Such a probabilistic model that acts directly on the signal waveform implicitly captures correlation between a signal's amplitude and phase information (Turner, 2010), which has the major implication that time-domain synthesis does not require the phase-reconstruction stage necessary in many traditional methods, which often introduces artefacts. Fitting the model parameters to the signal provides us with the ability to sample new data from the underlying generative model, making missing data imputation and denoising tasks intuitive.

Despite these benefits, existing probabilistic TF models are still not widely used, perhaps due to their higher computational complexity and because they are formulated in such a way that they can be difficult to interpret and understand.

3.2 Spectral Mixture Gaussian Processes

In Wilson and Adams (2013), GPs, along with their neural network counterparts, are presented as *"intelligent agents"* capable of automating the learning and decision making process. It is shown how detailed prior knowledge can be encoded in the system by constructing new covariance functions composed of the sum and product of simpler ones, resulting in spectral mixture kernels $C_{\mathrm{sm}}(t, t')$, Eq. (2.41).

These flexible multi-component kernel structures were initially conceived for the general task of automatic pattern detection, addressing the fact that in many modelling tasks it is unclear what covariance functions should be used. However, Alvarado et al. (2019) adapted them to modelling musical audio signals by treating each of the components as a harmonic of the signal. This method outperformed spectrogram-based techniques in a source separation task of uncovering the individual notes played during a simple musical sequence of two-note chords.

Until now, this Gaussian process model has not been formulated as an SDE, and inference has been carried out via the inducing point method which suffers from the issues outlined in section 2.4. Additionally, the fact that spectral mixture models are a sum of periodic components suggests a connection to the PPV, which has regularly been noted to be a GP model (see, for example, Turner and Sahani (2014)), but the precise relationship between these two methods has not previously been explored. In the next section we address both these shortcomings.

3.3 Unifying Probabilistic Models for Time-Frequency Analysis

Here we show that probabilistic TF analysis and Matérn spectral mixture GPs are in fact equivalent. In other words, spectral mixture kernels are probabilistic filter banks. By doing so we reinterpret TF modelling assumptions under the GP paradigm and provide a general procedure for rewriting spectral mixture GPs in discrete state space form, such that more complex TF models can be easily constructed, and inference can be performed efficiently via Kalman smoothing, whose computational complexity scales linearly in the number of time steps T and cubicly in state dimensionality M, $\mathcal{O}(M^3T)$.

We start by recognising that Eq. (3.1) represents a (discrete) complex first-order autoregressive process, and hence it can be written in state space form if we construct a state vector by stacking the real and imaginary components, $\tilde{\mathbf{s}}_k = \begin{pmatrix}\mathrm{Re}[s_{1,k}] & \mathrm{Im}[s_{1,k}] & \dots & \mathrm{Re}[s_{D,k}] & \mathrm{Im}[s_{D,k}]\end{pmatrix}^\top$, such that

$$\tilde{\mathbf{s}}_{k+1} = \mathbf{A}\tilde{\mathbf{s}}_k + \mathbf{q}_k, \qquad \mathbf{q}_k \sim \mathrm{N}(\mathbf{0}, \mathbf{Q}), \tag{3.2a}$$

$$y_k = \mathbf{H}\tilde{\mathbf{s}}_k + \sigma_{y_k}\varepsilon_k. \tag{3.2b}$$

The measurement model selects the real components and sums them, $\mathbf{H} = \begin{pmatrix}1 & 0 & \dots & 1 & 0\end{pmatrix}$, and the transition and process noise covariance matrices are

$$\mathbf{A} = \begin{pmatrix} \psi_1 \mathbf{R}(\omega_1) & & 0 \\ & \ddots & \\ 0 & & \psi_D \mathbf{R}(\omega_D) \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} \rho_1^2 \mathbf{I}_2 & & 0 \\ & \ddots & \\ 0 & & \rho_D^2 \mathbf{I}_2 \end{pmatrix}, \tag{3.3}$$

for rotation matrix $\mathbf{R}(\omega_d) = \begin{pmatrix} \cos\omega_d & -\sin\omega_d \\ \sin\omega_d & \cos\omega_d \end{pmatrix}$. This model is now in the form Eq. (2.20), and as such, inference can proceed via Kalman filtering and smoothing.

The PPV as a spectral mixture GP The transition matrix $\mathbf{A}$ in Eq. (3.3) hints at a connection to the quasi-periodic kernel in Eq. (2.40) if we notice that the rotation $\mathbf{R}(\omega_d)$ is the discrete version of the cosine kernel feedback $\mathbf{F} = \left(\begin{smallmatrix} 0 & -\omega_d \\ \omega_d & 0 \end{smallmatrix}\right)$. It turns out that the model in Eq. (3.2) is in fact the discrete form of a Matérn spectral mixture GP, Eq. (2.41), which we write down now before going on to show their equivalence:

$$f(t) \sim \mathrm{GP}\left(0, \sum_{d=1}^{D} C_{\mathrm{q-per}}^{(d)}(t,t')\right), \tag{3.4a}$$

$$y_k = f(t_k) + \sigma_{y_k}\, \varepsilon_k, \tag{3.4b}$$

for $C_{\mathrm{q-per}}^{(d)}(t,t') = C_{\cos}^{(d)}(t,t')\, C_{\exp}^{(d)}(t,t') = \sigma_d^2 \cos\left(\omega_d(t-t')\right) \exp\left(-\frac{|t-t'|}{\ell_d}\right)$. The kernel for each frequency channel d is a product of the cosine kernel and the exponential kernel (the Matérn-1/2).

Encoded in this model is the assumption that the signal is made up of a sum of D latent components, and that these latent components are periodic, as determined by the cosine kernel whose function realisations are pure sinusoids. But they are not perfectly periodic — the covariance between neighbouring time steps decays exponentially with the gap between steps, as described by $C_{\exp}(t,t')$.

The continuous state space model We now utilise the relationship between periodic kernels and state space models (Solin and Särkkä, 2014b) to write down the SDE version of Eq. (3.4). The model matrices corresponding to the cosine and exponential kernels for the $d = 1, \ldots, D$

quasi-periodic components are

$$
\begin{aligned}
\mathbf{F}^{(d)}_{\cos} &= \begin{pmatrix} 0 & -\omega_d \\ \omega_d & 0 \end{pmatrix}, & \mathbf{F}^{(d)}_{\exp} &= -\frac{1}{\ell_d}, \\
\mathbf{Q}^{(d)}_{c,\cos} &= \text{N/A}, & \mathbf{Q}^{(d)}_{c,\exp} &= \frac{2\sigma_d^2}{\ell_d}, \\
\mathbf{L}^{(d)}_{\cos} &= \text{N/A}, & \mathbf{L}^{(d)}_{\exp} &= 1, \\
\mathbf{P}^{(d)}_{\infty,\cos} &= \mathbf{I}_2, & \mathbf{P}^{(d)}_{\infty,\exp} &= \sigma_d^2, \\
\mathbf{H}^{(d)}_{\cos} &= \begin{pmatrix} 1 & 0 \end{pmatrix}, & \mathbf{H}^{(d)}_{\exp} &= 1.
\end{aligned}
\tag{3.5}
$$

The cosine kernel represents a deterministic process, therefore its SDE form does not have a diffusion term and so $\mathbf{Q}^{(d)}_{c,\cos}$ and $\mathbf{L}^{(d)}_{\cos}$ are not defined.

The product of two kernels can be calculated via the Kronecker sum of the component feedback matrices and the Kronecker product of the remaining component matrices, which gives

$$
\begin{aligned}
\mathbf{F}^{(d)} &= \mathbf{F}^{(d)}_{\cos} \oplus \mathbf{F}^{(d)}_{\exp} = \mathbf{F}^{(d)}_{\cos} \otimes \mathbf{I}_2 + \mathbf{I}_1 \otimes \mathbf{F}^{(d)}_{\exp} = \begin{pmatrix} -\frac{1}{\ell_d} & -\omega_d \\ \omega_d & -\frac{1}{\ell_d} \end{pmatrix}, \\
\mathbf{Q}^{(d)}_{c} &= \mathbf{I}_2 \otimes \mathbf{Q}^{(d)}_{c,\exp} = \frac{2\sigma_d^2}{\ell_d}\mathbf{I}_2, \\
\mathbf{L}^{(d)} &= \mathbf{I}_2 \otimes \mathbf{L}^{(d)}_{\exp} = \mathbf{I}_2, \\
\mathbf{P}^{(d)}_{\infty} &= \mathbf{P}^{(d)}_{\infty,\cos} \otimes \mathbf{P}^{(d)}_{\infty,\exp} = \sigma_d^2\mathbf{I}_2, \\
\mathbf{H}^{(d)} &= \mathbf{H}^{(d)}_{\cos} \otimes \mathbf{H}^{(d)}_{\exp} = \begin{pmatrix} 1 & 0 \end{pmatrix}.
\end{aligned}
\tag{3.6}
$$

These submatrices can now be used to construct the full continuous SDE form of the spectral mixture GP, Eq. (3.4), which is

$$
\frac{\mathrm{d}\tilde{\mathbf{f}}(t)}{\mathrm{d}t} = \mathbf{F}\tilde{\mathbf{f}}(t) + \mathbf{L}\mathbf{w}(t), \qquad \mathbf{w}(t) \sim \mathrm{N}(0, \mathbf{Q}_c) \tag{3.7a}
$$

$$
y_k = \mathbf{H}\tilde{\mathbf{f}}(t_k) + \sigma_y \varepsilon_k, \tag{3.7b}
$$

for state vector $\tilde{\mathbf{f}} = \begin{pmatrix} \mathrm{Re}[f_1(t)] & \mathrm{Im}[f_1(t)] & \dots & \mathrm{Re}[f_D(t)] & \mathrm{Im}[f_D(t)] \end{pmatrix}^\top \in \mathbb{R}^M$ where $M = 2D$, and the model matrices have block-diagonal structure:

$$
\begin{aligned}
\mathbf{F} &= \begin{pmatrix} \mathbf{F}^{(1)} & & & \\ & \mathbf{F}^{(2)} & & \\ & & \ddots & \\ & & & \mathbf{F}^{(D)} \end{pmatrix}, \quad
\mathbf{Q}_c = \begin{pmatrix} \mathbf{Q}_c^{(1)} & & & \\ & \mathbf{Q}_c^{(2)} & & \\ & & \ddots & \\ & & & \mathbf{Q}_c^{(D)} \end{pmatrix}, \\
\mathbf{L} &= \begin{pmatrix} \mathbf{L}^{(1)} & & & \\ & \mathbf{L}^{(2)} & & \\ & & \ddots & \\ & & & \mathbf{L}^{(D)} \end{pmatrix}, \quad
\mathbf{P}_\infty = \begin{pmatrix} \mathbf{P}_\infty^{(1)} & & & \\ & \mathbf{P}_\infty^{(2)} & & \\ & & \ddots & \\ & & & \mathbf{P}_\infty^{(D)} \end{pmatrix}, \\
\mathbf{H} &= \begin{pmatrix} \mathbf{H}^{(1)} & \mathbf{H}^{(2)} & \dots & \mathbf{H}^{(D)} \end{pmatrix}.
\end{aligned} \tag{3.8}
$$

It is important to note that this method of writing the spectral mixture model in SDE form is not specific to the exponential kernel. For example, if the Matérn-3/2 kernel was used instead, we could combine $\mathbf{F}^{(d)}_{\cos}$ and $\mathbf{F}^{(d)}_{\text{Mat3/2}}$ (as well as the other model components) in a similar manner. In this case, since the Matérn-3/2 kernel has second-order state space form, the state vector would additionally contain the real and imaginary parts of the first time derivative and would have dimensionality $M = 4D$.

Returning to discrete state space form LTI SDE models in the form of Eq. (3.7) have an exact discrete-time solution, and the corresponding state space model is given by

$$
\tilde{\mathbf{f}}(t_{k+1}) = \mathbf{A}\tilde{\mathbf{f}}(t_k) + \mathbf{q}_k, \qquad \mathbf{q}_k \sim \mathrm{N}(0, \mathbf{Q}) \tag{3.9a}
$$

$$
y_k = \mathbf{H}\tilde{\mathbf{f}}(t_k) + \sigma_y \varepsilon_k, \tag{3.9b}
$$

where $\mathbf{A} = \exp(\mathbf{F}\Delta t)$ for time step size $\Delta t = t_k - t_{k-1}$ which we assume to be constant, and $\mathbf{Q} = \mathbf{P}_\infty - \mathbf{A}\mathbf{P}_\infty\mathbf{A}^\top$ such that

$$
\begin{aligned}
\mathbf{A} &= \begin{pmatrix} \exp\left(-\frac{\Delta t}{\ell_1}\right)\mathbf{R}(\omega_1 \Delta t) & & \\ & \ddots & \\ & & \exp\left(-\frac{\Delta t}{\ell_D}\right)\mathbf{R}(\omega_D \Delta t) \end{pmatrix}, \\
\mathbf{Q} &= \begin{pmatrix} \sigma_1^2(1 - \exp(-\frac{2\Delta t}{\ell_1}))\mathbf{I}_2 & & \\ & \ddots & \\ & & \sigma_D^2(1 - \exp(-\frac{2\Delta t}{\ell_D}))\mathbf{I}_2 \end{pmatrix},
\end{aligned} \tag{3.10}
$$

where we have used the identity $\exp(X+Y) = \exp(X)\exp(Y)$ to obtain

$$\begin{aligned}\mathbf{A}^{(d)} = \exp(\mathbf{F}^{(d)}\Delta t) &= \exp(-\frac{1}{\ell_d}\mathbf{I}_2\Delta t)\exp(\left(\begin{smallmatrix} 0 & -\omega_d \\ \omega_d & 0 \end{smallmatrix}\right)\Delta t) \\ &= \exp(-\frac{\Delta t}{\ell_d})\mathbf{I}_2 \begin{pmatrix} \cos(\omega_d\Delta t) & -\sin(\omega_d\Delta t) \\ \sin(\omega_d\Delta t) & \cos(\omega_d\Delta t) \end{pmatrix} \\ &= \exp(-\frac{\Delta t}{\ell_d})\mathbf{R}(\omega_d\Delta t).\end{aligned} \tag{3.11}$$

The measurement matrix remains $\mathbf{H} = \begin{pmatrix} 1 & 0 & \ldots & 1 & 0 \end{pmatrix}$.

Written in this form it becomes clear that this Matérn spectral mixture GP is in fact equivalent to the PPV model in Eq. (3.3) if we select the PPV hyperparameters to be $\psi_d = \exp(-\Delta t/\ell_d)$ and $\rho_d = \sigma_d^2(1 - \exp(-2\Delta t/\ell_d))$. If $\Delta t \neq 1$ then we must also scale the frequency parameters by the step size for equivalence: $\omega_d^{(\text{PPV})} = \omega_d^{(\text{SM})}\Delta t$.

This result shows that, despite being developed from a very different perspective, the PPV is a special case of the spectral mixture GP. Looking at the hyperparameter mappings above, a long lengthscale ℓ_d results in high process variance ψ_d^2 and low noise variance ρ_d^2, which characterises a smoothly varying process. Hence we can now also interpret spectral mixture GPs as probabilistic filter banks in which the lengthscales determine the *bandwidth* of the filters.

We will now investigate the benefits of drawing such a link. As we will see, inference methods developed from the signal processing perspective are advantageous, but hyperparameter tuning and modelling flexibility benefit from the GP perspective, making these two paradigms complimentary to each other.

3.4 Hyperparameter Learning in the Frequency Domain

The model proposed above retains a linear Gaussian form, and inference can be carried out in the state space model via Kalman filtering and RTS smoothing (see section 2.3), which scales linearly in time. Additionally, we are able to tune the hyperparameters (filter bandwidths, centre frequency and scale) by minimising the energy function, Eq. (2.32), the negative log marginal likelihood.

However, despite the linear computational scaling, iterated filtering of the entire time domain signal can still be prohibitive for long time series. For this reason, we utilise Bayesian spectrum analysis (Bretthorst,

2013) to tune the hyperparameters in the frequency domain. To do so, we must first consider the spectral properties of our model, and a beneficial side effect of our new unifying perspective is that the kernel-based spectral mixture representation provides us with a straightforward way to calculate the frequency-domain properties of the system via the spectral density of the covariance functions.

Spectral density of the spectral mixture As well as helping us to tune the hyperparameters, calculating the spectral density of the GP prior covariance provides us with a further way to interpret our modelling assumptions. The spectral density is the Fourier transform of the covariance function (letting $\tau = |t - t'|$),

$$S(\omega) = \int_{-\infty}^{\infty} C(\tau)\mathrm{e}^{-i\omega\tau}\,\mathrm{d}\tau. \tag{3.12}$$

The Fourier transform of the exponential kernel, $C_{\text{exp}}^{(d)} = \sigma_d^2 \exp(-\tau/\ell_d)$, results in the spectral density

$$S_{\text{exp}}^{(d)}(\omega) = 2\sigma_d^2\lambda_d(\lambda_d^2 + \omega^2)^{-1}, \qquad \lambda_d = \ell_d^{-1} \tag{3.13}$$

which is a Cauchy-Lorentz function centred at the origin $\omega = 0$. The cosine kernel, whose realisations are pure sinusoids, is represented by a dual peak in the spectral domain at $\pm\omega_d$:

$$S_{\text{cos}}^{(d)}(\omega) = \frac{1}{2}(\delta(\omega - \omega_d) + \delta(\omega + \omega_d)). \tag{3.14}$$

The product of these two kernels in the time domain is equivalent to their convolution in the frequency domain, hence the spectral density of mixture component d is a frequency shifted version of the exponential:

$$\begin{aligned} S_{\text{sm}}^{(d)}(\omega) &= \frac{1}{2}(S_{\text{exp}}^{(d)}(\omega - \omega_d) + S_{\text{exp}}^{(d)}(\omega + \omega_d)) \\ &= \sigma_d^2\lambda_d\left((\lambda_d^2 + (\omega - \omega_d)^2)^{-1} + (\lambda_d^2 + (\omega + \omega_d)^2)^{-1}\right), \end{aligned} \tag{3.15}$$

again with $\lambda_d = \ell_d^{-1}$. A demonstration of this is shown in Figure 3.2.

Bayesian spectrum analysis Now that we have access to the spectral density of the full spectral mixture model, $S_{\text{sm}}(\omega) = \sum_{d=1}^{D} S_{\text{sm}}^{(d)}(\omega)$, we follow the approach of Turner and Sahani (2014) to fit the parame-

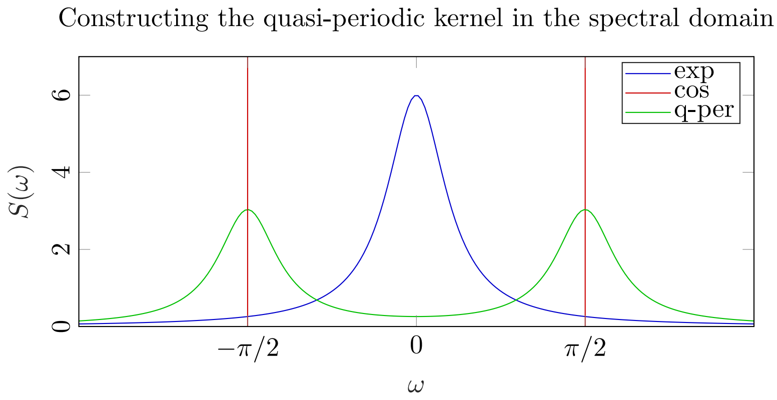


Figure 3.2: The cosine kernel acts as a frequency shift operator on the exponential kernel to produce the quasi-periodic kernel, i.e. one component of the spectral mixture kernel ($\sigma_d^2 = 1$, $\ell_d = 3$, $\omega_d = \pi/2$).

ters of the model via a maximum likelihood approach in the frequency domain.

We take the discrete Fourier transform of the observed signal to calculate the power in frequency bin j as $|\tilde{y}_j|^2 = |\sum_{k=1}^{T} \mathrm{FT}_{j,k} y_k|^2$. Now, letting $\gamma_{y,j}(\boldsymbol{\theta}) = S_{\mathrm{sm}}(\omega_j) + T\sigma_y^2$ where ω_j is the centre frequency of bin j, we obtain the frequency domain form of the log marginal likelihood,

$$\log p(\mathbf{y} \,|\, \boldsymbol{\theta}) = c_{\mathrm{prior}} - \frac{1}{2} \sum_{j=1}^{T} \left(\log(\gamma_{y,j}(\boldsymbol{\theta})) + \frac{|\tilde{y}_j|^2}{\gamma_{y,j}(\boldsymbol{\theta})} \right), \tag{3.16}$$

where c_{prior} is the hyper-prior contribution (a Gamma prior is placed over the variance hyperparameters). Calculating Eq. (3.16) is much more efficient than running the Kalman filter; however a practical modification to the learning algorithm is made by Turner and Sahani (2014) in which the signal spectrum is smoothed by replacing $|\tilde{y}_j|^2$ with Welch's periodogram computed on multiple data segments. This helps avoid local optima during the training process, with its effect being annealed over time by reducing the number of data segments used until eventually the process fits to the original noisy spectrum.

A final practical consideration is that hyperparameter optimisation benefits from parametrising the model with the marginal variances $\sigma_{d,\mathrm{marg}}^2$ rather than the conditional variances σ_d^2, where $\sigma_{d,\mathrm{marg}}^2 = \sigma_d^2/(1-\lambda_d^2)$. The parameters are better behaved in the marginal space since, at least for the Matérn class of kernels, this mapping accounts for the dependence between σ_d^2 and λ_d.

The benefits of using the kernel spectral density The frequency domain learning outlined above demonstrates the benefits of considering the signal processing perspective for GP models. However, the GP kernels themselves provide a significant advantage in this setting. When the PPV model was not known to be a spectral mixture GP, its kernel representation was unclear, and the spectral density S_{sm} had to be calculated via consideration of the model's autocorrelation function. This is straightforward for a first- or second-order autoregressive process, but as the order increases this becomes much more cumbersome.

Furthermore, in the autoregressive filter setting, stationarity of the filters must be ensured by calculating and implementing parameter constraints for the model coefficients ψ_d, ρ_d (Turner, 2010). The GP kernel approach sidesteps both these practical issues because calculation of the spectral density is done via the Fourier transform, and kernel stationarity is guaranteed by design.

3.5 Modifying the Time-Frequency Kernel

We have seen some benefits of the unified treatment of these probabilistic TF analysis models in terms of inference, but perhaps the most significant advantage is to modelling flexibility, since all the stationary kernels designed in the GP community can now be applied in the TF analysis setting. Extensions to the PPV model in its standard form involved constructing higher-order autoregressive processes, but we can now get the same effect via kernels that admit higher-order Markov representations.

Consider again the assumptions encoded in the model prior, Eq. (3.1). Its first-order state space form implies that the instantaneous frequency is not correlated through time, since the gradient of the process at time step t_k has no influence over the gradient at step t_{k+1}. A higher-order model would produce smoother sample paths that exhibit slowly varying instantaneous frequencies, a feature of real-world audio signals that should be leveraged to aid the highly ill-posed task of inferring a TF representation from data.

Therefore, one intuitive example of a way to update the model is to swap the exponential kernel (Matérn-$1/2$, with state dimensionality $M = 2D$) with a similar function that admits a higher-order state space representation. This corresponds to a filter bank whose filter transfer

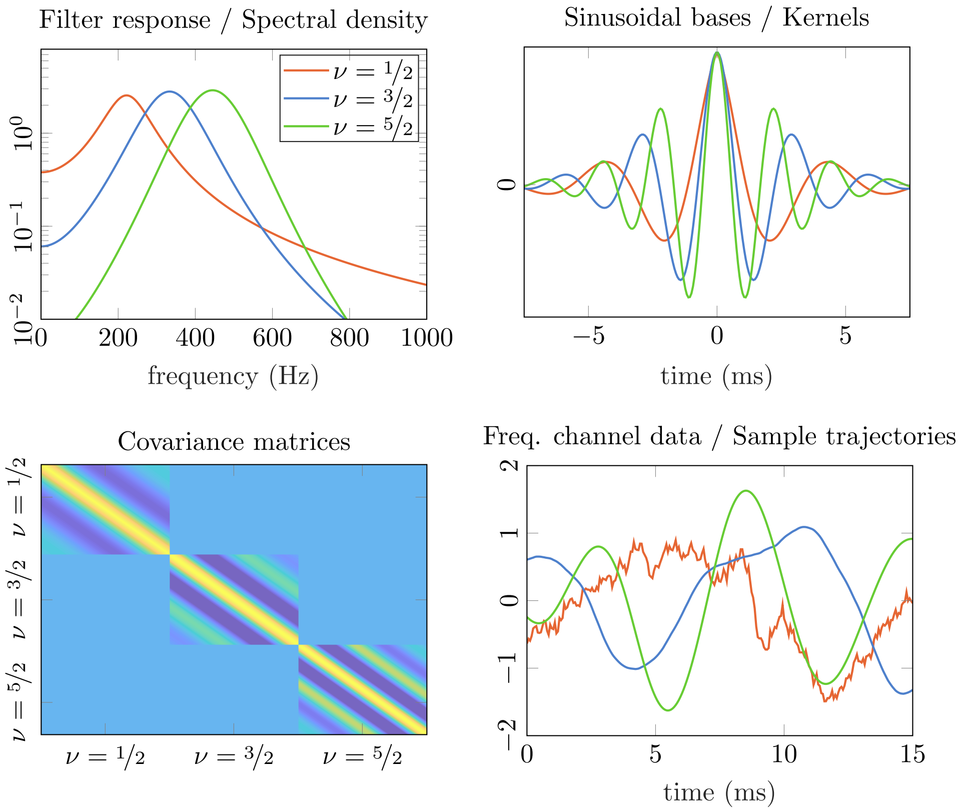


Figure 3.3: Four representations of the same Gaussian process-based probabilistic filter bank (with three filters). Each filter / process is a frequency shifted Matérn-ν GP. All three filters have the same lengthscale (bandwidth) parameter, but they exhibit quite different spectral densities (**top-left**). See main text for demonstration of how filter banks can be represented in canonical GP form, such as with kernels (**top-right**) and covariance matrices (**bottom-left**). Samples from the prior vary in smoothness (**bottom-right**), suggesting that the choice of ν will affect how the model fits the signal.

functions are no longer first-order autoregressive processes, but take a more complex form. We use the Matérn-3/2 ($M = 4D$) and Matérn-5/2 ($M = 6D$) kernels, which correspond to second- and third-order filter banks respectively and whose spectral densities have flatter tails and taller peaks (see Figure 3.3).

The Matérn-3/2 kernel, $C^{(d)}_{\mathrm{Mat}3/2}(\tau) = \sigma_d^2 (1 + \lambda_d \tau) \exp(-\lambda_d \tau)$, has feedback matrices $\mathbf{F}^{(d)} = \begin{pmatrix} 0 & 1 \\ -\lambda_d^2 & -2\lambda_d \end{pmatrix}$ and its spectral density is

$$S^{(d)}_{\mathrm{Mat}3/2}(\omega) = 4\sigma_d^2 \lambda_d^3 (\lambda_d^2 + \omega^2)^{-2}, \qquad \lambda_d = \sqrt{3}\ell_d^{-1} \tag{3.17}$$

The noise effect matrix, $\mathbf{L}^{(d)} = \begin{pmatrix} 0 & 1 \end{pmatrix}^\top$, states that the second order term, i.e. the first time derivative of the process, is influenced by the process noise $\mathbf{w}(t)$, which results in a smoother prior over functions f.

For a Matérn-5/2 kernel, $C^{(d)}_{\mathrm{Mat}5/2}(\tau) = \sigma_d^2 \left(1 + \lambda_d \tau + \frac{\lambda_d^2}{3}\right) \exp\left(-\lambda_d \tau\right)$, it is the second time derivative that is influenced by the noise, $\mathbf{L}^{(d)} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^\top$, and the spectral density is

$$S^{(d)}_{\mathrm{Mat}5/2}(\omega) = \frac{16}{3}\sigma_d^2 \lambda_d^5 (\lambda_d^2 + \omega^2)^{-3}, \qquad \lambda_d = \sqrt{5}\ell_d^{-1} \tag{3.18}$$

with feedback matrices $\mathbf{F}^{(d)} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\lambda_d^3 & -3\lambda_d^2 & -3\lambda \end{pmatrix}$.

Missing data synthesis experiment One way to evaluate updates to the TF model that incorporate these kernels is on a missing data synthesis, or audio inpainting (Adler et al., 2012), task. In general, the better our generative TF model is at representing audio data, the more capable it will be at predicting missing or corrupted segments of the signal. Audio inpainting is useful for a number of real-world applications, including de-clipping, de-clicking, and interference removal.

Each version of the model (Matérn-1/2, Matérn-3/2, Matérn-5/2), with $D = 40$ filters, was trained on 10 short speech excerpts (between 1 and 2 seconds in duration) and then used to calculate the posterior predictive distribution for versions of the recordings in which some data had been removed. Practically, this involves calculating the smoothing distribution, $p(\mathbf{s}_{1:D,1:T} \,|\, \mathbf{y})$, whilst skipping the Kalman update step at the time locations where the data is missing or corrupted.

Missing data gaps of between 1 ms and 20 ms were studied, with the results shown in Figure 3.4. Whilst the differences are subtle (the overall models are similar), the higher-order models' reconstruction achieved an improved signal to noise ratio for all missing data durations averaged across the 10 speakers. We also calculated the PESQ score (Rix et al., 2001) (a standardised perceptual speech quality metric), which demonstrated some signs of improvement, however all models performed similarly for large gap durations.

We found that training the model with second- and third-order kernels was less stable than with the first-order one (the standard PPV). In particular, it was much more sensitive to initial parameter settings. For this reason, in the above experiment we always pre-trained the model with the Matérn-1/2, and used the learnt parameters as the initial setting

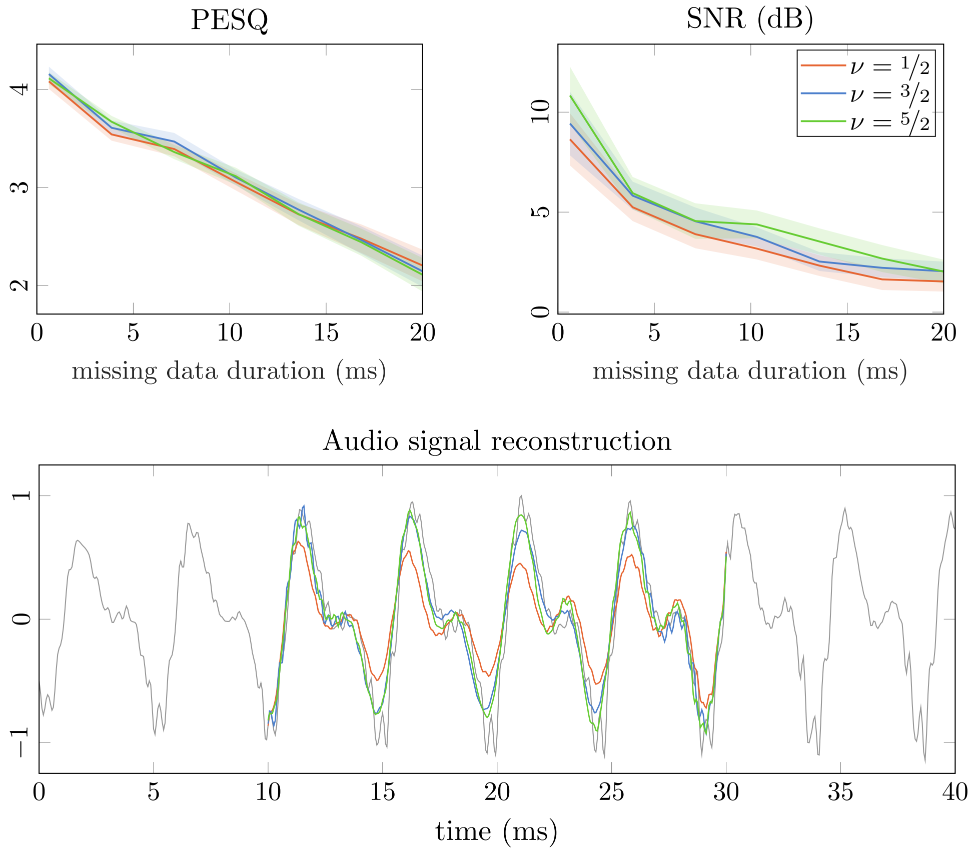


Figure 3.4: Missing data synthesis results for three Matérn-ν probabilistic time-frequency models. Segments of data were removed from 10 speech recordings. Performance measured via perceptual quality metric (**top-left**) and signal-to-noise ratio (**top-right**) as a function of gap duration. Median value across speakers shown (shaded area is standard error). A reconstruction example (**bottom**) shows how the higher-order models ($\nu = 3/2, 5/2$) recover the overall shape in clearer detail (ground truth in grey). Matérn-$1/2$ is the standard probabilistic phase vocoder.

for training with the kernel of interest.

The small differences in results shown in Figure 3.4 are somewhat surprising, given that the stiffer model prior described by the higher-order kernels should better describe the true behaviour of an audio signal. To investigate further, we compare the spectral density of the three model variants in Figure 3.5. The higher-order kernels have rounder peaks but narrower bandwidths in general and flatter tails, as we would expect from smoothly varying processes. This leads to a good representation of the data in most regions of the frequency domain, but in some areas the Matérn-$1/2$ model seems to fit the spectrum more tightly. These observations suggest that there is room for improvement in GP kernel

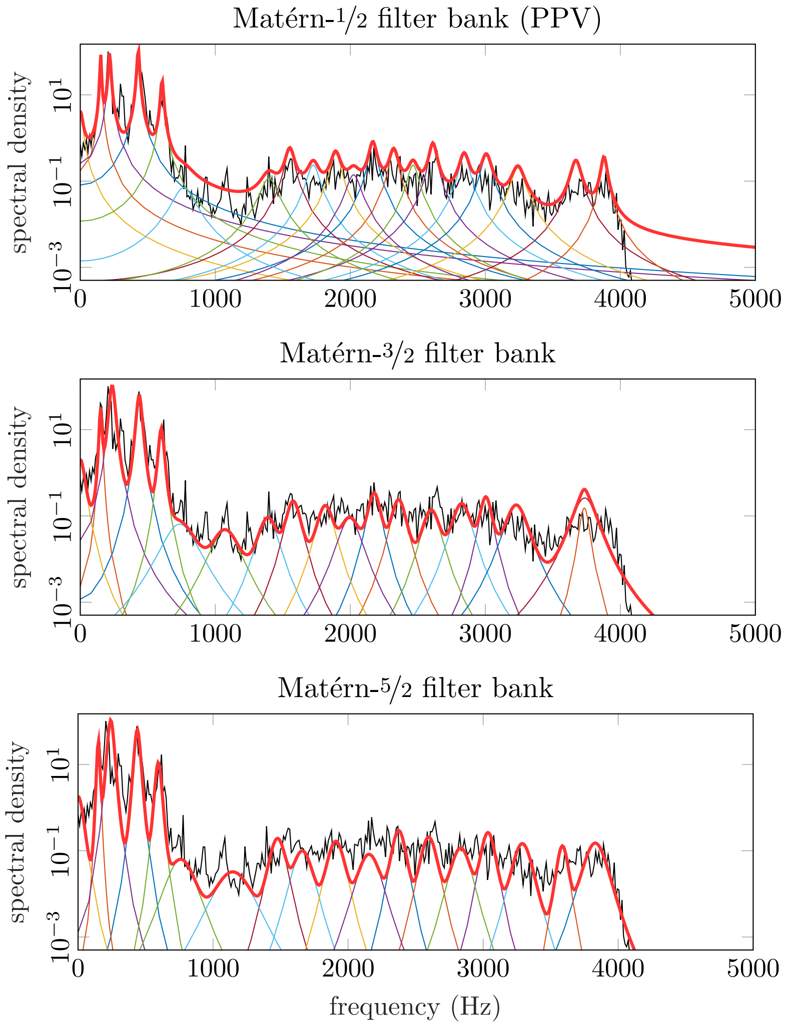


Figure 3.5: Three versions of the probabilistic filter bank / spectral mixture GP are fit to a speech signal (spectrum shown in black). The red lines show the sum of the spectral densities of the component filters / GPs. The Matérn-1/2 model (**top**) has narrower peaks and fatter tails. The Matérn-3/2 (**middle**) and Matérn-5/2 (**bottom**) models have rounder peaks but their tails are flat. The higher-order kernels allow for a more accurate fit in some regions and exhibit smoother sample paths, but more often exhibit pathological behaviour such as the stacking of two filters around 3,700 Hz in the middle plot.

design for audio data. Additionally, the noisy signal spectrum suggests that there is natural variation in the signal not being captured in the model, potentially due to nonstationary behaviour.

3.6 Conclusion

We have unified the theory surrounding probabilistic TF analysis and explained clearly how it relates to Gaussian process modelling, with the aim to motivate further research at the intersection of these fields. The general framework outlined here for converting spectral mixture GP models to a state space form enables efficient frequency domain optimisation and efficient time domain filtering and prediction, showing how these two perspectives are complementary to each other. We applied the framework to Matérn spectral mixture GPs and demonstrated improved performance over the standard probabilistic phase vocoder on a generative task.

The improved modelling flexibility allowed us to make clear comparisons between the competing TF models, but the relatively modest performance gains motivate further work on kernel design specifically for audio signals. Alternatively, it may prove more fruitful to learn the kernel itself (or its spectral density) from the data in a nonparametric fashion, an approach proposed by Tobar et al. (2015).

Practical limitations of probabilistic TF models still remain due to the Kalman smoother's cubic computational scaling in the state dimensionality and from the significant memory requirements involved in storing the entire covariance structure for every time step.

The methods presented here also assume independence across frequency channels and don't explicitly model time-varying amplitude behaviour. Our state space framework provides a foundation on which to construct more complex models that incorporate these features, which we explore next in chapter 4.

Chapter 4

End-to-End Probabilistic Inference for Nonstationary Audio Analysis

A typical (non-probabilistic) way to perform feature analysis on an audio signal is to apply nonnegative matrix factorisation (NMF) to the amplitude components $\mathbf{a}_d$ of a time-frequency (TF) representation – the spectrogram. This disjoint approach has limitations, as outlined in Turner and Sahani (2014). It discards phase information calculated during the TF stage, as well as dependencies between TF coefficients, and it fails to capture and share any uncertainty information between the analysis stages, which could be useful in determining an appropriate signal decomposition in such an ill-posed setting.

Moreover, the map that takes the waveform to the space of TF coefficients is not a bijection. This means that any function operating on the signal in the TF domain (e.g. noise removal), especially those that act only on the magnitude component, might push the signal outside the manifold of realisable waveforms (Turner, 2010). Hence, the modified TF representation must be projected back to the manifold of valid TF representations before the waveform can be re-synthesized (see, e.g., Griffin and Lim (1984)). This projection might distort the signal and introduce undesirable artefacts.

These issues have motivated a large body of research on probabilistic models that operate directly on signal waveforms rather than on TF representations. Such models have been shown to outperform their spectrogram-based counterparts on several tasks, including source separation (Liutkus et al., 2011; Alvarado et al., 2019; Magron and Virtanen, 2019), audio inpainting and denoising (Badeau and Plumbley, 2014; Turner and Sahani, 2014).

In chapter 3 we showed that probabilistic TF analysis can be performed using a GP model whose kernel is a sum of quasi-periodic functions. An extension to this approach, which we introduced in section 2.2, is the Gaussian time-frequency NMF (GTF-NMF) model for joint TF analysis and spectrogram feature analysis. The observation mechanism in this joint model is a nonlinear function of the latent components, which makes inference non-trivial and previous work relies on a suboptimal inference scheme, where the separate model components are updated independently in an iterative fashion.

In this chapter we devise a fully probabilistic joint inference method for the GTF-NMF model based on power expectation propagation in the Kalman smoother setting. First we construct its state space form, showing how it can be viewed as a spectral mixture GP with nonstationary NMF priors over the amplitude variance parameters, (see Figure 4.1 for an overview of the idea).

We also consider new ways to deal with computational issues that arise in this nonlinear model. We construct its infinite-horizon GP representation (Solin et al., 2018), which scales as $\mathcal{O}(M^2T)$ in complexity and $\mathcal{O}(MT)$ in memory, where M is the dimensionality of the state and T the number of time steps. Then we show performance of our approximate inference scheme on various tasks, and compare it to a classical signal processing approach: the iterated extended Kalman filter.

4.1 The Gaussian Time-Frequency NMF Model

We aim to decompose the signal $\mathbf{y}$ into D unknown frequency (oscillator) channels, whose relative amplitudes are modulated by N temporal NMF components. The GP *priors* for the $D + N$ latent model component functions are:

$$g_n(t) \sim \mathrm{GP}(0, C_{\mathrm{g}}^{(n)}(t,t')), \qquad n = 1, 2, \dots, N, \tag{4.1a}$$

$$z_d(t) \sim \mathrm{GP}(0, C_{\mathrm{z}}^{(d)}(t,t')), \qquad d = 1, 2, \dots, D, \tag{4.1b}$$

where $g_n(t)$ denotes the n^{th} temporal NMF component function and $z_d(t)$ the d^{th} frequency channel. The kernel $C_z^{(d)} = C_{\text{q-per}}^{(d)}$ is chosen to be

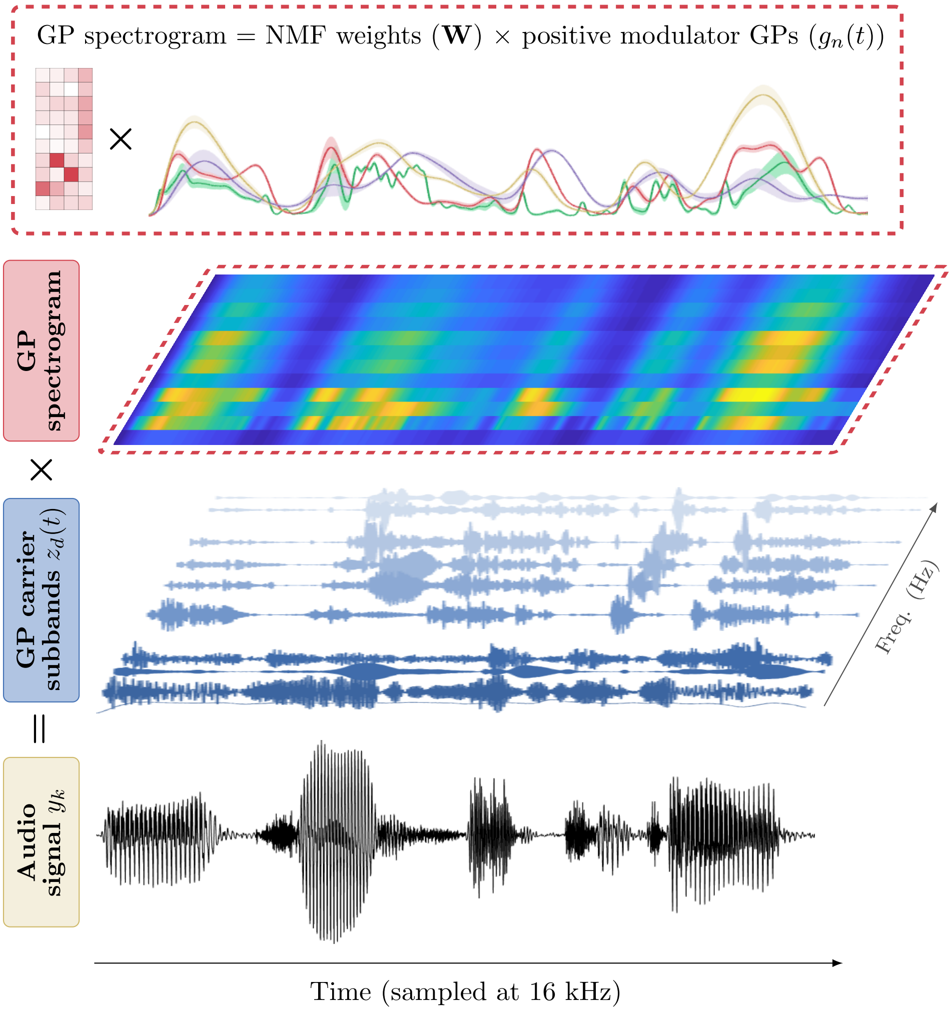

GP spectrogram = NMF weights ($\mathbf{W}$) × positive modulator GPs ($g_n(t)$)
×
GP spectrogram
×
GP carrier subbands $z_d(t)$
=
Audio signal y_k
Freq. (Hz)
Time (sampled at 16 kHz)

Figure 4.1: Nonstationary modelling of audio data. The input (**bottom**) is a sound recording of female speech. We seek to decompose the signal into Gaussian process carrier waveforms (blue block) multiplied by a spectrogram (red block). The spectrogram is learned from the data as a nonnegative matrix of weights times positive modulators (**top**).

a quasi-periodic function, i.e. the d^{th} component of a spectral mixture. $C_g^{(n)}$ should be determined by our assumptions about the behaviour of the amplitude modulators, such as their smoothness properties.

The likelihood (observation) model, letting $a_{d,k} = a_d(t_k)$ and $z_{d,k} = z_d(t_k)$, is given by:

$$y_k = \sum_d a_{d,k}\, z_{d,k} + \sigma_y\, \varepsilon_k, \tag{4.2}$$

for squared amplitudes (the magnitude spectrogram):

$$a_d^2(t_k) = \sum_n W_{d,n}\, \phi(g_n(t_k)). \tag{4.3}$$

We model the squared amplitudes in order to encourage the analogy between this model and a traditional approach, in which the spectrogram is the square of the STFT magnitude.

Positivity of the NMF components is enforced by a link function, in our case the softplus $\phi(g_n) = \log(1 + \mathrm{e}^{g_n})$. $\mathbf{W} \in \mathbb{R}^{D \times N}$ are the NMF weights determining how strongly each modulator affects each oscillator. Crucially, if we choose $N < D$, then the model captures amplitude behaviour shared across frequency channels, i.e. co-modulation, one of the important perceptual characteristics of sound discussed in section 2.2.

Note that if we set $a_d(t_k) = 1$, $\forall\, d, k$ then Eq. (4.2) reduces to standard probabilistic time-frequency analysis, the model given in chapter 3. If we discard $z_d(t_k)$ by calculating a fixed spectrogram, such that $a_d^2(t_k)$ become our observations, then Eq. (4.3) is standard temporal NMF (Bertin et al., 2010; Turner and Sahani, 2014). Further removing the GP prior over g_n brings us back to the NMF model in Eq. (2.14).

Figure 4.1 shows the model diagrammatically – the frequency channel subbands z_d are D independent, unit variance GPs with quasi-periodic kernel functions. The modulators g_n and the NMF weights constitute a model for the spectrogram, the squared amplitudes of the frequency channels.

The inference methods we will present next allow for any choice of C_g, C_z, so long as they can be written in state space form, either approximately or exactly (see section 2.5). We focus on the Matérn class of kernel functions due to their strong connection to autoregressive filters, and because their parameters have convenient interpretations for our task – their lengthscales and variances relate to the bandwidth and scale of the filters in a filter bank (see chapter 3).

4.2 Nonstationary Spectral Mixture GPs

If we write down our model in its hierarchical form, we observe a striking similarity to the nonstationary spectral mixture GPs presented in Remes et al. (2017). These nonstationary models treat the hyperparameters of a spectral mixture model as functions of time, and place a GP hyper-prior over them. Such a model can be written

$$s(t) \sim \mathrm{GP}\left(0, \sum_{d=1}^{D} \sigma_d(t)\sigma_d(t') \cos\big(\omega_d(t)t - \omega_d(t')t'\big) C_d\big(t, t' \,|\, \ell_d(t), \ell_d(t')\big)\right), \tag{4.4a}$$

$$y_k = s(t_k) + \sigma_y \, \varepsilon_k, \tag{4.4b}$$

with hyper-priors

$$\begin{aligned} \log \sigma_d^2(t) &\sim \mathrm{GP}(0, C_\sigma^{(d)}(t, t')), \\ \log \omega_d(t) &\sim \mathrm{GP}(0, C_\omega^{(d)}(t, t')), \\ \log \ell_d(t) &\sim \mathrm{GP}(0, C_\ell^{(d)}(t, t')). \end{aligned}$$

In this setting, the positive variance, frequency and lengthscale of the mixture components are allowed to vary smoothly over time.

By contrast the GTF-NMF model, Eqs. (4.1)-(4.3), keeps the lengthscales and frequencies fixed but introduces a similar nonstationary prior over the variances / amplitudes. Therefore its hierarchical form can be written in a similar way to the nonstationary spectral mixture, i.e. with a GP hyper-prior $g_n(t) \sim \mathrm{GP}(0, C_\mathrm{g}^{(n)}(t, t'))$ for each component with an NMF-like positivity mapping $\alpha_d^2(t) = \sum_n W_{d,n} \, \phi(g_n(t))$, such that

$$s(t) \sim \mathrm{GP}\left(0, \sum_{d=1}^{D} \alpha_d(t)\alpha_d(t') \cos\big(\omega_d(t - t')\big) C_d\big(t, t' \,|\, \ell_d\big)\right), \tag{4.6a}$$

$$y_k = s(t_k) + \sigma_y \, \varepsilon_k. \tag{4.6b}$$

We use the notation $\alpha_d^2(t)$ to represent the variance here rather than $a_d^2(t)$ since these parameters are not identical: multiplying the kernel by a real value is not the same as multiplying realisations of the GP by that value. However, these parameters play a similar role in the model and have a similar interpretation.

This equivalence means that the inference methods laid out in section 4.4 and section 4.5 also apply to some nonstationary spectral mixture models, as do their formulation as stochastic differential equations (we leave the SDE formulation of nonstationary frequencies and lengthscales to future work).

4.3 SDE Form of the Nonstationary Model

Following the approach outlined in section 3.3, we write the spectral mixture GP component (which we now call $\mathbf{z}_d$ since it represents the subband carrier signal) in its SDE form using a product of cosine and Matérn kernels. For example the feedback matrix is constructed via the Kronecker sum as $\mathbf{F}_{\text{sm}} = \text{blkdiag}\left(\mathbf{F}_{\cos}^{(1)} \oplus \mathbf{F}_{\text{Mat}}^{(1)}, \dots, \mathbf{F}_{\cos}^{(D)} \oplus \mathbf{F}_{\text{Mat}}^{(D)}\right)$, and the other model components ($\mathbf{L}_{\text{sm}}$, $\mathbf{Q}_{c,\text{sm}}$, $\mathbf{P}_{\infty,\text{sm}}$, $\mathbf{H}_{\text{sm}}$) are constructed via the Kronecker product of the cosine and Matérn parts, again in a block-diagonal structure.

We now append to the model the GP prior over the amplitudes, constructing a new state space model, Eq. (2.16), whose state vector $\tilde{\mathbf{f}}(t)$ is the concatenation of the subband state $\tilde{\mathbf{z}}(t)$ and the amplitude state $\tilde{\mathbf{g}}(t)$:

$$\tilde{\mathbf{f}}(t) = \begin{pmatrix} \tilde{\mathbf{z}}(t) \\ \tilde{\mathbf{g}}(t) \end{pmatrix} \in \mathbb{R}^M. \tag{4.7}$$

If we notate the amplitude prior model matrices $\mathbf{F}_{\text{amp}}$, $\mathbf{L}_{\text{amp}}$, etc., then the full model matrices are again constructed by stacking the components along the diagonal,:

$$\begin{aligned} \mathbf{F} &= \begin{pmatrix} \mathbf{F}_{\text{sm}} & 0 \\ 0 & \mathbf{F}_{\text{amp}} \end{pmatrix}, & \mathbf{L} &= \begin{pmatrix} \mathbf{L}_{\text{sm}} & 0 \\ 0 & \mathbf{L}_{\text{amp}} \end{pmatrix}, \\ \mathbf{Q}_c &= \begin{pmatrix} \mathbf{Q}_{c,\text{sm}} & 0 \\ 0 & \mathbf{Q}_{c,\text{amp}} \end{pmatrix}, & \mathbf{P}_\infty &= \begin{pmatrix} \mathbf{P}_{\infty,\text{sm}} & 0 \\ 0 & \mathbf{P}_{\infty,\text{amp}} \end{pmatrix}. \end{aligned} \tag{4.8}$$

However, since the likelihood model is now a nonlinear mixture of $\mathbf{z}_d$ and $\mathbf{g}_n$, we replace the linear observation matrix $\mathbf{H}$ with a nonlinear operator $\mathcal{H}$ that takes the carrier and amplitude states and outputs the sum of the subbands: $\sum_d s_{d,k} = \mathcal{H}(\tilde{\mathbf{f}}(t_k))$. In the GTF-NMF model, Eqs. (4.1)-(4.3), this is given by:

$$\begin{aligned} \mathcal{H}(\tilde{\mathbf{f}}(t_k)) &= \sum_{d=1}^{D} a_{d,k} z_{d,k} \\ &= \boldsymbol{\mu}_{1:D}^\top \mathbf{W} \phi\big(\boldsymbol{\mu}_{D+1:D+N}\big), \qquad \text{for } \boldsymbol{\mu} = \mathbf{H}\tilde{\mathbf{f}}(t_k) \in \mathbb{R}^{D+N\times 1} \end{aligned} \tag{4.9}$$

with NMF weights $\mathbf{W} \in \mathbb{R}^{D\times N}$ and softplus function $\phi(\cdot)$. The matrix $\mathbf{H}$ selects and stores the real-valued first-order terms corresponding to $\mathbf{z}$ and $\mathbf{g}$ from the state vector $\tilde{\mathbf{f}}$, i.e. $\boldsymbol{\mu} = \mathbf{H}\tilde{\mathbf{f}} = \begin{pmatrix} \mathbf{z}_{1:D} & \mathbf{g}_{1:N} \end{pmatrix}^\top$. In terms of the initial model we can interpret these likelihood components as the amplitudes $\mathbf{a}_{1:D} = \mathbf{W}\phi\big(\boldsymbol{\mu}_{D+1:D+N}\big)$ and the carriers $\mathbf{z}_{1:D} = \boldsymbol{\mu}_{1:D}$. Now the continuous-discrete SDE is written

$$\frac{\mathrm{d}\tilde{\mathbf{f}}(t)}{\mathrm{d}t} = \mathbf{F}\tilde{\mathbf{f}}(t) + \mathbf{L}\mathbf{w}(t), \tag{4.10a}$$

$$y_k = \mathcal{H}(\tilde{\mathbf{f}}(t_k)) + \sigma_y \varepsilon_k. \tag{4.10b}$$

An example state space model The SDE formulation is made clearer by considering an example model in which the carrier priors have a Matérn-3/2 kernel, $C_z^{(d)}(t,t') = C_{\mathrm{Mat3/2}}(t,t')$, and the amplitude priors have a Matérn-5/2 kernel, $C_g^{(n)}(t,t') = C_{\mathrm{Mat5/2}}(t,t')$.

In this case, using the dot notation for the time derivative $\dot{z}_d(t)$, the spectral mixture model for the carriers has a second-order complex state space, $\tilde{\mathbf{z}}_d(t) = \big(\mathrm{Re}[z_d(t)] \quad \mathrm{Im}[z_d(t)] \quad \mathrm{Re}[\dot{z}_d(t)] \quad \mathrm{Im}[\dot{z}_d(t)]\big)^\top$. For the amplitude prior, the Matérn-5/2 admits a third-order real-valued state space, $\tilde{\mathbf{g}}_n(t) = \big(g_n(t) \quad \dot{g}_n(t) \quad \ddot{g}_n(t)\big)^\top$.

We must now write down the appropriate observation model to enable Eq. (4.9) to represent a product of the NMF-weighted amplitudes and the carriers. Letting $\mathbf{H}_4 = \begin{pmatrix}1 & 0 & 0 & 0\end{pmatrix}$ and $\mathbf{H}_3 = \begin{pmatrix}1 & 0 & 0\end{pmatrix}$, we stack the model components as follows:

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}_4^{(1)} & & & & & \\ & \ddots & & & & \\ & & \mathbf{H}_4^{(D)} & & & \\ & & & \mathbf{H}_3^{(1)} & & \\ & & & & \ddots & \\ & & & & & \mathbf{H}_3^{(N)} \end{pmatrix}, \quad \tilde{\mathbf{f}}(t) = \begin{pmatrix} \tilde{\mathbf{z}}_1(t) \\ \vdots \\ \tilde{\mathbf{z}}_D(t) \\ \tilde{\mathbf{g}}_1(t) \\ \vdots \\ \tilde{\mathbf{g}}_N(t) \end{pmatrix}, \tag{4.11}$$

which gives us the desired result.

4.4 Linearisation-Based Inference

The inference methods laid out in the remainder of this chapter generally act on time discretised versions of the above model, Eq. (4.10), and hence it is useful to define the notation $\tilde{\mathbf{f}}_k = \tilde{\mathbf{f}}(t_k)$ to represent the discrete state vector.

Given the model in Eq. (4.10), we now attempt again to perform inference via Kalman filtering. The Kalman filter prediction step (see

Eqs. (2.21)) can proceed as usual given the linear Gaussian form of the prior component, Eq. (4.10a). However now that our observation model, Eq. (4.10b), is nonlinear (and non-Gaussian since Gaussianity is not preserved through nonlinear operations) we must modify the update step to account for this nonlinearity. Typically this is done by approximating the state distribution $p(\tilde{\mathbf{f}}_{1:T} \,|\, \mathbf{y})$ via local Gaussian approximations to the time-marginals $q(\tilde{\mathbf{f}}_k \,|\, \mathbf{y}) \simeq \mathrm{N}(\tilde{\mathbf{f}}_k \,|\, \mathbf{m}_k, \mathbf{P}_k)$.

Perhaps the most widely used technique for calculating these local Gaussian approximations is the *extended Kalman filter* (EKF, Jazwinski, 1970; Bar-Shalom et al., 2001). The EKF, together with the backward-pass known as the extended Rauch–Tung–Striebel smoother, takes first-order Taylor series linearisations of the nonlinear components, replacing the feedback and measurement projections in the prediction and update steps with their Jacobian matrices evaluated at the mean: $\mathbf{F}_{\tilde{\mathbf{f}}}(\mathbf{m}_{k-1})$ and $\mathbf{H}_{\tilde{\mathbf{f}}}(\mathbf{m}_k^-)$.

For GPs, a related local linearisation scheme is known as the Laplace approximation, where the approximation is improved iteratively by mode-seeking. In signal processing, iterative versions of the EKF are known as iterated filters, where the iteration is typically in the inner update loop (local iterated EKF, Jazwinski, 1970; Maybeck, 1982). Outer-loop variants which—similar to the GP Laplace method—seek a global approximation are known as the global iterated EKF (Zhang, 1997).

To apply this to the GTF-NMF model we must calculate the Jacobian of the measurement function $\mathcal{H}(\tilde{\mathbf{f}})$, Eq. (4.9), which we will denote $\mathbf{H}_{\tilde{\mathbf{f}}} \in \mathbb{R}^M$ and is given via the chain rule as

$$\mathbf{H}_{\tilde{\mathbf{f}}} = \mathbf{H}^\top \begin{pmatrix} \mathbf{W}\phi(\boldsymbol{\mu}_{D+1:D+N}) \\ \mathrm{diag}\left(\boldsymbol{\mu}_{1:D}^\top \mathbf{W}\right) \dot{\phi}(\boldsymbol{\mu}_{D+1:D+N}) \end{pmatrix} \tag{4.12}$$

where $\dot{\phi}(x) = \mathrm{e}^x/(\mathrm{e}^x + 1)$ is the derivative of the softplus, i.e. the sigmoid function. The elements inside the parentheses in Eq. (4.12) correspond to the state dimensions of interest in the likelihood, $(\,\cdot\,) \in \mathbb{R}^{D+N\times 1}$. The EKF now essentially runs the standard Kalman filter steps, Eq. (2.22), using this updated (linearised) measurement matrix.

In Algorithm 1 we present an iterated (outer-loop) EKF scheme for Laplace-like approximate inference. The local linearisation is still performed according to Eq. (4.12), but once we have run the smoother

Algorithm 1 Linearisation-based inference (Laplace approximation scheme) formulated as a global iterated extended Kalman filter.

```
Input: $\{t_k, y_k\}_{k=1}^{T}$, $\mathbf{A}$, $\mathbf{Q}$, $\mathbf{P}_0$            data and discretised state space model
       $\mathcal{H}(\tilde{\mathbf{f}})$, $\mathbf{H}$, $\mathbf{H}_{\tilde{\mathbf{f}}}(\tilde{\mathbf{f}})$            measurement model and Jacobian
$\mathbf{m}_0 \leftarrow \mathbf{0}$            init state mean
while not converged do            iterated EKF loop
  for $k = 1$ to $T$ do            forward pass
    if $k == 1$ then
      $\mathbf{P}_k \leftarrow \mathbf{P}_0$            init state covariance
    else
      $\mathbf{m}_k \leftarrow \mathbf{A}\,\mathbf{m}_{k-1}$; $\mathbf{P}_k \leftarrow \mathbf{A}\,\mathbf{P}_{k-1}\,\mathbf{A}^\top + \mathbf{Q}$            predict
    end if
    if has label $y_k$ then
      $v_k \leftarrow y_k - \mathcal{H}(\mathbf{m}_k)$; $S_k \leftarrow \mathbf{H}_{\tilde{\mathbf{f}}}(\mathbf{m}_k)\,\mathbf{P}_k\,\mathbf{H}_{\tilde{\mathbf{f}}}^\top(\mathbf{m}_k) + \sigma_y^2$            inn.
      $\mathbf{k}_k \leftarrow \mathbf{P}_k\,\mathbf{H}_{\tilde{\mathbf{f}}}^\top(\mathbf{m}_k)\,S_k^{-1}$            gain
      $\mathbf{m}_k \leftarrow \mathbf{m}_k + \mathbf{k}_k\,v_k$; $\mathbf{P}_k \leftarrow \mathbf{P}_k - \mathbf{k}_k\,S_k\,\mathbf{k}_k^\top$
    end if
  end for
  for $k = T-1$ to $1$ do            backward pass
    $\mathbf{G}_k \leftarrow \mathbf{P}_k\,\mathbf{A}^\top\,(\mathbf{A}\,\mathbf{P}_k\,\mathbf{A}^\top + \mathbf{Q})^{-1}$            gain
    $\mathbf{m}_k \leftarrow \mathbf{m}_k + \mathbf{G}_k\,(\mathbf{m}_{k+1} - \mathbf{A}\,\mathbf{m}_k)$
    $\mathbf{P}_k \leftarrow \mathbf{P}_k + \mathbf{G}_k\,(\mathbf{P}_{k+1} - \mathbf{A}\,\mathbf{P}_k\,\mathbf{A}^\top - \mathbf{Q})\,\mathbf{G}_k^\top$
  end for
end while
rows of $\mathbf{H}$ select states of interest, e.g. $\mathbf{h}_n^{\mathrm{g}}$ corresponds to row for $g_n$
Return: $\mathbb{E}[g_n(t_k)] = \mathbf{h}_n^{\mathrm{g}}\mathbf{m}_k$; $\mathbb{V}[g_n(t_k)] = \mathbf{h}_n^{\mathrm{g}}\mathbf{P}_k\mathbf{h}_n^{\mathrm{g}\top}$
        $\mathbb{E}[z_d(t_k)] = \mathbf{h}_d^{\mathrm{z}}\mathbf{m}_k$; $\mathbb{V}[z_d(t_k)] = \mathbf{h}_d^{\mathrm{z}}\mathbf{P}_k\mathbf{h}_d^{\mathrm{z}\top}$
        $\log p(\mathbf{y}\,|\,\boldsymbol{\theta}) \simeq -\sum_{k=1}^{T}\frac{1}{2}(\log 2\pi S_k + v_k^2/S_k)$
```

we use the current mean, $\mathbf{m}_1$, as the starting point for the next run of the EKF, repeating this process until convergence. We consider this algorithm as the baseline for our experiments in section 4.7. Its use as a baseline is motivated by its ubiquity in signal processing, as well as by recent work showing that the iterated EKF can outperform other inference methods such as EP for many tasks (Tronarp et al., 2018; García-Fernández et al., 2019).

4.5 Expectation Propagation for GTF-NMF

The inference methods laid out so far in this thesis, despite being scalable and efficient, are limited to systems that are well approximated by linear models and they are in general not capable of producing accurate inference in the presence of strong nonlinear dependencies such as in the model presented in Eq. (4.10).

Nickisch et al. (2018) combine classical methods with modern tools for approximate inference, e.g. variational Bayes and assumed density filtering (ADF), to enable handling of more complex models. We generalise this work by extending the ADF algorithm to expectation propagation and thus combining the best methods from the signal processing and machine learning communities.

Expectation propagation (EP) and power expectation propagation (PEP) are methods for approximating intractable probability distributions using tractable distributions from the exponential family. EP is a generalisation of ADF and works by minimising local Kullback-Leibler (KL) divergences in an iterative fashion. PEP can be seen as a further generalisation of EP that minimises local α-divergences rather than KL divergences (Minka, 2005).

Using PEP, we approximate the intractable likelihood terms as follows:

$$p(y_k \,|\, \mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k}) \approx q_k(\mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k}), \tag{4.13}$$

where each site approximation q_k is Gaussian. Specifically, we assume that q_k factorises across the latent components (as it would if they were linearly mixed rather than nonlinearly) and takes the form

$$q_k(\mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k}) = \prod_{d=1}^{D} \mathrm{N}(z_{d,k} \,|\, \nu^z_{d,k}, \tau^z_{d,k}) \prod_{n=1}^{N} \mathrm{N}(g_{n,k} \,|\, \nu^g_{n,k}, \tau^g_{n,k}), \tag{4.14}$$

where $\nu^z_{d,k}$ and $\tau^z_{d,k}$ are the precision-adjusted mean and precision, respectively, for $z_{d,k}$ etc. This choice leads to a joint Gaussian posterior approximation. Rather than simply matching moments of the two distributions in Eq. (4.13), the EP algorithm iteratively refines the posterior approximation by updating each site approximation q_k in the context of the so-called *cavity distribution* q_{-k}. The cavity distribution for the k^{th} observation is defined by removing the contribution of the k^{th} site approximation from the posterior approximation $q(\mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k} \,|\, \mathbf{y})$. That is,

$$q_{-k}(\mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k}) \propto \frac{q(\mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k} \,|\, \mathbf{y})}{q_k(\mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k})^{\eta}} \tag{4.15}$$

for $\eta \in (0, 1]$, where $\eta = 1$ corresponds to regular EP and $\eta < 1$ to PEP.

In the Kalman filtering and smoothing paradigm we are interested in, the smoothing distribution is used for the marginal posterior approximation q. Crucially, on the first pass through the data, the cavity distributions can be approximated by the Kalman filter predictions $q(\mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k} \,|\, \mathbf{y}_{1:k-1})$ which can be interpreted as the prior over the latents conditioned on the past data. On subsequent EP iterations, the cavities can be calculated by removing the site approximations from the smoothing distribution as usual (see Algorithm 2).

The k^{th} site approximation q_k is then updated by minimising the KL-divergence between the *tilted distribution*,

$$\hat{p}_k = \frac{1}{Z_k} p(y_k \,|\, \mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k})^\eta q_{-k}(\mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k}), \tag{4.16}$$

i.e. the true Bayesian local update, and the PEP approximation $q_k(\mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k})^\eta q_{-k}(\mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k})$, i.e. the estimate given by rearranging Eq. (4.15), to obtain our new posterior approximation:

$$q_k^* \left(\mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k} \,|\, \mathbf{y}\right) = \arg\min_{q_k} \mathrm{D}_{\mathrm{KL}} \left[\hat{p}_k \,\|\, q_k^\eta q_{-k}\right]. \tag{4.17}$$

The normalisation constant Z_k is given by

$$Z_k = \mathbb{E}_{q_{-k}} \left[p(y_k \,|\, \mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k})^\eta\right]. \tag{4.18}$$

Minimising the KL divergence between Gaussians is equivalent to matching their first two moments, hence algorithmically the EP updates correspond to a moment matching procedure. The moments of the tilted distribution can be obtained from the first two partial derivatives of $\log Z_k$ with respect to two sets of cavity mean parameters $\{\mu^g_{n,-k}\}_{n=1}^N$ and $\{\mu^z_{d,-k}\}_{d=1}^D$:

$$\begin{aligned} Z_k = \quad & \mathbb{E}_{q_{-k}} \left[p(y_k \,|\, \mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k})^\eta\right] \\ = \quad & \int \dots \int \mathrm{N}\Big(y_k | \sum_d \sum_n z_{d,k} W_{d,n} \phi(g_{n,k}), \sigma_y^2\Big)^\eta \\ & \times \prod_d \mathrm{N}\big(z_{d,k} \,|\, \mu^z_{d,-k}, \zeta^z_{d,-k}\big) \\ & \times \prod_n \mathrm{N}\big(g_{n,k} \,|\, \mu^g_{n,-k}, \zeta^g_{n,-k}\big) \, \mathrm{d}z_{1,k} \dots \mathrm{d}z_{D,k} \, \mathrm{d}g_{1,k} \dots \mathrm{d}g_{N,k} \end{aligned} \tag{4.19}$$

$$
\begin{aligned}
= \quad & \mathrm{const}_\eta \int \ldots \int \mathrm{N}\Big(y_k \,|\, \sum_d \sum_n \mu^z_{d,-k} W_{d,n} \phi(g_{n,k}), \\
& \sigma_y^2 + \sum_d \sum_n \zeta^z_{d,-k} W^2_{d,n} \phi(g_{n,k})^2\Big) \\
& \times \prod_n \mathrm{N}\big(g_{n,k} \,|\, \mu^g_{n,-k}, \zeta^g_{n,-k}\big)\, \mathrm{d}g_{1,k} \ldots \mathrm{d}g_{N,k}
\end{aligned}
$$

for $\mathrm{const}_\eta = (2\pi\sigma_y^2)^{1/2(1-\eta)}\eta^{-1/2}$ and where we have used the marginalisation properties of the Gaussian distribution to obtain the last line. Setting

$$
m_y = \sum_d \sum_n \mu^z_{d,-k} W_{d,n} \phi(g_{n,k})
$$

and

$$
v_y = \sigma_y^2 + \sum_d \sum_n \zeta^z_{d,-k} W^2_{d,n} \phi(g_{n,k})^2
$$

and differentiating w.r.t. μ^z, μ^g, we get

$$
\begin{aligned}
\frac{\mathrm{d}Z_k}{\mathrm{d}\mu^z_{d,-k}} = \quad & \mathrm{const}_\eta \int \ldots \int \mathrm{N}\big(y_k \,|\, m_y, v_y\big) \\
& \times \sum_n W_{d,n}\phi(g_{n,k}) \frac{y - m_y}{v_y} \\
& \times \prod_n \mathrm{N}\big(g_{n,k} | \mu^g_{n,-k}, \zeta^g_{n,-k}\big)\, \mathrm{d}g_{1,k} \ldots \mathrm{d}g_{N,k},
\end{aligned} \tag{4.20}
$$

$$
\begin{aligned}
\frac{\mathrm{d}Z_k}{\mathrm{d}\mu^g_{n,-k}} = \quad & \mathrm{const}_\eta \int \ldots \int \mathrm{N}\big(y_k \,|\, m_y, v_y\big) \frac{g_{n,k} - \mu^g_{n,-k}}{\zeta^g_{n,-k}} \\
& \times \prod_n \mathrm{N}\big(g_{n,k} \,|\, \mu^g_{n,-k}, \zeta^g_{n,-k}\big)\, \mathrm{d}g_{1,k} \ldots \mathrm{d}g_{N,k},
\end{aligned} \tag{4.21}
$$

$$
\begin{aligned}
\frac{\mathrm{d}^2 Z_k}{\mathrm{d}{\mu^z_{d,-k}}^2} = \quad & \mathrm{const}_\eta \int \ldots \int \mathrm{N}\big(y_k \,|\, m_y, v_y\big) \\
& \times \sum_n (W_{d,n}\phi(g_{n,k}))^2 \left[\left(\frac{y - m_y}{v_y}\right)^2 - \frac{1}{v_y}\right] \\
& \times \prod_n \mathrm{N}\big(g_{n,k} \,|\, \mu^g_{n,-k}, \zeta^g_{n,-k}\big)\, \mathrm{d}g_{1,k} \ldots \mathrm{d}g_{N,k},
\end{aligned} \tag{4.22}
$$

$$\begin{aligned}\frac{\mathrm{d}^2 Z_k}{\mathrm{d}{\mu^g_{n,-k}}^2} = \;& \mathrm{const}_\eta \int \dots \int \mathrm{N}\big(y_k \,|\, m_y, v_y\big) \\ & \times \left[\left(\frac{g_{n,k} - \mu^g_{n,-k}}{\zeta^g_{n,-k}} \right)^2 - \frac{1}{\zeta^g_{n,-k}} \right] \\ & \times \prod_n \mathrm{N}\big(g_{n,k} \,|\; \mu^g_{n,-k}, \zeta^g_{n,-k}\big)\, \mathrm{d}g_{1,k} \dots \mathrm{d}g_{N,k}.\end{aligned} \tag{4.23}$$

We can see from the above that all the required integrals are N-dimensional, where N is the number of NMF components. The partial derivatives of $\log Z_k$ can be obtained from the equations above using the chain rule:

$$\begin{aligned}\frac{\mathrm{dlog}\, Z_k}{\mathrm{d}\mu^z_{d,-k}} &= \frac{1}{Z_k} \frac{\mathrm{d}Z_k}{\mathrm{d}\mu^z_{d,-k}}, \\ \frac{\mathrm{dlog}\, Z_k}{\mathrm{d}\mu^g_{n,-k}} &= \frac{1}{Z_k} \frac{\mathrm{d}Z_k}{\mathrm{d}\mu^g_{n,-k}}, \\ \frac{\mathrm{d}^2\mathrm{log}\, Z_k}{\mathrm{d}{\mu^z_{d,-k}}^2} &= -\frac{1}{Z_k^2} \left(\frac{\mathrm{d}Z_k}{\mathrm{d}\mu^z_{d,-k}} \right)^2 + \frac{1}{Z_k} \frac{\mathrm{d}^2 Z_k}{\mathrm{d}{\mu^z_{d,-k}}^2}, \\ \frac{\mathrm{d}^2\mathrm{log}\, Z_k}{\mathrm{d}{\mu^g_{n,-k}}^2} &= -\frac{1}{Z_k^2} \left(\frac{\mathrm{d}Z_k}{\mathrm{d}\mu^g_{n,-k}} \right)^2 + \frac{1}{Z_k} \frac{\mathrm{d}^2 Z_k}{\mathrm{d}{\mu^g_{n,-k}}^2}.\end{aligned} \tag{4.24}$$

We use these derivatives of the log-partition function to update the site parameters in Eq. (4.14), whilst also converting them back to the precision-adjusted (natural) parameter space, via the following mapping (Seeger, 2005): letting $b_{d,k} = \frac{\mathrm{dlog}\, Z_k}{\mathrm{d}\mu^z_{d,-k}}$ and $c_{d,k} = \frac{\mathrm{d}^2\mathrm{log}\, Z_k}{\mathrm{d}{\mu^z_{d,-k}}^2}$ and for damping parameter ρ,

$$\tau^z_{d,k} = (1-\rho)\tau^z_{d,k} + \frac{\rho}{\eta} \left(\frac{-c_{d,k}}{1 + \zeta_{d,-k} c_{d,k}} \right), \tag{4.25a}$$

$$\nu^z_{d,k} = (1-\rho)\nu^z_{d,k} + \frac{\rho}{\eta} \left(\frac{b_{d,k} - \mu_{d,-k} c_{d,k}}{1 + \zeta_{d,-k} c_{d,k}} \right). \tag{4.25b}$$

Mapping to the natural parameter space in this way makes the updates in the EP algorithm more straightforward (see Algorithm 2). The updates for $\tau^g_{n,k}$ and $\nu^g_{n,k}$ are carried out similarly using the derivatives with respect to $\mu^g_{n,-k}$.

We numerically approximate the N-dimensional integrals required for moment matching with 9th-order sigma-point methods (McNamee and

Stenger, 1967; Kokkala et al., 2016). However, the number of sigma-points required in this 9th-order approximation scales poorly with the number of NMF components, $\frac{1}{2}(2N^4 - 4N^3 + 22N^2 - 8N + 3)$, which slows down inference for large N. Lower-order approximations, e.g. 5th- and 7th-order sigma point methods, are sufficient in many cases and scale much more efficiently, however in the experiments laid out in section 4.7 we found that the 9th-order approach was required to obtain consistent results.

The proposed algorithm is prone to convergence issues due to the complexity of the data, ambiguities / non-identifiability in the model, and the nonlinearity in the likelihood. To prevent EP from oscillating we use *damped* updates for the site parameters (Minka and Lafferty, 2002). That is, the site parameters are updated as a convex combination of the current parameter values and the new parameters values, i.e. the first and second terms in Eq. (4.25). Given the large amount of damping required, we generally had to run EP for 20 iterations to reach convergence, more than the 5-10 that is often reported for simpler models.

Standard EP scales cubicly in the number of observations. However, by using the RTS smoother, Eq. (2.23), to approximate the marginal posterior distributions $q(\mathbf{z}_{1:D,k}, \mathbf{g}_{1:N,k} \,|\, \mathbf{y})$ in Eq. (4.15), we can reduce the complexity of the algorithm to be linear in the number of observations. The EP algorithm is summarised in Algorithm 2.

Hyperparameter Tuning Model learning is difficult in this setting due to the highly correlated nature of the kernel hyperparameters and the non-identifiability of the NMF mapping (due to the fact that there are two sources of amplitude variation in the model: the envelopes themselves, $\mathbf{a}_d$, but also the natural variation in the quasi-periodic subbands, $\mathbf{z}_d$). We initialise the parameters via frequency domain fitting with the standard probabilistic TF model, as outlined in chapter 3, which is fast and gives an accurate estimate of the subband frequencies and lengthscales. We initialise the NMF weights using standard NMF applied to a spectrogram calculated with our subband model. Further tuning is then carried out by direct optimisation of the (log) marginal likelihood, $\log p(\mathbf{y} \,|\, \boldsymbol{\theta})$, which is calculated during Kalman smoothing as shown in Algorithm 2.

Algorithm 2 Expectation propagation using Kalman smoothing

```
Input: $\{t_k, y_k\}_{k=1}^{T}$                                              training inputs and targets
       $\mathbf{A}$, $\mathbf{Q}$, $\mathbf{H}$, $\mathcal{H}(\tilde{\mathbf{f}})$, $\mathbf{P}_0$                           discretised state space model
$\boldsymbol{\tau} \leftarrow \mathbf{0}$, $\boldsymbol{\nu} \leftarrow \mathbf{0}$                                   likelihood eff. precision and location
while EP not converged do                                                    EP loop
  for $k = 1$ to $T$ do                                                      forward pass
    if $k == 1$ then
      $\mathbf{m}_k \leftarrow \mathbf{0}$;  $\mathbf{P}_k \leftarrow \mathbf{P}_0$                                          init
    else
      $\mathbf{m}_k \leftarrow \mathbf{A}\mathbf{m}_{k-1}$; $\mathbf{P}_k \leftarrow \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^\top + \mathbf{Q}$                   predict
    end if
    if has label $y_k$ then
      $\boldsymbol{\mu} \leftarrow \mathbf{H}\mathbf{m}_k$;  $\mathbf{U} \leftarrow \mathbf{P}_k\mathbf{H}^\top$;  $\boldsymbol{\sigma}^2 \leftarrow \mathrm{diag}\,(\mathbf{H}\mathbf{U})$              latent
      if first EP iteration then
        $\boldsymbol{\tau}_{-k} \leftarrow \boldsymbol{\sigma}^2$;  $\boldsymbol{\nu}_{-k} \leftarrow \boldsymbol{\mu}$                                       cavity
        set $(\boldsymbol{\nu}_k, \boldsymbol{\tau}_k)$ to minimise the KL div. in Eq. (4.17) by calculating $Z_k$ and its gradients via Eqs. (4.19)-(4.25)
      end if
      $\mathbf{c}_k \leftarrow \boldsymbol{\mu} \oslash \boldsymbol{\tau}_k - \boldsymbol{\nu}_k$
      $\mathbf{K}_k \leftarrow \mathbf{U}\ (\boldsymbol{\sigma}^2 + \mathbf{1} \oslash \boldsymbol{\tau}_k)^{-1}$            (multiplication is column-wise)
      $\mathbf{P}_k \leftarrow \mathbf{P}_k - \mathbf{K}_k\mathbf{U}^\top$                                        variance
      $\mathbf{m}_k \leftarrow \mathbf{m}_k + \mathbf{K}_k\mathbf{c}_k$                                         mean
    end if
  end for
  for $k = T - 1$ to $1$ do                                                   backward pass
    $\mathbf{G}_k \leftarrow \mathbf{P}_k\,\mathbf{A}^\top\,(\mathbf{A}\,\mathbf{P}_k\,\mathbf{A}^\top + \mathbf{Q})^{-1}$                                 gain
    $\mathbf{m}_k \leftarrow \mathbf{m}_k + \mathbf{G}_k\,(\mathbf{m}_{k+1} - \mathbf{A}\,\mathbf{m}_k)$
    $\mathbf{P}_k \leftarrow \mathbf{P}_k + \mathbf{G}_k\,(\mathbf{P}_{k+1} - \mathbf{A}\,\mathbf{P}_k\,\mathbf{A}^\top - \mathbf{Q})\,\mathbf{G}_k^\top$
    $\boldsymbol{\mu} \leftarrow \mathbf{H}\mathbf{m}_k$;  $\boldsymbol{\sigma}^2 \leftarrow \mathrm{diag}\left(\mathbf{H}\mathbf{P}_k\mathbf{H}^\top\right)$                                latent
    $\boldsymbol{\tau}_{-k} \leftarrow \mathbf{1} \oslash \boldsymbol{\sigma}^2 - \eta\boldsymbol{\tau}_k$;  $\boldsymbol{\nu}_{-k} \leftarrow \boldsymbol{\mu} \oslash \boldsymbol{\sigma}^2 - \eta\boldsymbol{\nu}_k$                   cavity
    set $(\boldsymbol{\nu}_k, \boldsymbol{\tau}_k)$ to minimise the KL div. in Eq. (4.17) by calculating $Z_k$ and its gradients via Eqs. (4.19)-(4.25)
  end for
end while
rows of $\mathbf{H}$ select states of interest, e.g. $\mathbf{h}_n^{\mathrm{g}}$ corresponds to row for $g_n$
Return: $\mathbb{E}[g_n(t_k)] = \mathbf{h}_n^{\mathrm{g}}\mathbf{m}_k$; $\mathbb{V}[g_n(t_k)] = \mathbf{h}_n^{\mathrm{g}}\mathbf{P}_k\mathbf{h}_n^{\mathrm{g}\top}$
        $\mathbb{E}[z_d(t_k)] = \mathbf{h}_d^{\mathrm{z}}\mathbf{m}_k$; $\mathbb{V}[z_d(t_k)] = \mathbf{h}_d^{\mathrm{z}}\mathbf{P}_k\mathbf{h}_d^{\mathrm{z}\top}$
        $\log p(\mathbf{y} \,|\, \boldsymbol{\theta}) \simeq \sum_k \log Z_k$
```

Notation: $\mathbf{a} \circ \mathbf{b}$ and $\mathbf{a} \oslash \mathbf{b}$ denote the element-wise multiplication and element-wise divison of the vectors $\mathbf{a}$ and $\mathbf{b}$, respectively.

4.6 Infinite-Horizon Gaussian Processes

The inference method laid out above has linear time complexity, $\mathcal{O}(TM^3)$ (with $M \ll T$), with respect to the number of data points T, and state dimensionality M. The memory scaling is $\mathcal{O}(TM^2)$ due to the need for storing the state covariances at every time step. However, in the case of audio data T can be tens or hundreds of thousands even for short audio segments. This is mainly problematic with regards to the required memory (M typically in the range of 100–1000). For example, for $M = 100$, the required memory is in the range of 1.2 Gb per second of data.

To mitigate the memory bottleneck, we use the infinite-horizon GP (IHGP) framework proposed by Solin et al. (2018), where the GP is approximated by finding an associated posterior steady state of the filter for each of the $D+N$ latent functions. This way the propagation of the covariance terms in Algorithm 2 can be simplified, leading to a computational time scaling of $\mathcal{O}(TM^2)$ and memory scaling $\mathcal{O}(TM)$. Solin et al. (2018) derived their method to work with ADF, but our EP formulation directly lends itself to the approach by using the cavity parameters for updating the likelihood variance terms. With these changes, the required memory drops by orders of magnitude to 12.2 Mb per second of data.

The main idea is to drop the dependence on time of the state covariance $\mathbf{P}_k$, replacing it with a dependence only on the likelihood variance of the individual $i = 1, \dots, D+N$ model components, which can be estimated as $\sigma^2_{i,k} = 1/\tau_{i,k}$, treating it as a function $\mathbf{P}^i(\sigma^2_{i,k})$. A steady state solution exists when the covariance doesn't change between time steps, hence we can calculate $\mathbf{P}^i(\sigma^2_{i,k})$ by writing down one full recursion of the Kalman filter prediction and update steps to get

$$\begin{aligned}\mathbf{P}^i(\sigma^2_{i,k}) =& \mathbf{A}\mathbf{P}^i(\sigma^2_{i,k})\mathbf{A}^\top \\ & - \mathbf{A}\mathbf{P}^i(\sigma^2_{i,k})\mathbf{h}_i^\top(\mathbf{h}_i\mathbf{P}^i(\sigma^2_{i,k})\mathbf{h}_i^\top + \boldsymbol{\sigma}^2_k)^{-1}\mathbf{h}_i\mathbf{P}^i(\sigma^2_{i,k})\mathbf{A}^\top + \mathbf{Q},\end{aligned} \tag{4.26}$$

where $\mathbf{h}_i$ is the row of $\mathbf{H}$ corresponding to the i^{th} latent component. Eq. (4.26) is the form of a discrete algebraic Riccati equation (DARE, see Lancaster and Rodman, 1995), which can be solved by the Schur method (Laub, 1979) in $\mathcal{O}(M^3)$. Instead of solving the DARE at every time step, an interpolation method is used in which, prior to inference, multiple DAREs are solved for a range of likelihood variance values and their associated steady state covariances $\mathbf{P}^i$ are stored in a look-up table.

This approach adds a computational overhead to the inference scheme, but now that the covariance does not depend on time, the Kalman filter equations can be simplified such that they no longer rely on matrix multiplications, but rather only on vector-matrix multiplications which leads to computational scaling of $\mathcal{O}(TM^2)$. Crucially, since we no longer need to store a large covariance matrix at every time step, we achieve the reduced memory requirement of $\mathcal{O}(TM)$.

4.7 Comparing Inference Schemes for GTF-NMF

In this section we compare the proposed inference methods, showing that fully iterated EP is absolutely necessary for inference in the GTF-NMF model, since the iterated EKF and single-sweep EP approaches fail to uncover the latent functions with sufficient accuracy and show inferior performance in signal processing applications. Our generative model is extremely flexible, and we demonstrate here how it can be applied to three different real world tasks (and one simulated task) with no adjustment of the model or algorithm: missing data synthesis, denoising and source separation. The GTF-NMF performs on a similar level to application specific algorithms (better in missing data imputation, worse in denoising), whilst being much more general.

For ease of comparison, in all the real-world experiments we set $D = 16$, $N = 3$ and tune the parameters via single-sweep EP (ADF), with $\eta = 0.75$ and damping of $\rho = 0.1$. The benefit of adapting the model to the signal is that even just 16 filters can be sufficient to describe the data, however computational challenges were also considered when choosing these settings. We use the learnt parameters to directly compare the different inference methods (with the exception of the simulated data experiment where we use the known parameters). We use the exponential and Matérn-5/2 kernels for κ_d and κ_g. The advantages of the infinite-horizon approach become clear when we consider the source separation problem, in which the mixture signal contains multiple sources (leading to a very high-dimensional state space $M = 123$), and is 6 seconds in duration ($T = 96{,}000$).

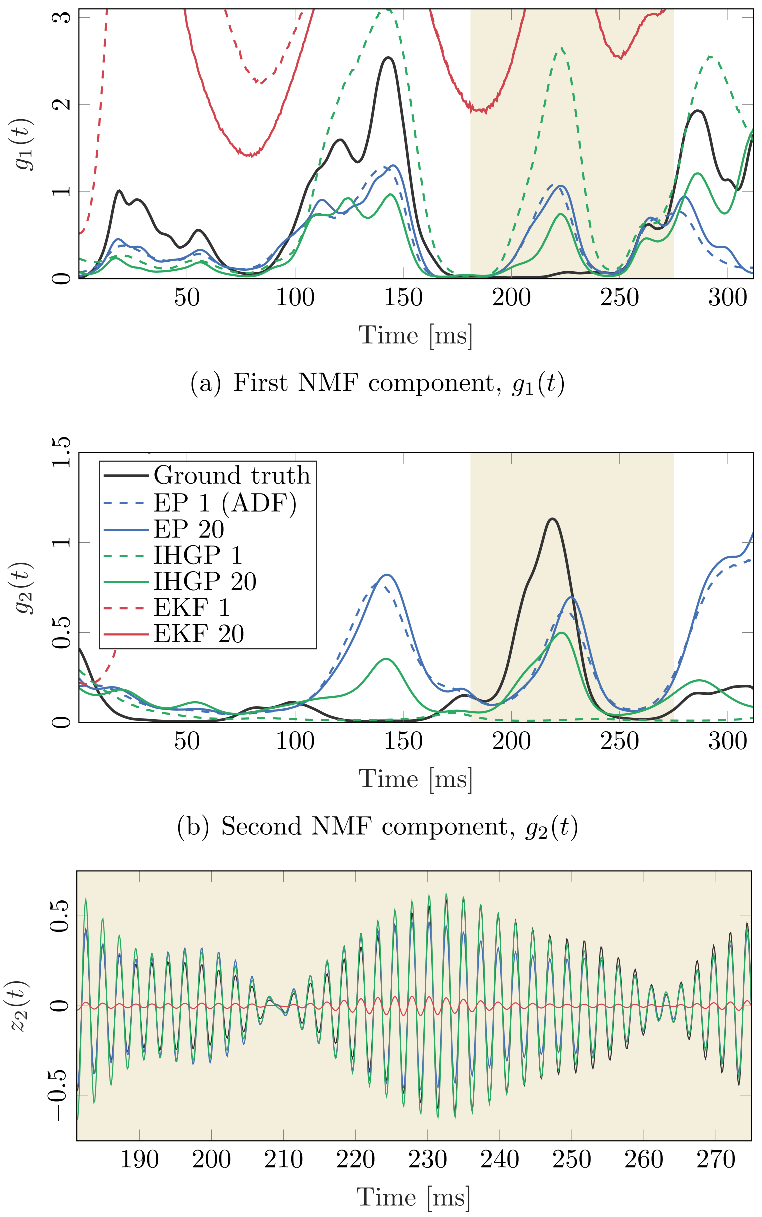
	EP1	EP20	IHGP1	IHGP20	EKF1	EKF20	MP
RMSE (sim.)	0.044	**0.003**	0.042	0.029	0.124	0.128	—
SNR (mis.)	7.494	**8.087**	4.520	4.591	3.716	3.735	5.232
RMSE (mis.)	0.590	**0.551**	0.720	0.716	0.746	0.743	0.761

Table 4.1: Performance measures for each inference scheme. *'sim.'* shows fit to observed data $\mathbf{y}$ in the simulated data experiment (likelihood noise variance is $\sigma_y^2 = 10^{-4}$). *'mis.'* shows mean missing data imputation results on a dataset of 10 musical instrument sounds, with segments of 20ms removed. Signal-to-noise ratio (in dB, larger is better) and root mean square error (smaller is better). Based on predictive mean. MP is the matching pursuit baseline.

Simulated Data Experiment We set $D = 5$, $N = 2$ and fix the hyperparameters by hand, before sampling from the generative model to create synthetic data. Figure 4.2 shows how each of the proposed inference methods estimates the hidden subband signals and NMF modulators. Uncovering the latents is a highly non-identifiable problem, especially due to the ambiguous nature of the model in which amplitude variation can occur due to variance in the subbands or the modulators. However, EP finds a much better match to the ground truth than EKF, and we see that iterating the IHGP method resolves part of the ambiguity. Table 4.1 shows how closely the approximate inference methods are able to fit the training data. Note that the likelihood noise variance is $\sigma_y^2 = 10^{-4}$, and hence we would hope the RMSE to be below $\sigma_y = 0.01$, a feat which only full EP manages.

Missing Data Imputation The generative model handles missing data synthesis naturally by treating the time steps where there are missing data as test locations and making predictions as usual. Table 4.1 shows the results of the prediction task on a dataset of 10 musical instrument recordings. Figure 4.3 shows an example segment from a recording of a bamboo flute. As a baseline for comparison we compare our methods to a well known matching pursuit algorithm (Adler et al., 2012), designed for denoising tasks such as de-clipping, de-clicking and interference removal. This baseline was outperformed by the iterated EP scheme, and exhibited results roughly in line with the IHGP approach.

Denoising Assuming a signal is corrupted by Gaussian noise of known variance, the GTF-NMF model can be adapted to a denoising task by



(a) First NMF component, $g_1(t)$

(b) Second NMF component, $g_2(t)$

(c) Short segment of one of the subband signals, $z_4(t)$

Figure 4.2: A simulated data experiment examining the ability of various inference methods to uncover the spectral components z_d (one example shown in the **bottom** plot) and NMF components g_n (**top two** plots) when the true parameters are known. Simulated ground truth is in **black**. Due to the ambiguity inherent in the model, (multiple sources of amplitude modulation), uncovering the latents is a difficult task. Standard EP and the IHGP methods far outperform EKF. "EP 1" relates to inference with one EP iteration (ADF). The iterated methods (dashed lines, each using 20 iterations) resolve the ambiguity better than the single sweep approach, except in the EKF case. Only the mean of the predictive distributions is shown.

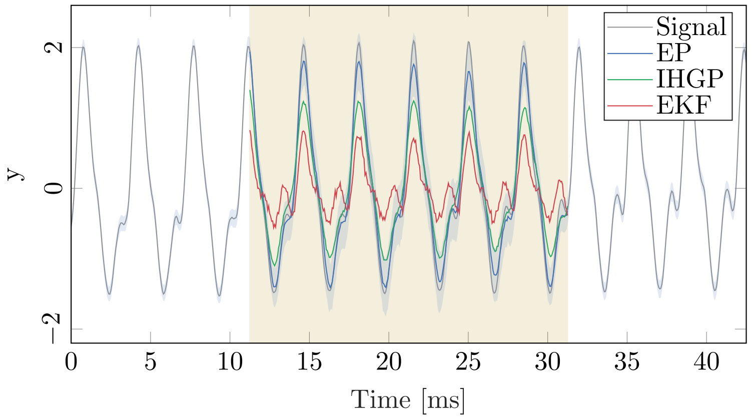


Figure 4.3: An example of missing data imputation with the GTF-NMF model for each inference method with 20 iterations. Grey signal is the ground truth, a recording of a bamboo flute. The yellow shaded region indicates where the data is missing. Blue shaded area is the 95% confidence region for the EP method.

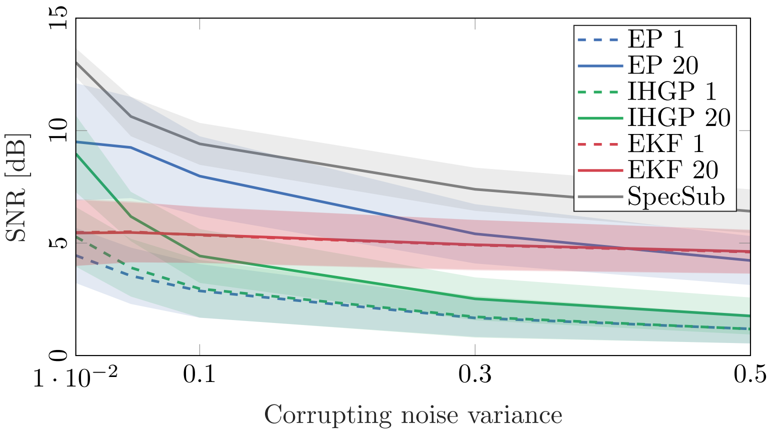


Figure 4.4: Denoising with various inference methods across five levels of corruption noise variance (0.01–0.5). y-axis is the signal-to-noise ratio of the recovered waveform. Mean values across 10 speech signals are shown. Shaded areas are standard error. SpecSub is the spectral subtraction baseline.

setting the measurement noise variance σ_y^2 to the appropriate level. Figure 4.4 shows the denoising results for the various inference methods for five different noise levels. Here we also compare against a spectral subtraction baseline algorithm (Ephraim and Malah, 1984) commonly used for denoising tasks. Figure 4.5 is an example of denoising a speech recording, where the clean signal is corrupted with $\sigma_y^2 = 0.3$. GP models are expected to deal with Gaussian noise well, however the approximate nature of inference in the GTF-NMF, as well as the potential for model

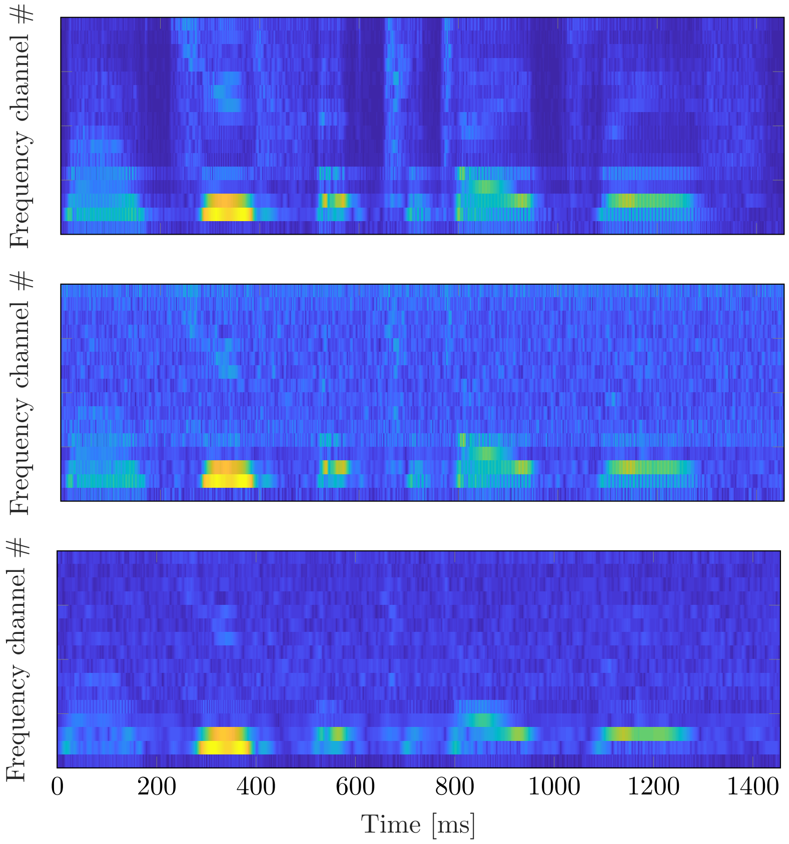


Figure 4.5: Spectrograms of a clean, corrupted, and reconstructed signal (from top to bottom) for audio denoising in the GTF-NMF model with inference via EP, applied to a speech signal.

misspecification when the optimisation procedure gets stuck in local minima, prevents it from outperforming the application-specific baseline.

Source Separation As a further demonstration, we follow the approach taken in Alvarado et al. (2019) by training the model on musical instrument notes (sources), and then attempting to uncover these sources when they are mixed via summation of their waveforms in a series of two-note chords. The only inference method capable of processing these series of notes is IHGP, due to the computation and memory requirements of stacking the sources in a state space model for 6 seconds of data (sampled at 16 kHz, $T = 96{,}000$, $M = 123$). Therefore we cannot compare performance on this task, but we show an example separation result in Figure 4.6.

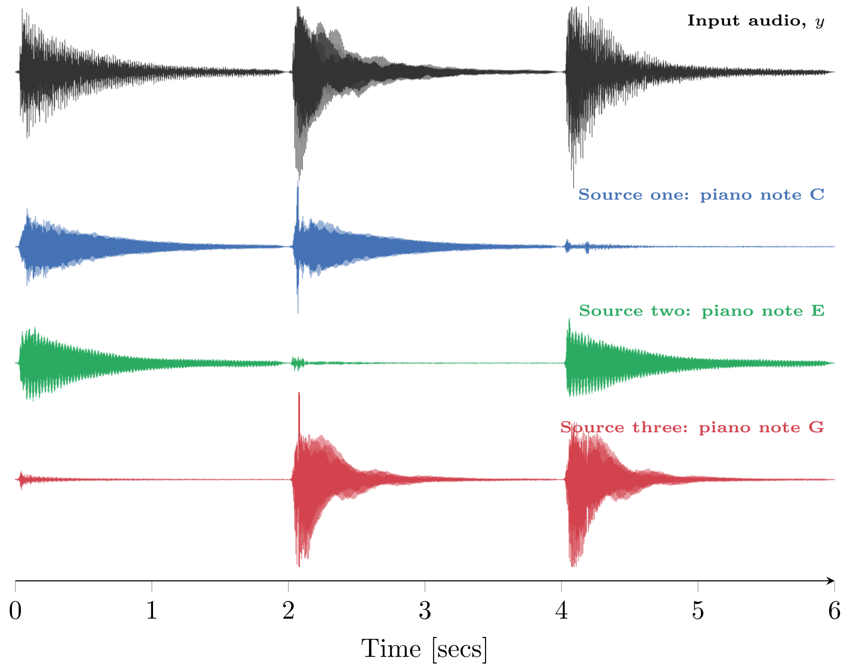


Figure 4.6: Source separation example using infinite-horizon Gaussian processes, showing three piano notes (sources) recovered from a mixture signal (top), where two notes are played at a time in the original recording.

4.8 Conclusion

We have constructed a novel scheme for inference in the Gaussian time-frequency NMF model based on power expectation propagation and leveraging infinite-horizon GPs, leading to an end-to-end probabilistic approach for audio modelling that goes beyond the disjoint analysis approach of Turner and Sahani (2014). By outlining how this model is similar to a nonstationary spectral mixture GP, we have further unified the theory connecting probabilistic machine learning and signal processing.

We demonstrated that our inference scheme consistently outperforms extended Kalman filtering. Recent work comparing the iterated EKF and EP approaches in a more general setting showed that for many models / datasets the EKF approach outperforms EP (Tronarp et al., 2018; García-Fernández et al., 2019). The significant benefits observed with EP in the GTF-NMF therefore suggest that those results are extremely model-dependent, and further investigation into the specific modelling scenarios that lend themselves to either approach is required.

Our results suggest that it is indeed necessary to go beyond classical signal processing techniques if we are to build more in-depth nonstationary methods for audio analysis, and that probabilistic modelling has much potential in this domain. By considering various real world tasks, we have shown the flexibility of such end-to-end generative models.

To extend this work, it is necessary to further reduce the inherent computational burden. One approach could be via the use of banded matrix operators which make GP models amenable to automatic differentiation (Durrande et al., 2019), significantly reducing the computational overhead involved in iteratively running the Kalman filter and RTS smoother. Speeding up the numerical integration step would have a significant effect on processing time, and sampling methods may be more accurate than our sigma-point approach.

Another avenue for investigation is an online algorithm that can handle data streaming and signals of longer duration. It has been shown that the IHGP method is suitable for data streaming since its steady state covariance reduces the risk of edge effects at the real-time frame boundaries (Solin et al., 2018). For such an approach to be practical, it may be necessary to improve the convergence properties of the EP algorithm, which currently requires many iterations and large damping.

The development of a more efficient and robust parameter learning scheme would also allow the GTF-NMF model to become more widely used: it is currently dependent on pre-processing via NMF and standard probabilistic TF analysis, and the fully probabilistic approach of maximising the marginal likelihood is very susceptible to getting stuck in local minima due to noisy signal spectra.

Chapter 5

Latent Force Models for Sound

The model considered in chapter 4 assumes the amplitude envelopes of the subbands of an audio signal, $\mathbf{a}_d$, come about as a linear sum of isotropic GPs projected through a positivity-enforcing link function. Whilst this link function provides some amount of anisotropic behaviour, NMF remains a very simplistic model for the amplitudes. Additionally, inferring the latent activations and the weights simultaneously is a highly ill-posed task, and it is common to leverage assumptions regarding smoothness, nonnegativity, sparsity etc. to aid in uncovering something representative of the "true" sound production mechanism. In this chapter we consider whether we can go beyond a scalar weighting of GPs, and incorporate *physical knowledge* about how audio signals behave into the prior.

One significant drawback of the NMF model considered so far is its symmetry in time. That is, the covariance of the latent process is the same forward in time as it is backwards in time. For audio signals, this is an unrealistic assumption since the attack and decay of a sound event can exhibit very different behaviour. Figure 5.1 shows the amplitude envelopes of a recording of a metal impact sound, which have been smoothed to clearly show the behaviour over time. This example, like many natural sounds, has a fast attack in which the envelopes are highly correlated and a slower decay in which the envelopes' decay rates vary and some envelopes modulate independently of the others.

In the next section we will motivate and implement a latent force model (LFM, see section 2.6) that takes into account these features.

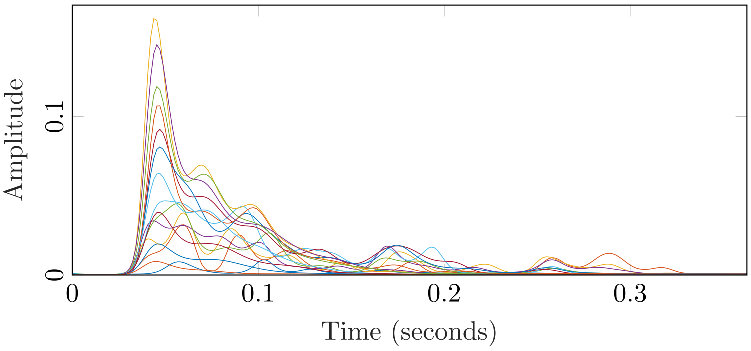


Figure 5.1: Amplitude envelopes of a metal impact sound.

In order to study the efficacy of a physical amplitude model prior, we isolate the envelopes by applying a fixed time-frequency analysis method and treating $\mathbf{a}_d$ as observations. Hence this chapter differs from the preceding two in that the models considered here are applied in the spectrogram domain rather than the waveform domain.

5.1 Learning Physical Parameters of Modal Synthesis

Physics-based approaches to sound synthesis vary from detailed numerical simulation of the sound production mechanism represented by differential equations (Trautmann and Rabenstein, 1999; Bensa et al., 2003), to standard digital filtering techniques informed by those same differential equations (Cook, 2002; Smith, 2010). 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 (Adrien and Ducasse, 1989; Cook, 1997). The forced mass-spring-damper corresponding to the d^{th} mode has coefficients relating to mass M_d, springiness (or stiffness) S_d and damping G_d:

$$M_d \frac{\mathrm{d}^2 x_d(t)}{\mathrm{d}t^2} + S_d \frac{\mathrm{d}x_d(t)}{\mathrm{d}t} + G_d x_d(t) = f(t), \tag{5.1}$$

where $f(t)$ 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,

$$x_d(t) = a_d(t) sin(2\pi\omega_d t + \psi_d), \tag{5.2}$$

for $d = 1, \ldots, D$, with time-varying amplitude $a_d(t)$, frequency ω_d and initial phase ψ_d, referred to as damped sinusoids, or damped oscillators. In traditional modal synthesis $f(t)$ is assumed to be an impulse, and we obtain the solution $a_d(t) = \alpha_d \mathrm{e}^{-\beta_d t}$ where α_d and β_d are the amplitude and damping of the mode respectively. If we allow $f(t)$ to be unconstrained, then no analytical solution for the amplitude exists. Here we will constrain $f(t)$ by placing a GP prior over its possible values.

Sinusoidal modelling Sinusoidal modelling (McAulay and Quatieri, 1986) 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 Eq. (5.2). 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. In the remainder of this section we use the *Spear* software (Klingbeil, 2005) to obtain the partials from an audio recording.

Modelling the amplitude data Our approach is to view sinusoidal amplitude data as the output of a series of digital filters representing the amplitudes $a_d(t)$ of the physical modes. This motivates the introduction of such filters (in ODE form) into the prior for a GP model looking to infer knowledge from audio recordings. In order for synthesis in this setting to be intuitively controllable, parameters must be physically meaningful and the learnt latent function $f(t)$ must also be interpretable in a physical sense.

We assume that the physical modes in Eq. (5.2) have fixed frequencies ω_d, and will limit any experiments in this chapter to sounds whose frequencies can reasonably be assumed not to vary significantly over time. We will further assume in this section that there exists a single forcing function driving the system, $f(t)$. Given this assumption, the problem becomes how to model $a_d(t)$.

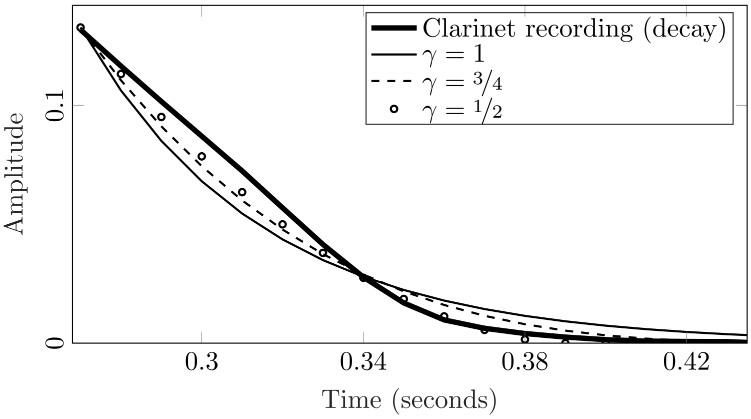


Figure 5.2: Comparison of amplitude model choice: $\gamma = 1$ represents the standard model for the amplitude of a sinusoid. Selecting $\gamma < 1$ alters the decay behaviour to more closely represent the real data obtained from the decay section of the second harmonic of a recording of a clarinet.

The analytical solution to this problem when $f(t)$ is an impulse is $x_d(t) = \alpha_d \mathrm{e}^{-\beta_d t}$. 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, Eq. (5.1). By doing so we obtain the model (letting $M_d = 1$ and replacing $x_d(t)$ with $a_d(t)$ to make it explicit that we are now modelling amplitudes),

$$\frac{\mathrm{d}a_d(t)}{\mathrm{d}t} + U_d a_d(t) = V_d f(t), \tag{5.3}$$

where $V_d = G_d/S_d$ and $V_d = 1/S_d$ are physically relevant parameters related to damping G_d and stiffness S_d of the system.

In practice, when observing real amplitude data (for which $f(t)$ will never truly be an impulse), we found that partials tend to decrease in a more linear fashion than can be described by equation Eq. (5.3). Therefore we propose an alternative model containing a parameter γ which alters the "linearity" of the decay of the signal,

$$\frac{\mathrm{d}a_d(t)}{\mathrm{d}t} + U_d a_d^{\gamma}(t) = V_d f(t). \tag{5.4}$$

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. Figure 5.2 shows the comparison between different choices of γ.

We aim to learn meaningful parameters representing damped modes which reduce to zero in the absence of input. As such it is beneficial to enforce a positivity constraint on input $f(t)$ via a link function $\phi(\cdot)$. This has two major benefits. Firstly, the new excitation force $\phi(f(t))$ becomes interpretable as a physical entity; positive energy driving the system. Secondly, it encourages the optimiser to learn damping coefficients U_d that are more physically realistic (i.e. larger / more damped), since they must enable the system to reduce to zero when $\phi(f(t)) = 0$, whereas in the unconstrained case this could be achieved via negative inputs rather than damping. As in previous chapters, we use the softplus $\phi(f(t)) = \log(1 + \mathrm{e}^{f(t)})$.

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

$$\frac{\mathrm{d}a_d(t)}{\mathrm{d}t} + U_d a_d^{\gamma}(t) = V_d \phi(f(t)). \tag{5.5}$$

If we place a GP prior over $f(t)$, then Eq. (5.5) becomes an SDE and is of the form described in section 2.6, a nonlinear latent force model.

Inference in nonlinear latent force models Now that our model is based on nonlinear functions of both the amplitudes $a_d(t)$ and the latent forcing function $f(t)$, inference is no longer straightforward and, as in chapter 4, we must use approximate Gaussian filtering methods.

Constructing our SDE model in a general form by stacking the state vectors for the output processes, $\tilde{\mathbf{a}}_d(t)$, and the latent GP, $\tilde{\mathbf{f}}(t)$, in a new vector $\tilde{\mathbf{z}}(t) = \left(\tilde{\mathbf{a}}_1(t), \ldots, \tilde{\mathbf{a}}_D(t), \tilde{\mathbf{f}}(t)\right)^{\top}$, we get

$$\frac{\mathrm{d}\tilde{\mathbf{z}}(t)}{\mathrm{d}t} = \mathcal{F}(\tilde{\mathbf{z}}(t)) + \mathcal{L}(\tilde{\mathbf{z}}(t))\mathbf{w}(t), \tag{5.6a}$$

$$\mathbf{y}_k = \mathbf{H}\tilde{\mathbf{z}}(t_k) + \sigma_y \boldsymbol{\varepsilon}_k \tag{5.6b}$$

for nonlinear functions $\mathcal{F}$ and $\mathcal{L}$, chosen to represent our model in Eq. (5.5). $\mathbf{w}(t)$ is a white noise process and $\boldsymbol{\varepsilon}_k$ is D-dimensional i.i.d. Gaussian noise. It is important to note that our observations $\mathbf{y}_k \in \mathbb{R}^D$ are the *observed amplitudes* obtained after applying sinusoidal modelling to the audio signal, not the signal itself.

The time derivatives of the Kalman filter mean and covariance, which for the linear case are given by Eq. (2.17), must now be reformulated in terms of our new state space model. They can be written as (Hartikainen et al., 2012),

$$\frac{\mathrm{d}\mathbf{m}(t)}{\mathrm{d}t} = \mathbb{E}\left[\mathcal{F}(\tilde{\mathbf{z}}(t))\right], \tag{5.7a}$$

$$\begin{aligned}\frac{\mathrm{d}\mathbf{P}(t)}{\mathrm{d}t} &= \mathbb{E}\left[(\tilde{\mathbf{z}}(t)) - \mathbf{m}(t)\mathcal{F}(\tilde{\mathbf{z}}(t))^{\top}\right] + \mathbb{E}\left[(\mathcal{F}(\tilde{\mathbf{z}}(t))\tilde{\mathbf{z}}(t)) - \mathbf{m}(t)^{\top}\right] \\ &\quad + \mathbb{E}\left[\mathcal{L}(\tilde{\mathbf{z}}(t))\mathbf{Q}_c\mathcal{L}(\tilde{\mathbf{z}}(t))^{\top}\right].\end{aligned} \tag{5.7b}$$

We follow Hartikainen et al. (2012), solving these differential equations by first approximating the integrals required in the expectations with sigma-point methods (as in section 4.5), and then applying a numerical ODE solver in Matlab.

Learning the parameters in our LFM framework proceeds as usual by maximising the marginal likelihood, but we now have a high-dimensional optimisation problem, since we have parameters U_d and V_d to estimate for all $d = 1, \ldots, D$ outputs in addition to the hyperparameters of the GP kernel for the latent input (we use the Matérn-$5/2$ kernel). 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.

Selecting the modes In this section, we aim to isolate the true vibrational modes of an audio recording, and discard the signal content relating to broadband noise. This is largely due to the fact that modelling all of the sinusoids that relate to the noisy, broadband content would be impractical. A separate noise-based approach could be used to model the remaining sinusoids, but we don't address that here (see section 5.3 for an extension of the approach which captures the entire signal content).

We must therefore identify which partials in the sinusoidal model are representative of the vibrational modes. If our analysis signal has strong harmonic content (as in 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 D partials. In Figure 5.3, 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

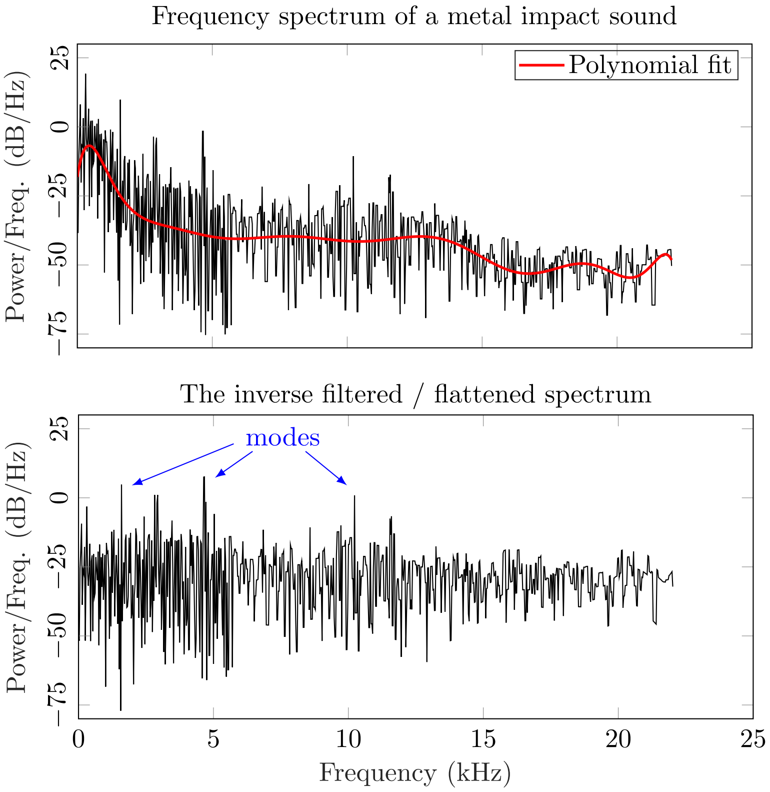


Figure 5.3: A filter is designed by fitting a polynomial to the shape of the frequency spectrum (**top**). The filter is inverted and applied to the signal to flatten the spectrum (**bottom**). Peaks in the flattened spectrum are then used to pick the vibrational modes of the signal.

the peaks of the filtered spectrum. Once we have selected our D modes, we calculate the median frequency value for each partial, and then treat that frequency as fixed.

Resynthesis with the state space model After tuning the model parameters and inferring a posterior over the outputs and the latent force, it is also useful to project the latent force through the physical system, i.e. through a discretised state space version of Eq. (5.5), in order to determine how much of the amplitude behaviour has been encoded. This is *not* the same as studying the posterior mean of the outputs, since the likelihood component is not taken into account (so the values are not updated based on the observations). Performing synthesis in this manner allows us to compare our method to other dimensionality reduction techniques, and also to synthesise novel sounds by passing new latent forces through the model.

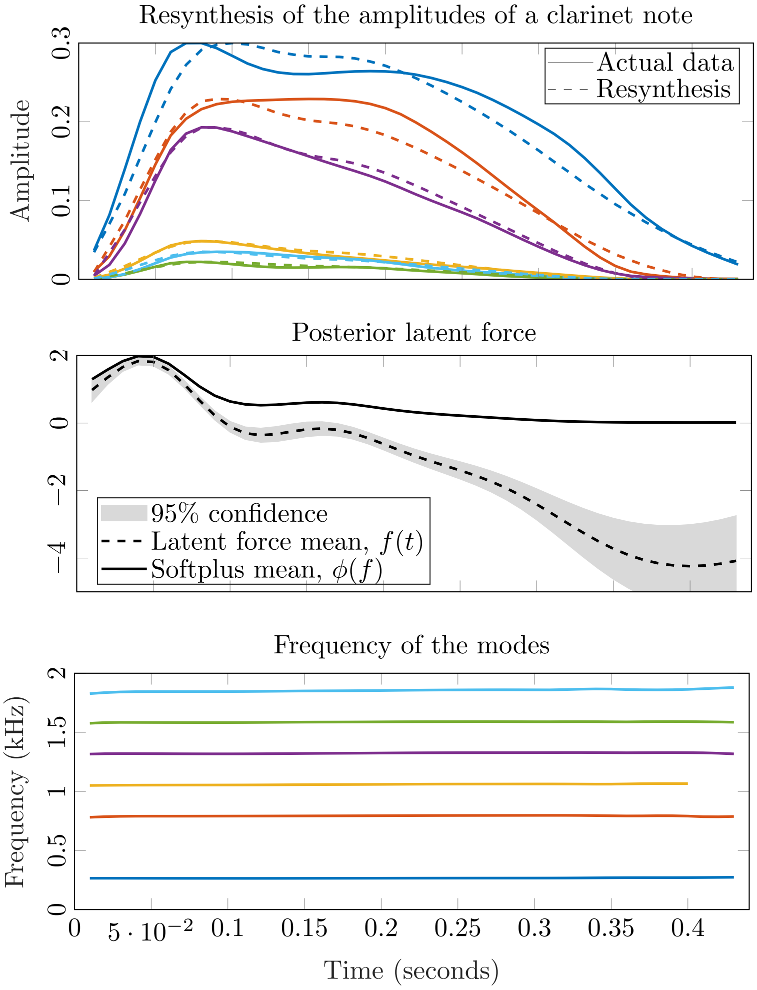


Figure 5.4: Latent force modelling of a clarinet note. Six modes are picked based on their amplitude, and resynthesis is performed by propagating the latent force (**middle**) through the state space model. The output is shown in comparison to the real data (**top**). The frequency data (**bottom**) shows the modes are, in order of magnitude, the 1st, 3rd, 5th, 4th, 7th and 6th harmonics. The mean and 95% confidence interval (uncertainty) of the latent input $f(t)$ is shown.

The discrete form of the model for a single output a_d can be written

$$\begin{pmatrix} a_d(t_k) \\ \dot{a}_d(t_k) \end{pmatrix} = \begin{pmatrix} 1 & \Delta t_k \\ 0 & 0 \end{pmatrix} \begin{pmatrix} a_d(t_{k-1}) \\ \dot{a}_d(t_{k-1}) \end{pmatrix} + \begin{pmatrix} 0 \\ -U_d \end{pmatrix} a_d^{\gamma}(t_{k-1}) + \begin{pmatrix} 0 \\ -V_d \end{pmatrix} \phi(f(t_k)) \tag{5.8}$$

for time step size Δt_k.

Figure 5.4 shows the result of the LFM applied to a clarinet note. The amplitude of six harmonics are modelled, and the learnt latent force is propagated through the model to resynthesise the outputs. Whilst not all the detail is captured, we can see that the outputs exhibit variable

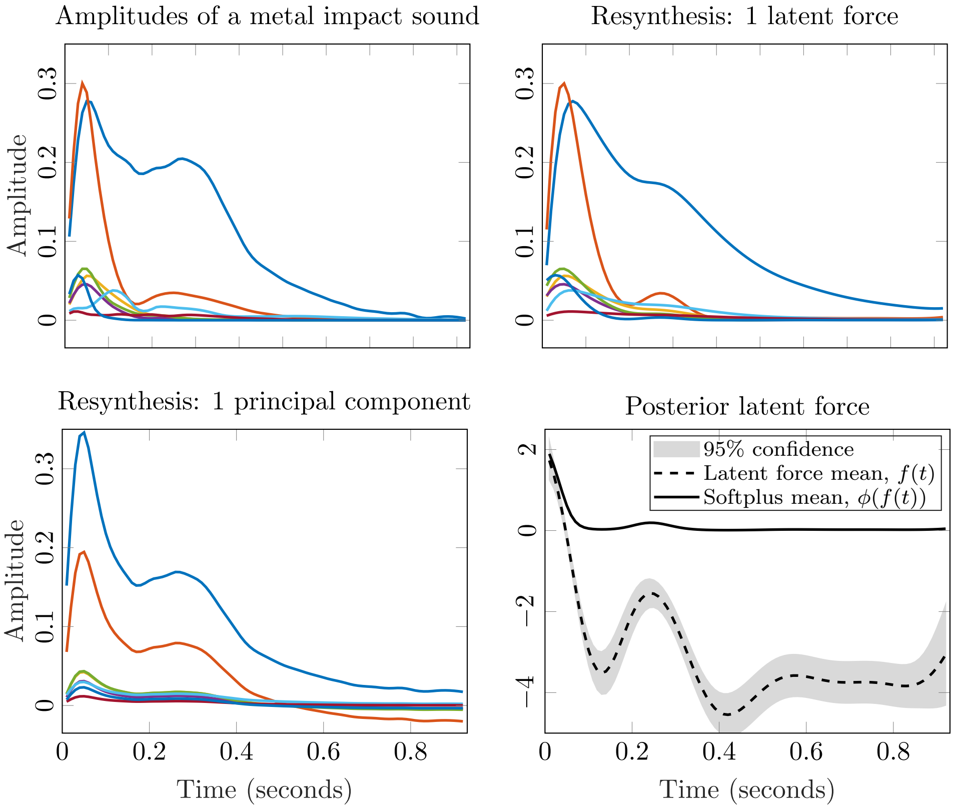


Figure 5.5: Latent force modelling of a metal impact sound. The real data shows some variation in behaviour between modes (**top left**). An increase in uncertainty in the latent posterior after 0.1s reflects this fact (**bottom right**). The learnt latent force is fed through the state space model, 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**).

damping rates.

Because of this ability to capture variable behaviour across modes, we expect our one-dimensional LFM to perform a more effective data compression than other dimensionality reduction techniques that reduce high-dimensional data down to a one-dimensional manifold. In Figure 5.5 we compare the resynthesised data with the LFM to the result when we apply PCA to the amplitude data and then reproduce them using just one principal component. The PCA approach cannot capture variable behaviour between modes since every output is modelled as a scaled version of the principal component. Table 5.1 shows the RMS error comparison using this approach for 5 musical instrument notes.

	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 5.1: Root-mean-square (RMS) error between modal amplitude data and outputs of the latent force model (LFM) and principal component analysis (PCA) with one principal component. The LFM outperforms PCA when disparate behaviour across dimensions is observed.

5.2 Real-Time Synthesis and Sound Morphing

After learning the parameters of the hybrid physical-statistical model we can adapt the model beyond simple reconstruction of the original amplitude envelopes. In the previous section Δt_k was fixed at the analysis time step size, corresponding to frame-wise modelling. During synthesis we can set the step size to be as large or small as required, based on our desired sampling frequency, 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. Synthesis of novel signals 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 implemented a system that controls 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[1].

[1] The interactive model (which includes the ability to morph between sounds) can be found at http://c4dm.eecs.qmul.ac.uk/audioengineering/latent-force-synthesis/

Sound morphing Our 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 d^{th} harmonic in model A to the d^{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 the modes have been paired, we perform linear interpolation of physical parameters U_d, V_d and the initial conditions, and logarithmic interpolation of the frequency. Synthesis in this manner negates the need for cumbersome time-domain modification (such as time-stretching) usually associated with morphing (Caetano and Rodet, 2013).

Figure 5.6 shows an example of sound morphing between an oboe and a clarinet. A manually-drawn excitation function is used for the morphed signal. Observing the newly synthesised amplitude envelopes (top row, middle column) we can see that the relative magnitudes of the envelopes has been modified and that the damping rates correspond to the mid-point between the highly-damped oboe and the lesser-damped clarinet. Importantly, the new signal is not constrained in duration by either of the recordings (note the different x-axis scales).

5.3 A Generative Model for Natural Sounds

In this section we look to build upon the methods laid out above, making them applicable to a larger class of natural sounds. Instead of focusing on controllable synthesis and physical interpretations, we look to extend the model in a number of ways. First, we model the amplitude envelopes of the outputs of a auditory filterbank, and synthesise the carrier envelopes with a sinusoids-plus-noise approach. This enables us to analyse the

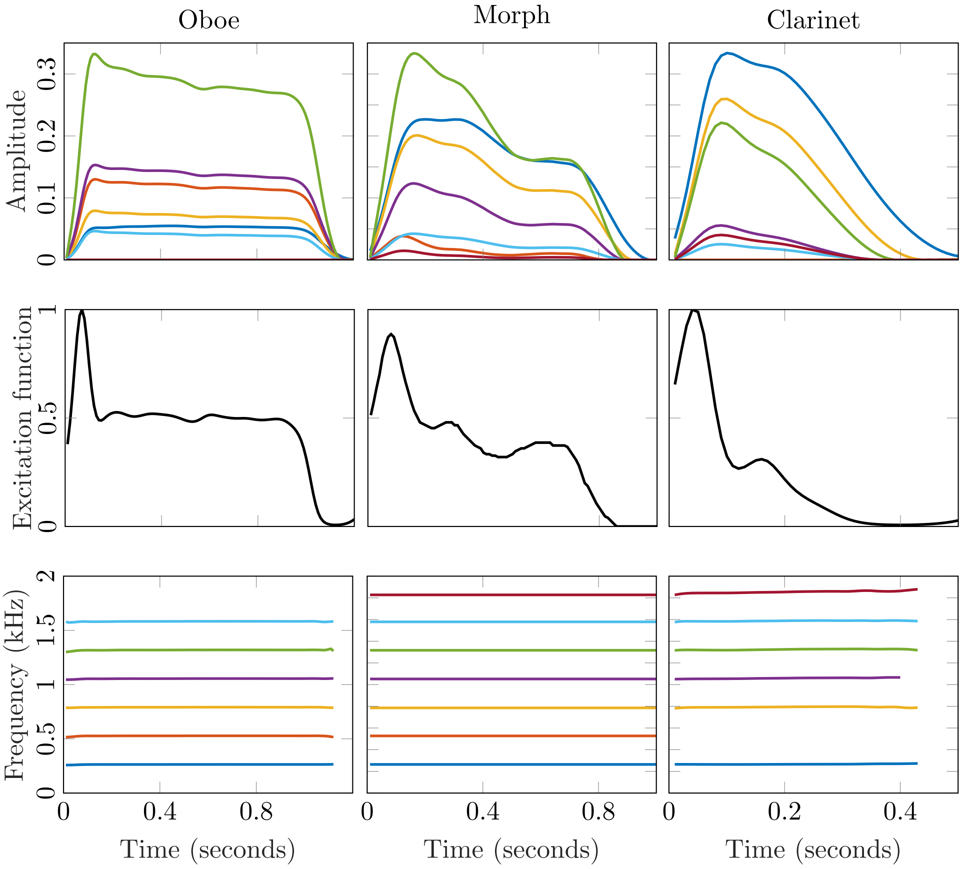


Figure 5.6: Sound morphing between an oboe and a clarinet. The modes of an oboe (**left column**) are matched with the modes of a clarinet (**right column**) and colour-coded based on their pairings. Since the modes represent harmonics, it is important to maintain the harmonic structure, so the 2nd mode 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 column**).

behaviour of the entire signal, rather than just the most important modes. We also allow the model to have multiple latent forces, greatly increasing the level of detail that can be captured. Finally, we include higher-order feedback and delay terms in the state space model, such that behaviour at a given time step can be affected by energy in the latent force and the outputs from multiple time steps in the past.

To obtain amplitude data in the desired form we pass an audio signal through an equivalent rectangular bandwidth (ERB) filter bank. We then use Gaussian process probabilistic amplitude demodulation

(GPPAD) (Turner and Sahani, 2011) to calculate the subband envelopes and their corresponding carrier signals. GPPAD allows for control over demodulation time-scales via GP lengthscale hyperparameters. We are concerned with slowly varying behaviour correlated across the frequency spectrum, in accordance with the observation that the human auditory system summarises sound statistics over time (see section 2.2). Fast-varying behaviour is relegated to the carrier signal and will be modelled as independent filtered noise.

The number of channels in the filter bank and the demodulation lengthscales must be set manually during this first analysis stage. We set the number of filters to be 16 in order to prevent the state space model from getting too large (since the Kalman filter methods scale cubically in the state dimensionality). We choose the GP lengthscales such that we capture amplitude behaviour occurring over durations of 10ms and slower.

Augmented latent force models for amplitude envelopes The ODE model presented so far, Eq. (5.5) is, overly simplistic in that it does not take into account variable decay behaviour due to internal damping or feedback and other nonstationary effects which occur as a sound is generated and propagates towards a listener.

To account for this complex behaviour, we extend our *discrete* model such that predictions at the current time step t_k can be influenced explicitly by predictions from multiple time steps in the past. Our final discrete model, which now allows for multiple forces $f_n(t_k)$, can be described as

$$\dot{a}_d(t_k) = -\hat{U}_d a_d^{\gamma_d}(t_k) + \sum_{p=1}^{P} \hat{Z}_{d,p} a_d(t_{k-p}) + \sum_{q=1}^{Q} \sum_{n=1}^{N} \hat{V}_{d,n,q} \phi(f_n(t_{k-q})). \quad (5.9)$$

Parameters $\hat{Z}_{d,p}$ are *feedback* coefficients which determine how the current output is affected by output behaviour from p time steps in the past. $\hat{V}_{d,n,q}$ are discrete *lag* parameters which determine how sensitive the current output is to input n from q time steps ago. $\hat{U}$ is the discrete version of the damping parameter.

The lag term is important since modes of vibration in a sounding object tend to be activated at slightly different times due to deformations in the object as it vibrates, and due to the interaction of multiple modes of vibration. It can also capture effects due to reverberation. The feedback terms allow for long and varied decay behaviour that can't be described by simple exponential decay.

The challenge is to incorporate Eq. (5.9) into our inference procedure. We do this by augmenting our state vector $\tilde{\mathbf{z}}(t_k)$ and transition model $\mathcal{F}(\tilde{\mathbf{z}}(t))$ with new rows corresponding to the delayed terms. After each time step the current states $\{a_d(t_k)\}_{d=1}^{D}, \{f_n(t_k)\}_{n=1}^{N}$ are "passed down" such that at the next time step they are in the locations corresponding to feedback and lag terms. When performing the Kalman filter prediction step, augmented states are included since they influence predictions for the current state, however the predictions for these augmented entries are simply exact copies from the previous time step.

Figure 5.7 shows the latent posterior for a metal impact sound with one latent force, $N = 1$. The mean of the distribution (the minimum least squares error estimate) is passed through the discrete model, Eq. (5.9), to reconstruct the amplitude envelopes. Despite the single latent force, we observe that some of the complex modulation behaviour has been learnt. Additionally, the latent force is both smooth and sparse, and the reconstructed envelopes have a slow decay despite this sparsity.

Generating novel instances of natural sounds A significant benefit of generative probabilistic methods such as the LFM is that, as well as providing us with uncertainty information about our predictions, they provide the means to sample new latent functions from the learnt *prior* distribution. That is, the model prior after the parameters have been optimised. By passing these new functions through the model we can generate novel amplitude envelopes. These envelopes modulate carrier signals produced using a sinusoids-plus-noise approach based on analysis of the original carriers. The subbands are then summed to create a new synthetic audio signal distinct from the original but with similar characteristics.

Sampling from the prior generates functions with appropriate smoothness and magnitude, however the desired energy sparsity is not guaranteed. Latent functions are modelled independently, but in practice they tend to co-occur and are activated in similar regions of the signal. We use GPPAD again to demodulate our latent functions with a slowly varying envelope, then fit a GP with a exponentiated quadratic covari-

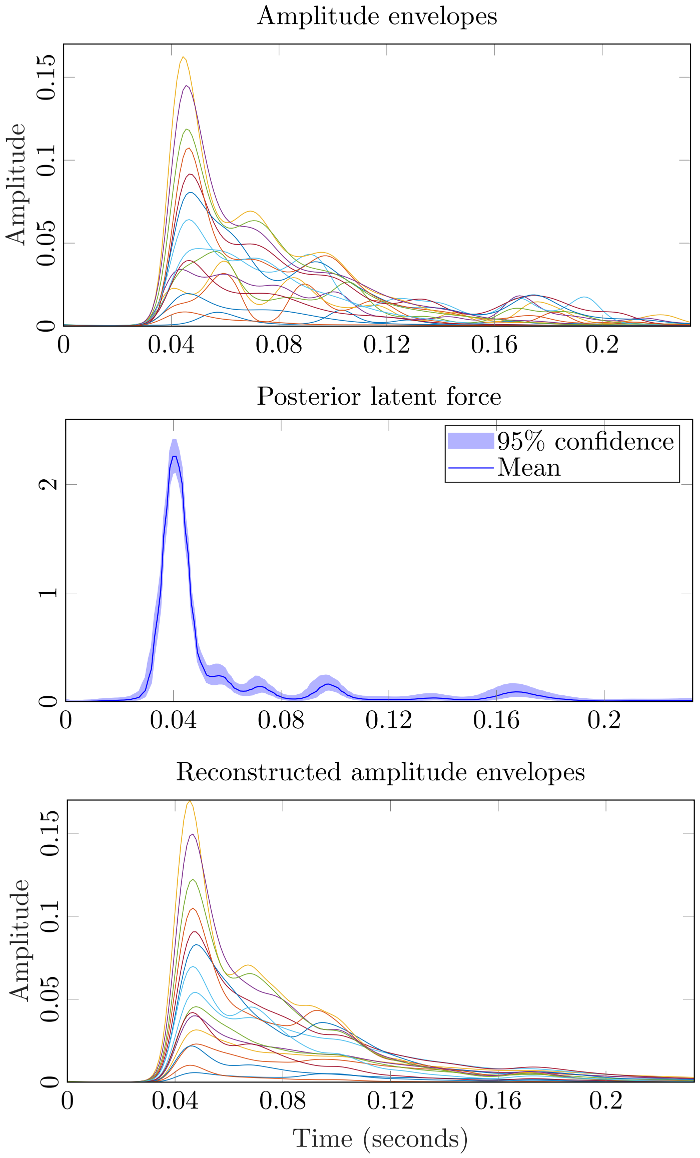
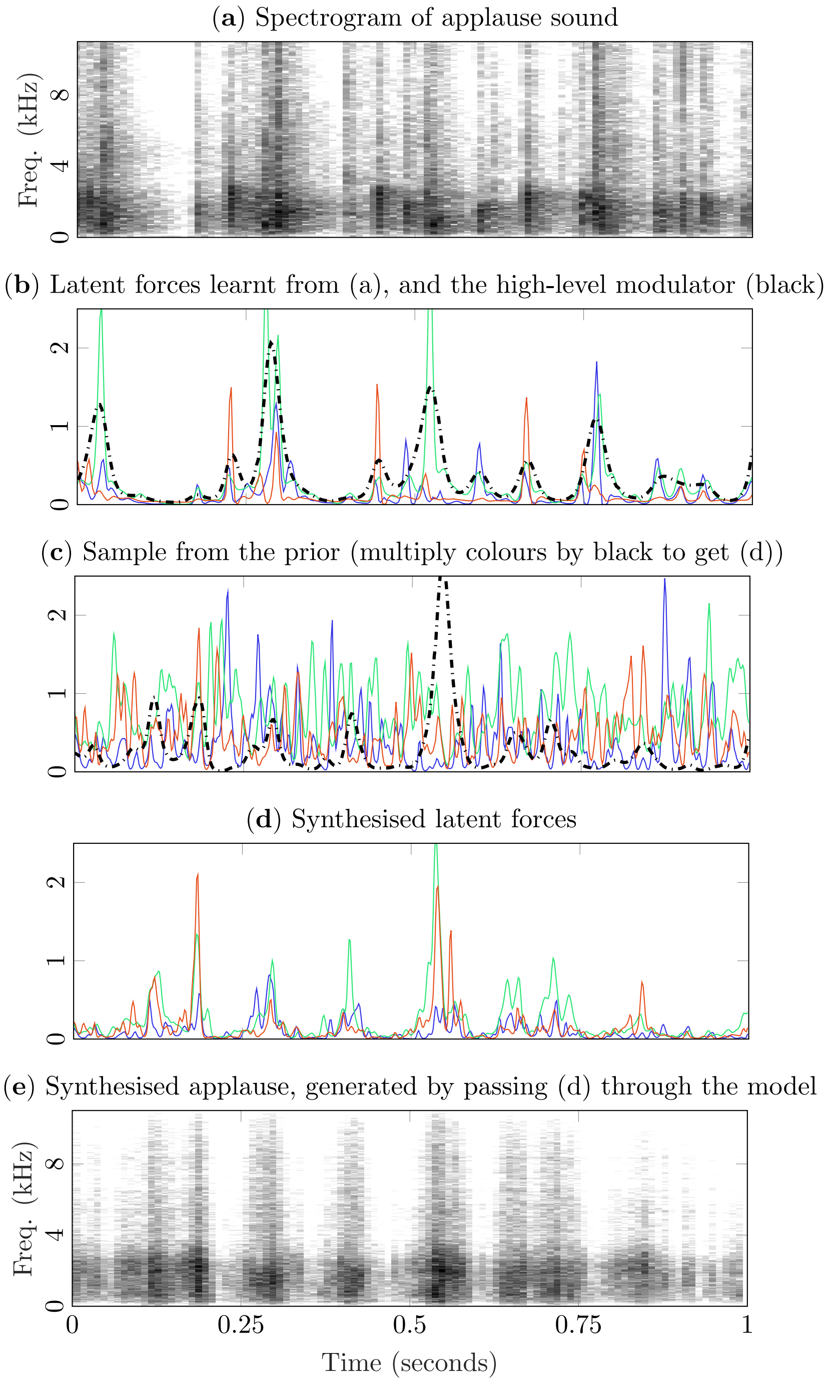


Figure 5.7: LFM applied to a metal impact sound, with mean and 95% confidence of the latent posterior shown. The mean is passed through the discrete model (Eq. (5.9)) to reconstruct the envelopes. Some complex behaviour in the decay section of the amplitudes is maintained despite using a single excitation force.

ance function to this envelope. We sample from this high-level process and use it to modulate our newly generated latent functions; the result of this product is latent behaviour with sparse energy, as demonstrated in Figure 5.8.

Optimisation settings The full set of model parameters including GP lengthscales ℓ_n, $\{\hat{U}_d, \hat{Z}_{d,p}, \hat{V}_{d,n,q}, \gamma_d, \ell_n\}$, becomes large as P, Q and N increase. To alleviate issues that occur when our parameter space



(**a**) Spectrogram of applause sound

(**b**) Latent forces learnt from (a), and the high-level modulator (black)

(**c**) Sample from the prior (multiply colours by black to get (d))

(**d**) Synthesised latent forces

(**e**) Synthesised applause, generated by passing (d) through the model

Figure 5.8: LFM generative model with 3 latent forces applied to an applause sound (**a**). The high-level modulator (black line in (**b**)) is calculated by demodulating the latent forces. New latent functions are sampled from the prior (**c**), and then multiplied by a newly sampled modulator to ensure they co-occur and have the correct sparsity. The resulting functions (**d**) are propagated through the state space model to synthesise the output amplitudes (**e**).

becomes large we sparsify the feedback and sensitivity parameters. For example, if $P = 10$, we may manually fix $\hat{Z}_{d,p}$ to zero for $p \in [3, 4, 6, 7, 9]$ such that only half the parameters are included in the optimisation procedure.

Reliability of the optimisation procedure suffers as the number of parameters increases, so in practice all D frequency channels are not optimised together. We select the 6 envelopes contributing the most energy and train the model on the observations from only these channels. The remaining channels are then appended and optimised whilst keeping the already-trained parameters fixed. This improves reliability but prioritises envelopes of high energy. We also skip prediction steps for periods of the signal that are of very low amplitude, which speeds up the filtering step. Despite these adjustments, optimisation still takes up to 36 hours for a 2 second sound sample due to the need to run the full Kalman filter process multiple times in each iteration to estimate the parameter gradients.

5.4 Evaluation of Latent Force Models for Natural Sound

To evaluate our method we collated a set of 20 audio recordings, selected as being representative of everyday natural sounds[2]. Music and speech sounds were not included, nor were sounds with significant frequency modulation, since our model doesn't capture this behaviour.

Objective evaluation: reconstruction error of original sound We analyse our ability to reconstruct the original data by projecting the latent representation back to the output space. For the LFM this means passing the mean of the latent posterior through the state space model. Figure 5.9 shows reconstruction RMS error and cosine distance of the LFM compared to similar generative models based on temporal NMF (tNMF, see section 2.2) and NMF for the 20 recordings. The smoothness constraint enforced by placing a GP prior over the latent functions negatively impacts the reconstruction. This is demonstrated by the fact that tNMF performs poorly from an RMS error perspective. Despite this, the LFM has much descriptive power, and is sometimes capable of achieving a lower RMS error than the unconstrained NMF.

[2] From freesound.org and from the Natural Sound Stimulus set: mcdermottlab.mit.edu/svnh/Natural-Sound/Stimuli.html

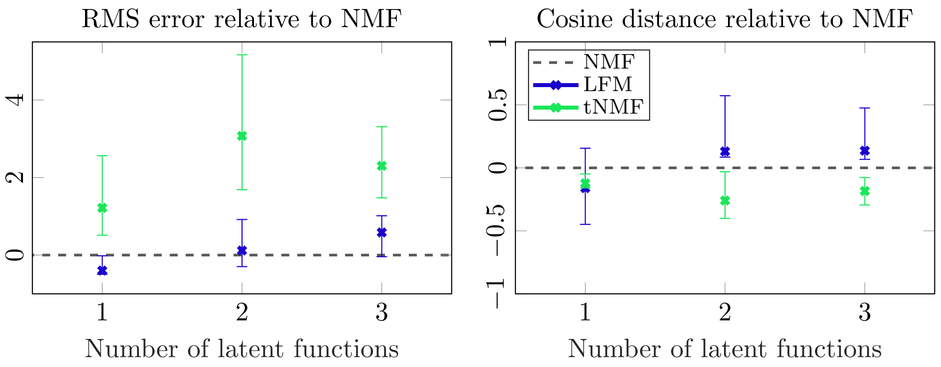


Figure 5.9: Reconstruction error of LFM and tNMF plotted relative to NMF. Crosses represent the median, error bars range from first to third quartile.

Interestingly however, tNMF consistently outperforms the other two models based on cosine distance.

Subjective evaluation: listening test for novel sounds Objective results suggest that smoothness constraints harm reconstruction of the original signal. However, our aim is to learn realistic latent representations that will be the foundation of a generative model. To test their suitability, we designed an experiment to compare generative models based on LFM, NMF and tNMF. The approach outlined in section 5.3 was used for all model types. Since NMF is non-probabilistic, it does not provide an immediate way in which to sample new data, therefore GPs were fit to the latent functions after analysis.

Our experiment followed a multi-stimulus subjective quality rating paradigm[3]: 24 participants were shown 20 pages (order randomised), one per sound example, and asked to listen to the reference recording and then rate 7 generated sounds (2 from each model plus an anchor, presented in random order) based on their credibility as a new sound of the same type as the reference. Ratings were on a scale of 0 to 1, with a score of 1 representing a very realistic sound. Figure 5.10 shows the mean realism ratings. Whilst variation was large between sound examples, LFM was generally rated as more realistic than the other methods.

We applied a generalised linear mixed effects model (GLMM), with beta regression, in which *sound example* and *participant* were treated

[3] The test was run online and implemented with the Web Audio Evaluation Tool: github.com/BrechtDeMan/WebAudioEvaluationTool

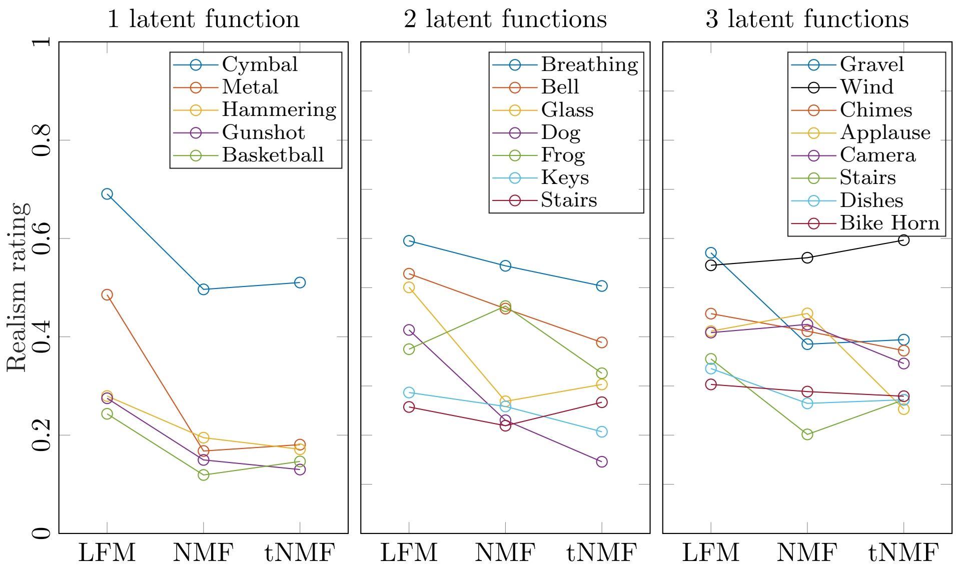


Figure 5.10: Mean realism ratings obtained from the listening test.

		LFM vs. NMF	LFM vs. tNMF	NMF vs. tNMF
All sounds	Estimate	0.3839	0.4987	0.1148
	p-value	**<1e-04**	**<1e-04**	0.3750
1 latent fn.	Estimate	0.8248	0.7976	-0.0272
	p-value	**<1e-05**	**<1e-05**	0.9980
2 latent fns.	Estimate	0.3140	0.5134	0.1994
	p-value	**0.0448**	**<0.001**	0.3218
3 latent fns.	Estimate	0.2052	0.3243	0.1191
	p-value	0.2867	**0.0285**	0.7154

Table 5.2: GLMM with three-way comparison applied to listening test results. LFM received higher mean ratings, but confidence decreases with number of latent forces, indicated by increasing *p-values*. *Estimate* can be interpreted as the ratio increase in realism rating when choosing model A over model B.

as random effects. Table 5.2 shows that the mean realism rating was highest for LFM regardless of the number of latent functions. The difference was significant at a 5% level except for LFM vs. NMF with 3 latent functions. This suggests that for sounds requiring many latent functions to capture their behaviour, such as textural sounds, LFM may not offer a significant gain over purely statistical approaches. For example, the wind recording, a textural sound whose envelopes do not exhibit clear exponential decay, was captured best with tNMF.

5.5 Conclusion

In the first half of the chapter, section 5.1 and section 5.2, motivated by physical observations about how sound behaves, and drawing explicit links with existing physically-inspired sound synthesis methods, i.e. modal synthesis, we incorporated exponential decay into a model for the vibrational models of an audio signal. In the case of sound events that we can safely assume have a single excitation force, such as an impact sound, we see some clear benefits of the method as a dimensionality reduction technique as well as a tool for synthesis and sound morphing. For many example recordings we tested, the amount of information captured in the model with a single input exceeded that captured by PCA or NMF.

In the second half of the chapter, section 5.3 and section 5.4, we extended these ideas to come up with a comprehensive generative model for natural audio signals. Again we see from inspecting the synthesised outputs that interesting behaviour is modelled; variable decay rates and complex modulation patterns.

How then, do we reconcile this knowledge with the fact that for more complex sound events that require multiple forces, the LFM does not consistently outperform NMF from a reconstruction RMSE perspective? The smoothness constraints over the latent forces certainly contribute to this result. However, our approximate inference methods also come at a cost in terms of speed and reliability. Numerically solving ODEs at every time step, whilst also using sigma-point methods to approximate the required integrals is not only slow, but also leads to numerical issues particularly when the signal falls below the noise floor and correlation structure is lost.

These issues mean that subjective evaluation was required to demonstrate the benefits of the proposed statistical-physical model. Our experiment showed that listeners consistently rated synthesis with the LFM as more realistic than similar generative models that don't exhibit any temporal asymmetry in their envelope patterns. This suggests that exponential decay is an important perceptual characteristic of natural sound.

By demonstrating the benefits of physical assumptions in this manner, we show that it is important to go beyond simplistic linear models for spectrograms, but that more research is needed in developing efficient and reliable inference methods in such a setting. Once achieved, the LFM could be combined with probabilistic time-frequency analysis in a manner similar to chapter 4, leading to a physically inspired generative model acting directly on the waveform.

Chapter 6

Deep Gaussian Processes as a Nonlinear Model for Audio Spectrograms

In this chapter we study the potential of *nonlinear* models for audio spectrograms. Although the LFM in the previous chapter contained a nonlinear mapping in order to enforce the latents to be positive, both the LFM and NMF assume that the amplitude envelopes are produced by a linear mixing of positive latent functions.

In general, the sound production mechanism is likely to be nonlinear and nonstationary due to the complex, sometimes chaotic, interaction between objects, materials or surfaces. This motivates the replacement of the linear mapping from latents to outputs with a nonlinear function characterised by another Gaussian process. This results in a two-layer instantiation of a deep GP (Damianou and Lawrence, 2013), the name used to describe the composition of multiple layers of GP functions.

We present a deep GP model for nonnegative temporal data, and we also propose some modifications and constraints to the standard approach based on empirical observations and our knowledge about how natural sound behaves. Most notably we incorporate monotonicity information into the multi-layer model. The validity of these modifications is assessed through a missing data synthesis task applied to spectrograms of both speech and sound texture recordings.

6.1 Temporal Deep Gaussian Processes for Nonnegative Data

Taking the model in Eq. (2.44) with just two layers, $L = 2$, and treating time, t, as the input provides us with a nonlinear model for audio amplitude data. Since the function that maps the latents to the outputs is learnt from nonnegative observations, we may expect the GP to always output positive predictions. In practice however, we found that when sampling from the latent processes and passing them through the second layer, negative values were often generated.

Another difficulty with expecting the GP to learn a positive mapping is that it makes the choice of mean function non-trivial. A linear mean function can encourage negative values, whilst a zero mean function is a poor choice when the outputs are large, and doesn't strongly discourage values below zero.

These reasons motivate the use of a likelihood model that explicitly describes the nonnegativity of the data. As in previous chapters, we impose this property via the softplus mapping, $\phi(\cdot)$, and we obtain our proposed nonlinear model for audio spectrograms:

$$f(t) \sim \mathrm{GP}\left(\mu_f(t), C_f(t, t')\right), \tag{6.1a}$$

$$g(t) \sim \mathrm{GP}\left(\mu_g(f(t)), C_g(f(t), f(t'))\right), \tag{6.1b}$$

$$\mathbf{a}_{1:D} \sim \prod_{k=1}^{T} \mathrm{N}(\phi(g(t_k)), \sigma_y^2), \tag{6.1c}$$

for observed amplitude envelopes $\mathbf{a}_d$, $d = 1, \ldots, D$. We have omitted any noise between the layers since this can be folded into the kernel C_f.

In this model $f : \mathbb{R} \to \mathbb{R}^N$ and $g : \mathbb{R}^N \to \mathbb{R}^D$ are multi-output GPs. As such, all N latent dimensions from the first layer share a single set of hyperparameters (e.g. lengthscale and variance). This means that all the detailed time-varying behaviour in the data must be described by the nonlinear mapping g.

This contrasts with the LFM and tNMF models discussed previously, whose latent dimensions were modelled with N separate one-dimensional GPs, each with its own lengthscale. Therefore, in line with our motivation for the deep GP model as a nonlinear extension of these methods, in which the mapping to the outputs is another GP, we also implement an alternative version where the input to the second layer is a concatenation of the outputs of multiple GPs, $f_n : \mathbb{R} \rightarrow \mathbb{R}$, giving

$$
\begin{aligned}
f_n(t) &\sim \mathrm{GP}\left(\mu_{f_n}(t), C_{f_n}(t, t')\right), && (6.2a)\\
\mathbf{f}(t) &= \left(f_1(t), \ldots, f_N(t)\right)^\top, && (6.2b)\\
g(t) &\sim \mathrm{GP}\left(\mu_g(\mathbf{f}(t)), C_g(\mathbf{f}(t), \mathbf{f}(t'))\right), && (6.2c)\\
\mathbf{a}_{1:D} &\sim \prod_{k=1}^{T} \mathrm{N}\left(\phi(g(t_k)), \sigma_y^2\right). && (6.2d)
\end{aligned}
$$

This model requires optimisation of a larger number of hyperparameters compared to Eq. (6.1), but enables us to capture multiple lengthscales in the latent functions, which can lead to a more expressive model[1]. From here on, we will call this model **DGP-multi**, reflecting the fact that it has multiple independent latent GPs. It is possible that similar expressiveness could be obtained by increasing the depth of the model ($L > 2$), but for ease of comparison we restrict our analysis to the two-layer case.

6.2 Approximate Inference in the Deep GP Model

The posterior distribution of interest, $p(\mathbf{f}, g \mid \mathbf{a}_{1:D})$, is intractable, as is the case with all deep GP models since they involve Gaussian distributions being propagated through nonlinearities. Hence we must resort to approximate inference.

We extend the doubly stochastic variational inference framework based on inducing points for both proposed models (Salimbeni and Deisenroth, 2017), which assumes a factorisation between layers and leads to the following joint distribution for the standard model,

$$
p(\mathbf{a}_{1:D}, f, g) = p(f)p(g)p(\mathbf{u}_f)p(\mathbf{u}_g)\prod_{k=1}^{T} p(\mathbf{a}_{1:D,t} \mid \phi(g(t_k))), \qquad (6.3)
$$

where $\mathbf{u}_f$ and $\mathbf{u}_g$ are the inducing points for f and g respectively. For

[1] Eq. (6.2) can now be seen as an extension of a latent variable model, say NMF. The importance of the nonnegativity constraint on the latents in NMF motivated us to experiment with imposing nonnegativity on the outputs of f_n. However, we found little benefit from doing so, and the inclusion of the sotfplus mapping in the likelihood means that this is not required.

the model with multiple independent latent GPs this is

$$p(\mathbf{a}_{1:D}, \mathbf{f}, g) = p(f_1) \dots p(f_N) p(g) p(\mathbf{u}_{f_1}) \dots p(\mathbf{u}_{f_N}) p(\mathbf{u}_g) \\ \times \prod_{k=1}^{T} p(\mathbf{a}_{1:D,t} \,|\, \phi(g(t_k))). \tag{6.4}$$

As is generally the case in the literature, a Gaussian approximation is used for each set of inducing points, $q(\mathbf{u}_i) = \mathrm{N}(\mathbf{m}_i, \mathbf{S}_i)$, and $\mathbf{m}_i$, $\mathbf{S}_i$ are treated as parameters of the model to be optimised. Then the approximate posterior has a simple factorised form given by

$$q(\mathbf{f}, g \,|\, \mathbf{a}_{1:D}) = p(f_1) \dots p(f_N) p(g) q(\mathbf{u}_{f_1}) \dots q(\mathbf{u}_{f_N}) q(\mathbf{u}_g). \tag{6.5}$$

Even given the above approximations, it is not possible to calculate the marginal densities or the marginal likelihood analytically. Instead, a nested Monte Carlo method is used for both quantities (hence the term *doubly stochastic*). We omit the full details here, since we use the approach described by Salimbeni and Deisenroth (2017), but we refer the reader to Chapter 4 of Bui (2018) for a detailed discussion of the method and a derivation of the variational lower bound on the marginal likelihood.

The difference in our case is twofold. Firstly, for DGP-multi we must construct the variational distributions for all the dimensions of $\mathbf{f}$ and then draw samples from these processes individually, after which we concatenate the results before propagating to the next layer.

Secondly, we must account for the softplus mapping, $\phi(g)$, which makes the likelihood non-Gaussian in the latent process g. To do so, we implement a Gaussian likelihood model which utilises a softplus link function to propagate samples in the desired fashion. In this case, the expectations required to calculate the marginal likelihood, which are (see Bui (2018))

$$\sum_{k=1}^{T} \mathbb{E}_{q(\mathbf{f}, g \,|\, \mathbf{a}_{1:D,k})} \left[\log p(\mathbf{a}_{1:D,k} \,|\, \phi(g(t_k))) \right], \tag{6.6}$$

can no longer be derived in closed form, so a Gauss-Hermite quadrature routine is employed (recall that $q(\mathbf{f}, g \,|\, \mathbf{a}_{1:D,k})$ is Gaussian).

	Nonlinear?	Uncertainty?	Interpolation?	example RMSE
NMF	✘	✘	✘	0.0174
tNMF	✘	✓	✓	0.0351
LFM	✘	✓	✓	0.0234
GP-LVM	✓	✓	✘	0.0016
DGP	✓	✓	✓	**0.0007**
DGP-multi	✓	✓	✓	0.0020

Table 6.1: A comparison of the proposed spectrogram models. Example RMSE is the reconstruction error when the latents are passed through the model for the glass breaking sound used in the case study. The deep GP models are expressive enough to fit the data well, whilst also being able to make predictions about unseen data and provide uncertainty estimates.

6.3 A Case Study of Linear and Nonlinear Spectrogram Models

In this section we elucidate the benefits of a nonlinear spectrogram model via visualisations and a case study comparing it to the linear models previously discussed. Throughout we will consider an example sound recording of *glass breaking* – a signal which exhibits nonstationary behaviour and multiple distinct events.

For ease of visualisation and comparison we use 16 frequency channels (D=16) and smooth the envelopes with GPPAD as in chapter 5. The methods we compare are NMF, tNMF, LFM, GP-LVM (Lawrence, 2005), DGP and DGP-multi. The GP-LVM is a nonlinear extension of probabilistic PCA that places a GP prior over the mapping from latent variables to observations.

Table 6.1 compares the methods' respective attributes in terms of whether they are linear / nonlinear, capable of quantifying uncertainty, and whether they allow for interpolation in the latent space to make predictions about unseen data points. We additionally demonstrate their expressivity as a dimensionality reduction technique, i.e. their ability to reconstruct the 16-dimensional glass-breaking signal from a 2-dimensional latent representation.

Figure 6.1 visualises the latent variables / functions for each of the methods. The first three, NMF, tNMF and LFM, all have linear mappings from the latent space to the outputs, and constrain the latents to be nonnegative. The last three, GP-LVM, DGP and DGP-multi, all have nonlinear mappings (via GPs) and do not constrain the latents

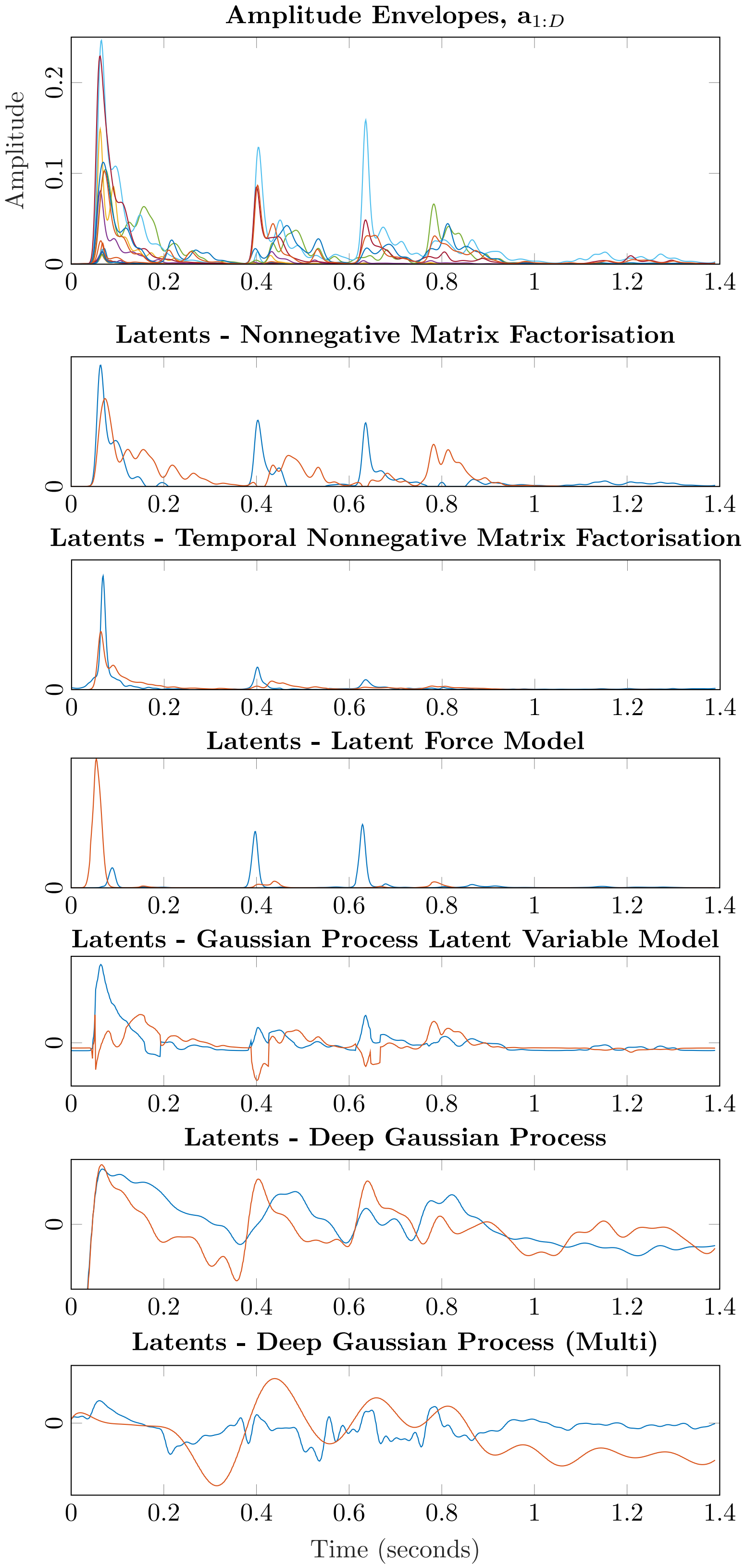


Figure 6.1: Comparison of latent variable models applied to the amplitude envelopes of a glass breaking sound (**top**). Latent means shown.

to be nonnegative. We can clearly see the effect of the GP prior over the latents of tNMF and LFM, since they exhibit much smoother behaviour than NMF. This smoothness is prohibitive in getting a close fit to the data, as demonstrated by their respective RMSE values shown in Table 6.1.

These observations suggest that a smoothness prior over latents in combination with a linear mapping does not result in model which is expressive enough to capture the behaviour of an audio signal. The GP-LVM is capable of fitting the data very well, since there are no smoothness constraints on the latents, however this leads them to exhibit discontinuities over time. Additionally, there is now no clear way to sample from the latent distribution in order to interpolate or generate data, since the prior over the latent variables are independent one-dimensional Gaussians.

The DGP models address these issues by allowing for temporal smoothness constraints and a nonlinear mapping, which means they can fit the data well, and also allow for interpolation or extrapolation. We can also see that the DGP-multi model now allows for the two latent dimensions to have different lengthscales and variances, however we notice that this doesn't always lead to an improved RMSE.

Figure 6.2 shows an example of the nonlinear GP mappings from the 2-dimensional latent space to a 16-dimensional output space for the standard DGP model, evaluated on a grid. These contour plots demonstrate clearly how expressive nonlinear behaviour is generated. It is possible to consider each of the surfaces as representing a *vibrational mode* of the sounding object. When a common force acts on these surfaces (the red line representing the 2-dimensional latent function), the vibrational modes are activated accordingly.

In observing Figure 6.2 we can see that the mapping is highly nonlinear with significant variation in regions where we have not previously observed data. We hypothesise that these drastic jumps can lead to unpredictable and unreliable interpolation in this space, which motivates a model which imposes additional constraints on the mapping in the second GP layer.

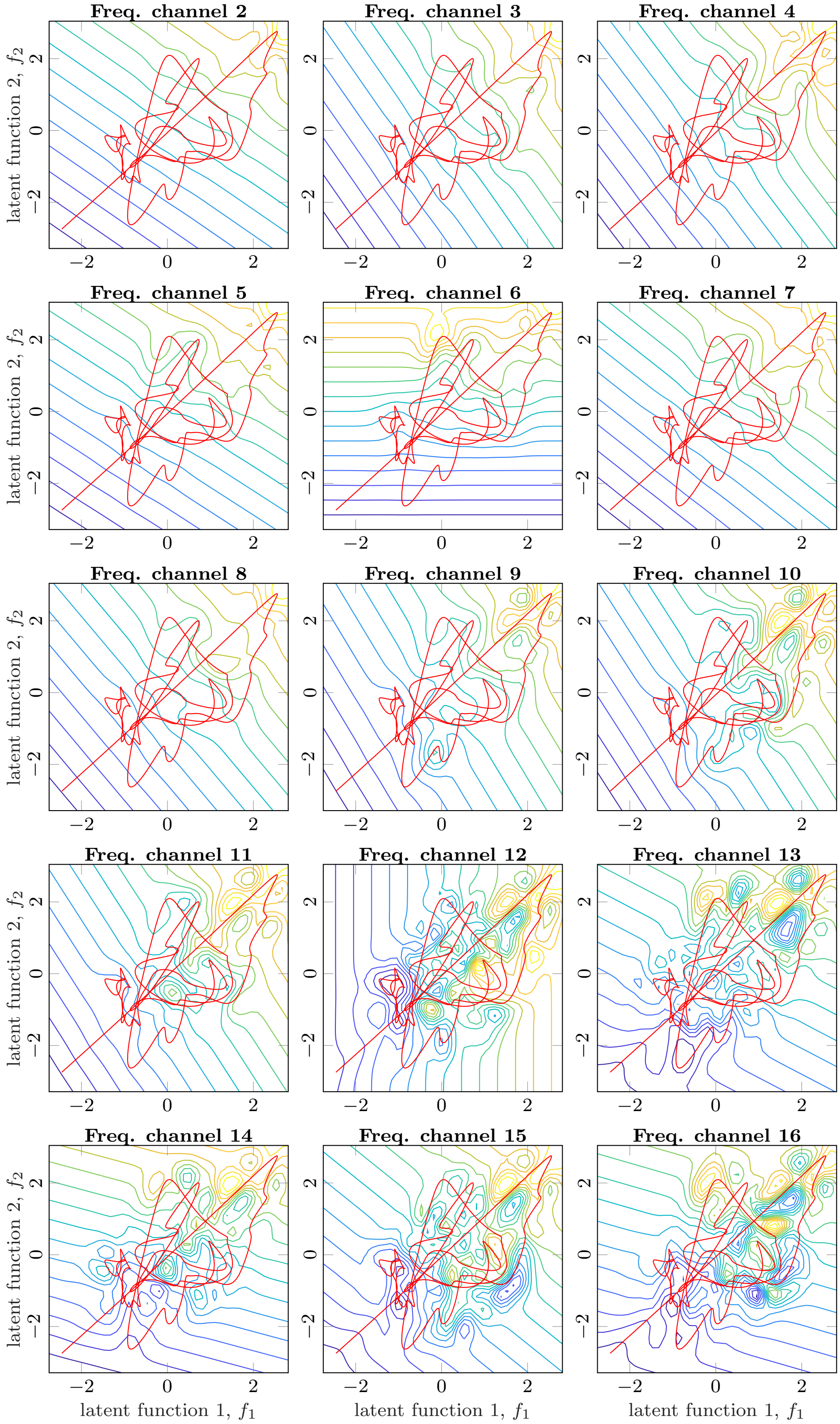


Figure 6.2: Deep GP contours - the predictive mean of 15 outputs plotted on a grid. The red line is the sample path, shared among outputs, which activates the modes of vibration of the sounding object. The 16 envelopes plotted through time can be seen in Figure 6.1.

6.4 Monotonic Deep Gaussian Processes

As a way of constraining the multi-layer GP in order to prevent undesirable behaviour, such as jumps in amplitude not observed in the data, we propose a deep GP model which incorporates monotonicity information. An additional motivation for such an approach is the pathological behaviour typically exhibited in deep models due to the non-injective mappings generated by unconstrained nonlinearities, as discussed in Salimbeni and Deisenroth (2017) and Duvenaud et al. (2014).

It is also insightful to consider the modelling assumptions implied by standard linear models such as NMF. The nonnegativity constraint on the spectral mapping (in addition to the temporal mapping) leads to a positively linear, and hence *positively monotonic*, function mapping from the latents to the outputs. Noticing this fact allows us to consider that positively monotonic nonlinear mappings could achieve a good middle ground between the interpretability / reliability of linear models and the expressivity of their nonlinear counterparts.

We follow Riihimäki and Vehtari (2010) by using virtual derivative observations to impose monotonicity on g, extending their method to the multi-layer case. The central idea is that Gaussian processes are closed under linear operations and partial differentiation is a linear operator. This implies that the joint distribution of a Gaussian process and its partial derivatives is Gaussian distributed if the covariance function of the GP is sufficiently smooth. Now let $g_k^{(j)}$ denote the j^{th} partial derivative at some position $f(t_k)$ and let $m_k \in \{-1, 1\}$ denote the desired monotonicity direction, then we can induce monotonicity at input position $f(t_k)$ using the following likelihood

$$p(m_k \,|\, g_k^{(j)}) = \Phi\left(\frac{1}{\nu} m_k g_k^{(j)}\right), \tag{6.7}$$

where Φ is the probit likelihood function and $\nu > 0$ is a parameter controlling the strictness of the monotonicity constraint. For $m_k = 1$, this likelihood ensures that the posterior distribution of g only contains significant probability mass on functions where $g_k^{(j)} \geq 0$ (See Riihimäki and Vehtari (2010) for a more detailed explanation, and for the derivative calculations for the exponentiated quadratic kernel). The likelihood for the *virtual derivative observation*, $p(m_k \,|\, g_k^{(j)})$, approaches an indicator function for $g_k^{(j)} > 0$ as $\nu \to 0$.

The joint model becomes, letting $\mathbf{m} = (m_1, \ldots, m_T)^\top$,

$$\mathbf{a}_{1:D} \mid f, g \sim \prod_{k=1}^{T} \mathrm{N}\left(g(f(t_k)), \sigma_y^2\right), \tag{6.8a}$$

$$\mathbf{m} \mid g_k^{(j)} \sim \prod_{k=1}^{T} \Phi\left(\frac{1}{\nu} m_k g_k^{(j)}\right), \tag{6.8b}$$

$$g, g^{(1)}, \ldots, g^{(J)} \sim \mathrm{GP}(0, \kappa_{g'}), \tag{6.8c}$$

$$f \sim \mathrm{GP}(0, \kappa_f), \tag{6.8d}$$

where $\kappa_{g'}$ is the joint covariance function for the function g and its partial derivatives $g^{(1)}, \ldots, g^{(J)}$ and $\{(f(t_k), m_k)\}_{k=1}^{T}$ denotes the set of virtual derivative locations and observations.

The derivative observations now contribute to the marginal likelihood calculations, and therefore increasing the number of derivative observations leads to not only a more dense set of locations at which to encourage monotonicity, but also to a larger contribution in the marginal likelihood. Hence choosing the number of observation points amounts to another control over the monotonicity strictness. Note that no changes to the monotonicity information model, Eq. (6.8b), are required to additionally enforce monotonicity in the DGP-Multi model.

Implementation details Further details that required consideration include the need to minibatch the derivative points during optimisation (i.e. only consider a random subset of points during each iteration), especially when using multiple latent dimensions where the size of the latent space (and hence the number of points needed to cover the space) grows exponentially with the dimensionality.

We found in practice that the derivative locations needed to be spaced on a grid in the latent space, rather than on a temporal grid and projected through the first GP layer, to prevent the GP learning to project the points onto an already-monotonic region of the second layer. We additionally included the derivative contribution from the mean function, to extend the model capacity beyond the zero-mean case. During implementation, we evaluated the monotonicity predictions using finite difference methods.

6.5 Missing Data Synthesis with Deep Gaussian Processes

In order to evaluate our proposed methods, in terms of their ability to model the detailed behaviour that occurs in the magnitude spectrogram of an audio signal, we ran a missing data synthesis experiment on two datasets of natural sound: 20 recordings of human speech, and 20 recordings of sound textures.[2] Each recording is approximately 2 seconds in duration. We are interested in their ability to interpolate and generate significant portions of audio, but we must keep the gaps small enough such that ground truth comparisons are at least partially meaningful. Hence we used missing data gaps of 50ms, which amounts to a few frames of data in a typical time-frequency representation, depending on the frame size.

We compare all four versions of the deep GP model (DGP, DGP-multi and their monotonic counterparts mDGP, mDGP-multi) against one another and against tNMF, which also has generative capabilities. Table 6.2 shows the RMSE and SNR of these methods with respect to the ground truth spectrogram. We observe that for the speech data the DGP model estimates the missing data most effectively, but for the sound textures dataset tNMF outperforms the nonlinear models.

However, since the missing data gap is potentially much larger than the lengthscales of the signal, a simple RMSE or SNR metric with respect to the ground truth is insufficient to compare performance. Therefore we also use the perceptual statistics discussed in section 2.2 as a performance metric. Table 6.3 shows the mean across all 10 perceptual statistics proposed by McDermott et al. (2009) that relate to amplitude envelopes. These statistics show that, perceptually, the data generated in the gap by the nonlinear models far outperform those from tNMF.

Whilst the standard deep GP model generally outperforms the others, in some situations it was beneficial to have multiple independent latent functions. Our hypothesis that constraining the second layer mapping would be of benefit turned out to be false in this experimental setting. The expressivity trade-off is too large, and hence it is outperformed in terms of ground truth comparison (Table 6.2) and perceptual measures

[2] 10 speech and 10 texture recordings were obtained from https://freesound.org/. 10 speech recordings were chosen at random from the TIMIT dataset, and 10 texture recordings were chosen from the Sound Texture repository: http://mcdermottlab.mit.edu/downloads.html.

		DGP	DGP-multi	mDGP	mDGP-multi	tNMF
Textures	RMSE	0.104	0.109	0.137	0.137	**0.068**
	SNR	4.287	4.130	3.114	2.446	**8.394**
Speech	RMSE	**0.087**	0.091	0.091	0.131	0.108
	SNR	**5.887**	5.717	5.602	2.791	4.621

Table 6.2: Mean performance of spectrogram missing data synthesis for each model type. mDGP is the DGP model with monotonicity constraints.

		DGP	DGP-multi	mDGP	mDGP-multi	tNMF
Textures	SNR	**16.463**	15.320	11.113	9.761	6.260
Speech	SNR	**14.860**	13.896	12.251	10.683	5.192

Table 6.3: Mean value of all texture statistic SNRs when performing missing data synthesis. mDGP is the DGP model with monotonicity constraints.

(Table 6.3). In addition to this, the monotonic model tended to learn shorter lengthscales, since more detail had to be captured in the latent functions rather than the constrained spectral mapping, which resulted in higher uncertainty in the missing segments and hence poorer performance relative to the ground truth.

Figure 6.3 shows a comparison of the individual texture statistics for the two datasets. We see a consistent and significant benefit to the deep GP models in terms of synthesising realistic spectrogram gaps. These plots suggest that the monotonicity constraints act as a compromise between the linear tNMF model and the nonlinear deep GP models, *trading-off linearity with expressivity*, which could motivate their use in other situations where unconstrained nonlinearity is not desirable.

Figure 6.4 demonstrates the application of the DGP model to missing data synthesis for a sound texture example, a recording of rain falling on a hard surface, and to female speech. The speech data is clearly nonstationary but exhibits somewhat smooth variation over time (there is low uncertainty in the missing regions), and the model is capable of *interpolating* between the observed data. However the rain sound, whilst stationary, is fast varying (higher uncertainty in the missing regions), hence samples from the model can be interpreted as *generating* plausible data in the gaps.

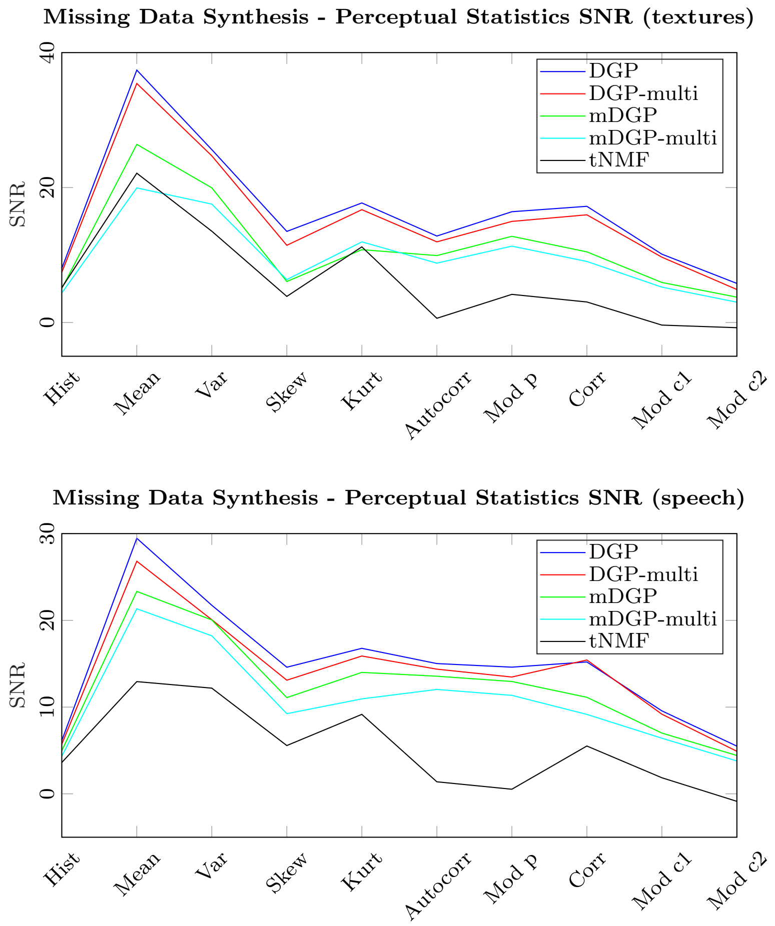


Figure 6.3: Signal to noise ratio of the perceptual statistics of the reconstructed signal vs. the true signal in a missing data synthesis task. The envelope statistics are, from left to right, the histogram, mean, variance, skew, kurtosis, autocorrelation, the modulation power, the across-band correlation, within-band modulator correlation, and the across-band modulator correlation.

6.6 Conclusion

In this chapter we investigated whether the data generating mechanism for audio spectrograms can be learnt using multi-layer Gaussian processes. Implementing such an approach involved modifications to the deep GP model to enforce positivity of the outputs, and can be seen as a nonlinear extension of classical methods such as temporal NMF.

Our examples and empirical analysis suggest that such an approach is effective in a missing data synthesis task, capable of generating plausible amplitude data as evaluated by the perceptual metrics discussed in section 2.2. Drawing analogies with classical signal processing methods, we proposed further modifications based on monotonicity and multiple

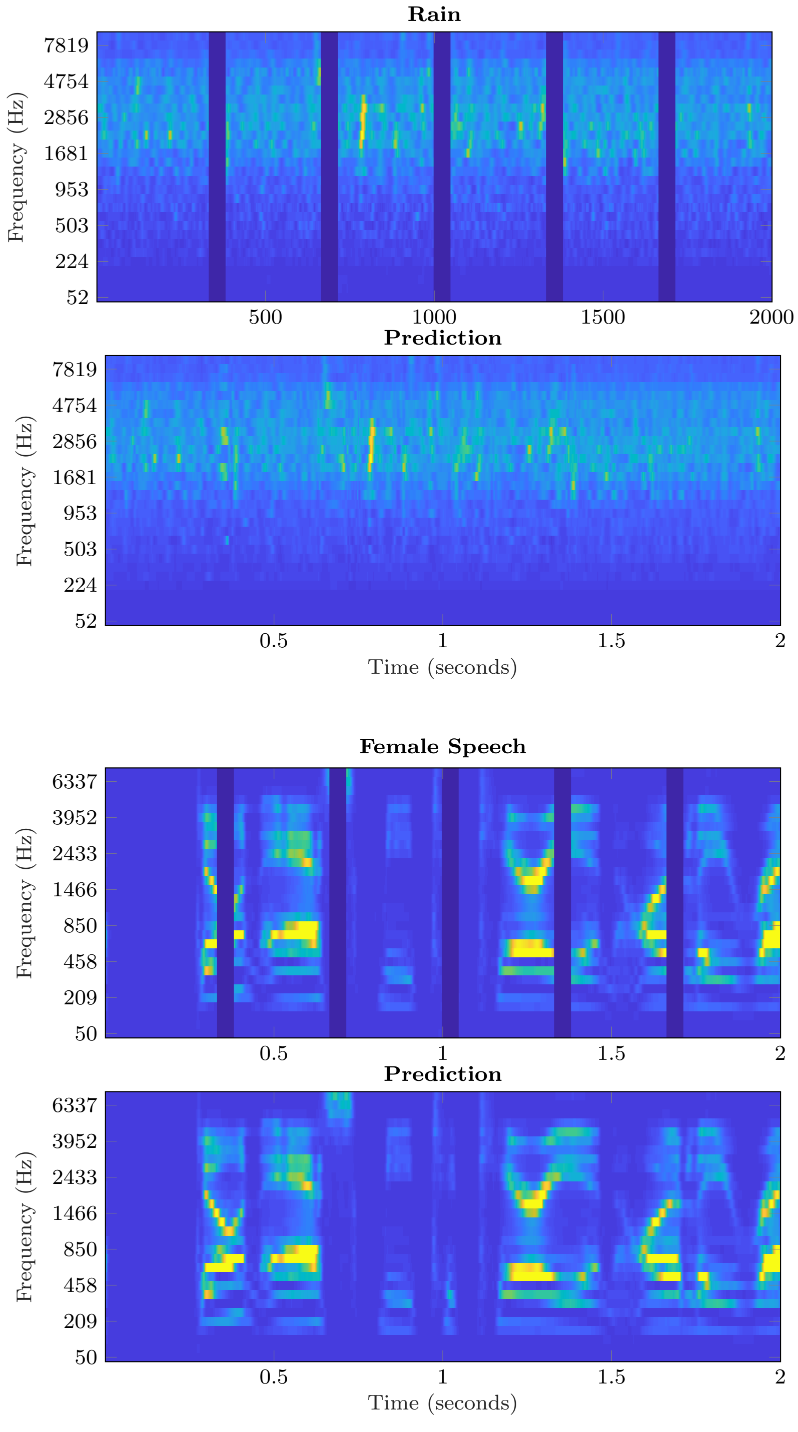


Figure 6.4: Missing data synthesis applied to a sound texture recording (rain, **top half**) and female speech (**bottom half**). The deep GP is trained on the spectrogram with 50ms segments removed (dark blue vertical bars), and is able to generate / interpolate authentic data in the gaps. The speech signal is smooth but nonstationary, whereas the texture signal is fast-varying but stationary, but the model is able to make good predictions in both cases.

lengthscale components, but these additions did not result in significant or consistent gains in performance.

The success of the unconstrained version of the deep GP suggests that a very powerful way to generate data and make predictions over long time periods is to model smooth latent functions being mapped through fast-varying (non-monotonic) spectral mappings.

However, drawing general conclusions from our experiments is challenging due to the significant number of moving parts involved. We must select the number of frequency components, the number of layers, the number of latent components, inducing points and derivative locations. Furthermore, the most effective way to perform inference in these settings is still an open research question (state of the art work on deep GPs has progressed since the completion of this work, see Havasi et al. (2018) and Salimbeni et al. (2019)), and the variational inducing point method utilised here is not a natural fit for time series.

These issues suggest that further research should focus first on developing scalable inference methods well suited to temporal data with complex and nonstationary structure. However, the promising results we present do motivate a return to these ideas when such methods are available, since it is now clear that linear models for audio data are insufficient, and improved performance in tasks such as missing data synthesis is indeed possible.

Finally, our evaluation using perceptual statistics allows us to consider the extent to which our Gaussian process models encode the important characteristics of sound. Our results suggest that these models do not fully capture the texture statistics proposed by McDermott et al. (2009), particularly the across-band and within-band modulation correlation metrics. This motivates two possible directions for future work: the explicit incorporation of such statistics into a probabilistic model for sound, or an increase in depth and scale of the models used. Whilst the latter is certainly more in line with current trends in the machine learning community, the former may lead to scalable methods with stronger ties to traditional signal processing.

Chapter 7

Conclusion

7.1 Summary and Discussion

This thesis can be viewed in two parts. In the first half, chapter 3 and chapter 4, we focus on *inference* in time series models that apply to sound, and we offer a new perspective on Gaussian process models for time-frequency analysis. We show how state of the art inference techniques can be formulated in this problem domain, in a way that is suited to long temporal data with complex structure. In doing so, we advance the state of the art of GP regression on audio signals, whilst also providing new insights into the links between probabilistic machine learning and signal processing.

In the second half of the thesis, chapter 5 and chapter 6, we turn our attention to *modelling* of audio spectrograms. We propose two novel models, one based on physical assumptions about sound, the other based on learning flexible nonlinear spectral mappings, both capable of adapting to data. In each case we demonstrated how the approach can lead to significant improvements on generative tasks. However, we also came across issues in the scalability and reliability of inference in both cases, and identified areas for improvement and future work that could pave the way for real advances in the practicality of such methods.

Viewed as a whole, we have demonstrated the effectiveness of GP models for audio, and implemented new and improved inference schemes for these models, before going on to demonstrate how they can be applied to signal processing tasks. Our results, and new insights on existing models, motivate further research around probabilistic treatment of audio, and bring us closer to the significant goal of being able to place a probability distribution over an audio signal in an effective and tractable manner.

Our main conclusions can be summarised as follows:

- Spectral mixture Gaussian processes can be seen as a probabilistic extension of filter banks, but allow for natural modification of the assumptions embedded in the filter coefficients, allow us to characterise uncertainty and to make predictions about unseen data points.

- Joint NMF and time-frequency analysis can be formulated as a nonstationary spectral mixture GP, and inference can (and arguably must) be performed via state of the art methods in the Kalman filter setting, which is a good fit to time series data.

- We show that expectation propagation can be performed in such a setting.

- We show that GP regression can be constructed in a way that is suited to audio signals, despite the poor temporal scaling usually associated with GP methods. However further research in this area is required.

- Physical assumptions about audio signals can be incorporated into probabilistic models for audio spectrograms, and this allows for effective dimensionality reduction and generative capabilities, as demonstrated by listening tests with human participants.

- These listening tests suggest that exponential decay of amplitude envelopes is an important component of auditory perception.

- Deep Gaussian processes can be seen as a nonlinear extension to (temporal) matrix factorisation.

- Therefore it follows that deep GPs can be a useful model for audio spectrograms, and improve performance on missing data synthesis for a range of signal types, based on perceptual characteristics.

- Monotonicity information can be incorporated into deep GP models, and provide a means by which to trade off interpretability with expressivity, but do not improve performance on missing data synthesis.

7.2 Future Work

Throughout this thesis we have identified some keys areas for improvement required to aid the application of Gaussian processes to signal processing tasks.

Whilst recent advances in scalable inference are encouraging, it is possible that batch processing of the entire data will always remain problematic in an application domain where we typically acquire new data at a rate of many thousands of samples per second, for a possibly unbounded duration. For this reason, a key area for future work should be in improving and formalising online learning methods, which adapt over time and don't require permanent storage of large covariance matrices that grow with the length of the signal.

In addition, the inducing points method used in chapter 6 is not a good conceptual fit for time series, where the data grows in an unbounded fashion. However, as we demonstrated, deep models can provide great benefits, and so an exciting area for investigation is in adapting the state space SDE approach to deep models. The application of these models directly in the waveform domain would likely be beneficial, albeit computationally challenging.

As we discussed in chapter 4, nonstationary behaviour is an important feature of real-world signals that should be captured during analysis. One way to advance nonstationary models would be to extend and generalise our proposed methods such that all the kernel hyperparameters can be time-varying. Fully probabilistic inference in this setting is still an unsolved problem, and parameter learning in existing approaches is somewhat unstable. These issues must be addressed in order for such methods to become practical and be applicable to real-world problems.

Another way to improve modelling flexibility is to learn the kernel, or its spectral density, with another GP. This can be thought of as an alternative to the deep models used in this thesis in which the kernel is learnt in a nonparametric fashion rather than warping the inputs to a parametric kernel. It remains to be seen which of these approaches will prove most fruitful.

Bibliography

Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Courier Corporation, 1965.

Amir Adler, Valentin Emiya, Maria G Jafari, Michael Elad, Rémi Gribonval, and Mark D Plumbley. Audio inpainting. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(3):922–932, 2012.

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.

Pablo A. Alvarado and Dan Stowell. Efficient learning of harmonic priors for pitch detection in polyphonic music. *arXiv preprint arXiv:1705.07104*, 2017.

Pablo A Alvarado, Mauricio A Álvarez, and Dan Stowell. Sparse Gaussian process audio source separation using spectrum priors in the time-domain. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019.

Mauricio A. Alvarez, David Luengo, and Neil D Lawrence. Latent force models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 12, pages 9–16, 2009.

Mauricio A Alvarez, Lorenzo Rosasco, Neil D Lawrence, et al. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.

Mauricio A. Alvarez, David Luengo, and Neil D. Lawrence. Linear latent force models using gaussian processes. *IEEE Transactions on pattern analysis and machine intelligence*, 35(11):2693–2705, 2013.

Joseph Antognini, Matt Hoffman, and Ron J Weiss. Synthesizing diverse, high-quality audio textures. *arXiv preprint arXiv:1806.08002*, 2018.

Roland Badeau and Mark D Plumbley. Multichannel high-resolution nmf for modeling convolutive mixtures of non-stationary signals in the time-frequency domain. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(11):1670–1680, 2014.

Yaakov Bar-Shalom, Xiao-Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley-Interscience, New York, 2001.

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*, 114(2):1095–1107, 2003.

Nancy Bertin, Roland Badeau, and Emmanuel Vincent. Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):538–549, 2010.

Larry G. Bretthorst. *Bayesian spectrum analysis and parameter estimation*, volume 48. Springer Science & Business Media, 2013.

Thang Bui, Daniel Hernández-Lobato, Jose Hernandez-Lobato, Yingzhen Li, and Richard Turner. Deep gaussian processes for regression using approximate expectation propagation. In *International Conference on Machine Learning*, pages 1472–1481, 2016.

Thang D Bui, Josiah Yan, and Richard E Turner. A unifying framework for gaussian process pseudo-point approximations using power expectation propagation. *The Journal of Machine Learning Research*, 18(1):3649–3720, 2017.

Thang Duc Bui. *Efficient Deterministic Approximate Bayesian Inference for Gaussian Process models*. PhD thesis, University of Cambridge, 2018.

Marcelo Caetano and Xavier Rodet. Musical instrument sound morphing guided by perceptually motivated features. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(8):1666–1675, 2013.

Ali Taylan Cemgil and Simon J. Godsill. Probabilistic phase vocoder and its application to interpolation of missing values in audio signals. In *European Signal Processing Conference (EUSIPCO)*, pages 1–4, 2005.

Leon Cohen. *Time-frequency Analysis: Theory and Applications.* Prentice Hall, USA, 1995.

Perry R. Cook. Physically informed sonic modeling (PhISM): Synthesis of percussive sounds. *Computer Music Journal*, 21(3):38–49, 1997.

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

Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. *Neural computation*, 14(3):641–668, 2002.

Andreas Damianou and Neil Lawrence. Deep gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.

Nicolas Durrande, Vincent Adam, Lucas Bordeaux, Stefanos Eleftheriadis, and James Hensman. Banded matrix operators for gaussian markov models in the automatic differentiation era. *arXiv preprint arXiv:1902.10078*, 2019.

David Duvenaud, Oren Rippel, Ryan Adams, and Zoubin Ghahramani. Avoiding pathologies in very deep networks. In *Artificial Intelligence and Statistics*, pages 202–210, 2014.

Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. Neural audio synthesis of musical notes with wavenet autoencoders. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1068–1077. JMLR, 2017.

Yariv Ephraim and David Malah. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on acoustics, speech, and signal processing*, 32(6):1109–1121, 1984.

Ángel F García-Fernández, Filip Tronarp, and Simo Sarkka. Gaussian process classification using posterior linearisation. *IEEE Signal Processing Letters*, 2019.

Andrew Gelman, Hal S Stern, John B Carlin, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

Mark N Gibbs and David JC MacKay. Variational gaussian process classifiers. *IEEE Transactions on Neural Networks*, 11(6):1458–1464, 2000.

Bradford W. Gillespie and Les E. Atlas. Optimizing time-frequency kernels for classification. *IEEE Transactions on Signal Processing*, 49 (3):485–496, 2001.

Brian R Glasberg and Brian CJ Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing research*, 47(1-2):103–138, 1990.

Philip A Gomersall, Richard E Turner, David M Baguley, John M Deeks, Hedwig E Gockel, and Robert P Carlyon. Perception of stochastic envelopes by normal-hearing and cochlear-implant listeners. *Hearing research*, 333:8–24, 2016.

Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

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

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)*, pages 903–910, 2012.

Marton Havasi, José Miguel Hernández-Lobato, and Juan José Murillo-Fuentes. Inference in deep gaussian processes using stochastic gradient hamiltonian monte carlo. In *Advances in Neural Information Processing Systems*, pages 7517–7527, 2018.

James Hensman, Nicolò Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence (UAI)*, pages 282–290. AUAI Press, 2013.

James Hensman, Nicolas Durrande, and Arno Solin. Variational Fourier features for Gaussian processes. *Journal of Machine Learning Research (JMLR)*, 18(151):1–52, 2018.

Andrew H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, New York, 1970.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

Alexander JE Kell, Daniel LK Yamins, Erica N Shook, Sam V Norman-Haignere, and Josh H McDermott. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644, 2018.

Michael Klingbeil. Software for spectral analysis, editing, and synthesis. In *International Computer Music Conference (ICMC)*, 2005.

Juho Kokkala, Arno Solin, and Simo Särkkä. Sigma-point filtering and smoothing based parameter estimation in nonlinear dynamic systems. *Journal of Advances in Information Fusion*, 11(1):15–30, 2016. ISSN 15576418.

Karl Krauth, Edwin V Bonilla, Kurt Cutajar, and Maurizio Filippone. AutoGP: Exploring the capabilities and limitations of Gaussian process models. In *Uncertainty in Artificial Intelligence (UAI).* AUAI Press, 2017.

Peter Lancaster and Leiba Rodman. *Algebraic riccati equations.* Clarendon press, 1995.

Alan Laub. A schur method for solving algebraic riccati equations. *IEEE Transactions on automatic control*, 24(6):913–921, 1979.

Neil Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of machine learning research*, 6(Nov):1783–1816, 2005.

Miguel Lázaro-Gredilla, Joaquin Quiñonero-Candela, Carl E. Rasmussen, and Aníbal R. Figueiras-Vidal. Sparse spectrum Gaussian process regression. *Journal of Machine Learning Research (JMLR)*, 11:1865–1881, June 2010.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

Antoine Liutkus, Roland Badeau, and Gäel Richard. Gaussian processes for underdetermined source separation. *IEEE Transactions on Signal Processing*, 59(7):3155–3167, 2011.

Paul Magron and Tuomas Virtanen. Complex ISNMF: a phase-aware model for monaural audio source separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(1):20–31, 2019.

Bertil Matérn. *Spatial variation: Stochastic models and their applications to some problems in forest surveys and other sampling investigations.* Meddelanden från statens skogsforskningsinstitut, 1960.

Peter S. Maybeck. *Stochastic Models, Estimation and Control*, volume 2. Academic Press, New York, NY, 1982.

Isambi S Mbalawata, Simo Särkkä, and Heikki Haario. Parameter estimation in stochastic differential equations with markov chain monte carlo and non-linear kalman filtering. *Computational Statistics*, 28(3):1195–1223, 2013.

Robert McAulay and Thomas Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):744–754, 1986.

Josh H McDermott and Eero P Simoncelli. Sound texture perception via statistics of the auditory periphery: Evidence from sound synthesis. *Neuron*, 71(5):926–940, 2011.

Josh H McDermott, Andrew J Oxenham, and Eero P Simoncelli. Sound texture synthesis via filter statistics. In *2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 297–300. IEEE, 2009.

Josh H McDermott, Michael Schemitsch, and Eero P Simoncelli. Summary statistics in auditory perception. *Nature Neuroscience*, 16(4): 493, 2013.

J. McNamee and F. Stenger. Construction of fully symmetric numerical integration formulas of fully symmetric numerical integration formulas. *Numerische Mathematik*, 10(4):327–344, Nov 1967.

Thomas Minka. Power EP. *Dep. Statistics, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep*, 2004.

Thomas Minka. Divergence measures and message passing. Technical report, 2005.

Thomas Minka and John Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 352–359. Morgan Kaufmann Publishers Inc., 2002.

Thomas P Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369, 2001.

Duy Nguyen-Tuong, Jan R Peters, and Matthias Seeger. Local gaussian process regression for real time online model learning. In *Advances in Neural Information Processing Systems*, pages 1193–1200, 2009.

Hannes Nickisch, Arno Solin, and Alexander Grigorievskiy. State space Gaussian processes with non-Gaussian likelihood. In *International Conference on Machine Learning (ICML)*, volume 80 of *PMLR*, pages 3789–3798, 2018.

Yuan Qi, Thomas P. Minka, and Rosalind W. Picara. Bayesian spectrum estimation of unevenly sampled nonstationary data. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages II–1473. IEEE, 2002.

Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research (JMLR)*, 6(Dec):1939–1959, 2005.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006.

Sami Remes, Markus Heinonen, and Samuel Kaski. Non-stationary spectral kernels. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 4642–4651. Curran Associates, Inc., 2017.

Jaakko Riihimäki and Aki Vehtari. Gaussian processes with monotonicity information. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 645–652, 2010.

Antony W. Rix, John G. Beerends, Michael P. Hollier, and Andries P. Hekstra. Perceptual evaluation of speech quality (PESQ)–a new method for speech quality assessment of telephone networks and codecs. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 749–752. IEEE, 2001.

Yunus Saatçi. *Scalable Inference for Structured Gaussian Process Models.* PhD thesis, University of Cambridge, UK, 2012.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep gaussian processes. In *Advances in Neural Information Processing Systems*, pages 4588–4599, 2017.

Hugh Salimbeni, Vincent Dutordoir, James Hensman, and Marc Deisenroth. Deep gaussian processes with importance-weighted variational inference. In *International Conference on Machine Learning*, 2019.

Simo Särkkä. *Bayesian Filtering and Smoothing.* Cambridge University Press, 2013.

Simo Särkkä and Robert Piché. On convergence and accuracy of state-space approximations of squared exponential covariance functions. In *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2014.

Simo Särkkä and Arno Solin. *Applied Stochastic Differential Equations.* Cambridge University Press, 2019.

Matthias Seeger. Expectation propagation for exponential families. Technical report, 2005.

Ervin Sejdić, Igor Djurović, and Jin Jiang. Time–frequency feature representation using energy concentration: An overview of recent advances. *Digital Signal Processing*, 19(1):153–183, 2009.

Robert V Shannon, Fan-Gang Zeng, Vivek Kamath, John Wygonski, and Michael Ekelid. Speech recognition with primarily temporal cues. *Science*, 270(5234):303–304, 1995.

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

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1257–1264. Curran Associates, Inc., 2006.

Arno Solin. *Stochastic Differential Equation Methods for Spatio-Temporal Gaussian Process Regression.* Doctoral dissertation, Aalto University, Helsinki, Finland, 2016.

Arno Solin and Simo Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *arXiv preprint arXiv:1401.5508*, 2014a.

Arno Solin and Simo Särkkä. Explicit link between periodic covariance functions and state space models. In *Artificial Intelligence and Statistics*, pages 904–912, 2014b.

Arno Solin, James Hensman, and Richard E Turner. Infinite-horizon Gaussian processes. In *Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 3490–3499. Curran Associates, Inc., 2018.

Michalis K Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 5 of *PMLR*, pages 567–574, 2009.

Felipe Tobar, Thang D Bui, and Richard E Turner. Learning stationary time series using gaussian processes with nonparametric kernels. In *Advances in Neural Information Processing Systems*, pages 3501–3509, 2015.

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*, pages 83–86, 1999.

Filip Tronarp, Ángel F García-Fernández, and Simo Särkkä. Iterative filtering and smoothing in nonlinear and non-gaussian systems using conditional moments. *IEEE Signal Processing Letters*, 25(3):408–412, 2018.

Richard E. Turner. *Statistical Models for Natural Sounds.* PhD thesis, University College London, UK, 2010.

Richard E Turner and Maneesh Sahani. Demodulation as probabilistic inference. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(8):2398–2411, 2011.

Richard E Turner and Maneesh Sahani. Time-frequency analysis as probabilistic inference. *IEEE Transactions on Signal Processing*, 62 (23):6171–6183, 2014.

George E Uhlenbeck and Leonard S Ornstein. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016.

Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pages 153–158. Ieee, 2000.

Christopher KI Williams and David Barber. Bayesian classification with gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

Andrew Gordon Wilson and Hannes Nickisch. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In *International Conference on Machine Learning (ICML)*, volume 37 of *PMLR*, pages 1775–1784, 2015.

Zhengyou Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76, 1997.

Jingang Zhong and Yu Huang. Time-frequency representation based on an adaptive short-time fourier transform. *IEEE Transactions on Signal Processing*, 58(10):5118–5128, 2010.