Non-stationary, online variational Bayesian learning, with circular variables

J Christmas[1,*]

University of Exeter, Exeter, UK

Abstract

We introduce an online variational Bayesian model for tracking changes in a non-stationary, multivariate, temporal signal, using as an example the changing frequency and amplitude of a noisy sinusoidal signal over time. The model incorporates each observation as it arrives and then discards it, and places priors over precision hyperparameters to ensure that (i) the posterior probability distributions do not become overly tight, which would impede its ability to recognise and track changes, and (ii) no values in the system are able to continuously increase and hence exceed the numerical representation of the programming language. It is thus able to perform truly online processing for an infinitely long set of observations. Only a single round of updates in the variational Bayesian scheme per observation is used, and the complexity of the algorithm is constant in time. The proposed method is demonstrated on a large number of synthetic datasets, comparing the results from the full model (with precision hyperparameters as variables with priors) with those from the base model where the precision hyperparameters are fixed values. The full model is also demonstrated on a set of real climate data.

Keywords: online learning/processing, variational methods, Bayes procedures.

1. Introduction

When noisy time series data are generated, perhaps by a sensor of some sort, there is a choice to be made between storing the original, raw data for later analysis, or processing them on the fly and recording some more useful or convenient summary information in place of the raw data. Choosing the latter, an obvious candidate for maintaining summary information is online Bayesian sequential learning, where the current state of a Bayesian model summarises the raw data, each observation is incorporated into the model as it arrives, and the posterior probability distributions after one observation become the priors for

*Corresponding author

Email address: J.T.Christmas@exeter.ac.uk (J Christmas)

the next one. Often though, the integrals required for exact Bayesian inference are intractable, so some sort of approximation is required.

We present a simple such system to which we choose to apply the variational Bayesian (VB) approximation method, and in the process of developing the model, overcome a number of obstacles and challenges that this method introduces. To facilitate the description of these challenges, and how we overcome them, we start by describing an illustrative system and our chosen mathematical representation of it. In this example we assume the noise to be Gaussian distributed, but in a real-world problem we might choose to assume it to be, for example, Student-t distributed to make it robust to outliers [1].

In our example system, a sensor buoy is floating at the surface of the sea, continuously measuring the vertical component of the motion of the waves at a fixed sampling rate, f_s, for an indefinite period of time. For the purposes of this paper, assume that the water's surface is perturbed by a single sinusoidal wave of unknown amplitude, A, and angular frequency, ω, onto which the sensor superimposes white noise of precision (inverse variance) λ. The sea is not statistically stationary, so over time the amplitude, frequency and noise precision may change slowly and smoothly, and we wish to track those changes. If the buoy's nth observation, y_n, is recorded at time t_n, then our selected model is as follows:

$$y_n = A\cos(\phi + \omega t_n) + \epsilon_n \tag{1}$$

where ϕ is the phase offset at time t_0, and ϵ_n the noise. The ϕ variable allows us to state without loss of generality that $t_0 = 0$, and hence $t_n = n\delta t$, where $\delta t = 1/f_s$ is the fixed time interval between observations. In this context the challenges for VB, and our responses to them, are as follows.

Challenge 1: sufficient processing speed for online learning. In order to maintain the online model, each observation must be incorporated into the model within the sampling interval, δt. While faster than Markov chain Monte Carlo (MCMC), VB is an interative process where the hyperparameters of each posterior are updated in turn until convergence. The number of iterations may be fixed, which at least makes the computation time predictable, or the system may be iterative until some convergence threshold is reached; either case raises an additional challenge of determining either what is the best number of iterations, or what the convergence threshold should be. We avoid the problem simply by not iterating the VB update process; we perform a single "iteration" to incorporate each observation.

Challenge 2: unknown number of observations. In a traditional VB framework, the complete set of observations is available to train the model, and it is assumed to be small enough to fit in the computer's memory during that training process. The model may be trained by sweeping backwards and forwards through the observations multiple times until convergence. In our case the total number of observations is unknown, and may be considered to be infinite, and we wish to retain in memory only the single current observation. In our system, all past observations are represented in the current state of the model, and all future observations have not yet been made.

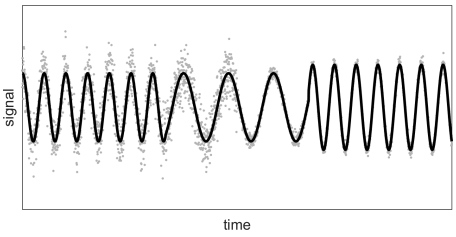


Figure 1: An example of the type of non-stationary sinusoidal signal modelled by the proposed method. The true signal is shown in black, with variations in both angular frequency and amplitude over time. The noisy observations are shown as grey dots, with the noise variance also changing over time.

Challenge 3: a non-stationary target. Our system is statistically non-stationary; the key variables change over time and we want to track the changes, as illustrated in figure 1. If a standard VB model is to capture these changes, and is trained on a complete set of observations, then the model itself must contain extra variables to track that variation. We do not include extra variables, but enable the posterior distributions to change over time so that at any given instant they represent the current state of the system.

Challenge 4: over-tight posteriors. The VB method is known to result in over-tight posterior distributions, i.e. posteriors whose precisions are unrealistically high. During the VB update procedure for a given variable, all other variables (we will assume a full factorisation of the approximate posteriors) are represented by their expectations rather than their full posterior probability distribution, leading to a loss of variance. Once the posteriors have become too tight, the model becomes unable to evolve as the underlying statistics of the system change. We prevent this from happening by placing priors over key precision hyperparameters.

Challenge 5: infinitely increasing values. When a system is running for, effectively, an infinite period of time, any values that are constantly increasing will eventually cause the program to fail when those values exceed the computer's ability to represent them. An obvious example in our system is the parameter t_n, a constantly increasing timestamp. In this instance we can replace t_n with a function of δt (though obviously not with $n\delta t$), but the VB updates for the precision posteriors all exhibit the same problem. We apply the same solution as for challenge 4, i.e. we place priors on the precision variables to limit their growth.

Challenge 6: circular variables. Using [2]'s VB approach requires us to replace the full distribution of a variable with its expected value when reevaluating the posteriors for other variables. Our model includes circular variables whose posteriors turn out to be Generalised von Mises distributions of order 2, which may be bimodal and hence present a problem in representing them with single expectations. We approximate them with unimodal ones, and provide a new approximation for the inverse of the modified Bessel function of the first kind.

In section 2 we start by surveying some of the relevant literature. We follow this, in section 3, with an overview of the circular distributions used in this paper: the von Mises, the use of the wrapped Normal distribution to approximate the sum of two von Mises distributed variables, and the Generalised von Mises distribution of order 2. The model based on (1) is introduced in two parts. Firstly (section 4), we describe the *base model*, providing an alternative arrangement of (1) that better suits the online VB method, specifying the priors, and deriving the VB update scheme. At this point we do not limit the precisions (see challenges 4 and 5 above), but, in section 5, provide some experimental results to show how the base model operates and how the ever-increasing precisions affect the learning process. Secondly (section 6), we describe the methods for limiting these ever-increasing precisions (the *full model*) and, in section 7 show how the updated model operates and how its performance differs from the base model. Conclusions are drawn in section 9.

2. Background

To avoid the computational expense of Markov chain Monte Carlo (MCMC) methods, here we use the variational Bayesian technique for finding approximations to the posterior distribution [3, 4, 5, 2]. This method minimises the Kullback-Leibler divergence between the approximate and actual posterior distributions to determine the optimal hyperparameter values for the approximations; for tutorials see [6], and [7, chapter 10]. Attias [2] (see also [8, 9]) exploits the factorisation of the posterior (in this paper we assume a full factorisation) to find a general expression for the minimisation in a mean-field sense. If conjugate priors are chosen for each variable, the approximate posteriors turn out to have the same functional form as the priors [2, 10], and the variational approximations may thus be found by evaluating each approximate posterior in turn. The hyperparameters of the posterior distribution for one variable will generally depend upon the hyperparameters for other variables, so the parameters for each variable are evaluated iteratively until convergence. [10] show that this scheme converges to a local minimum of the Kullback-Leibler divergence.

Generally the VB algorithms operate in batch mode, that is, they are trained by traversing the complete set of observations, often multiple times, to give the sought-after posterior distributions for each variable (e.g. [1, 11]). This approach is fine for the retrospective processing of datasets that are small enough to fit into the computer's memory, but is not suitable for online processing, or for very large (possible infinite) datasets, or for datasets whose final size is unknown.

The Bayesian approach of combining priors with an observation to form posteriors, which in turn become the priors for the next observation, naturally lends itself to an online system, particularly when, as mentioned above, conjugate priors lead to the posteriors having the same functional form as those priors [2, 10]. Now we may maintain a model of the current state and update it with each observation in turn, keeping only the current observation in memory.

[12] consider that VB applied in this way may well get stuck in a local optimum from which it cannot escape. They split the variables in a time series model into *local* (those that are time dependent) and *global* (not time dependent), updating the global variables once per observation, and iterating over the local ones multiple times until convergence. They control the period of the model's memory by including an explicit, tunable *forgetting* method, that demotes the terms of the likelihood that correspond to the oldest observations. They suggest that incorporating observations one at a time might be extreme and that processing the data in small batches might be an effective compromise.

A form of batching is used by [13], who constrain the overall problem by assuming that the time series is piecewise stationary, and use Monte Carlo techniques to model the unknown size and number of segments. [14] also use batching, and explicitly model temporal changes in the system parameters using adaptive forgetting rates.

The iterative coordinate ascent process whereby each approximate posterior is updated in turn while all other posteriors are held fixed, can make VB slow to converge to its final solution. [15, 16] replace it with a stochastic gradient-based algorithm for updating the hyperparameters; the latter paper names this method Stochastic Variational Inference (SVI). They still iterate the updates until convergence, but the gradient ascent means that convergence is achieved in fewer iterations. They also make use of a forgetting function for their key global variable, and update the model using observation *minibatches*. Their model is not dependent on the ordering of the data, so their minibatches are made up of uniformly randomly selected observations. Although each minibatch is incorporated into the model as it is "received", and then released from memory, the algorithm depends on knowing the total number of observations. [17] extend SVI to time series data, but still depend on sampling a sequence of observations to update the model.

Streaming, Distributed, Asynchronous Bayes (SDA-Bayes) [18] moves away from the requirement of knowing the total number of observations, but the distributed, asynchronous nature makes it unsuitable for time series data, especially when the system is not statistically stationary. In addition to sensors, process control is another area where streamed observations need to be modelled, and where the underlying system may not be statistically stationary. In this context, [19] propose an adaptive method based on minibatches, where each new batch is assessed to determine whether it conforms to the statistics of previous data; if it does not, then the model priors are re-initialised. Thus they support non-stationarity, but they also propose a parallel processes strategy which is unsuitable for time series data.

[20] describe an online VB model for time series regression that aims to conform much more closely to the classical Bayesian updating approach, with no requirement to know how many obervations there are. However, while their model is incorporating each single observation in turn, it includes parameters whose values continuously increase (for example, the counter n, which feeds into both $\mu_{q(1/\sigma^2)}$ and $q^*(\sigma^2)$, and the precision $\Sigma^{-1}_{q(\beta)}$ in algorithm 2).

Priors over the precisions (inverse standard deviations) are most often used in the context of automatic relevance determination (ARD) (Mackay 1994, Neal 1995) to "switch off" coefficients for which there is no evidence in the data. Rather than fixing the prior for a precision variable, the parameters of that prior become variables with their own priors. This provides the flexibility for the precision to become very large, constraining a zero-mean Gaussian variable to have a value very close to zero. This result may be used to explicitly estimate model order (e.g. [1]), or, more generally, to promote sparsity in latent variables (e.g. [21]). Other, related methods incorporate this approach, but also apply a penalty function to achieve sparsity (e.g. [22]). However, in each of these cases the underlying system is considered to be statistically stationary and with finite (and relatively small) sets of observations, and thus the ability of the precision variables to become very large (meaning that the model becomes very certain in its estimations) might be considered to be desirable. In a non-stationary system, this over-precision means that the model is unable to react to changes. In this paper, in the "full model", we are trying to achieve the opposite: we want to *relax* the precision so that nonstationarity in the underlying latent variables can be tracked and not locked out by over-precision.

In this paper we stay with the classical Bayesian update approach, and apply it to time series data where the ordering of the observations is critical. Each observation is incorporated into the model one at a time, in time order, and the hyperparameters of the approximate posteriors are updated once for each observation. We ensure that the algorithm does not contain any values that continuously increase over time.

In order to maintain clarity, we use mean-field approximation with a full posterior factorisation and a coordinate ascent update approach.

2.1. Sinusoid frequency estimation

While the system used in this paper has been selected for illustrative purposes only, it is worth very briefly reviewing Bayesian approaches to it.

The selected model, described by equation (1), has been investigated using various Bayesian approaches since [23]. While many of the papers are concerned with spectra and not just single sinusoids (e.g. [24], and [25, 26], who also investigate the optimal number of frequencies to be modelled), there is an evolution of Bayesian approaches to this specific problem: Gibbs sampling [27], RJ-MCMC [28], MAP/ML [29], and a form of expectation propagation [30].

[31, 32] employ a state space model, with Gibbs sampling, to learn the parameters of temporally nonstationary sinusoidal signals, which is closely related to our work, but we avoid using computationally expensive procedures such as Markov chain Monte Carlo.

The circular nature of the angular frequency leads to several methods for incorporating the von Mises (see section 3.1) distribution into a Bayesian context. [30] introduce an auto-regressive generalisation of the von Mises distribution, using expectation propagation to iteratively refine localised Gaussian approximations. While [33] are learning phase rather than frequency, and assume that

the frequency is known, they establish a tractable conjugate system based on a von Mises distribution, as do [34]. Working with frequencies, [26] approximate von Mises distributions with mixtures of von Mises. The main problem for Bayesian inference is the intractability of the Bessel function terms. [35] approximate the Bessel functions with Taylor series expansions.

3. Circular probability distributions

A continuous random variable that describes an angle, i.e. one whose values are constrained to be within a 2π interval, is best represented by a circular probability distribution. There are three such distributions used in this paper: the unimodal von Mises, which is summarised in section 3.1, the wrapped Normal, which is used in the approximation of the sum of two von Mises distributions, covered in section 3.2, and the Generalised von Mises of order 2, which is a generalisation of the von Mises to allow bimodal circular distributions, covered in section 3.3.

3.1. von Mises

The von Mises distribution [36] is a symmetrical, unimodal distribution for a circular variable and is defined as:

$$\mathcal{M}(\theta \,|\, \mu, \kappa) = \frac{1}{2\pi I_0(\kappa)} \exp\big(\kappa \cos(\theta - \mu)\big) \tag{2}$$

where $I_x(\cdot)$ is the modified Bessel function of the first kind, of order x, μ is the mean, and the concentration parameter, $\kappa > 0$, acts like a precision; as $\kappa \to 0$ the distribution tends to a Uniform distribution across the 2π range, while as κ increases the distribution tends to a Gaussian distribution. Useful expectations (denoted by $\langle\cdot\rangle$) for this distribution are as follows:

$$\langle \cos(n\theta) \rangle = \cos(n\mu) \frac{I_n(\kappa)}{I_0(\kappa)} \tag{3}$$

$$\langle \sin(n\theta) \rangle = \sin(n\mu) \frac{I_n(\kappa)}{I_0(\kappa)} \tag{4}$$

$$\langle \cos^2(n\theta) \rangle = \frac{1}{2}\Big(1 + \langle \cos(2n\theta) \rangle\Big) \tag{5}$$

$$\langle \sin^2(n\theta) \rangle = \frac{1}{2}\Big(1 - \langle \cos(2n\theta) \rangle\Big) \tag{6}$$

An alternative expression for the same distribution is

$$\mathcal{M}(\theta \,|\, \alpha, \beta) = \frac{1}{2\pi I_0(\kappa)} \exp\Big(\alpha \cos(\theta) + \beta \sin(\theta)\Big) \tag{7}$$

where

$$\alpha = \kappa \cos(\mu) \qquad\qquad \beta = \kappa \sin(\mu) \tag{8}$$

and, alternatively,

$$\mu = \operatorname{atan}\left(\frac{\beta}{\alpha}\right) \qquad\qquad \kappa = \frac{\alpha + \beta}{\cos(\mu) + \sin(\mu)} \tag{9}$$

3.2. The sum of two von Mises

The sum of two von Mises distributed variables is achieved by first approximating each of them by a wrapped Normal distribution, summing the wrapped Normals to give another wrapped Normal, and then returning the latter to a von Mises [36]. Denoting the wrapped Normal distribution as $w\mathcal{N}(\cdot)$:

$$\mathcal{M}\big(\theta \,|\, \mu, \kappa\big) \approx w\mathcal{N}\big(\theta \,|\, \mu, \mathrm{B}(\kappa)\big) \tag{10}$$

where

$$\mathrm{B}(\kappa) = \frac{I_1(\kappa)}{I_0(\kappa)} \tag{11}$$

If we have two von Mises variables, θ_1 and θ_2, with distributions $\mathcal{M}\big(\theta_1 \,|\, \kappa_1, \mu_1\big)$ and $\mathcal{M}\big(\theta_2 \,|\, \kappa_2, \mu_2\big)$, then their sum is given by

$$\mathrm{p}(\theta_1 + \theta_2) \approx w\mathcal{N}\big(\mu_1 + \mu_2, \mathrm{B}(\kappa_1)\mathrm{B}(\kappa_2)\big) \tag{12}$$

$$\approx \mathcal{M}\Big(\mu_1 + \mu_2, \mathrm{B}^{-1}\big(\mathrm{B}(\kappa_1)\mathrm{B}(\kappa_2)\big)\Big) \tag{13}$$

While calculating $\mathrm{B}(\cdot)$ is straightforward, there does not appear to be a solution for its inverse, $\mathrm{B}^{-1}(\cdot)$. A numerical solution is possible, but with a computational expense we cannot afford. Instead we approximate it using the following definition, which was discovered using a genetic program:

$$\mathrm{B}^{-1}(x) \approx \left(\frac{1}{\operatorname{acos}(x)}\right)^2 \tag{14}$$

3.3. Generalised von Mises of order 2 (GvM_2)

The Generalised von Mises distribution of order 2 (GvM_2) [37, 38, 39] is a potentially asymmetrical, potentially bimodal distribution for a circular variable and is defined as:

$$\begin{aligned} &\mathcal{G}v\mathcal{M}_2(\theta \,|\, \boldsymbol{\mu}, \boldsymbol{\kappa}) \\ &= \frac{1}{2\pi G_0(\delta_\mu, \boldsymbol{\kappa})} \exp\left(\sum_{m=1}^{2} \kappa_m \cos\big(m(\theta - \mu_m)\big)\right) \end{aligned} \tag{15}$$

where $G_0(\cdot)$ is an analytically intractable integral, $\delta_\mu = \mod(\mu_1 - \mu_2, \pi)$ and $\kappa_m > 0$. Note that the Generalised von Mises of order 1 is identical to the von Mises. The definition in (15) can equivalently be expressed as

$$\begin{aligned} &\mathcal{G}v\mathcal{M}_2(\theta \,|\, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &= \frac{1}{2\pi G_0(\cdot)} \exp\left(\sum_{m=1}^{2} \alpha_m \cos(m\theta) + \beta_m \sin(m\theta)\right) \end{aligned} \tag{16}$$

where

$$\alpha_1 = \kappa_1 \cos(\mu_1) \qquad \beta_1 = \kappa_1 \sin(\mu_1) \tag{17}$$

$$\alpha_2 = \kappa_2 \cos(2\mu_2) \qquad \beta_2 = \kappa_2 \sin(2\mu_2) \tag{18}$$

and, alternatively,

$$\mu_1 = \operatorname{atan}\left(\frac{\beta_1}{\alpha_1}\right) \qquad \kappa_1 = \frac{\alpha_1 + \beta_1}{\cos(\mu_1) + \sin(\mu_1)} \tag{19}$$

$$\mu_2 = \frac{1}{2}\operatorname{atan}\left(\frac{\beta_2}{\alpha_2}\right) \qquad \kappa_2 = \frac{\alpha_2 + \beta_2}{\cos(2\mu_2) + \sin(2\mu_2)} \tag{20}$$

We can see that μ_1 is within a 2π interval, while μ_2 is in a π interval.

4. The Base Model

In this paper we follow standard notation whereby $\mathrm{p}(\cdot)$ denotes an exact probability density function, while $\mathrm{q}(\cdot)$ denotes the Variational Bayes (VB) approximation of a posterior distribution. The superscript "(n)", e.g. $\mathrm{q}^{(n)}(\cdot)$, denotes the state after the nth observation has been incorporated into the model. For brevity, in the VB updates the expectation $\langle\cdot\rangle$ represents the expected value of the stated variable after the previous observation has been incorporated.

The original model in (1) is inconvenient for two reasons. The first is that t_n is a continuously incrementing value, which we need to avoid in our infinitely running system. The other is that the angular frequency in our model, ω, is a circular variable, but rather than having the range $-\pi$ to π, has the range from minus to plus the Nyquist frequency, which means we cannot give it a convenient conjugate von Mises prior. We solve the latter problem by defining a new variable, $\nu = \omega\delta t$, whose range is $-\pi$ to π, and redefine the model in terms of ν instead of ω. Since $\langle\nu\rangle = \langle\omega\delta t\rangle$ is just $\langle\omega\rangle\delta t$, we can easily calculate $\langle\omega\rangle$. The updated model now looks like this, where $n \geq 0$:

$$y_n = A\cos(\phi + n\nu) + \epsilon_n \tag{21}$$

Although the inconvenient t_n term has disappeared, it has been replaced by the similarly inconvenient n. To get around this we introduce a new variable, defined as follows:

$$\tau_n = \phi + \sum_{i=1}^{n} \nu^{(i)} \tag{22}$$

Rather than τ_n being an independently evaluated variable, we use the method described in section 3.2 to keep a running total of the sum defined in (22). Since this is a circular variable, although the summation might be infinite, the expectation is limited to the range $-\pi$ to π.

With this change, the final model to be evaluated using VB approximation looks like this:

$$y_0 = A\cos(\tau_0) + \epsilon_n \tag{23}$$
$$y_{n>0} = A\cos(\tau_{n-1} + \nu) + \epsilon_n \tag{24}$$

Before any observations have been made, we only have prior distributions for each of the four variables, the amplitude A, the phase ϕ, the angular frequency substitute ν, and the noise precision λ. When the first observation is incorporated, we obtain a first set of posterior distributions for A, ϕ and λ, and we set $\tau_0 = \phi$. The posteriors for A and λ become the priors for the next observation, y_1, and we now obtain posteriors for A, ν and λ. We set $\tau_1 = \tau_0 + \nu$ (using the method described in 3.2), and the posteriors become the priors for the next observation. Each subsequent observation is incorporated in the same way.

Note that for each observation, only a single round of updates is performed; there is no iterative series of updates as is more usual with VB. The update scheme is summarised in algorithm 1.

There are two ambiguities in this system. The first is that a signal with amplitude A and phase ϕ is exactly equivalent to one with amplitude $-A$ and phase $\pi + \phi$. We avoid this by ensuring that A is always greater than or equal to zero. The second is that a signal with phase ϕ and angular frequency ω gives exactly the same set of observations as a signal with phase $2\pi - \phi$ and frequency $-\omega$, the difference being the direction of travel of the wave.

With the base model defined, i.e. the one with no limitations on the posterior precisions, we now consider the priors for each variable, the initialisation of the variables, and then the VB update scheme.

4.1. Priors

In this paper we will assume the simplest noise model, i.e. a zero mean Gaussian distribution, so that the proposed method can be more clearly described. To make it more robust to outliers, we might choose in future to assume a Student-t distribution [1], which is effectively an infinite mixture of Gaussians; this could easily be incorporated into this model.

Since the noise in the system is assumed to be Gaussian, the likelihood, derived from (23) and (24), is

$$\mathrm{p}(y_0 \mid A, \phi, \lambda) = \mathcal{N}\big(y_0 \mid A\cos(\phi), \lambda^{-1}\big) \tag{25}$$
$$\mathrm{p}(y_n \mid A, \tau_{n-1}, \nu, \lambda) = \mathcal{N}\big(y_n \mid A\cos(\tau_{n-1} + \nu), \lambda^{-1}\big) \tag{26}$$

and we define the prior for the noise precision, λ, as a conjugate Gamma distribution[1]

$$\mathrm{p}(\lambda) = \mathcal{G}\big(\lambda \mid a_\lambda, b_\lambda\big) \tag{27}$$

[1] defined as: $\mathcal{G}(x \mid a, b) = \frac{1}{\Gamma(a)} b^a x^{1-a} \exp(-bx)$

For the amplitude, A, we would like to use a prior that enforces the condition that $A \geq 0$, such as a Rayleigh or Gamma distribution, but they are not conjugates in this case. A half-Gaussian always has its maximum value at zero, and both the mean and the spread are expressed in terms of a single scale parameter. From this it is impossible to obtain a meaningful measure of certainty from the posterior distribution, as a high mean value must also have a high spread. Quantifying the uncertainty in a meaningful way is very important for the intended applications of this work, so instead we define a Gaussian prior with mean μ_A and precision ρ_A, as follows:

$$\mathrm{p}(A) = \mathcal{N}\big(A \,|\, \mu_A, \rho_A^{-1}\big) \tag{28}$$

and discuss the potential negative values of the expectation in section 4.3.3. With no other information available, our priors for ϕ and ν are Uniform distributions over the interval $-\pi$ to π. For the initial phase offset, ϕ, this is sufficient, so we define:

$$\mathrm{p}(\phi) = \mathcal{U}\big(\phi \,|\, -\pi, \pi\big) \tag{29}$$

However, for the angular frequency substitute, ν, the posterior will turn out to be a GvM_2, so, for conjugacy later, we specify the prior in those terms instead:

$$\mathrm{p}(\nu) = \mathcal{GvM}_2\big(\nu \,|\, \boldsymbol{\kappa}, \boldsymbol{\mu}\big) \tag{30}$$

where the two concentration values in $\boldsymbol{\kappa}$ are set to zero to give the equivalent of the Uniform distribution $\mathcal{U}(\nu \,|\, -\pi, \pi)$.

4.2. Initialisation

One method for initialising the system would be to perform a Fourier Transform on the first set of observations, identify the peak frequency, set the prior expectations $\langle A \rangle$, $\langle \phi \rangle$, $\langle \nu \rangle$ and $\langle \lambda \rangle$ appropriately, and define the prior precisions to be, perhaps, relatively large.

However, we have chosen to define uninformative priors to demonstrate that the system still converges to good solutions. The hyperparameters are set as follows:

$$\lambda : \qquad a_\lambda = 1, \qquad b_\lambda = 1 \tag{31}$$

$$A : \qquad \mu_A = 1, \qquad \rho_A = 1 \tag{32}$$

$$\nu : \qquad \boldsymbol{\kappa} = \mathbf{0}, \qquad \boldsymbol{\mu} = \mathbf{0} \tag{33}$$

and the initial value of each variable's expectation is set to that of its prior. Note that these priors are only explicitly assimilated into the model when the first observation is incorporated, which explains why the model is insensitive to the values.

4.3. Variational approximations

Following [2], with a fully factorised posterior, we now derive VB updates for the four variables in the base model, ϕ, ν, A and λ, and then describe the method for updating the probability distribution for τ and calculating its various expectations.

In the following derivations we make use of the standard trigonometrical identity

$$\cos^2(\theta) = \frac{1 + \cos(2\theta)}{2} \tag{34}$$

4.3.1. Phase offset, ϕ

This variable is only estimated once, when the first observation, y_0, is incorporated into the model based on the likelihood in (23). In fact, ϕ is just τ_0. Following the VB process:

$$\begin{aligned}
&\log\big(\mathrm{q}(\phi)\big) \\
&= \log\big(\mathrm{p}(y_0 \mid A, \phi, \lambda)\,\mathrm{p}(A)\,\mathrm{p}(\phi)\,\mathrm{p}(\lambda)\big) && (35)\\
&= \log\big(\mathcal{N}(y_0 \mid A\cos(\phi), \lambda^{-1})\,\mathcal{U}(\phi \mid -\pi, \pi)\big) + \text{constant} && (36)
\end{aligned}$$

At each stage of the calculation, all terms not dependent on ϕ are absorbed into the constant term. For clarity, this term is omitted from now on. Since the prior for ϕ is Uniform, it contains no terms in ϕ, so we are left with:

$$\begin{aligned}
\log\big(\mathrm{q}(\phi)\big) &= -\frac{\lambda}{2}\Big(y_0 - A\cos(\phi)\Big)^2 && (37)\\
&= -\frac{\lambda A^2}{4}\cos(2\phi) + \lambda y_0 A\cos(\phi) && (38)
\end{aligned}$$

which has the form of a GvM_2, so the approximate posterior is $\mathcal{G}v\mathcal{M}_2(\boldsymbol{\alpha}, \boldsymbol{\beta})$, where, replacing all other variables with their current expectations:

$$\begin{aligned}
\alpha_1 &= \langle\lambda\rangle y_0 \langle A\rangle & \beta_1 &= 0 && (39)\\
\alpha_2 &= -\frac{1}{4}\langle\lambda\rangle\langle A^2\rangle & \beta_2 &= 0 && (40)
\end{aligned}$$

Since both $\langle\lambda\rangle$ and $\langle A\rangle$ (and hence $\langle A^2\rangle$) are positive values, α_1 is positive and α_2 negative.

The fact that the two posterior β values are zero means that the value for μ_1 is always either 0 or π, and that for μ_2 is always $\pi/2$. This tends to result in unimodal distributions with a mean of either 0 or π, but not necessarily.

4.3.2. Angular frequency substitute, ν

This variable is only evaluated when the second and subsequent observations (i.e. $n > 0$) are incorporated into the model, based on the likelihood in (24). If this is the first evaluation of ν ($n = 1$), then we must set $\mathrm{q}^{(0)}(\nu) = \mathrm{p}(\nu)$.

Following a similar VB process as that described above, in section 4.3.1, we arrive at an approximate posterior of $\mathrm{q}^{(n)}(\nu) = \mathcal{G}v\mathcal{M}_2(\boldsymbol{\alpha}^{(n)}, \boldsymbol{\beta}^{(n)})$, where

$$\alpha_1^{(n)} = \langle\lambda\rangle y_n \langle A\rangle \langle\cos(\tau)\rangle + \kappa_1^{(n-1)} \cos(\mu_1^{(n-1)}) \tag{41}$$

$$\alpha_2^{(n)} = -\frac{\langle\lambda\rangle\langle A^2\rangle}{4} \langle\cos(2\tau)\rangle + \kappa_2^{(n-1)} \cos(2\mu_2^{(n-1)}) \tag{42}$$

$$\beta_1^{(n)} = -\langle\lambda\rangle y_n \langle A\rangle \langle\sin(\tau)\rangle + \kappa_1^{(n-1)} \sin(\mu_1^{(n-1)}) \tag{43}$$

$$\beta_2^{(n)} = \frac{\langle\lambda\rangle\langle A^2\rangle}{4} \langle\sin(2\tau)\rangle + \kappa_2^{(n-1)} \sin(2\mu_2^{(n-1)}) \tag{44}$$

from which, using (17–20), we may calculate $\boldsymbol{\kappa}^{(n)}$ and $\boldsymbol{\mu}^{(n)}$.

4.3.3. Amplitude, A

If this is the first evaluation of A ($n = 0$), then we must set $\mathrm{q}^{(-1)}(A) = \mathrm{p}(A)$. Following the VB procedure, the approximate posterior for A is the Gaussian $\mathrm{q}^{(n)}(A) = \mathcal{N}(\mu_A^{(n)}, \rho_A^{(n)})$, with

$$\rho_A^{(n)} = \langle\lambda^{(n-1)}\rangle \langle\cos^2(\tau^{(n-1)})\rangle + \rho_A^{(n-1)} \tag{45}$$

$$\mu_A^{(n)} = \left(\rho_A^{(n)}\right)^{-1} \left(\langle\lambda^{(n-1)}\rangle y_n \langle\cos(\tau^{(n-1)})\rangle + \rho_A^{(n-1)} \mu_A^{(n-1)} \right) \tag{46}$$

This Gaussian distribution admits the possibility of negative values for $\langle A\rangle$. This is not in itself a problem as a sinusoid with amplitude $-A$ and phase ϕ is equivalent to one with amplitude A and phase $\phi + \pi$. However, amplitudes with small magnitudes may oscillate between negative and positive values during the update process, causing big jumps in other variables and the model to not converge properly. To prevent this we use the absolute value of $\langle A\rangle$ in place of $\langle A\rangle$ in each of the other posterior expressions, as in [40].

4.3.4. Noise precision, λ

If this is the first evaluation of λ ($n = 0$), then we must set $\mathrm{q}^{(-1)}(\lambda) = \mathrm{p}(\lambda)$. Following the VB procedure, the approximate posterior for λ is the Gamma $\mathrm{q}^{(n)}(\lambda) = \mathcal{G}(a_\lambda^{(n)}, b_\lambda^{(n)})$, with

$$a_\lambda^{(n)} = a_\lambda^{(n-1)} + 1 \tag{47}$$

$$b_\lambda^{(n)} = b_\lambda^{(n-1)} + \frac{1}{2}\left[y_n^2 + \langle A^2\rangle \langle\cos^2(\tau)\rangle - 2 y_n \langle A\rangle \langle\cos(\tau)\rangle \right] \tag{48}$$

4.3.5. Running phase, τ_n

The running phase, τ_n, is not a variable whose posterior is updated in the VB scheme, but the sum of a number of GvM_2 distributed variables:

$$\tau^{(0)} = \phi \tag{49}$$

$$\tau^{(n)} = \tau^{(n-1)} + \nu^{(n)} \qquad \text{for } n > 0 \tag{50}$$

With no known method for calculating the sum of two GvM_2 distributions, and with the necessity for us to define expectations for this variable to be used in the VB updates for the other variables, we want to maintain the posterior of τ as a (unimodal) von Mises distribution, with the posterior GvM_2 distributions for ϕ and $\nu^{(n)}$ approximated by von Mises distributions before the summation.

A bimodal GvM_2 distribution has two mean values: μ_1, which has a 2π range, and μ_2, which only has a range of π. We approximate the distribution as follows:

$$\mathcal{GvM}_2\left(\begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix}, \begin{bmatrix}\kappa_1\\ \kappa_2\end{bmatrix}\right) \approx \mathcal{M}(\mu_1, \kappa_1) \tag{51}$$

Note that while we use this approximation in the calculation of τ, we maintain the full GvM_2 in future calculations of ν. Note that there is a real and unavoidable ambiguity in the angular frequency ν, where a sinusoidal wave travelling in one direction, and one of the same absolute frequency travelling in the opposite direction will provide identical observations and differ only in the sign of the frequency. By selecting one of the means for updating τ we are effectively choosing one direction; which one is seeded by the very earliest observations and repeated selection reinforces it, resulting in the GvM_2 becoming unimodal.

For ϕ, the posterior distribution is somewhat meaningless, because it is calculated on the basis of a single observation. We therefore choose to fix the "posterior" of ϕ as a von Mises distribution with mean zero, and do not calculate the posterior. So now we may rewrite (49)–(50) as

$$\tau^{(0)} \sim \mathcal{M}(0, 10^{-3}) \tag{52}$$

$$\tau^{(n)} \sim \mathcal{M}\left(\mu_\tau^{(n-1)} + \mu_\nu^{(n)}, \mathrm{B}^{-1}\big(\mathrm{B}(\kappa_\tau^{(n-1)})\mathrm{B}(\kappa_\nu^{(n)})\big)\right) \tag{53}$$

where the $\mathcal{M}(0, 10^-3)$ is an uninformative prior with low concentration (in fact, the value of the concentration seems to make no material difference to the model learning, certainly in the range 10^-3 to 10^3). For the VB updates of the other variables in the model, we need to calculate $\langle\cos(\tau)\rangle$, $\langle\cos(2\tau)\rangle$, $\langle\sin(\tau)\rangle$ and $\langle\sin(2\tau)\rangle$, which we can do using (3) and (4). Note that as the concentration parameter κ tends towards zero, the Bessel fraction $\mathrm{B}(\kappa) = I_n(\kappa)/I_0(\kappa)$ tends to zero, and so all of these expectations also tend to zero. While the model does converge on a concentration value for τ, the value is low, typically 0.6, and the value of A is consistently underestimated. Fixing this concentration at a high value, in other words saying that we are very certain about the current estimate for $\langle\tau\rangle$, fixes this problem by forcing the uncertainty to be absorbed into the posterior of ν. The final resulting definition for τ is, therefore:

$$\tau^{(0)} \sim \mathcal{M}(0, 100) \tag{54}$$

$$\tau^{(n)} \sim \mathcal{M}\big(\mu_\tau^{(n-1)} + \mu_\nu^{(n)}, 1000\big) \tag{55}$$

which has the additional benefit of avoiding the computational expense of calculating the Bessel fraction, $\mathrm{B}(\cdot)$, and its inverse.

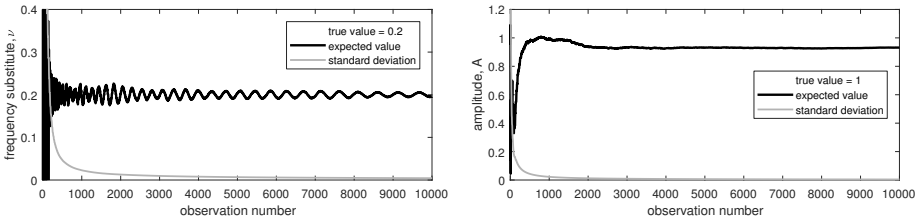


Figure 2: An example of (left) frequency subtitute ν and (right) amplitude A converging over time for a statistically stationary system.

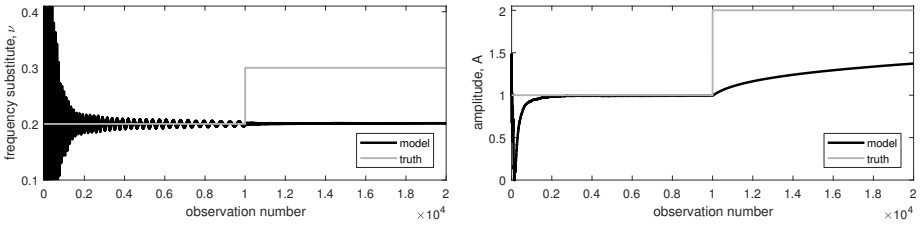


Figure 3: The base model converging on the true values for (left) frequency substitute ν and (right) amplitude A, before failing to adapt a change in the underlying system.

5. Base Model: experimental results

Figure 2 shows $\langle\nu\rangle$ and $\langle A\rangle$ converging for a statistically stationary signal. The value of $\langle\nu\rangle$ continues to oscillate, with a decreasing size of variation, and this oscillation is probably the cause of $\langle A\rangle$ converging to a value slightly lower than the truth. Also shown is the standard deviation in each case, which drops off very quickly, resulting in tight posterior distributions.

So, the base model learns in a stationary system, but as we can see from figure 3 where the signals undergo abrupt changes in amplitude and frequency, if the underlying system changes, it has becomes so certain that it is unable to track those changes.

6. The Full Model

It is clear from equations (47) and (48) that the values of a_λ and b_λ carry on increasing as new observations are incorporated. Not so obvious is that the values of ρ_A in (45) and $\boldsymbol{\kappa}$ in section 4.3.2 also continuously increase over time. Eventually the values of these variables will exceed the maximum value able to be represented by the software and the system will fail.

If we consider the precision λ and the variational updates shown in equations (47)–(48) we note that the values of parameters $a_\lambda^{(n)}$ and $b_\lambda^{(n)}$ are incremented with positive values at each time step. If we were to include a Gamma prior into these equations, this will only increment the values further (by fixed values), unless we were to use an improper prior with negative parameters. So with a

proper prior we cannot limit the continuous increase of the posterior parameters. Instead, we achieve the desired effect by applying priors to the parameters of these precision variables to limit their growth. Since the maximum values are constrained to some upper bound, the precisions themselves are also constrained.

Selecting suitable hyperparameter values for the priors has the added benefit of limiting the tightness of the precisions' posterior distributions and thus enabling the model to respond to nonstationarity. This is a trade-off situation: if the limit on precision variables is too tight, then we have the same situation as that of the base model; if the limit is too loose, then the system allows too much variability in the model, which can lead to, for example, a wrong value for ν being compensated for by variability in A. The values detailed in the following sections have been found by experimentation to provide the right level of trade-off for all the datasets the model has been tested on so far, but further work is required to establish a specific method for choosing them.

6.1. Variational approximations: online adjustments

We start by considering the noise hyperparameters, a_λ and b_λ, and the amplitude precision ρ_A, for which the approximate posterior distributions may be calculated in relatively formal ways. The form of the GvM_2 distribution means that there is no conjugate prior for the concentration variables, $\boldsymbol{\kappa}$, so these are treated using a more empirical method.

6.1.1. Noise precision hyperparameters, a_λ and b_λ

The two posterior hyperparameters for the noise precision, λ, must both be positive values, so we assign them Gamma priors: $\mathcal{G}(a_\lambda \,|\, c_a, d_a)$ and $\mathcal{G}(b_\lambda \,|\, c_b, d_b)$ respectively. Combining these priors with the results in (47)–(48), we may now apply the VB process to obtain posterior distributions. Starting with the easier b_λ, and dropping the observation number superscript, for clarity:

$$\begin{aligned}\log(\mathrm{q}(b_\lambda)) &= \log\big(\mathbb{E}[\mathrm{q}(\lambda \,|\, a_\lambda, b_\lambda)]\mathbb{E}[\mathrm{p}(b_\lambda \,|\, c_b, d_b)]\big) && (56)\\ &= (c_b + \langle a_\lambda\rangle - 1)\log(b_\lambda) - b_\lambda(d_b + \langle\lambda\rangle) && (57)\end{aligned}$$

which is the Gamma distribution

$$\mathcal{G}\big(b_\lambda \,|\, c_b + \langle a_\lambda\rangle, d_b + \langle\lambda\rangle\big) \tag{58}$$

Following the same process for a_λ:

$$\begin{aligned}\log(\mathrm{q}(a_\lambda)) &= \log\big(\mathbb{E}[\mathrm{q}(\lambda \,|\, a_\lambda, b_\lambda)]\mathbb{E}[\mathrm{p}(a_\lambda \,|\, c_a, d_a)]\big) && (59)\\ &= a_\lambda\langle\log(b_\lambda)\rangle - \log(\Gamma(a_\lambda)) + a_\lambda\langle\log(\lambda)\rangle \\ &\quad + (c_a - 1)\log(a_\lambda) - d_a a_\lambda && (60)\end{aligned}$$

Following [1] we use Stirling's first order approximation for the $\log(\Gamma(a_\lambda))$ term:

$$\log(\Gamma(a_\lambda)) \approx \left(a_\lambda - \frac{1}{2}\right)\log(a_\lambda) - a_\lambda \tag{61}$$

giving us, for (60):

$$\log(\mathrm{q}(a_\lambda)) = \left(c_a - a_\lambda + \frac{1}{2} - 1\right)\log(a_\lambda) - \left(d_a - \langle\log(b_\lambda)\rangle - 1 - \langle\log(\lambda)\rangle\right)a_\lambda \tag{62}$$

This is a Gamma distribution, apart from the $a_\lambda \log(a_\lambda)$ term. We avoid this complication by replacing this with $\langle a_\lambda\rangle \log(a_\lambda)$, where $\langle a_\lambda\rangle$ is the expected value from the previous estimate of the posterior. So now have the Gamma posterior

$$\mathrm{q}(a_\lambda) = \mathcal{G}\big(a_\lambda \,|\, c_a - \langle a_\lambda\rangle + \frac{1}{2}, d_a - \langle\log(b_\lambda)\rangle - 1 - \langle\log(\lambda)\rangle\big) \tag{63}$$

Where, for example, $\langle\log(\lambda)\rangle = \psi(a_\lambda) - \log(b_\lambda)$, and $\psi(\cdot)$ is the digamma function.

For the results shown in this paper, the following values were used:

$$c_a = d_a = c_b = d_b = 1 \tag{64}$$

6.1.2. Amplitude precision hyperparameters, c_ρ and d_ρ

We assign the amplitude precision, ρ_A, the Gamma prior $\mathcal{G}(\rho_A \,|\, c_\rho, d_\rho)$. Combining this with the results from (45–46), we apply the usual VB process to obtain a posterior:

$$\log(\mathrm{q}(\rho_A))$$

$$= \log(\rho_A) - \rho_A \frac{(A-\mu_A)^2}{2} + (c_\rho - 1)\log(\rho_A) - d_\rho\rho_A \tag{65}$$

$$= (c_\rho + 1 - 1)\log(\rho_A) - \rho_A\left(d_\rho + \frac{1}{2}(A-\mu_A)^2\right) \tag{66}$$

Given that $\langle A^2\rangle = \rho_A^{-1} + \mu_A^2$ and $\langle A\rangle = \mu_A$, we obtain the Gamma distribution

$$\mathcal{G}(\rho_A \,|\, c_\rho + 1, d_\rho + \frac{1}{2\langle\rho_A\rangle}) \tag{67}$$

where $\langle\rho_A\rangle$ is the expectation of that variable from the previous iteration.

For the results shown in this paper, the following values were used:

$$c_\rho = 10^{-3} \tag{68}$$

$$d_\rho = 10^{-2} \tag{69}$$

6.1.3. Frequency precision hyperparameters, $\boldsymbol{\kappa}$

Since we are approximating the posterior GvM_2 distribution for ν as a unimodal von Mises, but maintaining the GvM_2 for the updates, the first concentration, κ_1, dominates the second, κ_2. We cannot apply the same limiting process to both concentrations as this causes κ_2 to "catch up" with κ_1, resulting in significant changes to the shape of the distribution. We could apply different priors for the two variables, but instead we choose to apply the prior to κ_1 only, and update κ_2 proportionally.

We would choose to assign κ_1 a Gamma prior and combine it with the GvM_2 posterior for ν, but the normalising factor in the GvM_2 (the $G_0(\cdot)$ term in (15)) is a complicated function of κ_1 that does not lend itself to the VB process. Another thought is to take the route through the von Mises to the wrapped Normal, as described in section 3.2, but this gives us a Bessel function of κ_1, which again is not amenable to the VB process. Instead we resort to a more empirical method, related to the precision of the sum of two Gaussian distributions, one having precision κ_1 and the other, the "prior", having precision r:

$$\kappa_1^{\text{new}} = \left(\frac{1}{r} + \frac{1}{\kappa_1^{\text{old}}} \right) \tag{70}$$

The update for κ_2 then becomes

$$\kappa_2^{\text{new}} = \kappa_2^{\text{old}} \frac{\kappa_1^{\text{new}}}{\kappa_1^{\text{old}}} \tag{71}$$

For the results shown in this paper, a value of $r = 10^5$ has been used.

6.2. Full algorithm

The algorithm for the full model is shown in algorithm 1. Note that the complexity of the algorithm is constant in time, i.e. $\mathcal{O}(1)$ (as, in fact, is the base model), so provided that each "iteration" can be completed within the observation time interval, δt, the model is able to run in real-time indefinitely.

7. Full Model: experimental results

With limits on the model's precision variables, it is clear that it is likely to converge more loosely on the true values, depending, of course, on the selected priors for those variables. Figure 4 shows this effect, with the model estimates not fully converging on the truth. But note how the system is now able to track abrupt changes in these underlying values, where before (see figure 3) the tightness of the precisions caused the model to fail to adapt to these same changes.

Figure 5 shows the model tracking continuous changes to both frequency and amplitude. For the first 10,000 observations, and observations 15,000 to 20,000 the system is stationary, with, in the two cases, different random selections of the true variables: $A \sim \mathcal{U}(0, 10)$, $\omega \sim \mathcal{U}(-\pi, \pi)$ and $\phi \sim \mathcal{U}(-\pi, \pi)$. The

Algorithm 1 The VB update scheme for both models; the base model does not include the final *if* statement.

define the prior hyperparameters using (31–33)
initialise the value of each variable from its prior
for n=0 to ∞ **do**
  if n==0 **then**
    calculate $\tau^{(0)}$ using (54)
  else
    calculate/update the posterior for ν using (41–44)
    calculate $\tau^{(n)}$ using (55)
  end if
  calculate the expectations for $\tau^{(n)}$ using (3–4)
  calculate/update the posterior for A using (45–46)
  calculate/update the posterior for λ using (47–48)
  if full model **then**
    update the λ precision using (58–63)
    update the ρ precision using (67)
    update the κ precision using (70–71)
  end if
end for

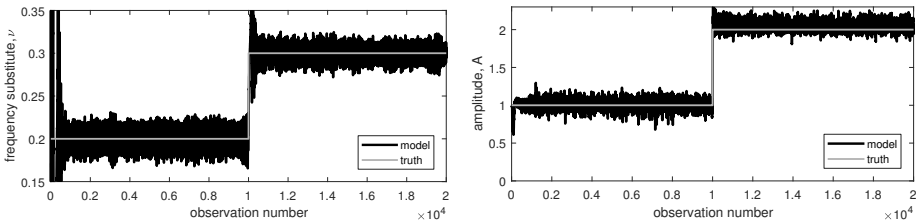


Figure 4: An example of (left) frequency substitute ν and (right) amplitude A converging with abrupt changes in the underlying system.

standard deviation of the noise is a Uniformly random value between 0 and the minimum of the two randomly-selected amplitudes. Between observations 10,000 and 15,000 the values of A and ω transition linearly between their two values. Apart from a short initial stabilisation period, both A and ω are tracked rather closely, apart from when the true value of ω transitions through zero. In this particular example the model's estimate of ω has the same sign as the true value; the probability of the estimated sign being correct is 0.5 as the model cannot distinguish the direction of travel of the wave.

For this example set of observations, the model was trained 1,000 times with the values in expressions (31)–(33) set to random values between 0 and 10, or 0 and 2π for the mean of ν. Given these priors are only explicitly assimilated into the model when the first observation is incorporated, we would expect the model to be insensitive to the values, which turns out to be the case. However,

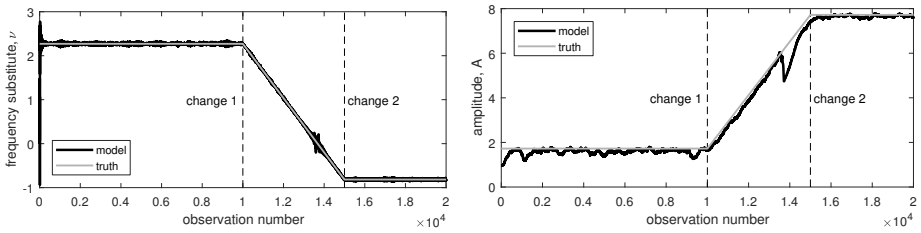


Figure 5: An example of (left) frequency substitute ν and (right) amplitude A constantly changing to track underlying changes in both the frequency and amplitude of the true signal. At approximately observation number 14,000 the frequency passes through zero, causing a wobble in the estimations.

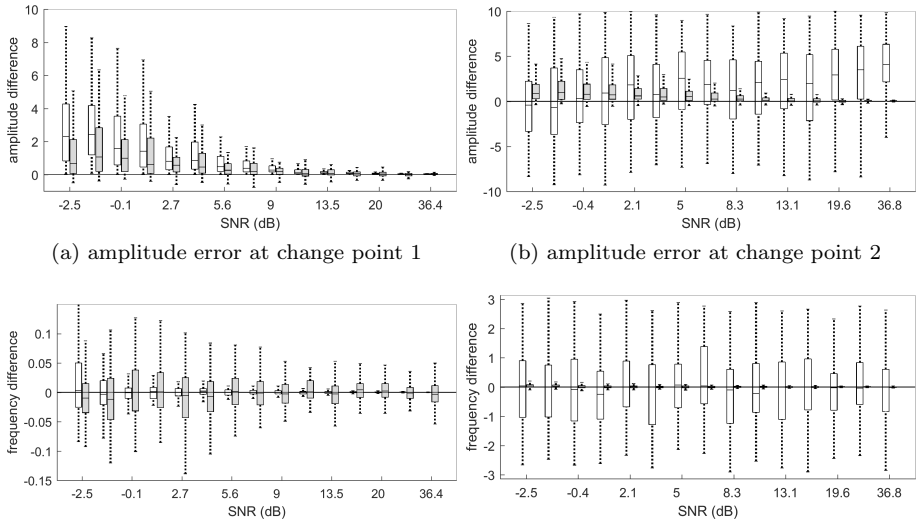
in 3% of the tests the model completely failed to track the signal, but this does not seem to be related to any specific pattern in the setting of these values.

In order to compare the functionality of the base and full models, 1,200 datasets of the form described above (figure 5) were generated and both models trained against each of them. In other words, each test dataset contained 20,000 observations, the first 10,000 observations from one set of randomly assigned values for A and ω, the last 5,000 observations from a different set of randomly assigned values for A and ω, and linear transition between the two sets of values for the 5,000 observations inbetween. Single values for ϕ and λ were randomly selected separately for each dataset.

Clearly the ability of the models to estimate the true values of the system variables is affected by the signal-to-noise ratio (SNR) in each case. The boxplots in figure 6 summarise the results by SNR. Each box represents the extent of the 25^{th} to 75^{th} percentile, with the median marked within the box, and whiskers extending to cover approximately 99% of the values. There is a pair of boxes for each SNR value; the left, white box is from the base model, while the right, grey box is the full model. For amplitude the value summarised is $A_{\text{true}} - \langle A \rangle$; for frequency, in order to ignore the inability of the model to determine the sign, the value is $|\omega_{\text{true}}| - |\langle \nu \rangle|$.

The effect of SNR is clearly seen in figure 6a which compares amplitude at changepoint 1, where both models show decreasing errors for increasing SNR. The errors for the full model tend to be smaller, but while the base model always underestimates the true value, the full model does not. At changepoint 2 (6b) the full model displays the same characteristics as before, but the base model is clearly unable to track the change, even at high SNR. For the frequency the same two plots are shown (figures 6c and 6d), though note the very different scales on the y axes in these two plots.

At changepoint 1 the full model is slightly worse than the base model, but it is considerably better at changepoint 2. Not shown here, but giving the same results as for changepoint 2, are the errors at the final observation. In the base model, because there is no constraint on how big the precision can be, in a stationary system, like that up to changepoint 1, the precision continues to increase over time, causing the posteriors to become tightly distributed around



(a) amplitude error at change point 1

(b) amplitude error at change point 2

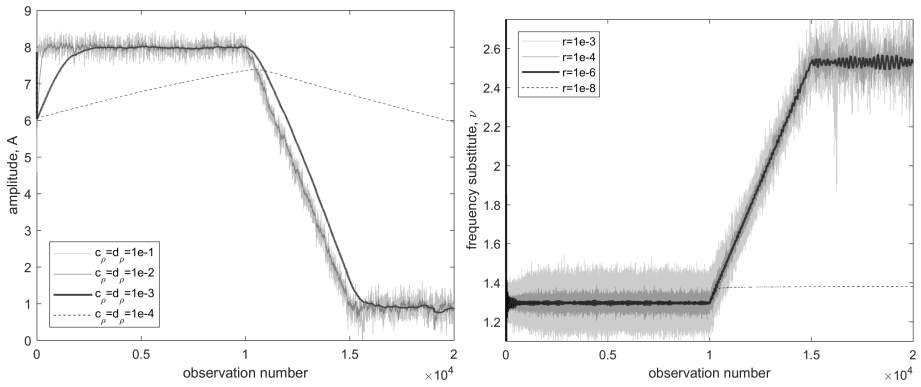
(c) frequency error at change point 1

(d) frequency error at change point 2

Figure 6: Summary boxplots of the results of running the base and full models against the same set of 1,200 randomly-generated sets of observations of the form shown in figure 4. At each SNR value there are two boxes: the left, white box is from the base model, while the right, grey box is from the full model.

the true value. The full model limits this increase in precision, so the posteriors never become very tight, leading to wobbles around the true value and hence worse accuracy at this point than the base model. However, this limiting of the precision means that when the underlying system changes, as it does between the two changepoints, the full model is able to respond and track those changes, whereas the base model has become so sure that it is unable to react to the changes (as shown in figure 3). Hence the full model is considerably better at changepoint 2.

As stated in section 4.2, the model is insensitive to the hyperparameters for A, λ and ν, as these contribute to the model only when the first observation is incorporated. There is sensitivity to the precision hyperparameters, because these control the upper limit placed on the precisions. If these limits are too high, then the full model converges onto the base model, which is good at learning the underlying system when it is statistically stationary, but, as shown in figure 3, when the underlying system then changes, it is unable to track those changes. Equally, if these limits are too low, then the model never converges on anything. The "Goldilocks" solution makes the limits high enough that the model can learn, and loose enough that it continues to be able to respond to changes. This loosening has the effect seen in figure 4, where the model is tracking even through an abrupt change, but the cost is that it does not full converge onto the truth (the black line is noisy around the gray truth in each case).



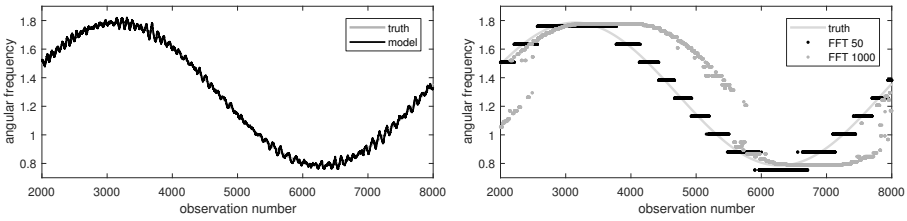
(a) amplitude precision hyperparameters (b) frequency precision hyperparameter

Figure 7: Plots showing how the model convergence varies when (a) the amplitude precision hyperparameters, c_ρ and d_ρ, and (b) the frequency precision hyperparameter, r, are varied.

Experimentally, the model seems to be sensitive only to the values of the amplitude precision hyperparameters, c_ρ and d_ρ (section 6.1.2), and to the value of r in the frequency precision (section 6.1.3). Figure 7a shows how changing the amplitude hyperparameters impacts the model's convergence on the true amplitude. In this case the true amplitude is 8.047 up to changepoint 1 and 0.980 after changepoint 2 (the true frequency is 1.296 and 2.529 respectively). In each of four examples, $c_\rho = d_\rho$, so the mean of the Gamma distribution is always 1, but the values shown are 10^{-1}, 10^{-2}, 10^{-3} and 10^{-4}, equating to different variances. The cases of 10^{-3} and 10^{-4} are not sufficiently constrained, as the model is lagging significantly behind changes to the truth because it is too certain of its estimates. The case of 10^{-1} is too constrained; the truth is well tracked, but there is a lot of variation in the estimate over time. The best of these options is 10^{-3}, as it follows the signal with low variation over time, and tracks closely when it changes. It should be noted that all four examples provide the same close estimates of the true frequency.

Figure 7b shows how changing the frequency precision hyperparameter r in equation (70) impacts the model's convergence on the true frequency, using the same signal as in the previous example. The case of 10^{-8} is not sufficiently constrained, as the model is lagging significantly behind changes to the truth because it is too certain of its estimates. The cases of 10^{-3} and 10^{-4} are too constrained; the truth is well tracked, but there is a lot of variation in the estimate over time. The best of these options is 10^{-7}, as it follows the signal with low variation over time, and tracks closely when it changes.

An initial attempt to track amplitude and frequency might be to repeatedly use the Fast discrete Fourier Transform (FFT) [41, 42] on the most recent set of observations. The size of this observation window would need to be optimised to trade off the competing requirements of a long window to get the best estimate of frequency and a short window to track changes to it over time. Since the



(a) the full model (b) FFT with window sizes 50 and 1000

Figure 8: For a system where the frequency of the underlying signal is constantly changing (the grey line), these plots compare how (a) the model tracks the frequency changes, with (b) how FFT with two different window sizes (50 and 1000 observations respectively) track those changes. For the FFT, the window size of 50 is clearly too small, with too few Fourier frequencies to capture the truth, while a window size of 1000 is too large, leading to a significant lag in the tracking.

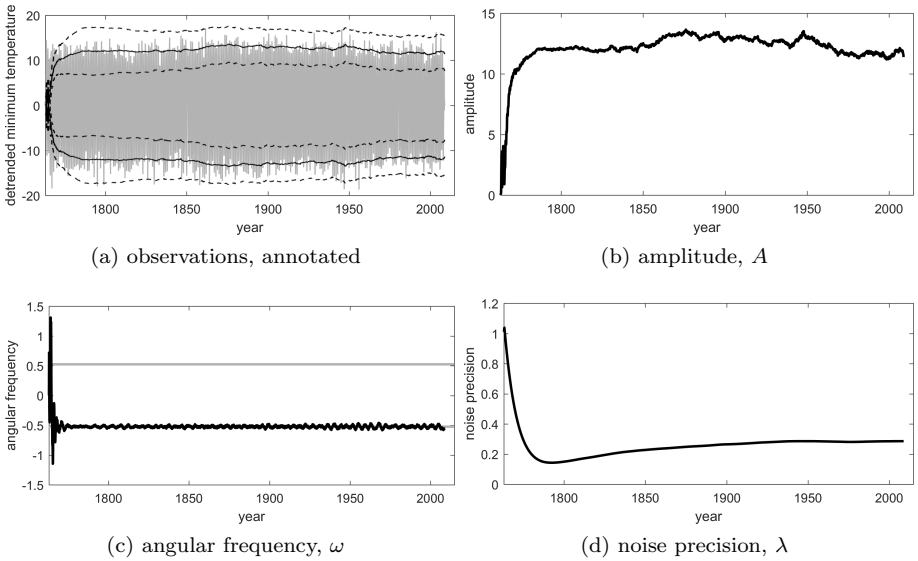
Fourier Transform assumes that the signal is periodic, we would also need to mitigate the effects that windowing has on the resulting spectrum (see, for example, [43]). With no prior knowledge of the variability of the true signal, it would be impossible to optimise the FFT window size and windowing function. Figure 8 compares the frequency estimated by FFT for two different window sizes (50 and 1000), and that estimated by the model, against the true frequency for a system where the true frequency is slowly evolving over time. Note that an FFT window size of 50 is clearly too small, with too few Fourier frequencies to capture the truth, while a window size of 1000 is too large, leading to a significant lag in the tracking. The model, in contrast, is tracking these changes rather closely.

8. Results on real data

The full model was trained on the 2,951 obervations of the monthly average of maximum daily temperature recorded in Milan between January 1763 and November 2008 [44]. Since the model assumes the observations to be sinusoidal about zero, the trend away from zero has been removed by subtracting the 20 year moving average.

The observations are shown as the grey line in figure 9a. This has been overlaid with two solid black lines showing plus and minus the model's expected amplitude over time, i.e. $\pm\langle A\rangle$. On either side of these solid black lines are two dashed black lines, indicating the amplitude plus and minus two times the model's expected noise standard deviation, i.e. $\pm 2/\sqrt{\langle\lambda\rangle}$. The expected amplitude and noise precision over time are also shown separately in figures 9b and 9d. We can see that the model quickly learns the amplitude, which varies over time and there is a "bounce" in the noise precision which is adjusted once the amplitude has settled down.

Figure 9c shows the expected angular frequency over time, i.e. $\langle\omega\rangle$. The true angular frequency for these monthly data is $2\pi/12 = 0.5236$ radians per



(a) observations, annotated

(b) amplitude, A

(c) angular frequency, ω

(d) noise precision, λ

Figure 9: Results from running the full model on real data, showing: (a) the observations in grey, overlaid with $\pm$ the model amplitude as solid black lines and each of these with $\pm$ two standard deviations of the noise ($2/\sqrt{\lambda}$) as dashed black lines; (b) the amplitude, A, over time; (c) the angular frequency, w, over time in black, with $\pm$ the true monthly frequency in grey; (d) the noise precision, λ, over time.

month. Here the model has settled on the negative value. The sinusoids could be "travelling" in either direction and this is an ambiguity in the system that the model is unable to resolve. However, as expected given the data, the angular frequency does not vary significantly over time. The small deviations are caused by the limitation placed on the concentration variable by the full model (section 6.1.3), allowing a small amount of exploration.

9. Conclusions

Despite the many and various approximations applied to this model, from the variational Bayesian approximation, to the use of unimodal von Mises distributions in place of the potentially bimodal GvM_2, it is still clearly able to learn the underlying system parameters and to keep track of changes to them. Each observation is incorporated as it arrives and then discarded, and none of the values involved in the processing continuously increase, so the program can run for ever without running out of memory or exceeding the numerical limits of the programming language. Since the complexity is constant in time, provided each observation is incorporated with the time interval, δt, then the model can run for an indefinite period of time.

For statistically stationary data, the VB algorithm is guaranteed to converge to a local optimum [10]. However, where the underlying true system is

not statistically stationary, proving convergence is challenged by the fact that divergence might be caused by a failure of the model or as a result of the non-stationarity of the true system. In section 7 it was observed that in only 3% of the 1,000 synthetic tests, the model failed to track the signal. This is most likely to happen when the angular frequency is at or very close to zero, and in these cases the whole signal has been captured by the noise variables.

Future work is focussing on how the parameters of the priors introduced in the full model (i.e. those in (64) and (68)–(69)) should be selected and the precise effect of the values. This is essentially a multi-objective optimisation problem, trading off certainty *vs* flexibility for each variable, and certainty between the different variables.

The methods introduced in this paper are demonstrated on the simplest possible model, of a noisy sinusoidal signal. However, they are much more widely applicable. For example, a continuous wave lidar measures a three-dimensional wind vector at a single point in space by rotating a lidar beam in a cone around the point and recording one-dimensional measurements around the disk centred on that point. At present the devices take perhaps 100 measurements per rotation and then fit the parameters of the known mathematical model, from which the wind vector is then extracted. The devices are run continously for long periods of time (possibly years). With the method introduced in this paper, each individual measurement is incorporated in real time, giving a higher temporal resolution, and the model allows for outliers in the data by assuming a Student-t distribution of the noise (paper in preparation).

Though not particularly tuned, the Matlab program running on a standard Microsoft Windows laptop processes 100,000 observations in approximately 9 s, which is a rate of more than 11 kHz. In other words, as long as the observation time interval, δt, is greater than 0.000 09 s, the model will run for an indefinitely long period of time, incorporating each observation as it arrives.

References

[1] J. Christmas, R. Everson, Robust autoregression: Student-t innovations using variational Bayes, IEEE Transactions on Signal Processing 59 (1) (2011) 48–57.

[2] H. Attias, A variational Bayesian framework for graphical models, Advances in Neural Information Processing Systems 12 (2000) 209–215.

[3] D. Mackay, Ensemble learning and evidence maximisation, Tech. rep., Cavendish Laboratory, University of Cambridge (1995).

[4] D. Mackay, Ensemble learning for hidden Markov models, Tech. rep., Cavendish Laboratory, University of Cambridge (1997).

[5] M. Jordan, Z. Ghahramani, T. Jaakkola, L. Saul, An introduction to variational methods for graphical models, Machine Learning 37 (2) (1999) 183–233.

[6] H. Lappalainen, J. Miskin, Ensemble learning, in: Advances in Independent Component Analysis, Springer-Verlag, Berlin, 2000, pp. 75–92.

[7] C. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2006.

[8] M. Beal, Z. Ghahramani, The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures, in: Bayesian Statistics, Vol. 7, Oxford University Press, 2002, pp. 453–464.

[9] M. Beal, Variational algorithms for approximate Bayesian inference, Ph.D. thesis, University College London (2003).

[10] Z. Ghahramani, M. Beal, Propagation algorithms for variational Bayesian learning, in: Advances in Neural Information Processing Systems, Vol. 13, MIT Press, 2001, pp. 507–513.

[11] J. Wang, W. Shao, X. Zhang, Z. Song, Dynamic variational Bayesian Student's T mixture regression with hidden variables propagation for industrial inferential sensor development, IEEE Transactions on Industrial Informatics 17 (8) (2021) 5314–5324.

[12] A. Honkela, H. Valpola, On-line variational Bayesian learning, in: 4th International Symposium on Independent Component Analysis and Blind Signal Separation, 2003, pp. 803–808.

[13] Z. Li, O. Rosen, F. Ferrarelli, R. Krafty, Adaptive Bayesian spectral analysis of high-dimensional nonstationary time series, Journal of Computational and Graphical Statistics 0 (0) (2021) 1–14.

[14] A. Masegosa, D. Ramos-López, A. Salmerón, H. Langseth, T. Nielsen, Variational inference over nonstationary data streams for exponential family models, Mathematics 8 (11) (2020) 1942.

[15] M. Hoffman, F. Bach, D. Blei, Online learning for Latent Dirichlet Allocation, in: J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, A. Culotta (Eds.), Advances in Neural Information Processing Systems 23, Curran Associates, Inc., 2010, pp. 856–864.

[16] M. Hoffman, D. Blei, C. Wang, J. Paisley, Stochastic variational inference, Journal of Machine Learning Research 14 (1) (2013) 1303–1347.

[17] M. Johnson, A. Willsky, Stochastic variational inference for bayesian time series models, in: International Conference on Machine Learning, 2014, pp. 1854–1862.

[18] T. Broderick, N. Boyd, A. W. A. Wilson, M. Jordan, Streaming Variational Bayes, in: C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26, Curran Associates, Inc., 2013, pp. 1727–1735.

[19] Z. Yang, L. Yao, Z. Ge, Streaming parallel variational Bayesian supervised factor analysis for adaptive soft sensor modeling with big process data, Journal of Process Control 85 (2020) 52–64.

[20] J. Luts, T. Broderick, M. Wand, Real-time semiparametric regression, Journal of Computational and Graphical Statistics 23 (3) (2014) 589–615.

[21] L. Zhang, W. Wei, Y. Zhang, C. Shen, A. Van Den Hengel, Q. Shi, Cluster sparsity field: An internal hyperspectral imagery prior for reconstruction, International Journal of Computer Vision 126 (8) (2018) 797–821.

[22] D. Wipf, D. Bhaskar, S. Nagarajan, Latent variable Bayesian models for promoting sparsity, IEEE Transactions on Information Theory 57 (9) (2011) 6236–6255.

[23] E. Jaynes, Bayesian spectrum and chirp analysis, in: Proceedings of the Third Workshop on Maximum Entropy and Bayesian Method, Laramie, Wyoming, 1983.

[24] G. Bretthorst, Excerpts from Bayesian spectrum analysis and parameter estimation, Maximum Entropy and Bayesian Methods in Science and Engineering 1 (1988) 75–145.

[25] P. Djurić, Simultaneous detection and frequency estimation of sinusoidal signals (1993).

[26] M. Badiu, T. Hansen, B. Fleury, Variational Bayesian inference of line spectra, IEEE Transactions on Signal Processing 65 (9) (2017) 2247–2261.

[27] L. Dou, R. Hodgson, Bayesian inference and Gibbs sampling in spectral analysis and parameter estimation. I, Inverse Problems 11 (5) (1995) 1069–1085.

[28] C. Andrieu, A. Doucet, Joint Bayesian model selection and estimation of noisy sinusoids via Reversible Jump MCMC, IEEE Transactions on Signal Processing 47 (10) (1999) 2667–2676.

[29] H. Fu, P. Kam, MAP/ML estimation of the frequency and phase of a single sinusoid in noise, IEEE Transactions on Signal Processing 55 (3) (2007) 834–845.

[30] R. Turner, M. Sahani, Probabilistic amplitude and frequency demodulation, Advances in Neural Information Processing Systems (2011) 981–989.

[31] J. Nielsen, M. Christensen, A. Cemgil, S. Godsill, S. Jensen, Bayesian interpolation and parameter estimation in a dynamic sinusoidal model, IEEE Transactions on Audio Speech and Language Processing 19 (7) (2011) 1986–1998.

[32] J. Nielsen, Sinusoidal parameter estimation - a Bayesian approach, Master's thesis, Aalborg University (2009).

[33] A. Quinn, J. Barbot, P. Larzabal, The Bayesian inference of phase, in: Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 2011, pp. 4276–4279.

[34] T. Tsiligkaridis, K. Forsythe, A sequential Bayesian inference framework for blind frequency offset estimation, in: Proceedings of the 2015 IEEE International Workshop on Machine Learning for Signal Processing, Boston, USA, 2015.

[35] J. Taghia, Z. Ma, A. Leijon, Bayesian estimation of the von-Mises Fisher mixture model with variational inference, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (9) (2014) 1701–1715.

[36] K. Mardia, P. Jupp, Directional statistics, John Wiley & Sons, Ltd, Chichester, 2000.

[37] R. Gatto, S. Jammalamadaka, The generalized von Mises distribution, Statistical Methodology 4 (2007) 341–353.

[38] V. Maksimov, Necessary and sufficient statistics for the family of shifts of probability distributions on continuous bicompact groups, Theory of Probability and its Applications 12 (1967) 267–280, (translated by A.R. Kraiman).

[39] V. Maksimov, Necessary and sufficient statistics for the family of shifts of probability distributions on continuous bicompact groups, Theoria Verojatna 12 (1967) 307–321, (in Russian).

[40] J. Christmas, Bayesian spectral analysis with Student-t noise, IEEE Transactions on Signal Processing 62 (11) (2014) 2871–2878.

[41] J. Cooley, J. Tukey, An algorithm for the machine computation of complex Fourier series, Mathematical Computation 19 (1965) 297–301.

[42] J. Fourier, Théorie Analytique de la Chaleur, F. Didot, Paris (English translation: The Analytical Theory of Heat, translated by Alexander Freeman, Cambridge University Press, 1878), 1822.

[43] F. Harris, On the use of windows for harmonic analysis with the Discrete Fourier Transform, Proceedings of the IEEE 66 (1) (1978) 51–83.

[44] M. Menne, I. Durre, B. Korzeniewski, S. McNeal, K. Thomas, X. Yin, S. Anthony, R. Ray, R. Vose, B.E.Gleason, T. Houston, Global Historical Climatology Network – Daily (GHCN-Daily), version 3.26, `http://doi.org/10.7289/V5D21VHZ`, last accessed 1 May 2020 (Nov. 2012).