2023 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2023, ROME, ITALY

GAUSSIAN PROCESS AMPLITUDE DEMODULATION BY MESSAGE-PASSING

Hoang H.M. Nguyen[1] *İsmail Şenöz*[1,2] *Bert de Vries*[1,2,3]

[1]Department of Electrical Engineering, Eindhoven University of Technology
[2]Lazy Dynamics
[3]GN Hearing
{m.h.n.hoang, i.senoz, bert.de.vries}@tue.nl

ABSTRACT

Gaussian Process Amplitude Modulation (GPAM) is a probabilistic model that assigns Gaussian Process priors to the modulator and the carrier and allows us to solve the amplitude demodulation (AD) problem by using inference methods in probability theory. Inference in GPAM results in Gaussian Process Probabilistic Amplitude Demodulation (GP-PAD). However, the mostly used inference technique for GP-PAD is maximum a posteriori (MAP), a point estimate method that is not entirely representative of Bayesian methods in general. In this paper, we provide a full Bayesian inference approach to GP-PAD model. More specifically, we represent the GP-PAD model as a factor graph and use message-passing rules, namely Belief Propagation (BP) and Expectation Propagation (EP), to infer the marginal posteriors of the modulator and the carrier. Furthermore, we employ the Kalman smoothing solution to temporal GP regression models to achieve fast inference for GP models. We compare our approach to the baseline, popular demodulation methods in synthetic and real data experiments. The result shows that our method outperforms the baseline methods and converges.

Index Terms— Probabilistic amplitude demodulation, Gaussian Processes, Belief Propagation, Expectation Propagation, Gaussian linear-state space model

1 Introduction

Amplitude demodulation (AD) is a signal processing problem in which we would like to decompose a signal into a product of a positive, slowly varying envelope (modulator) and a quickly varying signal (carrier). This is an ill-posed problem that cannot be solved without prior assumptions about the modulator or the carrier. In [1] Turner introduced a probabilistic generative process, called *Gaussian Process Probabilistic Amplitude Demodulation* (GP-PAD), that assigns Gaussian Process priors to the modulator and the carrier and thus opens the way for Bayesian inference methods to address the AD problem. The GP-PAD model is shown to outperform traditional AD methods such as Hilbert Envelope (HE) [2] and Square and Low-pass filter (SLP) [3] in [1] and has been widely applied to many audio signal processing tasks such as speech synthesis, speech rhythm analysis, etc. [4–6]. However, maximum a posteriori (MAP) is mostly used to perform the inference procedure in GP-PAD. This approach is a point estimate method and not entirely representative of Bayesian inference in general.

In this paper, we provide a full Bayesian inference approach to the GP-PAD model. More specifically, we represent the GP-PAD as a factor graph and employ message-passing techniques, namely Belief Propagation (BP) [7] and Expectation Propagation (EP) [8] to perform inference. Furthermore, to achieve a fast inference for the GP models, we utilize the Kalman smoothing solution to temporal GP regression models [9–11], a technique that transforms a GP into a linear Gaussian state-space model and solves GP regression by Kalman filtering and Rauch–Tung–Striebel (RTS) smoothing. The computational complexity of this method grows as $\mathcal{O}(N)$, where N is the number of observations, which is very suitable to handle sound signals with thousands of samples in a few seconds. The contributions of the paper are as follows:

- We solve the AD problem by message-passing on factor graph in Section 2.3 via Kalman filtering solution to temporal Gaussian processes.
- We validate our approach in Section 3 by two experiments and compare the results to the baseline methods HE and SLP.

2 GP-PAD model

The amplitude demodulation task concerns the problem of decomposing a signal $y(\tau)$ into the product of a modulator (or envelope) $a(\tau)$ and a carrier $c(\tau)$. The GP-PAD [1] model assumes that $a(\tau)$ is produced by taking a real-valued process $x(\tau)$ through a positive non-linear mapping. The model further assumes that the process $x(\tau)$ and the carrier $c(\tau)$ are Gaussian Processes (GPs). From these assumptions, we can write the GP-PAD model as follows

$$\begin{aligned} x(\tau) &\sim \mathcal{GP}(0, k_x(\tau, \tau')), \\ c(\tau) &\sim \mathcal{GP}(0, k_c(\tau, \tau')), \\ a(\tau) &= \exp(x(\tau)), \\ y(\tau) &= a(\tau)\, c(\tau), \end{aligned} \tag{1}$$

979-8-3503-2411-2/23/$31.00 ©2023 IEEE

where $k_x(\cdot,\cdot)$ and $k_c(\cdot,\cdot)$ refer to the kernel functions of $x(\tau)$ and $c(\tau)$ respectively, each with their own hyper-parameters. Often, we acquire noisy observations at discrete time intervals via a sampling mechanism. We denote the discretized index by t and define $y_t \triangleq y(\tau_t)$ with discretization step $\Delta\tau_t \triangleq \tau_t - \tau_{t-1}$. We can augment the continuous time GP-PAD specification (1) with a likelihood to reflect uncertainties in the measurements. In the original work [1], instead of exponential non-linearity

Solving the demodulation problem is equivalent to implementing Bayesian inference, which uses Bayes theorem to compute the posterior distributions of $x(\tau)$ and $c(\tau)$ [12]. In [1] Turner has used MAP to approximate the values of $x(\tau)$ and $c(\tau)$, and has also resorted to a Laplace's method to estimate the uncertainties in the results. Although this way of addressing the inference problem works well in practice, it consists of ad-hoc methods to Bayesian inference and does not provide probability distribution functions for the posteriors of $x(\tau)$ and $c(\tau)$.

In the following subsections, we will show how a message-passing framework on a factor graph can efficiently implement the inference procedure amounting to a probabilistic demodulation process.

2.1 Gaussian processes

Let us first briefly recap the concept of Gaussian processes. A stochastic process is called a Gaussian process (GP) if a finite set of its function values follows a joint Gaussian distribution [13]. Specifically, let $f \sim \mathcal{GP}(m(\tau), k(\tau,\tau'))$ be a GP with a mean function $m(\cdot)$ and a kernel function $k(\cdot,\cdot)$, then the collection of N function values $\boldsymbol{f} = (f(\tau_1),\ldots,f(\tau_N))^\top$ evaluated at the inputs $\boldsymbol{\tau} = (\tau_1,\ldots,\tau_N)^\top$ satisfies that

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{m}, \mathbf{K}) \tag{2}$$

where $\boldsymbol{m} = (m(\tau_1),\ldots,m(\tau_N))^\top$ and $\mathbf{K}_{ij} = k(\tau_i,\tau_j)$. In regression problems, we can predict the function value f_* at a new input τ_* by computing the predictive distribution

$$f_* \sim \mathcal{N}(m_*, k_*), \tag{3}$$

where

$$m_* = m(\tau_*) + \boldsymbol{k}(\boldsymbol{\tau},\tau_*)\mathbf{K}^{-1}(\boldsymbol{f} - \boldsymbol{m}), \tag{4}$$

$$k_* = k(\tau_*,\tau_*) - \boldsymbol{k}(\tau_*,\boldsymbol{\tau})\mathbf{K}^{-1}\boldsymbol{k}(\boldsymbol{\tau},\tau_*). \tag{5}$$

The inverse of the covariance matrix $\mathbf{K}$ in (4) and (5) has a cubic computational complexity with respect to the number of observations, i.e., $\mathcal{O}(N^3)$. This becomes a severe problem when we work with sound signals, which usually have thousands of samples in a few seconds. Turner [1] has got around this obstacle smartly by using a circulant matrix trick. In this paper we employ the Kalman smoothing solution to temporal GP regression models [9–11] to tackle the problem.

2.2 Kalman smoothing solution to temporal GP regression

For certain classes of kernel functions, Gaussian Process can be regarded as the solution of an mth-order linear stochastic differential equation (SDE) [9–11]

$$a_0 f(\tau) + a_1 \frac{df(\tau)}{d\tau} + \cdots + a_{m-1}\frac{d^{m-1}f(\tau)}{d\tau^{m-1}} + \frac{d^m f(\tau)}{d\tau^m} = w(\tau) \tag{6}$$

where $w(\tau)$ is a zero-mean white noise process with spectral density Q_w. If we define a vector-valued function

$$\boldsymbol{f}(\tau) = \left(f(\tau), \frac{df(\tau)}{d\tau}, \ldots, \frac{d^{m-1}f(\tau)}{d\tau^{m-1}}\right)^\top \tag{7}$$

then we can rewrite (6) as a first-order linear SDE

$$\frac{d\boldsymbol{f}(\tau)}{d\tau} = \mathbf{F}\boldsymbol{f}(\tau) + \mathbf{L}w(\tau) \tag{8}$$

where

$$\mathbf{F} = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -a_0 & -a_1 & \ldots & -a_{m-1} \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}. \tag{9}$$

Extracting $f(\tau)$ from $\boldsymbol{f}(\tau)$ can be easily done by a linear operation

$$f(\tau) = \mathbf{H}\,\boldsymbol{f}(\tau), \tag{10}$$

where $\mathbf{H} = (1\ 0\ \ldots\ 0)$. In this paper, we use the Matern-52 kernel [13] for GPs. The kernel is defined as follows

$$k(r) = \sigma_M^2 \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right)\exp\left(-\frac{\sqrt{5}r}{l}\right), \tag{11}$$

where l and σ_M are the length-scale and the standard deviation of the kernel, respectively, and $r = |\tau - \tau'|$. The corresponding values of $\mathbf{F}$ and Q_w are

$$\mathbf{F} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\lambda^3 & -3\lambda^2 & -3\lambda \end{pmatrix}, \quad Q_w = \frac{16}{3}\sigma_M^2\,\lambda^5, \tag{12}$$

where $\lambda = \sqrt{5}/l$.

From (8) and (10) we can define a discrete state-space model as follows [9–11]

$$\begin{aligned} \boldsymbol{f}_t &= \mathbf{A}_t\,\boldsymbol{f}_{t-1} + \boldsymbol{q}_t, \quad \boldsymbol{q}_t \sim \mathcal{N}(0, \mathbf{Q}_t) \\ y_t &= \mathbf{H}\,\boldsymbol{f}_t + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0,\sigma^2) \end{aligned} \tag{13}$$

where the transition matrix $\mathbf{A}_t$ is computed as

$$\mathbf{A}_t = \exp(\mathbf{F}\,\Delta\tau_t), \quad \Delta\tau_t = \tau_t - \tau_{t-1}, \tag{14}$$

With the use of Matern kernel, the noise covariance matrix $\mathbf{Q}_t$ is found by the following formula

$$\mathbf{Q}_t = \mathbf{P}_\infty - \mathbf{A}_t\mathbf{P}_\infty\mathbf{A}_t^\top. \tag{15}$$

where $\mathbf{P}_\infty$ is called the stationary covariance of $\boldsymbol{f}(\tau)$. This is because the Matern-52 kernel is stationary and defines a stationary state $\boldsymbol{f}_\infty \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_\infty)$. The matrix $\mathbf{P}_\infty$ can be found by solving the following Lyapunov differential equation

$$\frac{d\mathbf{P}_\infty}{d\tau} = \mathbf{F}\,\mathbf{P}_\infty + \mathbf{P}_\infty\,\mathbf{F}^\top + \mathbf{L}\,Q_w\,\mathbf{L}^\top = 0, \tag{16}$$

which gives the solution

$$\mathrm{vec}\,(\mathbf{P}_\infty) = (\mathbf{I}\otimes\mathbf{F} + \mathbf{F}\otimes\mathbf{I})^{-1}\ \mathrm{vec}\left(-\mathbf{L}\,Q_w\,\mathbf{L}^\top\right), \tag{17}$$

where vec denotes the vectorization operator and $\otimes$ denotes the Kronecker product.

By expressing GPs as the state-space model (13), we can solve the GP regression problem by Kalman filtering and RTS smoothing, which has computational complexity only propor-

tional to $\mathcal{O}(N)$. For a more rigorous and comprehensive explanation of the Kalman smoothing solution to temporal GP regression, we refer to [9–11].

2.3 GP-PAD inference by message-passing

In this section, we show how the GP-PAD model can be solved by inference with messages passing on a factor graph. This paper uses Forney-style Factor Graph (FFG) [14–16] to represent models. An FFG is a graphical model that visualizes the factorization of a function as a graph with nodes and edges, where each node represents a factor (i.e. function), each edge represents a variable, and an edge connects to a node if and only if the variable on that edge is an argument of the function inside that node. For more details about FFG, we refer to [15, 16].

Recall that the modulator x_t and the carrier c_t are assigned to GP priors. Since GPs can be expressed as state-space models, we can rewrite the GPs in (1) as follows

$$c_t = \mathbf{H}\,\boldsymbol{z}_t + \epsilon_c, \quad \epsilon_c \sim \mathcal{N}(0, \sigma_c^2), \tag{18}$$

$$x_t = \mathbf{H}\,\boldsymbol{f}_t + \epsilon_x, \quad \epsilon_x \sim \mathcal{N}(0, \sigma_x^2), \tag{19}$$

$$\boldsymbol{z}_t = \mathbf{A}_z\,\boldsymbol{z}_{t-1} + \boldsymbol{q}_z, \quad \boldsymbol{q}_z \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_z), \tag{20}$$

$$\boldsymbol{f}_t = \mathbf{A}_f\,\boldsymbol{f}_{t-1} + \boldsymbol{q}_f, \quad \boldsymbol{q}_f \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_f), \tag{21}$$

where $\mathbf{A}_z, \mathbf{Q}_z$ are matrices corresponding to the process $z(\tau)$ and are defined in (14) and (15), respectively. Similarly, $\mathbf{A}_f$ and $\mathbf{Q}_f$ correspond to the process $f(\tau)$ and have the same computation as that of $z(\tau)$. Now we can rewrite the generative model of GP-PAD in state space form

$$p(\boldsymbol{y}, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{x}, \boldsymbol{f}, \boldsymbol{z}) = p(\boldsymbol{f}_0)p(\boldsymbol{z}_0)\prod_{t=1}^{T} p(y_t|a_t, c_t)p(a_t|x_t) \\ p(x_t|\boldsymbol{f}_t)p(\boldsymbol{f}_t|\boldsymbol{f}_{t-1})p(c_t|\boldsymbol{z}_t)p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}) \tag{22}$$

where the factors of (22) are defined as below[1]

$$p(y_t|a_t, c_t) = \delta(y_t - a_t\,c_t), \tag{23}$$

$$p(a_t|x_t) = \delta(a_t - \exp(x_t)), \tag{24}$$

$$p(x_t|\boldsymbol{f}_t) = \mathcal{N}(x_t|\mathbf{H}\,\boldsymbol{f}_t, \sigma_x^2), \tag{25}$$

$$p(\boldsymbol{f}_t|\boldsymbol{f}_{t-1}) = \mathcal{N}(\boldsymbol{f}_t|\boldsymbol{A}_f\,\boldsymbol{f}_{t-1}, \mathbf{Q}_f), \tag{26}$$

$$p(c_t|\boldsymbol{z}_t) = \mathcal{N}(c_t|\mathbf{H}\,\boldsymbol{z}_t, \sigma_c^2), \tag{27}$$

$$p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}) = \mathcal{N}(\boldsymbol{z}_t|\boldsymbol{A}_z\,\boldsymbol{z}_{t-1}, \mathbf{Q}_z), \tag{28}$$

$$p(\boldsymbol{f}_0) = \mathcal{N}(\boldsymbol{f}_0|\mathbf{0}, \boldsymbol{P}_{\infty f}), \tag{29}$$

$$p(\boldsymbol{z}_0) = \mathcal{N}(\boldsymbol{z}_0|\mathbf{0}, \boldsymbol{P}_{\infty z}). \tag{30}$$

The factor graph of (22) at the time t is shown in Fig. 1 with messages on edges. The inference procedure amounts to finding the posterior distributions $p(\boldsymbol{f}_t|\boldsymbol{y})$ and $p(\boldsymbol{z}_t|\boldsymbol{y})$, and in factor graph framework these distributions can be computed by multiplying all messages on the edges, i.e.

$$p(\boldsymbol{f}_t|\boldsymbol{y}) = \vec{\mu}_1(\boldsymbol{f}_t)\,\overleftarrow{\mu}_2(\boldsymbol{f}_t)\,\overleftarrow{\mu}_3(\boldsymbol{f}_t) \tag{31}$$

$$p(\boldsymbol{z}_t|\boldsymbol{y}) = \vec{\mu}_1(\boldsymbol{z}_t)\,\overleftarrow{\mu}_2(\boldsymbol{z}_t)\,\overleftarrow{\mu}_3(\boldsymbol{z}_t). \tag{32}$$

[1] All $\mathbf{A}$ and $\mathbf{Q}$ matrices change with time, but we omit the time dependence for brevity.

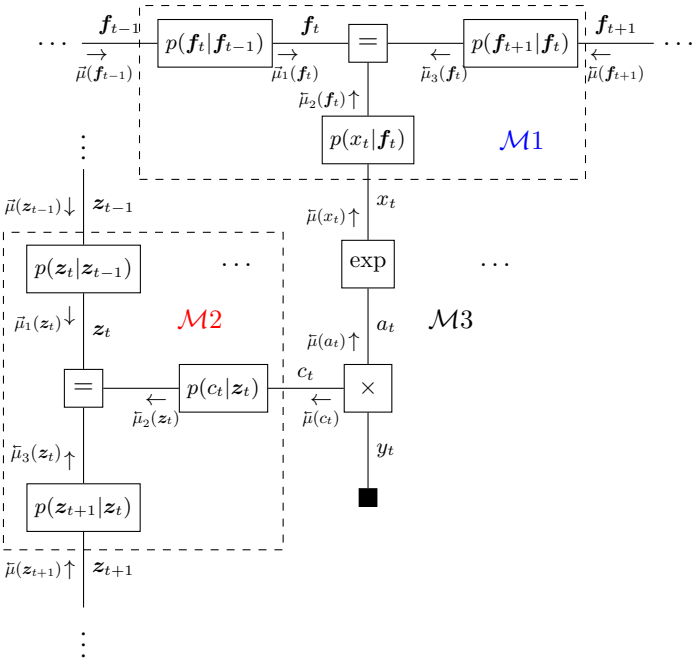


Fig. 1: The FFG of the GP-PAD model (22).

For the convenience of message computation, we split the graph into three sub-graphs $\mathcal{M}1$, $\mathcal{M}2$ and $\mathcal{M}3$ as shown in Fig. 1. We notice that both $\mathcal{M}1$ and $\mathcal{M}2$ encapsulate linear Gaussian state-space models since all the factors inside them are Gaussian distributions. Consequently, the messages in (31) and (32) are Gaussian and can be computed by the sum-product rule or Belief Propagation [7]. The difficult part lies in the sub-graph $\mathcal{M}3$, which contains non-Gaussian factors, namely the exponential function and the multiplication function, yielding non-Gaussian messages that will lead to solutions that are not in closed form. Therefore, we apply Expectation Propagation (EP) [8] method to approximate messages in $\mathcal{M}_3$ as Gaussian.

Now we elaborate on the computation of the messages. The sum-product update rules for $\vec{\mu}_1(\boldsymbol{f}_t)$, $\vec{\mu}_1(\boldsymbol{z}_t)$, $\overleftarrow{\mu}_3(\boldsymbol{f}_t)$ and $\overleftarrow{\mu}_3(\boldsymbol{z}_t)$ are computed in [16]. The update rules for $\overleftarrow{\mu}_2(\boldsymbol{f}_t)$ and $\overleftarrow{\mu}_2(\boldsymbol{z}_t)$ are

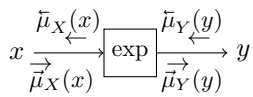
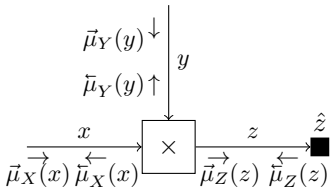
$$\overleftarrow{\mu}_2(\boldsymbol{f}_t) = \int \overleftarrow{\mu}(x_t)p(x_t|\boldsymbol{f}_t)dx_t \tag{33}$$

$$\overleftarrow{\mu}_2(\boldsymbol{z}_t) = \int \overleftarrow{\mu}(c_t)p(c_t|\boldsymbol{z}_t)dc_t. \tag{34}$$

Since $p(x_t|\boldsymbol{f}_t)$ and $p(c_t|\boldsymbol{z}_t)$ are Gaussian distributions, the above equations yield Gaussian messages only if $\overleftarrow{\mu}(x_t)$ and $\overleftarrow{\mu}(c_t)$ are Gaussian. However, the messages come from an exponential factor node and a multiplication factor node. Therefore, their functional forms are non-Gaussian. The specific message update rules[2] for the exponential and multiplication nodes are provided in Table.1. Note that we have used the result of product of two Gaussian random variables in [17]

[2] All derivations of the rules are available at this link.

Table 1: Sum-product update rules for Exponential and Multiplication nodes. $K_v(\cdot)$ denotes the modified Bessel function of the second kind of order v.

Node	Incoming messages	Outgoing messages
 $f(x,y) = \delta(y - \exp(x))$	$\vec{\mu}_X(x) = \mathcal{N}(x\|m_x, v_x)$	$\vec{\mu}_Y(y) = \text{Lognormal}(y\|m_x, v_x)$
	$\overleftarrow{\mu}_Y(y)$	$\overleftarrow{\mu}_X(x) = \overleftarrow{\mu}_Y(\exp(x))$
 $f(x,y,z) = \delta(z - x\,y)$	$\vec{\mu}_X(x) = \mathcal{N}(x\|m_x, \sigma_x^2)$ $\vec{\mu}_Y(y) = \mathcal{N}(y\|m_y, \sigma_y^2)$	$\vec{\mu}_Z(z) = \exp\left(-\frac{1}{2}\left(\frac{m_x^2}{\sigma_x^2} + \frac{m_y^2}{\sigma_y^2}\right)\right) \times$ $\sum_{n=0}^{\infty}\sum_{m=0}^{2n} \frac{z^{2n-m}\lvert z\rvert^{m-n}\sigma_x^{m-n-1}}{\pi\,(2n)!\,(\sigma_y)^{m-n+1}} \left(\frac{m_x}{\sigma_x^2}\right)^m \times$ $\binom{n}{r}\left(\frac{m_y}{\sigma_y^2}\right)^{2n-m} K_{m-n}\left(\frac{\lvert z\rvert}{\sigma_x\,\sigma_y}\right)$
	$\vec{\mu}_Y(y)$ $\overleftarrow{\mu}_Z(z) = \delta(z - \hat{z})$	$\overleftarrow{\mu}_X(x) = \frac{1}{\lvert x\rvert}\vec{\mu}_Y\left(\frac{\hat{z}}{x}\right)$

for the computational rule of the outgoing message at the multiplication node. To address the non-Gausianity problem, we resort to EP to approximate these messages as Gaussian. Fortunately, in factor graph framework the EP method can be implemented by specifying local constraints around nodes as described in [18]. Following [18], we can achieve EP messages for the exponential node and the multiplication node, and this also guarantees the convergence of our approach.

3 Experiments

In this section, we perform two experiments with the GP-PAD model on synthetic and real data and compare its results with the two baseline methods, namely Square and Low-Pass filter (SLP) and Hilbert Envelope (HE). We use Signal-to-Noise ratio (SNR) in decibels to compare the performance of the three methods on the synthetic data. To illustrate the convergence of our method we employ Bethe Free Energy (BFE) [19] quantity, an approximation of negative log-evidence. All experiments are performed by scientific programming language `Julia` [20] with the programming probabilistic package `RxInfer.jl` [21] for inference. All experiments are available at `github.com/MLSP-2023/GPAD`.

3.1 Synthetic data

Table 2: SNR of the three methods.

	HE	SLP	PAD
Envelope SNR (dB)	7.062	13.398	15.677
Carrier SNR (dB)	5.970	10.443	12.689

We use the GP-PAD model (1) to generate synthetic data. Both GP priors assigned to the modulator and the carrier have zero-mean function and Matern-52 kernel. The kernel of the GP modulator has length-scale $l_m = 1$, variance $\sigma_{Mm}^2 = 1$, while the kernel of the GP carrier has $l_c = 0.01$ and $\sigma_{Mc}^2 = 0.5$. The generated data is shown in Fig. 3a, which includes 500 samples in 5 second. For the model (22), we set $\sigma_x^2 = \sigma_c^2 = 0.1$. The inference of GP-PAD is run with 20 iterations.

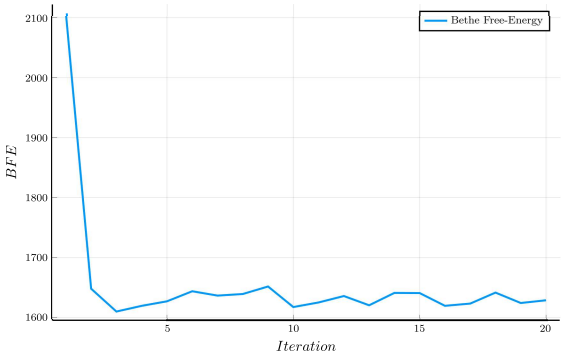
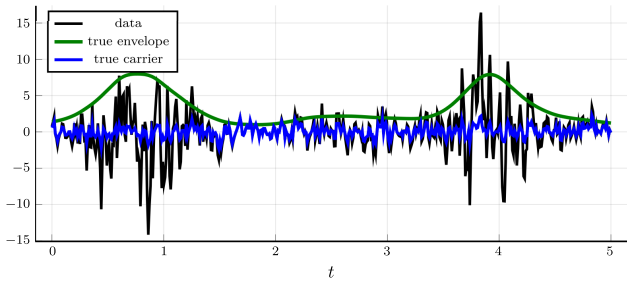


Fig. 2: The evolution of Bethe Free Energy during inference in the experiment with synthetic data.

The result is shown in Fig.4. Intuitively, the HE method has the poorest performance as expected since HE only works well with sinusoidal signal. Meanwhile the SLP and the GP-PAD can recover pretty well the true modulator and carrier. Indeed, the SNR results in Table. 2 shows that the HE method gives the smallest SNR value (7.062dB for envelope), while those numbers of SLP and GP-PAD are 13.398 dB and 15.677 dB, respectively. The numbers also tell us GP-PAD has the best performance. The SNR values for the carrier also have the same behaviour. Furthermore, Fig. 2 shows the evolution of BFE value during inference and we can observe a convergence. Although there are fluctuations, this behaviour is expected since EP relies on sampling methods for approxima-

Fig. 3: On the left plot the green curve is the envelope, the blue curve is the carrier, and the black curve is the modulated data. Results of estimation on this data are given in Fig. 4. On the right plot, we have 1 second of an audio recording of a bubbling water sound.

(a) Synthetic data generation process.



(b) Bubbling water sound data.

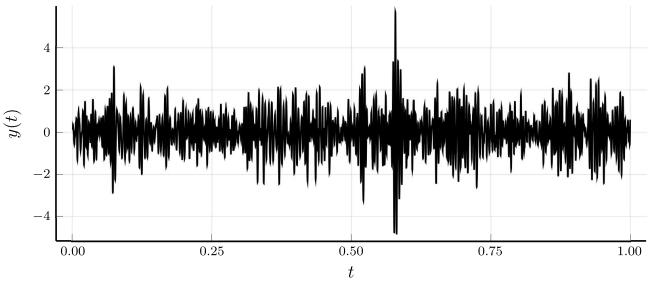
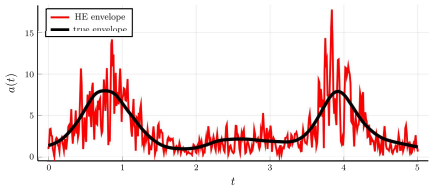
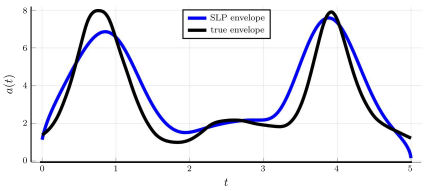


Fig. 4: Results of estimates by HE, SLP, and PAD methods on the synthetic data Fig. 3a. We color code the estimates returned by HE with red, SLP with blue, and PAD with green. Shaded regions for the PAD estimates correspond to standard deviation.

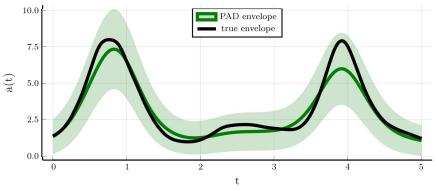
(a) Envelope estimate by Hilbert method



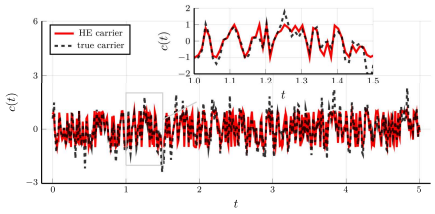
(b) Envelope estimate by SLP method



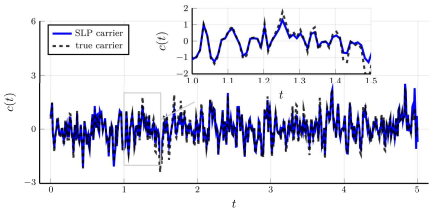
(c) Envelope estimate by PAD method



(d) Carrier estimate by Hilbert method



(e) Carrier estimate by SLP method



(f) Carrier estimate by PAD method

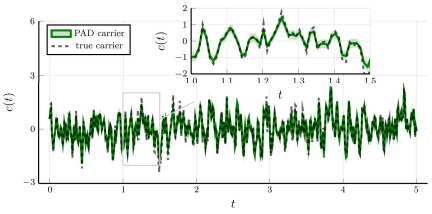
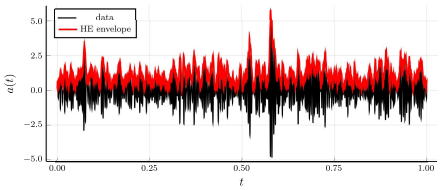
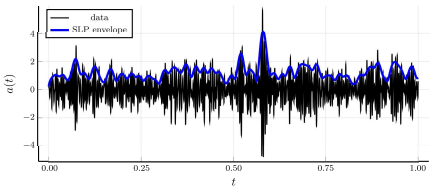


Fig. 5: Results of estimates by HE, SLP, and PAD methods on a bubbling sound data Fig. 3b. We color code the estimates returned by HE with red, SLP with blue, and PAD with green. Shaded regions for the PAD estimates correspond to standard deviation.

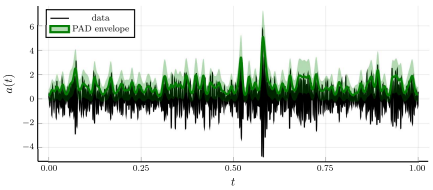
(a) Envelope estimate by Hilbert method



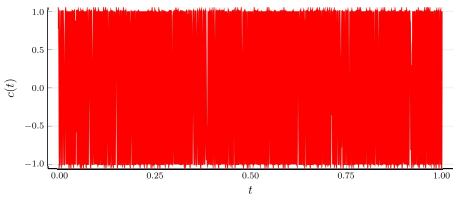
(b) Envelope estimate by SLP method



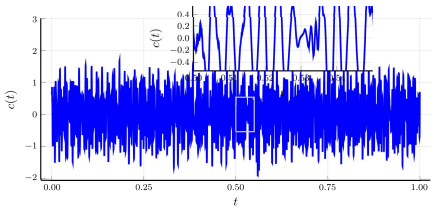
(c) Envelope estimate by PAD method



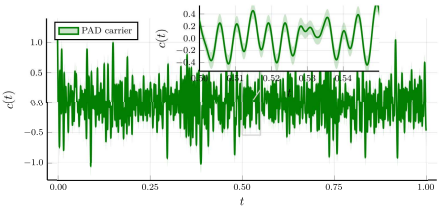
(d) Carrier estimate by Hilbert method



(e) Carrier estimate by SLP method



(f) Carrier estimate by PAD method



tion.

3.2 Real data

In this experiment, we test the GP-PAD model on real data, namely bubbling water sound, which is shown in Fig. 3b. The configuration of GP-PAD is as follows: the kernel of the GP modulator has $l_m = 0.015$, $\sigma^2_{Mm} = 1.$, while the kernel of the GP carrier has $l_c = 0.001$, $\sigma^2_{Mc} = 0.5$; $\sigma^2_x = \sigma^2_c = 0.6$.

The result is shown in Fig.5. The HE method has the poorest performance as in the experiment with synthetic data. Its recovered envelope is simply the peaks of the signal. On the other hand, both SLP and GP-PAD provide smoother envelopes. Furthermore, compared to HE and SLP, GP-PAD can also yield a carrier with a structure, indicating its advantages over the traditional methods [1].

4 Conclusions

This paper applies the complete Bayesian inference approach to the Gaussian Process Amplitude Demodulation (GP-PAD) model in [1] for the amplitude demodulation problem. We have shown that the GP-PAD model can be expressed in terms of factor graphs, and the inference procedure is thus performed by message-passing methods, namely Belief Propagation and Expectation Propagation. We perform two experiments with synthetic and natural data and compare our approach with traditional methods, Hilbert Envelope and Square-Low Pass filter. The experiment result not only shows that our approach outperforms the conventional methods, but also illustrates the convergence of our method.

Nevertheless, our model uses fixed parameters, which is not ideal in real applications. Further improvements should include parameter learning by message-passing framework for our approach.

Acknowledgements

This work is partially financed by contributions from GN Hearing, PPS subsidy from Holland High Tech and the EAISI institute at TU Eindhoven.

5 References

[1] Richard E. Turner, *Statistical Models for Natural Sounds*, Ph.D. thesis, Gatsby Computational Neuroscience Unit, UCL, 2010.

[2] D. Vakman, "On the analytic signal, the Teager-Kaiser energy algorithm, and other methods for defining amplitude and frequency," *IEEE Transactions on Signal Processing*, vol. 44, no. 4, pp. 791–797, Apr. 1996.

[3] Robert Libbey, *Signal and image processing sourcebook*, Springer Science & Business Media, 1994.

[4] Victoria Leong and Usha Goswami, "Impaired extraction of speech rhythm from temporal modulation patterns in speech in developmental dyslexia," *Frontiers in Human Neuroscience*, vol. 8, 2014.

[5] Alexandros Lazaridis, Milos Cernak, and Philip N. Garner, "Probabilistic Amplitude Demodulation Features in Speech Synthesis for Improving Prosody," in *Interspeech 2016*. Sept. 2016, pp. 2298–2302, ISCA.

[6] Tatsuya Daikoku and Usha Goswami, "The Hierarchical Structure of Temporal Modulations in Music is Universal across Genres and matches Infant-Directed Speech," *bioRxiv*, 2020.

[7] Judea Pearl, "Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach," in *Proceedings of the Second AAAI Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, 1982, AAAI'82, pp. 133–136, AAAI Press.

[8] Thomas P. Minka, "Expectation Propagation for Approximate Bayesian Inference," in *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, USA, 2001, UAI'01, pp. 362–369, Morgan Kaufmann Publishers Inc.

[9] Jouni Hartikainen and Simo Sarkka, "Kalman filtering and smoothing solutions to temporal Gaussian process regression models," in *2010 IEEE International Workshop on Machine Learning for Signal Processing*, Kittila, Finland, Aug. 2010, pp. 379–384, IEEE.

[10] Simo Sarkka, Arno Solin, and Jouni Hartikainen, "Spatiotemporal Learning via Infinite-Dimensional Bayesian Filtering and Smoothing: A Look at Gaussian Process Regression Through Kalman Filtering," *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 51–61, July 2013.

[11] Arno Solin, *Stochastic differential equation methods for spatio-temporal Gaussian process regression*, Ph.D. thesis, Aalto University, 2016, Publisher: Aalto University.

[12] R. E. Turner and M. Sahani, "Demodulation as Probabilistic Inference," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2398–2411, Nov. 2011.

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

[14] G.David Forney, "Codes on graphs: normal realizations," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 520–548, Feb. 2001.

[15] Hans-Andrea Loeliger, "An introduction to factor graphs," *Signal Processing Magazine, IEEE*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

[16] Hans-Andrea Loeliger, Justin Dauwels, Junli Hu, Sascha Korl, Li Ping, and Frank R. Kschischang, "The Factor Graph Approach to Model-Based Signal Processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, June 2007.

[17] G. Cui, X. Yu, S. Iommelli, and L. Kong, "Exact Distribution for the Product of Two Correlated Gaussian Random Variables," *IEEE Signal Processing Letters*, vol. 23, no. 11, pp. 1662–1666, Nov. 2016, Conference Name: IEEE Signal Processing Letters.

[18] İsmail Şenöz, Thijs van de Laar, Dmitry Bagaev, and Bert de Vries, "Variational Message Passing and Local Constraint Manipulation in Factor Graphs," *Entropy*, vol. 23, no. 7, pp. 807, July 2021, Publisher: Multidisciplinary Digital Publishing Institute.

[19] Jonathan S Yedidia, William T Freeman, and Yair Weiss, "Bethe free energy, Kikuchi approximations, and belief propagation algorithms," *Advances in neural information processing systems*, vol. 13, pp. 24, 2001.

[20] J. Bezanson, A. Edelman, S. Karpinski, and V. Shah, "Julia: A Fresh Approach to Numerical Computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, Jan. 2017.

[21] Dmitry Bagaev, Albert Podusenko, and Bert De Vries, "RxInfer: A Julia package for reactive real-time Bayesian inference," *Journal of Open Source Software*, vol. 8, no. 84, pp. 5161, 2023.