CHALMERS

Chalmers Publication Library

Backward particle message passing

This document has been downloaded from Chalmers Publication Library (CPL). It is the author´s version of a work that was accepted for publication in:

Proc. IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), June 2015

Citation for the published paper:
Wymeersch, H. ; Irukulapati, N. ; Sackey, I. et al. (2015) "Backward particle message passing". Proc. IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), June 2015

http://dx.doi.org/10.1109/SPAWC.2015.7227078

Downloaded from: http://publications.lib.chalmers.se/publication/224270

Notice: Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source. Please note that access to the published version might require a subscription.

Chalmers Publication Library (CPL) offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all types of publications: articles, dissertations, licentiate theses, masters theses, conference papers, reports etc. Since 2006 it is the official tool for Chalmers official publication statistics. To ensure that Chalmers research results are disseminated as widely as possible, an Open Access Policy has been adopted.
The CPL service is administrated and maintained by Chalmers Library.

(article starts on next page)

Backward Particle Message Passing

Henk Wymeersch, Naga V. Irukulapati, Isaac A. Sackey, Pontus Johannisson, and Erik Agrell
Chalmers University of Technology, Gothenburg, Sweden
e-mail: {henkw,vnaga,pontus.johannisson,agrell}@chalmers.se, isaacs@student.chalmers.se

***Abstract*—Particle methods are an established way to represent messages and perform message passing in factor graphs. Despite their common use, there are several cases for which messages are hard to compute, even in linear models. Building on results from Gaussian message passing, we demonstrate how backward particle-based messages can be computed and describe a practical application in the context of fiber-optical communications.**

I. Introduction

Factor graphs have received considerable attention in the communications and signal processing community, for their ability to express problems and algorithms in a systematic and unified manner [1]. A variety of message passing algorithms can be executed on a factor graph to solve inference problems. Among these methods, the sum–product algorithm (belief propagation) is arguably the most common. However, belief propagation on factor graphs is plagued by two difficulties: (i) the presence of cycles; (ii) representation of messages related to continuous variables. This paper concerns the latter issue.

Representation of messages of continuous variables can be accomplished in a number of ways, including quantizing/gridding the message domain, approximation by distributions from a parametric family (e.g., Gaussian), projection onto a suitable set of basis functions, so that the message representation are the coordinates along the basis functions, and, finally, a particle representation of the message by a sequence of weighted particles. While all methods are widely used, theoretical results have mainly focused on Gaussian representations, in particular for linear models. Particles were considered in [2] for a factor graph-based design of a detector for nonlinear optical communication. However, [2] relied on a conventional matched filter to cope with non-invertible linear transformations, which may not be optimal.

In this paper, in order compare the matched filtering approach from [2] (which we will term "projection") with an approach based on the sum–product algorithm (which we will term "slicing"), we build on [3], [4] and relate several results from [1] to particle representations in linear systems in combination with specific non-linearities. We consider different approaches for the explicit computation of backward messages for linear models using particles. We apply these techniques to the problem considered in [2], indicating that the matched filtering approach from [2] is suboptimal and that slicing leads to improved performance.

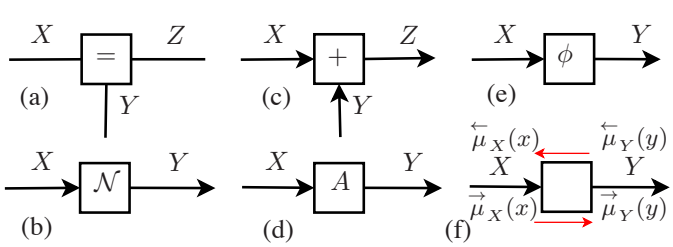

Figure 1. Basic building blocks, including (a) 3-way equality; (b) addition of noise; (c) summation; (d) matrix multiplication; (e) a general nonlinear function. Inset (f) shows leftward and rightward messages.

II. Problem Formulation

We recall that in a factor graph, a function $f(x_1, x_2, \ldots, x_N) \in \mathbb{R}_{\geq 0}$ with arbitrary variables[1] is represented by a factor vertex (generally labeled by the function) and N edges, each labeled with one of the variables (generally in capitals, to distinguish the variable from its value). Given incoming messages $\mu^{\text{in}}_{X_i}(x_i)$, the outgoing message on edge X_j is computed as

$$\mu^{\text{out}}_{X_j}(x_j) \propto \int f(x_1, x_2, \ldots, x_N) \prod_{i \neq j} \mu^{\text{in}}_{X_i}(x_i) \mathrm{d}x_{i \neq j}, \quad (1)$$

where $\propto$ indicates that the message can be normalized.

Basic building blocks/factors include the following [1]:

(a) *3-way equality:*

$$f(x, y, z) = \delta(x - y)\delta(x - z). \quad (2)$$

(b) *Addition of noise:*

$$f(x, y) = \mathcal{N}(y|x, \Sigma), \quad (3)$$

where $\mathcal{N}(y; x, \Sigma)$ denotes a Gaussian distribution in y with mean x and covariance matrix Σ.

(c) *Summation:*

$$f(x, y, z) = \delta(z - (x + y)). \quad (4)$$

(d) *Matrix multiplication:*

$$f(x, y) = \delta(y - Ax), \quad (5)$$

where A is an $N \times M$ matrix.

(e) *Nonlinear function*[2]*:*

$$f(x, y) = \delta(y - \phi(x)), \quad (6)$$

in which we will limit ourselves to functions for which, for any y, $\mathcal{X}(y) = \{x|\phi(x) = y\}$ is either finite or empty.

Factor graphs of these functions are shown in Fig. 1. Note that the large black arrows are part of the representation and

[1]The variables x_i could be real, complex, binary, vectors, etc.

[2]Note that we will designate both $f(x, y)$ and $\phi(x)$ as functions. From the context it should be clear what is meant.

are not indicative of the message flow and that the labels in the vertices are only intended to convey which factor they correspond to (e.g., the vertex labeled "A" corresponds to the function $\delta(y - Ax)$). The small red arrows and the arrows above the message notation refer to message flow (e.g., $\vec{\mu}_X(x)$ for a message flowing from left to right) .

For the first four basic building blocks, when incoming messages (from any direction) are Gaussian, the outgoing messages are also Gaussian and can be computed in closed form [1]. For the nonlinear function, this is no longer true. In the presence of nonlinear functions over continuous variables, Gaussianity is not preserved and other message representations are required. In particular, we consider particle representations where a message, say $\mu_X(x)$, is represented by a list of K samples $x^{(k)}$ and corresponding weights $w_X^{(k)}$, in which $\sum_{k=1}^K w_X^{(k)} = 1$. The representation of $\mu_X(x)$ by $\{w_X^{(k)}, x^{(k)}\}$ should be understood in the sense that for any integrable function $g(x)$

$$\int g(x)\mu_X(x)\mathrm{d}x \approx \sum_{k=1}^{K} w_X^{(k)} g(x^{(k)}). \tag{7}$$

We can thus express the representation as an approximation of the form:

$$\mu_X(x) \approx \sum_{k=1}^{K} w_X^{(k)} \delta(x - x^{(k)}). \tag{8}$$

Our objective is to compute particle representations of outgoing messages (both leftward and rightward), given particle representations of incoming messages for the functions (2)–(6).

III. Message Computation

A. 3-way equality

In the three-way equality, the incoming messages $\mu_X(x)$ and $\mu_Y(y)$ are given in particle form as in (8).[3] Based on (1), the outgoing message is given by

$$\begin{aligned}\mu_Z(z) &= \iint \delta(x-z)\delta(y-z)\mu_X(x)\mu_Y(y)\mathrm{d}x\mathrm{d}y \\ &= \mu_X(z)\mu_Y(z).\end{aligned} \tag{9}$$

The problem now arises that $\mu_X(z)$ and $\mu_Y(z)$ are not available for the same z. The problem can be avoided by smoothing the particle distribution and attaching a kernel to each particle [4]:

$$\tilde{\mu}_X(x) = \sum_{k=1}^{K} w_X^{(k)} h_\sigma(x - x^{(k)}), \tag{10}$$

in which the kernel function $h_\sigma(x)$ can be a Gaussian

$$h_\sigma(x) = \frac{1}{(2\pi)^{N/2}\sigma^N} \exp\left(-\frac{\|x\|^2}{2\sigma^2}\right) = \mathcal{N}(x; 0, \sigma^2 I), \tag{11}$$

where I denotes the identity matrix of proper size and σ is a parameter that should be chosen appropriately. In fact, optimization of σ is nontrivial and results are only known for low-dimensional scenarios. For that reason, σ is often set using Silverman's rule of thumb [5]. Note that the particle representation is recovered when $\sigma \to 0$. Note that (10) can be evaluated for *any* x.

A particle representation of $\mu_Z(z)$ can thus be found as follows:

(1) Draw samples $z^{(1)}, \ldots, z^{(K)}$ from a proposal distribution[4] $q_Z(z)$
(2) Compute weights using the smoothened distributions
$$w_Z^{(k)} \propto \frac{\tilde{\mu}_X(z^{(k)})\tilde{\mu}_Y(z^{(k)})}{q_Z(z^{(k)})}. \tag{12}$$
(3) Normalize the weights so they add up to one. The message $\mu_Z(z)$ can be approximated using weighted particles $\{w_Z^{(k)}, z^{(k)}\}$.

B. Addition of noise

When a particle representation of $\vec{\mu}_X(x)$ is available, we generate an internal particle $e^{(k)} \sim \mathcal{N}(e; 0, \Sigma)$ for each $x^{(k)}$ and set $y^{(k)} = x^{(k)} + e^{(k)}$ to obtain $\vec{\mu}_Y(y)$. Conversely, when a particle representation of $\overleftarrow{\mu}_Y(y)$ is available, we generate an internal particle $e^{(k)} \sim \mathcal{N}(e; 0, \Sigma)$ for each $y^{(k)}$ and set $x^{(k)} = y^{(k)} + e^{(k)}$ to obtain $\overleftarrow{\mu}_X(x)$. Note that we have used the symmetry of the Gaussian distribution and that the particle weights are unchanged.

C. Summation

Let us first consider the case where the incoming messages are $\mu_X(x)$ and $\mu_Z(z)$ and are given in particle form. Assume that $x, y, z \in \mathbb{R}^N$. The summation factor can be expressed $f(x, y, z) = \delta(y - A[z^{\mathrm{T}}\, x^{\mathrm{T}}]^{\mathrm{T}})$, in which A is a $1 \times 2N$ vector of N ones followed by N minus ones.[5] If instead we consider $\mu_X(x)$ and $\mu_Y(y)$ as incoming messages, the summation factor can be expressed as $f(x, y, z) = \delta(z - A[x\, y]^T)$, in which A is a vector of $2N$ ones. In any case, we see that summation is a special case of a linear function, so we will defer the message computation until subsections III-D and III-F.

D. Forward message for (non)linear functions

For a general function $y = \phi(x)$, when the incoming messages $\vec{\mu}_X(x)$ is given, the outgoing message $\vec{\mu}_Y(y)$ is of the form [6]

$$\vec{\mu}_Y(y) = \sum_{x \in \mathcal{X}(y)} \frac{\vec{\mu}_X(x)}{|\phi'(x)|}, \tag{13}$$

in which $\mathcal{X}(y) = \{x | \phi(x) = y\}$ and $|\phi'(x)|$ denotes the absolute value determinant of the Jacobian. In particle representations, the outgoing message $\vec{\mu}_Y(y)$ is found by pushing the particles through the function [3], so that

$$\begin{aligned}\vec{\mu}_Y(y) &= \int \delta(y - \phi(x))\vec{\mu}_X(x)\mathrm{d}x \\ &\approx \sum_{k=1}^{K} w_X^{(k)} \delta(y - \phi(x^{(k)})).\end{aligned} \tag{14}$$

[3]Note that the roles of x, y, and z are interchangeable.

[4]By choosing the proposal distribution as either $\mu_X(x)$ or $\mu_Y(y)$, the computation of the weights can be simplified. The choice of proposal distribution is often application-specific.

[5]Note that, for $\mathbf{1}$ an $N \times 1$ all-ones vector, $[\mathbf{1}^{\mathrm{T}}\ -\mathbf{1}^{\mathrm{T}}][z^{\mathrm{T}}\, x^{\mathrm{T}}]^{\mathrm{T}} = z - x$, so that $\delta(y - A[z^{\mathrm{T}}\, x^{\mathrm{T}}]^{\mathrm{T}}) = \delta(y - (z - x)) = \delta(z - (x + y))$.

In other words, $\{w_Y^{(k)} = w_X^{(k)}, y^{(k)} = \phi(x^{(k)})\}$ forms a particle representation of $\overrightarrow{\mu}_Y(y)$. This result is of course valid for functions of the form $\phi(x) = Ax$, including the summation operation from Section III-C.

E. Backward message for (non)linear functions

For the backward messages (both here and in Section III-F), we will make an additional assumption: *the underlying message $\overleftarrow{\mu}_Y(y)$ from which the particles are obtained is such that $\overleftarrow{\mu}_Y(y) > 0$, for all y.*

For a function of the form $y = \phi(x)$, given a particle representation $\{w_Y^{(k)}, y^{(k)}\}$ of the incoming message $\overleftarrow{\mu}_Y(y)$, the outgoing message, $\overleftarrow{\mu}_X(x)$, is given by

$$\begin{aligned}\overleftarrow{\mu}_X(x) &= \int \delta(y - \phi(x))\overleftarrow{\mu}_Y(y)\mathrm{d}y \\ &\approx \sum_{k=1}^{K} w_Y^{(k)} \delta(y^{(k)} - \phi(x)). \end{aligned} \tag{15}$$

We see that $\overleftarrow{\mu}_X(x)$ only has mass for $x \in \mathcal{X}(y^{(k)})$. Denoting the cardinality of the set $\mathcal{X}(y^{(k)})$ by $\left|\mathcal{X}(y^{(k)})\right|$, we can construct the following particle representations of $\overleftarrow{\mu}_X(x)$: for each $y^{(k)}$ we have $\left|\mathcal{X}(y^{(k)})\right|$ particles (say $x^{(k,1)}, \ldots, x^{(k,|\mathcal{X}(y^{(k)})|)}$), all with weight $w_Y^{(k)}$, leading to a representation with $\sum_{k=1}^{K} \left|\mathcal{X}(y^{(k)})\right|$ particles.

For general functions $\phi(x)$, determining the sets $\mathcal{X}(y^{(k)})$ may be computationally hard. More problematically, in some cases $\mathcal{X}(y^{(k)})$ may be empty, possibly for all k, precluding the use of the above method (this happens when $\phi(x)$ is an injection).

There are two special cases: (i) when $\phi(x)$ is an invertible function, $\left|\mathcal{X}(y^{(k)})\right| = 1$, so the particle representation is of the form $\{w_Y^{(k)}, \phi^{-1}(y^{(k)})\}$; (ii) when $\phi(x) = Ax$, which is detailed in the next section.

F. Backward message for linear functions

1) Gaussian message passing: We have seen that, in general, computing backward messages is hard. However, for the case of Gaussian messages and linear functions ($y = Ax$), backward messages can be evaluated in closed form. We consider the case where A is an $N \times M$ matrix. From [1], [7], we recall that when $\overleftarrow{\mu}_Y(y) = \mathcal{N}(y; m_Y, \Sigma_Y)$ then

$$\begin{aligned}\overleftarrow{\mu}_X(x) &= \int \delta(y - Ax)\mathcal{N}(y; m_Y, \Sigma_Y)\mathrm{d}y \\ &= \mathcal{N}(Ax; m_Y, \Sigma_Y) \propto \mathcal{N}(x; m_X, \Sigma_X), \end{aligned} \tag{16}$$

in which [7, Table 3]

$$m_X = \left(A^{\mathrm{T}}\Sigma_Y^{-1}A\right)^{\#} A^{\mathrm{T}}\Sigma_Y^{-1}m_Y \tag{17}$$

$$\Sigma_X^{-1} = A^{\mathrm{T}}\Sigma_Y^{-1}A, \tag{18}$$

where $(\cdot)^{\#}$ denotes the Moore–Penrose pseudo-inverse. The expression (16) thus reveals that $\overleftarrow{\mu}_X(x)$ is obtained by evaluating $\overleftarrow{\mu}_Y(y)$ along the subspace $y = Ax$. For injective A, we can interpret this as *slicing* $\overleftarrow{\mu}_Y(y)$ along this subspace.

When $\overleftarrow{\mu}_Y(y)$ is a *mixture* of Gaussians, we can apply the above procedure to each mixture component, since integration is a linear operation. Thus, when

$$\overleftarrow{\mu}_Y(y) = \sum_{k=1}^{\tilde{K}} w_Y^{(k)} \mathcal{N}(y, m_Y^{(k)}, \Sigma_Y^{(k)}), \tag{19}$$

then

$$\begin{aligned}\overleftarrow{\mu}_X(x) &= \sum_{k=1}^{\tilde{K}} w_Y^{(k)} \mathcal{N}(Ax, m_Y^{(k)}, \Sigma_Y^{(k)}) \\ &= \sum_{k=1}^{\tilde{K}} w_Y^{(k)} s^{(k)} \mathcal{N}(x, m_X^{(k)}, \Sigma_X^{(k)}), \end{aligned} \tag{20}$$

where each Gaussian mixture component is scaled by[6]

$$\begin{aligned} s^{(k)} &= \exp\left(-\frac{1}{2}\left(\left(m_Y^{(k)}\right)^{\mathrm{T}}\left(\Sigma_Y^{(k)}\right)^{-1} m_Y^{(k)}\right)\right) \\ &\times \exp\left(\frac{1}{2}\left(\left(m_X^{(k)}\right)^{\mathrm{T}}\left(\Sigma_X^{(k)}\right)^{-1} m_X^{(k)}\right)\right), \end{aligned} \tag{21}$$

and $m_X^{(k)}$ and $\Sigma_X^{(k)}$ are computed from (17)–(18) by substituting m_Y by $m_Y^{(k)}$ and Σ_Y by $\Sigma_Y^{(k)}$. The weights can be normalized such that $\overleftarrow{\mu}_X(x)$ can be interpreted as a distribution, i.e.,

$$w_X^{(k)} = \frac{w_Y^{(k)} s^{(k)}}{\sum_{k'} w_Y^{(k')} s^{(k')}}. \tag{22}$$

2) Particle message passing: We now specialize the above expressions to the particular case when $\Sigma_Y^{(k)} = \sigma^2 I$, $m_Y^{(k)} = y^{(k)}$, and $\tilde{K} = K$, so that

$$m_X^{(k)} = \left(A^{\mathrm{T}}A\right)^{\#} A^{\mathrm{T}}y^{(k)} \tag{23}$$

$$\Sigma_X^{-1} = \frac{1}{\sigma^2}A^{\mathrm{T}}A = Q\mathrm{diag}\left[\frac{\lambda_1}{\sigma^2}\ldots\frac{\lambda_L}{\sigma^2}\ 0\ldots 0\right]Q^{\mathrm{T}}, \tag{24}$$

in which Q is an orthogonal matrix, $\lambda_1, \ldots, \lambda_L$ are the $L \leq M$ strictly positive eigenvalues of $A^{\mathrm{T}}A$, and $\mathrm{diag}\,[\cdot]$ generates a diagonal matrix from its vector argument. In other words, the last $M - L$ columns of Q span the null-space of $A^{\mathrm{T}}A$.

We can express the relation (19) as

$$\overleftarrow{\mu}_Y(y) = \sum_{k=1}^{K} w_Y^{(k)} \mathcal{N}(y, y^{(k)}, \sigma^2 I) \to \sum_{k=1}^{K} w_Y^{(k)} \delta(y - y^{(k)}),$$

as $\sigma \to 0$, which is exactly the form of a particle representation. The outgoing message can be expressed as

$$\overleftarrow{\mu}_X(x) = \sum_{k=1}^{K} w_Y^{(k)} s^{(k)} \mathcal{N}(x, \left(A^{\mathrm{T}}A\right)^{\#} A^{\mathrm{T}}y^{(k)}, \Sigma_X^{(k)}), \tag{25}$$

in which we tacitly assumed $\sum_{k=1}^{K} w_Y^{(k)} s^{(k)} = 1$, wherein (due to $\Sigma_Y^{(k)} = \sigma^2 I$)

$$s^{(k)} = \exp\left(-\frac{1}{2\sigma^2}(\|m_Y^{(k)}\|^2 - \|Am_X^{(k)}\|^2)\right). \tag{26}$$

[6] When expressing $\mathcal{N}(Ax; m_Y, \Sigma_Y)$ as $\mathcal{N}(x; m_X, \Sigma_X)$, there is a scaling factor. This scaling can be absorbed in the normalization when there is only one mixture component (as was implicitly done in (16)), but must be accounted for explicitly when there are multiple Gaussian mixture components with different scaling.

Letting $\sigma \to 0$ does not necessarily recover a meaningful particle representation of $\overleftarrow{\mu}_X(x)$. To understand this, we consider three cases, depending on whether A is bijective, injective, or surjective.

A is bijective: Since $m_Y^{(k)} = Am_X^{(k)}$, it follows that $s^{(k)} = 1, \forall k$. Hence, letting $\sigma \to 0$ in (25) yields

$$\overleftarrow{\mu}_X(x) \to \sum_{k=1}^{K} w_Y^{(k)} \delta(x - \left(A^{\mathrm{T}}A\right)^{\#} A^{\mathrm{T}} y^{(k)}), \qquad (27)$$

so $x^{(k)} = \left(A^{\mathrm{T}}A\right)^{\#} A^{\mathrm{T}} y^{(k)}$. Note that $\left(A^{\mathrm{T}}A\right)^{\#} A^{\mathrm{T}} = A^{-1}$. This is merely a special case of an invertible function, discussed in Section III-E. We thus find that each particle $y^{(k)}$ is transformed to $x^{(k)} = A^{-1}y^{(k)}$.

A is surjective: This happens, e.g., for fat matrices. In this case, the null-space of $A^{\mathrm{T}}A$ is not empty, so that $L < M$ (see (24)). Furthermore, $s^{(k)} = 1, \forall k$. The message $\overleftarrow{\mu}_X(x)$ can still be expressed in the form (27), though now each particle $x^{(k)} = \left(A^{\mathrm{T}}A\right)^{\#} A^{\mathrm{T}} y^{(k)}$ represents an $(M-L)$-dimensional space of the form

$$x = x^{(k)} + \sum_{l=L+1}^{M} \alpha_l q_l, \qquad (28)$$

in which α_l are arbitrary scalars and $\{q_l\}_{l=L+1}^{M}$ is a basis for the null-space of $A^{\mathrm{T}}A$. Hence, the concept of a particle must be extended to account for any point given by (28). We can denote the particle representation of $\overleftarrow{\mu}_X(x)$ by $\{w_X^{(k)}, x^{(k)}, \{q_{L+1}, \ldots, q_M\}\}$, in which $w_X^{(k)} = w_Y^{(k)}$.

A is injective: This happens, e.g., for tall matrices. In this case $s^{(k)}$ is of the general form (26). We introduce $d^{(k)} = ||m_Y^{(k)}||^2 - ||Am_X^{(k)}||^2 \geq 0$. We further introduce $k^* = \arg\min_k d^{(k)}$. Letting $\sigma \to 0$ will lead to

$$\overleftarrow{\mu}_X(x) \to \delta(x - \left(A^{\mathrm{T}}A\right)^{\#} A^{\mathrm{T}} y^{(k^*)}), \qquad (29)$$

so that only the particle $y^{(k)}$ closest to the space $y = Ax$ is relevant, and is *projected* onto the space $x = \left(A^{\mathrm{T}}A\right)^{\#} A^{\mathrm{T}} y$. This is clearly not a meaningful particle representation, moreover it is not consistent with the Gaussian case from Section III-F1 (i.e., when the particles $y^{(k)}$ are drawn from a Gaussian, we would expect the particle representation of $\overleftarrow{\mu}_X(x)$ correspond to particles from a Gaussian). The situation can be improved by increasing σ. However, as we have seen in Section III-A, an optimal choice of σ is not obvious. An alternative approach is to approximate the particles with a Gaussian distribution and apply the message passing rules (i.e., *slicing*) from Section III-F1.

IV. Numerical Examples

A. Illustrating example

In this section we will demonstrate the ideas for backward message passing for an injective transformation of the form $y = Ax$ with $A = [1\ 2]^{\mathrm{T}}$. As before, we assume $\overleftarrow{\mu}_Y(y) > 0$ for all y and is represented by a large number of particles $\{w_Y^{(k)}, y^{(k)}\}$. Now, $A^{\mathrm{T}}A = 5$ and $\left(A^{\mathrm{T}}A\right)^{\#} A^{\mathrm{T}} = [1\ 2]/5$. An example of the $\overleftarrow{\mu}_Y(y)$ and $\overleftarrow{\mu}_X(x)$ is shown in Fig. 2, according to two approaches, both of which are based on first approximating the particles $\{y^{(k)}\}$ with a Gaussian $\mathcal{N}(y, m_Y, \Sigma_Y)$. The red $\overleftarrow{\mu}_X(x)$ is obtained by applying the slicing operation (17)–(18), so that $m_X = \left(A^{\mathrm{T}}A\right)^{\#} A^{\mathrm{T}} m_Y$ and $\Sigma_X^{-1} = A^{\mathrm{T}}\Sigma_Y^{-1}A$, which for this specific case yields the correct result when the number of particles is large. The blue $\overleftarrow{\mu}_X(x)$ is obtained by projecting the individual particles to the space $y = Ax$. Clearly, this second approach leads to an incorrect distribution, in terms of both the mean and variance. Note, however, that when $\overleftarrow{\mu}_Y(y)$ is a Gaussian with Σ_Y a scaled identity matrix, slicing and projection give the same result.

B. Coherent optical receiver

An application of practical importance is the design of a digital communication receiver, in particular for situations where the optimal receiver may not be known in closed form, e.g., in fiber-optical communication systems [2]. We consider the setup based on [8, Fig. 1], where a single polarization 16-QAM sequence x is modulated using a root raised cosine pulse with roll-off of 25%, leading to a signal y, and fed into a fiber-optical channel with 60 spans, each span comprising 80 km standard single mode fiber, an amplifier, and dispersion compensating fiber. The channel introduces dispersion, nonlinearity due to the Kerr effect, and noise due to the amplifiers. The fiber and system parameters can be found in [8, Table 1]. The received signal is denoted by r. The goal of the receiver is to recover x from r, by computing $p(x|r)$ using a factor graph.

The received signal is processed with stochastic digital backpropagation (SDBP) [2], which performs message passing on a factor graph model of the channel, starting with the received signal r. In particular, each fiber is modeled using the split step Fourier method, where the fiber is split into many small segments and each segment consists of linear and nonlinear blocks as in Fig. 1 (d) and Fig. 1 (e). The amplifier can be modeled similar to Fig. 1 (c). The variables in the factor graph are the waveforms before and after each segment, before and after each amplifier, as well as x and y. The messages are represented using particles. After the particles are passed backward through each of the blocks of the system, we obtain K particles $\{y^{(k)}\}$ representing $\overleftarrow{\mu}_Y(y)$, describing the uncertainty regarding y. The relation between x and y can be modeled as $y = Ax$ as in Fig. 1 (d), where A represents the pulse shape including up-sampling. To obtain $\overleftarrow{\mu}_X(x)$, we consider two approaches, which start with a Gaussian approximation[7] of $\{y^{(k)}\}$ with mean m_Y and covariance Σ_Y. We then apply the slicing approach (leading to $m_X = \left(A^{\mathrm{T}}\Sigma_Y^{-1}A\right)^{\#} A^{\mathrm{T}}\Sigma_Y^{-1} m_Y$ and $\Sigma_X^{-1} = A^{\mathrm{T}}\Sigma_Y^{-1}A$) as well as a projection-style approach in which we project the mean $m_X = \left(A^{\mathrm{T}}A\right)^{\#} A^{\mathrm{T}} m_Y$, but keep $\Sigma_X^{-1} = A^{\mathrm{T}}\Sigma_Y^{-1}A$, both followed by symbol-by-symbol decisions. Note that in our application, the number of particles K is much smaller

[7] We separate real and imaginary parts of $y^{(k)}$ to capture correlations between them.

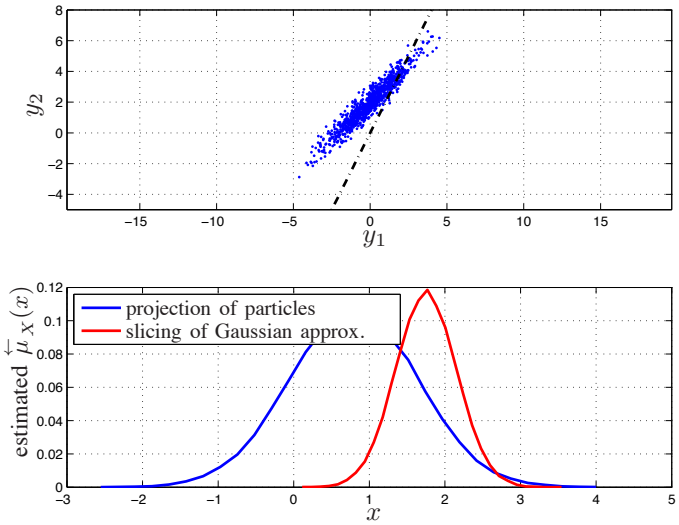

Figure 2. Example: $y = [x\ 2x]^{\mathrm{T}}$. We set $\overleftarrow{\mu}_Y(y)$ as $\mathcal{N}(y; [0\ 2]^{\mathrm{T}}, [2\ 1.9; 1.9\ 2])$ and generate $K = 10^6$ samples, a scatter plot of which is shown in the top figure, along with the dash-dotted line $y = [x\ 2x]^{\mathrm{T}}$. The bottom figure shows the (scaled) message $\overleftarrow{\mu}_X(x)$, obtained by approximating $\overleftarrow{\mu}_Y(y)$ by a Gaussian and applying the slicing operation (red) or projecting the particles onto the space $y = [x\ 2x]^{\mathrm{T}}$ (blue).

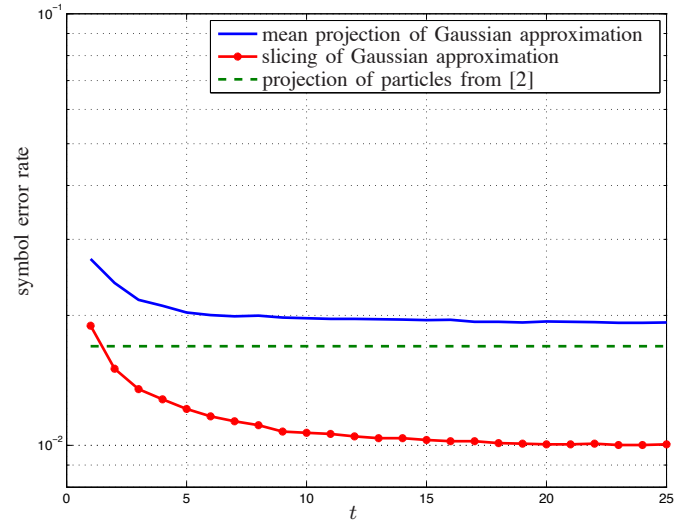

Figure 3. Symbol error rate as a function of the taper size t for a fiber optical communication receiver using SDBP [2]. We considered 16-QAM transmission over 60 spans of 80 km fiber at an input power of -3 dBm.

than the dimension of y. To avoid singular estimates of Σ_Y, we taper Σ_Y around its main diagonal, i.e., we replace $[\Sigma_Y]_{ij}$ with $[\Sigma_Y]_{ij} \times \exp(-|i-j|/t)$, for a tapering factor $t > 0$ [9]. We compare with the reference approach from [2], where each $y^{(k)}$ is projected onto $y = Ax$, leading to particles $x^{(k)} = \left(A^{\mathrm{T}}A\right)^{\#} A^{\mathrm{T}} y^{(k)}$, corresponding to matched filtering followed by symbol rate sampling of each $y^{(k)}$. For each symbol, the corresponding particles are approximated with a bivariate Gaussian distribution based on which a symbol-by-symbol decision is made.

In Fig. 3, we compare all these approaches in terms of the symbol error rate (SER). We observe that projection of the particles, as proposed in [2], leads to approximately the same performance as projection of the Gaussian approximation of the particles, for large enough t. Moreover, slicing the Gaussian approximation of the particles, as dictated by the sum–product algorithm, leads to superior performance. This indicates that matched filtering followed by symbol rate sampling is in general suboptimal and that the sub-optimality is due to an incorrect computation of the mean, not the covariance.

V. Conclusions

We have considered the problem of backward message passing in factor graphs using particles for certain linear and non-linear factors. We show that the problem is hard, even for factors that describe linear transformation. To address this, we have proposed a Gaussian approximation of the particles, and utilize a slicing and projection operation for message computation. The slicing operation follows from the sum–product algorithm, while the projection operation is an approximation. We have applied these to the problem of receiver design for coherent optical communication over a channel that includes dispersion, nonlinearities, and noise. We found that the slicing operation outperforms the benchmark method, though further analysis is needed to quantify these gains in a variety of scenarios.

Acknowledgment

We would like to thank Hans-Andrea Loeliger and Vincent Tan for discussions that helped to shape this research. This research was supported by the Swedish Research Council (VR) under grants 2010–4236 and 2013–5642, and the European Research Council, under Grant No. 258418 (COOPNET).

References

[1] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proc. IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007.

[2] N. V. Irukulapati, H. Wymeersch, P. Johannisson, and E. Agrell, "Stochastic digital backpropagation," *IEEE Trans. Commun.*, vol. 62, no. 11, pp. 3956–3968, 2014.

[3] J. Dauwels, S. Korl, and H.-A. Loeliger, "Particle methods as message passing," in *IEEE International Symposium on Information Theory*, pp. 2052–2056, IEEE, 2006.

[4] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," in *IEEE Conference on Computer Vision & Pattern Recognition*, pp. 605–612, 2003.

[5] S. J. Sheather *et al.*, "Density estimation," *Statistical Science*, vol. 19, no. 4, pp. 588–597, 2004.

[6] C. Au and J. Tam, "Transforming variables using the Dirac generalized function," *The American Statistician*, vol. 53, no. 3, pp. 270–272, 1999.

[7] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, 2004.

[8] N. V. Irukulapati, D. Marsella, P. Johannison, M. Secondini, H. Wymeersch, E. Agrell, and E. Forestieri, "On maximum likelihood sequence detection for single-channel coherent optical communications," in *European Conference on Optical Communication (ECOC)*, 2014.

[9] P. J. Bickel and E. Levina, "Regularized estimation of large covariance matrices," *The Annals of Statistics*, pp. 199–227, 2008.