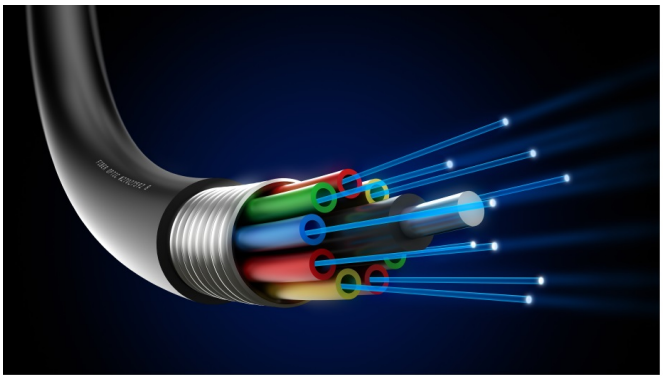
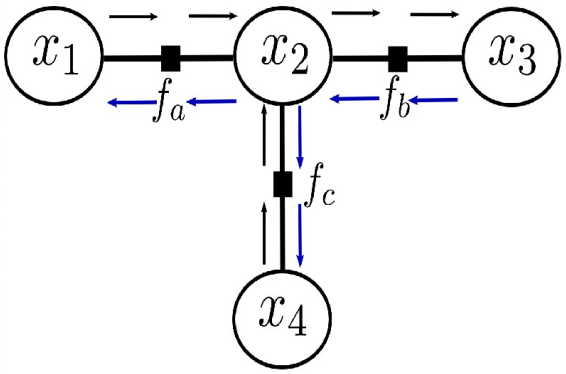
CHALMERS





Backward Gaussian particle message passing with applications to fiber optic communications

ISAAC A. SACKEY

Department of Signals and Systems
Chalmers University of Technology
Gothenburg, Sweden 2015
Master's Thesis 2015

Backward Gaussian particle message passing with applications to fiber optic communications

Isaac A. Sackey

Examiner: Henk Wymeersch, Associate Professor, Signals and Systems, Chalmers University of Technology

Supervisor: Naga V. Irukulapati, PhD candidate, Signals and Systems, Chalmers University of Technology

CHALMERS

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2015

Abstract

Optical communication systems provide data rates that far exceeds alternate systems for on-premises data links and equipment wiring and also for long-haul communications. Fiber is mostly preferred not only for its extremely large data-carrying capacity, but also for its small attenuation over large bandwidths, small physical size and immunity to interference from extraneous ambient electrical signals.

However, inherent intensity-dependent nonlinearities in fiber tend to reduce system performance for high input power ranges. The state-of-the-art design to compensate for these nonlinearities is the digital back propagation (DBP) (digital-signal processing (DSP) - based approach), which does not account for noise that is present in real life systems with periodic amplification. The stochastic digital back propagation (SDBP) technique extends the DBP technique to account for noise in addition to compensating for deterministic linear and nonlinear impairments.

In this work we show that, the decision unit of the SDBP receiver is not optimal since it neglects any residual memory that may be present. We extend the SDBP algorithm by applying a maximum a posterior (MAP)-based criterion using factor graphs and the sum product algorithm (SPA) to develop a novel detector that accounts for non-optimality in the stochastic digital back propagation (SDBP). We shown that the proposed method performs significantly better than the original SDBP. For inline dispersion-managed links, the proposed algorithm has up to 2 times lower symbol error rate (SER) than the original SDBP and up to 9 times lower SER than DBP in the nonlinear power regime for 16-QAM modulation.

Acknowledgements

This thesis is the concluding part of my studies at the Master's Programme in Communication Engineering at Chalmers University of Technology. The thesis work has been conducted at the department of Signals and Systems at Chalmers University of Technology with the aid of a student scholarship grant from the Swedish Institute.

I would like to first and foremost give my sincere and unreserved gratitude to God for his guidance, provision and blessing me with the enormous amount of support I have received throughout my degree and especially during this thesis work.

I would like to thank Prof. Henk Wymeersch for his motivation, immense knowledge and patience. And for always being so eager to help.

I would also like to thank my supervisor, Naga V. Irukulapati at Chalmers University of Technology for his tireless efforts, insightful remarks and relentless support during this thesis work.

I would also want to show appreciation to the SI for their kindness, hospitality and generosity during my studies.

Last but not the least, my sincere thanks goes to all my family, friends and loved ones for their constant support and encouragement during my studies.

Isaac A. Sackey,

Gothenburg, October 11, 2015

Contents

Acronyms

ASE Amplified Spontaneous Emission.

AWGN Additive White Gaussian Noise.

CD Channel Dispersion.

DBP Digital Back Propagation.

DCF Dispersion Compensation Fiber.

DM Dispersion Managed.

DOK Dictionary of Keys.

DSP Digital Signal Processing.

EDFA Erbium Doped Fiber Amplifier.

FFG Forney-Factor Graph.

FG Factor Graph.

FWM Four Wave Mixing.

GMP Gaussian Message Passing.

GVD Group Velocity Dispersion.

ISI Inter-Symbol Interference.

LDPC Low-Density Parity-Check.

MAP Maximum A Posteriori.

MF Matched Filtering.

NLPN Nonlinear Phase Noise.

NLSE Nonlinear Schrodinger Equation.

NSNI Nonlinear Signal to Noise Interaction.

PDF Probability Density Function.

PMD Polarization Mode Dispersion.

PR Particle Representation.

PSK Phase-Shift Keying.

QAM Channel Dispersion.

SBS Symbol-by-Symbol.

SDBP Stochastic Digital Back Propagation.

SER Symbol Error Rate.

SMF Single-mode Fiber.

SNR Signal to Noise Ratio.

SPA Sum Product Algorithm.

SPM Self Phase Modulation.

SSFM Split Step Fourier Method.

WDM Wavelength Division Multiplexing.

XPM Cross Phase Modulation.

1

Introduction

1.1 Background and motivation

Over the past decades, demand for fast reliable communications has risen to an all time high. High speed, data consuming, and power hungry electronic applications have driven the need for high-efficient transmission and detection algorithms to extraordinary limits. Forecasts however show that this ever increasing global data traffic is bound for a thousand-fold rise by the year 2020 [1] with 24 billion globally networked devices and connections by 2019, representing a 71% raise from 2014 [2].

Meanwhile, the advent of semiconductor and electronic architecture technology has bolstered the efficiency of digital signal processing (DSP)-based informational signal processing leading to increased data rates [3], whilst the data-carying capacity of some transmission media have tremendously improved over the years. In that vein, fiber-optic communication based systems have attracted immense attention for high-speed inter-continental backbone communications. This is because fiber has a very high bandwidth (several Terabits/sec [4]), has low attenuation factor (about 0.2 dB/km [4], over several THz), is small in size [10] (with a cross sectional area of about $30\mu m^2$ - $100\mu m^2$ [4]), and is immune to other electromagnetic interference (which implies that electrical cables can be run along with it without any electrical disturbance).

Due to the increasing aggregate of global data traffic, exploration of efficient ways to achieve higher data rates in fiber has driven its transmission capacity to increase by a factor of approximately 10 every four years as shown in Fig. 1.1 [9]. Key technological breakthroughs in fiber-optic communication systems include the development of low-loss single-mode fiber (SMF), the erbium doped fiber amplifier (EDFA), wavelength division multiplexing (WDM) and high-spectral-efficiency coding through DSP-enabled coherent transmission [9]. Exploration of innovative ways however present challenges, which were nearly impossible to tackle in the early years of fiber-optic communications due to extremely low level DSP technology available at the time [4].

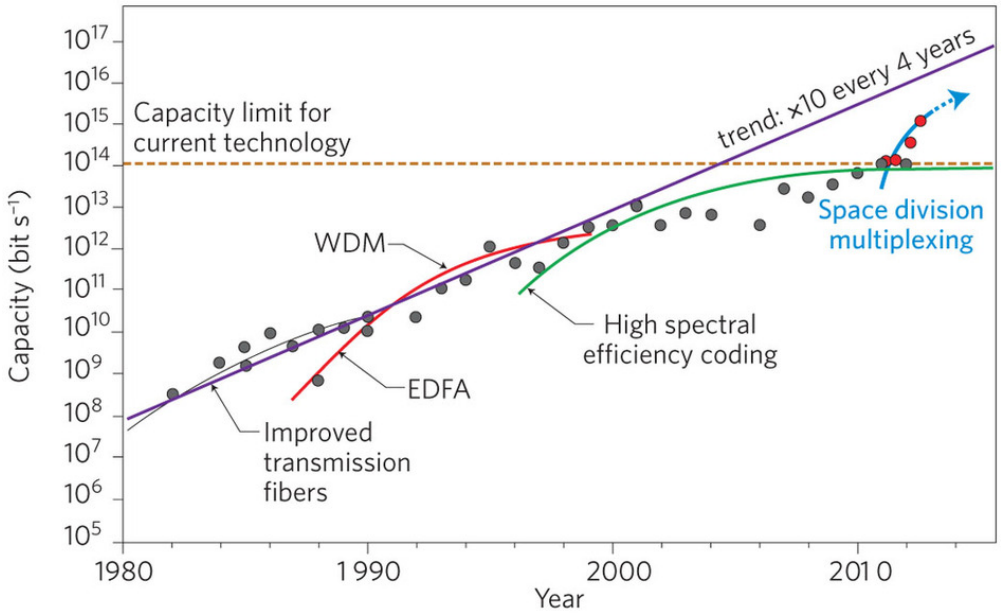


Figure 1.1: Improvement of fiber systems over the years [9].

In general, response of any dielectric material to light becomes nonlinear for high intensity electromagnetic fields and fiber-optic materials are no exception [4]. This can be explained fundamentally by the anharmonic motion of bound electrons of the fiber atoms with increasing input power intensity [24], resulting in a nonlinear susceptibility and eventually leading to the so called Kerr effect. The induced refractive index variations translate into changes in the signal's optical phase [10]. Thus, the signal is distorted (in the frequency domain) and the effective symbol error rate (SER) of the system increases. Fiber systems have an optimum transmit power beyond which the SER begins to increase due to the nonlinearity domination in the fiber. This is shown in the graph in Fig. 1.2

In order to address nonlinearities in the fiber, early lightwave systems applied input power back-off from the nonlinear dominated power regions, and used binary modulation formats at the transmitter with fixed threshold detection at the receiver. Although communication distances were in the hundreds of kilometers, the power levels were too low for any accumulated nonlinear impairments to have any significant impact on the transmission. Hence the only main problems that confronted communications were linear impairments like dispersion and power attenuation [4]. High spectral efficiency coding (e.g. through high order constellation schemes with coherent detection) require exploring different high input power options in order to maintain a specific SER performance for example. However, as seen in Fig. 1.2 high input powers lead to operation in the nonlinear dominated region.

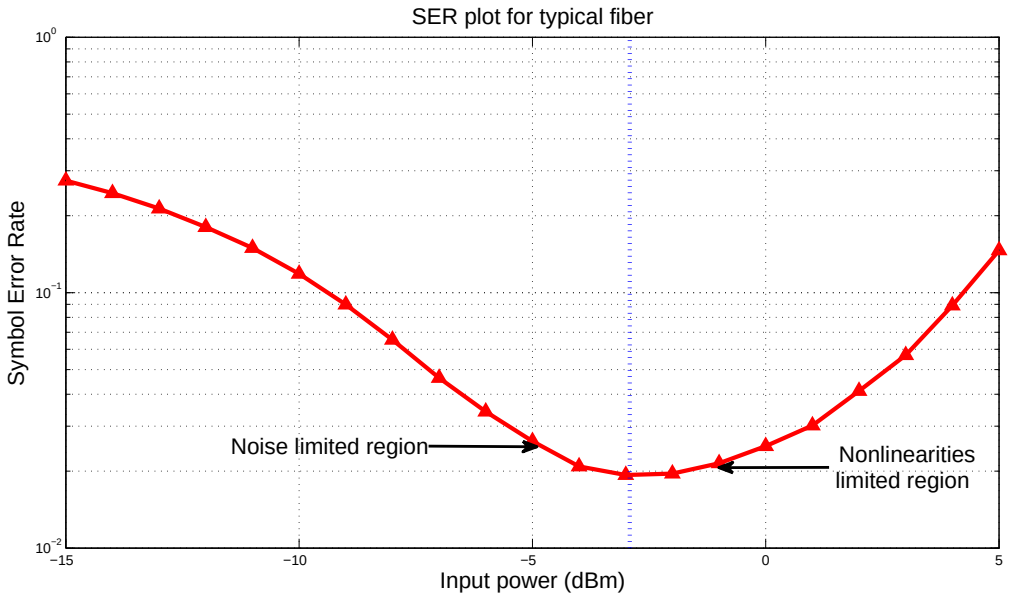


Figure 1.2: Effects of nonlinearities on symbol error rate in optical communication systems.

DSP receiver-based efforts have been made to find different ways of compensating for the impact of the inherent nonlinearities in the fiber, by processing the received signal before making best effort decisions to minimize errors. These DSP methods include channel phase conjugation techniques such as the nonlinear phase noise (NLPN) compensation technique [27], which applies phase conjugation to the received signal at the receiver to remove the nonlinear phase noise, channel inversion techniques such as digital back propagation (DBP) [11] and the SDBP [7] techniques, which back propagates the received signal through a virtual inverted channel to remove the effects of the inherent fiber impairments and other methods based on Voltera kernels [14]. Other contributions pertinent to this work are referenced in relevant sections of the thesis. Three of such DSP based optimization techniques are considered in Chapter 3. We will however be particularly concerned with one of these techniques, namely the SDBP, which uses factor graphs, particle methods and message passing in the design of communication detection algorithms. In the next section, we briefly describe the principle behind back propagation and highlight the main problem, which we wish to tackle in this work.

1.2 Existing approaches

Light propagating through an optical fiber can be modeled with the nonlinear schrödinger equation (NLSE). Closed-form solutions to the NLSE exist if either the linear or nonlinear impairments is negligible. However when both linear and nonlinear impairments are present, the resulting nonlinear partial differential equation does not generally lend itself

to a closed-form solution . In that case, the split step Fourier method (SSFM) can be used as a numerical method to simulate the NLSE [4]. The SSFM approximates NLSE by assuming linear and nonlinear effects act independently along a very small segment of the fiber. Given the transmitted signal that propagates through a fiber cable, the received signal can be approximated with simulations just as in practice with the SSFM. DBP is a receiver-based signal processing technique to jointly compensate for linear and nonlinear impairments in a fiber-optic cable by applying the SSFM solution to the received optical signal with opposite optical fiber propagation parameters [11]. A more detailed mathematical description of this algorithm is given in Chapter 3.

In long-haul lightwave systems, optical amplifiers are used to reinforce weak signals. Optical amplifiers often introduce noise through amplified spontaneous emission (ASE), which distorts the optical signal. For such systems the DBP technique cannot perform optimally since it does not account for noise. Irukulapati et. al. [7], proposed a technique to improve on DBP. Having the maximum a posteriori (MAP) criterion as the starting point and with factor graphs and a message passing algorithm, namely the sum product algorithm (SPA), they arrive at a solution whose primary objective is to minimize the probability of error. It assumes that the ASE noise from the optical amplifiers is white, Gaussian and additive. In the detection unit of [7], in order to maximize the signal to noise ratio, a filter matched to the pulse shape and a symbol rate sampler at optimal sampling times is applied (assuming perfect synchronization) [30]. The matched filtering technique here is only applied heuristically. This thesis work investigates the optimality of the matched filtering (MF) in the SDBP, and attempts to develop a novel detector that avoids heuristics and hence improves on the SER within the nonlinear dominant power regime. The tools we employ are highlighted in the next section.

1.3 Our methodology: message passing on factor graphs

The graphical model approach (such as factor graph (FG)) to problem solving is increasingly becoming a universal way of representing complex signal processing, channel coding and machine learning problems in a comprehensive and systematic manner [6]. Just as Tanner graphs are associated with low-density parity-check (LDPC) and Trellis graphs are associated with the Viterbi algorithm, it is also common to associate factor graphs with a number of message passing algorithms. The SPA or belief propagation and the maximum product or minsum algorithms are such algorithms of which the SPA is arguably the most common [20].

Factor graphs works well with the SPA message passing algorithms in Bayesian inference problems [6]. The SPA solves inference problems by computing the posterior marginal of the variable of interest, provided there are no cycles in the graph and all variables are discrete. Particle representation is a way to represent messages as a list of particles with weights [23]. It has been shown in [23] that particle representations are a full member of the family of message-passing methods that enables for a number of possibilities including: applying particle methods at nodes in factor graphs to determine forward and backward messages and applying particle methods to factor graphs with

cycles.
In the following Chapters, we exploit these tools to tackle the SDBP detection problem, which was highlighted in the previous section.

1.4 Objective and outline

SDBP is one of the few techniques that accounts for ASE noise in its operation. At the detection unit, a matched filter and symbol rate sampling is used to maximize signal to noise ratio (SNR). This method was only applied based on heuristics and need not be optimal. This thesis work aims to find out the following:

1. if the matched filtering and symbol rate sampling employed by the SDBP is optimal and if not
2. develop an algorithm that will minimize SER better than matched filtering and sampling and
3. investigate whether the proposed algorithm is optimal.

The structure of this work is as follows:
Chapter 2 provides a mathematical background in fiber-optical communication and the SPA. It entails a mathematical representation of the system and channel models, the NLSE, and an introduction to linear factor graphs models as well as the SPA.

In Chapter 3, we review and compare existing receiver-based algorithms that jointly compensate for linear and nonlinear impairments. Following that, is a formal formulation of the problem statement.

In Chapter 4, we derive the proposed technique and verify its performance improvement over SDBP. Some implementation problems are also discussed and solved.

Finally, we draw our conclusions in Chapter 5 and provide suggestions for future work.

2

Background review

2.1 System model

Notation: $x(t)$ represents a single-polarized continuous-time optical signal. Its vector representation, which is an over-sampled version (sequence of data symbols, spaced at symbol time T_s apart), and in general all vectors are represented with lower case bold letter $\mathbf{x}$. $\tilde{\mathbf{x}}$ is the frequency domain representation of $\mathbf{x}$. Δz represents an infinitesimal width along a distance variable, z, often too small for any depending variable to vary along that distance. The Hermitian conjugate of a complex matrix A is denoted as A^H and transpose of a real matrix, M is M^T. The Moore-Penrose pseudo inverse is denoted as $(\bullet)^{\#}$. The Dirac delta function is also written as $\delta(\bullet)$. A multivariate Gaussian probability density function (PDF) of a random variable, $\mathbf{X}$ with mean $\mathbf{m}_X$ and covariance Σ_X is denoted as $\mathcal{N}(\mathbf{x};\mathbf{m}_X,\Sigma_X)$.

2.1.1 Transmitter

The transmitter system connects to the receiver system via the N_{span} spans of optical fiber and EDFAs. A binary sequence $b[n] = \{b[0], b[1],...,b[L_b - 1]\}$ of length L_b, is modulated using a Quadrature Amplitude Modulation (QAM) or phase-shift keying (PSK) modulator where the available symbol set is given as Ω, $\log_2(\Omega) = L_b/M$. M is the length of the resulting complex symbol sequence, $\mathbf{s} = \{s[0], s[1],...,s[M - 1]\} \in \Omega^M$ and $\mathbf{s}[n] \in \mathbb{C}$. $\mathbf{s}$ is then upsampled with upsampling factor q and filtered with a continuous time pulse, $p(t)$, usually a root-raised-cosine pulse. The continuous-time complex baseband signal propagated through the fiber-optical communication system from the output of the pulse shaper is given as:

$$v(t,l_{1,1}) = \sum_{n=0}^{M-1} s[n]p(t - nT_s) \tag{2.1}$$

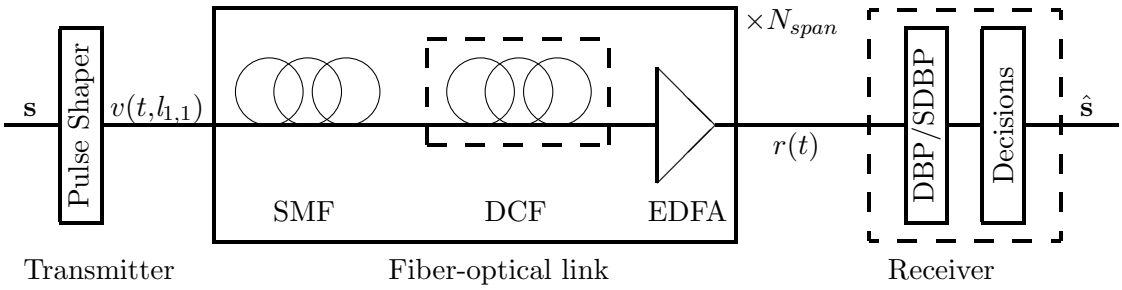


Figure 2.1: System model consisting of a fiber link with N_{span} span concatenations of an SMF, a dispersion compensation fiber (DCF) (for dispersion managed (DM) links), and an EDFA [8].

where $p(t)$ is the impulse response of filter. $v(t,l_{1,1})$ is the signal at the output of the pulse shaping and is different from the signal in other segments of the fiber spans, which are denoted as $v(t,l_{n\neq1,m\neq1})$.

2.1.2 Receiver

The receiver consists of a matched filter that filters the received signal, $r(t)$ with $p(t)$ to avoid inter-symbol interference (ISI) and downsamples the resulting signal at symbol rate. Its overall aim is to optimally determine $\mathbf{s}$ given $r(t)$ based on an optimality criterion. In this work we consider the MAP criterion. A MAP detection criterion will make the decision according to:

$$\hat{\mathbf{s}} = \underset{\mathbf{s}\in\Omega^M}{\operatorname{argmax}}\, p(\mathbf{s}|\mathbf{r}) \tag{2.2}$$

$$= \underset{\mathbf{s}\in\Omega^M}{\operatorname{argmax}}\, \frac{p(\mathbf{r}|\mathbf{s})p(\mathbf{s})}{p(\mathbf{r})} \tag{2.3}$$

where $p(\mathbf{s}|\mathbf{r})$ is the posterior distribution, $p(\mathbf{r}|\mathbf{s})$ is the likelihood function and the $p(\mathbf{s})$ is the a priori distribution. To reduce complexity a symbol-by-symbol detection could be used:

$$\hat{s}_i = \underset{s_i\in\Omega}{\operatorname{argmax}}\, p(s_i|\mathbf{r}) \tag{2.4}$$

The QAM or PSK detected symbols are demodulated according to the symbol mapping in the transmitter to yield $\hat{\mathbf{b}}$.

2.1.3 Channel model

A. Signal Propagation in fiber

Signal propagation in an optical channel can be modeled with the NLSE as [7]:

$$\frac{\partial \mathbf{v}}{\partial z} = (\hat{N} + \hat{D})\mathbf{v} \qquad where: \quad \hat{N} = j\gamma|\mathbf{v}|^2, \quad \hat{D} = -j\frac{\beta_2}{2}\frac{\partial^2}{\partial t^2} - \frac{\alpha}{2} \tag{2.5}$$

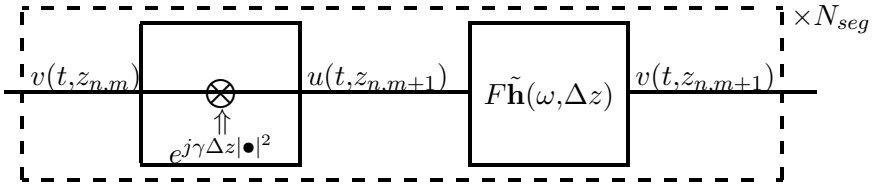


Figure 2.2: SSFM model for one fiber segment with nonlinear and linear impairments, where $h(\omega,\Delta z) = \exp(j\beta_2\omega^2\Delta z/2)$ and $F = \exp(\alpha\Delta z/2)$.

where $\mathbf{v}$ is the electric field, $|\mathbf{v}|^2$ is the signal power, γ is the nonlinear parameter, β_2 is the group velocity dispersion (GVD) and α is the attenuation factor. The linear differential operator, $\hat{D}$, operates on linear impairments parameters whilst $\hat{N}$, represents the nonlinear operation. The exact analytic solution when all parameters are non-zero is not available in closed form except for some specific instances like soliton transmission. It is possible however, to determine the solution when the nonlinear factor $\gamma = 0$ or when GVD, β_2 is set to zero. When all parameters are non-zero, it is normal to resort to approximate numerical solutions to describe the optical signal evolution. One of the most common methods is the SSFM. The SSMF is based on two assumptions: (i) linearities and nonlinearities act independently in a small enough segment of the fiber. (ii) the phase change in the time domain due to the Kerr effect, and the phase change due to the channel dispersion (CD) in the frequency domain is constant over a very small distance Δz. In this case, the size of the chosen segment size, Δz is crucial and determines the accuracy of the approximation. With the SSFM, the transmitted signal can be retrieved from the received signal by the inverse NLSE solution: this is given by solving for $\mathbf{v}(z,t)$ in the equation:

$$\mathbf{v}(t,z_{n,m+1}) \approx \exp\left(\Delta z\hat{D}\right)\exp\left(\Delta z\hat{N}\right)\mathbf{v}(t,z_{n,m}) \tag{2.6}$$

where $\Delta z = z_{n,m+1} - z_{n,m}$ is the step size, $\hat{D}$ and $\hat{N}$ are defined as in equation 2.5. An iterative process is then used to solve the equation. (2.6) is referred to as the asymmetric SSFM as the nonlinearity is added at the boundary of the segment. In the symmetric SSFM, the nonlinear operation is in the middle of the segment whilst symmetric linear exponential operators are at the boundaries. The exponential operators, $\exp\left(\Delta z\hat{D}\right)$ and $\exp\left(\Delta z\hat{N}\right)$, can be interpreted as a phase shift, in the frequency domain (in the case of the linear operator) or in the time domain (in the case of the nonlinear operator). In the limit of Δz approaching zero, both the symmetric and asymmetric SSFM approach the true NLSE solution [11]. In long haul optical communication systems, N_{span} fiber spans, each of length L, consisting of an SMF and DCF is concatenated into small segments of size Δz (applies to both DCF and SMF fibers). For each segment, linear effects and nonlinear effects are treated independently assuming Δz is chosen to be small enough. We can therefore refer to the distance of the input of a segment, $m \in \{1,2,\cdots,N_{seg}\}$ in fiber span $n \in \{1,2,\cdots,N_{span}\}$ as: $z_{n,m}$. The input-output signal relationship of

segment m in span n according to the SSFM algorithm is given as:

$$u(t,z_{n,m+1}) = v(t,z_{n,m}) \exp\left(j\gamma\Delta z|v(t,z_{n,m})|^2\right) \tag{2.7}$$

and

$$\tilde{v}(\omega,z_{n,m+1}) = \tilde{u}(\omega,z_{n,m+1}) \times \exp\left(-j(\frac{\alpha}{2} + \frac{\beta_2\omega^2}{2})\Delta z\right) \tag{2.8}$$

where $z_{n,m} = (m-1)\Delta z+(n-1)L$, $\tilde{u}(\omega,z_{n,m})$ is the Fourier transform of the intermediary optical intensity, $\mathbf{u}(t,z+\Delta z)$, between the linear and nonlinear sections of each segment. Equation 2.8 can be represented in the time domain as:

$$v(t,z_{n,m+1}) = Fu(t,z_{n,m+1}) * \frac{e^{-jt^2/(2\beta_2\Delta z)}}{\sqrt{j2\pi\beta_2\Delta z}} \tag{2.9}$$

Where $*$ is the convolution operation and $F \triangleq e^{(-\alpha\Delta z/2)}$. β_2 is the GVD parameter, and γ is the nonlinear paramter. The operation of the nonlinear effects before linear effects per segment is based on the assumption that since signal intensity is highest at the beginning of each fiber segment, nonlinear effects are are also highest at those points. From (2.7) - (2.8) it can be observed that the time spread caused by the GVD is as a result of phase change in the frequency domain, (2.8), whilst the spectrum spread due to the nonlinearities is due to phase change in the time domain as expressed in equation (2.7).

B. SMF

The SMF allows for a single mode of light to propagate. The input-output relationship in the SMF is given as:

Algorithm 1 Signal propagation in SMF using SSFM

procedure SIGNAL PROPAGATION SMF
  for each segment $m \in \{1,...,N_{seg}\}$ **do**
    $\mathbf{u}_{SMF}(z_{n,m+1}) = \mathbf{v}_{SMF}(z_{n,m}) \times \exp\left(k_{SMF,NL}|\mathbf{v}_{SMF}(z_{n,m=1})|^2\right)$
    $\tilde{\mathbf{v}}_{SMF}(z_{n,m+1}) = \tilde{\mathbf{u}}_{SMF}(z_{n,m+1}) \times \exp\left(k_{SMF,L}\right)$
  end for
  $\mathbf{v}_{DCF}(z_{n,m=1}) = \mathbf{v}_{SMF}(z_{n,m=N_{seg}+1})$
end procedure

where $\tilde{\mathbf{v}}_{SMF}(z_{n,m})$ is the frequency domain representation of $\mathbf{v}_{SMF}(z_{n,m})$, $k_{SMF,L} = (-j(\frac{\alpha}{2} + \frac{\beta_2\omega^2}{2})\Delta z)$ and $k_{SMF,NL} = \exp\left(j\gamma\Delta z|\mathbf{v}(t,z_{n,m})|^2\right)$ are SMF linear and nonlinear constants respectively.

C. DCF

The DCF also allows for only a single mode of light to propagate. In our system model in Fig. 2.1, the output of the SMF (which is input to the DCF) is related to the output

Algorithm 2 Signal propagation in SMF using SSFM

procedure SIGNAL PROPAGATION SMF
  for each segment $m \in \{1,...,N_{seg}\}$ **do**
    $\mathbf{u}_{DCF}(z_{n,m+1}) = \mathbf{v}_{DCF}(z_{n,m}) \times \exp\left(k_{DCF,NL}|\mathbf{v}_{DCF}(z_{n,m=1})|^2\right)$
    $\tilde{\mathbf{v}}_{DCF}(z_{n,m+1}) = \tilde{\mathbf{u}}_{DCF}(z_{n,m+1}) \times \exp\left(k_{DCF,L}\right)$
  end for
  $\mathbf{v}_{SMF}(z_{n+1,m=1}) = \mathbf{v}_{DCF}(z_{n,m=N_{seg}+1})$
end procedure

of the DCF according to:

where $\tilde{\mathbf{v}}_{DCF}(z_{n,m})$ is the frequency domain representation of $\mathbf{v}_{DCF}(n,m)$, $k_{DCF,L} = (-j(\frac{\alpha}{2} + \frac{\beta_2\omega^2}{2})\Delta z)$ and $k_{DCF,NL} = \exp\left(j\gamma\Delta z|\mathbf{v}(t,z_{n,m})|^2\right)$ are respectively, DCF linear and nonlinear constants. The main difference between the DCF and the SMF is that their dispersion parameters are opposite in signs, and the magnitude of the DCF's dispersion parameter is much larger than that of the SMF, i.e., $|D_{DCF}| >> |D_{SMF}|$.

D. EDFA

EDFAs are by far the most common types of optical amplifiers, where accumulated degradation is compensated by signal regeneration with fibers doped with erbium ions $\approx$ 10m long [4]. Amplification in the EDFA occurs by the principle of stimulated emission. However ASE can occur when electrons in higher-energy levels of the erbium-ions decay at random and emit photons at random phases and directions. Some of which end up in the fiber core and is amplified by stimulated emission. ASE is the main limiting factor in light-wave communications. The power spectral density of the ASE noise is given as: $S_p = GF_nh\nu$, where G is the gain of the amplifier, $h\nu$ is the photon energy and F_n is the noise figure of the amplifier given to be [4]: $F_n = 2n_{sp}(1 - G^{-1})$. n_{sp} is the spontaneous emission factor that depends on the inversion population. In this work we compensate attenuation losses completely. Since optical power decreases according to: $P_{out} = P_{in}e^{-\alpha z}$ [4] in lumped amplification with amplifier spacing L_A, the amplifier gain is designed so that: $G = e^{(\alpha L_A)}$.

Therefore in the EDFA unit of our system model, if the incoming signal is given as: $\mathbf{z}(z_{n,N_{seg}})$, the output signal is given as: $\mathbf{r}(z_{n,N_{seg}}) = G\mathbf{z}(z_{n,N_{seg}}) + n$ where $n \sim \mathcal{N}(n;0,\Sigma)$.

In the next section, we introduce the tools that are needed in later sections to describe the development of the SDBP algorithm, from which we derive our problem statement.

2.2 Message passing on FG

Message passing processing refers to a collection of low complexity processes (blocks) working in a distributed fashion by sending messages to each another and relying on these

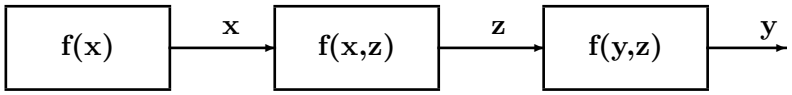


Figure 2.3: Example: Factor graph representation of factorization.

messages to perform a task [5]. Graphical representations such as factor graphs allows us to represent these passing of messages in a visually comprehensive and systematic manner [20]. Each block performs a process based on an algorithm whose output (message) may be the input of another process. In the next sections we introduce a special type of factor graphs called the Forney-factor graph (FFG) as a graphical representation model and show how the SPA algorithm is used to compute messages on it.

Notation: Messages are represented with overhead arrows, $\overrightarrow{\mu}$, $\overleftarrow{\mu}$ for rightward and leftward messages respectively. The direction of the arrow is the direction of the message. A general function, f of variables $\mathbf{x}$,$\mathbf{y}$,$\mathbf{z}$ is represented as $f(\mathbf{x},\mathbf{y},\mathbf{z})$. A linear function of $\mathbf{x}$ is denoted $f(\mathbf{x},\mathbf{y}) = \delta(\mathbf{y}-A\mathbf{x})$, whilst a nonlinear function is represented as $f(\mathbf{x},\mathbf{y}) = \delta(\mathbf{y}-\psi(\mathbf{x}))$. The multivariate Gaussian PDF of a variable $\mathbf{y}$ with covariance Σ_Y, conditioned on $\mathbf{x}$ and related to it by noise addition, $\mathbf{y} = \mathbf{x} + \mathbf{n}$ is denoted as $\mathcal{N}(\mathbf{y}|\mathbf{x},\Sigma)$ where n is a zero mean Gaussian noise process and $\mathcal{N}(\mathbf{y};\mathbf{m}_Y,\Sigma_Y)$ denotes a Gaussian distribution of $\mathbf{y}$ with mean and covariance $\mathbf{m}_Y$ and Σ_Y respectively.

2.2.1 Factor graphs

FG have a long standing history in error-coding theory back as early as in 1960s. Just as it is common to relate the trellis diagram to the Viterbi algorithm, it is quite common to also relate the SPA and max-product algorithm to factor graphs. In this work we will consider only the FFGs and execute the SPA algorithm on it. We first present a set of linear factor blocks and show with an example how the SPA can be applied in the case Gaussian messages.
For random variables, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, assume that a generic function $f(\mathbf{x},\mathbf{y},\mathbf{z})$ can be factored as:

$$f(\mathbf{x},\mathbf{y},\mathbf{z}) = f_1(\mathbf{x})f_2(\mathbf{x},\mathbf{z})f_3(\mathbf{y},\mathbf{z}) \tag{2.10}$$

This factorization is expressed by the Forney-style FG in Fig. 2.3. We note however that this factorization is not unique and can always be varied in ways so as to simplify the problem. Each factor represents a node and each variable represents an edge. Each edge connected to two factors is referred to as a full edge (i.e $\mathbf{x}$ and $\mathbf{z}$ in Fig. 2.3) whilst an edge connected to one factor is a half edge (i.e $\mathbf{y}$ in Fig. (2.3)). The obvious limitation is that, a variable cannot appear in factor more than twice. We show how to overcome

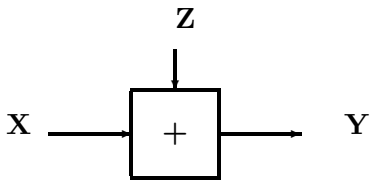


Figure 2.4: The 3-way addition block.

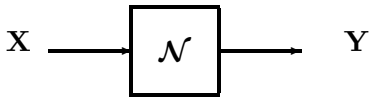


Figure 2.5: Noise addition block.

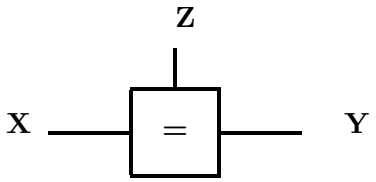


Figure 2.6: The equality block.

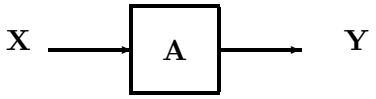


Figure 2.7: Matrix multiplication block.

this problem in the next section.
In linear models, the factor blocks commonly used are shown in Figs. 2.4 - 2.7 and are explained in details below.

Summation: the factor function is given as: $f(\mathbf{x},\mathbf{y},\mathbf{z}) = \delta(\mathbf{z} - (\mathbf{x} + \mathbf{y}))$ and the function block is shown in Fig. 2.4. We assume that $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ are of the same lengths.

Noise addition: the function block for the addition of noise is represented in Fig. 2.5. The factor function is: $f(\mathbf{x},\mathbf{y}) = \mathcal{N}(\mathbf{y}|\mathbf{x},\mathbf{\Sigma}_Y)$. Here again, it is assumed that $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are of the same lengths. Where $\mathcal{N}(\mathbf{y};\mathbf{m}_Y,\Sigma_Y)$ is a Gaussian distribution of $\mathbf{y}$ with mean $\mathbf{m}_Y$ and covariance Σ_Y.

3-way equality: the function block for the 3-way equality is shown in Fig. 2.6 and the factor graph function representation is given by: $f(\mathbf{x},\mathbf{y},\mathbf{z}) = \delta(\mathbf{x}-\mathbf{y})\delta(\mathbf{x}-\mathbf{z}) = \delta(\mathbf{x}-\mathbf{z})\delta(\mathbf{y}-\mathbf{z}) = \delta(\mathbf{y}-\mathbf{z})\delta(\mathbf{x}-\mathbf{y})$. It is assumed that $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ have the same lengths. When a variable appears in more than 2 functions, the 3-way equality is used to connect the all equivalent variables together.

Matrix Multiplication: the factor block representation is shown in Fig. 2.7. The matrix represents a linear operation as applied in the numerical examples presented in section 4.1.5. If the message is Gaussian, both forward and backward messages can be computed in closed form [6]. This however is not the case for nonlinear systems. The factor representation is given as $\delta(\mathbf{y} - f(\mathbf{x}))$. Where $\mathbf{y}$ and $\mathbf{x}$ need not be the same size. If $A_{N\times M}$, then $\mathbf{y}$ is of dimension N, and $\mathbf{x}$ is of dimension M.

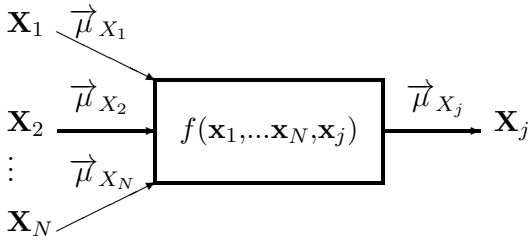


Figure 2.8: Message passing representation for N+1 variables. $\overrightarrow{\mu}_{X_j}$ is computed with the SPA using $\overrightarrow{\mu}_{X_1},...,\overrightarrow{\mu}_{X_N}$.

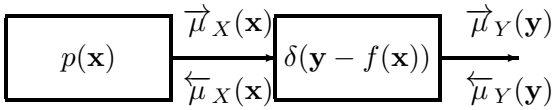


Figure 2.9: Incoming ($\overrightarrow{\mu}_X$, $\overleftarrow{\mu}_Y$), outgoing ($\overrightarrow{\mu}_Y$, $\overleftarrow{\mu}_X$), forward ($\overrightarrow{\mu}_Y$) and backward ($\overleftarrow{\mu}_X$) messages.

2.2.2 The SPA algorithm

The SPA algorithm, also known as belief propagation, is a message passing iterative algorithm that is used to perform inference problems to find marginal distributions of unobserved nodes on factor graphs. It is ubiquitous in artificial intelligence algorithms, as well as LDPC and turbo decoding algorithms in information technology. It has been shown that Fourier Analysis [33] and convolution theory algorithms and many other well known algorithms are all special cases of the SPA algorithm.

We recall that each edge on a factor graph corresponds to a variable. Consider the Fig.2.8 for a factor node with N+1 variables. An outgoing message is any message leaving the function block and is the output of the SPA algorithm whilst the incoming messages are the messages entering the function block whose product are involved in the integration operation to compute the outgoing message. In Fig. 2.8, $\overrightarrow{\mu}_{X_1},\overrightarrow{\mu}_{X_2},...,\overrightarrow{\mu}_{X_N}$ are incoming messages whilst $\overrightarrow{\mu}_{X_j}$ is the outgoing message. For the general function block, $\delta(\mathbf{y}-f(\mathbf{x}))$ in Fig. 2.9, the forward message is denoted as $\overrightarrow{\mu}_Y$ and backward message is denoted as $\overleftarrow{\mu}_X$, hence an outgoing message can be a forward or backward message depending on the functional relation of variables.

The computation of the outgoing message, $\overrightarrow{\mu}_{X_j}(\mathbf{x}_j)$, is given by the SPA algorithm as:

$$\overrightarrow{\mu}_{X_j}(\mathbf{x}_j) \propto \sum_{x_{i\neq j}} f(\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_{N+1}) \prod_{i\neq j} \overrightarrow{\mu}_{X_i}(\mathbf{x}_i) \tag{2.11}$$

where summation is replaced by integration for continuous variables.

On the FG in Fig. 2.9, we can compute messages from right, $\overleftarrow{\mu}_Y(\mathbf{y})$ to left, $\overleftarrow{\mu}_X(\mathbf{x})$. Assuming, $f(\mathbf{x},\mathbf{y}) = f(\mathbf{x})f(\mathbf{x},\mathbf{y})$, $\overleftarrow{\mu}_Y(\mathbf{y}) = \mathcal{N}(\mathbf{y},\mathbf{m}_Y,\Sigma_Y)$, $f(\mathbf{x},\mathbf{y}) = \delta(\mathbf{y}-A\mathbf{x})$. To compute the message $\overleftarrow{\mu}_X(\mathbf{x})$:

$$\overleftarrow{\mu}_X(\mathbf{x}) = \int \delta(\mathbf{y}-A\mathbf{x})\,\mathcal{N}(\mathbf{y},\mathbf{m}_Y,\Sigma_Y)\,d\mathbf{y} \tag{2.12}$$

$$= \mathcal{N}(A\mathbf{x},\mathbf{m}_Y,\Sigma_Y) \tag{2.13}$$

2.2.3 Particle representation

Particle representation is a way of expressing general functions or probability density (or mass) functions (usually scaled) as a list of samples with corresponding weights. The weights are assigned to have the property such that $\sum_{k=1}^{K} w_X^k = 1$. Let the particle representation of a general message $\mu_X(x)$ corresponding to a variable $\mathbf{x}$, be given as: $\text{PR}\{\mu_X(x)\} = \mathcal{L}_f = \{x^{(k)}, w^{(k)}\}_{k=1}^{K}$ where $x^{(i)} \in \chi$ is the i^{th} sample drawn from $\mu_X(x)$, χ is the set of all $\mathbf{x}$ realizations, and $w^{(k)} \in \mathbb{R}^+$ are the weights . The weights are such that, for any integrable function $g(x)$ [23]:

$$\int g(x)\mu_X(x)dx \approx \sum_{k=1}^{K} w_X^{(k)} g(x^{(k)}) \qquad \mu_X(x) \approx \sum_{k=1}^{K} w_X^{(k)} \delta(x - x^{(k)}) \tag{2.14}$$

to evaluate the expectation of $g(x)$. Depending on whether the mass points are drawn from a distribution or a general function, the weights can be uniform or non-uniform [23]. It is shown in [23] that particle methods are a full member of message passing techniques such as the SPA algorithm and is even applicable to factor graphs with cycles. It is also shown that when the straight forward application of the SPA is unavailable, (this happens when some of the variables, X_k in the system is continuous and hence the messages can not be determined at that node or that the alphabets of the variable, X_k, is too large) the marginals of variables can still be approximated with particle methods. In this work, we mainly focus on discrete variables, and hence a straight forward application of ([23] section III-A). The main difference in this work however is that each particle $x^{(k)} \in \mathbb{C}^N$, is a waveform and represented as a multidimensional variable of size N.

3

Nonlinear compensation

We recall that one cause of the Kerr effects is high power intensity introduced over a relatively small cross section area of the fiber. Therefore one way to reduce this effect is to increase the cross-sectional area of the fiber optic cable as a way to reduce the magnitude of the nonlinear parameter However, this technique has serious implications and limitations as wider fiber aperture means more modes of propagation which can be detrimental to SMFs. Currently quite a number of receiver-based DSP techniques exist to compensate nonlinear fiber impairments in the electrical domain. In this next section, we will describe some of these DSP techniques and formulate the problem statement of the thesis.

3.1 Nonlinear phase compensation technique

For certain optimized channels, a method described as the NLPN compensation is used to compensate for nonlinear phase shift. We refer to the model described in Fig. 2.1 [27]. Attempts to reduce the effect of the nonlinear phase shift has been made by estimating the phase shift and de-rotating the phase of the received symbols in DSP. Ho and Khan [27], designed a model to compensate the nonlinear phase shift in the absence of CD and polarization mode dispersion (PMD). In zero-compensation fibers the nonlinear phase shift impact on received symbols is given by:

$$\phi_{NL,n} = \gamma L_{eff} \sum_{k=1}^{N_{span}} |x_n + \sum_{l=1}^{k} m_{l,n}|^2 \qquad L_{eff} = \frac{(1 - e^{-\alpha z})}{\alpha} \tag{3.1}$$

where L_{eff} is the effective length, γ is the nonlinear parameter, α is the attenuation factor and the n^{th} transmitted symbol (x_n) and the noise effect on it by the l^{th} amplifier is given by $|x_n + \sum_{l=1}^{k} m_{l,n}|^2$. The noise is an additive white Gaussian noise process drawn

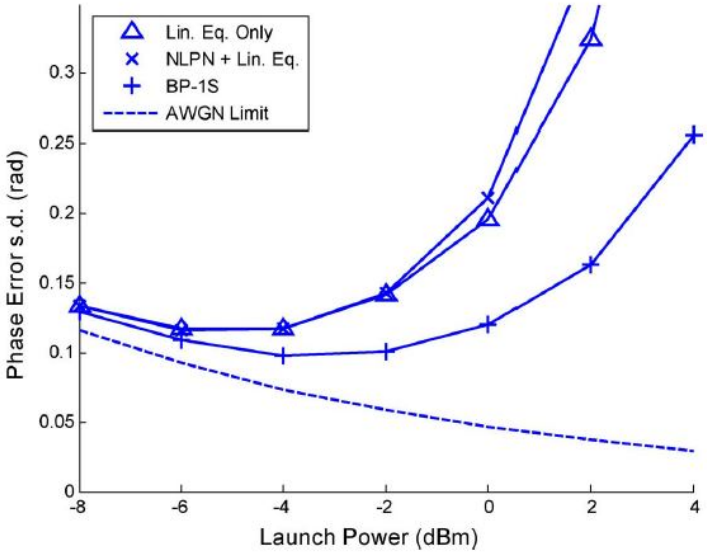


Figure 3.1: Comparison of NLNC and DBP [11].

from $m_{l,n} \sim \mathcal{N}(m; 0, 2\sigma^2)$. At the receiver, compensation is done according to:

$$\hat{\psi}_n = arg(y_n) - \xi |y_n|^2 \tag{3.2}$$

where $arg(y_n)$ is the phase of the n^{th} received symbol, ξ is an optimization parameter whose optimal value is $\xi_{opt} \approx -\gamma L_{eff}(N_{span}span + 1)/2$. It has been shown that this scheme reduces the impact of NLPN by cutting down the noise by a factor of 4 [11]. In the next section, we describe the DBP algorithm and compare its performance with the NLPN compensation technique.

3.2 Digital back propagation

DBP has received a great deal of attention over recent years and is increasingly considered the universal standard for joint linear and nonlinear compensation of optical fiber impairments. The DBP algorithm is based directly on the inverse solution of the NLSE equation using SSFM. In its rigorous form, the DBP operates by applying the inverse to the segments of size Δz, of each fiber in the long haul of concatenated SMF-DCF units. However variations exits that simplify the complexity of original algorithm [11]. The DBP applies opposite transmission parameters $(-\beta_2, -\alpha, -\gamma)$ in the SSFM to undo the linear and nonlinear effects experienced by the received optical pulse. This is done according to:

In the DCF:

$$\tilde{\mathbf{u}}(\omega,z_{n,m}) = \tilde{\mathbf{v}}(\omega,z_{n,m+1}) \exp\left(\frac{j\beta_{2,DCF}L_{DCF}\omega^2}{2}\right) \tag{3.3}$$

$$\mathbf{v}(t,z_{n,m}) = \mathbf{u}(t,z_{n,m}) \exp(-j\xi_{DCF}|\mathbf{u}(t,z_{n,m})|^2) \tag{3.4}$$

$$\xi_{NL,DCF} = \xi\gamma_{DCF}L_{eff,DCF}G \exp\left(-\alpha_{SMF}L_{SMF}\right) \tag{3.5}$$

In the SMF

$$\tilde{\mathbf{u}}(\omega,z_{n,m}) = \tilde{\mathbf{v}}(\omega,z_{n,m+1}) \exp\left(\frac{j\beta_{2,SMF}L_{SMF}\omega^2}{2}\right) \tag{3.6}$$

$$\mathbf{v}(t,z_{n,m}) = \mathbf{u}(t,z_{n,m}) \exp(-j\xi_{SMF}|\mathbf{u}(t,z_{n,m})|^2) \tag{3.7}$$

$$\xi_{NL,SMF} = \xi\gamma_{SMF}L_{eff,SMF} \tag{3.8}$$

where $\mathbf{v}$ and $\mathbf{u}$ are given in Fig. 2.2. A recursive process is applied $L_{n,dcf}/\Delta z$ times in each DCF and $L_{n,SMF}/\Delta z$ in each SMF staring from $\mathbf{v}(t,z_{N_{span},N_{seg}})$ for N_{span} spans until the transmitted signal, $\mathbf{v}(t,z_{0,0})$ is computed. Whether the DCF or SMF subroutine is computed first, depends on the system architecture. In (3.3) - (3.8), the DCF subroutine is executed first in each fiber span because it was designed according to the system model in 2.1 (post-compensation link). The presence of noise from the EDFAs (when present), cause the resulting computed $\hat{\mathbf{v}}(t,z_{1,1})$ to deviate from the original transmitted optical signal, $\mathbf{v}(t,z_{1,1})$.

Results: In Fig. 3.1, a comparison of the phase errors of detected symbols of three algorithms are shown, namely; a linear equalizer which compensates only for linear impairments (shown as "Lin. Eq. Only"), the nonlinear phase noise compensation technique (shown as "NLPN + Lin. Eq") and the DBP algorithm ("BP - 1S"). The experiment consists of N_{span} spans of SMF-EDFA-DCF-EDFA units which are concatenated to form a hypothetical long haul terrestrial link. $D_{SMF} = 17$ps/nm/km with $L_{SMF} = 80$km. $D_{DCF} = -80$ps/nm/km with $L_{DCF} = 15.3$km. Hence $D_{DCF}L_{DCF} + D_{SMF}L_{SMF} = 136 \neq 0$ which is an under-compensation of $L_{DCF} = 1.7$km or $\frac{136}{1360} \times 100 = 10\%$.
In the linear region, all the algorithms perform close to the additive white Gaussian noise (AWGN) limit. However, in the nonlinear power dependent region, as signal propagates through long distances, non-zero nonlinear effects and significant CD accumulates in the fiber and can not be ignored. Hence the NLPN compensation technique phase model presented in (3.1) is far from correct in such under-compensated channels. In that regime, the DBP algorithm outperforms the NLPN algorithm. [11] shows that better performance of the DBP is achieved when the dispersion in the SMF is under-compensated whilst the linear compensation is performed in DSP.

However, for channels with periodic amplification using optical amplifiers such as EDFAs, the DBP algorithm is sub-optimal since it does not accounted for the presence of noise in the system. In the next section, we study the SDBP algorithm, which aims at optimally minimizing detection errors in systems that consists of optical amplifier noise.

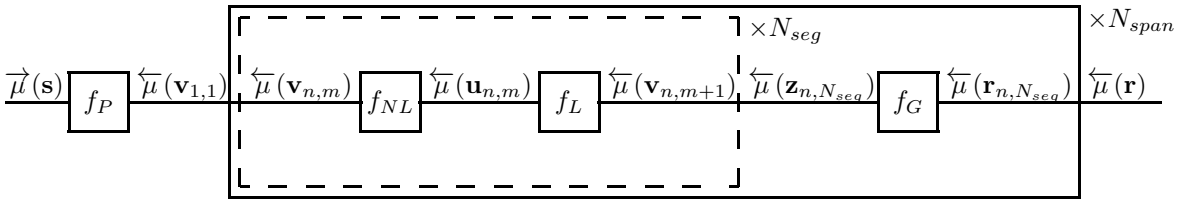


Figure 3.2: Factor graph representation of system model. f_P represents the pulse shaping block, f_{NL} represents the nonlinear block, f_L represents the linear block and f_G corresponds to the EDFA block

.

3.3 Stochastic DBP

It is seen that none of the methods described above account for the nonlinear signal to (ASE) noise interaction (nonlinear signal to noise interaction (NSNI)). The ASE noise is a result of the spontaneous emissions from the amplifier, characterized by a large bandwidth. However a low pass filter is used to filter out the noise outside the bandwidth of the optical signal leaving only the NSNI noise. Is has been shown that the NSNI has significant penalty on the system performance in proportion to symbol rate and modulation formats for dispersion managed links systems with inline amplification [7]. In the SDBP algorithm, the NSNI is factored into developing a novel receiver by accounting for ASE noise. It is shown that the SDBP method outperforms the DBP algorithm in some power regimes. It is also shown that in non-dispersion managed links, the difference in performance of the two algorithms is even more significant [7]. Unlike the DBP, the SDBP is developed with the aid of message passing on factor graphs and the SPA algorithm since it accounts for the stochastic property of the information signal due to the presence of the amplifier noise. The algorithm approximates each message with particles of size N_p. Similar to the DBP, the SDBP performs an iterative back propagation by segmentation the fiber into N_{seg} sections each of size Δz. However, unlike the DBP where optical signals are computed at each stage, distributions (messages) are computed in the SDBP algorithm. The message computation is done for each sub-block; each consisting of building blocks in the SSFM i.e, linear block (f_L), a nonlinear block (f_{NL}), and an EDFA sub-block (f_G) as seen in Fig. 3.2. We show how the messages are computed in the system model blocks in Fig. 3.2.

Message computation in EDFA (f_G)

In Fig. 3.2, the message $\overleftarrow{\mu}(r) = \overleftarrow{\mu}(r_{N_{span},N_{seg}})$ is represented in a particle representation form as :
$\frac{1}{N_p}\sum_{k=1}^{N_p}\delta(\mathbf{r}_{n,N_{seg}} - \mathbf{r}^{(k)}_{n,N_{seg}})$. We also write that $\mathbf{r}_{n,N_{seg}} = \sqrt{G}\mathbf{z}_{n,N_{seg}} + n$ for $n \sim \mathcal{N}(n;0,\Sigma)$. Hence the factor function representation of the EDFA block, f_G is given as $f_G =$

$\mathcal{N}(\mathbf{r}_{n,N_{seg}}; \sqrt{G}\mathbf{z}_{n,N_{seg}}, \mathbf{\Sigma})$. Then:

$$\mu_{\mathbf{Z}_{n,N_{seg}}}(\mathbf{z}_{n,N_{seg}}) = \frac{1}{N_p}\int \mathcal{N}(\mathbf{r}_{n,N_{seg}}; \sqrt{G}\mathbf{z}_{n,N_{seg}}, \mathbf{\Sigma}) \sum_{k=1}^{N_p} \delta(\mathbf{r}_{n,N_{seg}} - \mathbf{r}^{(k)}_{n,N_{seg}}) d\mathbf{r}_{n,N_{seg}}$$
$$= \frac{1}{N_p}\sum_{k=1}^{N_p} \mathcal{N}(\mathbf{r}^{(k)}_{n,N_{seg}}; \sqrt{G}\mathbf{z}_{n,N_{seg}}, \Sigma)$$

then N_p particles of $\mathbf{z}_{n,N_{seg}}$ are generated according to $\mathbf{z}^{(k)}_{n,N_{seg}} \sim \mathcal{N}(\mathbf{z}_{n,N_{seg}}; \mathbf{r}^{(k)}_{n,N_{seg}}/\sqrt{G}, \Sigma/G)$.

Message computation in linear block (f_L)

From Fig. 3.2 we have that, $\mathbf{z}_{n,N_{seg}} = \mathbf{v}_{n,m+1}$ when $m = N_{seg}$. In the linear block, our aim is to compute the particle representation (PR) for $\overleftarrow{\mu}_{\mathbf{U}_{n,m}}$ given the PR of $\overleftarrow{\mu}_{\mathbf{V}_{n,m+1}}$. The two variables are related by: $\mathbf{v}^{(k)}_{n,m+1} = F\mathbf{u}^{(k)}_{n,m} * h(t,\Delta z)$, hence the factor block representation $f_L : p(\mathbf{v}_{n,m+1}|\mathbf{u}_{n,m}) = \delta(\mathbf{v}_{n,m+1} - F\mathbf{u}_{m,n} * \mathbf{h}(\Delta z))$. Then solving for $\mathbf{u}^{(k)}_{n,m}$ will involve a deconvolution operation. Since it is computationally effective to work in the frequency domain, the FFT of the input-output variables will be used. This is done as follows [7]:
let PR $\{\overleftarrow{\mu}_{\mathbf{U}_{n,m}}\}$ be the particle list of $\overleftarrow{\mu}_{\mathbf{U}_{n,m}}(\mathbf{u}_{n,m})$ given as $\{\mathbf{u}^{(k)}_{n,m}\}_{k=1}^{N_p}$. The IFFT and FFT of $\mathbf{v}_{n,m}$ be IFFT($\mathbf{v}_{n,m}$) and FFT($\mathbf{v}_{n,m}$) respectively.

$$PR\{\overleftarrow{\mu}_{\tilde{\mathbf{U}}_{n,m+1}}\} = \{FFT(\mathbf{v}^{(k)}_{n,m+1})\}_{k=1}^{N_p}$$

$$PR\{\overleftarrow{\mu}_{\tilde{\mathbf{U}}_{n,m}}\} = \{F^{-1}\tilde{\mathbf{h}}^{-1}(\omega,\Delta z)\tilde{\mathbf{u}}^{(k)}_{n,m+1}\}_{k=1}^{N_p}$$

$$PR\{\overleftarrow{\mu}_{\mathbf{U}_{n,m}}\} = \{IFFT(\tilde{\mathbf{u}}^{(k)}_{n,m})\}_{k=1}^{N_p}$$

since FFT($\bullet$) and IFFT($\bullet$) are bijective functions on vectors.

Message computation in nonlinear block (f_{NL})

In this block we wish to compute PR $\{\overleftarrow{\mu}_{\mathbf{V}_{n,m}}\} = \{\mathbf{v}^{(k)}_{n,m}\}_{k=1}^{N_p}$ when we have PR $\{\overleftarrow{\mu}_{\mathbf{U}_{n,m}}\}$ $= \{\mathbf{u}^{(k)}_{n,m}\}_{k=1}^{N_p}$. From Fig. 2.2, we know that the input-output relationship for the nonlinear block is given as:

$$\mathbf{v}^{(k)}_{n,m} = \mathbf{u}^{(k)}_{n,m}\exp(-j\gamma\Delta z|\mathbf{u}^{(k)}_{n,m}|^2) \tag{3.10}$$

Hence the factor graph function representation:

$$f_{NL} : p(\mathbf{u}_{n,m}|\mathbf{v}_{n,m}) = \delta(\mathbf{u}_{n,m} - \mathbf{v}_{n,m}\exp(j\gamma\Delta z)|\mathbf{v}_{n,m}|^2)$$

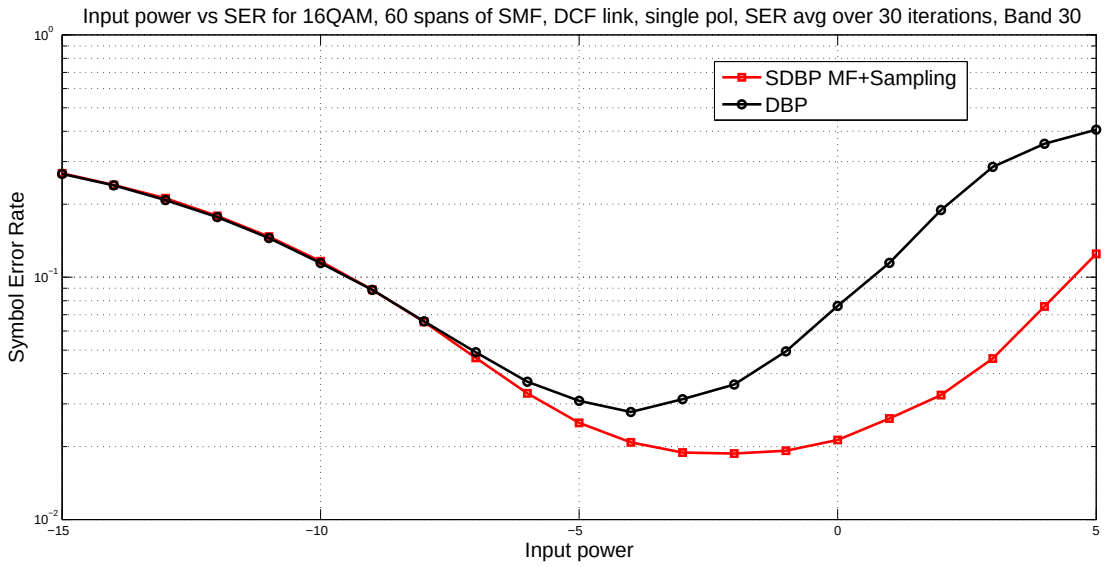


Figure 3.3: Comparison of SDBP with DBP. At 0dBm, SDBP has 3.5 times less errors compared to DBP.

meaning for every particle, $\mathbf{u}_{n,m}^{(k)}$, we de-rotate the phase using the product in the exponential argument which is proportional to the power of the waveform.

We continue the iterative process for all N_{span} fiber spans and for each span n, iterating for N_{seg} segments until we arrive with PR$\{\overleftarrow{\mu}_{\mathbf{v}_{1,1}}\}$.

Detection:

In general, to compute the marginal distribution for a variable $\mathbf{S}$ in factor graphs, we compute the product of the messages in the two directions on the edge, i.e, $\mu_{\mathbf{S}}(s) = \overrightarrow{\mu}_{\mathbf{S}}(s) \times \overleftarrow{\mu}_{\mathbf{S}}(s)$. In the SDBP algorithm, since back propagation is always done with a fixed $\mathbf{r}$, the product of $\overrightarrow{\mu}_{\mathbf{S}}(s) \times \overleftarrow{\mu}_{\mathbf{S}}(s)$ yields the a posteriori distribution, $p(\mathbf{s}|\mathbf{r})$. The aim is to make decisions on the most likely transmitted symbol sequence, by maximizing the posterior distribution over $\mathbf{s}$, i.e $\hat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmax}}\, p(\mathbf{s}|\mathbf{r})$. However $p(\mathbf{s}|\mathbf{r})$ is not readily available.

Since $\mathbf{v}_{1,1} = \mathbf{s} * \mathbf{p}$ (factor graph representation is $f_P : p(\mathbf{v}_{1,1}|\mathbf{s}) = \delta(\mathbf{v}_{1,1} - \mathbf{s} * \mathbf{p})$), the availability of $p(\mathbf{v}_{1,1}|\mathbf{r})$ gives rise to $p(\mathbf{s}|\mathbf{r})$ directly. The posterior distribution $p(\mathbf{v}_{1,1}|\mathbf{r})$ can be found by multiplying the forward and backward messages on the edge of the variable $\mathbf{v}_{1,1}$, $\overleftarrow{\mu}_{\mathbf{v}_{1,1}} \times \overrightarrow{\mu}_{\mathbf{v}_{1,1}}$. PR$\{\mu_{\mathbf{v}_{1,1}}\}$ is the particle representation of $\overleftarrow{\mu}_{\mathbf{v}_{1,1}}$ whilst $\overrightarrow{\mu}_{\mathbf{v}_{1,1}}$ comes directly from $\overrightarrow{\mu_{\mathbf{s}}}$. However a multiplication of the two messages ($\overleftarrow{\mu}_{\mathbf{v}_{1,1}} \times \overrightarrow{\mu}_{\mathbf{v}_{1,1}}$) requires that both messages are available for the same $\mathbf{v}_{1,1}$. PR$\{\mu_{\mathbf{v}_{1,1}}\}$ can be reconstructed into a continuous function, by the use of kernel density estimation [28] or a Gaussian process [29].

But the problem arises where the complexity of the maximization operation, i.e

Algorithm 3 Detection in SDBP receiver using matched filtering and symbol rate sampling

1: **procedure** SYMBOL-BY-SYMBOL (SBS) SDBP
2: **for** each particle $(k) \in \{1,...,N_p\}$ **do**
3: $e_{MF}^{(k)}(t) = \mathbf{v}_{1,1}^{(k)} * p(t)$
4: $r_{MF}^{(k)}[n] = e_{MF}^{(k)}(nT_s)$
5: **end for**
6: **end procedure**

$\underset{\mathbf{s}}{\text{argmax}}\, p(\mathbf{s}|\mathbf{r})$, increases exponentially with M, i.e $|\Omega|^M$, where M is the dimension of $\mathbf{s}$. In the SDBP algorithm, complexity is reduced to a symbol by symbol detector. Each particle in PR $\{\mu_{\mathbf{v}_{1,1}}\}$ $\mathbf{v}_{1,1}^{(k)} \in \mathbb{C}^N$ is filtered with a filter that is matched to the transmit pulse. This is then followed by symbol rate sampling at optimal sampling times to maximize the SNR [30], [31] as shown in Algorithm 3. The output of the sampler for particle (k) is $\mathbf{r}_{MF}^{(k)} = [r_{MF}^{(k)}[1],...,r_{MF}^{(k)}[M]]$ where $r_{MF}^{(k)}[n] \in \mathbb{C}$ corresponding to symbol $\mathbf{s}[n]$. The final product is a particle representation, $\{r_{MF}^{(k)}[n]\}_{(k=1)}^{N_p}$ representing $p(r_{MF}[n]|\mathbf{s}[n])$. In this way, complexity increases linearly as $M \times |\Omega|$. r_{MF} is assumed to be Gaussian whose variance and mean of each sample $(r_{MF}[n])$ is given as:

$$\Sigma[n] = \frac{1}{N_p}\sum_{k=1}^{N_p}(r_{MF}^{(k)}[n] - m[n])(r_{MF}^{(k)}[n] - m[n])^T \qquad m[n] = \frac{1}{N_p}\sum_{\forall k} r_{MF}^{(k)}[n] \quad (3.11)$$

We know that,

$$p(s[n]|\mathbf{r}) = \frac{c \times \exp\left(-\frac{1}{2}(s[n] - m[n])^T\Sigma[n]^{-1}(s[n] - m[n])\right) \times p(\mathbf{s}[n])}{p(\mathbf{r})} \quad (3.12)$$

where c is a constant. But since $p(r[n])$ is the same in the maximization operation, we ignore it in the argmax operation. We also assume a uniform distribution of $p(s[n])$ which can also be ignored in the argmax operation. Hence:

$$p(s[n]|\mathbf{r}) \approx c \times \exp\left(-\frac{1}{2}(s[n] - m[n])^T\Sigma[n]^{-1}(s[n] - m[n])\right) \quad (3.13)$$

Results: Fig. 3.3, shows the results of a simulation of symbol error rate (SER) vs launch power in an optical communication system to compare the performance of the SDBP and DBP algorithms. 60 spans of SMF-DCF-EDFA units are used, the optical signal is transmitted on a single polarization and all PMDs are absent. The SER is averaged over 30 iterations and the modulation format is 16 QAM. An oversampling rate of 4 is applied at the transmitter. The receiver pulse shaping is a root raised cosine

pulse and N_p=500 particles are used in the SDBP. In both cases a matched filter is applied, however in the case of the SDBP, the matched filter is applied to all particles and detection is done according to (3.13). We observe than in the linear region where the AWGN dominates, both algorithms have almost the same performance. However within the power regimes of -2 dB to 5dB, when the nonlinearities begin to dominate the SDBP is more tolerant to the nonlinearities compared to the DBP and outperforms the DBP.

3.4 Limitation of SDBP

In the detection unit of the SDBP algorithm, we can write in message passing terms, the backward message of the pulse shaping function $p(\mathbf{r}|\mathbf{s})/p(\mathbf{r})$, as $\overleftarrow{\mu}_{\mathbf{s}}(\mathbf{s})$. Whilst $p(\mathbf{s})$ is the incoming message to the pulse function, $\overrightarrow{\mu}_{\mathbf{s}}(\mathbf{s})$ as seen in Fig. 3.2. If PR$\{\mu_{\mathbf{v}_{1,1}}\}$ is assumed to be Gaussian with sample mean $\boldsymbol{m}_{\mathbf{V}1,1}$ and covariance $\Sigma_{\mathbf{V}_{1,1}}$, then the matched filtering operation according to algorithm 3 will yield a Gaussian PR$\{r_{MF}^{(k)}\}$, with mean and covariance in vector-matrix representation [APPENDIX A]:

$$\hat{\boldsymbol{m}}_{\mathbf{r}_{MF}} = (A^T A)^{\#} A^T \boldsymbol{m}_{\mathbf{V}_{1,1}} \tag{3.14}$$

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{r}_{MF}} = (A^T A)^{\#} A^T \Sigma_{\mathbf{V}_{1,1}} A (A^T A)^{\#} \tag{3.15}$$

where $(*)^{\#}$ is the Moore-Penrose inverse and A is the matrix representation of the pulse shaping. If A is a root-raised-cosine pulse, $A^{\#} = A^T$ and $\hat{\boldsymbol{m}}_{\mathbf{r}_{MF}}[n] = A^T \boldsymbol{m}_{\mathbf{V}_{1,1}}$, $\hat{\boldsymbol{\Sigma}}_{\mathbf{r}_{MF}} = A^T \Sigma_{\mathbf{V}_{1,1}} A$. The derivation of $p(\mathbf{s}|\mathbf{r})$ based on algorithm 3, i.e using matched filtering and symbol rate sampling, is based on heuristics hence there is no claim that it optimally computes the correct backward message, $\overleftarrow{\mu}_S(\mathbf{s})$.

In the next chapter, we investigate if the matched filtering with symbol rate sampling are optimal methods for deriving the true posteriori distribution in the SDBP detector. We also derive a true near-MAP inference detector using message passing and the SPA algorithm on FFGs by computing the backward message, $\overleftarrow{\mu}_S(\mathbf{s})$ from the particle representation PR$\{\overleftarrow{\mu}_{V_{1,1}}(\mathbf{v}_{1,1})\}$. Following that, we introduce some implementation challenges and propose solutions to them.

4

Proposed Approach

4.1 Computing messages

In this chapter, we show how to compute forward and backward messages in linear and nonlinear models and highlight the reason why backward message computation for nonlinear models can be tedious. We also show the special case when all messages are Gaussian in a linear model and address some implementation issues such as the small sample size problem. With the help of some numerical examples, we demonstrate the computation of backward messages in different linear scenarios and finally apply the operation to a practical optical communication system in comparison with some benchmark methods such as the DBP and the SDBP with matched filtering and symbol rate sampling.

Notation: As in the previous chapter, messages are represented with overhead arrows, with incoming messages of $\mathbf{y}$ being denoted as $\overleftarrow{\mu}_Y$ and outgoing messages of $\mathbf{y}$ denoted as $\overrightarrow{\mu}_Y$ as shown in Fig. 2.9. $f(\mathbf{x},\mathbf{y})$ denotes a general function (linear and nonlinear) of $\mathbf{x}$ and $\mathbf{y}$. $\mathbf{y} = A\mathbf{x}$ represents a linear function (which could either be a matrix or a vector) whilst $\mathbf{y} = \psi(\mathbf{x})$ represents a nonlinear function. $\chi(\mathbf{y})$ represents the set of all possible values of $\mathbf{x}$ that gives $\mathbf{y}$ for the function $\mathbf{y} = f(\mathbf{x})$. $|\chi(\mathbf{y})|$ denotes the cardinality of the set $\chi(\mathbf{y})$.

4.1.1 Forward message for (non)linear channels

Forward computation is relatively simple to compute. The outgoing message $\overrightarrow{\mu}_Y(\mathbf{y})$, from a function block $\mathbf{y} = \psi(\mathbf{x})$ given an incoming message, $\overrightarrow{\mu}_X(\mathbf{x})$, is given as [6]:

$$\overrightarrow{\mu}_Y(\mathbf{y}) = \sum_{\mathbf{x}\in\chi(\mathbf{y})} \frac{\overrightarrow{\mu}_X(\mathbf{x})}{|J(\psi(\mathbf{x}))|} \tag{4.1}$$

where $\chi(\mathbf{y}) = \mathbf{x}|\psi(\mathbf{x}) = \mathbf{y}$ is finite ($0<|\chi(\mathbf{y})|< \infty$) or empty, $|J(\psi(\mathbf{x}))|$ is the absolute determinant of the Jacobian of the function $\mathbf{y} = \psi(\mathbf{x})$.

If the incoming message is given in a particle list i.e, $\mathrm{PR}\{\overrightarrow{\mu}_X(\mathbf{x})\} = \{\mathbf{x}^{(k)}, w^{(k)}\}_{k=1}^{K}$. Hence $\overrightarrow{\mu}_X(\mathbf{x}) = \sum_{k=1}^{K} w_X^{(k)} \delta(\mathbf{x} - \mathbf{x}^{(k)})$ the outgoing message is given as [6]:

$$\overrightarrow{\mu}_Y(\mathbf{y}) \approx \sum_{k=1}^{K} w_X^{(k)} \delta(\mathbf{y} - \psi(\mathbf{x}^{(k)}))$$

with the accuracy of the computed message approaching the real value as number of particles K, approaches ∞. Note here however that the weights in the particle list for outgoing and incoming messages are the same, i.e $\{ w_Y^{(k)} = w_X^{(k)}, \mathbf{y}^{(k)} = \psi(\mathbf{x}^{(k)})\}$ where particle list of message $\overrightarrow{\mu}_Y(\mathbf{y})$, $\mathrm{PR}\{\overrightarrow{\mu}_Y(\mathbf{y})\}$ is $\{ w_Y^{(k)}, \mathbf{y}^{(k)}\}_{k=1}^{K}$. This applies to both linear and nonlinear functions.

4.1.2 Backward message for (non)linear models

We first consider the general model case and later consider the linear Gaussian case. In general, computing the backward message can be tedious especially for surjection functions and with intractable message representations. Here we say the incoming message is $\overleftarrow{\mu}_Y(\mathbf{y})$ whilst the outgoing message is given by $\overleftarrow{\mu}_X(x)$. If $\overleftarrow{\mu}_Y(\mathbf{y})$ is given as a particle list $\mathrm{PR}\{\overleftarrow{\mu}_Y(\mathbf{y})\} = \{ w_Y^{(k)}, \mathbf{y}^{(k)}\}_{k=1}^{K}$, i.e $\overrightarrow{\mu}_Y(\mathbf{y}) = \sum_{k=1}^{K} w_Y^{(k)} \delta(\mathbf{y} - \mathbf{y}^{(k)})$, then the outgoing message is given as [6]:

$$\overleftarrow{\mu}_X(\mathbf{x}) = \int \delta(\mathbf{y} - \psi(\mathbf{x})) \overleftarrow{\mu}_Y(\mathbf{y}) d\mathbf{y} \tag{4.2}$$

$$\approx \sum_{k=1}^{K} w_Y^{(k)} \delta(\mathbf{y}^{(k)} - \psi(\mathbf{x})) \tag{4.3}$$

For each particle $\mathbf{y}^{(k)}$, there exist $|\chi(\mathbf{y}^{(k)})|$ particles according to the inverse function $\psi^{-1}(\mathbf{y}^{(k)})$ with weights equal to $w_Y^{(k)}$. The particle list is of $\overleftarrow{\mu}_X(\mathbf{x})$ is therefore given as: $\{ \mathbf{x}^{(k,1)}, \mathbf{x}^{(k,2)}, ..., \mathbf{x}^{(k,|\chi(\mathbf{y}^{(k)})|)} : w_X^k = w_Y^k \}$. In this case there are a total of $\sum_{k=1}^{K} |\chi(\mathbf{y}^{(k)})|$ particles. If $|\chi(\mathbf{y}^{(k)})| > 1$ for any $\mathbf{y}$ the function $\psi(\mathbf{x})$ is surjective. If $|\chi(\mathbf{y}^{(k)})| \leq 1$, $\psi(\mathbf{x})$ is injective and if $|\chi(\mathbf{y}^{(k)})| = 1$, $\psi(\mathbf{x})$ is bijective and invertible.

It is clear from the above that tractable forms of expressing $\psi^{-1}(\mathbf{y})$ and a non zero $\chi(\mathbf{y}^{(k)})$ set is necessary for computing the outgoing particle list $\mu_X(\mathbf{x})$. In the case of an injective function when $|\chi(\mathbf{y}^{(k)})| = 0$ for all or for a certain (k) i.e particle(s) do not exist on the $\psi(\mathbf{x})$ subspace, $\mathbf{x}^{(k)}$ has no mass and can not be represented in particle representation (PR). This problem is overcome when $\psi(\mathbf{x})$ is surjective or bijective.

However for linear models with Gaussian messages, a closed form solution is possible for both forward and backward message passing [6]. The forward case is according to

section 4.1.1 where the Jacobian becomes the matrix representation of the linear function. Consider a linear system in the form $\mathbf{y} = A\mathbf{x}$, for $A_{N\times M}$, if $\mathbf{x}$ is Gaussian, then $\mathbf{y}$ will also be Gaussian (and vice versa) according to equation 4.1 (since the determinant of the Jacobian is the determinant of A). In the backward case, if $f(\mathbf{x}) = \delta(\mathbf{y} - A\mathbf{x})$ and the incoming message, $\overleftarrow{\mu}_Y$ is given as $\mathcal{N}(\mathbf{y}; \mathbf{m}_Y, \Sigma_Y)$, we can apply the SPA algorithm to compute the backward message $\overleftarrow{\mu}_X(\mathbf{x})$, as follows:

$$\overleftarrow{\mu}_X(\mathbf{x}) = \int \delta(\mathbf{y} - A\mathbf{x})\mathcal{N}(\mathbf{y}; \mathbf{m}_Y, \Sigma) \tag{4.4}$$

$$= \mathcal{N}(A\mathbf{x}; \mathbf{m}_Y, \Sigma_Y) \propto \mathcal{N}(\mathbf{x}; \mathbf{m}_X, \Sigma_X) \tag{4.5}$$

$$= s \times \mathcal{N}\big(\mathbf{x}; \mathbf{m}_X, \Sigma_X\big) \tag{4.6}$$

where the constant of proportionality, s is given as:

$$s = \exp\left(-\frac{1}{2}(\mathbf{m}_Y^T \Sigma_Y^{-1} \mathbf{m}_Y - \mathbf{m}_X^T \Sigma_X^{-1} \mathbf{m}_X)\right) \tag{4.7}$$

In [6], the equations used for computing the mean and covariances are:

$$\Sigma_X^{-1} = A^H \Sigma_Y^{-1} A \tag{4.8}$$

$$\mathbf{m}_X = (A^H \Sigma_Y^{-1} A)^{\#} A^H \Sigma_Y^{-1} \mathbf{m}_Y \tag{4.9}$$

where Σ_X is covariance of X and Σ_Y is the covariance of Y. The $(A)^{\#}$ is the Moore-Penrose pseudo-inverse whose definition is given as $A^{\#} = (A^*A)^{-1}A^*$ when (A) has full column rank and $A^{\#} = A^*(AA^*)^{-1}$ when (A) has full row rank.

4.1.3 Backward Gaussian particle message passing for linear models

GMP - Slicing: If the incoming message, $\overleftarrow{\mu}_Y(\mathbf{y})$ is represented by a particle list, we can approximate the outgoing message $\overleftarrow{\mu}_X(\mathbf{x})$ in closed form. By assuming that the particles are drawn from a Gaussian distribution [6], the outgoing message can be computed by estimating the incoming message $\overleftarrow{\mu}_Y(\mathbf{y})$ according to:

$$\mathbf{m}_Y = \frac{1}{K}\sum_{k=1}^{K} \mathbf{y}^k \qquad \hat{\Sigma}_Y = \frac{1}{K-1}\sum_{k=1}^{K} (\mathbf{y}^k - \mathbf{m}_Y)(\mathbf{y}^k - \mathbf{m}_Y)^T \tag{4.10}$$

where $Y \in \mathbb{C}^N$ and K is the number of samples (particles). This sample covariance, $\hat{\Sigma}_Y$, is the Maximum Likelihood estimate (MLE) of the covariance Σ_Y. We then apply (4.8) and (4.9).
If $\mu_Y(y)$ is a mixture of Gaussian components we can:

- apply a uniform weight, $w^{(k)}$ to each mixture component in the linear combination. Thus we have:

$$\overleftarrow{\mu}_Y(\mathbf{y}) = \sum_{k=1}^{\tilde{K}} w_Y^{(k)} \mathcal{N}(\mathbf{y}, \mathbf{m}_Y^{(k)}, \Sigma_Y^{(k)}) \tag{4.11}$$

- apply (4.4) to $\overleftarrow{\mu}_Y(\mathbf{y})$ to compute outgoing message, $\overleftarrow{\mu}_X(\mathbf{x})$

$$\overleftarrow{\mu}_X(\mathbf{x}) = \int \delta(\mathbf{y} - A\mathbf{x}) \sum_{k=1}^{\tilde{K}} w_Y^{(k)} \mathcal{N}(\mathbf{y}, \mathbf{m}_Y^{(k)}, \Sigma_Y^{(k)}) dx \tag{4.12}$$

$$= \sum_{k=1}^{\tilde{K}} \int \delta(\mathbf{y} - A\mathbf{x}) w_Y^{(k)} \mathcal{N}(\mathbf{y}, \mathbf{m}_Y^{(k)}, \Sigma_Y^{(k)}) d\mathbf{x} \tag{4.13}$$

$$= \sum_{k=1}^{\tilde{K}} w_Y^{(k)} \mathcal{N}(A\mathbf{x}; \mathbf{m}_Y^{(k)}, \Sigma_Y^{(k)}) = \sum_{k=1}^{\tilde{K}} w_Y^{(k)} s^{(k)} \mathcal{N}(\mathbf{x}; \mathbf{m}_X^{(k)}, \Sigma_X^{(k)}) \tag{4.14}$$

$$= \sum_{k=1}^{\tilde{K}} w_Y^{(k)} s^{(k)} \mathcal{N}(\mathbf{x}; (A^H (\Sigma_Y^{(k)})^{-1} A)^{\#} A^H (\Sigma_Y^{(k)})^{-1} \mathbf{m}_Y^{(k)},\ A^H (\Sigma_Y^{(k)})^{-1} A) \tag{4.15}$$

each Gaussian is scaled by the proportionality factor, s, since each Gaussian mixture will have different contributions to the $A\mathbf{x}$ subspace depending on its distance from $A\mathbf{x}$:

$$s^{(k)} = \exp\left(-\frac{1}{2} (\mathbf{m}_Y^{(k),T} (\Sigma_Y^{(k)})^{-1} \mathbf{m}_Y^{(k)} - \mathbf{m}_X^{(k),T} (\Sigma_Y^{(k)})^{-1} \mathbf{m}_X^{(k)}) \right). \tag{4.16}$$

We see from (4.8) - (4.9) that the mean and covariance of $\overleftarrow{\mu_X}(\mathbf{x})$ is a function of the inverse covariance of $\overleftarrow{\mu}_Y(\mathbf{y})$, i.e $\{\mathbf{m}_X, \Sigma_X\} = f(\Sigma_Y^{-1})$. The accuracy of $\overleftarrow{\mu}_X(x)$ therefore heavily depends on the accuracy on estimation of Σ_Y.
It is seen from (4.4) that the evaluation of $\overleftarrow{\mu}_X(\mathbf{x})$ can be done by evaluating $\overleftarrow{\mu}_Y(\mathbf{y})$ on the $\mathbf{y} = A\mathbf{x}$ subspace for the case when the linear function A is injective, which can be interpreted as "slicing" $\overleftarrow{\mu}_Y(\mathbf{y})$ along $\mathbf{y} = A\mathbf{x}$ subspace.

Projection: Consider the special case when $\Sigma_Y^{(k)} = \sigma^2 I$, $\mathbf{m}_X^{(k)} = \mathbf{y}^{(k)}$, and $K = 1$ (one Gaussian PDF). In this case [6]:

$$\mathbf{m}_X = \left(A^T \frac{I}{\sigma^2} A \right)^{\#} A^T \left(\frac{I}{\sigma^2} \right) \mathbf{y} = \left(A^T A \right)^{\#} A^T \mathbf{y} \tag{4.17}$$

$$\Sigma_X^{-1} = (A^T \Sigma_Y^{-1} A) \equiv \frac{1}{\sigma^2} (A^T A) \equiv Q \mathbf{diag} \left[\frac{\lambda_1}{\sigma^2} \cdots \frac{\lambda_L}{\sigma^2} 0 \cdots 0 \right] Q^T \tag{4.18}$$

where λ_1,...λ_L are the strictly positive eigenvalues of $A^T A$ and Q contains the corresponding eigenvectors. The last $M - L$ columns of Q are the eigenvectors that span the null space of $A^T A$ thereby making $\mathbf{z}^T A^T A \mathbf{z} = 0$ for those eigenvectors, $\mathbf{z}$.
Applying the SPA for the above case gives:

$$\overleftarrow{\mu}_X(\mathbf{x}) = s\, w_Y \mathcal{N}\left(\mathbf{x}; (A^T A)^{\#} A^T \mathbf{y},\ \sigma^2 (A^T A)^{-1} \right) \tag{4.19}$$

where $s\, w_Y = 1$.

This means that for non-bijective (only surjective or only injective) linear functions, the projection technique can not yield a meaningful Gaussian PR since (4.19) can not have a valid covariance matrix. One way of looking at (4.19) is as follows: to compute $\overleftarrow{\mu}_X(\mathbf{x})$, each particle, $\mathbf{y}^{(k)}$, which may not exist on the $A\mathbf{x}$ subspace is projected onto it using (4.17) and becomes the mean of the PDF.

At this point, we observe that (4.17) - (4.18)) i.e projection technique is the same as (3.14) - (3.15) in Chapter 2, section 3.3. In order words, performing a projection of particles is equivalent to matched filtering and sampling. Since the projection is only applicable for the case where Σ_Y is a diagonal matrix, and since the more general technique is the GMP technique, we conclude that matched filtering is not optimal in computing the posterior distribution in the SDBP algorithm. However, if the incoming message particles can be approximated with a Gaussian distribution, the backward outgoing message can be computed with the slicing (GMP) method yielding an optimal solution since Gaussian message passing is exact in linear models. We illustrate these ideas in the examples in section 4.1.5.

4.1.4 Computation of GMP for large dimensions

We have seen a way to compute backward messages in linear models for particles list that are drawn from a Gaussian distribution. Equations 4.8 - 4.9 is referred to hereafter as the slicing operation or GMP. The resulting message, $\overleftarrow{\mu}_X(\mathbf{x})$, is Gaussian whose mean and covariance is scaled with the inverse covariance, Σ_Y^{-1}, of the incoming message, $\overleftarrow{\mu}_Y(\mathbf{y})$ (4.12). If the incoming message is represented by a particle list and a Gaussian approximation of the incoming message is done according to the GMP technique, the mean and covariance of the incoming message, $\overleftarrow{\mu}_Y(y)$, can be computed with 4.10.

In the case where $K < N$, where K is the number of particles and N is the dimension of the variable Y, we have the small sample size problem: the observation sample size is smaller than the number of classes (dimensions). The resulting sample covariance, $\hat{\Sigma}_Y$, will then be rank deficient and with linearly dependent rows (or columns). Hence $\hat{\Sigma}_Y$ will be singular and noninvertible. Even in the case when $K > N$, the fraction, $\frac{N}{K}$ will have to be extremely small ($\frac{\log N}{K} \to 0$) for the sample covariance to be properly scaled and well conditioned. If not, computing the inverse of $\hat{\Sigma}_Y$ will give unstable results, which will yield infinite elements of $\hat{\Sigma}_Y^{-1}$. This is observed in Fig. 4.1 where the least $N - K$ eigenvectors correspond to zero eigenvalues of the undersampled covariance matrix. We will now investigate how to solve this problem of under-sampled covariance matrix for large dimensions.

A. Banding

We know that a covariance matrix must: (i) be positive semi-definite and (ii) be symmetric. A positive definite matrix, $\hat{\Sigma}_Y$, always satisfies the condition: $\mathbf{z}^T\Sigma_Y\mathbf{z} > 0$, for any non zero vector $\mathbf{z}$. A positive (or negative) semi-definite matrix will give a non-negative (or non-positive respectively) value for the same product. The product of eigenvalues of a positive definite matrix is positive. By this definition, a positive definite matrix is

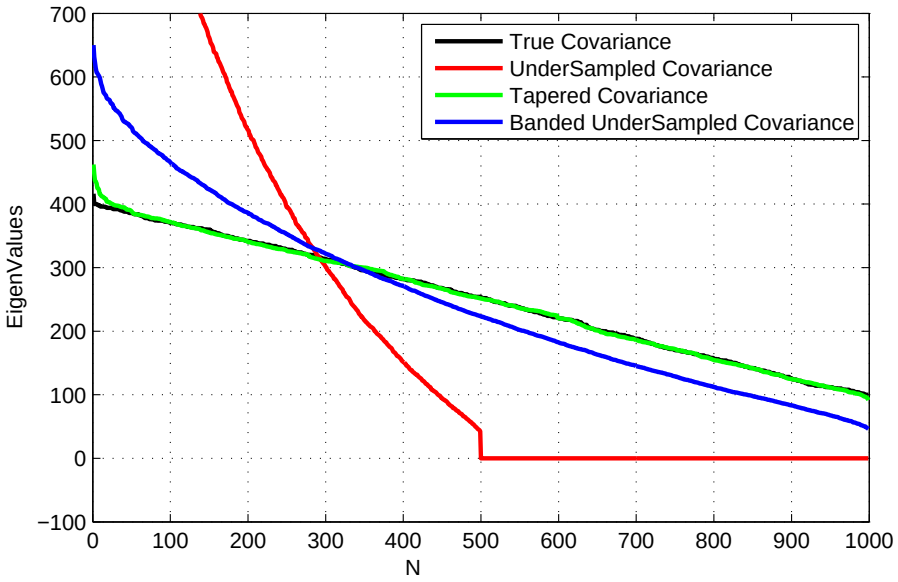


Figure 4.1: Eigenvalues plot showing banding effect on singularity. $K = 500$ and $N = 1000$ and $\Gamma = 0.05 \times N$, where N is the size of the matrix, K is the particle size and Γ is the bandwidth of the simple banded technique.

necessarily invertible whilst this cannot be said for the case of a positive semi-definite matrix. A positive semi-definite matrix that satisfies the equality sign in: $\mathbf{z}^T \Sigma_Y \mathbf{z} \geq 0$ for a certain non-zero z (specifically if z is the eigenvector corresponding to the 0 eigenvalue) is non invertible. A symmetric matrix on the other hand is a square matrix, $A_{M \times M}$ in which $[a_{i,j}] = [a_{j,i}]$ for all $i,j = 1,..,M$.

In [18], different methods are used to well condition a sample covariance matrix computed from a data, which does not satisfy $\frac{N}{K} \to \infty$. The simplest method of well conditioning covariance matrices is to set all off diagonals to zero leaving only the main digonal. This assumption is based on the fact that correlation decreases away from the main diagonal. Such methods aims at biasing the sample covariance matrix towards the main diagonal. [18] suggests a form of banding off diagonals according to $|i - j| > \Gamma \to \sigma_{i,j} = 0$, where i,j are the indices of the matrix and Γ is a certain predefined bandwidth, to remove the linear dependency. Linear shrinkage estimators make a convex combination of the sample covariance (4.10) and a regularization matrix that is varied to approach a specified target covariance matrix with the aim to minimize the Frobenius risk [18].

The first of the two algorithm proposed in [18], involves defining a function,$\mathbf{B}_\Gamma^{sim}()$:

$$\mathbf{B}_\Gamma^{sim}(\hat{\Sigma}_Y) \equiv \sigma_{i,j} = 0, \tag{4.20}$$

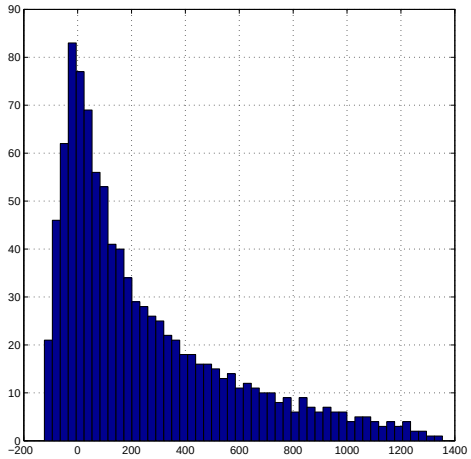


Figure 4.2: Histogram of eigenvalues of simple banded matrix, $\mathbf{B}^{sim}_{\Gamma=0.5\times N}(\Sigma_Y)$.

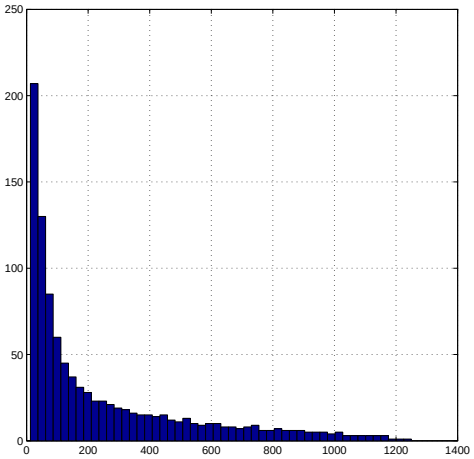


Figure 4.3: Histogram of eigenvalues of tapered matrix, $\mathbf{B}^{tap}_{\Gamma=N}(\Sigma_Y)$.

$$for\ j < i - \Gamma\ and\ j > i + \Gamma, \forall\ \{i,j\}.$$

The bandwidth determine the sparsity of the matrix. This has an advantage of not only taking away the singularity property but it also saves memory space (matrix can be stored as a sparse matrix) with a trade-off of losing some covariance information.

Recall that a covariance matrix is always symmetric, $\Sigma_Y^T = \Sigma_Y$, and positive semi-definite, $\mathbf{z}\Sigma_Y\mathbf{z} \geq 0$ for all $\mathbf{z}$.

Example1: $K = 500$ particles are drawn from a multivariate Gaussian variable $\mathbf{y} \in R^{N=1000}$ with a narrow covariance matrix bandwidth. The estimated covariance is a sample covariance according to 4.10 (brown in Fig. 4.1) and is banded according to (4.20) for $\Gamma = 0.05 \times N$, the symmetric property is preserved and the determinant of Σ_Y $|\Sigma_Y|, > 0$ (invertible). This is seen in Fig. 4.1 (blue).

However we recall also that all eigenvalues of a positive semi-definite matrix are non-negative. A simple banding of Σ_Y according to (4.20) can result in negative eigenvalues (depending on the bandwidth) as shown in a histogram plot of the eigenvalues of $\hat{\Sigma}_Y$ in Fig. 4.2 when the bandwidth was set at $\Gamma = 0.5 \times N$. The conclusion therefore is that a simple banding of the covariance matrix according to 4.20 does not guarantee positive semi-definiteness (Fig. 4.2) and hence an assured valid covariance matrix, although it may guarantee invertibility (Fig. 4.1). We note that Fig. 4.2 - Fig. 4.3 are histograms. Hence the frequency bar showing at the 0 eigenvalue point does not necessarily mean there exists a zero eigenvalue but that between an interval of say 0 and the histogram bin sizeΔb, there exists a number of eigenvalues.

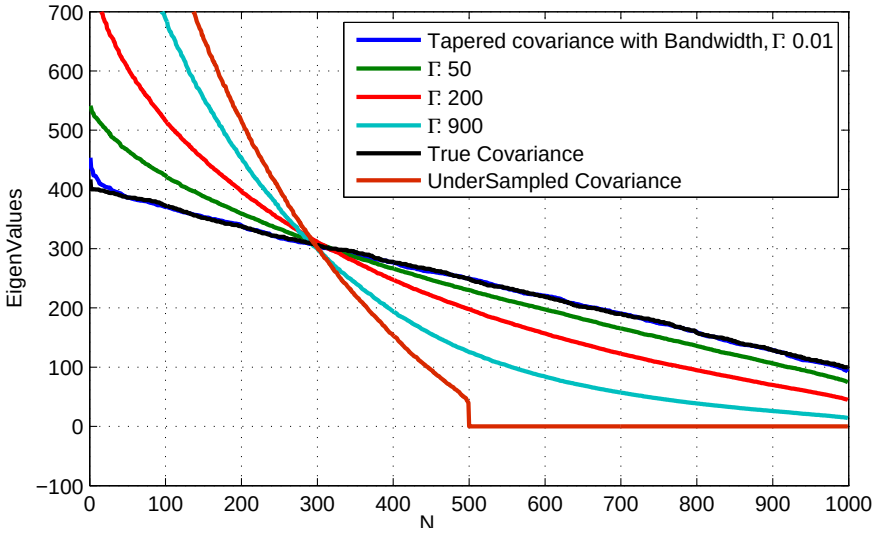


Figure 4.4: Eigenvalues of "tapered" covariance for different Γ.

B. Tapering

A regularization approach proposed in [18], to address the lack of assured positive definiteness in a simple banded covariance matrix is to apply a tapering function. Furrer and Bengtsson [19] have showed that positive definiteness can be preserved by "tapering" the covariance matrix. Since the coordinate-wise product of positive definite matrices is also positive definite, the "tapering" is done by performing a coordinate wise multiplication of the sample covariance with a function R where R is a positive definite matrix. Let us assume that $R = [r_{i,j}]_{i,j \in R^+}$ where $r_{i,j} = g(\delta(i,j))$. If δ is a distance parameter such that, $\delta(i,j) = 0$ when $i = j$, and increases as $\Delta(i-j)$ increases, and also $g(0) = 1$, $g(\infty) = 0$, then $\mathbf{B}_{\Gamma}^{tap}(\Sigma_Y) = \Sigma_Y \bigodot R$ is a regularization of Σ_Y where $\bigodot$ is a coordinate-wise multiplication and Γ is a tapering factor. One of the suggestions in [18] for R is:

$$r_{\Gamma}(i,j) = \exp\left(\frac{-|i-j|}{\Gamma}\right) \tag{4.21}$$

We apply the tapering operation in 4.21 for $\Gamma = 0.01$ to example Example 1 and compare the resulting eigenvalue structure with the simple banded method. The result is shown in Fig. 4.1 for bandwidth, $\Gamma = 0.01,100,1000,100000$. We observe that, all eigenvalues are positive and nonzero. We also observe smaller the value of Γ, the better the estimation. In Fig. 4.3, we show a histogram for the case of a large bandwidth, $\Gamma = N$. Again we observe no eigenvalue is non-positive. Hence tapering an undersampled covariance matrix yields an assured valid symmetric positive-definite covariance matrix (regardless of Γ; except when $\Gamma \to \infty$ in which case the sample covariance matrix is unregularized) whilst the simple banded technique does not always yield a positive semi-definite covariance

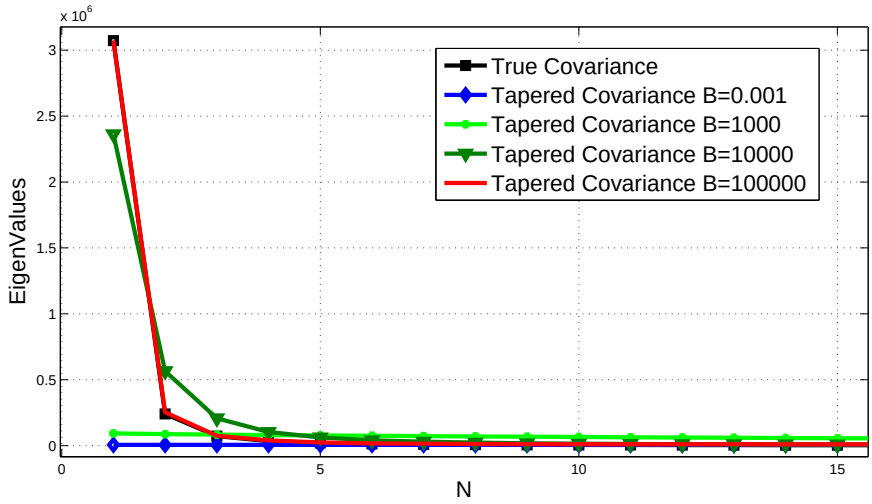


Figure 4.5: Largest 15 eigenvalues of tapered covariances matrices of bandwidth, Γ=0.001,1000,10000,100000.

matrix. We also observe in Fig. 4.4 that the estimation of the eigenvalues (and hence the matrix) of the tapered matrix, approaches the true covariance when the bandwidth is reduced. It is worth mentioning that in cases where the true covariance is a diagonal matrix or close, the estimate of the simple banded operation approaches that of the tapering method if the bandwidth of the simple banded covariance is reduced sufficiently enough, albeit highly unstable and lacks assurance of positive definiteness.

Example 2: In another example, $K = 500$ particles are drawn from a multivariate Gaussian variable $\mathbf{y} \in R^{N=1000}$ with a non-diagonal covariance matrix shown in Fig. 4.6 (unlike in the previous example). Figs. $4.7 - 4.9$ show the structures of the estimated covariance estimates for different Γ whilst Fig. 4.5 shows the estimates of their eigenvalues. We observe in this case that the larger the bandwidth of the tapering function, Γ, the better the estimation since a larger bandwidth will preserve its element's relative values and smoothen (well condition) the structure (Fig. 4.7) whilst a small bandwidth will taper away the off-diagonal correlation elements and yield a poor estimation (Fig. 4.9). Therefore from the Examples 1 and 2, we see that the bandwidth of the tapering function Γ, should be optimized for different covariance matrices.

C. Handling large matrices

Dealing with very large matrices where only a few elements are non-zero is common in many applications. In such cases, memory consumption can be minimized and performance increased by applying a specialized representation of storing only the nonzero elements. Such a matrix is called a sparse matrix. There are several ways to implement

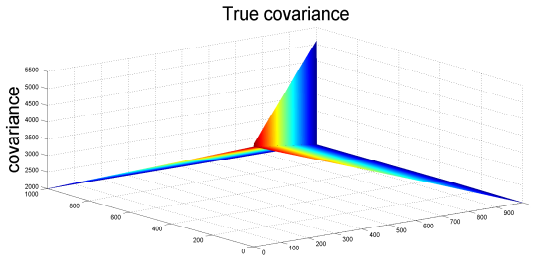


Figure 4.6: True covariance matrix.

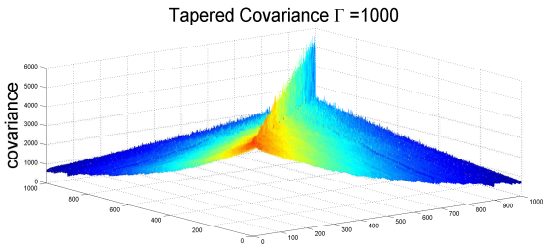


Figure 4.7: Tapered covariance matrix estimate with Γ: 1000.

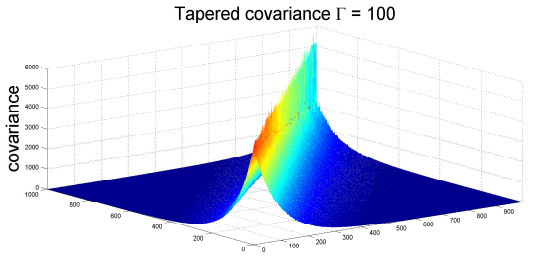


Figure 4.8: Tapered covariance matrix estimate with Γ: 100.

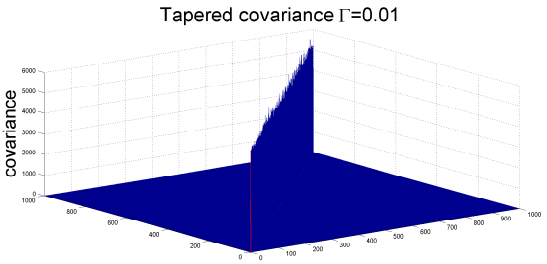


Figure 4.9: Tapered covariance matrix estimate with Γ: 0.001.

the storing and manipulation of a sparse matrix in DSP. Some are best for saving memory whilst others are better suited for computations. In the dictionary of Keys (DOK) for example, only the non-zero elements together with their positions are stored [35]. The required memory for storing such a matrix is given as: $8\times(nnz) + 4\times(nnz + n + 1)$, where n is the number of columns and nnz is the number of non-zero elements.

Another method is called the Yale method in which an $m \times n$ sparse matrix is stored in 3 one-dimensional arrays: A, IA,JA. Where the array A is of length NNZ and holds all the nonzero entries of M in left-to-right top-to-bottom ("row-major") order. The array IA is of length m + 1 and contains the index in A of the first element in each row, followed by the total number of nonzero elements nnz [36]. The third array, JA, contains the column index in M of each element of A and hence is of length nnz. This method is often useful when, $nnz \leq \frac{(m(n-1)-1)}{2}$, else the memory consumed in storing the sparse matrix exceeds the memory size if it had not been stored as a sparse matrix. These methods of saving memory works best for the simple banded method of sample matrix regularization than the tapering method since the band cutting-off of the banded covariance matrix is sharp whereas the tapering method 'tapers off' and hence does not generally lend itself to sparse matrix operations. However since simulations were done in matlab, a method similar to the DOK was used to compress the large covariance matrices, which were sparse.

In computing Σ_Y^{-1} for very large dimensions, various techniques can be used to exploit the banded nature of the matrix to reduce the complexity of the computation. The results of a simulation is shown in Fig. 4.10. We investigate for a very large matrix by de-

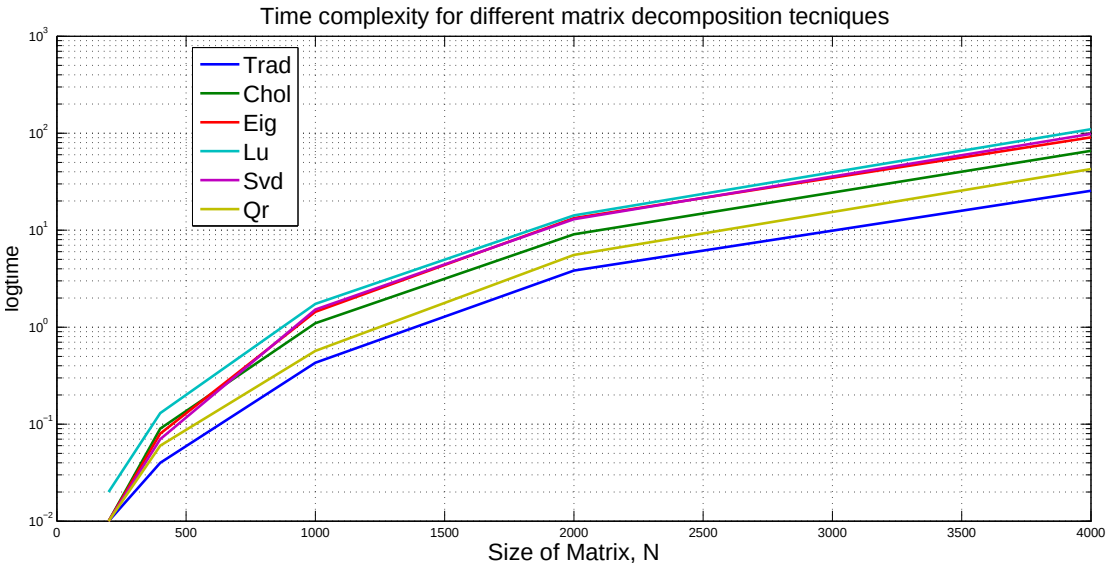


Figure 4.10: Time complexity for LU, QR, Cholesky, Eigenvalue and SVD decomposition.

composing the matrix to find out if it can reduce time complexity of inverse computation complexity as compared to when using traditional methods (Gauss Jordan elimination). Time taken to decompose various sizes of Σ'_Y for different decomposition techniques are shown in Fig. 4.10. The motivation for investigation is as follows: (i) most matrices can be decomposed into submatrices $A = B \times C \times D$ where B and D are orthogonal (ii) it takes lesser computations to transpose than to perform Gauss elimination (for inverse). For example to apply an eigenvalue decomposition to compute the inverse of a large matrix will be as follows:

$$[V\ D] = \mathbf{EIG}(\Sigma_Y) \tag{4.22}$$

$$\Sigma_Y = V^T \times D \times V \tag{4.23}$$

$$\Sigma_Y^{-1} = (V^T \times D \times V)^{-1} \tag{4.24}$$

$$= (V)^{-1} \times D^{-1} \times (V^T)^{-1} \tag{4.25}$$

$$= (V)^T \times D^{-1} \times V \tag{4.26}$$

where V is orthogonal and D is diagonal. Since computing the inverse of a diagonal matrix simply requires finding the inverse of the individual diagonal elements and computing the inverse of an orthogonal matrix is by simply transposing, once the matrix has been decomposed it takes a simple step to compute the inverse. However in the Fig. 4.10 we see that traditional Gauss-Jordan inverse computation gives the minimum time complexity in computing the inverse of a very large covariance matrix (upto 4000×4000). This is because the time it takes to compute the traditional inverse using the Gauss-Jordan elimination technique is almost the time it takes to decompose the matrix and

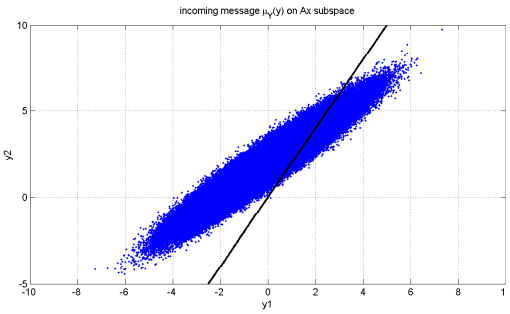


Figure 4.11: Scattered plot representation of PR$\{\overleftarrow{\mu}_Y(\mathbf{y})\} = \{\mathbf{y}^{(k)}\}_{k=1}^{10^4}$, $\mathbf{y}^{(k)} \sim \mathcal{N}$ ($\mathbf{y}$,[0; 2],[2 1.9; 1.9 2]).

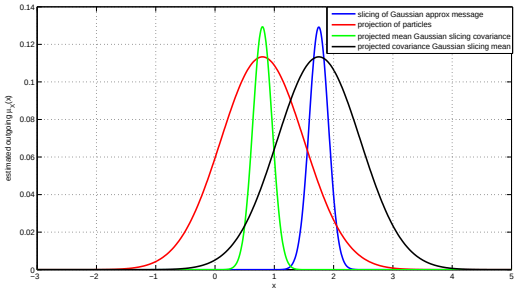


Figure 4.12: Computed outgoing message $\overleftarrow{\mu}_X(x)$, for projection, GMP and reference techniques.

the additional transpose, inverse diagonal and multiplication operations requires extra time hence the relatively longer time.

In the next section we will apply the GMP and the projection operations to simple numerical examples to illustrate the performance of GMP algorithm, and also we show the condition under which both methods are equivalent.

4.1.5 Numerical examples

Injection transformation:

This example illustrates the slicing method (backward Gaussian message passing) compared with the projection method. We show that the GMP (slicing) method gives the true outgoing message and is optimal if the incoming messages are indeed Gaussian and that the projection method is not. We also find that the only scenario when the two algorithms converge is in case when the incoming variable (whose incoming message is observable) is an i.i.d process with a diagonal covariance matrix.
Consider a linear function $\mathbf{y} = Ax$, where A is tall matrix (an injective transformation) and thus $\chi(\mathbf{y}^{(k)}) \leq 1$ and Y is a multivariate Gaussian variable of dimension $\mathbb{R}^2$. We also assume that $|\mu_Y(\mathbf{y})| > 0$ and is represented by $K = 10^4$ particle list with uniform weight as illustrated in the scattered plot 4.11. We simulate two scenarios. In the first case, PR $\{\overleftarrow{\mu}_Y(\mathbf{y})\}$ is a 2D particle list drawn from a Gaussian distribution with mean, $m_X = [\begin{smallmatrix}0\\2\end{smallmatrix}]$ and covariance $\Sigma_Y = [\begin{smallmatrix}2 & 1.9\\1.9 & 2\end{smallmatrix}]$ thus PR$\{\overleftarrow{\mu}_Y(\mathbf{y})\} = \{y^{(k)}\}_{k=1}^{10^4}$ (uniform weights are ignored) drawn from $\mathcal{N}$ ($\mathbf{y}$,$[\begin{smallmatrix}0\\2\end{smallmatrix}]$,$[\begin{smallmatrix}2 & 1.9\\1.9 & 2\end{smallmatrix}]$). The SPA algorithm is applied according to (4.4) to determine the outgoing message, $\overleftarrow{\mu}_X(x)$. The mean and covariance is found according to (4.10). Thus we have:

$$\Sigma_X^{-1} = [1\,2] \times [\begin{smallmatrix}2 & 1.9\\1.9 & 2\end{smallmatrix}]^{(-1)} \times [1\,2]^T \tag{4.27}$$

$$\mathbf{m}_X = (\Sigma_Y^{-1})^{\#} \times [1\,2]^T \times [\begin{smallmatrix}2 & 1.9\\1.9 & 2\end{smallmatrix}]^{(-1)} \times \mathbf{m}_Y \tag{4.28}$$

This gives the result shown in blue in Fig. 4.12. In the same figure, we see the results of projecting the individual particles onto the $\mathbf{y} = Ax$ subspace. This is equivalent to

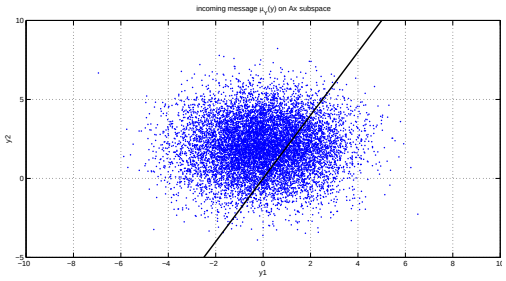


Figure 4.13: Scattered plot representation of $PR\{\overleftarrow{\mu}_Y(\mathbf{y})\} = \{\mathbf{y}^{(k)}\}_{k=1}^{10^4}$, $\mathbf{y}^{(k)} \sim \mathcal{N}(y,[0\,2]^T,[2\,0;0\,2]$.

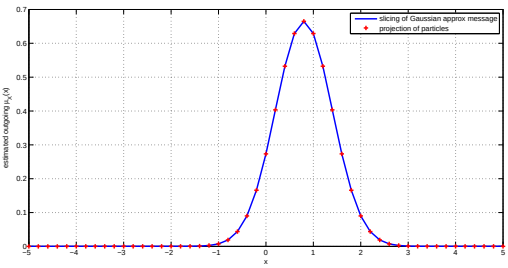


Figure 4.14: Resulting outgoing message $\overleftarrow{\mu}_X(\mathbf{x})$ for projection (red) and GMP (blue).

applying (4.17) with $s = 1$. For each particle, $\mathbf{y}^{(k)}$, there exist exactly one particle, $x^{(k)} = (A^H A)^{\#} A^H \mathbf{y}^{(k)} = [1\,2] \times \frac{1}{5} \times \mathbf{y}^{(k)}$. The result is shown in red as "projection of particles" in the same figure. It clear that, the "sliced Gaussian approximation" method provides the true message (when the number of particles, K is large) whilst in this particular case the projection method does not. In addition to the projection and GMP techniques are two other reference techniques: one reference takes the mean of the projection method and covariance of the GMP method (green), whilst the other takes the mean from the GMP method and covariance from the projection method (black). This reference computation helps us to notice that the correct mean computation is much more important than the correct covariance computation in determining the true outgoing message. Hence the suboptimality of the projection method lies more in the incorrect computation of the mean than the covariance.
In the second instance, $\overleftarrow{\mu}_Y(\mathbf{y}) = \mathcal{N}\left(\mathbf{y},\left[\begin{smallmatrix}0\\2\end{smallmatrix}\right],\left[\begin{smallmatrix}2&0\\0&2\end{smallmatrix}\right]\right)$ (uncorrelated samples of $\mathbf{y}^{(k)} = [y_1^{(k)}\, y_2^{(k)}] \in \mathbb{R}^2$). A scattered plot representation is shown in Fig. 4.13 with the $\mathbf{y} = Ax$ subspace shown in black in the same figure. The slicing (GMP) and projection methods are applied as before and the results are shown on 4.14. Here we observe that the two methods produce the same result in terms of mean and variance. Which means whereas the GMP (slicing) technique considers the correlation between samples to determine the outgoing message $\overleftarrow{\mu}_X(\mathbf{x})$, the projection method determines the outgoing message on the assumption that, samples are uncorrelated thus it merely projects the message particles onto the $A\mathbf{x}$ subspace.

4.2 Application to a coherent optical system

In this simulation, we investigate the performance of the slicing algorithm (GMP) by comparing it to the standard matched filtering and sampling method, which we have identified as a projection technique. We refer to the canonical optical communication model described in Chapter 2. In the receiver, three other different receiver algorithms are used for comparison: DBP, SDBP with MF and a reference technique, which applies the mean of the projection method whilst maintaining the covariance of the GMP. We

Table 4.1: Fiber and EDFA parameters.

	SMF	DCF
D (ps/nm/km)	1.6	-120
$\gamma(1/W/km)$	1.3	5.2
$\alpha(dB/km)$	0.2	0.6
$L_{SMF} + L_{DCF}(km)$	80	

refer to this receiver as the projection-style reference receiver.

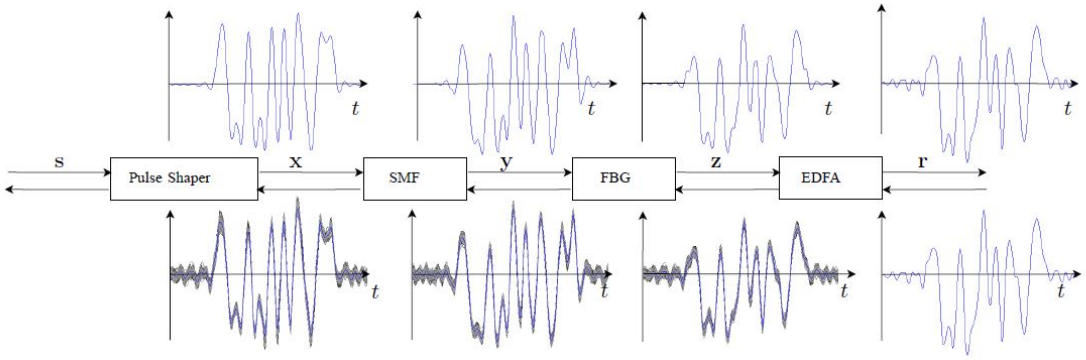


Figure 4.15: SDBP paticle evolution through the fiber [7].

4.2.1 System description

Fig. 4.15 shows the signal evolution from transmission up to back propagation at the pulse shaper output, which corresponds to $\mathrm{PR}\{\overleftarrow{\mu}_{\mathbf{V}}(\mathbf{v}_{1,1})\}$ in section 3.3. The received signal $r(t)$, which has no uncertainty, yet distorted by both linear and nonlinear impairments is back propagated through a virtual channel to yield a signal with a certain level of uncertainty (due to the optical amplifiers; shown by the grey area in Fig. 4.15). Our goal is to compute the correct outgoing message, $\overleftarrow{\mu}_{\mathbf{S}}(\mathbf{s})$ given $\overleftarrow{\mu}_{\mathbf{V}}(\mathbf{v}_{1,1})$.

Transmitter: 16 QAM symbol sequence, $\mathbf{s} \in \mathbb{C}^M$, where $M = 512$, is upsampled with oversampling rate, $q = 4$, and filtered with a root raised cosine pulse with a roll-off factor of 0.25. The output of the pulse shaping is an over-sampled continuous signal, $\mathbf{v}_{1,1} \in \mathbb{C}^N$, where N = 2000, and is fed into a concatenation of fiber spans.

Channel: The experiment is limited to a single polarization transmission. The channel is concatenation of $N_{span} = 60$ SMF-DCF-EDFA units whose parameters are presented in Table 4.1. The SMF introduces power dependent nonlinearities, attenuation

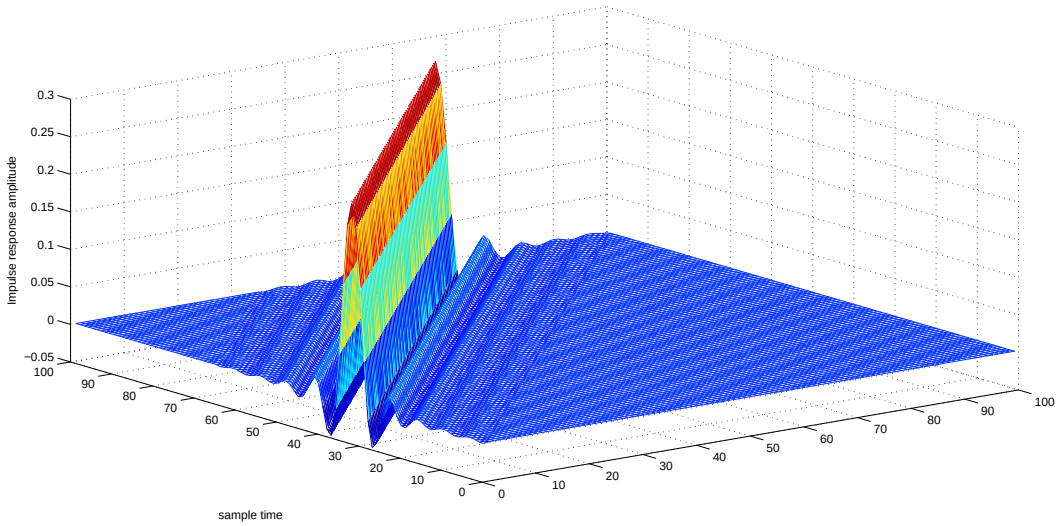


Figure 4.16: Convolution matrix representation of pulse shaper.

and chromatic dispersion. The DCF compensates for the dispersion from the SMF whilst the EDFA reinforces the weak signal from the DCF in the optical domain.

Receiver: In the absence of PMD, the received signal, $r(t) \in \mathbb{C}^N$ is distorted by fiber nonlinear phase shift due to Kerr effect, pulse spreading due to under-compensation by the DCF(s) and noise due to the amplifiers. The aim is to recover the transmitted symbols, $\hat{\mathbf{s}}$, from the received signal $\mathbf{r}$ by computing $p(\mathbf{s}|\mathbf{r})$. Hence with the received signal, $\mathbf{r}(t)$, which contains no uncertainty, we apply SDBP by back propagating the message through the hidden states in the 60 fiber spans, we retrieve $\overleftarrow{\mu}_{\mathbf{Y}}$, corresponding to the message at the output of the pulse shaping (and equivalent to $\mu_{\mathbf{V}_{1,1}}(\mathbf{v}_{1,1})$ in the SDBP algorithm in Chapter 2 section 3.3). $\overleftarrow{\mu}_{Y}$ is represented by a particle list, $\{y^{(k)}\}_{k=1}^{N_p}$ of size, $N_p = 500$. Due to the nature of the algorithms we wish to implement, we rearrange the received signal so as to set the real and imaginary parts of the waveform as shown in (4.29). Assuming $\mathbf{y}$ is an over-sampled complex signal where $y_{real}[n]$ is the real part of the n^{th} symbol, the rearrangement in general is done as follows:

$$\mathbf{y}^{(k)} = [y_{real}[1], y_{imag}[1], ..., y_{real}[N], y_{imag}[N]] \tag{4.29}$$

In this way we can take advantage of the correlation between the real and imaginary parts of the signal for detection. In the MF and sampling detector, $\overleftarrow{\mu}_{\mathbf{S}}$ is computed by matched filtering and downsampling followed by the application of equations: 4.17, which is equivalent to projecting each particle, $\mathbf{y}^{(k)}$, unto the Ax subspace, i.e $(A^TA)^{\#}A^T\mathbf{y}^{(k)}$ and applying (3.11) - (3.12) to determine $p(\mathbf{s}|\mathbf{r})$. In the GMP slicing detector, we apply (4.12) - (4.15) to PR$\{\overleftarrow{\mu}_{\mathbf{V}1,1}\}$ to derive $\overleftarrow{\mu}_{\mathbf{S}}$.

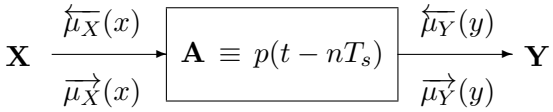


Figure 4.17: Pulse shaping represented as linear function in factor graphs.

In order to apply the GMP function to the canonical coherent optical communication system in Fig. 2.1, the pulse shaping $p(t)$, needs to be represented by a linear transformation function i.e matrix as illustrated in Fig. 4.17. A is a matrix representation of the up-sampling and filtering process hence it is a tall (injective) matrix of real elements that projects an M dimensional space onto an N dimensional space.

Since we apply an orthogonal pulse $p(t)$ at the transmitter to up-sample and filter the sequence of symbols $s[n]$, a convolution matrix, C is constructed from $p(t)$ (C is a circulant matrix, which when pre-multiplied by an arbitrary vector $\mathbf{x}$ is equivalent to filtering the vector with the filter $p(t)$). A 3-D plot of C is shown in Fig. 4.16. In order to account for the up-sample in the pulse shaper, C is up-sampled by a factor of q, which is equivalent to post-multiplying C by B i.e $A'_{N\times M} = C \times B$ where B is an identity matrix with $(q-1) \times M$ zeros inserted between each row.

In the receiver, each (k) particle is re-arranged as shown in (4.29), we therefore need to rearrange $\mathbf{A}$ to correspond to this arrangement. This involves upsampling matrix $\mathbf{A}'$ by a factor of $q = 2$, repeating each column of A next to each other and cyclic shifting the repeated columns downwards by one place. This will yield $\mathbf{A}$ that is used in the GMP algorithm.

4.2.2 Simulation results

Mean and covariance of particle symbols

In the SDBP receiver, after back-propagating the received signal, $r(t)$ according the section 3.3 to yield PR $\{\mu(\mathbf{v}_{1,1})\}$, the mean and covariance of the outgoing message, $\overleftarrow{\mu}_S(\mathbf{s})$ is computed according to (4.17) - (4.19) and (4.15) for the matched filter and the GMP (slicing) receiver respectively and inference is done according to 4.24. In this experiment, an input power of 0 dbm is used and the nonlinear parameters of the SMF and DCF as well as the amplifier gain are shown in Table 4.1. For the GMP algorithm, a bandwidth of 30 is applied. In Figs. 4.18 and 4.19, we see an IQ (16-QAM) plot of the mean outgoing message of the projection technique (left) as well as the GMP (slicing, right) for $M = 512$ symbols. We see that the Gaussian approximation method yields an improved symbol mean scatter plot of the projection method and is better well conditioned.
In Figs. 4.20 and 4.21, the PDF contours of particle representation $\{\mathbf{r}_{MF}^{(k)}[n]\}_{k=1}^{N_p}$ for one symbol is shown for matched filtering and GMP. We use a Gaussian approximation to

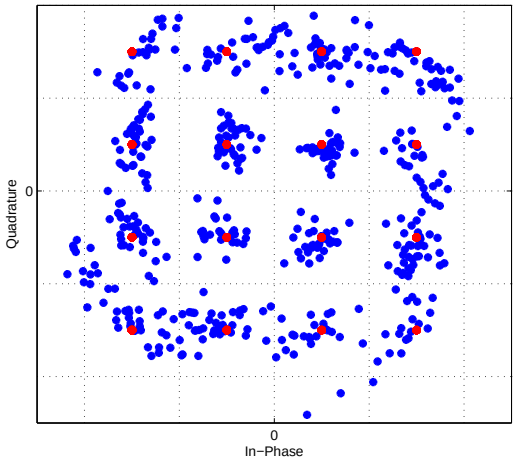


Figure 4.18: Scatterd plot representation of evaluated outgoing mean message (blue) with SDBP with matched filtering. Mean is highly dispersed from the originally transmitted 16-QAM symbols (red).

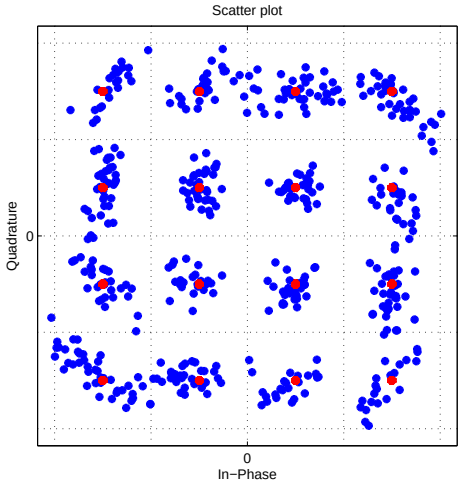


Figure 4.19: Scatterd plot representation of evaluated outgoing mean message (blue) with SDBP with Gaussian message passing. Mean is an optimized version of the matched filtering case 16-QAM.

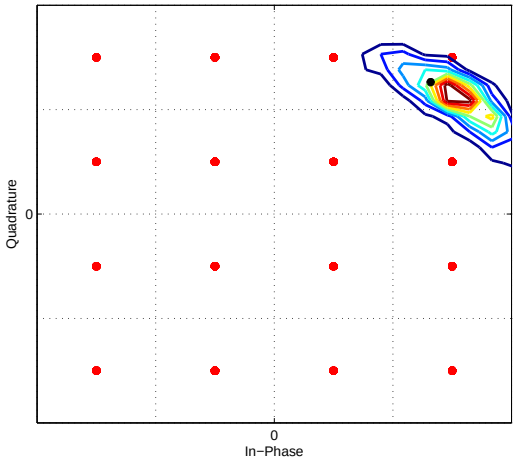


Figure 4.20: Gaussian PDF contours of particle representation for SDBP with matched filtering.

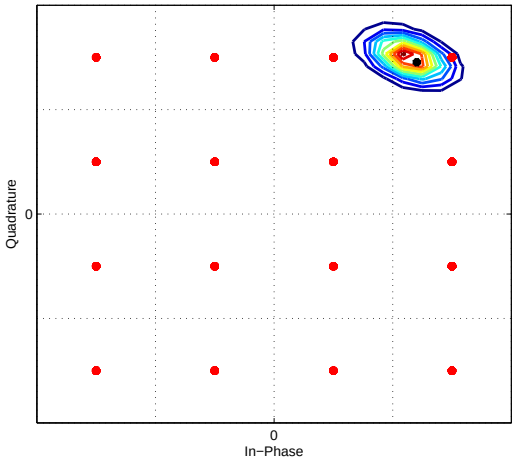


Figure 4.21: Gaussian PDF contours of particle representation for SDBP with Gaussian message passing.

plot the contours for both cases and the black point indicates the mean of the distribution. It is clear that in this instance $\hat{\mathbf{s}}$ in the MAP computation, will be $\hat{\mathbf{s}} = \underset{\mathbf{s}\in\Omega^M}{\operatorname{argmax}}\, p(\mathbf{r}|\mathbf{s})$ $= 3 + j3$ in both algorithms.

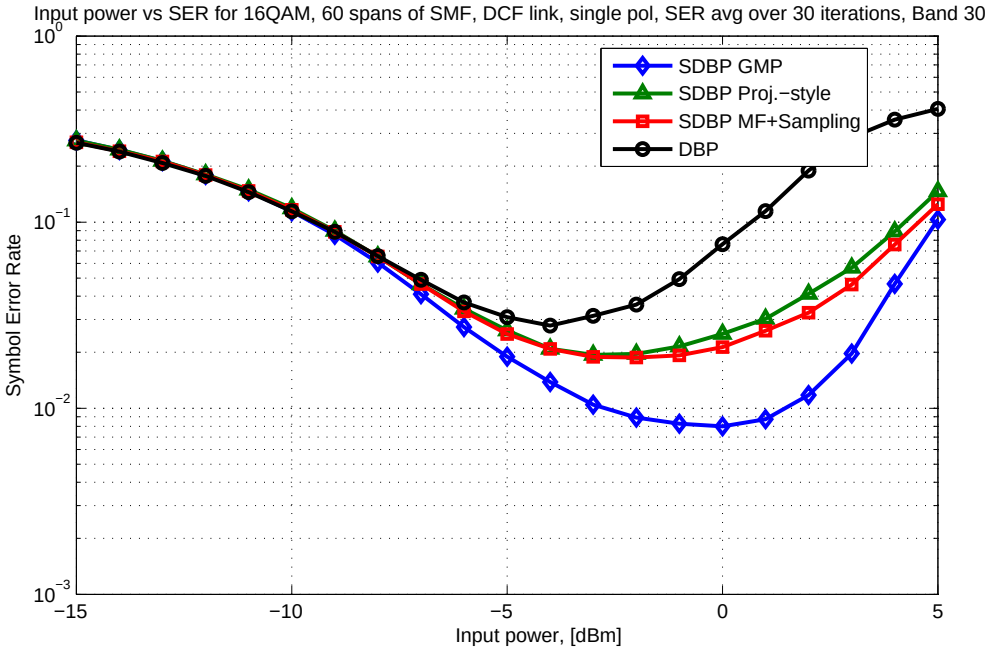


Figure 4.22: Input power versus SER.

SER versus input power

In this experiment, the bandwidth, Γ is kept at 30, the number of particles, N_p is also kept at 500, whilst the input power is varied for the system parameters in Table 4.1. The SER versus input power is shown in Fig. 4.22. We notice significant gains over DBP and SDBP at the nonlinear dominated region for the Gaussian slicing receiver whilst, performance is the same in the noise limited region. We also observe that, the projection-style method works better than the DBP algorithm but not as well as the SDBP matched filtering method. If we consider the 0 dBm input power level, the SER of the DBP is SER_{DBP}= 0.761, the SER of the SDBP and matched filtering is $\text{SER}_{SDBP-MF} = 0.0214$, whilst the SER of the GMP SDBP is $\text{SER}_{GMP-SDBP} = 0.008$. Therefore we say that $\text{SER}_{SDBP+MF} = 2 \times \text{SER}_{GMP-SDBP}$ whilst $\text{SER}_{DBP} = 9 \times \text{SER}_{GMP-SDBP}$.

SER versus bandwidth

First, the bandwidth, Γ, of the covariance matrix was varied with a constant input power of -3dBm and also $N_p = 500$, number of particles kept constant. The results are shown in Fig. 4.23. The left region i.e small bandwidth region whilst the right region represents the large bandwidth region. As the true covariance, Σ_Y contains significant off-diagonal elements (i.e received signal samples are correlated), increasing the bandwidth, Γ of the tapering function, $\mathbf{B}_{\Gamma}^{tap}(\Sigma_Y)$, yields a better estimation of the covariance matrix

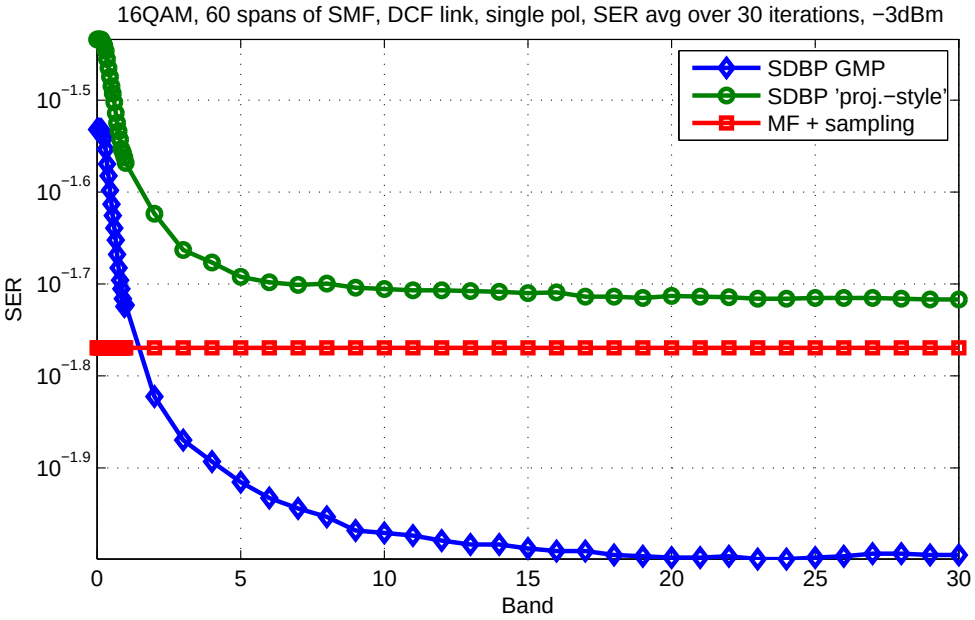


Figure 4.23: Bandwidth versus SER.

and hence improves the SER of the system. In Fig. 4.23, the reference detector whose mean is the mean of the matched filter technique, shows results worst than matched-filter technique. Its performance however improves with increasing bandwidth but never reaches the matched filter's performance. This shows that although the performance of the GMP operation is increases with accuracy of the covariance matrix estimation, performance is improved on a smaller scale compared to the improvement introduced by the correct mean computation. Therefore we conclude that the correct computation of the mean, is the major contributor to the optimality of the GMP receiver.

SER versus number of particles

We further investigate the effect of particle number N_p, on the SER performance. Power is kept at -3dBm whilst the bandwidth, Γ is 30. The DBP is also shown as reference. From Fig. 4.24, we observe that the SER improves with increasing particle number and saturates after about $N_p = 500$. This is true for both projection and slicing. However slicing only improves over matched filter after $N_p = 100$ particles. Since particles are used to estimate the message, $(\sum_{k=1}^{N_p} w_X^{(k)} \delta(x - x^{(k)}) \rightarrow \mu_X(x)$ as $N_p \rightarrow \infty$), increasing the particle number yields a more accurate message (mean and covariance estimation of $\overleftarrow{\mu}_V(v_{1,1})$ discussed under section 4.1.4), hence the improved SER. When the number of particles is less than 100, the condition $\frac{\log(N)}{K} \rightarrow 0$, is hardly approached and the tapering function is even helpless in solving the small sample problem, therefore the

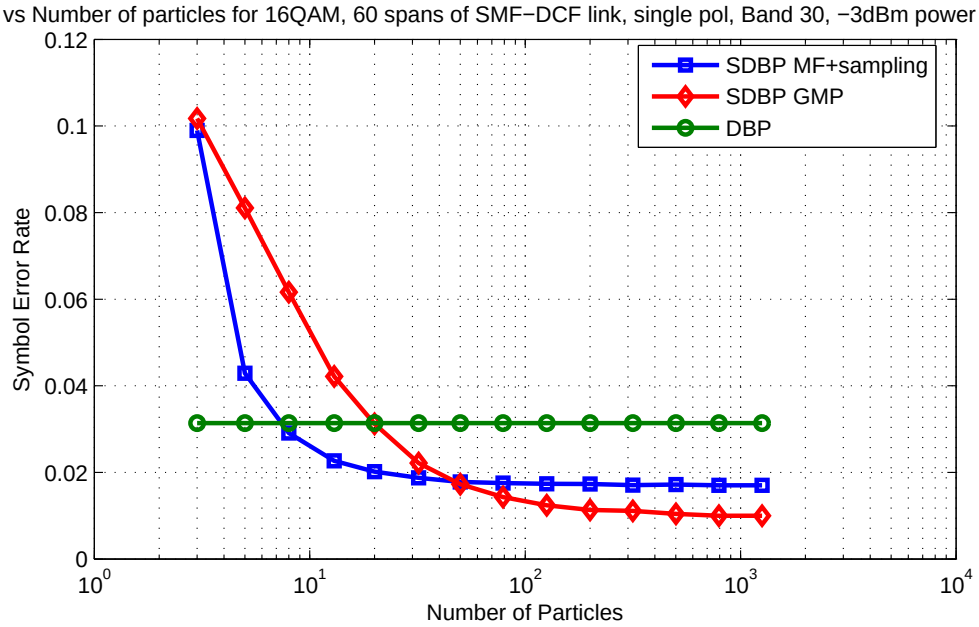


Figure 4.24: Number of particles, N_p versus SER.

Σ estimation is incorrect leading to the high SER in the low particle number region. Similarly in the matched filter, the mean and covariance derived from $\mathbf{r}_{MF}[n]$ is not correct hence the high SER. The DBP algorithm however is not affected by this since it has no stochastic property involved in its development as it does not include the ASE noise in its formulation; it remains unchanged regardless of the particle number.

4.3 Discussion

In the noise-dominated regime in Fig. 4.22, the ultimate limitation of the system is noise and the effects of the nonlinearities are insignificant. Since we can hardly do better than the AWGN limit, all the algorithms perform along that limit. Beyond the noise-limited regime, nonlinearities begin to dominate and the slicing algorithm is ultimately limited by nonlinearities and any non-optimal processing in SDBP. Fig. 4.22 shows that the slicing function performs better than the DBP by several orders and the projection method by about 2 times in the nonlinear region. This improvement was introduced by the correct message computation in the pulse shaping block and hence shows the importance of correct computation of each outgoing message of the hidden blocks in the algorithm.

Apart from bandwidth, Γ and sample number N_p, which can be varied to improve the system performance, the sampling factor, q, can also improve on performance. Increasing this value will increase the degree of freedom and provide more covariance information

with which to compute a more accurate outgoing message however with a trade-off of a some computational complexity.

Although the message estimate is made better by including more particles, this also implies increased computational complexity. A particle size of $N_p = 500$ as used in the simulation is practical not only for complexity reasons, but also system performance (SER) saturates after that value. Typically, in DSP, communication systems will use a greater number of signal samples N, than number of SDBP particles N_p in dsp, in which case a badly scaled matrix results. The tapering function then becomes advantageous.

The main difference between the GMP and the projection is that, GMP computes backward message by considering received signal sample correlations. However, if the correlation is ignored so that the estimated covariance is a scaled diagonal matrix, the slicing method collapses into the projection method. The bandwidth varies the size of correlation samples that is used to compute the backward message. However if the bandwidth is reduced to be sufficiently small, memory consumption of the covariance matrix can be reduced by using the sparse matrix to compress the data at the expense of performance. In Fig. 4.23 reducing the bandwidth to a certain band almost makes the covariance matrix a diagonal matrix, which explains the crossing point between the GMP and the projection methods.

As the bandwidth is increased further, the SER saturates because correlation between the samples slowly reduces after a certain range. Beyond a short range of about $\Gamma = 19$ as seen in Fig. 4.23, Σ_Y makes no further improvement in the message computation. Since $\Gamma \to \infty$ means no regularization of the sample covariance, increasing the bandwidth further towards N implies high SER. Hence bandwidth Γ, needs to be optimized for each covariance matrix.

5

Conclusion and Summary

In this work, we considered the process of message passing in linear and nonlinear models when messages are given in a particle form. We see that, particle message passing is a full member of message passing algorithms and we demonstrate this with the SPA algorithm on factor graphs. We also show why the problem of particle message passing can be difficult for the injective function case. However, the solution is closed in linear models when a Gaussian assumption is made. We called this proposed method, the GMP and considered a special case called projection method. We see that the projection of particles is equivalent to matched filtering with symbol rate down sampling in the SDBP algorithm and is sub-optimal in computing backward messages for cases where the incoming random process is correlated.

We further applied the GMP method in a fiber optic communication receiver to improve the performance of the original SDBP algorithm, which jointly compensates for fiber linear (ASE noise, CD, attenuation) and nonlinear (self phase modulation (SPM)) impairments. We see that, compared to DBP and SDBP with matched filtering, the slicing operation performs better in terms of SER in the nonlinear regime.

Since in this work, we investigated the GMP-SDBP method for only for attenuation, CD and SPM losses, future work can investigate the effects of PMD, and higher order nonlinear impairments such as cross phase modulation (XPM), four wave mixing (FWM), Brillioun and Raman scaterring processes. Also analysis could be made to quantify the gains of the GMP detector over the DBP and SDBP with matched filtering techniques in WDM systems.

Appendices

A

Appendix A

Assume a linear model in the form, $Y = AX$. Where $A \in \mathbb{R}^{M \times N}$. If $X = \{X_1, \cdots, X_M\}^T$ and $Y = \{Y_1, \cdots, Y_N\}^T$ are Gaussian distributions with mean $\mathbf{m}_X$ and $\mathbf{m}_Y$ respectively and covariance $\mathbf{\Sigma}_X$ and $\mathbf{\Sigma}_Y$ respectively. As a linear combination:

$$\begin{aligned} Y_1 &= a_{1,1}X_1 + a_{1,2}X_2 + \cdots + a_{1,M}X_M \\ Y_2 &= a_{2,1}X_1 + a_{2,2}X_2 + \cdots + a_{2,M}X_M \\ &\vdots \\ Y_N &= a_{N,1}X_1 + a_{N,2}X_2 + \cdots + a_{N,M}X_M \end{aligned}$$

$$\text{A}= \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,M-1} & a_{1,M} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,M-1} & a_{2,M} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{N-1,1} & a_{N-1,2} & \cdots & a_{N-1,M-1} & a_{N-1,M} \\ a_{N,1} & a_{N,2} & \cdots & a_{N,M-1} & a_{N,M} \end{pmatrix}$$

Let $E[X] = \mathbf{m}_X$ and $E[Y] = \mathbf{m}_Y$. Then the relation between $\mathbf{m}_X$ and $\mathbf{m}_Y$ is:

$$\mathbf{m}_Y = A\mathbf{m}_X \tag{A.1}$$

$$A^T\mathbf{m}_Y = A^TA\mathbf{m}_X \tag{A.2}$$

$$(A^TA)^{\#}A^T\mathbf{m}_Y = \mathbf{m}_X \tag{A.3}$$

And that of $\mathbf{\Sigma}_X$ and $\mathbf{\Sigma}_Y$:

$$\mathbf{\Sigma}_Y = A\mathbf{\Sigma}_X A^T \tag{A.4}$$

$$A^T \Sigma_Y A = A^T A \mathbf{\Sigma}_X A^T A \tag{A.5}$$

$$(A^T A)^{\#} A^T \mathbf{\Sigma}_Y A (A^T A)^{\#} = \mathbf{\Sigma}_X \tag{A.6}$$

Bibliography

[1] E. Dahlman, G. Mildh, J. Peisa, J. Sachs, Y. Selen, and S. Parkvall, "5G radio access," Ericsson Review, pp. 1-7, June 2014.

[2] Cisco (2015, October 29). VNI forecast highlights [Online]. Available:

`http://www.cisco.com/assets/sol/sp/vni/forecast_highlights_mobile/index.html.`

[3] Shivi Chaturvedi, "The role of digital signal processors (DSP) for 3G mobile communication systems," International Journal on Emerging Technologies vol 1, no. 1, pp. 23-26, 2010.

[4] Govind P. Agrawal, "Fiber-optic communications," John Wiley and Sons, Inc., publication, 2002.

[5] William E.Ryan and Shu Lin, "Channel codes, classical and modern," Cambridge Uni. press, 2009.

[6] H. Wymeersch, N. V. Irukulapati, Isaac A. Sackey, P. Johannisson, E. Agrell, "Backward particle message passing," Signal Processing Advances in Wireless Communications (SPAWC), pp. 450 - 454 , June 2015.

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

[8] N. V. Irukulapati, D. Marsella, P. Johannisson, E. Agrell, M. Secondini, H. Wymeersch, "Stochastic digital backpropagation with residual memory compensation," Journal of Lightwave Technology, vol. 33, no. 24, pp. 1-7, 2015.

[9] D. J. Richardson, J. M. Fini L. E. Nelson. "Space-division multiplexing in optical fibres," Nature Photonics 7, pp. 354–362, 29 April 2013

[10] Ivan P. Kaminow, Tingye Li Alan E. Willner, "Optical fiber telecommunications V B," Systems and Networks, Academic Press, 2008.

[11] Ezra Ip and Joseph M. Kahn, "Impairments using digital backpropagation," Journal of Lightwave Technology, vol. 26, no. 20, pp. 3416-3425, 2008.

[12] 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," Optics Express, vol. 19, no. 23, pp. 22600-22606, 2011.

[13] A. Yariv, D.Fekete, and D.M.Pepper, "Compensation for channel dispersion by nonlinear optical phase conjugation," Optics Letters, vol. 4, no. 2, pp.52-54, Feb. 1979.

[14] K. Peddanarappagari and M. Brandt-Pearce, "Voltera series transfer function of single-mode fibres," Journal of Lightwave Technology, vol. 15, no.12, pp. 2232-2241, 1997.

[15] R. Buczynski, "Photonic crystal fibers," Acta Physica Polonica, vol. 106, no. 2, 2004.

[16] D. Marsella, M. Secondini, E. Forestieri, and R. Magri, "Detection strategies in the presence of fibre nonlinear effects," Optical Communications (ECOC), 38th European Conference and Exhibition, pp. 1-3, 2012.

[17] H. B. Song and M. Brandt-Pearce, "A 2-D discrete-time model of physical impairments in wavelength-division multiplexing systems," Journal of Lightwave Technology, vol. 30, no. 5, pp.713-726, Mar. 2012.

[18] P. J. Bickel and E. Levina, "Regularized estimation of large covariance matrices," The Annals of Statistics, vol. 36, no. 1, pp.199–227, 2008.

[19] Furrer, R. and Bengtsson, T., "Estimation of high-dimensional prior and posteriori covariance matrices in Kalman filter variants," J. Multivariate Anal., vol. 98, no.2, pp.227-255, 2006.

[20] 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.

[21] Tajima K, Zhou J, Nakajima K, Sato K, "Ultralow loss and long length photonic crystal fiber," Journal of Lightwave Technology vol 22, pp. 7–10, 2003.

[22] S. P. Singh and N. Singh, "Nonlinear effects in optical fibers: Origin, management and application," Progress In Electromagnetics Research, PIER vol. 73, pp. 249–275, 2007.

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

[24] Govind P. Agrawal, "Nonlinear fiber optics," Academic press, 2001.

[25] K. Kikuchi, M. Fukase, and S. Kim, "Electronic post-compensation for nonlinear phase noise in a 1000-km 20-Gb/s optical QPSK transmission system using the homodyne receiver with digital signal processing," Proc. Opt. Fiber Commun. Conf, Paper OTuA2, 2007,

[26] G. Charlet, N. Maaref, J. Renaudier,H.Mardoyan, P. Tran, and S. Bigo, "Transmission of 40 Gb/s QPSK with coherent detection over ultralong distance improved by nonlinearity mitigation," IEEE/LEOS Summer Topical Meetings, 2007.

[27] K.-P. Ho and J. M. Kahn, "Electronic compensation technique to mitigate nonlinear phase noise," J. Lightw. Technol., vol. 22, no. 3, pp. 779–783, Mar. 2004.

[28] C. Herzet, N. Noels, V. Lottici, H. Wymeersch, M. Luise, M. Moeneclaey, and L. Vandendorpe, "Code-aided turbo synchronization," Proceedings of the IEEE, vol. 95, no. 6, pp. 1255–1271, 2007.

[29] G. Colavolpe, A. Barbieri, and G. Caire, "Algorithms for iterative decoding in the presence of strong phase noise," IEEE J. Sel. Areas Commun., vol. 23, no. 9, pp. 1748–1757, 2005.

[30] B. P. Smith and F. R. Kschischang, "Future prospects for FEC in fiberoptic communications," IEEE Journal of Selected Topics in Quantum Electronics, vol. 16, no. 5, pp. 1245–1257, Sep. 2010.

[31] I. B. Djordjevic, L. L. Minkov, and H. G. Batshon, "Mitigation of linear and nonlinear impairments in high-speed optical networks by using LDPC-coded turbo equalization," IEEE J. Sel. Areas Commun., vol. 26, no. 6, pp. 73–83, Aug. 2008.

[32] E.E. Narimanov "The channel capacity of a fiber optics communication system: perturbation theory," Journal of lightwave technology, vol. 20, no. 3, pp.530 -537 2002.

[33] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[34] Pontus Johanisson, "Fiber optical communication (Lecture notes)," 2015, unpublished.

[35] Claus Führer, Jan Erik Solem, Olivier Verdier, Tony Stillfjord, Christian Andersson,"Advanced numerical algorithms with Python/Scipy sparse matrices numerical analysis (Lecture Notes)," 2011, unpublished.

[36] S.C. Eisenstat, M.C.Gursky, M.H.Schultz and A.H. Sherman, "Yale sparse matrix package," International journal for numeric methods in engineering, vol. 18, no. 8, pp. 1145-1151, 1982.