TEL AVIV UNIVERSITY

FACULTY OF ENGINEERING

אוניברסיטת תל-אביב

הפאקולטה להנדסה

THE IBY AND ALADAR FLEISCHMAN FACULTY OF ENGINEERING
THE ZANDMAN-SLANER SCHOOL OF GRADUATE STUDIES
THE SCHOOL OF ELECTRICAL ENGINEERING

Message Passing Algorithms for Phase Noise Tracking Using Tikhonov Mixtures

Thesis submitted toward the degree of
Master of Science in Electrical and Electronic Engineering

by

Shachar Shayovitz

July, 2013

TEL AVIV UNIVERSITY אוניברסיטת תל-אביב
FACULTY OF ENGINEERING הפאקולטה להנדסה

THE IBY AND ALADAR FLEISCHMAN FACULTY OF ENGINEERING
THE ZANDMAN-SLANER SCHOOL OF GRADUATE STUDIES
THE SCHOOL OF ELECTRICAL ENGINEERING

Message Passing Algorithms for Phase Noise Tracking Using Tikhonov Mixtures

Thesis submitted toward the degree of
Master of Science in Electrical and Electronic Engineering

by

Shachar Shayovitz

This research was carried out at the
School of Electrical Engineering,
Tel-Aviv University

Advisor:
Dr. Dan Raphaeli

July, 2013

Acknowledgements

I would like to thank my advisor, Dr. Dan Raphaeli, for his great support. His valuable advice and intuition made me a better researcher and engineer.

Most importantly, I would like to thank my future wife, Zohar for her endless patience and support. I couldn't have done this without you!

Abstract

In this work, a new low complexity iterative algorithm for decoding data transmitted over strong phase noise channels is presented. The algorithm is based on the Sum & Product Algorithm (SPA) with phase noise messages modeled as Tikhonov mixtures.

Since mixture based Bayesian inference such as SPA, creates an exponential increase in mixture order for consecutive messages, mixture reduction is necessary. We propose a low complexity mixture reduction algorithm which finds a reduced order mixture whose dissimilarity metric is mathematically proven to be upper bounded by a given threshold. As a part of the mixture reduction, we provide a new method for optimal clustering, which finds the closest Tikhonov distribution, in Kullback Leibler sense, to any circular distribution. We further show a method for limiting the number of mixture components and further complexity reduction.

We present simulation results and complexity analysis for the proposed algorithm which show better performance than other state of the art low complexity algorithms. The algorithm has shown superior performance in high code rate, strong phase noise, high order constellations and sparse pilot patterns scenarios.

Finally, we show that the Tikhonov mixture approximation of SPA messages is equivalent to the tracking of multiple phase trajectories, or also can be looked as smart multiple phase locked loops (PLL).

Young man, in mathematics you don't understand things. You just get used to them.

John von Neumann

Contents

List of Figures

Chapter 1

Introduction

The subject of this thesis is **Message Passing Algorithms for Phase Noise Tracking Using Tikhonov Mixtures**. The goal is to derive low complexity algorithms for joint decoding and estimation of coded information in phase noise channels.

Many high frequency communication systems operating today employ low cost upconverters or downconverters which create phase noise. Phase noise can severely limit the information rate of a communications system and pose a serious challenge for the detection systems. Moreover, simple solutions for phase noise tracking such as PLL either require low phase noise or otherwise require many pilot symbols which reduce the effective data rate.

In the last decade we have witnessed a significant amount of research done on joint estimation and decoding of phase noise and coded information. The use of LDPC or Turbo decoders, as part of iterative message passing schemes, allows the receiver to operate in low SNR regions while requiring less pilot symbols. In [54], the authors proposed an iterative decoding scheme which is based on the factor graph representation of the joint posterior and allows the design of efficient message passing algorithms which incorporate both the code graph and the channel graph.

In order to perform MAP decoding of the code symbols, the SPA is applied to the factor graph. The SP algorithm is a message passing algorithm which computes the exact marginal for each code symbol, provided there are no cycles in the factor graph. In the case of phase noise channels, the messages related to the phase are continuous, thus recursive computation of messages requires computation of integrals which have no analytical solution and the direct application of this algorithm is not feasible. A possible approximation of MAP detection is to quantize the phase noise and perform an approximated SP. The channel phase takes only a finite number of values L, thus creating a trellis diagram representing the random walk. If we suppose a forward - backward scheduling, the SPA reduces to a BCJR run on this trellis following LDPC decoding. This algorithm (called DP - discrete phase in this work) requires large computational resources (large L) to reach high accuracy, rendering it not practical for some real world applications.

In this contribution, a new approach for approximating the phase noise forward and backward messages using Tikhonov mixtures is proposed. Since SP recursion equations create an exponential increase in the number of mixture components, a mixture reduction

algorithm is needed at each phase message calculation to keep the mixture order small. We have tested a few state of the art clustering algorithms which failed for this task and cannot provide proven accuracy under any optimality criterion. Therefore we have derived a new clustering algorithm. A distinct property of the new algorithm is its ability to provide adaptive mixture order, while keeping a specified accuracy constraint (where the accuracy is the Kullback Leibler (KL) divergence between the original and the clustered pdfs). A proof for the accuracy of this mixture reduction algorithm is also presented in this work. We show that the process of hypothesis expansion followed by clustering is equivalent to a sophisticated tracker which can track most of the multiple hypotheses of possible phase trajectories. Occasionally, the number of hypotheses grows, and more options for phase trajectories emerge. Each such event causes the tracker to create another tracking loop. In other occasions, two trajectories are merged into one. We show, as an approximation, the tracking of each isolated phase trajectory is equivalent to a PLL and a split event is equivalent to a point in time when a phase slip may happen.

In the second part, we use a limited order Tikhonov mixture. This limitation may cause the tracking algorithm to lose tracking of the correct phase trajectory, and is analogous to a cycle slip in PLL. We propose a method to combat these slips with only a slight increase in complexity. The principle operation of the method is that each time some hypothesis is abandoned, we can calculate the probability of being in the correct trajectory and we can use this information wisely in the calculation of the messages. We provide further complexity reduction approaches. One of these approaches is to abandon the clustering altogether, and replace it by component selection algorithm, which maintains the specified accuracy but requires more components in return. Now the complexity of clustering is traded against the complexity of other tasks. Finally, we show simulations results which demonstrate that the proposed scheme's Packet Error Rate (PER) are comparable to the DP algorithm and that the resulting computational complexity is much lower than DP and in fact is comparable to the algorithm proposed in [7].

In this thesis we present the following main results:

- A low complexity joint decoding and estimation algorithm for strong phase noise channels with excellent performance in,

    - High code rates
    - Low pilot frequency
    - High order constellations
    - Strong phase noise

- A new approach for mixture dimension reduction which guarantees the KL divergence between the original mixture and the reduced mixture be upper bounded.

- A novel approach for combating cycle slip events in phase tracking.

- A new theorem in directional statistics for clustering circular mixtures.

- We provide insight to the underlying dynamics of Bayesian phase tracking

Chapter 2

Preliminaries

2.1 Directional Statistics

Directional statistics is a branch of mathematics which studies random variables defined on circles and spheres. For example, the probability of the wind to blow at a certain direction. The *circular* mean and variance of a circular random variable θ, are defined in [29], as

$$\mu_C = \angle \mathbb{E}(e^{j\theta}) \tag{2.1}$$

$$\sigma_C^2 = \mathbb{E}(1 - cos(\theta - \mu_C)) \tag{2.2}$$

One can see that for small angle variations around the *circular* mean, the definition of the *circular* variance coincides with the standard definition of the variance of a random variable defined on the real axis, since $1 - cos(\theta - \mu_C) \approx (\theta - \mu_C)^2$. One of the most commonly used circular distributions is the Tikhonov distribution and is defined as,

$$g(\theta) = \frac{e^{Re[\kappa_g e^{-j(\theta - \mu_g)}]}}{2\pi I_0(\kappa_g)} \tag{2.3}$$

According to (2.1) and (2.2), the *circular* mean and *circular* variance of a Tikhonov distribution are,

$$\mu_C = \mu_g \tag{2.4}$$

$$\sigma_C^2 = 1 - \frac{I_1(\kappa_g)}{I_0(\kappa_g)} \tag{2.5}$$

where $I_0(x)$ and $I_1(x)$ are the modified Bessel function of the first kind of the zero and first order, respectively. An alternative formulation for the Tikhonov pdf uses a single complex parameter $z = \kappa_g e^{j\mu_g}$ residual phase noise in a first order PLL when the input phase noise is constant is the tikhonov distribtion.

2.2 Circular Mean & Variance Matching

In this section we will present a new theorem in directional statistics. The theorem states that the nearest Tikhonov distribution, $g(\theta)$, to any circular distribution, $f(\theta)$ (in a Kullback Liebler (KL) sense), has its circular mean and variance matched to those of the circular distribution . The Kullback Liebler (KL) divergence is a common information theoretic measure of similarity between probability distributions, and is defined as [21],

$$D(f||g) \triangleq \int_0^{2\pi} f(\theta) \log \frac{f(\theta)}{g(\theta)} d\theta \tag{2.6}$$

Definition 1. *We define the operator $g(\theta) = \mathsf{CMVM}[f(\theta)]$ (Circular Mean and Variance Matching), to take a circular pdf - $f(\theta)$ and create a Tikhonov pdf $g(\theta)$ with the same circular mean and variance.*

Theorem 2.2.1. (CMVM): *Let $f(\theta)$ be a circular distribution, then the Tikhonov distribution $g(\theta)$ which minimizes $D(f||g)$ is,*

$$g(\theta) = \mathsf{CMVM}[f(\theta)] \tag{2.7}$$

The proof can be found in appendix A.

2.3 Helpful Results for KL Divergence

We introduce the reader to three results related to the Kullback-Leibler Divergence which will prove helpful in the next sections.

Lemma 2.3.1. *Suppose we have two distributions, $f(\theta)$ and $g(\theta)$,*

$$f(\theta) = \sum_{i=1}^{M} \alpha_i f_i(\theta)$$

$$D_{KL}(\sum_{i=1}^{M} \alpha_i f_i(\theta)||g(\theta)) \leq \sum_{i=1}^{M} \alpha_i D_{KL}(f_i(\theta)||g(\theta)) \tag{2.8}$$

The proof of this bound can be found in [47] and is based on the Jensen inequality.

Lemma 2.3.2. *Suppose we have three distributions, $f(\theta)$,$g(\theta)$ and $h(\theta)$. We define the following mixtures,*

$$f_1(\theta) = \alpha f(\theta) + (1 - \alpha)g(\theta) \tag{2.9}$$

$$f_2(\theta) = \alpha f(\theta) + (1 - \alpha)h(\theta)) \tag{2.10}$$

for $0 \leq \alpha \leq 1$
  Then,

$$D_{KL}(f_1(\theta)||f_2(\theta)) \leq (1 - \alpha)D_{KL}(g(\theta)||h(\theta)) \tag{2.11}$$

5

The proof for this identity can also be found in [47].

Lemma 2.3.3. *Suppose we have two mixtures, $f(\theta)$ and $g(\theta)$, of the same order M,*

$$f(\theta) = \sum_{i=1}^{M} \alpha_i f_i(\theta)$$

and

$$g(\theta) = \sum_{j=1}^{M} \beta_i g_i(\theta)$$

Then the KL divergence between them can be upper bounded by,

$$D_{KL}(f(\theta)||g(\theta)) \le D_{KL}(\alpha||\beta) + \sum_{i=1}^{M} \alpha_i D_{KL}(f_i(\theta)||g_i(\theta)) \tag{2.12}$$

where $D_{KL}(\alpha||\beta)$ is the KL divergence between the probability mass functions defined by all the coefficients α_i and βi. The proof of this bound uses the sum log inequality and can be found in [9].

Chapter 3

System Model

In this chapter we present the system model used throughout this work. We begin by introducing the reader to phase noise channels and the factor graph framework for joint decoding and estimation in such channels. Later, we show the specific system model for this work.

3.1 Phase Noise Channel

Oscillators suffer from imperfections which create random phase and frequency instabilities. These instabilities are manifested as a spectrum of noise around the oscillator's operating frequency. Denote the instantaneous frequency noise in an oscillator as $\Phi(\tau)$, then the phase noise component in the output signal is,

$$\phi(t) = 2\pi \int_0^t \Phi(\tau)d\tau \tag{3.1}$$

The analog output of a quadrature modulation digital communications system using a noisy oscillator can be written as,

$$s(t) = \Re\{(I(t) + jQ(t))e^{j(2\pi f_c t + \phi(t))}\} \tag{3.2}$$

where $I(t)$ and $Q(t)$ are the in-phase and quadrature components. We write (3.2) as,

$$s(t) = \Re\{[(I(t) + jQ(t))p(t)]e^{j(2\pi f_c t)}\} \tag{3.3}$$

where,

$$p(t) = e^{j\phi(t)} \tag{3.4}$$

and

$$\mathbf{E}\{p(t)p^*(t+\tau)\} = \mathbf{E}\{e^{j(\phi(t)-\phi(t+\tau))}\} \tag{3.5}$$

Thus the spectrum of $s(t)$ is a convolution of the spectrum of $(I(t) + jQ(t))$ and the spectrum of $p(t)$ - $P(f)$. Using the fact that the power spectrum is the fourier transform of the autocorrelation function for a wide sense stationary process, we will compute the spectrum $P(f)$ using its autocorrelation.

Suppose $\Phi(\tau)$ is a Gaussian white process with,

$$\Phi(\tau) \sim \mathcal{N}(0, N_0) \tag{3.6}$$

then using (3.1), we get that (3.5) is the characteristic function of the Gaussian random variable $\phi(t) - \phi(t + \tau)$, whose mean and variance are zero and $N_0|\tau|$ respectively,

$$\mathbf{E}\{p(t)p^*(t+\tau)\} = e^{\frac{-N_0|\tau|}{2}} \tag{3.7}$$

Therefore the spectrum $P(f)$ is,

$$P(f) = \frac{N_0}{\pi^2 N_0^2 + f^2} \tag{3.8}$$

Assuming the phase noise is almost constant during a symbol duration, then after matched filtering and sampling every T_s seconds, we get a discrete time phase noise Wiener process,

$$\theta_k = \phi(kT_s) \tag{3.9}$$

where

$$\theta_k = \theta_{k-1} + \Delta_k \tag{3.10}$$

and Δ_k is a real, i.i.d gaussian sequence with $\Delta_k \sim \mathcal{N}(0, N_0 T_s)$.

The variance of Δ_k was computed using the fact that power spectrum of the discrete time series $e^{j\theta_k}$ - $S_\theta(f)$, satisfies the following relation,

$$S_\theta(2\pi f) = \frac{1}{T_s} P\left(\frac{2\pi f}{T_s}\right) \tag{3.11}$$

which means that

$$S_\theta(2\pi f) = \frac{1}{T_s} \frac{N_0}{\pi^2 N_0^2 + \left(\frac{2\pi f}{T_s}\right)^2} \tag{3.12}$$

thus,

$$S_\theta(f) = \frac{T_s N_0}{\pi^2 N_0^2 T_s^2 + (f)^2} \tag{3.13}$$

and using (3.8), we get that

$$\mathbf{E}\{\Delta_k^2\} = N_0 T_s \tag{3.14}$$

3.1.1   Phase Noise Baseband Model

For the duration of this work, we assume the following system model. A sequence of data bits is encoded using an LDPC code and then mapped to a complex signal constellation $\mathbb{A}$ of size M, resulting in a sequence of complex modulation symbols $\mathbf{c} = (c_0, c_1, ..., c_{K-1})$. This sequence is transmitted over an AWGN channel affected by carrier phase noise. Since
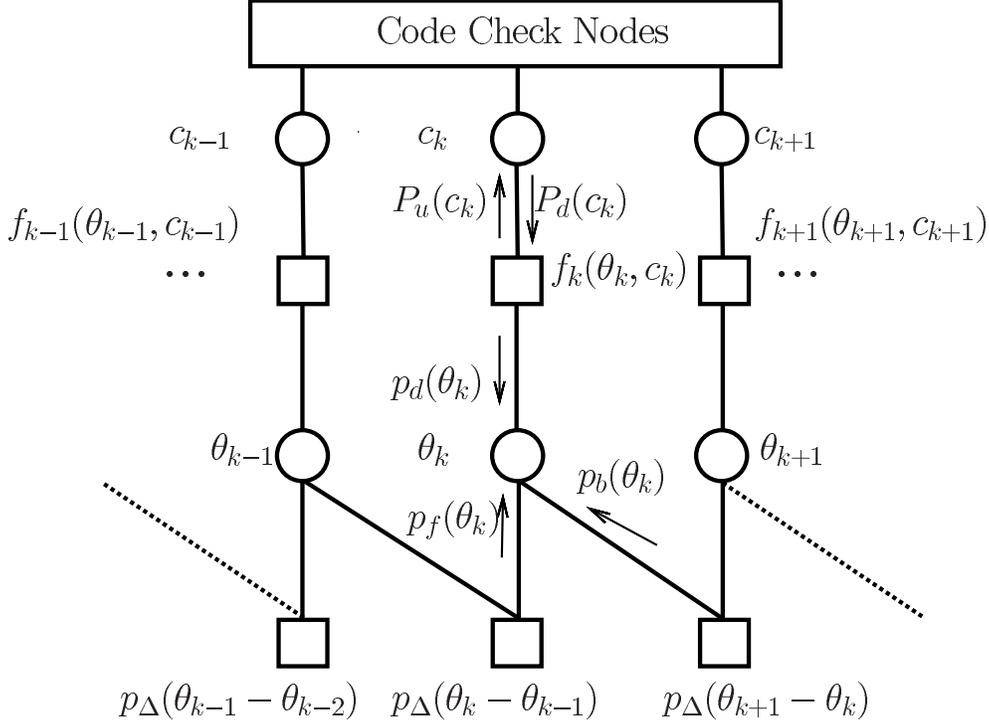
Figure 3.1: Factor graph representation of the joint posterior distribution

we use a long LDPC code, we can assume the symbols are drawn independency from the constellation. The discrete-time baseband complex equivalent channel model at the receiver is given by:

$$r_k = c_k e^{j\theta_k} + n_k \quad k = 0, 1, ..., K - 1. \tag{3.15}$$

where K is the length of the transmitted sequence of complex symbols.

The phase noise stochastic model is a Wiener process

$$\theta_k = \theta_{k-1} + \Delta_k \tag{3.16}$$

where Δ_k is a real, i.i.d gaussian sequence with $\Delta_k \sim \mathcal{N}(0, \sigma_\Delta^2)$ and $\theta_0 \sim \mathcal{U}[0, 2\pi)$. For the sake of clarity we define pilots as transmitted symbols which are known to both the transmitter and receiver and are repeated in the transmitted block every known number of data symbols. We also define a preamble as a sequence of pilots in the beginning of a transmitted block. We assume that the transmitted sequence is padded with pilot symbols in order to bootstrap the algorithms and maintain the tracking.

3.2   Factor Graphs and the Sum Product Algorithm

Since we are interested in optimal MAP detection, we will use the framework defined in [54], compute the SPA equations and thus perform approximate MAP detection. The factor graph representation of the joint posterior distribution was given in [7] and is shown in Fig. 3.1. The resulting Sum & Product messages are computed by

9

$$p_f(\theta_k) = \int_0^{2\pi} p_f(\theta_{k-1})p_d(\theta_{k-1})p_\Delta(\theta_k - \theta_{k-1})d\theta_{k-1} \tag{3.17}$$

$$p_b(\theta_k) = \int_0^{2\pi} p_b(\theta_{k+1})p_d(\theta_{k+1})p_\Delta(\theta_{k+1} - \theta_k)d\theta_{k+1} \tag{3.18}$$

$$p_d(\theta_k) = \sum_{x\in\mathbb{A}} P_d(c_k = x)e_k(c_k, \theta_k) \tag{3.19}$$

$$P_u(c_k) = \int_0^{2\pi} p_f(\theta_k)p_b(\theta_k)e_k(c_k, \theta_k)d\theta_k \tag{3.20}$$

$$e_k(c_k, \theta_k) \propto \exp\{-\frac{|r_k - c_k e^{j\theta_k}|^2}{2\sigma^2}\} \tag{3.21}$$

$$p_\Delta(\theta_k) = \sum_{l=-\infty}^{\infty} g(0, \sigma_\Delta^2, \theta_k - l2\pi) \tag{3.22}$$

Where r_k, P_d, σ^2 and $g(0, \sigma_\Delta^2, \theta)$ are the received base band signal, symbol soft information from LDPC decoder, AWGN variance and Gaussian distribution, respectively. The messages $p_f(\theta_k)$ and $p_b(\theta_k)$ are called in this paper the forward and backward phase noise SP messages, respectively.

The detection process starts with the channel section providing the first LLRs $(P_u(c_k))$ to the LDPC decoder, and so on. A different scheduling could be applied on a general setting, but this will not be possible with the algorithms in this paper. Due to the fact that the phase symbols are continuous random variables, a direct implementation of these equations is not possible and approximations are unavoidable. Assuming enough quantization levels, the DP algorithm can approximate the above equations as close as we wish. However, this algorithm requires large computational resources to reach high accuracy, rendering it not practical for some real world applications. In [48],[50] and [49], modified Tikhonov approximations were used for the messages in the SPA which lead to a very simple and fast algorithm. In this paper, an approximate inference algorithm is proposed which better balances the tradeoff between accuracy and complexity for strong phase noise channels.

Chapter 4

Previous Work

Phase noise has attracted the attention of many researchers. The combination of phase noise and LDPC/Turbo codes is especially interesting due to the fact that these codes perform very well in low SNR regions. In this chapter we will overview previous work done in this area. The solutions can be divided as follows:

- Standard LDPC with pilots

- Special LDPC designs

- Concatenation of LDPC with Rotationally Invariant (RI) Codes

- LDPC with differential encoding

Each of these categories can be further subdivided into a multitude of algorithms in the receiver. In the following sections, we will describe shortly these solutions.

4.1   Standard LDPC with Pilots

In these solutions there is no constraint on the LDPC and in order to resolve phase ambiguity and enable startup of any iterative decoding scheme, pilots are needed. The pilots are usually a single symbol every N symbols. The various algorithms are different with regards to the ability to accommodate large phase noise and complexity. Here is the list of algorithms, divided into groups of common denominator.

4.1.1   Constant Phase Over the Block

These algorithms assume that the phase is unknown but constant over the burst. These algorithms are usually very simple to implement but are not suited for real world applications and the setting assumed in this work. However, we will highlight the algorithms that enable division of the block into several small sub-blocks, and in this case they can be applied to the case of low phase noise. The practical and intuitive idea is using soft information from the code in order to improve the phase estimation and vise versa. In [37], Noels et al. provide a

good theoretical basis to these algorithms. Various versions are described in [43], [30], [36], [17], [27], [55], [28], [26], [59], [10], [58], [46], [5], [16] and [51].

4.1.2 Decision Aided PLL

When dynamic phase is to be tracked, many papers suggest a PLL that tracks the phase within the block, and each iteration does a better job due to better symbol estimations. However, PLL is not suitable for coded systems in high phase noise, since he decoder requires a delay in decisions making, hence updating the PLL with relatively reliable decisions of the decoder requires a delay in the loop. This imposes narrowing the PLL loop filter, causing the PLL to fail in tracking large phase fluctuations. On the other hand, updating the PLL with the tentative zero delayed decisions (i.e. not from the decoder output) will increase the probability of errors in the decision making. Various implementations are described in [40], [11], [2], [53], [38], [56] and [39].

4.1.3 Noncoherent Methods

Even though there is no differential encoding in the case of standard LDPC, noncoherent type of metrics can still be used. These metrics are developed on the basis of the assumption that the phase is constant over a block of N symbols which can be overlapping or not. In [11], Ferrari et al. compare a non-coherent metric to PSP (PLL per state) and the noncoherent is shown to be better by about 1dB from coherent in 5 degrees rms for BPSK. The authors applied this on turbo code but it can be applied on LDPC as well. [41],[42] and [57] assume independent phase per block, so they lose performance unless N is large enough. In [34] Idin and Anastasopoulos present a low complexity non-coherent detection scheme for each block of N symbols, with one pilot. This algorithm works by maximizing GLRT over phase and due to low complexity we can use larger N for small phase noise and reduce its loss. [32] shows excellent results by using noncoherent metrics and proper insertion into factor graphs. [33] uses similar metrics and also gets good results. Using these metrics we can approach coherent detection by using larger N if smaller phase noise is expected. It should be noted that non coherent methods suffer from an intrinsic degradation with respect to the coherent methods and are less flexible when combined with LDPC or turbo codes.

4.1.4 Model based

These methods assume a model for the phase noise posterior and apply Bayesian inference to perform approximated MAP decoding of the code symbols. For example, [7] and [6] use the code as part of an iterative decoding scheme in order to perform joint decoding and estimation. The idea is to create a factor graph representation of the channel with the code and apply the Sum and Product Algorithm (SPA) on this graph to compute the marginal posteriors. However, as seen earlier, the resulting SPA messages are continuous and involve integrals rendering these methods unfeasible. One solution is to quantize the phase messages and perform approximated SPA. This solution requires many quantization levels in order to achieve high accuracy rendering it not practical for real world applications. In [6], Colavolpe presents an algorithm which uses channel memory truncation and a Tikhonov

model for the phase posterior. This approach performs well for low phase noise, but fails for strong phase noise or high modulations since the finite time window does not allow the estimator to effectively track the phase noise variations. In [7], section B., Colavolpe and Barbieri present an algorithm which efficiently balances the tradeoff between accuracy and complexity (called BARB in this paper). BARB uses Tikhonov distribution parameterizations (canonical model) for all the SPA messages concerning a phase node. Canonical models are distributions which are characterized by a finite set of parameters. Thus if the correct canonical model is used to describe the SPA phase messages, then one only needs to compute its finite parameters and propagate them across the factor graph. This approach is the one which was chosen in this thesis.

However, the approximation as defined in [7], is only good when the information from the LDPC decoder is good (high reliability). In the first iteration the approximation is poor, and in fact exists only for pilot symbols. The LLR messages related to the received symbols which are not pilots are essentially zero (no information). This inability to accurately approximate the messages in the first iterations causes many errors and can create an error floor. This problem is intensified when using either low code rate or high code rate. In the first case, the pilots are less significant, since their energy is reduced. In the second case, the poor estimation of the symbols far away from the pilots cannot be overcome by the error correcting capacity of the code. In order to overcome this limitation, BARB relies on the insertion of frequent pilots to the transmitted block causing a reduction of the information rate.

More examples of the model based methods include [35] which uses variational bounding (mean field approximation) coupled with a linearized phase noise model to produce an extended Kalman filter. For low order constellations, this algorithm provides BER levels comparable to the Tikhonov algorithm in [7] with similar complexity but for high phase noise the two schemes perform poorly. [52] and [23] use the phase quantization technique and try to reduce complexity, although it still remains high, by sampling the probabilities using particle filtering which is basically a sequential MCMC method. The Monte Carlo methods perform very well but require very high computational complexity rendering them not suited for our setting. In [45], the author proposed a hard decision directed extended Kalman filter (EKF) to track time varying phase noise for an un-coded system. Moreover, in the same contribution, an iterative receiver algorithm performing code-aided turbo synchronization was derived using the expectation maximization (EM) framework and showed good results. However, since these algorithms assume a linearized model they do not perform well in strong phase noise.

In [24], Lehmann proposed a Gaussian mixture approach to approximate the phase posteriors and derived a mixture reduction algorithm which showed good results for Turbo codes. The mixture reduction algorithm divides the interval $0 - 2\pi$ to N non overlapping sections and clusters all the components in each section to one Gaussian thus producing a mixture of N components. This approach is very problematic since the author does not provide any analysis as to how to choose the number N. This issue can affect the complexity and accuracy of the mixture reduction algorithm a great deal since a large N can be too computational consuming, while a small N would not provide good enough mixture reduction. Moreover, there are implementation issues such as similar components on the edge of two sections and

wrap around. These issues are dealt in this thesis where we came up with the idea of an adaptive mixture order which finds a small reduced mixture satisfying an accuracy criterion coupled with a mathematical proof for this. Moreover, due to the modulo-2π nature of the phase, we propose the Tikhonov distribution which is much more convenient than Gaussian.

4.2 Special LDPC Designs

[18], [20] show a method to design LDPC such that pilots are not needed. They use local check nodes for sub-blocks and then design all other check nodes such that each sub-block is rotationally invariant. This method applies only to BPSK. However, in [19], Sridhar and Cowley show how to apply the technique to higher modulations and show results on phase noise channel and some low complexity receivers.

4.3 Concatenation of LDPC with Rotationally Invariant (RI) Codes

Another attractive code design option is a serial concatenation of an outer code (which can be either LDPC or convolutional code) with inner convolutional trellis code which has the rotational invariance property. This property enables users to design high performance iterative phase noise tracking algorithms independent of the outer code. [13] shows how to design turbo differential (RI) specifically tailored for the phase noise channels, and shows excellent results. RI codes are essentially non coherent codes and as such posses the same intrinsic degradation with respect to coherent methods discussed earlier.

4.4 LDPC with Differential Encoding

Appending LDPC with differential encoding (DE) is a natural way to combat phase noise. DE does not necessarily lead to degradation if proper receiver and code are applied. [14] and [1] show that LDPC with DE is not worse than coherent for MPSK. [44] deals with a class of codes called Repeat accumulate (RA) in which DE is natural. RA codes are good well known codes and are easy to decode. [4] shows that such codes can be very efficient also for MPSK. Moreover, iterative detection of LDPC/DE is treated in [44], [12], [22], [25], [31] and [3] which present low complexity decoding with high performance on high phase noise. However, iterative differential decoding requires many iteration in order to converge thus creating a complexity issue.

4.5 Summary

The methods detailed above have very good performance in specific settings. However, they do not perform well for,

- High order constellations

- Strong phase noise

- High code rate

- Few pilots

In this work, we have developed a low complexity iterative decoding scheme whose performance is very good in the above demanding scenarios. This scheme is based on the SPA algorithm with Tikhonov mixture approximation for the phase noise messages and will be detailed in the next chapter.

Chapter 5

Tikhonov Mixture Canonical Model

In this chapter we will present the Tikhonov mixture canonical model for approximating the forward and backward phase noise SP messages. Firstly, we will give insight to the motivation of using a mixture model for $p_f(\theta_k)$ and $p_b(\theta_k)$. The message, $p_f(\theta_k)$, is the posterior phase distribution given the causal information $(r_0, ..., r_{k-1})$. If we look at the (local) maximum over time we observe a phase trajectory. A phase trajectory is an hypothesis about the phase noise process given the data. In case of zero a priori information, there will be a $\frac{2\pi}{M}$ ambiguity in the phase trajectory, i.e. there will be M parallel phase trajectories with $\frac{2\pi}{M}$ separation between them.

Having a priori information on the data, such as preamble or pilots, can strengthen the correct hypothesis and gradually remove wrong trajectories. However, as we get far away from the known data, more hypotheses emerge. This dynamics is illustrated in Fig. 5.1 where we have plotted in three dimensions the forward phase noise messages $(p_f(\theta_k))$ of the DP algorithm. The DP algorithm computes the phase forward messages (3.17) on a quantized phase space. The axes represent the time sample index, the quantized phase for each symbol and the Z-axis is the posterior probability. In this figure there is only a small preamble in the beginning and the end of the block and thus the first forward messages are single mode Tikhonov distributions, which form a single trajectory in the beginning of the figure and converges to a single trajectory in the end. After the preamble, due to additive noise and phase noise, occasionally the algorithm cannot decide which is the correct phase trajectory due to ambiguity in the symbols, thus it suggests to continue with two trajectories each with its relative probability of occurring. This point is a split in the phase trajectories and is analogous to a cycle slip in a PLL. If we approximate the messages at each point in time as a a Tikhonov mixture with varying order, then each time we have a split, more components are added to the mixture, and each time there is a merge, the number of components decreases. This understating of the underlying structure of the phase messages is one of the most important contributions of this paper and is the basis of the mixture model approach.

The advantage of using mixtures is in the ability to track several phase trajectories simultaneously and provide better extrinsic information to the LDPC decoder, which in turn will provide better information on the code symbols to the phase estimator. In this way the joint detection and estimation will converge quickly and avoid error floors. However, as will be shown in a later section, the approximation of SP messages using mixtures is a
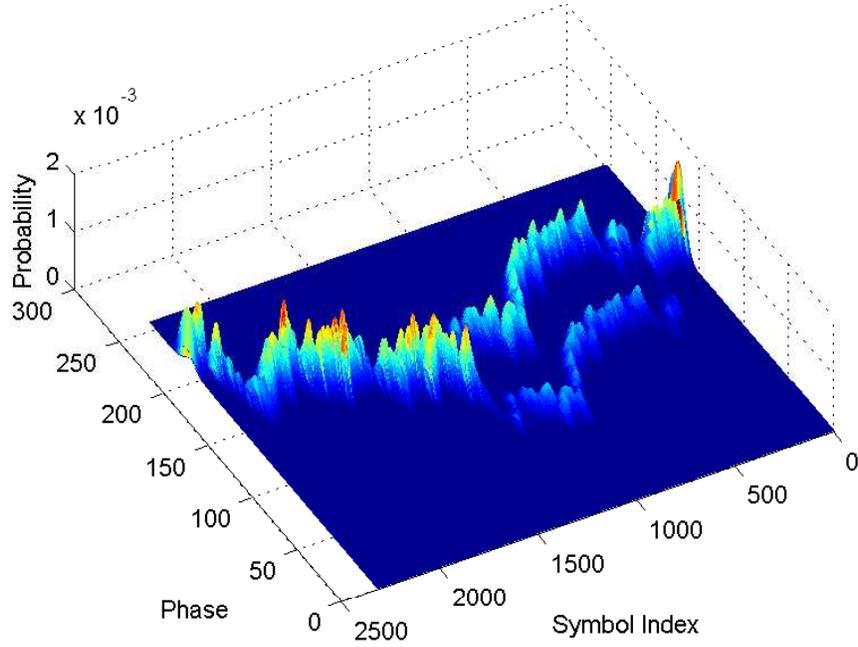
Figure 5.1: SP Phase Noise Forward Messages

very difficult task since the mixture order increases exponentially as we progress the phase tracking along the received block. Therefore, there is a need for an efficient dimension reduction algorithm. In the following sections we will propose a mixture reduction algorithm for the adaptive mixture model. But first we will formulate the mixture reduction task mathematically and describe algorithms which attempt to accomplish this task.

5.1 Mixture Reduction - Problem Formulation

As proposed above, the forward and backward messages are approximated using Tikhonov mixtures,

$$p_f(\theta_k) = \sum_{i=1}^{N_f^k} \alpha_i^{k,f} t_i^{k,f}(\theta_k) \tag{5.1}$$

$$p_b(\theta_k) = \sum_{i=1}^{N_b^k} \alpha_i^{k,b} t_i^{k,b}(\theta_k) \tag{5.2}$$

where:

$$t_i^{k,f}(\theta_k) = \frac{e^{Re[z_i^{k,f} e^{-j\theta_k}]}}{2\pi I_0(|z_i^{k,f}|)} \tag{5.3}$$

17

$$t_i^{k,b}(\theta_k) = \frac{e^{Re[z_i^{k,b}e^{-j\theta_k}]}}{2\pi I_0(|z_i^{k,b}|)} \tag{5.4}$$

and, $\alpha_i^{k,f}$,$\alpha_i^{k,b}$,$z_i^{k,f}$,$z_i^{k,b}$ are the mixture coefficients and Tikhonov parameters of the forward and backward messages of the phase sample. If we insert approximations (5.1) and (5.2) in to the forward and backward recursion equations (3.17) and (3.18) respectively, we get,

$$\tilde{p}_f(\theta_k) = \sum_{i=1}^{N_f^{k-1}} \int_0^{2\pi} \alpha_i^{k-1,f} t_i^{k-1,f}(\theta_{k-1}) p_d(\theta_{k-1}) p_\Delta(\theta_k - \theta_{k-1}) d\theta_{k-1} \tag{5.5}$$

$$\tilde{p}_b(\theta_k) = \sum_{i=1}^{N_b^{k+1}} \int_0^{2\pi} \alpha_i^{k+1,b} t_i^{k+1,b}(\theta_{k+1}) p_d(\theta_{k+1}) p_\Delta(\theta_{k+1} - \theta_k) d\theta_{k+1} \tag{5.6}$$

It is shown in [7] that the convolution of a Tikhonov and a Gaussian distributions is a Tikhonov distribution,

$$\tilde{p}_f(\theta_k) = \sum_{i=1}^{N_f^{k-1}} \sum_{x \in \mathbb{A}} \alpha_i^{k-1,f} \lambda_{i,x}^{k-1,f} \frac{e^{Re[\gamma(\sigma_\Delta, \tilde{Z}_{i,x}^{k-1,f})e^{-j\theta_k}]}}{2\pi I_0(|\gamma(\sigma_\Delta, \tilde{Z}_{i,x}^{k-1,f})|)} \tag{5.7}$$

$$\tilde{p}_b(\theta_k) = \sum_{i=1}^{N_b^{k+1}} \sum_{x \in \mathbb{A}} \alpha_i^{k+1,b} \lambda_{i,x}^{k+1,b} \frac{e^{Re[\gamma(\sigma_\Delta, \tilde{Z}_{i,x}^{k+1,b})e^{-j\theta_k}]}}{2\pi I_0(|\gamma(\sigma_\Delta, \tilde{Z}_{i,x}^{k+1,b})|)} \tag{5.8}$$

where

$$\tilde{Z}_{i,x}^{k-1,f} = z_i^{k-1,f} + \frac{r_{k-1}x^*}{\sigma^2} \tag{5.9}$$

$$\lambda_{i,x}^{k-1,f} = \frac{1}{A} P_d(c_{k-1} = x) \frac{I_0(|\tilde{Z}_{i,x}^{k-1,f}|)}{I_0(|z_i^{k-1,f}|)} \tag{5.10}$$

$$\tilde{Z}_{i,x}^{k+1,b} = z_i^{k+1,b} + \frac{r_{k+1}x^*}{\sigma^2} \tag{5.11}$$

$$\lambda_{i,x}^{k+1,b} = \frac{1}{B} P_d(c_{k+1} = x) \frac{I_0(|\tilde{Z}_{i,x}^{k+1,b}|)}{I_0(|z_i^{k+1,b}|)} \tag{5.12}$$

$$\gamma(\sigma_\Delta, Z) = \frac{Z}{1 + |Z|\sigma_\Delta^2} \tag{5.13}$$

where A and B are a normalizing constants.

Therefore, equations (5.7) and (5.8) are Tikhonov mixtures of order $N_f^k M$ and $N_b^k M$. Since we do not want to increase the mixture order every symbol, a mixture dimension reduction algorithm must be derived which captures "most" of the information in the mixtures $\tilde{p}_f(\theta_k)$ and $\tilde{p}_b(\theta_k)$, while keeping the computational complexity low. From now on, we will present only the forward approximations, but the same applies for the backward.

There are many metrics used for mixture reduction. The two most commonly used are the Integral Squared Error (ISE) and the KL. The ISE metric is defined for mixtures $f(\theta)$ and $g(\theta)$ as follows,

$$D_{ISE}(f(\theta)||g(\theta)) = \int_0^{2\pi} (f(\theta) - g(\theta))^2 d\theta \qquad (5.14)$$

We chose the KL divergence for the cost function between the reduced mixture and the original mixture rather than ISE, since the former is expected to get better results. For example, assume a scenario where there is a low probability isolated cluster of components, then if the reduction algorithm would prune that cluster the ISE based cost will not be effected. However, the KL based reduction will have to assign a cluster since the cost of not approximating it, is very high. In general, the KL divergence does not take in to account the probability of the components while the ISE does. This feature of KL is useful since we wish to track all the significant phase trajectories regardless of their probability.

We define the following mixture reduction task using the Kullback Leibler divergence - *Given a Tikhonov mixture $f(\theta)$ of order L, find a Tikhonov mixture $g(\theta)$ of order N (L > N), which minimizes,*

$$D_{KL}(f(\theta)||g(\theta)) \qquad (5.15)$$

where,

$$f(\theta) = \sum_{i=1}^{L} \alpha_i f_i(\theta) \qquad (5.16)$$

$$g(\theta) = \sum_{j=1}^{N} \beta_j g_j(\theta) \qquad (5.17)$$

where $f(\theta)$ is the mixture $\tilde{p}_f(\theta_k)$ and the reduced order mixture $g(\theta)$ will be the next forward message, $p_f(\theta_k)$. We would like to provide an additional insight to choosing KL. The information theoretic meaning of KL divergence is that we wish that the loss in bits when compressing a source of probability $f(\theta)$, with a code matched to the probability $g(\theta)$ will be not larger than ϵ. Thus, we wish to find a lower order mixture $f(\theta)$ which is a compressed version of $f(\theta)$.

5.2 Mixture Reduction algorithms - Review

There is no analytical solution for (5.15), but there are many mixture reduction algorithms which provide a suboptimal solution for it. They can be generally classified in to two groups, *local* and *global* algorithms. The *global* algorithms attempt to solve (5.15) by gradient descent type solutions which are very computationally demanding. The *local* algorithms usually start from a large mixture and prune out components/merge similar components, according to some rule, until a target mixture order is reached. A very good summary of many of these algorithms can be found in [8]. The *global* algorithms do not deal with KL divergence and thus are not suited for our problem. We will review two *local* algorithms in the following section which provide the best performance in the sense of best balancing the tradeoff between complexity and accuracy, and show why they fail for our case. The first algorithm is the one proposed in [47]. This algorithm minimizes a *local* problem, which *sometimes* provides

a good approximation for (5.15).

Given (5.16), the algorithm finds a pair of mixture components, f_{i^*} and f_{k^*} which satisfy,

$$[i^*, k^*] = arg \min_{i,k} D_{KL}(\alpha f_i + (1-\alpha)f_k || g_j(\theta)) \tag{5.18}$$

where,

$$g_j(\theta) = CMVM(\alpha f_i + (1-\alpha)f_k) \tag{5.19}$$

and α is normalized probability of f_i after dividing by the sum of the probabilities of f_i and f_k. The algorithm merges the two components to $g_j(\theta)$, thus the order of (5.16) has now decreased by one. This procedure is now repeated on the new mixture iteratively to find another optimal pair until the target mixture order is reached. It should be noted that the component's probability influences the metric (5.18). Suppose we have two very different components, one with high probability and another with very low probability, which is the correct hypothesis. Then the algorithm may choose to cluster them, and the low probability component will be lost which may be the correct trajectory. Another algorithm is the one proposed in [15], which also does not directly solve (5.15), but defines another metric which is much easier to handle mathematically. The algorithm's operation is very similar to the K-means algorithm. It first chooses an initial reduced mixture $g(\theta)$ and then iteratively performs the following,

1. *Select the clusters* - Map all f_i to the g_j which minimizes $D_{KL}(f_i || g_j)$

2. *Regroup* - For all j, optimally cluster the elements f_i which were mapped to each g_j to create the new $g(\theta)$

This algorithm is dependent on initial conditions in order to converge to the lowest mixture. Also, the iterative process increases the computational complexity significantly. In [15] and [47], the Gaussian case was considered, thus the clustering was performed using Gaussian moment matching. For our setting, we have taken the liberty to change the moment matching to CMVM, since we have Tikhonov distributions and not Gaussian. Note that in both algorithms, the target order must be defined before operation, since they have to know when to stop. Selecting the proper target mixture order is a difficult task. On one hand, if we choose a large target order, then the complexity will be too high. On the other hand, if we choose the order to be low then the algorithm may cluster components which clearly need not be merged but since they provide the minimal KL divergence, they are clustered. Therefore, in order to maintain a good level of accuracy, the task should be to guarantee an upper bound on the KL divergence and not try to unsuccessfully minimize it. Moreover, it should be noted that in our setting the mixture reduction task (5.15), is performed many times and not once. Therefore, there may not be a need to have the same reduced mixture order for each symbol. These ideas will lead us to the approach presented in the next chapter of the *adaptive* mixture canonical model.

Chapter 6

A New Approach to Mixture Reduction

We have seen that the current state of the art low complexity mixture reduction algorithms based on a fixed target mixture order do not provide good enough approximations to (5.15). Moreover, the choice of the mixture order plays a crucial part in the clustering task. On one hand, a small mixture will provide poor SP message approximation which will propagate over the factor graph and cause a degradation in performance. On the other hand, a large mixture order will demand too many computational resources. Instead of reducing (5.7) and (5.8) to a fixed order, we propose a new approach which has better accuracy while keeping low complexity. Since we are performing Bayesian inference on a large data block, we have many mixture reductions to perform rather than just a single reduction. Therefore, in terms of computational complexity, it is useful to use different mixture orders for different symbols and look at the average number of components as a measure of complexity. This new observation is critical in achieving high accuracy and low PER while keeping computational complexity low. We define the *new mixture reduction task - Given a Tikhonov mixture $f(\theta)$,*

$$f(\theta) = \sum_{i=1}^{L} \alpha_i f_i(\theta) \tag{6.1}$$

Find the Tikhonov mixture $g(\theta)$ with the minimum number of components N

$$g(\theta) = \sum_{j=1}^{N} \beta_j g_j(\theta) \tag{6.2}$$

which satisfy,

$$D_{KL}(f(\theta)||g(\theta)) \leq \epsilon \tag{6.3}$$

Solving this new task will guarantee that the accuracy of the approximation is upper bounded so we can keep the PER levels low. Moreover, simulations show that the resulting mixtures are of very small sizes. In the following section, we will show a low complexity algorithm which finds a mixture $g(\theta)$ whose average number of mixture components is low.

6.1 Mixture Reduction Algorithm

In this section, a mixture reduction algorithm is proposed which is suboptimal in the sense that it does not have the minimal number of components, but finds a low order mixture which satisfies (6.3), for any ϵ. The algorithm, whose details are given in pseudo-code in Algorithm 1, uses the CMVM approach, for optimally merging a Tikhonov mixture to a single Tikhonov distribution.

Algorithm 1 Mixture Reduction Algorithm

$j \leftarrow 1$
while $|f(\theta)| > 0$ **do**
    $lead \leftarrow argmax\{\underline{\alpha}\}$
    for $i = 1 \rightarrow |f(\theta)|$ **do**
        if $D_{KL}(f_i(\theta)||f_{lead}(\theta)) \leq \epsilon$ **then**
            $idx \leftarrow [idx, i]$
        end if
    end for
    $\beta_j \leftarrow \sum_{i \in idx} \alpha_i$
    $g_j(\theta) \leftarrow CMVM(\sum_{i \in idx} \frac{\alpha_i}{\beta_j} f_i(\theta))$
    $f(\theta) \leftarrow f(\theta) - \sum_{i \in idx} \alpha_i f_i(\theta)$
    $j \leftarrow j + 1$
end while

The input to this algorithm, $f(\theta)$, is the Tikhonov mixture (5.7) and the output Tikhonov mixture $g(\theta)$ is a reduced version of $f(\theta)$ and approximates the next forward or backward messages. Note that the function $|f(\theta)|$ outputs the number of Tikhonov components in the Tikhonov mixture $f(\theta)$. The computations of $D_{KL}(f_i(\theta)||f_{lead}(\theta))$ and $CMVM(\sum_{i \in idx} \frac{\alpha_i}{\beta_j} f_i(\theta))$ are detailed in appendices (C) and (B). In the beginning of each iteration, the algorithm selects the highest probability mixture component and clusters it with all the components which are similar to it (KL sense). It then finds the next highest probability component and performs the same until there are no components left to cluster. We will now show that for any ϵ, the algorithm satisfies (6.3).

Theorem 6.1.1. (Mixture Reduction Accuracy): *Let $f(\theta)$ be a Tikhonov mixture of order L and ϵ be a real positive number. Then, applying the Mixture Reduction Algorithm 1 to $f(\theta)$ using ϵ, produces a Tikhonov mixture $g(\theta)$, of order N which satisfies,*

$$D_{KL}(f(\theta)||g(\theta)) \leq \epsilon \tag{6.4}$$

Proof. In the first iteration, the algorithm selects the highest probability mixture component of (6.1) and denotes it as $f_{lead}(\theta)$. Let M_0, be the set of mixture components $f_i(\theta)$ selected for clustering,

$$M_0 = \{f_i(\theta) \mid D_{KL}(f_i(\theta)||f_{lead}(\theta)) \leq \epsilon\} \tag{6.5}$$

and M_1 be the set of mixture components which were not selected,

$$M_1 = \{f_i(\theta) \mid D_{KL}(f_i(\theta)||f_{lead}(\theta)) > \epsilon\} \tag{6.6}$$

Thus,

$$\sum_{i \in M_0} \frac{\alpha_i}{\beta_1} D_{KL}(f_i(\theta) || f_{lead}(\theta)) \leq \epsilon \tag{6.7}$$

where,

$$\beta_1 = \sum_{i \in M_0} \alpha_i \tag{6.8}$$

Using Lemma (2.3.1),

$$D_{KL}\left(\sum_{i \in M_0} \frac{\alpha_i}{\beta_1} f_i(\theta) || f_{lead}(\theta)\right) \leq \epsilon \tag{6.9}$$

The algorithm then clusters all the distributions in M_0 using CMVM,

$$g_1(\theta) = CMVM\left(\sum_{i \in M_0} \frac{\alpha_i}{\beta_1} f_i(\theta)\right) \tag{6.10}$$

then, using Theorem (2.2.1),

$$D_{KL}\left(\sum_{i \in M_0} \frac{\alpha_i}{\beta_1} f_i(\theta) || g_1(\theta)\right) \leq D_{KL}\left(\sum_{i \in M_0} \frac{\alpha_i}{\beta_1} f_i(\theta) || f_{lead}(\theta)\right) \tag{6.11}$$

which means that,

$$D_{KL}\left(\sum_{i \in M_0} \frac{\alpha_i}{\beta_1} f_i(\theta) || g_1(\theta)\right) \leq \epsilon \tag{6.12}$$

We can rewrite the mixtures $f(\theta)$ and $g(\theta)$ in the following way,

$$f(\theta) = \alpha_{M_0} f_{M_0}(\theta) + \alpha_{M_1} f_{M_1}(\theta) \tag{6.13}$$

$$g(\theta) = \beta_1 g_1(\theta) + (1 - \beta_1) h(\theta) \tag{6.14}$$

where,

$$\alpha_{M_0} = \sum_{i \in M_0} \alpha_i \tag{6.15}$$

$$\alpha_{M_1} = \sum_{i \in M_1} \alpha_i \tag{6.16}$$

$$f_{M_i}(\theta) = \sum_{j \in M_i} \frac{\alpha_j}{\alpha_{M_i}} f_j(\theta) \tag{6.17}$$

Using (6.8),

$$\alpha_{M_i} = \beta_i \tag{6.18}$$

Therefore (6.13) and (6.14) are two mixtures of the same size and have exactly the same coefficients, thus the KL of the probability mass functions induced by the coefficients of both mixtures is zero. Using Lemma (2.3.3),

$$D_{KL}(f(\theta) || g(\theta)) \leq \beta_1 D_{KL}(f_{M_0}(\theta) || g_1(\theta)) + (1 - \beta_1) D_{KL}(f_{M_1}(\theta) || h(\theta)) \tag{6.19}$$

using (6.11) we get,

$$D_{KL}(f(\theta)||g(\theta)) \leq \beta_1\epsilon + (1 - \beta_1)D_{KL}(f_{M_1}(\theta)||h(\theta)) \tag{6.20}$$

If we find a Tikhonov mixture $h(\theta)$,which satisfies,

$$D_{KL}(f_{M_1}(\theta)||h(\theta)) \leq \epsilon \tag{6.21}$$

then we will prove the theorem. But (6.21) is exactly the same as the original problem, thus applying the same clustering steps as described earlier on the new mixture $f_{M_1}(\theta)$ will ultimately satisfy,

$$D_{KL}(f(\theta)||g(\theta)) \leq \epsilon \tag{6.22}$$

$\square$

6.2 Mixture Reduction As Phase Noise Tracking

Recall in Fig. 5.1, that the phase noise messages can be viewed as multiple separate phase trajectories, then the mixture reduction algorithm can be viewed as a scheme to map the different mixture components to different phase trajectories. The mixture reduction algorithm receives a mixture describing the next step of all the trajectories and assigns it to a specific trajectory, thus we are able to accurately track all the hypotheses for all the phase trajectories. Assuming slowly varying phase noise and high SNR, the mixture reduction tracking loop i, $\hat{\theta}_k^i$ for each trajectory can be computed in the following manner,

$$\hat{\theta}_k^i = \hat{\theta}_{k-1}^i + \frac{|r_{k-1}||c_t|}{G_{k-1}\sigma^2}(\angle r_{k-1} + \angle c_t - \hat{\theta}_{k-1}^i) \tag{6.23}$$

where, c_t and G_{k-1} are a soft decision of the constellation symbol and the inverse conditional MSE for $\hat{\theta}_{k-1}$, respectively. The proof for this claim is provided in appendix D. Thus the mixture reduction is equivalent to multiple soft decision first order PLLs with adaptive loop gains. Whenever the mixture components of the SPA message become too far apart, a split occurs and automatically the number of tracking loops increases in order to track the new trajectories.

6.3 Limited Order Adaptive Mixture

In the previous section, we have presented an algorithm which *adaptively* changes the canonical model's mixture order, with no upper bound. This enabled us to track *all* the significant phase trajectories in the SP messages. However, there may be scenarios with limited complexity, in which we are forced to have a limited number of mixture components, thus we can track only a limited number of phase trajectories. If the number of significant phase trajectories is larger than the maximum number of mixture components allowed, then we might miss the correct trajectory. For example, if we limit the number of tracked trajectories to one, we get an algorithm very close to a PLL. In this case whenever a split event occurs,

we have to choose one of the trajectories and abandon the other and in case we chose the wrong one, we experience a cycle slip. Analogously we can nickname the event of missing the right trajectory even when more than one trajectory is available, a cycle slip. In this section, assuming pilots are present, we propose an improvement to Algorithm 1, which provides a solution to the missed trajectories problem. The improved algorithm still uses a mixture canonical model for the approximation of messages in the SPA but with an additional variable ϕ_k^f (for backward recursions ϕ_k^b), which approximates, online, the probability that the tracked trajectories include the correct one. This approach enables us to track phase trajectories while maintaining a level of their confidence. We apply the previously used clustering based on the KL divergence in order to select which of the components of the mixture are going to be approximated by a Tikhonov mixture, while the rest of the components will be ignored, but their total probabilities will be accumulated. We then use pilot symbols and ϕ_k^f in order to regain tracking if a cycle slip has occurred. This approach proves to be robust to phase slips and provides a high level of accuracy while keeping a low computational load. The resulting algorithm was shown, in simulations, to provide very good performance in high phase noise level and very close to the performance of the optimal algorithm even for mixtures of order 1,2 and 3.

6.3.1   Modified Reduction Algorithm

We denote the modification of Algorithm 1 for limited complexity, as Algorithm 2. This algorithm selects **some** components from a Tikhonov mixture, $f(\theta)$ and clusters them to an output Tikhonov mixture $g(\theta)$ of maximum order L. We initialize $\phi_0^f = 1$, which means

---

Algorithm 2 Modified Mixture Reduction Algorithm

  $j \leftarrow 1$
  while $j \leq L$ or $|f(\theta)| > 0$ **do**
      $lead \leftarrow argmax\{\underline{\alpha}\}$
      for $i = 1 \rightarrow |f(\theta)|$ **do**
          if $D_{KL}(f_i(\theta)||f_{lead}(\theta)) \leq \epsilon$ **then**
              $idx \leftarrow [idx, i]$
          end if
      end for
      $\beta_j \leftarrow \sum_{i \in idx} \alpha_i$
      $g_j(\theta) \leftarrow CMVM(\sum_{i \in idx} \frac{\alpha_i}{\beta_j} f_i(\theta))$
      $f(\theta) \leftarrow f(\theta) - \sum_{i \in idx} \alpha_i f_i(\theta)$
      $j \leftarrow j + 1$
  end while
  $\phi_k^f \leftarrow (\sum_j \beta_j)\phi_{k-1}^f$

---

that in the first received sample, for the forward recursion, there is no cycle slip. Note that Algorithm 2, is identical to Algorithm 1 apart for the computation of ϕ_k^f. For each iteration, Algorithm 2, selects the most probable component in (5.7) and clusters all the mixture components similar to it. The algorithm then removes this cluster and finds another cluster

similarly. When there are no more components in $f(\theta)$ or the maximum allowed mixture order is reached, the algorithm computes ϕ_k^f. As discussed earlier, this variable represents the probability that a cycle slip has not occurred. The algorithm sums up the probabilities of the clustered components in $f(\theta)$ and multiplies that with ϕ_{k-1}^f to get ϕ_k^f. Suppose we have clustered all the components in $f(\theta)$, then ϕ_{k-1}^f will be equal to ϕ_k^f. That suggests that the probability that a cycle slip has occurred before sample $k-1$ is the same as for sample k. This is in agreement with the fact that no trajectories were ignored at the reduction from $k-1$ to k. For low enough ϵ, ϕ_k^f is a good approximation of that probability.

6.3.2 Recovering From Cycle Slips

In this section, we propose to use ϕ_{k-1}^f, the probability that a cycle has not occurred, and the information conveyed by pilots in order to combat cycle slips. In case of a cycle slip, the phase message estimator based on the tracked trajectories is useless and we need to find a better estimation of the phase message. We propose to estimate the message using **only** the pilot symbol, $p_d(\theta_{k-1})$. However, if a cycle slip has not occurred, then estimating the phase message based **only** on the pilot symbol might damage our tracking. Therefore, once a pilot symbol arrives, we will **average** the two proposed estimators according to ϕ_{k-1}^f,

$$q_f(\theta_{k-1}) = \phi_{k-1}^f p_f(\theta_{k-1}) + (1 - \phi_{k-1}^f)\frac{1}{2\pi} \tag{6.24}$$

If a cycle slip has occurred and ϕ_{k-1}^f is low, then the pilot will, in high probability, correct the tracking. We present the proposed approach in pesudo-code in Algorithm (3).

Algorithm 3 Forward Message Computation with Cycle Slip Recovery

$p_f(\theta_0) \leftarrow \frac{1}{2\pi}$
$\phi_0^f \leftarrow 1$
$k \leftarrow 1$
while $k \leq K$ **do**
    Compute $p_d(\theta_{k-1})$
    if c_{k-1} is a pilot **then**
        $q_f(\theta_{k-1}) \leftarrow \phi_{k-1}^f p_f(\theta_{k-1}) + (1 - \phi_{k-1}^f)\frac{1}{2\pi}$
        $t \leftarrow 1$
    else
        $q_f(\theta_{k-1}) \leftarrow p_f(\theta_{k-1})$
        $t \leftarrow \phi_{k-1}^f$
    end if
    $\tilde{p}_f(\theta_k) \leftarrow \int_0^{2\pi} q_f(\theta_{k-1})p_d(\theta_{k-1})p_\Delta(\theta_k - \theta_{k-1})d\theta_{k-1}$
    $[p_f(\theta_k), \phi_k^f] \leftarrow Algorithm2(\tilde{p}_f(\theta_k), t)$
    $k \leftarrow k + 1$
end while

6.4   Computation of $P_u(c_k)$

As discussed earlier, after computing the forward and backward messages, the next step of the SP algorithm is to compute $P_u(c_k)$. These messages describe the LLR of a code symbol based on the channel part of the factor graph. These messages are sent to the LDPC decoder and the correct approximation of these messages is crucial for the decoding of the LDPC. When using Algorithm 1 for the computation of the forward and backward messages, we use the reduced mixtures with (3.20) and analytically compute the message. However, when using a limited order mixture and Algorithm 2 with the cycle slip recovery method in Algorithm 3, we use ϕ_k^f and ϕ_k^b in order to better the estimation of the messages. Thus $P_u(c_k)$ is a weighted summation of four components which can be interpreted as conditioning on the probability that a phase slip has occurred for each recursion (forward and backward). This will ensure that the computation of $P_u(c_k)$ is based on the most reliable phase posterior estimations, even if a phase slip has occurred in a single recursion (forward or backward). We insert the mixture (6.24) into (3.20),

$$P_u(c_k) \propto \int_0^{2\pi} q_f(\theta_k)q_b(\theta_k)e_k(c_k,\theta_k)d\theta_k \tag{6.25}$$

where $q_f(\theta_k)$ and $q_b(\theta_k)$ are defined in Algorithm 3. We decompose the computation to a summation of four components,

$$P_u(c_k) \propto A + B + C + D \tag{6.26}$$

where

$$A = \phi_k^f \phi_k^b \int_0^{2\pi} p_f(\theta_k)p_b(\theta_k)e_k(c_k,\theta_k)d\theta_k \tag{6.27}$$

$$B = \phi_k^f(1 - \phi_k^b)\int_0^{2\pi} p_f(\theta_k)\frac{1}{2\pi}e_k(c_k,\theta_k)d\theta_k \tag{6.28}$$

$$C = (1 - \phi_k^f)\phi_k^b \int_0^{2\pi} \frac{1}{2\pi}p_b(\theta_k)e_k(c_k,\theta_k)d\theta_k \tag{6.29}$$

$$D = (1 - \phi_f^b)(1 - \phi_k^b)\int_0^{2\pi} \frac{1}{2\pi}\frac{1}{2\pi}e_k(c_k,\theta_k)d\theta_k \tag{6.30}$$

We will detail the computation of A, but the same applies to the other components of (6.26). We use the mixture form defined in (5.1) and (5.2).

We define the following,

$$Z_\psi = z_i^{k,f} + z_j^{k,b} + \frac{r_k c_k^*}{\sigma^2} \tag{6.31}$$

and get,

$$A = \sum_{i=1}^{N_f^k}\sum_{j=1}^{N_b^k} \alpha_i^{k,f}\alpha_j^{k,b}\frac{I_0(|Z_\psi|)}{2\pi I_0(|z_i^{k,f}|)I_0(|z_j^{k,b}|)} \tag{6.32}$$

When implementing the algorithm in log domain, we can simplify (6.32), by using (B.6),

$$\log\left(\frac{I_0(|Z_\psi|)}{2\pi I_0(|z_i^{k,f}|)I_0(|z_j^{k,b}|)}\right) \approx |Z_\psi|-|z_i^{k,f}|-|z_j^{k,b}|-\frac{1}{2}\log\left(\frac{|Z_\psi|}{|z_i^{k,f}||z_j^{k,b}|}\right) \tag{6.33}$$

and for large enough $|z_i^{k,f}|$ and $|z_j^{k,b}|$

$$\log\left(\frac{I_0(|Z_\psi|)}{2\pi I_0(|z_i^{k,f}|)I_0(|z_j^{k,b}|)}\right) \approx |Z_\psi|-|z_i^{k,f}|-|z_j^{k,b}| \tag{6.34}$$

Chapter 7

Complexity

In this section we will detail the computational complexity of the proposed algorithms and compare the complexity to the DP and BARB algorithms. Since the mixture order changes between symbols and LDPC iterations, we can not give an exact expression for the computational complexity. Therefore, in order to assess the complexity of the algorithms, we denote the average number of components in the canonical model per sample, as $\gamma(i)$, where i is the index of the LDPC iteration. $\gamma(i)$, decreases in consecutive LDPC iterations due to the fact that the LDPC decoder provides better soft information on the symbols thus resolving ambiguities and decreasing the required number of components in the mixture. This value, $\gamma(i)$, depends mainly on the number of ambiguities that the phase estimation algorithm suffers. These ambiguities are a function of the SNR, phase noise variance and algorithmic design parameters such as the number of LDPC iteration, KL threshold - ϵ and the pilot pattern.

The significant difference in computational complexity between the DP and the mixture based algorithms stems from the fact that multi modal SPA messages are not well characterized by a single Tikhonov and the DP algorithm must use many quantization levels to accurately describe them. However, the mixture algorithm is successful in characterizing these messages using few mixture parameters and this difference is very significant as the modulation order increases. The mixture algorithm starts out by approximating the forward and backward messages using Tikhonov mixtures. These mixtures are then inserted in to (3.17) and (3.18) to produce larger mixtures (5.7) and (5.8). Next, the mixture reduction scheme produces a reduced mixture which is used to compute $P_u(c_k)$. On average, for a given LDPC iteration i, the forward message, $p_f(\theta_k)$, is a Tikhonov mixture of order $\gamma(i)$. After applying (3.17), the mixture increases to order $M\gamma(i)$ and is sent to the mixture reduction algorithm. Also on average, the clustering algorithm performs $\gamma(i)$ clustering operations on M components. The clustered mixtures are then used to compute $P_u(c_k)$ which is a multiplication of the forward and backward mixtures. In appendices (B) and (C), we have described the computation of the KL divergence, $D_{KL}(f_i(\theta)||f_{lead}(\theta))$ and the application of the CMVM operator on the clustered components - $g_j(\theta) \leftarrow CMVM(\sum_{i \in idx} \frac{\alpha_i}{\beta_j} f_i(\theta))$. In order to further reduce the complexity of the proposed algorithm, the variables representing probabilities are stored in log domain and summation of these variables is approximated using the max operation. We also use the fact that for large x, $\log(I_0(x)) \approx x$ and approximate

the KL divergence in (C.7) as,

$$D_{KL} \approx |z_2|(1 - cos(\angle z_1 - \angle z_2))\qquad(7.1)$$

There is an option to abandon the clustering altogether, and replace it by a component selection algorithm, which maintains the specified accuracy but requires more components in return. Now the complexity of clustering is traded against the complexity of other tasks. The selection algorithm is a simple modification in the algorithm. Instead of using CMVM to cluster several close components, we simply choose $f_{lead}(\theta)$ as the result of the clustering. Recalling (6.11), we note that $f_{lead}(\theta)$ satisfies the accuracy condition and Theorem 6.1.1 still holds. Thus we will not suffer degradation in maximum error if we use this approximation and not CMVM. However, the mean number of mixture components will increase since we do not perform any clustering. The CMVM operator actually reduces the KL divergence between the original mixture and the reduced mixture to much less than ϵ. Therefore, when using CMVM, the reduced mixture is much smaller than needed to satisfy the accuracy condition. In order to get the same performance with the reduced algorithm, we need to decrease ϵ and use more components. The reduced complexity is summarized in Table 7.1, and compared to DP and BARB. Q is the number of quantization levels per constellation symbol in the DP algorithm. We only count multiplication and LUT operations since they are more costly than additions. We assume that the cosine operation is implemented using a look up table.

Table 7.1: Computational load per code symbol per iteration for M-PSK constellation

	DP	BARB	Limited Order
MULS	$4Q^2M^2+2M^2Q+6MQ+M$	$7M+5$	$4M\gamma(i)^2 + 2M(\gamma(i) + 1)$
LUT	QM	$3M$	$3M\gamma(i)^2 - \gamma(i)(2M - 1)$

Chapter 8

Numerical Results

In this section, we analyze the performance of the algorithms proposed in this paper. The performance metrics of a decoding scheme is comprised of two parameters - the Packet/Bit Error Rate (PER/BER) and the computational complexity. We use the DP algorithm as a benchmark for the lowest achievable PER and the algorithm proposed in [7], denoted before as BARB as a benchmark for a state of the art low complexity scheme. The phase noise model used in all the simulations is a Wiener process and the DP algorithm was simulated using 16 quantization levels between two constellation points. Also, note that the simulation results presented in this paper use an MPSK constellation but the algorithm can also be applied, with small changes, to QAM or any other constellation.

In Fig. 8.1 and 8.2, we show the BER and PER results for an 8PSK constellation with an LDPC code of length 4608 with code rate 0.89. We chose $\sigma_\Delta = 0.05$[rads/symbol] and a single pilot was inserted every 20 symbols.

The algorithms simulated were the unlimited order algorithm, the limited order algorithm with varying mixture orders (1,2 and 3) and the reduced complexity algorithm of Order 3 (denoted Reduced Complexity Size 3). We can see that the unlimited mixture, the limited order mixtures of order 2 and 3 and the reduced complexity algorithm provide almost identical results, which are close to the performance of the DP algorithm. On the other hand, the BARB algorithm has significant degradation with respect to all the algorithms. We note that a mixture with only one component can not describe the phase trajectory well enough to have PER levels like DP, but this algorithm is still better than BARB.

In Figs. 8.3,8.4 and 8.5 we show the PER results for a BPSK,QPSK and 32PSK constellations respectively with the same code used earlier. For the BPSK and QPSK scenarios we simulated the phase noise using $\sigma_\Delta = 0.1$[rads/symbol] and for 32PSK we used $\sigma_\Delta = 0.01$[rads/symbol]. A single pilot was inserted according to the pilot frequency detailed in each figure's caption.

We can see that the mixture of order 2 is close to the performance of the optimal algorithm, even when very few pilots are present and the code rate and constellation order are high. One should also observe that for the 32PSK scenario, the BARB algorithm demonstrates a high error floor. This is because of the large phase noise variance and large spacing between pilots which causes the SPA messages to become uniform and thus do not provide information for the LDPC decoder. The high code rate amplifies this problem. However, the limited algorithm with only one Tikhonov component performs almost as well as the DP
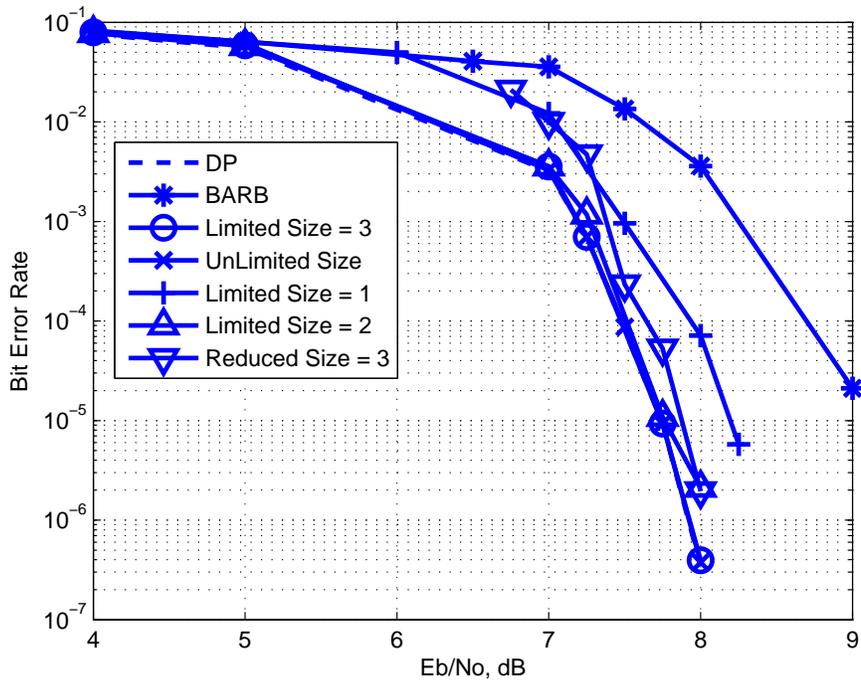
Figure 8.1: Bit Error Rate error rate - 8PSK , $\sigma_\Delta = 0.05$, Pilot Frequency $= 0.05$
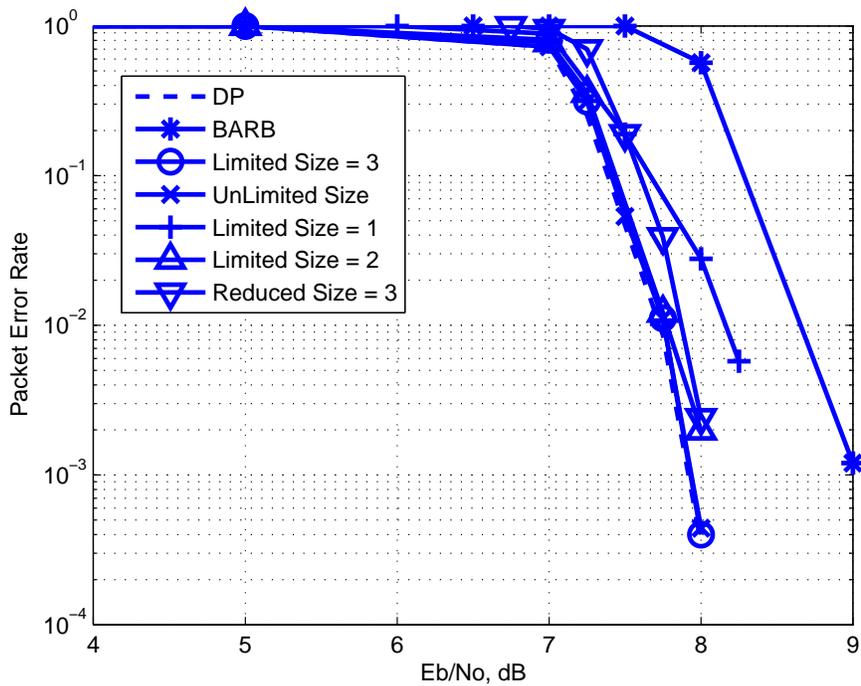
Figure 8.2: Packet Error Rate error rate - 8PSK , $\sigma_\Delta = 0.05$, Pilot Frequency $= 0.05$
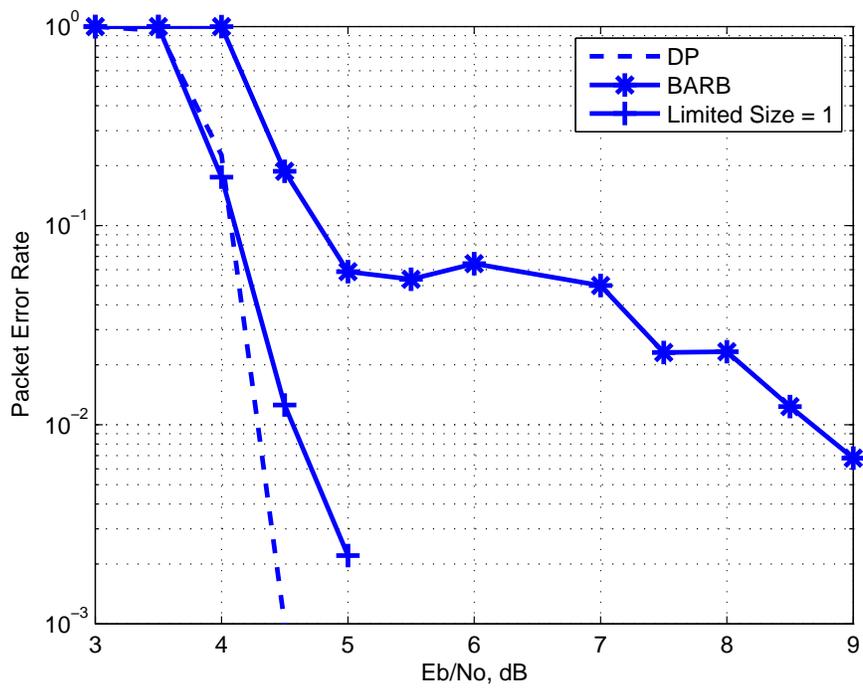
Figure 8.3: Packet Error Rate error rate - BPSK , $\sigma_\Delta = 0.1$, Pilot Frequency $= 0.0125$
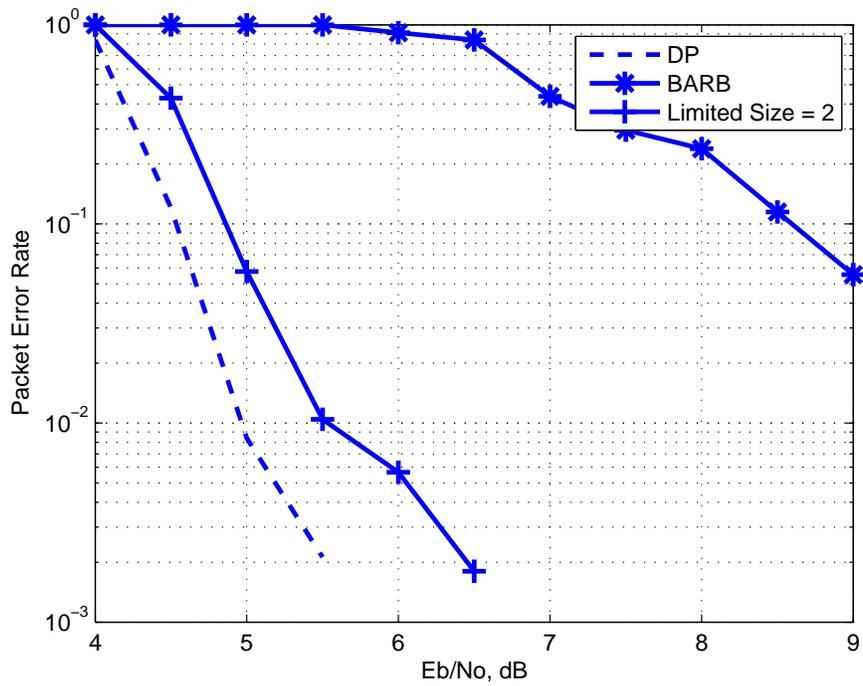
Figure 8.4: Packet Error Rate error rate - QPSK , $\sigma_\Delta = 0.1$, Pilot Frequency $= 0.05$
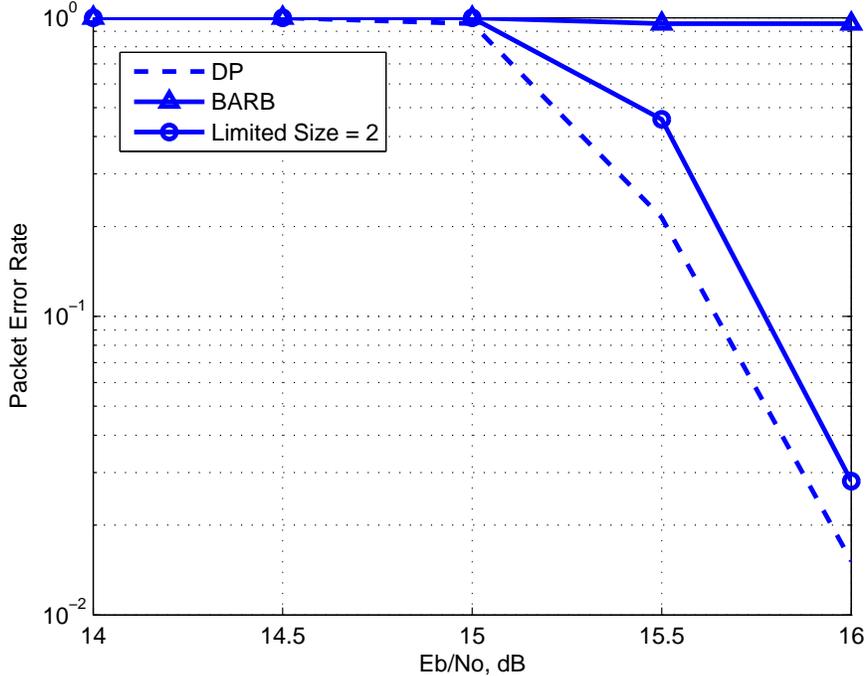
Figure 8.5: Packet error rate - 32PSK, $\sigma_\Delta = 0.01$, Pilot Frequency $= 0.025$

algorithm. This is due to the cycle slip recovery procedure we have presented earlier which enables the limited algorithm to regain tracking even after missing the correct trajectory.

In Fig. 8.6 we present the average number of mixture components, for different SNR and LDPC iterations for $\epsilon = 4$. It can be seen that for the first iteration, many components are needed since there is a high level of phase ambiguity. As the iterations progress the LDPC decoder sends better soft information for the code symbols, resolving these ambiguities. Therefore, the average number of mixture components becomes closer to 1.

In Fig. 8.7 we present the average number of mixture components for the reduced complexity algorithm, for different SNR and LDPC iterations for $\epsilon = 1$. We chose ϵ to be lower since we do not use the CMVM operator as described earlier. As shown in this figure, the mean number of components is larger than for $\epsilon = 4$ but the overall complexity is still manageable. In Table (8.1), the computational complexity of the reduced complexity algorithm is compared to the DP and BARB algorithms. We use the mean mixture in Fig. 8.7 as γ. We can see that the algorithms proposed in this contribution, have extremely less computational complexity than DP, while having comparable PER levels to it.

It should be noted, that the PER performance of the Unlimited algorithm, for small enough ϵ, is as good as the PER performance of the DP algorithm because the mixture algorithm tracks all the significant trajectories with no limit on the mixture order. The choice of the threshold ϵ in the algorithm is according to the level of distortion allowed for the reduced mixture with respect to the original mixture. If ϵ is very close to zero, then there will not be any components close enough and the mixture will not be reduced. Therefore, there is a tradeoff between complexity and accuracy in the selection of this parameter. This

34
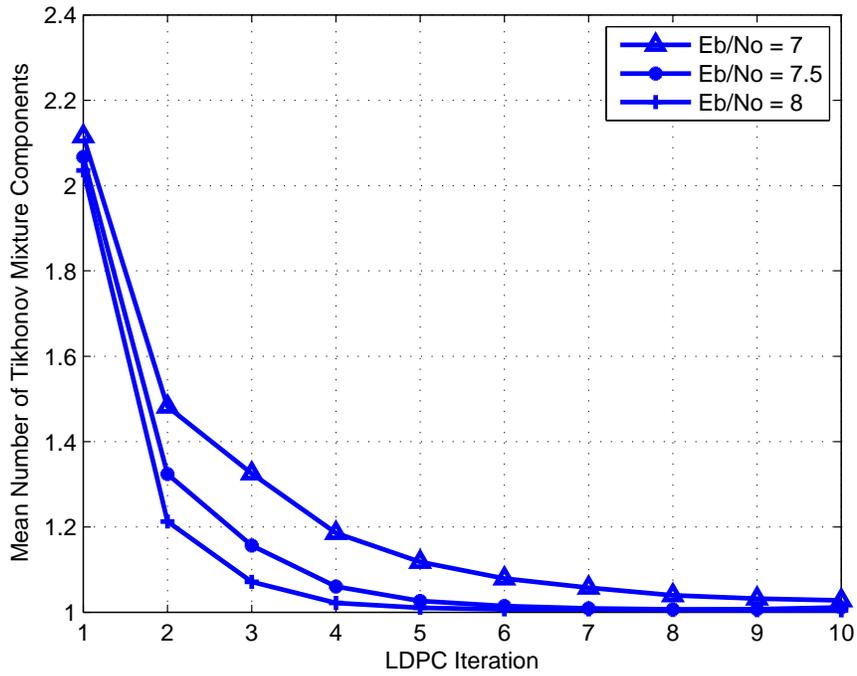
Figure 8.6: 8PSK Mean Number of Tikhonov Mixture Components - Full Algorihtm, Maximum 3 lobes

Figure 8.7: 8PSK Mean Number of Tikhonov Mixture Components - Reduced Complexity Algorithm, Maximum 3lobes

Table 8.1: Simulation Results - Computational load per code symbol for 8PSK constellation at $\frac{E_b}{N_0} = 8dB$

Algorithm	DP	BARB	Reduced Complexity, Order 3
Iteration	Constant for all iterations	Constant	1 2 3 4
MULS	68360	61	312 292 273 238
LUT	128	24	147 134 123 102
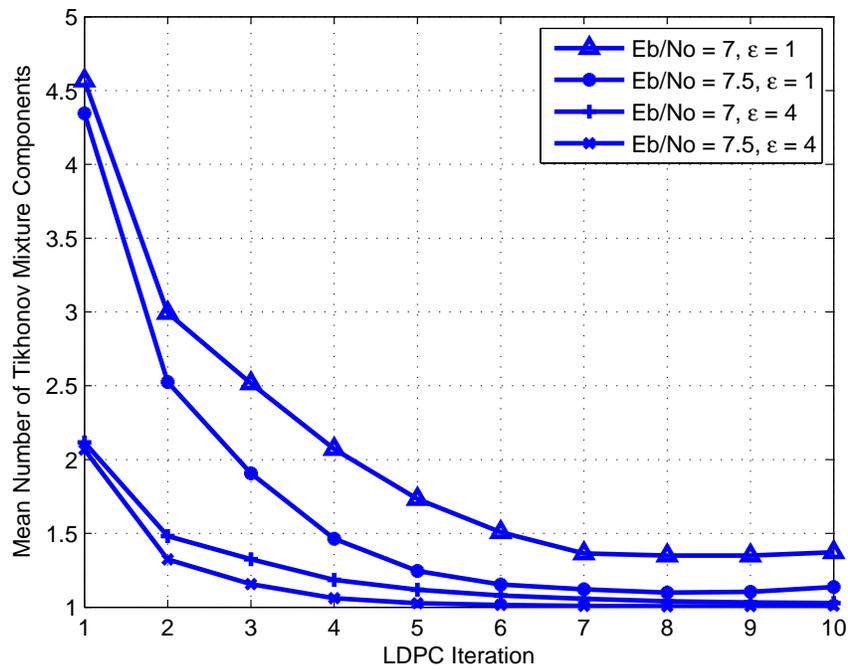
Figure 8.8: 8PSK Mean Number of Tikhonov Mixture Components - Unlimited Algorithm

tradeoff is illustrated in Fig. 8.8, where we have plotted the mean mixture order for the unlimited algorithm using $\epsilon = 1$ and $\epsilon = 4$. It should be noted that for these values and chosen SNRs, the unlimited algorithm has the same PER levels for both ϵ. However, choosing $\epsilon = 15$ with the same algorithm will increase the PER. Therefore, choosing the threshold too low might increase the mixture order with no actual need.

Chapter 9

Summary and Conclusions

In this work we have presented a new low complexity approach for joint decoding and estimation of LDPC coded communications in phase noise channels. The proposed algorithms are based on the approximation of SPA messages using Tikhonov mixture canonical models. We have presented an innovative approach for mixture dimension reduction which does not attempt to find a reduced mixture of fixed order but an adaptive order. Furthermore, we have presented a dimension reduction algorithm which is mathematically proven to provide a reduced order mixture whose KL divergence to the original mixture is upper bounded by an arbitrary threshold.

The decoding schemes proposed in this contribution are shown via simulations to have PER levels very close to the optimal algorithm (DP), even for very low mixture order. We have also introduced several approximations to the mixture algorithms which significantly reduce their complexity. These approximations cause a slight increase in the number of mixture components needed for optimal performance but the resulting mixture order is still very low.

We have shown that the estimation algorithm can be viewed as trajectory tracking, thus enabling the development of the mixture reduction and clustering algorithms which can be viewed as PLLs. It has been shown that the process of hypothesis expansion followed by clustering is equivalent to a sophisticated tracker which can track most of the multiple hypotheses of possible phase trajectories. Occasionally, the number of hypotheses grows, and more options for phase trajectories emerge. Each such event causes the tracker to create another tracking loop. We have shown, as an approximation, that the tracking of each isolated phase trajectory is equivalent to a PLL and a split event is equivalent to a point in time when a phase slip may happen. We have proposed an algorithm to combat these events, which has shown via simulations to perform very well. All this has lead to a better understanding of tracking problems using Bayesian methods.

An interesting topic for future study is to develop a mathematical model for computing the mean number of mixture components for given channel conditions and operating points. This will enable the system designer to pre-allocate memory for the estimation algorithm without taking spare space.

Appendix A

Proof of Theorem 2.2.1

Let $f(\theta)$ be any circular distribution defined on $[0, 2\pi)$ and $g(\theta)$ a Tikhonov distribution.

$$g(\theta) = \frac{e^{Re[\kappa e^{-j(\theta - \mu)}]}}{2\pi I_0(\kappa)} \tag{A.1}$$

We wish to find,

$$[\mu^*, \kappa^*] = arg \min_{\mu, \kappa} D_{KL}(f||g) \tag{A.2}$$

According to the definition of the KL divergence,

$$D_{KL}(f||g) = -h(f) - \int_0^{2\pi} f(\theta) \log g(\theta) d\theta \tag{A.3}$$

where the differential entropy of the circular distribution $f(\theta)$, $h(f)$ does not affect the optimization,

$$[\mu^*, \kappa^*] = arg \max_{\mu, \kappa} \int_0^{2\pi} f(\theta) \log g(\theta) d\theta \tag{A.4}$$

After the insertion of the Tikhonov form into (A.4), we get

$$[\mu^*, \kappa^*] = arg \max_{\mu, \kappa} \int_0^{2\pi} f(\theta) Re[\kappa e^{-j(\theta - \mu)}] d\theta \ - \log 2\pi I_0(\kappa) \tag{A.5}$$

Rewriting (A.5) as an expectation and maximizing over μ only,

$$\mu^* = arg \max_{\mu} \kappa \mathbb{E}(Re[e^{-j(\theta - \mu)}]) \tag{A.6}$$

Using the linearity of the expectation and real operators,

$$\mu^* = arg \max_{\mu} \kappa Re[\mathbb{E}(e^{j(\theta - \mu)})] \tag{A.7}$$

We can view (A.7) as an inner product operation and therefore, the maximal value of μ is obtained, according to the Cauchy-Schwartz inequality, for

$$\mu^* = \angle \mathbb{E}(e^{j(\theta)}) \tag{A.8}$$

38

Now we move on to finding the optimal κ, using the fact that we found the optimal μ. For μ^*, the optimal $g(\theta)$ needs to satisfy

$$\frac{\partial D(f||g)}{\partial \kappa} = 0 \tag{A.9}$$

After applying the partial derivative to (A.5), and using

$$\frac{dI_0(\kappa)}{d\kappa} = \frac{I_1(\kappa)}{I_0(\kappa)} \tag{A.10}$$

We get,

$$\mathbb{E}(Re[e^{-j(\theta-\mu^*)}]) = \frac{I_1(\kappa^*)}{I_0(\kappa^*)} \tag{A.11}$$

Recalling (2.1) and (2.2), we get that the optimal Tikhonov distribution $g(\theta)$ is given by matching its circular mean and variance to the circular mean and circular variance of the distribution $f(\theta)$. $\qquad\square$

Appendix B

Using The **CMVM** Operator to Cluster Tikhonov Mixture Components

In algorithms 1 & 2, at each clustering iteration, a set J of mixture components indices of the input Tikhonov mixture (6.1) is selected. The corresponding mixture components are clustered using the CMVM operator. In this appendix we will explicitly compute the application of the CMVM operator and introduce several approximations to speed up the computational complexity. For simplicity, assume that the mixture components in the set J are,

$$f^J(\theta_k) = \sum_{l \in J}^{|J|} \alpha_l \frac{e^{Re[Z_l e^{-j\theta_k}]}}{2\pi I_0(|Z_l|)} \tag{B.1}$$

Using Theorem (2.2.1) and skipping the algebraic details, the **CMVM** operator for (B.1), is:

$$\mathsf{CMVM}(f^J(\theta_k)) = \frac{e^{Re[Z_k^f e^{-j\theta_k}]}}{2\pi I_0(|Z_k^f|)} \tag{B.2}$$

where

$$Z_k^f = \hat{k} e^{j\hat{\mu}} \tag{B.3}$$

and

$$\hat{\mu} = \arg \sum_{l \in J}^{|J|} \alpha_l \frac{I_1(|Z_l|)}{I_0(|Z_l|)} e^{j \arg(Z_l)} \tag{B.4}$$

$$\frac{1}{2\hat{k}} = 1 - \sum_{l \in J}^{|J|} \alpha_l \frac{I_1(|Z_l|)}{I_0(|Z_l|)} Re[e^{j(\hat{\mu} - arg(Z_l))}] \tag{B.5}$$

Since implementing a modified bessel function is computationally prohibitive, we present the following approximation,

$$\log(I_0(k)) \approx k - \frac{1}{2}\log(k) - \frac{1}{2}\log(2\pi) \tag{B.6}$$

which holds for $k > 2$, i.e. reasonably narrow distributions.

Using the following relation,

$$I_1(x) = \frac{dI_0(x)}{dx} \tag{B.7}$$

We find that,

$$\frac{I_1(k)}{I_0(k)} = \frac{d}{dk}(\log(I_0(k))) \tag{B.8}$$

Therefore

$$\frac{I_1(k)}{I_0(k)} \approx 1 - \frac{1}{2k} \tag{B.9}$$

Thus, the approximated versions of (B.5) and (B.4) are

$$\hat{\mu} = \arg[\sum_{l \in J}^{|J|} \alpha_l (1 - \frac{1}{2|Z_l|}) e^{j \arg(Z_l)}] \tag{B.10}$$

$$\frac{1}{2\hat{k}} = 1 - \sum_{l \in J}^{|J|} \alpha_l (1 - \frac{1}{2|Z_l|}) \cos(\hat{\mu} - \arg(Z_l)) \tag{B.11}$$

We also use the approximation for the modified bessel function in the computation of α_l. For a small enough ϵ, $\cos(\hat{\mu} - \arg(Z_l)) \approx 1$, thus one can further reduce the complexity of (B.11)

$$\frac{1}{\hat{k}} = \sum_{l \in J}^{|J|} \alpha_l \frac{1}{|Z_l|} \tag{B.12}$$

which coincides with the computation of a variance of a Gaussian mixture.

Appendix C

Computation of the KL divergence between two Tikhonov Distributions

In this section we will provide the computation of the KL divergence between two Tikhonov distributions, which is a major part of both mixture reduction algorithms. We will also provide approximations used to better the computational complexity of this computation. Suppose two Tikhonov distributions $g_1(\theta)$ and $g_2(\theta)$, where

$$g_1(\theta) = \frac{e^{Re[z_1 e^{-j\theta}]}}{2\pi I_0(|z_1|)} \tag{C.1}$$

$$g_2(\theta) = \frac{e^{Re[z_2 e^{-j\theta}]}}{2\pi I_0(|z_2|)} \tag{C.2}$$

We wish to compute the following KL divergence,

$$D_{KL}(g_1(\theta)||g_2(\theta)) \tag{C.3}$$

which is,

$$D_{KL} = \int_0^{2\pi} g_1(\theta) \log(\frac{e^{Re[z_1 e^{-j\theta}]} I_0(|z_2|)}{e^{Re[z_2 e^{-j\theta}]} I_0(|z_1|)}) d\theta \tag{C.4}$$

Thus,

$$D_{KL} = \log(\frac{I_0(|z_2|)}{I_0(|z_1|)}) + \int_0^{2\pi} g_1(\theta)(Re[z_1 e^{-j\theta}] - Re[z_2 e^{-j\theta}]) d\theta \tag{C.5}$$

After some algebraic manipulations, we get

$$D_{KL} = \log(\frac{I_0(|z_2|)}{I_0(|z_1|)}) + \frac{I_1(|z_1|)}{I_0(|z_1|)}(|z_1| - |z_2|cos(\angle z_1 - \angle z_2)) \tag{C.6}$$

Using (B.9) and (B.6) we get

$$D_{KL} \approx |z_2|(1 - cos(\angle z_1 - \angle z_2)) - \frac{1}{2}\log(\frac{|z_2|}{|z_1|}) + \frac{|z_2|}{2|z_1|}cos(\angle z_1 - \angle z_2) \tag{C.7}$$

Appendix D

Proof of Theorem 6.2

In this section we will prove the claim presented in section 6.2, that under certain channel conditions, the mixture reduction algorithms can be viewed as multiple PLLs tracking the different phase trajectories. For reasons of simplicity, will only show the case where the mixture reduction algorithm converges to a single PLL (the generalization for more than one PLL is trivial, as long as there are no splits). As described earlier, we model the forward messages as Tikhonov mixtures. Suppose the mth component is,

$$p_f^m(\theta_{k-1}) = \frac{e^{Re[z_m^{k-1,f} e^{-j\theta_{k-1}}]}}{2\pi I_0(|z_m^{k-1,f}|)} \tag{D.1}$$

then using (3.17), we get a Tikhonov mixture $f(\theta_k)$,

$$f(\theta_k) = \sum_{i=1}^{M} \alpha_i f_i(\theta_k) \tag{D.2}$$

where,

$$f_i(\theta_k) = \frac{e^{Re[\tilde{z}_{m,i}^{k-1,f} e^{-j\theta_k}]}}{2\pi I_0(|\tilde{z}_{m,i}^{k-1,f}|)} \tag{D.3}$$

$$\tilde{z}_{m,i}^{k-1,f} = \frac{(z_m^{k-1,f} + \frac{r_{k-1} x_i^*}{\sigma^2})}{1 + \sigma_\Delta^2 |(z_m^{k-1,f} + \frac{r_{k-1} x_i^*}{\sigma^2})|} \tag{D.4}$$

and x_i is the ith constellation symbol. We insert (D.2) into the mixture reduction algorithms. Assuming slowly varying phase noise and high SNR, such that the mixture reduction will cluster all the mixture components, with non negligible probability, to one Tikhonov distribution. Then, the *circular* mean, $\hat{\theta}_k$, of the clustered Tikhonov distribution is computed according to,

$$\hat{\theta}_k = \angle \mathbb{E}(e^{j\theta_k}) \tag{D.5}$$

where the expectation is over the distribution $f(\theta_k)$. We note that for every complex

43

valued scalar z, the following holds

$$\angle z = \Im(\log z) \tag{D.6}$$

where $\Im$ denotes the imaginary part of a complex scalar. If we apply (D.6) to (D.5) we get,

$$\hat{\theta}_k = \Im\left(\log \sum_{i=1}^{M} \alpha_i \frac{\tilde{z}_{m,i}^{k-1,f}}{|\tilde{z}_{m,i}^{k-1,f}|}\right) \tag{D.7}$$

which can be rewritten as,

$$\hat{\theta}_k = \Im\left(\log \sum_{i=1}^{M} \alpha_i \frac{z_m^{k-1,f} + \frac{r_{k-1}x_i^*}{\sigma^2}}{|z_m^{k-1,f} + \frac{r_{k-1}x_i^*}{\sigma^2}|}\right) \tag{D.8}$$

we denote,

$$G_{k-1} = |z_m^{k-1,f} + \frac{r_{k-1}x_i^*}{\sigma^2}| \tag{D.9}$$

and assume that G_{k-1}, the conditional causal MSE of the phase estimation under mixture component $f_i(\theta_k)$, is constant for all significant components. Then,

$$\hat{\theta}_k \approx \hat{\theta}_{k-1} + \Im\left(\log\left(1 + \frac{r_{k-1}}{G_{k-1}z_m^{k-1,f}\sigma^2}\left(\sum_{i=1}^{M} \alpha_i x_i^*\right)\right)\right) \tag{D.10}$$

where,

$$\hat{\theta}_{k-1} = \angle z_m^{k-1,f} \tag{D.11}$$

We will define c_{soft} as the soft decision symbol using the significant components,

$$c_{soft} = \sum_{i=1}^{M} \alpha_i x_i \tag{D.12}$$

Since we assume high SNR and small phase noise variance, then the tracking conditional MSE will be low, i.e $|z_1^{k,f}|$ will be high. Using the fact that for small angles ϕ,

$$\angle(1 + \phi) \approx \Im(\phi) \tag{D.13}$$

Therefore,

$$\hat{\theta}_k \approx \hat{\theta}_{k-1} + \Im\left(\frac{r_{k-1}c_{soft}^*}{G_{k-1}z_m^{k-1,f}\sigma^2}\right) \tag{D.14}$$

Which, again for small angles x, $sin(x) \approx x$,

$$\hat{\theta}_k \approx \hat{\theta}_{k-1} + \frac{|r_{k-1}||c_{soft}^*|}{G_{k-1}|z_m^{k-1,f}|\sigma^2}(\angle r_{k-1} + \angle c_{soft}^* - \hat{\theta}_{k-1}) \tag{D.15}$$

Bibliography

[1] Zhu Aimin and Yao Senjie. Design of optimal ldpc code for the system of mdpsk concatenated with ldpc codes. In *Proceedings of the IEEE International Conference on Communication Software and Networks*, 2009.

[2] Achilleas Anastasopoulos and Keith M. Chugg. Adaptive iterative detection for phase tracking in turbo-coded systems. *IEEE Transactions on Communications*, 49, 2001.

[3] Alan Barbieri and Giulio Colavolpe. Soft-output decoding of rotationally invariant codes over channels with phase noise. *IEEE Transactions on Communications*, 55, 2007.

[4] Mao-Ching Chiu. Bandwidth-efficient modulation codes based on nonbinary irregular repeatâĂŞaccumulate codes. *IEEE Transactions on Information Theory*, 56, 2010.

[5] Kwonhue Choi. Residual frequency offset compensation-embedded turbo decoder. *IEEE Transactions on Vehicular Technology*, 57, 2008.

[6] Giulio Colavolpe. On ldpc codes over channels with memory. *IEEE Transactions on Wireless Communications*, 5:1757 –1766, July 2006.

[7] Giulio Colavolpe, Alan Barbieri, and Giuseppe Caire. Algorithms for iterative decoding in the presence of strong phase noise. *IEEE Journal on Selected Areas in Communications*, 23:1748 –1757, September 2005.

[8] David F. Crouse, Peter Willett, Krishna Pattipati, and Lennart Svensson. A look at gaussian mixture reduction algorithms. In *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, 2011.

[9] Minh N. Do. Fast approximation of kullback-leibler distance for dependence trees and hidden markov models. *IEEE Signal Processing Letters*, 10:115 – 118, April 2003.

[10] Peixin Dong, Pingyi Fan, and Li Yi. Iterative phase recovery method of turbo-coded 16qam based on soft symbol to bit mapping. In *Proceedings of the IEEE 5th International Symposium on Multi-Dimensional Mobile Communications*, 2004.

[11] Achilleas Anastasopoulos Giulio Colavolpe Ferrari, Gianluigi and Riccardo Raheli. Adaptive iterative detection for the phase-uncertain channel: Limited-tree-search versus truncated-memory detection. *IEEE Transactions on Vehicular Technology*, 53, 2004.

[12] Michele Franceschini, Gianluigi Ferrari, Riccardo Raheli, and Aldo Curtoni. A novel class of low-complexity siso algorithms for phase-uncertain communications. In *Proceedings of the IEEE International Symposium on Information Theory*, 2005.

[13] Michele Franceschini, Gianluigi Ferrari, Riccardo Raheli, and Aldo Curtoni. Serial concatenation of ldpc codes and differential modulations. *IEEE Journal on Selected Areas in Communications*, 23, 2005.

[14] Michele Franceschini, Gianluigi Ferrari, Riccardo Raheli, and Aldo Curtoni. Serial concatenation of ldpc codes and differential modulations. *IEEE Journal on Selected Areas in Communications*, 23, 2005.

[15] Jacob Goldberger and Sam Roweis. Hierarchical clustering of a mixture model. 2004.

[16] CÃľdric Herzet, ValÃľry Ramon, and Luc Vandendorpe. A theoretical framework for iterative synchronization based on the sumâĂŞproduct and the expectation-maximization algorithms. *IEEE Transactions on Signal Processing*, 5, 2007.

[17] Rodrigue Imad and Sebastien Houcke. Blind frame synchronization and phase offset estimation for coded systems. In *Proceedings of the IEEE 9th Workshop on Signal Processing Advances in Wireless Communications*, 2008.

[18] Sridhar Karuppasami and William G. Cowley. Ldpc code aided phase ambiguity resolution. In *Proceedings of the IEEE International Conference on Information, Decision and Control*, 2007.

[19] Sridhar Karuppasami and William G. Cowley. Construction and iterative decoding of ldpc codes over rings for phase-noisy channels. *EURASIP Journal on Wireless Communications and Networking*, January 2008.

[20] Sridhar Karuppasami, William G. Cowley, and Steven S. Pietrobon. Ldpc code construction and iterative receiver techniques for channels with phase noise. In *Proceedings of the IEEE International Conference on Vehicular Technology*, 2008.

[21] Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22:79–86, March 1951.

[22] LH-J. Lampe, , and Robert Schober. Decision-feedback differential demodulation of bit-interleaved coded mdpsk. *IEEE Electronics Letters*, 35, 1999.

[23] Didier Le Ruyet, Tanya Bertozzi, and Nicolas Paul. Particle filtering for iterative data and phase estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006.

[24] Frederic Lehmann. A gaussian sum approach to blind carrier phase estimation and data detection in turbo coded transmissions. *IEEE Transactions on Communications*, 57, 2009.

[25] Ching-Chi Lo and Szu-Lin Su. Siso fb-map detection for turbo-coded differential psk systems. *IEEE Transactions on Vehicular Technology*, 58, 2009.

[26] Benvenuti Lorenzo, Giugno Luca, Lottici Vincenzo, and Luise Marco. Code-aware carrier phase noise compensation on turbo-coded spectrally-efficient high-order modulations. In *Proceedings of the 8th International Workshop on Signal Processing for Space Communications*, 2003.

[27] Vincenzo Lottici and Marco Luise. Carrier phase recovery for turbo-coded linear modulations. In *Proceedings of the IEEE International Conference on Communications*, 2002.

[28] Vincenzo Lottici and Marco Luise. Embedding carrier phase recovery into iterative decoding of turbo-coded linear modulations. *IEEE Transactions on Communications*, 52, 2004.

[29] Kanti V. Mardia and Peter E. Jupp. *Directional Statistics*. John Wiley and Sons Ltd., 2000.

[30] Wataru Matsumoto and Hideki lmai. Blind synchronization with enhanced sum-product algorithm for low-density parity-check codes. In *Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications*, 2002.

[31] Elisa Mo and Pooi Yuen Kam. Log-likelihood metrics based on two-symbol-interval observations for ldpc codes with bdpsk transmission. In *Proceedings of the IEEE 68th Vehicular Technology Conference*, 2008.

[32] Elisa Mo and Pooi Yuen Kam. Log-likelihood ratios for ldpc codes with pilot-symbol-assisted bpsk transmission over the noncoherent channel. In *Proceedings of the IEEE International Conference on Wireless Communications and Networking*, 2009.

[33] Elisa Mo and Pooi Yuen Kam. Log-likelihood ratios for ldpc codes with pilot-symbol-assisted bpsk transmission over the noncoherent channel. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, 2009.

[34] Idin Motedayen and Achilleas Anastasopoulos. Polynomial-complexity, adaptive symbol-by-symbol soft-decision algorithms with application to non-coherent detection of ldpcc. In *Proceedings of the IEEE International Conference on Communications*, 2002.

[35] Mauri Nissila and Subbarayan Pasupathy. Adaptive iterative detectors for phase-uncertain channels via variational bounding. *IEEE Transactions on Communications*, 57, 2009.

[36] Mauri J. Nissila, Subbarayan Pasupathy, and Aarne Mammela. An em approach to carrier phase recovery in awgn channel. In *Proceedings of the IEEE International Conference on Communications*, 2001.

[37] Nele Noels, Vincenzo Lottici, Antoine Dejonghe, Heidi Steendam, Marc Moeneclaey, Marco Luise, and Luc Vandendorpe. A theoretical framework for soft-information-based synchronization in iterative (turbo) receivers. *EURASIP Journal on Wireless Communications and Networking*, April 2005.

[38] Nele Noels, Heidi Steendam, and Marc Moeneclaey. Carrier phase tracking from turbo and ldpc coded signals affected by a frequency offset. *IEEE Communications Letters*, 9, 2005.

[39] Nele Noels, Heidi Steendam, and Marc Moeneclaey. Performance analysis of ml-based feedback carrier phase synchronizers for coded signals. *IEEE Transactions on Signal Processing*, 55, 2007.

[40] Nele Noels, Heidi Steendam, Marc Moeneclaey, and Herwig Bruneel. A maximum-likelihood based feedback carrier synchronizer for turbo-coded systems. In *Proceedings of the 61st IEEE Vehicular Technology Conference*, 2005.

[41] Rza Nuriyev and Achilleas Anastasopoulos. Analysis of joint iterative decoding and phase estimation for the noncoherent awgn channel, using density evolution. In *Proceedings of the IEEE International Symposium on Information Theory*, 2002.

[42] Rza Nuriyev and Achilleas Anastasopoulos. Capacity and coding for the block-independent noncoherent awgn channel. *IEEE Transactions on Information Theory*, 51, 2005.

[43] Wangrok Oh and Kyungwhoon Cheun. Joint decoding and carrier phase recovery algorithm for turbo codes. *IEEE Communications Letters*, 5:375 –377, September 2001.

[44] Wangrok Oh and Kyungwhoon Cheun. Iterative decoding and channel parameter estimation algorithms for repeat-accumulate codes. *IEEE Transactions on Communications*, 53, 2005.

[45] Arif Onder Isikman and Hani Mehrpouyan. Phase noise estimation for uncoded/coded siso and mimo systems. 2012.

[46] Yossef Rahamim, Avraham Freedman, and Arie Reichman. Ml iterative soft-decision-directed (ml-isdd): a carrier synchronization system for short packet turbo coded communication. *IEEE Transactions on Communications*, 56, 2008.

[47] Andrew R. Runnalls. Kullback-leibler approach to gaussian mixture reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 43:989 –999, July 2007.

[48] Shachar Shayovitz and Dan Raphaeli. Efficient iterative decoding of ldpc in the presence of strong phase noise. In *Proceedings of The 7th International Symposium on Turbo Codes & Iterative Information Processing*, 2012.

[49] Shachar Shayovitz and Dan Raphaeli. Improved message passing algorithm for phase noise channels using optimal approximation of tikhonov mixtures. In *Proceedings of the 5th International Symposium on Communications, Control and Signal Processing, ISCCSP 2012, Rome, Italy, 2-4 May 2012*, 2012.

[50] Shachar Shayovitz and Dan Raphaeli. Multiple hypotheses iterative decoding of ldpc in the presence of strong phase noise. In *Proceedings of The 2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel*, 2012.

[51] Ilan Sutskover and Shlomo Shamai. Iterative decoding of low-density parity-check codes over compound channels. *IEEE Transactions on Communications*, 11, 2005.

[52] Isabel M. Tienda-Lunaa, Diego P. Ruizb, Yufei Huangc, and MarÃŋa C. CarriÃşnb. Iterative decoding in factor graph representation using particle filtering. In *Proceedings of the IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, 2005.

[53] Esteban L. VallẤes, Richard D. Wesel, John D. Villasenor, and Christopher R. Jones. Carrier and timing synchronization of bpsk via ldpc code feedback. In *Proceedings of the IEEE Fortieth Asilomar Conference on Signals, Systems and Computers*, 2006.

[54] Andrew P. Worthen and Wayne E. Stark. Unified design of iterative receivers using factor graphs. *IEEE Transactions on Information Theory*, 47:843 –849, February 2001.

[55] N. Wu, H. Wang, , and J-M. Kuang. Code-aided iterative carrier phase estimation using selective soft decision feedback. *IEEE Electronics Letters*, pages 691–692, June 2009.

[56] Xiaofu Wu and Haige Xiang. Iterative carrier phase recovery methods in turbo receivers. *IEEE Communications Letters*, 9, 2005.

[57] Xiaofu Wu, Haige Xiang, Cong Ling, Xiaohu You, and Shaoqian Li. Bounds on the decoding error probability of binary block codes over noncoherent block awgn and fading channels. *IEEE Transactions on Wireless Communications*, 5, 2006.

[58] Li Zhang and Alister Burr. Low-complexity iterative carrier phase and symbol timing recovery approach for turbo-codes system. In *Proceedings of the 5th European Personal Mobile Communications Conference*, 2003.

[59] Li Zhang and Alister G. Burr. Iterative carrier phase recovery suited to turbo-coded systems. *IEEE Transactions on Wireless Communications*, 3, 2004.

תקציר

בתיזה זו מוצג אלגוריתם איטרטיבי בעל סיבוכיות נמוכה לפענוח תקשורת מקודדת בערוצים עם רעש פאזה חזק. האלגוריתם מבוסס על אלגוריתם Sum & Product כאשר הודעות של צמתים המשויכים לרעש הפאזה ממודלים כמודל עירוב Tikhonov.

מאחר והסקה בייסיאנית כמו Sum & Product מגדילה את סדר מודל העירוב באופן אקספוננציאלי מהודעה אחת לבאה אחריה, נדרש אלגוריתם להורדת מימדיות. אנו מציעים אלגוריתם להורדת מימדיות אשר מוצא מודל עירוב קטן יותר שעבורו מטריקת חוסר הדימיון חסומה מלעיל ע״י סף נתון. כחלק מאלגוריתם הורדת המימדיות, אנו מספקים שיטה לאשכול אופטימלי אשר מוצאת את הפילוג טיכונוב הקרוב ביותר , במובן Kullback-Liebler, לכל פילוג מעגלי. כמו כן, אנו מציגים שיטה להגבלת סדר מודל העירוב ועוד שיטות להורדת סיבוכיות.

אנו מציגים תוצאות סימולציות וניתוח סיבוכיות עבור האלגוריתם המוצע אשר מראות ביצועים טובים יותר מאשר שיטות מתקדמות ביותר בעלות סיבוכיות נמוכה. האלגוריתם הציג ביצועים עליונים עבור קצב קוד גבוה, רעש פאזה חזק, קונסטלציות גבוהות ומעט פיילוטים.

לבסוף, אנו מראים כי קירוב מודל עירוב Tikhonov להודעות Sum & Product שקול למעקב אחר מספר מסלולי רעש פאזה אפשריים, או במידה מסויימת למספר חוגים עוקבי פאזה (PLL) חכמים.

אלגוריתמי העברת מסרים המבוססים על עירובי טיכונוב לטובת עקיבה אחר רעש פאזה

חיבור זה הוגש כעבודת גמר לקראת התואר "מוסמך אוניברסיטה"
בהנדסת חשמל ואלקטרוניקה

על-ידי

שחר שיוביץ

אב התשע"ג

אלגוריתמי העברת מסרים המבוססים על עירובי טיכונוב לטובת עקיבה אחר רעש פאזה

חיבור זה הוגש כעבודת גמר לקראת התואר ״מוסמך אוניברסיטה״
בהנדסת חשמל ואלקטרוניקה

על-ידי

שחר שיוביץ

אב התשע״ג