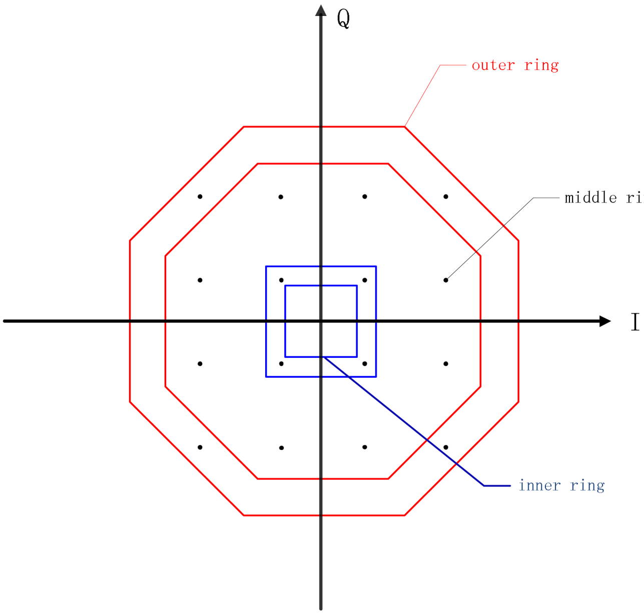
CHALMERS



Phase Noise Estimation for Coherent Fiber-Optic Communication

Master Thesis in Master Program Communications Engineering

Fangrong Peng

Devision of Communications
Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden

EXE01/2002

Abstract

Phase noise estimation is a significant issue in coherent fiber-optic communication systems. In this project we develop a symbol-by-symbol phase noise estimation algorithm for 16-QAM which is evaluated for coherent Polarization Multiplexing (POLMUX).

Two polarizations X and Y are affected by common phase noise, and each one has it own estimates of the common phase noise. On the other hand, the combination of these two estimates is also derived. It is demonstrated that the information fusion enhances performance of the fiber-optic communication system with respect to two criteria: probability of cycle slips and estimation error variance.

Through the Monte Carlo simulations, it is verified that the proposed phase noise estimator can cope with laser linewidths of up to 2.0 MHz in high SNR regimes, at 100 Gb/s.

KEYWORDS: fiber-optic communication system, symbol-by-symbol phase noise estimation, POLMUX 16QAM.

CONTENTS

1 INTRODUCTION 1

2 MODELING AND ALGORITHM DERIVATION 3

2.1 Background 3

2.2 Observation Model 3

2.3 Algorithms 4

2.3.1 Phase offset estimation 5

2.3.2 Symbol-by-symbol phase noise estimation 6

2.3.3 Polarization combination 13

2.4 Performance measures 15

2.4.1 Cycle slip detection 16

2.4.2 Estimation error variance 17

3 IMPLEMENTATION 19

3.1 Algorithms 19

3.1.1 Transmission channel 19

3.1.2 Initial phase offset 20

3.1.3 Raw estimation 20

3.1.4 Phase unwrapping 22

3.1.5 Phase combination 23

3.2 Performance measures 26

3.2.1 Cycle slip detection 26

3.2.2 Estimation error variance 27

4 DISCUSSION 29

4.1 Ring error probability 29

4.2 Cycle slip statistics 30

4.3 Estimation error variance statistics 30

4.4 Comparison with existing algorithms 31

4.5 Symbol error rate 31

5 CONCLUSION 35

5.1 Completed work 35

5.2 Findings 36

5.3 Future work 36

1 INTRODUCTION

16-QAM POLMUX transmission has the potential to double throughput compared to QPSK-POLMUX, but is more sensitive to impairments in the transmission link. In particular, phase noise caused by the non-zero linewidth of lasers is an impairment that needs to be compensated for prior to data detection. Phase noise is characterized by the product $\Delta_f T$, where Δ_f is the sum linewidth of the signal and local oscillator lasers, and $1/T$ is the baud rate.

Traditional technique to estimate the phase noise is feedback phase noise estimation using a phase-locked loop (PLL), however, it can only cope with small $\Delta_f T$, due to the large delay that PLL exhibits, while feed-forward (FF) phase noise estimation is promising for larger $\Delta_f T$ (Ip and Kahn 2007, Pfau *et al.* 2009). A demonstrated comparison between FF carrier recovery and PLL shows that FF carrier recovery can tolerate 1.5 to 2 times higher laser linewidth than PLL, even using an optimistic assumption for the performance of the PLL (Kahn and Ip 2009). Conventional FF estimation is based on block processing, considering the phase to be approximately constant over the duration of a block. As the block size decreases, faster adaptation is possible but with degraded tracking performance(Seimetz 2008). For QPSK, values of $\Delta_f T \approx 10^{-3}$ can be tolerated with negligible loss under FF estimation(Pfau *et al.* 2009). For 16-QAM, the reduced angular separation and the presence of multiple amplitude levels reduces the phase noise tolerance. Symbol-by-symbol phase estimators (SBSPE) are a potential way to cope with larger values of $\Delta_f T$. They can be seen as a special case of block-based estimators (Seimetz 2008, Fatadin *et al.* 2010), and have also received considerable interest on their own (Louchet *et al.* 2008).

In this thesis, we (i) provide the first explicit derivation of SBSPE; (ii) analyze the error probability of the ring detector (Seimetz 2008, Fatadin *et al.* 2010, Louchet *et al.* 2008); (iii) discuss the SNR and $\Delta_f T$ regime in which SBSPE can operate reliably, confirming previous experimental findings (Louchet *et al.* 2008); (iv) develop a novel algorithm to combine estimates from both polarizations. Through Monte Carlo simulations, we demonstrate the superior performance of the SBSPE, compared to state-of-the-art block-based estimators (Louchet *et al.* 2008).

The remainder of this document is structured as follows. In Chapter 2 we will describe the system model, algorithms of implementing the system and performance measures; in Chpater 3 we will introduce the implementation of system model, the proposed phase noise estimator, and the performance measures; in Chapter 4 we will further discuss regarding some specific issues in the implementation; in Chapter 5 we will conclude the whole thesis project.

2 MODELING AND ALGORITHM DERIVATION

In this chapter, we will introduce the description of the problem, including the background of the problem, the observation model and the algorithms of several issues to be solved in this project.

2.1 Background

Considering optical detection, there are usually three methods: non-coherent detection, differentially coherent detection and coherent detection. The focus of this thesis is on the third method, coherent detection. An optical coherent detection scheme detects not only the signal's amplitude but also phase and polarization, which increases the detection capacity and spectral efficiency.

The analytical model of coherent detection is shown in Fig.(2.1) (Kahn and Ip 2009). In this model, we have:

$$\mathbf{y}(t) = \mathbf{h}(t) * \mathbf{x}(t) + \mathbf{n}(t),$$

taking the Fourier transform on both side, we get:

$$\mathbf{Y}(\omega) = \mathbf{H}(\omega)\mathbf{X}(\omega) + \mathbf{N}(\omega),$$

so the channel frequency response is:

$$\mathbf{H}(\omega) = e^{-j\frac{1}{2}\beta_2 L\omega^2}\mathbf{R}_1^T\mathbf{D}(\omega)\mathbf{R}_2,$$

where the component $e^{-j\frac{1}{2}\beta_2 L\omega^2}$ is the chromatic dispersion, $\mathbf{R}_1^T\mathbf{D}(\omega)\mathbf{R}_2$ is the polarization mode dispersion, and $\mathbf{R}_1$ and $\mathbf{R}_2$ are two rotation matrices.

In this thesis, we are mainly interested in the Wiener phase noise, rather than other impairments, such as chromatic dispersion and polarization mode dispersion, so we only research on the model that these impairments have been compensated. The observation model of this thesis will be introduced in the next section.

2.2 Observation Model

We try to exploit POLMUX in coherent fiber optic communication systems to increase the data rate and perform better phase noise estimation. Specifically speaking, we use two polarizations, say X and Y, to transmit information symbols in the same channel, so the phase noises imposed on them are equal to each other.

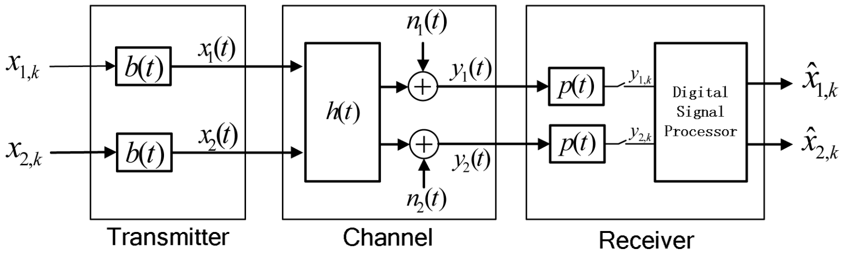


Figure 2.1. Analytical model of coherent system

The observation model is described as follows. We assume that other impairments have been compensated for by the receiver, including chromatic dispersion and polarization mode dispersion, then the received symbols at time k can be expressed as (Ip and Kahn 2007)

$$\mathbf{r}_k = \begin{bmatrix} a_k^{(\mathrm{X})} e^{j\phi + j\theta_k} \\ a_k^{(\mathrm{Y})} e^{j\theta_k} \end{bmatrix} + \mathbf{n}_k, \tag{2.1}$$

where $a_k^{(i)}$ is the kth 16-QAM symbol on polarization $i \in \{\mathrm{X}, \mathrm{Y}\}$, $\mathbf{n}_k$ is modeled as i.i.d. zero-mean complex Gaussian noise with variance σ^2 per real dimension, ϕ is a constant phase offset between polarization X and Y, and θ_k is the phase noise at time k. In optical communications, phase noise can be modeled as a Wiener process (Ip and Kahn 2007)

$$\theta_k = \sum_{m=-\infty}^{k} \nu_m, \tag{2.2}$$

where ν_m is previous phase noise and i.i.d with variance $\sigma_\theta^2 = 2\pi\Delta_f T$. The SNR is defined as $E_\mathrm{s}/(2\sigma^2)$, where E_s is the average energy per 16-QAM symbol per polarization.

From the observation model, we can easily find that these two polarizations have an identical phase except for a constant phase offset ϕ, which can be easily ignored after it has been accurately estimated.

2.3 Algorithms

This section is mainly concerned the algorithms of phase noise estimation, and our focus is on blind algorithms, which assume no a priori knowledge of the transmitted symbols. In this section, the project is divided into several parts and the algorithms of each part are described in detail. These parts are:

- Phase offset estimation
- Symbol-by-symbol phase noise estimation
    - Ring discrimination

  - Phase estimation
  - Phase unwrapping
- Polarization combination
- Phase noise filtering

According to the observation model, in the channel there are two independent polarizations with an identical phase noise and a phase offset. The phase offset is constant, and it can be accurately estimated by a long block of observations, which will be explained in detail in Section 2.3.1. So after the compensation for the constant phase offset, polarizations X and Y have an identical phase.

Then, a symbol-by-symbol phase noise estimation can be done on each polarization, with the procedures introduced in Section 2.3.2, including three steps described in corresponding subsections. Combining the estimates from two independent polarizations is one of the most important part in this project, which directly and significantly affects the performance of the whole project, and is demonstrated in Section 2.3.3.

To evaluate the performance of our algorithms, we need to detect and quantify the statistics of cycle slips and evaluate the variance of phase noise estimation, and the particular methods are described in Section 2.4.1 and Section 2.4.2.

2.3.1 Phase offset estimation

Before estimating the phase noise, we need to estimate the phase offset between polarizations X and Y. Since the 16-QAM constellation has a rotational symmetry of pi/2, we can only estimate the phase up to an ambiguity of $k \times \pi/2$, so we limit the phase offset ϕ in the range $[0, \pi/2]$, actually it can be in the range $[0, 2\pi)$. This ambiguity can be resolved in subsequent processing by considering training symbols or through differential detection.

The phase offset estimator operates as follows. First, the observation $r_k^{(\mathrm{X})}$ and $r_k^{(\mathrm{Y})}$ are broken into blocks of length N, with the phase offset estimate in the nth block given by:

$$\hat{\phi}_n = \frac{1}{4}\measuredangle\left(\sum_{l=s_n}^{s_n+N-1} r_l^{(\mathrm{X})4} - \sum_{l=s_n}^{s_n+N-1} r_l^{(\mathrm{Y})4}\right), \tag{2.3}$$

where $\measuredangle(\cdot)$ denotes the phase and s_n denotes the index of the first symbol in the nth block. Second, the phase offset estimates in all blocks are averaged:

$$\hat{\phi} = \frac{1}{K}\sum_{m=1}^{K} \hat{\phi}_n, \tag{2.4}$$

where K is the number of blocks. Note that the result may have up to $\pi/2$ ambiguity, because the function $\measuredangle(\cdot)$ always returns a value in the range $[0, \pi/2]$.

2.3.2 Symbol-by-symbol phase noise estimation

In this section we will introduce the algorithm of symbol-by-symbol phase estimation. First of all, we introduce a conventional FF estimator which operates as follows. First, the observation r is broken into blocks of length M, with the phase estimate in the mth block given by the Viterbi &Viterbi estimator:

$$\hat{\theta}_m^{\mathrm{r}} = \frac{1}{4}\measuredangle\left(\sum_{l=s_m}^{s_m+M-1} r_l^4\right), \tag{2.5}$$

where s_m is the index of the first symbol in the mth block, note that we focus on a single polarization and omit the superscripts X and Y. Since the phase estimates will be limited in the range $[0, \pi/2]$, they need to be unwrapped and low-pass filtered.

In this project, the method we choose, SBSPE, is a special case of (2.5), when $M = 1$. Our algorithm includes three steps: the first one is to discriminate on which circle the symbol lies because the amplitude of a symbol is useful information for estimating its phase in 16-QAM constellation; the second step is phase estimation based on the ring information we have obtained; and the third one is phase unwrapping, which is used to make the phase noise estimate sequence continuous.

Ring discrimination

In a 16-QAM constellation, symbols can be divided into three groups on the basis of their amplitudes, say outer ring, middle ring and inner ring, as shown in Fig. (2.2). The first step in the algorithm of symbol-by-symbol phase noise estimation is ring discrimination.

The model is shown as follows. Introducing $\rho_k = |r_k|^2$, for medium-to-high SNR, we can model $\rho_k \sim \mathcal{N}(\mu_{R,\sigma^2}, V_{R,\sigma^2})$, where R is the radius of the circle on which the kth transmitted symbol lies, σ^2 is the variance per real dimension of the channel noise, and μ_{R,σ^2} and V_{R,σ^2} are mean and variance of ρ_k, respectively.

We find that

$$\mu_{R,\sigma^2} = R^2 + 2\sigma^2, \tag{2.6}$$

$$V_{R,\sigma^2} = 4\sigma^4 + 4R^2\sigma^2. \tag{2.7}$$

To verify this conclusion, we find out 25128 symbols on the inner ring, 49993 on the middle ring and 24879 on the outer ring in total 100000 samples and plot histograms for each rings in Fig. (2.3) as the first three figures. Among them, the common point is that all of these figures can be considered as Gaussian distribution. Looking at them one by one, we observe that the mean values of them are approximately 0.2, 1.0 and 1.8, respectively, the same as the values of R_{i}^2, R_{m}^2 and R_{o}^2 (we have normalized the average energy of symbols to 1, which means $1/4R_{\mathrm{i}}^2 + 1/2R_{\mathrm{m}}^2 + 1/4R_{\mathrm{o}}^2 = 1$),

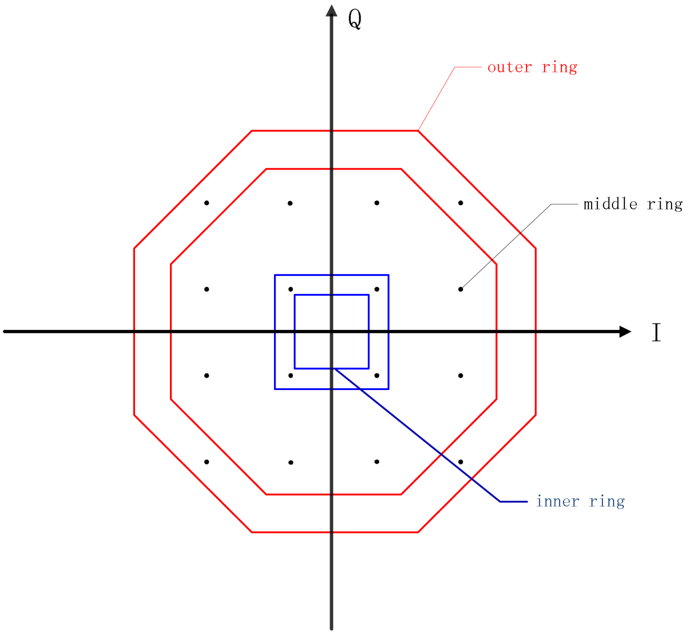


Figure 2.2. Ring division of 16-QAM constellation

which demonstrates the validity of (2.6). If we make the range of the axis the same for each ring and plot them in one figure, as shown in the forth figure in Fig. (2.3), we can see that the variance increases with the radius growing up, which accords to the conclusion we obtain in (2.7).

How (2.6) follows is shown. $\rho_k = |r_k|^2 = r_k \cdot r_k^* = |a_k|^2 + |n_k|^2 + 2\mathcal{R}e\{a_k \cdot e^{j\theta_k} \cdot n_k^*\}$, where n_k is the noise of one polarization at time k,

$$\begin{aligned}\mu_{R,\sigma^2} &= \mathbb{E}[{\rho_k}^2] \\ &= \mathbb{E}[|a_k|^2] + \mathbb{E}[|n_k|^2] + \mathbb{E}[2\mathcal{R}e\{a_k \cdot e^{j\theta_k} \cdot n_k^*\}] \\ &= R^2 + 2\sigma^2 + 2\mathbb{E}[\mathcal{R}e\{a_k \cdot e^{j\theta_k} \cdot n_k^*\}] \\ &= R^2 + 2\sigma^2 + 2\mathcal{R}e\{\mathbb{E}[a_k \cdot n_k^*]\} \\ &= R^2 + 2\sigma^2 + 2\mathcal{R}e\{\mathbb{E}[a_k] \cdot \mathbb{E}[n_k^*] \cdot \mathbb{E}[e^{j\theta_k}]\} \\ &= R^2 + 2\sigma^2, \end{aligned} \tag{2.8}$$

thus $\mu_{R,\sigma^2} = R^2 + 2\sigma^2$.

The second equation in (2.7) follows because:

$$\begin{aligned}V_{R,\sigma^2} &= \mathbb{E}[(\rho - \mu_{R,\sigma^2})^2] \\ &= \mathbb{E}[\rho^2] - \mu_{R,\sigma^2}^2 \\ &= \mathbb{E}[|a_k|^4 + |n_k|^4 + 4\mathcal{R}e^2\{a_k \cdot e^{j\theta_k} \cdot n_k^*\} + 2|a_k|^2|n_k|^2 \\ &\quad + 4|a_k|^2\mathcal{R}e\{a_k \cdot e^{j\theta_k} \cdot n_k^*\} + 4|n_k|^2\mathcal{R}e\{a_k \cdot e^{j\theta_k} \cdot n_k^*\}] - \mu_{R,\sigma^2}^2 \\ &= R^4 + 8\sigma^4 + 4R^2\sigma^2 + 4R^2\sigma^2 + 0 + 0 - (R^2 + 2\sigma^2)^2 \\ &= 4\sigma^4 + 4R^2\sigma^2, \end{aligned} \tag{2.9}$$

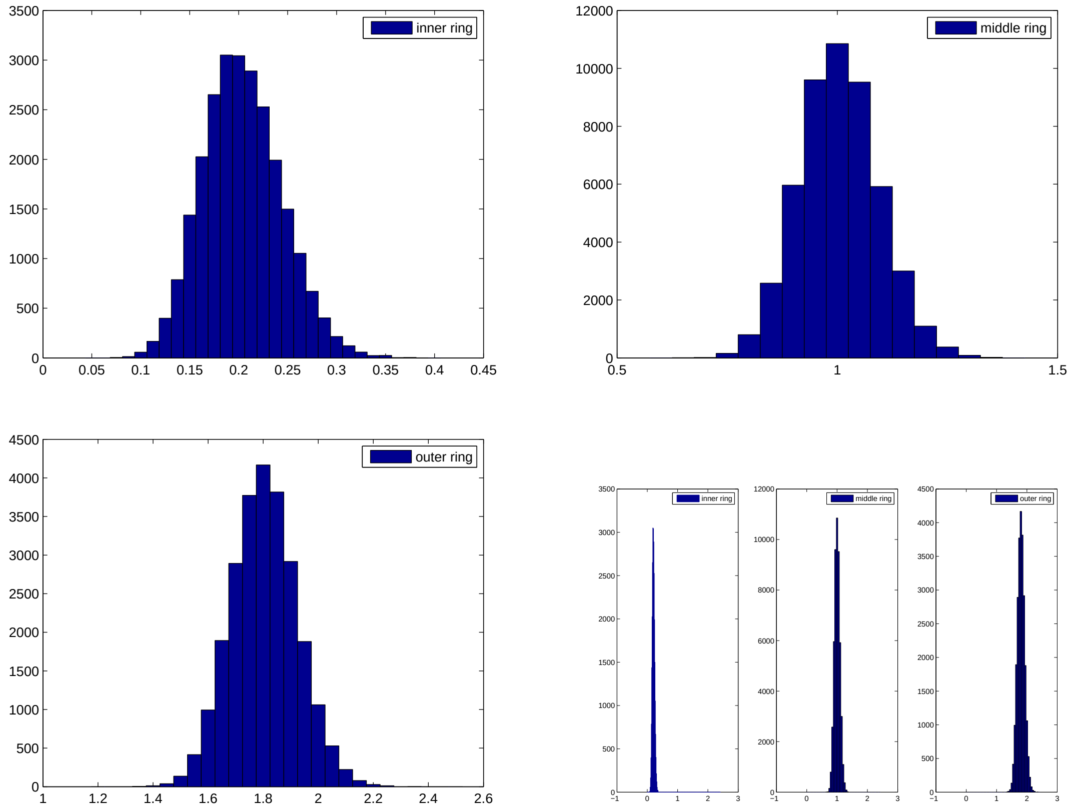


Figure 2.3. Histogram of different rings in received sequence

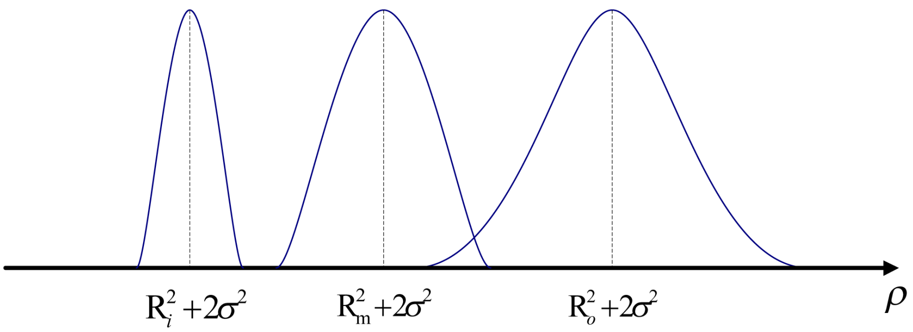


Figure 2.4. Probability distribution of three rings

thus $V_{R,\sigma^2} = 4\sigma^4 + 4R^2\sigma^2$.

The optimal, maximum a posteriori (MAP) estimate of the ring, given the observation ρ_k, is then

$$\hat{R}_k = \arg\max_r \left\{ \frac{p(R=r)}{\sqrt{2\pi}V_{R,\sigma^2}} e^{-\frac{\left(\rho_k-\mu_{R,\sigma^2}\right)^2}{2V_{R,\sigma^2}}} \right\}, \tag{2.10}$$

It is easier to solve this equation if taking logarithm function, and then the optimal estimate of the ring is:

$$\hat{R}_k = \arg\max_r \left\{ \log \frac{p(R=r)}{V_{R,\sigma^2}} - \frac{\left(\rho_k - \mu_{R,\sigma^2}\right)^2}{2V_{R,\sigma^2}} \right\}, \tag{2.11}$$

where r and $\hat{R}_k$ take their values in the set $\{R_\mathrm{i}, R_\mathrm{m}, R_\mathrm{o}\}$, which are the radiuses of the inner, middle, and outer ring, respectively. We write $p(R=r)$ as $p(r)$ for short, so the probability mass function of R is $p(R_\mathrm{i}) = 1/4$, $p(R_\mathrm{o}) = 1/4$, and $p(R_\mathrm{m}) = 1/2$.

From (2.7) it is observed that the outer ring, middle ring and inner ring have descending variances of ρ_k given σ, so we can predict that most of the error of ring discrimination would be located between the outer ring and the middle ring, as shown in Fig. (2.4). This will be discussed in the following chapters.

Phase estimation

Once the ring has been determined, our object becomes to estimate $\hat{\theta}_k$ from r_k, given $\hat{R}_k$ and $\hat{\theta}_{k-1}$. A block diagram of the phase noise estimator is shown in Fig. (2.5) (Ip and Kahn 2007).

According to the block diagram, the phase noise estimator is:

$$\hat{\theta}_k = \frac{1}{4} \measuredangle r_k^4. \tag{2.12}$$

In the following we explain the principles of this phase noise estimator. For 16-QAM, the whole constellation can be considered as a superposition of four components, two

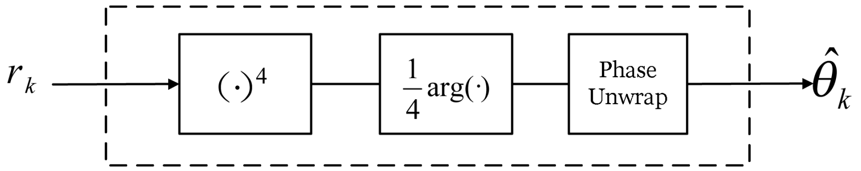


Figure 2.5. Phase Estimator

QPSK constellations (the outer and inner ring) and two rotated QPSK constellations (the middle ring), as shown in Fig. (2.6). Therefore, each symbol on the 16-QAM constellation can be represented as

$$a_m = Ae^{j(m\pi/2+\varphi)}, m = 0, 1, 2, 3 \tag{2.13}$$

where A is the amplitude of the symbol taking the value in the set $\{\sqrt{0.2}, 1, \sqrt{1.8}\}$, corresponding to the inner, middle and outer ring. φ is the rotation angle of a QPSK or rotated QPSK constellation, consider, for the inner and outer ring $\varphi = \pi/4$ (denoted as $\varphi_{\text{i,o}}$), and for the middle ring, φ has two potential values $\arctan(1/3)$ and $\arctan(3)$ because the middle ring is composed of two rotated QPSK constellations, and we denote these two values as $\varphi_{\text{m,1}}$ and $\varphi_{\text{m,2}}$, respectively. Now the received symbol becomes $r_k = Ae^{j(m\pi/2+\varphi)}e^{j\theta_k} + n_k$. Raising the received signal to its $4th$ power (Ip and Kahn 2007), we get

$$\begin{aligned} r_k^4 &= (Ae^{j(m\pi/2+\varphi)}e^{j\theta_k} + n_k)^4 \\ &= A^4 e^{j4\varphi}e^{j4\theta_k} + n_k^{'}, \end{aligned} \tag{2.14}$$

where $a_k^4 = A^4e^{j4\varphi}e^{j4\theta_k}$ is the desired term depending on phase noise θ_k, and $n_k^{'} = \sum_{p=1}^{4}\binom{4}{p}(a_ke^{j4\varphi}e^{j4\theta_k})^{4-p}n_k^p$ is the sum of the unwanted cross terms between the transmitted symbols and channel noise.

When we measure the phase of r_k^4, we get

$$\measuredangle\left(r_k^4\right) \approx 4\varphi + 4\theta_k + m_k^{'}, \tag{2.15}$$

where $m_k^{'}$ is the noise generated by the unwanted cross terms $n_k^{'}$. When SNR is high, $m_k^{'}$ can be demonstrated approximately as i.i.d zero-mean Gaussian process (Ip and Kahn 2007), so we can consider (2.15) as an unbiased estimate of $4\varphi + 4\theta_k$.

As we have discussed, φ has three possible values:

$$\begin{aligned} \varphi_{\text{i,o}} &= \frac{\pi}{4}, \\ \varphi_{\text{m,1}} &= \arctan(1/3), \\ \varphi_{\text{m,2}} &= \arctan(3), \end{aligned} \tag{2.16}$$

depending on which ring the symbol lies on, and has been determined in (2.11). Although the ring has been determined, there are still two cases to consider: (i) symbols on the outer or the inner ring; (ii) symbols on the middle ring. In case

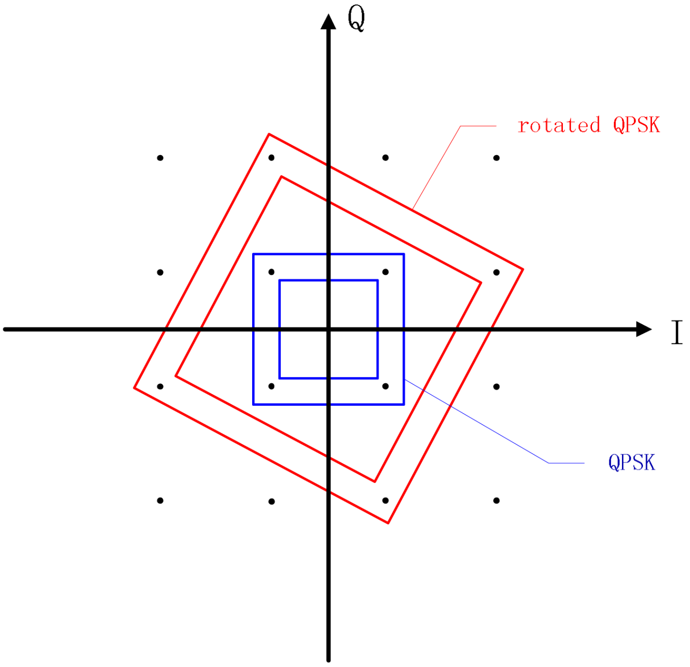


Figure 2.6. Decomposition of 16-QAM constellation

(i), it is very easy because there is only one potential value of φ, so we can simply estimate the phase noise as:

$$\hat{\theta}_k = \frac{1}{4}\left(\measuredangle\left(r_k^4\right) - 4\varphi_{\mathrm{i,o}}\right). \tag{2.17}$$

In case (ii), we obtain two possible phase noise estimates:

$$\begin{aligned}\hat{\theta}_{k,1} &= \frac{1}{4}\left(\measuredangle\left(r_k^4\right) - 4\varphi_{\mathrm{m,1}}\right)\\ \hat{\theta}_{k,2} &= \frac{1}{4}\left(\measuredangle\left(r_k^4\right) - 4\varphi_{\mathrm{m,2}}\right),\end{aligned} \tag{2.18}$$

and we should decide between them by unwrapping them based on $\hat{\theta}_{k-1}$, and choosing the most likely value.

Assuming that small phase noise steps are more likely than large ones, we need to unwrap $\hat{\theta}_{k,1}$ and $\hat{\theta}_{k,2}$ with $\hat{\theta}_{k-1}$ by using the function $\mathrm{U}(\cdot)$. This function means unwrapping the first argument by referencing the second one, and it will be specifically described in the next section. Now introducing $\alpha_k = \mathrm{U}(\hat{\theta}_{k,1}, \hat{\theta}_{k-1})$ and $\beta_k = \mathrm{U}(\hat{\theta}_{k,2}, \hat{\theta}_{k-1})$, we find that the phase noise estimate is:

$$\hat{\theta}_k = \arg\min_{\alpha_k,\beta_k}\left(|\alpha_k - \hat{\theta}_{k-1}|, |\beta_k - \hat{\theta}_{k-1}|\right). \tag{2.19}$$

Phase unwrapping

The objective of phase unwrapping is to remove 'discontinuity' in the raw estimated phase noise sequence (the results we obtain in (2.17) and (2.19)). In order to distin-

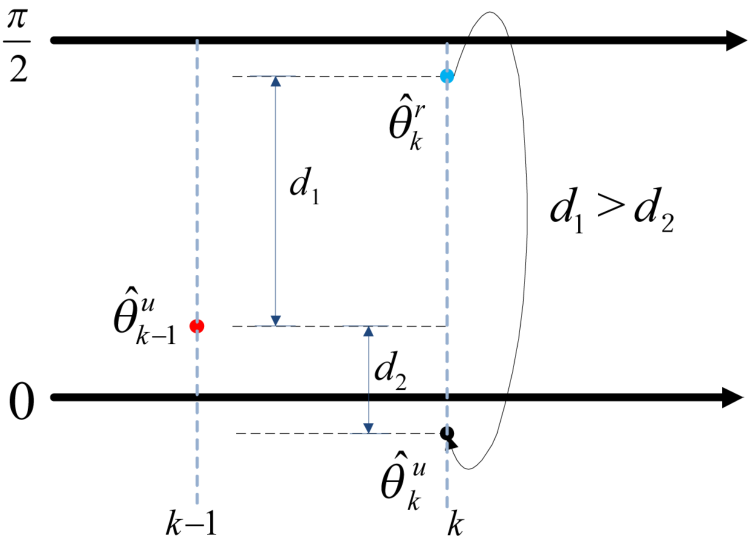


Figure 2.7. Description of Unwrapping function.

guish the raw and unwrapped phase noise estimates, we use the superscripts $\cdot^{\mathrm{r}}$ and $\cdot^{\mathrm{u}}$ to refer them, respectively.

The reason of this 'discontinuity' existing is that in practice the function $\measuredangle(\cdot)$ always returns a value between 0 and 2π, and in programming we limit the results of (2.15), (2.17) and (2.18) in the range of $[0, \pi/2]$.

Assuming that small phase noise step is more likely than large one, we need to find a value nearest to the previous phase noise estimate $\hat{\theta}_{k-1}^{\mathrm{u}}$ among $\hat{\theta}_k^{\mathrm{r}} + m \cdot \pi/2$, where m is an integer ranging from $-\infty$ to ∞, and take this value as the unwrapped phase noise estimate. For example, if we get the results as $\hat{\theta}_{k-1}^{\mathrm{u}} = \pi/10$ and $\hat{\theta}_k^{\mathrm{r}} = 9\pi/10$, we unwrap $\hat{\theta}_k^{\mathrm{r}}$ to $-\pi/10$ instead of keeping it $9\pi/10$, because the phase noise step $-\pi/5$ is more reasonable than $4\pi/5$. In general, the unwrapped phase noise estimate is:

$$\begin{aligned} \hat{\theta}_k^{\mathrm{u}} &= \mathrm{U}(\hat{\theta}_k^{\mathrm{r}}, \hat{\theta}_{k-1}^{u}) \\ &= \arg \min_{\hat{\theta}_k^{\mathrm{r}} + m \cdot \frac{\pi}{2}} \left(|\hat{\theta}_{k-1}^{\mathrm{u}} - (\hat{\theta}_k^{\mathrm{r}} + m \cdot \frac{\pi}{2})| \right), m = -\infty, ..., \infty. \end{aligned} \tag{2.20}$$

An intuitive description of the function is shown in Fig. (2.7), where the red point $(\hat{\theta}_{k-1}^{u})$ serves as the reference, and the blue point $(\hat{\theta}_k^{r})$ is dragged down by $\pi/2$ to the black point $(\hat{\theta}_k^{u})$ because d_1 (phase noise step) is greater than d_2. We apply this algorithm to the whole raw phase noise estimates.

Phase noise filtering

After phase unwrapping, the phase noise estimates can be further improved by low-pass filtering (Ip and Kahn 2007), such as Wiener filter or Kalman filter. However, Phase filtering was beyond the scope of this thesis project.

2.3.3 Polarization combination

On each polarization, we get an independent estimate of the *common* phase noise θ_k (denoted as $\hat{\theta}_k^{(\mathrm{X})}$ and $\hat{\theta}_k^{(\mathrm{Y})}$, and the superscripts $\cdot^{\mathrm{r}}$ and $\cdot^{\mathrm{u}}$ are ignored in this case). The quality of the estimate depends on the ring on which the transmitted symbol lies: on the outer or the inner ring, symbols have at least $\pi/2$ angular separation, while in the middle ring this value is reduced much smaller, so obviously the symbols on the outer or the inner ring are less likely to give rise to 'bad' estimates. To evaluate the quality of the estimates, the variance of the estimate is one of the criteria of performance, which is demonstrated in the next paragraph.

In the observation model, $r_k = a_k e^{j\theta_k} + n_k$, and according to (2.13), using R, the radius of the symbol instead of A we get:

$$\begin{aligned}
r_k^4 &= (Re^{j(m\pi/2+\varphi)}e^{j\theta_k} + n_k)^4 \\
&= R^4 e^{j4\varphi} e^{j4\theta_k} \left(1 + \frac{n_k}{Re^{j(m\pi/2+\varphi)}e^{j\theta_k}}\right)^4 \\
&= R^4 e^{j4\varphi} e^{j4\theta_k} (1 + w_k)^4 \text{ (let } w_k = \frac{n_k}{Re^{j(m\pi/2+\varphi)}e^{j\theta_k}}) \\
&\approx R^4 e^{j4\varphi} e^{j4\theta_k} (1 + 4w_k)
\end{aligned} \tag{2.21}$$

The last approximate equation follows because for medium-to-high SNR, w_k is small and we can neglect the higher order terms. Since n_k is complex normal distributed with zero-mean and variance σ^2 per real dimension, w_k is complex normal distributed with zero-mean and variance σ^2/R^2 per real dimension. Then $\measuredangle(r_k^4) = 4\varphi + 4\theta_k + \measuredangle(1 + 4w_k)$, since 4φ and $4\theta_k$ are constant, the variance of $\measuredangle(r_k^4)$ is equal to the variance of $\measuredangle(1 + 4w_k)$, which is $16\sigma^2/R^2$ per real dimension. Finally, according to (2.20), the variance of $\hat{\theta}_k$ is σ^2/R^2. Now the estimates of symbols on the outer, middle and inner ring have variances $\sigma^2/R_{\mathrm{o}}^2$, $\sigma^2/R_{\mathrm{m}}^2$ and $\sigma^2/R_{\mathrm{i}}^2$, respectively, with ascending values. Although the estimate variance of symbols on the middle ring has smaller value than that of inner ring, we still consider the estimates on the inner ring are more reliable than those on the middle ring because the inner ring has larger angular separation.

Now we can consider the model of phase estimation as:

$$\begin{aligned}
\hat{\theta}_k^{(\mathrm{X})} &= \theta_k + n_x,\ n_x \sim \mathcal{N}(0, \sigma^2/(\hat{R}_k^{(\mathrm{X})})^2) \\
\hat{\theta}_k^{(\mathrm{Y})} &= \theta_k + n_y,\ n_y \sim \mathcal{N}(0, \sigma^2/(\hat{R}_k^{(\mathrm{Y})})^2),
\end{aligned} \tag{2.22}$$

Note that we do not use the superscripts $\cdot^{\mathrm{r}}$ and $\cdot^{\mathrm{u}}$ in this model, neither consider the initial difference between the raw/unwrapped phase noise estimates and the true phase noise θ_k, since our focus is on deducing the mathematical method of combination. Phase unwrapping will be taken into account in the discussion of each case of phase combination that will be explained later.

Given that the two observations are affected by two independent Gaussian noises n_x and n_y, the maximum log-likelihood function of θ_k is:

$$\begin{aligned}\Lambda(\theta_k) &= \ln p(\hat{\theta}_k^{(X)}, \hat{\theta}_k^{(Y)} | \theta_k)\\ &= \frac{1}{2}\ln p(\hat{\theta}_k^{(\mathrm{X})}|\theta_k) + \frac{1}{2}\ln p(\hat{\theta}_k^{(\mathrm{Y})}|\theta_k)\\ &= \frac{1}{2}\left\{\ln\frac{1}{\sqrt{2\pi\sigma^2/(\hat{R}_k^{(\mathrm{X})})^2}} - \frac{(\hat{\theta}_k^{(\mathrm{X})}-\theta_k)^2}{2\sigma^2/(\hat{R}_k^{(\mathrm{X})})^2}\right\}\\ &\quad+\frac{1}{2}\left\{\ln\frac{1}{\sqrt{2\pi\sigma^2/(\hat{R}_k^{(\mathrm{Y})})^2}} - \frac{(\hat{\theta}_k^{(\mathrm{Y})}-\theta_k)^2}{2\sigma^2/(\hat{R}_k^{(\mathrm{Y})})^2}\right\}\\ &= -\ln\sqrt{2\pi} - \ln\sigma + \frac{1}{2}\ln\hat{R}_k^{(\mathrm{X})} + \frac{1}{2}\ln\hat{R}_k^{(\mathrm{Y})}\\ &\quad-\frac{(\hat{R}_k^{(\mathrm{X})})^2}{4\sigma^2}(\hat{\theta}_k^{(\mathrm{X})}-\theta_k)^2 - \frac{(\hat{R}_k^{(\mathrm{Y})})^2}{4\sigma^2}(\hat{\theta}_k^{(\mathrm{Y})}-\theta_k)^2. \end{aligned}\tag{2.23}$$

Taking the derivative of $\Lambda(\theta_k)$ with respect to θ_k, we get:

$$\frac{\partial\Lambda(\theta_k)}{\partial\theta_k} = -\left(\frac{(\hat{R}_k^{(\mathrm{X})})^2}{2\sigma^2} + \frac{(\hat{R}_k^{(\mathrm{Y})})^2}{2\sigma^2}\right)\theta + \frac{(\hat{R}_k^{(\mathrm{X})})^2}{2\sigma^2}\hat{\theta_k}^{(\mathrm{X})} + \frac{(\hat{R}_k^{(\mathrm{Y})})^2}{2\sigma^2}\hat{\theta_k}^{(\mathrm{Y})}, \tag{2.24}$$

set this formula to zero, we obtain the maximum likelihood estimator:

$$\hat{\theta}_{k,mle} = \frac{(\hat{R}_k^{(\mathrm{X})})^2\hat{\theta}_k^{(\mathrm{X})} + (\hat{R}_k^{(\mathrm{Y})})^2\hat{\theta}_k^{(\mathrm{Y})}}{(\hat{R}_k^{(\mathrm{X})})^2 + (\hat{R}_k^{(\mathrm{Y})})^2}. \tag{2.25}$$

We can also look at this problem at different angle and solve it in another way. Our objective is to minimize the estimate variance by linearly combing these two observations, so the combination can be written as:

$$\hat{\theta}_{k,com} = \lambda\hat{\theta}_k^{(\mathrm{X})} + (1-\lambda)\hat{\theta}_k^{(\mathrm{Y})},\ 0 \le \lambda \le 1, \tag{2.26}$$

so the variance of $\hat{\theta}_{k,com}$ (denoted as $V_{k,com}$) is:

$$V_{k,com} = \lambda^2\sigma^2/(\hat{R}_k^{(\mathrm{X})})^2 + (1-\lambda)^2\sigma^2/(\hat{R}_k^{(\mathrm{Y})})^2, \tag{2.27}$$

taking derivative of (2.27) with respect to λ and setting it to zero, we get $\lambda = \frac{(\hat{R}_k^{(\mathrm{X})})^2}{(\hat{R}_k^{(\mathrm{X})})^2+(\hat{R}_k^{(\mathrm{Y})})^2}$, so the optimal combinator is:

$$\hat{\theta}_{k,com} = \frac{(\hat{R}_k^{(\mathrm{X})})^2\hat{\theta}_k^{(\mathrm{X})} + (\hat{R}_k^{(\mathrm{Y})})^2\hat{\theta}_k^{(\mathrm{Y})}}{(\hat{R}_k^{(\mathrm{X})})^2 + (\hat{R}_k^{(\mathrm{Y})})^2}, \tag{2.28}$$

which is identical to $\hat{\theta}_{k,mle}$ in (2.25).

There are four possible cases of polarization combination:

(i) $\hat{R}_k^{(\mathrm{X})} = \hat{R}_k^{(\mathrm{Y})} \in \{R_\mathrm{i}, R_\mathrm{o}\}$;

(ii) $\hat{R}_k^{(\mathrm{X})} = R_\mathrm{i}$ and $\hat{R}_k^{(\mathrm{Y})} = R_\mathrm{o}$ (or vice versa);

(iii) $\hat{R}_k^{(\mathrm{X})} \in \{R_\mathrm{i}, R_\mathrm{o}\}$ and $\hat{R}_k^{(\mathrm{Y})} = R_\mathrm{m}$ (or vice versa);

(iv) $\hat{R}_k^{(\mathrm{X})} = \hat{R}_k^{(\mathrm{Y})} = R_\mathrm{m}$.

Observe that now we add the superscripts $\cdot^\mathrm{r}$ and $\cdot^\mathrm{u}$ to differentiate the raw and unwrapped phase noise estimates.

In cases (i)-(ii), both estimates are considered as reliable, so we can combine them through proper weighting allocation (2.28) and as follows. Letting $\gamma_k = \mathrm{U}(\hat{\theta}_k^{\mathrm{r},(\mathrm{X})}, \hat{\theta}_k^{\mathrm{r},(\mathrm{Y})})$, we can find a joint estimate of θ_k as

$$\hat{\theta}_{k,com}^{\mathrm{r}} = \frac{(\hat{R}_k^{(\mathrm{X})})^2\gamma_k + (\hat{R}_k^{(\mathrm{Y})})^2\hat{\theta}_k^{\mathrm{r},(\mathrm{Y})}}{(\hat{R}_k^{(\mathrm{X})})^2 + (\hat{R}_k^{(\mathrm{Y})})^2}. \tag{2.29}$$

In case (iii), $\hat{\theta}_k^{\mathrm{r},(\mathrm{X})}$ is reliable while $\hat{\theta}_k^{\mathrm{r},(\mathrm{Y})}$ is not, so $\hat{\theta}_k^{\mathrm{r},(\mathrm{X})}$ is a more appropriate reference than $\hat{\theta}_{k-1}^{\mathrm{u}}$ for the polarization Y, so we utilize (2.19) to re-calculate $\hat{\theta}_k^{\mathrm{r},(\mathrm{Y})}$. In this case, we calculate $\alpha_k = \mathrm{U}(\hat{\theta}_{k,1}^{(\mathrm{Y})}, \hat{\theta}_k^{\mathrm{r},(\mathrm{X})})$ and $\beta_k = \mathrm{U}(\hat{\theta}_{k,2}^{(\mathrm{Y})}, \hat{\theta}_k^{\mathrm{r},(\mathrm{X})})$, and the raw phase noise estimate in polarization Y is:

$$\hat{\theta}_k^{\mathrm{r},(\mathrm{Y})} = \arg\min_{\alpha_k,\beta_k} \left(|\alpha_k - \hat{\theta}_k^{\mathrm{r},(\mathrm{X})}|, |\beta_k - \hat{\theta}_k^{\mathrm{r},(\mathrm{X})}| \right). \tag{2.30}$$

Then we apply (2.29) with appropriate weighting to $\hat{\theta}_k^{\mathrm{r},(\mathrm{X})}$ and $\hat{\theta}_k^{\mathrm{r},(\mathrm{Y})}$.

In case (iv), both estimates are on the middle ring and are not reliable, but we can not find another reference more appropriate than $\hat{\theta}_{k-1}^{\mathrm{u}}$, so we do not need to re-calculate any of them, and apply (2.29) with weights 1/2 to both estimates, and γ_k has the same meaning with cases (i) and (ii).

The combined phase noise estimates need to be unwrapped as well, with the same algorithm introduced in the last section.

2.4 Performance measures

In this section we will describe the methods of measuring the performance of our phase noise estimator. According to the criteria we utilize in performance measures, this section will be partitioned into the following two parts:

- Cycle slip detection
- Estimation error variance

Each of them will be particularly explained later in this section.

2.4.1 Cycle slip detection

Cycle slip is a common phenomenon in fiber-optic communication system, it is a change in the operating point that occurs during a measurement. Fig (2.8) illustrates the cycle slips occurring in both polarizations, with channel noise variance $\sigma^2 = 0.002$ and phase noise variance $\sigma_\theta^2 = 10^{-5}$. During normal operation, the difference between the true and estimated phase noise $\theta_k - \hat{\theta}_k^{\mathrm{u}}$ exhibits small variations around a stable operating point (De Jonghe and Moeneclaey 1994), as shown in Fig (2.8) the initial difference $\theta_k - \hat{\theta}_k^{\mathrm{u},(\mathrm{X})} = \theta_k - \hat{\theta}_k^{\mathrm{u},(\mathrm{Y})} = 3\pi/2$. However, due to noise disturbances, a sudden $\pi/2$ error may happen during phase unwrapping, and the following estimates will 'lock' on this wrong unwrapped estimate, therefore, the phase noise difference $\theta_k - \hat{\theta}_k^{\mathrm{u}}$ may slip from the initial stable operating point ($3\pi/2$) to an adjacent stable operating point, say 2π in case of positive slip, or π in case of negative slip (De Jonghe and Moeneclaey 1994). Since the system has an ambiguity up to $\pi/2$, the operating point always slip $\pi/2$ or $-\pi/2$ in phase noise difference, as in Fig (2.8) we can see that the first cycle slip occurs at around the $12000th$ symbol in polarization X, and there are 6 cycle slips in polarization X (the green curve), and 5 cycle slips in polarization Y (the red curve) in total, and each cycle slip leads to a catastrophic $\pi/2$ or $-\pi/2$ phase shift to the estimates. Therefore, to detect cycle slips and quantify their statistics are parts of the tasks in this project.

The probability of cycle slip is an important criterion to evaluate the performance of the SBSPE, so it is required to detect cycle slips accurately when we measure the performance, however, please note that the practical receiver can not detect the cycle slips. Our method is to detect $\pi/2$ or $-\pi/2$ jumps in the sequence of phase noise difference $\theta_k - \hat{\theta}_k^{\mathrm{u}}$, denoted as d (we drop the superscripts $\cdot^{(\mathrm{X})}$ and $\cdot^{(\mathrm{Y})}$ for notational convenience). The particular procedures follow.

We generate a sequence *diff*, whose nth element is the difference between the nth and the $(n + L)th$ elements of d, i.e.

$$\mathit{diff}(n) = d(n + L) - d(n), \tag{2.31}$$

where L is a constant to be decided. There is a trade-off between reliability and performance in deciding the value of L: if L is too small the jump value may not reach $\pi/2$ or $-\pi/2$ in the period of L symbols, so we fail to detect some cycle slips; if the value is too large, we can not determine the occurring positions of the cycle slips accurately, which will affect the performance of following functions, so the decision of L is an important issue in this section. In the simulation we found that one 'jump' (a cycle slip) is always completed during the period of 4 to 5 symbols; after several tests we have found that $L = 10$ is an appropriate choice.

According to the property of sequence d, the elements are valued around 0 in case of no cycle slip; oppositely, the values grow up to around $\pi/2$ or $-\pi/2$ when cycle slips occur, so when we detect 'impulse' absolutely valued $\pi/2$, we treat it as a cycle slip and record the index of this symbol, then we have a sequence (if there are any cycle slips) *index_cs*. However, we detect one cycle slip for several times

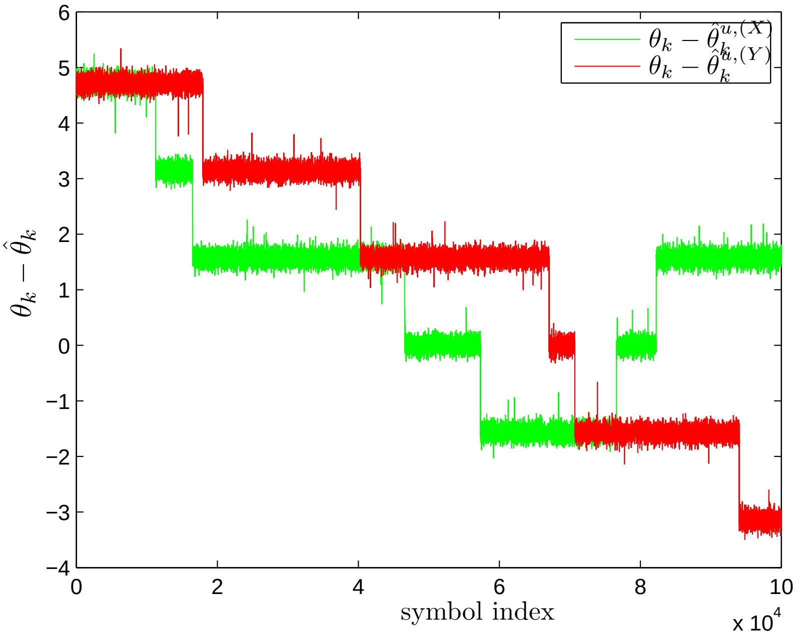


Figure 2.8. Cycle slips in polarization X and Y with $\sigma^2 = 0.002$ and $\sigma_\theta^2 = 10^{-5}$.

because one 'jump' is not completed in one, but 4 to 5 symbols; therefore, to remove these repetitions, we scan the sequence *index_cs* and search for consecutive numbers; finally we only record the first one of each string of consecutive numbers and remove the followings. For example, if a cycle slip occurs at the $12000th$ symbol, we may find impulses absolutely valued around $\pi/2$ at the $12000th$, $12001th$, $12002th$ and $12003th$ symbol in *diff* and record all of them in *index_cs*, but we should only record the first one, 12000, as the index of the cycle slip and remove the others. Now the sequence *index_cs* contains the occurring position of each cycle slip, and the length of this sequence is the times that cycle slip occurs.

16-QAM POLMUX technique from Section 2.3.3 helps reduce the probability of cycle slip in the system, and the outcome will be elaborated in the next chapter.

2.4.2 Estimation error variance

The variance of phase noise estimation error is another criterion of performance. To calculate the estimation error variance we do not take cycle slips into consideration. Instead we calculate the variance of estimation error for the periods between each two contiguous cycle slips separately.

Since in the previous section we have determined the positions of each cycle slip, we can divide the sequence *diff* into several blocks, and measure the variance for each block; finally we average them and obtain the estimation variance.

Similar to cycle slip, 16-QAM POLMUX is also of great assistance to reduce the estimation error variance, and the outcome is to be presented in the next chapter.

3 IMPLEMENTATION

In the last chapter the background, observation model and algorithms of the system were discussed, and in this chapter, the descriptions of simulation will be introduced step by step based on the theoretical solutions given in Section 2.3.

3.1 Algorithms

3.1.1 Transmission channel

The transmission channel in the simulation is assumed to be an Additive White Gaussian Noise (AWGN) channel according to the observation model (2.1). The transmitted symbols on polarizations X and Y are generated as follows:

```
aX = 2*(floor(rand(1,Ns)*M)-(M-1)/2);
aY = 2*(floor(rand(1,Ns)*M)-(M-1)/2);
x = aX+j*aY;  // transmitted symbols on polarization X
bX = 2*(floor(rand(1,Ns)*M)-(M-1)/2);
bY = 2*(floor(rand(1,Ns)*M)-(M-1)/2);
y = bX+j*bY;  // transmitted symbols on polarization Y
x_stats = unique(x(1,:));
E2 = mean(abs(x_stats).^2);
x = x/sqrt(E2);  // normalization
y = y/sqrt(E2);
x_stats = unique(x(1,:));
```

where `Ns` is the length of transmitted symbol sequence, and `M` $= 4$ for 16-QAM. For calculational convenience we normalize the symbols to make sure that the average energy of symbols in the constellation equals 1. The channel noise is simulated as i.i.d zero-mean complex Gaussian noise with variance σ^2 per real dimension by:

```
nx = randn(1,Ns)*sqrt(sigma2)+j*randn(1,Ns)*sqrt(sigma2);
ny = randn(1,Ns)*sqrt(sigma2)+j*randn(1,Ns)*sqrt(sigma2);
```

and the phase noise sequence is generated according to (2.2):

```
theta(1)=rand(1)*2*pi;
for k=2:Ns
    theta(k)=2*pi*deltaf+theta(k-1)+randn(1)*sqrt(sigma2_theta);
end;
```

where sigma2_theta is the variance of phase noise.

Sent over the AWGN channel, the received symbols are simulated as shown in model (2.1), where the initial phase offset is generated as a random number between 0 and $\pi/2$:

```
phi = rand(1)*pi/2; // phase offset
rX = x.*exp(j*(theta+phi))+nx;
rY = y.*exp(j*theta)+ny;
```

3.1.2 Initial phase offset

Estimating the initial phase offset is a significant issue for decoding the received sequence, and the method has been explained in Section 2.3.1. In our simulation, this algorithm is applied with the block size as 50, and the outcome is satisfactory. The code is shown as follows:

```
Ms=50; // block size
a=reshape(rX,Ms,Ns/Ms);
b=reshape(rY,Ms,Ns/Ms);
pd=mod(0.25*phase(sum(a.^4))-0.25*phase(sum(b.^4)),pi/2);
D=zeros(2,3);
for n=2:length(pd)  // unwrapping
     D(1,:)=ones(1,3)*pd(1);
     D(2,:)=[pd(n),pd(n)+pi/2,pd(n)-pi/2];
     d=abs(D(1,:)-D(2,:));
     [dum,index]=min(d);
     pd(n)=D(2,index);
end
phi_hat=mean(pd); // phase offset estimate
```

After being estimated, the initial phase offset can be compensated as: $\hat{r}_k^{(\mathrm{X})} = r_k^{(\mathrm{X})} e^{-j\hat{\phi}}$ using:

```
rX = rX*exp(-j*phi_hat);
```

3.1.3 Raw estimation

Raw estimation is the most important part in SBSPE, and is implemented by function estimatePhaseSymbolBySymbol(r, x_stat, sigma2) in the simulation, where the argument r is the received symbol sequence, x_stat is a 1×16 vector composed of the coordinates of the symbols in the 16-QAM constellation (origin as the centroid), and sigma2 is the AWGN noise variance. This function calls another two functions: determineRotations and estimatePhase.

Function determineRotations(x_stats, rho, sigma2) is designed to obtain the mean value as variable m, the variance as s2 of each ring by using (2.6) and (2.7), and the three angles in (2.16) as theta_corrA, theta_corrC1 and theta_corrC2, respectively.

```
for k=1:3
    index=find(abs(x_stats).^2==rho(k));  // index of each ring
    m(k)=2*sigma2+rho(k);  // mean
    s2(k)=4*sigma2*sigma2+4*sigma2*rho(k);  // variance
    phase(x_stats(index).^4);
    if (k==1 || k==3)
        // rotation angle of the inner ring and outter ring
        theta_corrA=phase(x_stats(index(1)).^4);
    else
        // rotation angles of the middle ring
        z=phase(x_stats(index).^4);
        zz=unique(z);
        theta_corrC1=zz(1);
        theta_corrC2=zz(2);
    end;
end;
```

Observe that the variables theta_corrA, theta_corrC1 and theta_corrC2 are 4 times of each rotation angle in (2.16). The argument rho is an 1×3 vector containing the square of amplitudes of the symbols in the 16-QAM constellation.

estimatePhase(r, theta_corrA, theta_corrC1, theta_corrC2, theta_prev, m, s2) is a function serving to estimate the phase noise for each symbol, where the arguments m and s2 have the same meanings with those in determineRotations. Equation (2.11) is taken as a likelihood estimator to determine on which ring the symbol is transmitted:

```
Prior=[0.25 0.5 0.25]; // probability distribution of three rings
for l=1:3
   // likelihood function
   L(l)=-1/(2*s2(l))*(abs(r)^2-m(l))^2-0.5*log(s2(l))+log(Prior(l));
end;
[dum,index]=max(L); // ring index in {1,2,3}
```

After on which ring has been determined, we apply (2.17), (2.18) and (2.19) to estimate the phase noise:

```
// case 1: inner ring or outer ring
if (index==1 || index==3)
   theta_hat=mod(0.25*(phase(r^4)-theta_corrA),pi/2);
else
// case 2: middle ring, we need to distinguish between two
// sub-constellations
try1=mod(0.25*(phase(r^4)-theta_corrC1),pi/2);
try2=mod(0.25*(phase(r^4)-theta_corrC2),pi/2);
  // to find estimate closest to theta_hat(k-1)
   D=zeros(1,6);
   D(1)=abs(try1-theta_prev);
   D(2)=abs(try2-theta_prev);
   D(3)=abs(try1-theta_prev-pi/2);
   D(4)=abs(try2-theta_prev-pi/2);
   D(5)=abs(try1-theta_prev+pi/2);
   D(6)=abs(try2-theta_prev+pi/2);
   [dum,index2]=min(D);
   if (mod(index2,2)==1)
        theta_hat=try1;
   else
        theta_hat=try2;
   end;
end;
```

The comparisons between true phase noise and raw estimates on both polarizations are shown in Fig. (3.1), from which we can tell that the raw estimates do not follow the true phase noise, rather, they jump around in the whole plane, so the raw estimates can not be used directly.

3.1.4 Phase unwrapping

The significance of phase unwrapping has been elaborated in previous parts, and the algorithm has been illustrated in Fig. (2.7), so this section is mainly about the simulation of phase unwrapping. Take polarization X as example:

```
theta_unwrap_x=zeros(1,Ns);
theta_unwrap_x(1)=theta_hat_x(1);
for k=2:Ns
        aa=(theta_unwrap_x(k-1)-theta_hat_x(k))/(pi/2);
        m1=floor(aa);
        m2=ceil(aa);
        t1=theta_hat_x(k)+m1*pi/2;
        t2=theta_hat_x(k)+m2*pi/2;
        // find the one nearest to the previous one
```

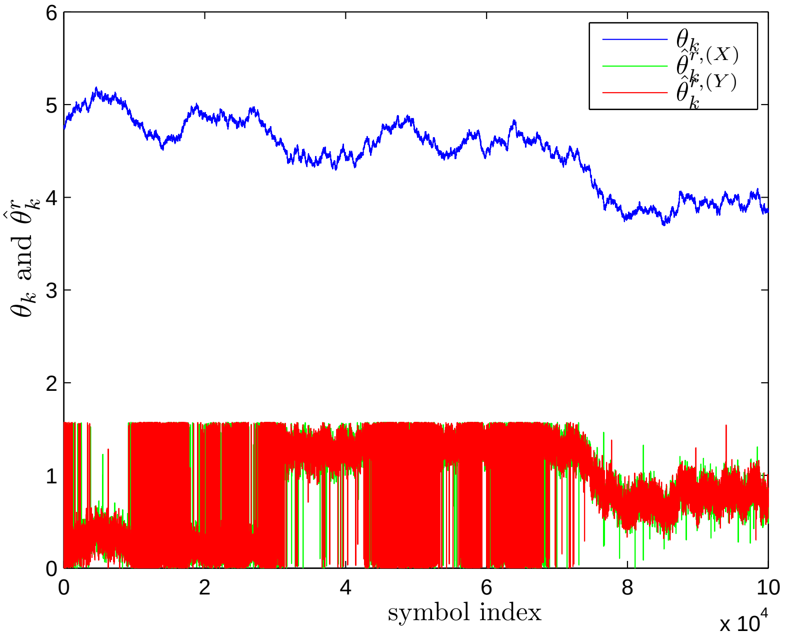


Figure 3.1. Phase noise and raw estimates

```
        if (abs(t1-theta_unwrap_x(k-1))<abs(t2-theta_unwrap_x(k-1)))
            theta_unwrap_x(k)=t1;
        else
            theta_unwrap_x(k)=t2;
        end;
end;
```

After being unwrapped, the estimates are shown in Fig. (3.2). In this figure, the estimates basically follow the phase noise, except for several cycle slips and the initial phase difference, which is inevitable in estimation and could be compensated by further functions.

3.1.5 Phase combination

Applying SBSPE to both polarizations, we have two independent estimates for the common phase noise, and we can combine them to obtain a better estimate. The procedures of phase combination were described in Section 2.3.3, while we realize it in the function `estimatePhaseSymbolBySymbolCombination` in our simulation. The function has six arguments: `rX` and `rY` are the received symbols on polarization X and Y, respectively; `x_stats` and `sigma2` have the same meaning with those in the previous functions; `theta_hat_x` and `theta_hat_y` are raw estimates on X and Y, also the results we obtain in Section 3.1.3. Similar to the function `estimatePhaseSymbolBySymbol`, this function also calls for another two functions:

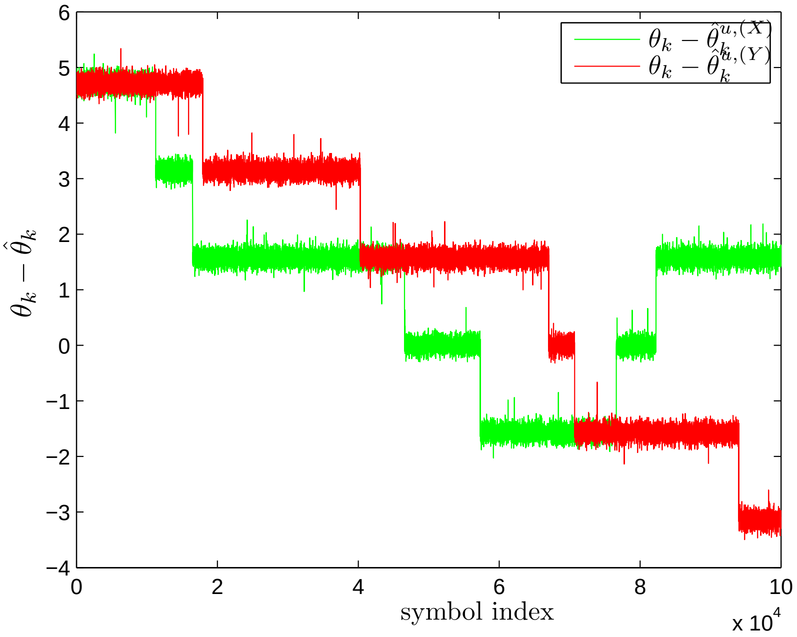


Figure 3.2. Cycle slips in polarization X and Y with $\sigma^2 = 0.002$ and $\sigma_\theta^2 = 10^{-5}$.

`determineRotations` and `estimatePhaseCombination`. The first one has been introduced in Section 3.1.3, and latter one is the focus of this section.

The function `estimatePhaseCombination` has nine arguments: `rX`, `rY`, `m`, `s2`, `rho`, `theta_corrC1`, `theta_corrC2`, `theta_hat_x`, `theta_hat_y`. The weights of two estimates are determined by:

```
// index=1,2,3 indicating inner, middle and outer ring
weight_x=rho(index_x)/(rho(index_x)+rho(index_y));
weight_y=rho(index_y)/(rho(index_x)+rho(index_y));
```

According to the four cases in Section 2.3.3, if both are or neither is on the middle ring, we simply combine the two independent estimates by using another function called `PhaseCombination`:

```
function theta_com=PhaseCombination(theta1,theta2,weight1,weight2)
d=zeros(1,3);
// unwrapping theta1 to theta2
d(1)=abs(theta1-theta2);
d(2)=abs(theta1-theta2-pi/2);
d(3)=abs(theta1-theta2+pi/2);
[dmin,index_min]=min(d);
if index_min==1
    theta_com=mod(weight1*theta1+weight2*theta2,pi/2);
```

```
elseif index_min==2
    theta_com=mod(weight1*theta1+weight2*(theta2+pi/2),pi/2);
else
    theta_com=mod(weight1*theta1+weight2*(theta2-pi/2),pi/2);
end;
```

If only one of the two estimates is on the middle ring, we have to re-estimate the symbol which is previously measured to be on the middle ring by taking the other estimate as reference instead of the previous estimate.

```
elseif (index_x~=2 && index_y==2)
    try_y1=mod(0.25*(phase(rY^4)-theta_corrC1),pi/2);
    try_y2=mod(0.25*(phase(rY^4)-theta_corrC2),pi/2);
    // find the estimation of y closest to x
    T=zeros(2,6);
    T(1,:)=ones(1,6)*theta_hat_x;
    T(2,:)=[try_y1,try_y1-pi/2,try_y1+pi/2,try_y2,try_y2-pi/2,try_y2+pi/2];
    D=zeros(1,6);
    D=abs(T(1,:)-T(2,:));
    [dum,index2]=min(D);
    theta_hat_com=mod(weight_x*T(1,index2)+weight_y*T(2,index2),pi/2);

elseif (index_x==2 && index_y~=2)
    try_x1=mod(0.25*(phase(rX^4)-theta_corrC1),pi/2);
    try_x2=mod(0.25*(phase(rX^4)-theta_corrC2),pi/2);
    // find the estimation of x closest to y
    T=zeros(2,6);
    T(1,:)=[try_x1,try_x1-pi/2,try_x1+pi/2,try_x2,try_x2-pi/2,try_x2+pi/2];
    T(2,:)=ones(1,6)*theta_hat_y;
    D=zeros(1,6);
    D=abs(T(1,:)-T(2,:));
    [dum,index2]=min(D);
    theta_hat_com=mod((weight_x*T(1,index2)+weight_y*T(2,index2)),pi/2);
```

The outcome of phase combination (after being unwrapped) is shown in Fig. (3.3). Drastic reduction in the number of cycle slips can be seen in the combined phase noise estimates (only 1 cycle slip) compared to single polarization (6 in polarization X and 5 in Y); moreover, the estimation variance also decreases to a lower level compared to those of single polarization. Although we do not have the exact values of the estimation variances in this phase, we still hold an intuitive judgment that the curve of the combination has small varying range than the other two, because the curve is 'narrower' than the others.

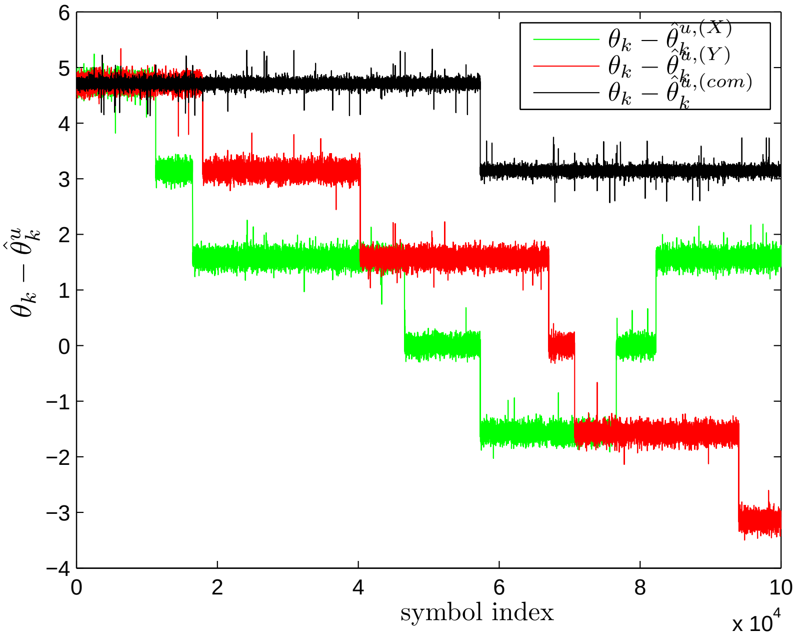


Figure 3.3. True and estimated phase noise

3.2 Performance measures

3.2.1 Cycle slip detection

Cycle slip is a significant issue in coherent optic communication, so the function `findCycleSlip`(`theta`, `theta_unwrap`, L) is designed to detect cycle slips. It has three arguments: `theta`, `theta_unwrap` and L, and returns two variables: integer `cnt`, the number of cycle slips in `theta_unwrap`, and vector `index_cs`, containing the occurring position of each cycle slip.

```
d=theta-theta_unwrap;
diff=zeros(1,length(d)-L);
for p=1:length(diff)
    diff(p)=d(L+p)-d(p);
end
index=find(abs(diff)>(pi/2-pi/8));
index_cs=[];
cnt=1-isempty(index);
if isempty(index)==0
    index_cs=[index(1)];
    for q=1:length(index)-1
        if index(q)+1<index(q+1)
            cnt=cnt+1;
            index_cs=[index_cs index(q+1)];
        else
        end
    end
end
```

When we run the code:

```
L=6; // detection step
[cnt_x,index_cs_x]=findCycleSlip(theta,theta_unwrap_x,L);
[cnt_y,index_cs_y]=findCycleSlip(theta,theta_unwrap_y,L);
[cnt_com,index_cs_com]=findCycleSlip(theta,theta_unwrap_com,L);
```

we get the results in the command window as:

```
cnt_x =     6
index_cs_x =   [11246    16487    46578    57316    76639    82252 ]

cnt_y =     5
index_cs_y =   [17934    40286    67048     70706    94025]

cnt_com =     1
index_cs_com =     57314
```

The numerical results accord to those in Fig. (3.3), which means that the function detects all the cycle slips, and exactly positions the occurring index of each one, paving a way for further actions and functions in this project.

3.2.2 Estimation error variance

Estimation error variance is a criterion to evaluate the quality of an estimate. An intuitive judgment is the linewidths of the curves in Fig. (3.3), where we can reach the decision that the black curve (combination) has less estimation error variance than the green and red ones (one polarization), however, using a numerical value is a more direct and convincing way to represent the estimation error variance, and provide an uniform standard for it.

To calculate the exact value of estimation error variance, it is unfair to take cycle slips into account, because a cycle slip leads to a $\pi/2$ or $-\pi/2$ jump and will make the variance very large, so we divide the sequence $\texttt{difflevel} = \texttt{theta} - \texttt{theta\_unwrap}$ into the number of $\texttt{cnt} + 1$ blocks by cutting it at the points whose indices are in the vector `index_cs`, and measure the variance of each block, then take the average value of them as the estimation error variance. Taking polarization X as example:

```
// case1: no cycle slip
if cnt_x==0
   EstimateVar_x(z)=var(difflevel_x);
// case2: non-zero cycle slips
else
   var_x=zeros(1,cnt_x+1);
   var_x(1)=var(difflevel_x(1:index_cs_x(1)));
   for m=1:cnt_x-1
   // kicking off the unstable period
      if index_cs_x(m)+L<index_cs_x(m+1)
         var_x(m+1)=var(difflevel_x(index_cs_x(m)+L:index_cs_x(m+1)));
      else
         var_x(m+1)=var(difflevel_x(index_cs_x(m):index_cs_x(m+1)));
      end;
    end;
    var_x(cnt_x+1)=var(difflevel_x(index_cs_x(cnt_x)+L:end));
    // averaging
    EstimateVar_x(z)=mean(var_x);
end
```

Running this piece of code, we can get the results in the command window:

```
EstimateVar_x =    0.0039

EstimateVar_y =    0.0039

EstimateVar_com =  7.4822e-004
```

which accord to the intuitive observation in Fig. (3.3): the two polarizations have the values in the same order, while the combination has a value reduced by a factor of 5.

4 DISCUSSION

In the previous two chapters, we gave a description of the models and algorithms, and represented the simulation results; in this chapter we will further discuss regarding ring discrimination, cycle slip, estimation error variance and symbol error rate of this system.

4.1 Ring error probability

Ring discrimination is a critical issue that directly determines whether the estimate is correct or not, so ring error probability is an important indicator for evaluating the quality of estimates.

Referring to Fig. 2.4, ring error mainly occurs between the outer and the middle ring, so we approximately consider the ring error probability as the probability of mistaking the outer and middle ring. In this section we use $p(R_\mathrm{o}|R_\mathrm{m})$ to indicate the probability that a symbol locating on the middle ring is estimated on the outer ring, and vice-versa.

The Maximum a posteriori estimator of the ring is given in (2.10), where the distribution of R is currently altered to $p(R_\mathrm{m}) = 2/3, p(R_\mathrm{o}) = 1/3$ in our approximation. Hence, the solution $\rho = \zeta$ to the equation

$$\frac{1}{3}\frac{1}{\sqrt{2\pi}V_{R_\mathrm{o},\sigma^2}}e^{-\frac{(\rho-\mu_{R_\mathrm{o},\sigma^2})^2}{2V_{R_\mathrm{o},\sigma^2}}} = \frac{2}{3}\frac{1}{\sqrt{2\pi}V_{R_\mathrm{m},\sigma^2}}e^{-\frac{(\rho-\mu_{R_\mathrm{m},\sigma^2})^2}{2V_{R_\mathrm{m},\sigma^2}}} \tag{4.1}$$

is the threshold value to distinguish R_m and R_o, which means that the decision will be R_o if ρ is greater than ζ, while the decision will be R_m if ρ is less than ζ.

Now the ring error probability can be calculated as:

$$\begin{aligned} p_{e,ring} &= p(R_\mathrm{m}) \cdot p(R_\mathrm{o}|R_\mathrm{m}) + p(R_\mathrm{o}) \cdot p(R_\mathrm{m}|R_\mathrm{o}) \\ &= 2/3p(R_\mathrm{o}|R_\mathrm{m}) + 1/3p(R_\mathrm{m}|R_\mathrm{o}) \\ &= 2/3Q\left(\frac{\zeta - \mu_{R_\mathrm{m},\sigma^2}}{\sqrt{V_{R_\mathrm{m},\sigma^2}}}\right) + 1/3Q\left(-\frac{\zeta - \mu_{R_\mathrm{o},\sigma^2}}{\sqrt{V_{R_\mathrm{o},\sigma^2}}}\right) \end{aligned} \tag{4.2}$$

We calculate the solution to (4.1) and (4.2) numerically by using Monte Carlo simulation method, and get the result plotted in Fig. 4.1 In this simulation we can conclude that the ring error probability goes down along with the increment of

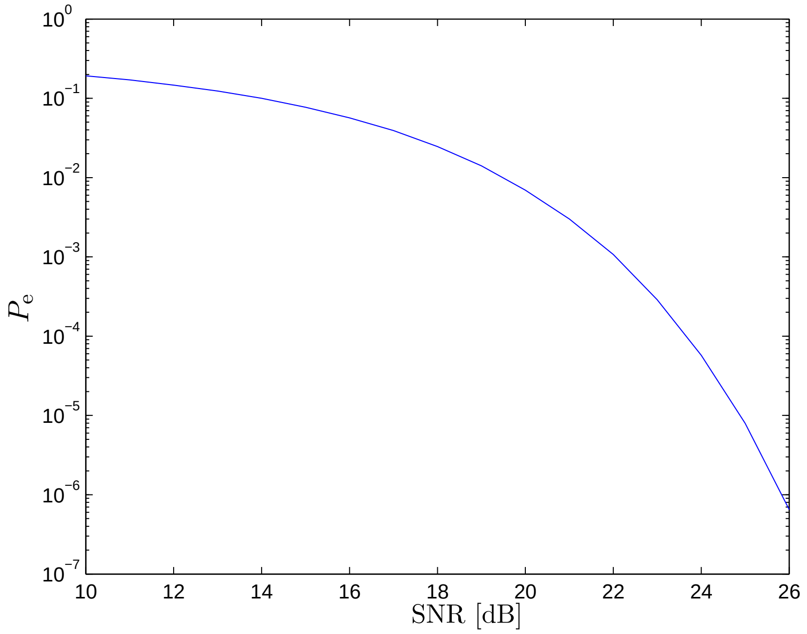


Figure 4.1. Error probability of the ring detector from (2.11).

SNR. Furthermore, relatively large SNR is required to keep the ring error probability low, for instance, it is over 10^{-2} around 19dB that is required for a BER of 10^{-3}. For the parameters we used in the simulation in the previous chapters, SNR approximately equals to 24dB, and the corresponding ring error probability equals to 6×10^{-5}.

4.2 Cycle slip statistics

We also do a Monte Carlo simulation of considering the probability of cycle slips as a function of σ_θ^2 in the system, while the results is illustrated in Fig. 4.2. In this figure we can conclude that the higher the SNR, the lower the probability of cycle slips, and cycle slips occur more often for $\sigma_\theta^2 > 10^{-3}$. For 100Gb/s communication, in each polarization the baud rate is 12.5Gbaud/s, so the corresponding laser linewidth $\Delta_f = \sigma_\theta^2/2\pi/T = 2MHz$, where $1/T$ is the baud rate. Moreover, the probability of cycle slips is reduced by approximately an order of magnitude by combining the information from both polarizations.

4.3 Estimation error variance statistics

In this section, we consider the estimation error variance as a function of σ_θ^2 for separately and jointly estimation, whose results have been plotted in Fig. 4.3, where

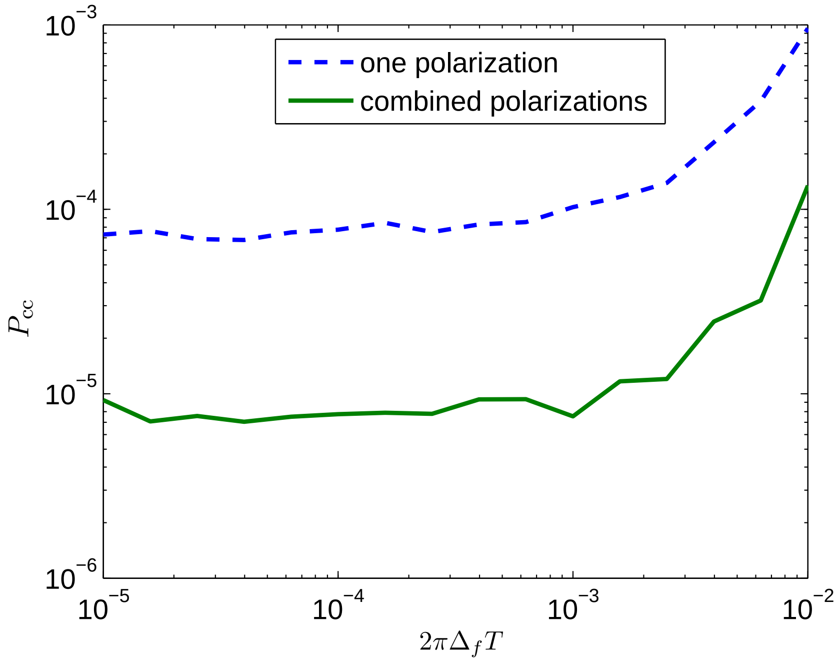


Figure 4.2. Probability of cycle slips, using one and two polarizations.

we can observe that the estimation error variance of the combined polarizations is always lower than that of a single polarization; furthermore, the estimation error variance is almost independent of σ_θ^2 up to $\sigma_\theta^2 > 10^{-3}$, above which the value increases rapidly.

4.4 Comparison with existing algorithms

Besides the SBSPE estimator, we realized two other types of estimators as well, and plotted all the performance in Fig. 4.4, they are estimator 2.5 for $M = 64$, and the estimator from (Seimetz 2008) with $M = 64$. These two estimators give a performance of $V_{ss} > 10^{-1}$, a relatively large value, for $\sigma_\theta^2 > 1.5 \times 10^{-3}$, thus we truncate the curve at $V_{ss} = 10^{-1}$ and $\sigma_\theta^2 = 10^{-2}$. In this figure, the estimator fusing information from both polarizations gives a performance as fairly independent of σ_θ^2 up to 10^{-2} and can tolerate orders of magnitude more laser linewidth than other estimators.

4.5 Symbol error rate

Upon the completion of phase noise estimation, we can de-rotate the received symbols and compensate for the phase noise: the original transmitted symbols can be

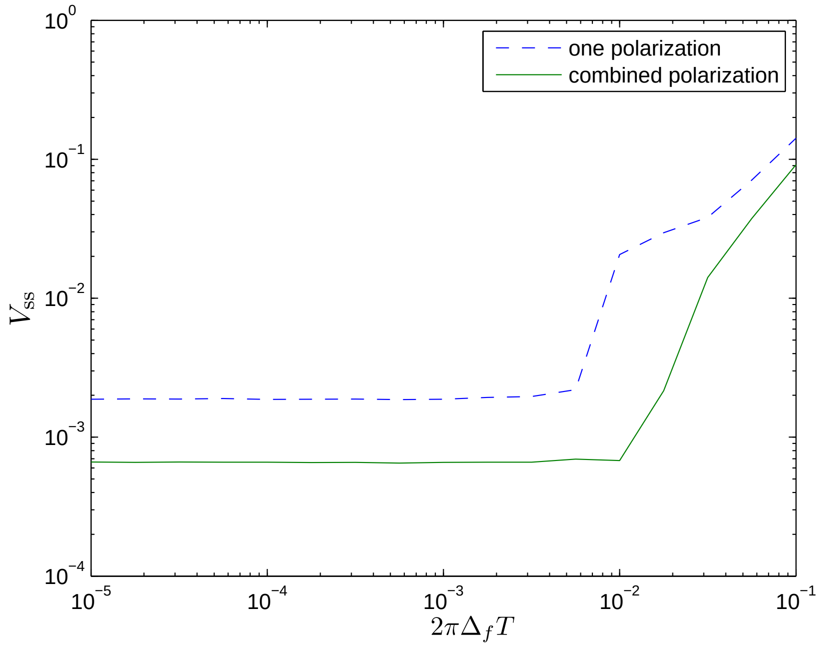


Figure 4.3. Steady-state error variance for different laser linewidths, using one and two polarizations.

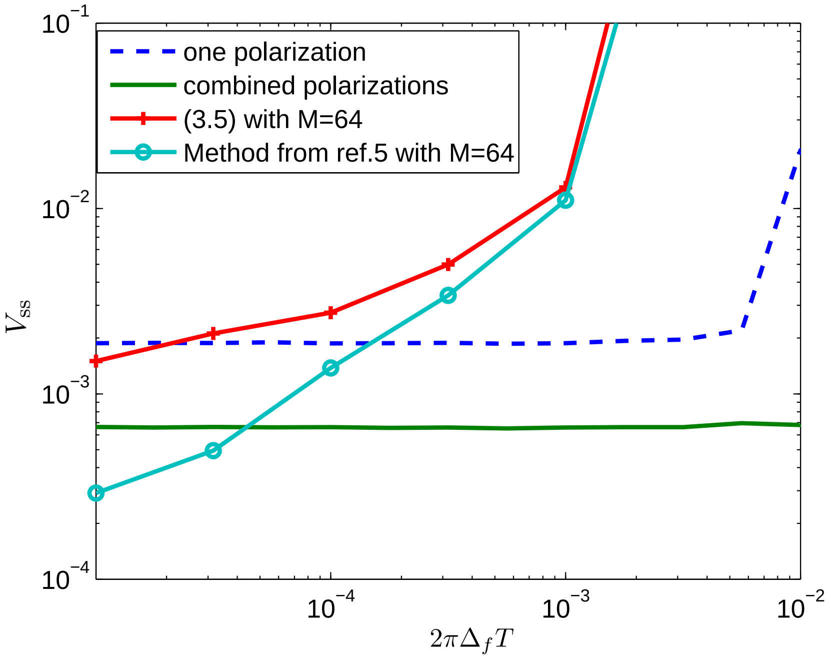


Figure 4.4. Steady-state error variance for different laser linewidths

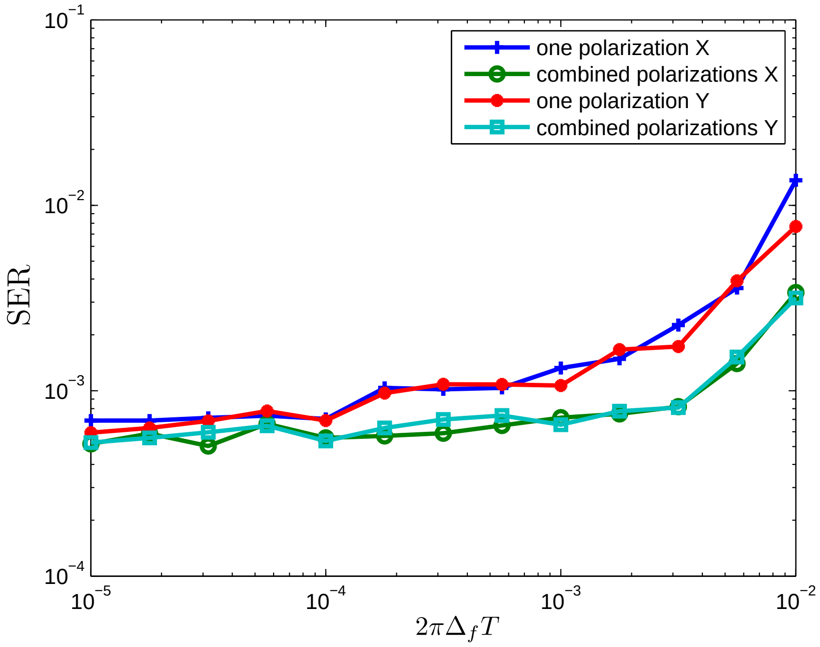


Figure 4.5. Symbol error rate, decoded by one and two polarizations

recovered as:

$$\hat{\mathbf{a}}_k^{'} = \begin{bmatrix} e^{-j\hat{\phi}-j\hat{\theta}_k} & 0 \\ 0 & e^{-j\hat{\theta}_k} \end{bmatrix} \cdot \mathbf{r}_k, \tag{4.3}$$

now we have rough estimates of transmitted symbols in $\hat{\mathbf{a}}_k^{'}$.

Since the complex numbers in $\hat{\mathbf{a}}_k^{'}$ are not necessarily the points in the 16-QAM constellation, we have to pass $\hat{\mathbf{a}}_k^{'}$ through a decision circuit to finally recover the original transmitted symbols. After the decision, we get $\hat{\mathbf{a}}_k = \begin{bmatrix} \hat{a}_k^{(X)} & \hat{a}_k^{(Y)} \end{bmatrix}^T$. Comparing $\hat{\mathbf{a}}_k$ to $\mathbf{a}_k$, the symbol error rate of the system is obtained. We considered the SER as a function of σ_θ^2, and plotted the result of this Monte Carlo simulation in Fig. 4.5. In this figure, the SER of using the phase noise estimates from one polarization grows up to a relatively large value, say around 10^{-2} for $\sigma_\theta^2 = 10^{-2}$, while it keeps 3×10^{-3} for that from both polarizations at the same level of σ_θ^2.

5 CONCLUSION

5.1 Completed work

We have considered feed-forward phase noise estimation for 16-QAM in coherent fiber-optic communication in this project, and a novel symbol-by-symbol phase noise estimation algorithm is introduced and analyzed. A brief description of the algorithm is listed as follows:

1. Phase offset estimation - to estimate the constant phase offset between two polarizations X and Y.
2. Symbol-by-symbol phase noise estimation - the primary part of the whole algorithm, to estimate the phase noise symbol-by-symbol, including three steps:
   (a) Ring discrimination - to determine on which ring of the 16-QAM constellation the symbol is being transmitted;
   (b) Phase estimation - to estimate the phase noise based on the information from ring discrimination;
   (c) Phase unwrapping - to remove the ’discontinuity’ component in the raw estimates.
3. Polarization combination - to fuse the information collected from both polarizations and achieve better performance.

To analyze the performance of the proposed algorithm, we have considered the following performance measures:

1. Cycle slip detection - to detect and quantify the statistics of cycle slips which lead to catastrophic $\pi/2$ phase shift in estimates.
2. Estimation error variance - to estimate the steady-state variance in between cycle slips to characterize the performance of the SBSPE algorithm.

In addition, some further issues are discussed as well:

1. Ring error probability - to analyze the the probability of mistaking on which ring the symbol is transmitted, which is a critical reason for wrong phase estimates.
2. Symbol error rate - to quantify the symbol error rate of this algorithm.

3. Cycle slip statistics - to analyze the probability of cycle slips as a function of the variance of phase noise.

4. Estimation error variance statistics - to analyze the estimation error variance as a function of the variance of phase noise.

5. Comparison with existing references - to compare the steady-state error variance for different laser linewidths of the SBSPE (with one and two polarizations) with other two estimators.

5.2 Findings

Based on the performance of the SBSPE algorithm and the results of comparison with other estimators, it is corroborated that (i) information fusion enhances the performance of the system in two characteristics, the probability of cycle slips and the estimation error variance; (ii) the SBSPE can tolerate orders of magnitude more laser linewidth than the other two estimators from (2.5) and (Seimetz 2008).

5.3 Future work

The algorithms have been implemented in this project, nevertheless, there are still improvements can be done. After the raw phase noise estimates have been unwrapped, filters - Kalman filter or Wiener filter can be used to low-pass filter the estimates, on the purpose of reducing the variance of estimation error.

Bibliography

De Jonghe, G. and Moeneclaey, M. (1994). Cycle slip analysis of the nda ff carrier synchronizer based on the viterbi amp; viterbi algorithm. In: *Communications, 1994. ICC '94, SUPERCOMM/ICC '94, Conference Record, 'Serving Humanity Through Communications.' IEEE International Conference on.* vol. 2. pp 880 –884.

Fatadin, I., Ives, D. and Savory, S. (2010). Laser linewidth tolerance for 16-qam coherent optical systems using qpsk partitioning. *Photonics Technology Letters, IEEE*, **22**(9), 631 –633.

Ip, E. and Kahn, J. (2007). Feedforward carrier recovery for coherent optical communications. *Lightwave Technology, Journal of*, **25**(9), 2675 –2692.

Kahn, Joseph M. and Ip, Ezra (2009). Principles of digital coherent receivers for optical communications. In: *Optical Fiber Communication - incudes post deadline papers, 2009. OFC 2009. Conference on.* pp 1 –48.

Louchet, H., Kuzmin, K. and Richter, A. (2008). Improved dsp algorithms for coherent 16-qam transmission. In: *Optical Communication, 2008. ECOC 2008. 34th European Conference on.* pp 1 –2.

Pfau, T., Hoffmann, S. and Noe, R. (2009). Hardware-efficient coherent digital receiver concept with feedforward carrier recovery for m -qam constellations. *Lightwave Technology, Journal of*, **27**(8), 989 –999.

Seimetz, M. (2008). Laser linewidth limitations for optical systems with high-order modulation employing feed forward digital carrier phase estimation. In: *Optical Fiber communication/National Fiber Optic Engineers Conference, 2008. OFC/NFOEC 2008. Conference on.* pp 1 –3.