

ABSTRACT

NONLINEAR ELECTRICAL COMPENSATION FOR THE COHERENT OPTICAL OFDM SYSTEM

by Jie Pan

One major drawback of a Coherent-Optical Orthogonal Frequency Division Multiplexing (CO-OFDM) system is its sensitivity to fiber nonlinearity. It has been shown that a Volterra series based nonlinear equalizer can compensate distortion introduced by the fiber nonlinearity. One major disadvantage of Volterra series is its complexity which can be reduced by removing its “unimportant” coefficients. The resulting Volterra system is called the sparse Volterra system. The Wiener-Hammerstein model is another popular nonlinear signal processing technique with a simpler structure. In this thesis, the nonlinear distortion of a CO-OFDM system is investigated, and equalizers based on Volterra model, sparse Volterra model and Wiener-Hammerstein model are designed. This is the first attempt to apply the Wiener-Hammerstein model to compensate nonlinear distortion in optical communication systems. A comparison between predistorters and equalizers is also presented. To the best of author’s knowledge, no such a study has ever been conducted for optical communication systems.

NONLINEAR ELECTRICAL COMPENSATION FOR THE COHERENT OPTICAL OFDM SYSTEM

A Thesis

Submitted to the

Faculty of Miami University

in partial fulfillment of

the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

by

Jie Pan

Miami University

Oxford, Ohio

2010

Advisor _____
Dr. Chi-Hao Cheng

Reader _____
Dr. Dmitriy Garmatyuk

Reader _____
Dr. Kuang-Yi Wu

TABLE OF CONTENTS

Chapter 1. Introduction	1
Chapter 2. Fiber Optics and Optical CO-OFDM	5
2.1 Fiber Optics	5
2.2 Principles of OFDM	7
2.3 Optical Detection Techniques	8
2.4 Fiber Nonlinearities	11
Chapter 3. Modeling Techniques	14
3.1 Linear Model	14
3.2 Volterra Model	14
3.3 Sparse Volterra Model	15
3.4 Pth-order Inverse of Volterra Series	19
3.5 Wiener-Hammerstein model	21
Chapter 4. Adaptive Algorithms	23
4.1 Least Mean Square (LMS) Algorithm	23
4.2 Recursive Least Square (RLS) Algorithm	25
Chapter 5. Channel Estimation and Compensators Design	30
5.1 Compensator Design Based on Transmission Channel Model	30
5.1.1 Compensator Design Based On the Linear Inverse Model	31
5.1.2 Compensator Design based on pth-order Volterra Inverse Model	33
5.2 Adaptive Equalizer Design Based on Inverse Channel Estimation	37
Chapter 6.Simulation System	42
Chapter 7.Results and Discussions	47
7.1 Equalization by Full Volterra and Sparse Volterra System	47
7.2 Equalization by <i>pth-order</i> Volterra Inverse System	54
7.3 Compensation by <i>pth-order</i> Volterra Inverse Equalizer and Predistorter	58
7.4 Equalization by Wiener-Hammerstein Equalizer	62
Chapter 8.Conclusions	66
Reference	68
Appendix	72

TABLE CAPTIONS

Table 1. LMS algorithm summary for real signal.....	25
Table 2 LMS algorithm summary for complex signal.....	25
Table 3 RLS algorithm summary.....	29
Table 4 <i>Pth-order</i> inverse summary for the third order Volterra series.	36
Table 5 Predistorter design based on the <i>pth-order</i> pre-inverse Volterra system.....	37
Table 6 LMS for the Wiener-Hammerstein model coefficients update.....	41

FIGURE CAPTIONS

Figure 2.1 Components of an optical communication system.	5
Figure 2.2 Block diagram for WDM system.	6
Figure 2.3 Coherent detection block diagram.	9
Figure 2.4 Direct detection block diagram.	11
Figure 3.1 Nonlinear system H connected with its inverse system K (post inverse system).	20
Figure 3.2 Wiener-Hammerstein system for channel modeling.	21
Figure 5.1 A nonlinear optical channel model.	30
Figure 5.2 Third order Volterra model for optical nonlinear channel.	33
Figure 5.3 Channel estimation process (training mode).	35
Figure 5.4 Tandem connection of the p th-order pre-inverse K and the nonlinear system H	36
Figure 5.5 Inverse system determination process (training mode).	38
Figure 5.6 Equalizer compensation mode.	38
Figure 6.1 Block diagram for numerical simulation of CO-OFDM system.	42
Figure 6.2 16-QAM encoders with Gray coding.	42
Figure 6.3 I/Q channel RF OFDM spectrum.	43
Figure 6.4 Optical OFDM spectrum after a pair of MZMs.	43
Figure 6.5 Optical OFDM spectrum after 800 km of optical fiber transmission.	44
Figure 6.6 Five channel WDM CO-OFDM signal at the transmitter.	45
Figure 7.1 Output signal constellation of 16 QAM OFDM system after 800km transmission with different launch power. A) 1dBm, B) 3dBm, C) 5dBm.	47
Figure 7.2 Output signal constellation of 16 QAM OFDM system at 3dBm launch power with after different transmission length. A) 400km, B) 800km, C) 1200km.	47
Figure 7.3 Output signal constellation of 16 QAM OFDM system (A) w/o equalizer, (B) with linear equalizer (C) with full Volterra equalizer, (D) with sparse Volterra equalizer.	48
Figure 7.4 BER of 16 QAM OFDM system w/o compensation and with linear/nonlinear compensation as a function of fiber launch power.	50
Figure 7.5 BER of 16 QAM OFDM system with full Volterra/sparse Volterra equalization as a function of fiber launch power.	50

Figure 7.6 BER of 16 QAM OFDM system w/o compensation and with linear/nonlinear equalization as a function of OSNR (fixed launch power).	51
Figure 7.7 Optimal fiber transmission distance at 10^{-3} BER vs. launch power with and w/o compensation.	52
Figure 7.8 BER of a 5 channel WDM 16-QAM OFDM system w/o compensation and with linear/nonlinear compensation as a function of fiber launch power.	53
Figure 7.9 NMSE of different channel model of the CO-OFDM system versus launch power after 800km of transmission.....	54
Figure 7.10 Received signal constellation of 16 QAM OFDM systems under 0 dBm launch power. (A) w/o equalizer, (B) with linear inverse equalizer (C) with <i>pth-order</i> full Volterra inverse equalizer, (D) with <i>pth-order</i> sparse Volterra inverse equalizer.	55
Figure 7.11 BER versus launch power of the CO-OFDM system with or without compensation after 800km transmission.....	56
Figure 7.12 Maximum transmission length of the OFDM system with or without compensation as a function of launch power.	57
Figure 7.13 Electrical signal spectrum of the OFDM system without compensation, with equalization at the receiver and with predistortion at the transmitter.	58
Figure 7.14 BER of the CO-OFDM system without equalization or with linear equalizer, <i>pth-order</i> Volterra inverse equalizer, <i>pth-order</i> full Volterra inverse predistorter, <i>pth-order</i> sparse Volterra inverse predistorter at difference launch power.	59
Figure 7.15 Signal power entering the fiber of CO-OFDM systems with <i>pth-order</i> Volterra inverse equalizer and with <i>pth-order</i> Volterra inverse predistorter under different launch power.	60
Figure 7.16 OSNR of the CO-OFDM system with <i>pth-order</i> Volterra inverse equalizer and with <i>pth-order</i> Volterra inverse predistorter under different launch power.	61
Figure 7.17 Maximum fiber transmission length versus launch power of the OFDM systems without compensation with <i>pth-order</i> inverse equalizer and <i>pth-order</i> Volterra predistorter.....	62
Figure 7.18 Received signal constellation of 16 QAM OFDM system. (A) w/o equalizer, (B) with linear equalizer (C) with Volterra equalizer, (D) with Wiener-Hammerstein equalizer.	63
Figure 7.19 BER of 16 QAM OFDM system w/o compensation and with linear/nonlinear compensation as a function of launch power.	64

Figure 7.20 BER of 16 QAM OFDM system w/o compensation and with linear/nonlinear compensation as a function of OSNR (fixed launch power).	65
---	----

ACKNOWLEDGEMENTS

I would like to thank my supervising professor Dr. Chi-Hao Cheng for his invaluable advice and support for my research work. I want to express my appreciation to my committee members Dr. Kuang-Yi Wu and Dr. Dmitriy Garmatyuk for serving as my thesis reader and giving useful comments.

I appreciate all faculty and staffs in ECE department for helping me complete all my master thesis work. I would like to thank all the graduate students for offering their help when I came across some obstacles. I want to thank Mr. Jason Pennington and Mr. Wei Mu for helping revise my thesis. I also want to express my gratitude to my family for their support.

Chapter 1. Introduction

Coherent Optical OFDM (CO-OFDM) is considered an enabling technology of the next generation optical communication system since it possesses the merits of both a coherent system and an OFDM system [1]. As a coherent system, the CO-OFDM system maintains both signal amplitude and phase [2], thus increasing bandwidth utilization. The coherent optical communication system makes full compensation of chromatic dispersion, after optical/electrical conversion, possible. The OFDM modulation scheme also leads to a high spectral efficiency because of its partially overlapping subcarriers [1]. Moreover, the cyclic prefix code of the CO-OFDM system makes the system more resistant to inter-symbol interference caused by chromatic dispersion and polarization mode dispersion (PMD) [1, 3].

One major concern people have about the CO-OFDM system is its vulnerability to fiber nonlinear effects such as self-phase modulation (SPM) and cross-phase modulation (XPM). Both of SPM and XPM are caused by the optical signal intensity fluctuation [4]. Since the OFDM system has a high peak to average power ratio (PAPR) [5], a CO-OFDM system has more severe SPM and XPM compared with traditional optical communication systems. Because OFDM is a multi-carrier modulation scheme, the four-wave mixing (FMW) among subcarriers within one channel also causes concerns among researchers [6]. As a result, nonlinearity compensation is a crucial component of the CO-OFDM system. In this thesis, we concentrate on intra-channel nonlinearity distortion of the CO-OFDM system caused by SPM and FWM among subcarriers and present several nonlinear signal processing schemes to compensate for intra-channel nonlinearity.

The Volterra model is a widely used nonlinear signal processing tool [7]. It has been used to model the optical communication system nonlinearity [8-12]. The Volterra model is also used to mitigate nonlinearity effects in optical communication systems [13] and design equalizers for optical systems with OOK and PSK modulation [6, 11, 14-16]. The nonlinear system identification based on Volterra models can be carried out in frequency domain or in time domain. In this work, we focus on the time domain analysis of the nonlinear channel.

The biggest disadvantage of a Volterra model based nonlinear compensator is its complexity. A considerable amount of Volterra model coefficients is usually required to model a nonlinear system. Consequently, it may not be feasible to apply a Volterra model based compensator in

real-time signal processing applications [17]. One possible solution is to identify the most significant coefficients of a Volterra model and delete all of the insignificant coefficients from the Volterra model [17-18]. The resulting Volterra model is referred to as a sparse Volterra model by some researchers [19].

The *pth-order* inverse is an important Volterra model based equalization technique [20]. Many researchers have used the *pth-order* inverse theory as a tool for equalizers or predistorters design [20-23]. The difference between the equalizer and the predistorter is that the equalizer compensates the signal distortion at the receiver and the predistorter pre-compensates the signal at the transmitter. A *pth-order* inverse Volterra equalizer is demonstrated to be capable of compensating the joint effects of inter-symbol interference, nonlinearities and noise at the receiver [20]. An issue associated with the equalizer is that the received signal is corrupted by noise, and the equalizer may amplify noise [20]. On the other hand, a *pth-order* inverse predistorter would vary the actual input signal to the nonlinear system, make the channel model inaccurate although the predistorters can circumvent the noise enhancement problems [21, 22] and compensate the nonlinear distortion before the addition of noise [21]. To fully explore the advantages and disadvantages of the *pth-order* inverse equalizer and the *pth-order* inverse predistorter, we carried out a thorough comparison about these two compensation techniques. To the best of author's knowledge, it is the first time such a comparison has ever been conducted for the CO-OFDM system.

An alternative way for nonlinear compensation is to use the Wiener-Hammerstein model. Wiener-Hammerstein model is one of the commonly used block-oriented nonlinear structures [24], which comprises a cascading connection of a linear system, a memoryless nonlinearity system and a second linear system. This kind of nonlinear system has been used in the physiological system modeling [25], the power amplifiers modeling [26], the power amplifiers nonlinearity compensation [27], the acoustic echo cancellation [28, 29], the biological applications [30], and etc. To the best of the author's knowledge, it is the first time that the Wiener-Hammerstein model has ever been used in the optical communication system for fiber nonlinearity modeling and compensation. The employment of Wiener-Hammerstein model in optical nonlinear distortion is accomplishable because many Wiener-Hammerstein models would have a corresponding Volterra model representation [24], and the Volterra model has been successfully demonstrated to be capable of compensating the fiber nonlinearity effect. The

benefits of using a Wiener-Hammerstein model are that the number of coefficients would be significantly reduced and it has a simpler structure compared with the Volterra model.

In this thesis, the nonlinear effect of a CO-OFDM system is investigated, and equalizers based on the linear model, conventional Volterra model, sparse Volterra model, and Wiener-Hammerstein model are designed and tested. The *pth-order* inverse theory is applied to design the Volterra inverse predistorter and Volterra inverse equalizer. The distinctions between *pth-order* Volterra inverse predistorter and *pth-order* Volterra inverse equalizer are investigated. The simulation results show that nonlinear equalizers outperform linear equalizers. The sparse Volterra model has comparable performance to the conventional Volterra model. Both the *pth-order* Volterra inverse equalizer and the *pth-order* Volterra inverse predistorter can reduce the signal nonlinear distortion. A *pth-order* Volterra inverse equalizer performance would be affected by the system launch power; however, the launch power would have limited performance on the *pth-order* inverse Volterra predistorter. The Wiener-Hammerstein equalizer works better than the linear equalizer and has comparable performance to the Volterra equalizer with fewer coefficients. The rest of this thesis is organized as follows:

In chapter 2, the single channel and the wavelength division multiplexing (WDM) fiber optics systems are introduced. The principles of coherent detection and direct detection are introduced. The nonlinearity effects, like self-phase modulation and cross-phase modulation are described.

Chapter 3 reviews the Volterra model which will be used to model and compensate the signal nonlinear distortions. It is also shown how the orthogonal search method can reduce the complexity of the Volterra model. The obtained model is called the sparse Volterra model. The *pth-order* inverse theory is discussed. The Wiener-Hammerstein model is also explained in this chapter.

In chapter 4, two adaptive signal processing algorithms, least mean square (LMS) and recursive least square (RLS) algorithms, are introduced. These two algorithms will be used to determine the channel model and compensator coefficients in this study.

In chapter 5, we describe how to model the optical transmission channel or compensate its nonlinear distortion using the conventional Volterra model, the sparse Volterra model and the Wiener-Hammerstein model. The procedure of applying the *pth-order* inverse theory to design the Volterra inverse predistorter and Volterra inverse equalizer is provided. The determination of equalizers based on adaptive signal processing algorithms is presented in this chapter as well.

In chapter 6, we describe the simulation setup in details. The simulation is done using commercial software, OptiSystemTM. The implementation of the nonlinear equalizers and the predistorters in the simulation setup are also discussed.

Simulation results are included in Chapter 7. The performance of different types of compensation schemes is presented and a detailed comparison between the equalizer and predistorter is provided.

Chapter 8 concludes this thesis.

Chapter 2. Fiber Optics and Optical CO-OFDM

The use of light for communication can be dated back to ancient times. As the technology has been improved through generations, massive information can be transmitted as optical waveforms. Comparing with other communication techniques, optical system enables a high bit rate signal transmission because of its ultra-high carrier frequency. This chapter provides a concise overview of the fiber optics communication system including optical OFDM and fiber nonlinearity.

2.1 Fiber Optics

Fiber optics communication systems are lightwave systems which transmit information through optical fibers [3]. In the optical system, information modulates optical carriers [31]. Similar to other communication systems, optical communication components include a transmitter, a receiver, and a transmitting channel, as shown in Fig.2.1.

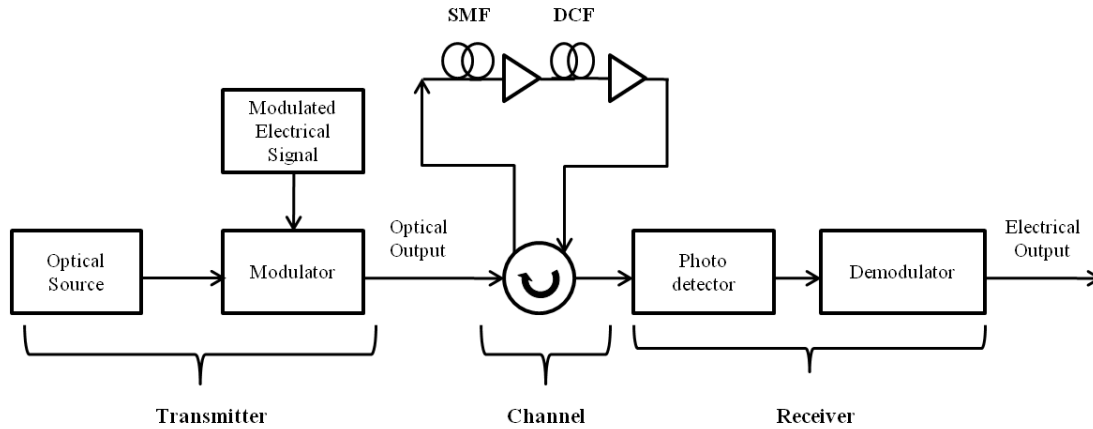


Figure 2.1 Components of an optical communication system.

The optical source can be the semiconductor laser or light-emitting diodes, which generates carrier waves at different frequencies. The electrical signal entering the modulator can be pre-coded by different schemes, for example On-Off Keying and PSK. The modulator, like the commonly used Mach-Zehnder modulator, is used to modulate the optical carrier and convert electrical signals to optical signals. The generated optical signals are transmitted through the optical fibers. At the receiver, direct detection or coherent detection technique can be used to

convert the optical signals back to electrical signals and those obtained signals will be demodulated and recovered. The detection techniques are described in details in section 2.2.

Optical fibers can be classified into multimode fiber and single mode fiber (SMF) depending on the fiber's core diameter. In our study, we only analyze the signals passing through SMF since SMF can support long distance transmission. When signals transmit through the fiber, they will undergo attenuation, chromatic dispersion and nonlinear distortion. Fiber attenuation is caused by the fluctuation of the refractive index and imperfections in the fiber [32]. Optical amplifiers are used to compensate attenuation, but the optical amplifier would generate amplified spontaneous emission (ASE) noise, decreasing the optical signal to noise ratio (OSNR). Chromatic dispersion comes from the fact that optical signals at different wavelengths will propagate at different group velocities [32], resulting in the broadening of an optical pulse. Dispersion compensation fiber (DCF) and the electrical dispersion compensator are two common tools for the chromatic dispersion compensation. In our study, DCF is used to compensate the chromatic dispersion in the transmission channel. The signal distortion caused by the fiber nonlinearity is another challenge in optical communication system. It will be discussed in the later sections.

Wavelength Division Multiplexing (WDM) technology can increase the bandwidth utilization of the fiber. In a WDM system, signals at different wavelengths travel in the same fiber [32], as illustrated in Fig.2.2. In each channel, different electrical signals modulate carriers with different wavelengths. Different wavelengths are coupled by a multiplexer and transmit through the fiber. At the receiver, those wavelengths are separated by the de-multiplexer and are demodulated separately.

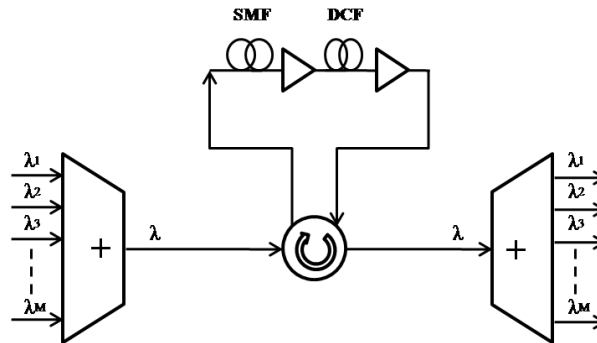


Figure 2.2 Block diagram for WDM system.

2.2 Principles of OFDM

In this study, we use the OFDM technique to modulate the electrical signal. In a multicarrier modulation system (MCM), the data stream is parsed into several parallel sub-streams and each sub-stream modulates one subcarrier. A MCM signal at transmitter can be written as [3]:

$$s(t) = \sum_{i=-\infty}^{+\infty} \sum_{k=1}^N c_{ki} s_k(t - iT_s) \quad (2.1)$$

$$s_k(t) = \Pi(t) e^{2\pi j f_k t} \quad (2.2)$$

$$\Pi(t) = \begin{cases} 1, & 0 < t \leq T_s \\ 0, & \text{otherwise} \end{cases} \quad (2.3)$$

s_k is the subcarrier waveform and c_{ki} is the information at the k th subcarrier. N is the number of subcarriers, and f_k is the corresponding frequency of the subcarrier. T_s is the symbol period, and $\Pi(t)$ is the pulse shaping function. If we sample $s(t)$ in Eq. 2.1 with sampling period of $\frac{T_s}{N}$, the m th sample can be represented as [3]:

$$s_m = \sum_{k=1}^N c_k \cdot e^{j2\pi f_k \frac{(m-1)T_s}{N}} \quad (2.4)$$

OFDM is a special class of MCM. In an OFDM system, different subcarrier carrier frequencies are chosen so that each subcarrier is orthogonal to each other. Because of the orthogonality of the OFDM subcarrier, we will have

$$f_k = \frac{k-1}{T_s} \quad (2.5)$$

Substituting Eq. 2.5 into Eq.2.4, we get Eq.2.6 [3]:

$$s_m = \sum_{k=1}^N c_k \cdot e^{j2\pi \frac{(m-1)(k-1)}{N}} \quad (2.6)$$

We can see s_m is the inverse Fourier transforms of input signal $\{c_k\}$. At the receiver, assuming perfect synchronization, after sampling with sampling period $\frac{T_s}{N}$, the received signals become $\{\hat{s}_m\}$. The recovered signals $\{\hat{c}_k\}$ would be the Fourier transforms of the received signal $\{\hat{s}_m\}$.

$$\hat{c}_k = \frac{1}{\sqrt{N}} \sum_{m=1}^N \hat{s}_m e^{-j2\pi \frac{(m-1)(k-1)}{N}} \quad (2.7)$$

By evaluating Eq. 2.6, it is obvious that OFDM signal is a summation of several subcarriers. As a result, OFDM signal would have a higher peak to average power ratio (defined in Eq. 2.8) than single carrier signals.

$$PAPR = \frac{\max\{|s(t)|^2\}}{E\{|s(t)|^2\}}, \quad t \in [0, T_s] \quad (2.8)$$

For the RF OFDM system, the power amplifier gain will saturate at high input power, causing nonlinear signal distortion [3], while for the optical OFDM system, the currently used optical amplifier is linear with its input signal power. However, the optical fiber nonlinearity is more severe under high input signal power. The nonlinearity effect in optical OFDM systems is an important drawback.

Since OFDM is a multicarrier system, when passing through a dispersive channel, different subcarrier will transmit at different group velocities, causing dispersions. To reduce the dispersion effect, a cyclic prefix or zero padding is usually added in the modulation block.

WDM optical OFDM is a WDM system, which uses OFDM as the electrical modulation scheme. In each channel, the signal is modulated by the OFDM technique. The OFDM signals from each channel modulate an optical carrier of different wavelength using a pair of Mach-Zehnder modulators (MZM). Those signals are multiplexed and then transmitted through the same fiber. At the receiver, the signals are de-multiplexed and detected separately.

2.3 Optical Detection Techniques

The optical signal is converted to electrical signal by photo-detectors. The detection technique is described in this section. After the OFDM modulation, the baseband electrical signal $s(t)$ is

transformed to the optical domain by a pair of MZMs. The resulting optical domain signal becomes:

$$E(t) = e^{j(w_{LD1}t + \phi_{LD1})}s(t) \quad (2.9)$$

where w_{LD1} and ϕ_{LD1} are the angular frequency and phase of the launch laser at the transmitter. After passing through the optical channel, whose impulse response is $h(t)$, the output optical signal becomes the convolution of input signal $E(t)$ and $h(t)$.

$$E_1(t) = e^{j(w_{LD1}t + \phi_{LD1})}s(t) \otimes h(t) \quad (2.10)$$

where \otimes denotes the convolution symbol. At the receiver, a hybrid coherent detection technique is used to convert the optical signals to the electrical signals. The in-phase (I) component and the quadrature (Q) component are the real part and imaginary part of the signals respectively. The main purpose of the coherent detection is to linearly recover I and Q channels, and to reduce the noise [3]. Compared with traditional direct detection techniques, coherent detection can fully recover the phase and amplitude information, enabling an accurate signal recovery. The coherent detection components consist of a 2×4 90 degree optical hybrid and two pairs of photo-detectors, as seen in Fig.2.3. E_1 is the incoming signal, and E_2 is the local oscillator (LO) laser signal, which can be written as Eq.2.11, where w_{LD2} and ϕ_{LD2} are the angular frequency and phase of the LO laser at the receiver.

$$E_2(t) = e^{j(w_{LD2}t + \phi_{LD2})} \quad (2.11)$$

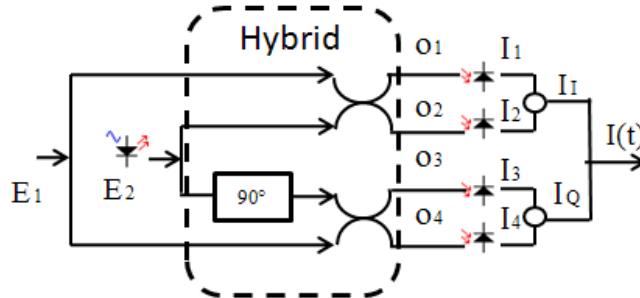


Figure 2.3 Coherent detection block diagram.

After the optical hybrid conversion, the output signals $O_{1\sim 4}$ from the four ports are as following [3]:

$$O_1 = \frac{1}{\sqrt{2}}(E_1 + E_2) \quad (2.12)$$

$$O_2 = \frac{1}{\sqrt{2}}(E_1 - E_2) \quad (2.13)$$

$$O_3 = \frac{1}{\sqrt{2}}(E_1 + jE_2) \quad (2.14)$$

$$O_4 = \frac{1}{\sqrt{2}}(-E_1 + jE_2) \quad (2.15)$$

Based on the square law detection, the corresponding I channel photocurrent I_{I-2} and the Q channel photocurrents I_{3-4} would become [3]:

$$I_1 = |O_1|^2 = 0.5(|E_1|^2 + |E_2|^2 + 2\text{Re}(E_1 E_2^*)) \quad (2.16)$$

$$I_2 = |O_2|^2 = 0.5(|E_1|^2 + |E_2|^2 - 2\text{Re}(E_1 E_2^*)) \quad (2.17)$$

$$I_3 = |O_3|^2 = 0.5(|E_1|^2 + |E_2|^2 + 2\text{Im}(E_1 E_2^*)) \quad (2.18)$$

$$I_4 = |O_4|^2 = 0.5(|E_1|^2 + |E_2|^2 - 2\text{Im}(E_1 E_2^*)) \quad (2.19)$$

Because of balanced detection, I and Q components become:

$$I_I(t) = I_1 - I_2 = 2\text{Re}(E_1 E_2^*) \quad (2.20)$$

$$I_Q(t) = I_3 - I_4 = 2\text{Im}(E_1 E_2^*) \quad (2.21)$$

The combination of I and Q photocurrent gives the complex signal:

$$I(t) = I_I(t) + jI_Q(t) = 2E_1E_2^* \quad (2.22)$$

Substituting Eq. 2.10 and Eq.2.11 into Eq. 2.22

$$I(t) = 2e^{j(\Delta\omega_{LD}t + \Delta\phi_{LD})}s(t) * h(t) \quad (2.23)$$

$$\Delta\omega_{LD} = \omega_{LD1} - \omega_{LD2} \quad (2.24)$$

$$\Delta\phi_{LD} = \phi_{LD1} - \phi_{LD2} \quad (2.25)$$

$\Delta\omega_{LD}$ and $\Delta\phi_{LD}$ are the frequency offset and phase offset between the transmitter and receiver laser respectively. Apparently, after coherent-detection the signal is a linear replica of the incoming complex signal [3], and the phase and amplitude of the incoming signal can be well maintained. At the receiver, if we apply the direct detection technique, the optical signal is detected by the photo-detection based on the square law detection, as shown in Fig.2.4. The output signal magnitude is proportional to the received optical signal power (Eq. 2.26); however, the signal phase information cannot be maintained.

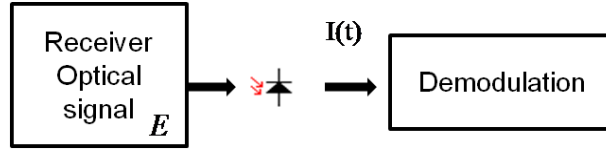


Figure 2.4 Direct detection block diagram.

$$I(t) = |E_1|^2 \quad (2.26)$$

2.4 Fiber Nonlinearities

Fiber nonlinearities can be classified into two categories. One is the Kerr nonlinearities and the other is stimulated scattering [3]. Kerr effect is the electrical perturbation on the material refractive index [33]. The refractive index of the material in the direction of electric field can be expressed by:

$$n(I) = n_0 + n_2I \quad (2.27)$$

where I is the light intensity. n_0 is the refractive index. n_2 is the nonlinear refractive index of the fiber (Kerr coefficient) and is positive for most materials. Since the refractive index is a function of the light intensity, a high launch power laser modulated by electrical signals would result in more variance in the refractive index and cause higher nonlinearity. Nonlinear Schrödinger (NLS) equation in Eq.2.28 is often used to model the fiber nonlinearity effect on the pulse propagating in optical fibers [34].

$$\frac{\partial A}{\partial z} = -\frac{\alpha}{2}A - j\frac{\beta_2}{2}\frac{\partial^2 A}{\partial t^2} + j\gamma|A|^2A \quad (2.28)$$

where $A(t, z)$ is the slowly varying complex envelope at position z in the fiber and time t . β_2 is the second-order dispersion parameter, describing the group velocity dispersion. γ is defined as the fiber nonlinearity coefficients. α is the attenuation coefficient.

In this work, we mainly focus on the self-phase modulation and cross-phase modulation caused by the Kerr nonlinearities. Self-phase modulation (SPM) is the most influential nonlinear effect in single channel optical transmission, which can cause the spectrum broadening. The total nonlinear phase shift caused by SPM based on the NLS equation can be described in Eqs.2.29-2.31, where P_0 is the launch power. A_{eff} is the fiber effective area. L_{eff} is the effective fiber length. λ is the reference wavelength. L is the fiber transmission length and n_2 is the fiber nonlinear index. It is apparent that with higher launch power, at longer transmission distance the phase shift due to SPM becomes more severe.

$$\Delta\phi = \gamma P_0(t)L_{eff} \quad (2.29)$$

$$L_{eff} = \frac{1 - e^{-\alpha L}}{\alpha} \quad (2.30)$$

$$\gamma = \frac{2\pi n_2}{\lambda A_{eff}} \quad (2.31)$$

In a WDM system, the cross-phase modulation (XPM) appears. For a certain channel, besides its own signal power, the signal phase shift of that particular channel after fiber transmission length L is also affected by the power of other channels. For a WDM system, the phase shift in the m th channel can be represented by,

$$\Delta\phi_m = \gamma L_{eff} [P_{0m}(t) + 2 \sum_{j=1, j \neq m}^M P_{0j}(t)] \quad (2.32)$$

where $P_{0m}(t)$ is the pulse shape in the m th channel. Apparently, the phase shift in WDM system is more severe than a single channel transmission. Another important nonlinear effect is the four-wave mixing (FWM). It is defined as when three optical signals with different carrier frequencies f_i, f_j, f_k interact, they tend to generate new frequency as described by Eq.2.33. FWM usually happens in a WDM system. It may also exist in a single channel OFDM system, since its subcarriers occupy different frequencies. Each subcarrier may interact with others and generate new frequencies.

$$f_{ijk} = f_i + f_j - f_k \quad (2.33)$$

Chapter 3. Modeling Techniques

In nonlinear systems, the outputs are nonlinear functions of input signals. In many cases, the linear model is not accurate enough to model the nonlinear channel. To solve this problem, researchers resort to different nonlinear filters and use them to model the channel and compensate the nonlinear distortions.

3.1 Linear Model

If we consider the optical channel as a linear time-invariant system, its input-output relation can be represented by the following equation:

$$y(n) = \sum_{i=0}^{L-1} w(i)x(n-i) \quad (3.1)$$

$x(n)$ is the filter input signal, w is the filter coefficient and $y(n)$ is the filter output signal. $L-1$ is the linear filter memory length. However, a linear filter is insufficient to model nonlinear fiber effects like the Kerr effects. Therefore, a nonlinear optical channel model is necessary and the popular nonlinear filters Volterra model and Wiener-Hammerstein model are studied in this thesis.

3.2 Volterra Model

For a continuous-time Volterra model, its output signal $y(t)$ can be represented by the sum of linear, quadratic, cubic and higher-order convolution integrals of input signal $x(t)$ [7]. The input-output relations of a continuous-time Volterra system can be described as [7]:

$$\begin{aligned} y(t) = & \int_{-\infty}^{+\infty} h_1(\tau_1)x(t-\tau_1) d\tau_1 + \iint_{-\infty}^{+\infty} h_2(\tau_1, \tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1d\tau_2 \\ & + \iiint_{-\infty}^{+\infty} h_3(\tau_1, \tau_2, \tau_3)x(t-\tau_1)x(t-\tau_2)x(t-\tau_3) d\tau_1d\tau_2d\tau_3 + \dots \end{aligned} \quad (3.2)$$

$h_j(\tau_1, \tau_2, \tau_3, \dots, \tau_j)$ is the j th order Volterra kernel. For a causal system, for any $\tau_j < 0$, $h_j(\tau_1, \tau_2, \tau_3, \dots, \tau_j) = 0$. Similarly, a causal discrete-time Volterra filter can be written as [7]:

$$\begin{aligned}
y(n) = & \sum_{k_1=0}^{\infty} h_1(k_1)x(n-k_1) + \sum_{k_1=0}^{\infty} \sum_{k_2=0}^{\infty} h_2(k_1, k_2)x(n-k_1)x(n-k_2) \\
& + \sum_{k_1=0}^{\infty} \sum_{k_2=0}^{\infty} \sum_{k_3=0}^{\infty} h_3(k_1, k_2, k_3)x(n-k_1)x(n-k_2)x(n-k_3) + \dots
\end{aligned} \tag{3.3}$$

A discrete nonlinear bandpass channel can be represented by a complex Volterra model whose input and output are signals' complex envelopes. Because of the bandpass nature [35], the even order terms in the Volterra transfer function will not generate in-band frequency components [36]. Thus, the even order terms in the bandpass Volterra series can be ignored. The third order bandpass Volterra model would be sufficient enough to model the nonlinear optical fiber channel. In our following simulation, we will represent the optical nonlinear channel using the causal, truncated, discrete time third-order bandpass Volterra system as follows [36, 37]:

$$\begin{aligned}
y(n) = & \sum_{k_1=0}^N h_1(k_1)x(n-k_1) \\
& + \sum_{k_1=0}^N \sum_{k_2=0}^N \sum_{k_3=0}^N h_3(k_1, k_2, k_3)x(n-k_1)x(n-k_2)x^*(n-k_3)
\end{aligned} \tag{3.4}$$

N is the memory length, and $()^*$ denotes the complex conjugate. $x(n)$ and $y(n)$ are the complex envelopes of input signal and model's output signals at time n . $h_1[]$ and $h_3[]$ are the linear and cubic Volterra kernels respectively. The total number of coefficients L would equals to $(N+1) + (N+1)^3$. Notice that the Volterra series is a linear combination of Volterra kernels. As a result, when a Volterra model is used to model a nonlinear system, the Volterra kernels can be updated using the least mean square (LMS) or the recursive least square (RLS) methods, which will be described in the next chapter.

3.3 Sparse Volterra Model

Usually, to model a highly distorted transmission with a Volterra model, a comparatively long memory length and high filter order would be chosen. However, an increase in memory length N or filter order will increase the number of kernels exponentially, resulting in complicated calculations. Moreover, the estimation of Volterra kernels may lose accuracy since distortion

will take place when estimating a large number of unwanted kernels [17]. As a result, it is necessary to remove the unwanted kernels to simplify the structure in practical applications.

To get a compact Volterra structure with the orthogonal search approach [17-18], the filter coefficients are updated and searched in the orthogonal domain, making the search process more straightforward. The detailed procedure is described as following. The nonlinear equation Eq. 3.4 can be expressed by [17]:

$$y(n) = \sum_{i=1}^L w_i u_i(n) + e(n) \quad (3.5)$$

where L is the total number of Volterra kernels, u_i is the linear or cubic input terms, and w_i is the corresponding Volterra kernels. To represent Eq.3.5 in a vector form:

$$Y = \bar{Y} + E = \sum_{i=1}^L w_i U_i + E \quad (3.6)$$

where,

$$Y = [y(1), y(2), \dots y(i) \dots]^T$$

$$U_i = [u_1(1), u_2(2), \dots u_2(i) \dots]^T$$

$$E = [e(1), e(2), \dots e(i) \dots]^T$$

$()^T$ demotes transpose. Since each coefficient w_i carries different weights, our goal is to identify the most important kernels and ignore the rest. We will solve this problem in the orthogonal domain. In orthogonal domain, Eq.3.6 is equivalent to

$$Y = \bar{Y} + E = \sum_{i=1}^L v_i Q_i + E \quad (3.7)$$

Where, Q is the orthogonalized matrix of vectors U . v_i is the orthogonal Volterra coefficients corresponding to Q_i . Since matrix Q is orthogonal, the coefficients v_i can be updated by [17]:

$$v_i = \frac{Y^T Q_i}{Q_i^T Q_i} \quad (3.8)$$

A simplified model needs to be constructed so that \bar{Y} approaches output Y with the minimum number of terms, which is equivalent to minimizing the normalized mean square error (NMSE) of the model.

$$NMSE = \frac{E^T E}{Y^T Y} = \frac{(Y - \sum_{i=1}^L v_i Q_i)^T (Y - \sum_{i=1}^L v_i Q_i)}{Y^T Y} = 1 - \sum_{i=1}^L D_i \quad (3.9)$$

$$D_i = \frac{v_i^2 Q_i^T Q_i}{Y^T Y} \quad (3.10)$$

In Eq.3.9, we find that the problem is simplified to identify the most significant v_i among the L coefficients which contributes to a large D_i in Eq.3.10. The orthogonalization procedure of converting U to Q can be completed by the modified Gram-Schmidt method. According to [17-18, 38], the modified Gram-Schmidt method is summarized as following.

```

Q = U;
for j = 1:L
    a = norm(Q(:, j))
    for i = 1:j-1
        R(i, j) = Q(:, i)' * Q(:, j)
    prob: Q(:, j) = Q(:, j) - R(i, j) * Q(:, i)
    end for
    b = norm(Q(:, j))
    R(j, j) = norm(Q(:, j))
    Q(:, j) = Q(:, j) / R(j, j)
end for

```

One problem associated with the modified Gram Schmidt method is that Q may lose orthogonality due to the intensive cancellation happened in line *prob* [39]. One way to solve this problem is to define a value ε and if $\|b\|_2 \leq \varepsilon \|a\|_2$, a reorthogonalization is needed [39]. The pseudo code above is then modified as [39]:

```

    Q=U;
    for j=1:L
        a=norm(Q(:,j))
        re_org: for i=1:j-1
            s=Q(:,i)'*Q(:,j)
            R(i,j)=R(i,j)+s
            Q(:,j)=Q(:,j)-s*Q(:,i)
        end for
        b=norm(Q(:,j))
        if b<εa then (a=b) and (goto re_org)
    R(j,j)=b;
    Q(:,j)=Q(:,j)/R(j,j)
    end for

```

When the input matrix is rank deficient, some vectors are linearly dependent. Those linear dependent vectors can be easily identified by evaluating the value of b . If the value b is small, then the row of R and the related column Q_i will be set to zero. Suppose the linear dependent vector indexes are k , and linear independent vector indexes are j , Eq.3.6 would become:

$$Y = \bar{Y} + E = \sum_j v_j Q_j + \sum_k v_k Q_k + E \quad (3.11)$$

Assume that M is the rank of the input matrix ($M \leq L$), the number of “valid” coefficients would be M , with the rest of coefficients set equal to zero. The kernels of the sparse Volterra model are constructed from these M coefficients. Since $Q_m^T Q_n = 0$ ($m \neq n$) and $Q_m^T Q_m = 1$, Eq. 3.8 and Eq.3.10 can be revised as:

$$v_i = Y^T Q_i \quad (3.11)$$

$$D_i = \frac{v_i^2}{Y^T Y} \quad (3.12)$$

Each time, the coefficient v_i which generates the biggest D_i , is selected and $NMSE = NMSE - D_i$. The search process ends once the NMSE meets the pre-determined value. To construct

sparse Volterra models, the orthogonal coefficients, will still have size of L with all the unimportant coefficients set to zero. Also the upper triangle matrix $R(i, j)$, the i th row which corresponding to the unimportant coefficient index i is also set to zero. The back substitution method shown below is then used to determine the sparse Volterra model kernels [38]. The resulting sparse Volterra model will have less non-zero kernels than the “full” Volterra model.

```

for i = L : 1
    w_i = v_i
    for j = (i + 1) : L
        w_i = w_i - R(i, j) * v_j
    end for
    w_i = w_i / R(i, i)
end for

```

In the back substitution algorithm above, if $R(i, i) = 0$, w_i will be set to be zero correspondingly.

3.4 P th-order Inverse of Volterra Series

People have used the nonlinear model based on Volterra series to analyze dispersion, attenuation and SPM effect in optical fibers. A compensator can be designed based on the inverse system of the existing nonlinear channel model. However, for some nonlinear systems, their inverse doesn't exist, while others possess an inverse system only for a restricted range of input amplitudes [40]. The p th-order inverse can be used to obtain an inverse system of a nonlinear channel in which the input amplitude range is not restricted.

To understand the p th-order Volterra inverse method, we write an M th-order Volterra series as [40]:

$$y(n) = H[x(n)] = \sum_{i=1}^M H_i[x(n)] \quad (3.13)$$

and

$$H_i[x(n)] = \sum_{k_1=0}^N \cdots \sum_{k_i=0}^N h_i(k_1, \dots, k_i) x(n - k_1) \dots x(n - k_i) \quad (3.14)$$

H is the system operator and H_i is the i th-order Volterra operator. The p th-order inverse of the

nonlinear system H is defined as a p th-order nonlinear system when connected in tandem with H , will result in a system, in which the second through the p th-order Volterra operators of the system are zero [40]. As shown in Fig.3.1, K is the p th-order Volterra inverse system of the nonlinear system H , and R represents the cascading system.

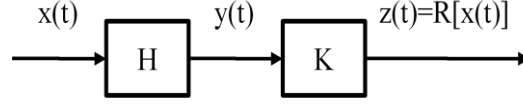


Figure 3.1 Nonlinear system H connected with its inverse system K (post inverse system).

The p th-order Volterra inverse K can be written as in Eq.3.15, where K_i is the i th-order Volterra inverse operator and the system R can be written as in Eq. 3.16, where R_i is the i th-order Volterra operator of system R .

$$K[y(n)] = \sum_{i=1}^p K_i[y(n)] \quad (3.15)$$

$$z(n) = R[x(n)] = x(n) + \sum_{i=p+1}^{PM} R_i[x(n)] \quad (3.16)$$

Our goal is to obtain the third order Volterra inverse of the bandpass Volterra system whose second order Volterra operator is ignored. The first and the third inverse Volterra operator can be derived as:

$$K_1 = H_1^{-1} \quad (3.17)$$

$$K_3 = -K_1 H_3 K_1 \quad (3.18)$$

To summarize, the third order Volterra inverse K would be:

$$K[y(n)] = H_1^{-1}[y(n)] - H_1^{-1}H_3H_1^{-1}[y(n)] \quad (3.19)$$

The p th-order inverse K followed by the nonlinear system H is referred to as the p th-order pre-inverse system, and the p th-order inverse K following the nonlinear system H is referred to as

the p th-order post-inverse system. As shown in reference 40, the p th-order pre-inverse of a system H is the same as the p th-order post inverse of H . We can design predistorters or equalizers based on the p th-order inverse theory to compensate nonlinear distortion in an optical communication system.

3.5 Wiener-Hammerstein model

In this section, the complex Wiener-Hammerstein model is used to represent the discrete nonlinear bandpass channel. In this model, the first subsystem is the FIR filter. The second subsystem is the nonlinear polynomial filter, and the third subsystem is another FIR filter. The configuration of the channel modeling performed by a Wiener-Hammerstein model is shown in Fig.3.2.

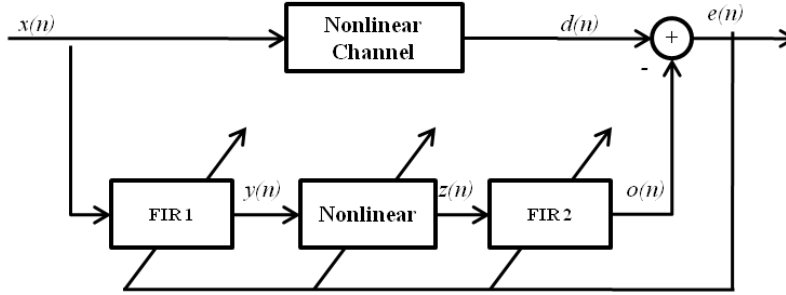


Figure 3.2 Wiener-Hammerstein system for channel modeling.

After passing through the first FIR filter, the input-output relation in a discrete and time invariant form is displayed as following, where $M_1 - 1$ is the first FIR filter memory length.

$$y(n) = \sum_{i=0}^{M_1-1} u(i)x(n-i) \quad (3.20)$$

Because the nonlinear optical channel is assumed to be bandpass, only the odd order terms in the “sandwiched” nonlinear subsystem can generate nonzero output and the even order terms is neglected. In the Wiener-Hammerstein model, the center nonlinear subsystem order would determine its equivalent Volterra system order; Several researchers has proved that a third order Volterra model is sufficient to model the nonlinear optical channel; As a result, the center nonlinear filter order is set at three. The output signal after passing through the center nonlinear filter is written as:

$$z(n) = v(0)y(n) + v(1)y^2(n)y^*(n)$$

(3.21)

The relationship between the output signals $o(n)$ and the input signals $z(n)$ of the second FIR filter is represented by Eq.3.22, where $M_2 - 1$ is the second FIR filter memory length.

$$o(n) = \sum_{i=0}^{M_2-1} w(i)z(n-i)$$

(3.22)

Chapter 4. Adaptive Algorithms

The adaptive Least Mean Square (LMS) and Recursive Least Square (RLS) algorithms are two most popular adaptive signal processing algorithms. Since the Volterra series is a linear combination of Volterra kernels as shown in Chapter 3, both LMS and RLS can be used to determine the coefficients of a Volterra series based nonlinear channel model or the compensator. The linear and nonlinear subsystem coefficients of the Wiener-Hammerstein model can be updated by the joint normalized LMS (joint NLMS) algorithm. Without losing generality, we use a linear filter to explain LMS algorithm and use a Volterra nonlinear filter to explain the RLS algorithm. Their detailed description is given below.

4.1 Least Mean Square (LMS) Algorithm

For an adaptive linear filter with L number of coefficients, its output signal $y(n)$ at certain time n can be represented as the follows:

$$y(n) = \sum_{i=0}^{L-1} w_i(n)x(n-i) \quad (4.1)$$

$x(n)$ is the filter input signal, $w_i(n)$ is the tap coefficient at time n . We can express all the coefficients $w_i(n)$ and the input signals $x(n)$ in a vector form, as shown in Eq.4.2 and 4.3:

$$\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_j(n), \dots, w_{L-1}(n)]^T \quad (4.2)$$

$$\mathbf{u}(n) = [x(n), x(n-1), \dots, x(n-j), \dots, x(n-L+1)]^T \quad (4.3)$$

The desired output vector D is:

$$D = [d(1), d(2), \dots, d(i), \dots] \quad (4.4)$$

Assume we use the linear filter to model an unknown system. At time n , the difference $e(n)$ between the desired signal $d(n)$ and the filter output $y(n)$ is:

$$\begin{aligned}
e(n) &= d(n) - y(n) \\
&= d(n) - \mathbf{w}^T(n)\mathbf{u}(n)
\end{aligned}
\tag{4.5}$$

The problem we have to solve is to update the adaptive filter coefficients in Eq.4.2 to minimize $E[e^2(n)]$. The coefficients vector can be updated by the gradient of the mean square error as shown in Eq. 4.6, where ε is the step size.

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \varepsilon \nabla\{E[e^2(n)]\} \tag{4.6}$$

Since $e(n)$ is a function of filter coefficients, the gradient of the mean square error can be estimated as follows

$$\begin{aligned}
\nabla\{E[e^2(n)]\} &\cong \nabla\{e^2(n)\} = 2e(n) \cdot \nabla\{e(n)\} \\
&= 2e(n) \cdot \begin{bmatrix} \frac{\partial e(n)}{\partial w_0(n)} \\ \frac{\partial e(n)}{\partial w_1(n)} \\ \vdots \\ \frac{\partial e(n)}{\partial w_{L-1}(n)} \end{bmatrix} \\
&= -2e(n) \cdot \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-L+1) \end{bmatrix}
\end{aligned}
\tag{4.7}$$

Suppose the coefficients vector needs to be estimated is $\hat{\mathbf{w}}$, the recursive equation for updating the filter coefficients can be summarized as in Table 1.

Table 1. LMS algorithm summary for real signal.

Initialization:
$\hat{\mathbf{w}} = \mathbf{0}$
Update procedure (n=1, 2, 3...):
Estimation error:
$e(n) = d(n) - \hat{\mathbf{w}}^T(n)\mathbf{u}(n)$
Coefficients update:
$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + 2\varepsilon e(n)\mathbf{u}(n)$
ε is a small positive number

For complex input signals, the coefficients $w_i(n)$ can be written in vector form as:

$$\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_j(n), \dots, w_{L-1}(n)]^H \quad (4.8)$$

The recursive equation for updating the filter coefficients can be summarized as in Table 2.

Table 2 LMS algorithm summary for complex signal.

Initialization:
$\hat{\mathbf{w}} = \mathbf{0}$
Update procedure (n=1, 2, 3...):
Estimation error:
$e(n) = d(n) - \hat{\mathbf{w}}^H \mathbf{u}(n)$
Coefficients update:
$\hat{\mathbf{w}} = \hat{\mathbf{w}} + 2\varepsilon e^*(n)\mathbf{u}(n)$
ε is a small positive number

4.2 Recursive Least Square (RLS) Algorithm

In a Volterra model, although the output signal is a polynomial combination of current and past input symbol, $x(n)$, its output symbol is linearly dependent on the Volterra filter kernels. The input-output relations of a complex third order adaptive Volterra model at time n can be

represented by Eq.4.9, where $h_i(n)$ are the linear kernels and the $h_{i,j,k}(n)$ are the third order kernels at time n .

$$y(n) = \sum_{i=0}^N h_i(n)x(n-i) + \sum_{i=0}^N \sum_{j=0}^N \sum_{k=0}^N h_{i,j,k}(n)x(n-i)x(n-j)x^*(n-k) \quad (4.9)$$

The RLS method converges faster than the LMS method at the expense of increasing computational complexity. In this section, we demonstrate how the RLS algorithm can be used to update the Volterra model coefficients. We use a third order Volterra filter to model an unknown system. In the Volterra model, the input vector is given in Eq.4.10 and the desired output vector D as shown in Eq.4.4.

$$\mathbf{u}[n] = [x(n), x(n-1), \dots, x(n-N), x^2(n)x^*(n), x^2(n)x^*(n-1), \dots, x^2(n-N)x^*(n-N)]^T \quad (4.10)$$

The coefficients vector is:

$$\mathbf{w} = [h_0(n), h_1(n), \dots, h_N(n), h_{000}(n), h_{001}(n), \dots, h_{0N0}(n), h_{0N1}(n), \dots, h_{NNN}(n)]^H \quad (4.11)$$

L is the total coefficient number. The difference between desired output and estimated output can be expressed by:

$$e(i) = d(i) - y(i) = d(i) - \mathbf{w}^H(n)\mathbf{u}(i) \quad (4.12)$$

The objective of least square algorithm is to minimize the cost function:

$$\varepsilon(n) = \sum_{i=1}^n \gamma^{n-i} |e(i)|^2 \quad (4.13)$$

where γ is the exponential weighting factor or forgetting factor, which can track the slow statistical variations of the channel [36]. It is a positive constant with value close to, but less than one [36, 42]. A special case is $\gamma = 1$, which corresponds to infinite memory [42], and can be used in a stationary environment [42]. The correlation matrix $\boldsymbol{\phi}$ of the input signal and the cross-correlation vector $\boldsymbol{\theta}$ between the filter input and the desired output signals can be defined as following:

$$\boldsymbol{\varphi}(n) = \sum_{i=1}^n \gamma^{n-i} \mathbf{u}(i) \mathbf{u}^H(i) \quad (4.14)$$

$$\boldsymbol{\theta}(n) = \sum_{i=1}^n \gamma^{n-i} \mathbf{u}(i) d^*(i) \quad (4.15)$$

The optimal value of $\mathbf{w}(n)$ which minimizes the cost function can be obtained from:

$$\boldsymbol{\theta}(n) = \boldsymbol{\varphi}^{-1}(n) \mathbf{w}(n) \quad (4.16)$$

The correlation matrix in Eq.4.14 and the cross-correlation vector in Eq.4.15 can be reorganized as following:

$$\boldsymbol{\varphi}(n) = \gamma \sum_{i=1}^{n-1} \gamma^{n-i-1} \mathbf{u}(i) \mathbf{u}^H(i) + \mathbf{u}(n) \mathbf{u}^H(n) = \gamma \boldsymbol{\varphi}(n-1) + \mathbf{u}(n) \mathbf{u}^H(n) \quad (4.17)$$

$$\boldsymbol{\theta}(n) = \gamma \sum_{i=1}^{n-1} \gamma^{n-i-1} \mathbf{u}(i) d^*(i) + \mathbf{u}(n) d^*(n) = \gamma \boldsymbol{\theta}(n-1) + \mathbf{u}(n) d^*(n) \quad (4.18)$$

We can use the matrix inversion lemma methods to get the inverse matrix of $\boldsymbol{\varphi}(n)$ [42].

Assume a positive definite matrix \mathbf{A} can be expressed as:

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C} \mathbf{D}^{-1} \mathbf{C}^H \quad (4.19)$$

Where, \mathbf{A} and \mathbf{B} are positive definite M-by-M matrices. \mathbf{D} is the N-by-M positive-definite matrix and \mathbf{C} is an M-by-N matrix. The inverse matrix of \mathbf{A} can be calculated as:

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B} \mathbf{C} (\mathbf{D} + \mathbf{C}^H \mathbf{B} \mathbf{C})^{-1} \mathbf{C}^H \mathbf{B} \quad (4.20)$$

If we let \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{D} equivalent to following:

$$\begin{aligned}
\mathbf{A} &= \boldsymbol{\varphi}(n) \\
\mathbf{B}^{-1} &= \gamma \boldsymbol{\varphi}(n-1) \\
\mathbf{C} &= \mathbf{u}(n) \\
\mathbf{D} &= \mathbf{1}
\end{aligned}
\tag{4.21}$$

Substituting Eq. 4.21 to Eq.4.20, we can get:

$$\boldsymbol{\varphi}^{-1}(n) = \frac{1}{\gamma \boldsymbol{\varphi}(n-1)} - \frac{\boldsymbol{\varphi}^{-1}(n-1) \mathbf{u}(n) \mathbf{u}^H(n) \boldsymbol{\varphi}^{-1}(n-1)}{\gamma^2 + \gamma \mathbf{u}^H(n) \boldsymbol{\varphi}^{-1}(n-1) \mathbf{u}(n)}
\tag{4.22}$$

Let the inverse correlation matrix $\boldsymbol{\vartheta}(n) = \boldsymbol{\varphi}^{-1}(n)$, and then we have:

$$\boldsymbol{\vartheta}(n) = \frac{1}{\gamma} \boldsymbol{\vartheta}(n-1) - \frac{1}{\gamma} \mathbf{k}(n) \mathbf{u}^H(n) \boldsymbol{\vartheta}(n-1)
\tag{4.23}$$

where $\mathbf{k}(n)$ is the gain vector, and

$$\mathbf{k}(n) = \frac{\boldsymbol{\vartheta}(n-1) \mathbf{u}(n)}{\gamma + \mathbf{u}^H(n) \boldsymbol{\vartheta}(n-1) \mathbf{u}(n)}
\tag{4.24}$$

The procedure of updating estimate of Volterra filter coefficient vector $\hat{\mathbf{w}}(n)$ with the RLS algorithm can then be derived and is summarized in Table 3 [36, 42].

Table 3 RLS algorithm summary.

Initialization:

$$\hat{\mathbf{w}} = \mathbf{0}$$

$$\boldsymbol{\vartheta}[0] = \frac{1}{\delta} * I$$

I is $L \times L$ identity matrix

$$\gamma = 1$$

δ is a small positive constant

Update procedure (n=1, 2, 3...):

Gain Vector:

$$\mathbf{k}(n) = \frac{\boldsymbol{\vartheta}(n-1)\mathbf{u}(n)}{\gamma + \mathbf{u}^H(n)\boldsymbol{\vartheta}(n-1)\mathbf{u}(n)}$$

Estimation error:

$$e(n) = d(n) - \hat{\mathbf{w}}^H \mathbf{u}(n)$$

Coefficients update:

$$\hat{\mathbf{w}} = \hat{\mathbf{w}} + \mathbf{k}(n)e^*(n)$$

Correlation matrix inverse:

$$\boldsymbol{\vartheta}(n) = \frac{1}{\gamma} \boldsymbol{\vartheta}(n-1) - \frac{1}{\gamma} \mathbf{k}(n)\mathbf{u}^H(n)\boldsymbol{\vartheta}(n-1)$$

Chapter 5. Channel Estimation and Compensators Design

In this chapter, we will introduce different nonlinearity compensation methods based on the Volterra and Wiener-Hammerstein models. The first method is to represent the nonlinear transmission channel by the p th-order Volterra model, and then design the equalizer and predistorter based on its p th-order Volterra inverse system. The second approach is to determine the Volterra series based equalizer using the adaptive signal processing method. The third approach is to design the equalizer based on the Wiener-Hammerstein model using the adaptive signal processing method. Linear equalizer is also included in simulations for comparison. The estimation and compensation process is performed in the electrical domain, since an electronic signal processor is more affordable and convenient.

5.1 Compensator Design Based on Transmission Channel Model

The discrete-time model of the optical communication system is shown in the diagram below.

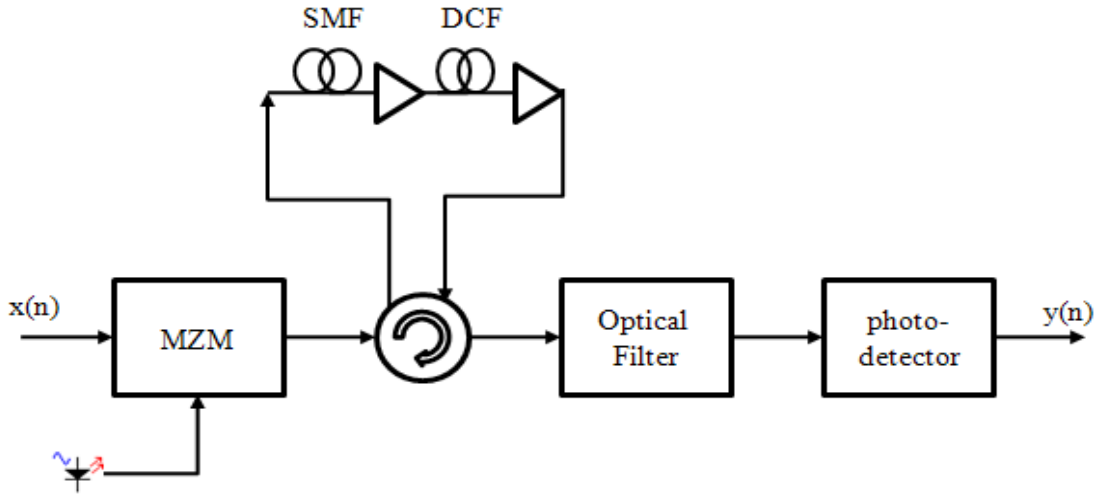


Figure 5.1 A nonlinear optical channel model.

The nonlinear channel includes the MZM, single mode fiber, amplifier, dispersion compensation fiber, optical filter and the photo-detector. The input signal to the channel is represented by $x(n)$, which is a sequence of complex valued 16-QAM encoded OFDM signals. The output signal $y(n)$ is the 16-QAM encoded OFDM signals after passing through the nonlinear channel.

Both RLS and LMS algorithm can be used to update the coefficients of the linear model and the nonlinear model, and both algorithms can give very close results. Without losing generality, we use the LMS algorithm to determine linear filter and the Wiener-Hammerstein filter coefficients and the RLS algorithm to determine the Volterra filter coefficients.

5.1.1 Compensator Design Based On the Linear Inverse Model

To demonstrate that a nonlinear equalizer can compensate the nonlinearity better than a linear equalizer, first of all, we use the linear model to model the optical channel. A discrete adaptive linear model is given in Eq.5.1.

$$\begin{aligned} y(n) &= \sum_{i=0}^{L-1} w_i(n)x(n-i) + e(n) \\ &= \hat{y}(n) + e(n) \end{aligned} \quad (5.1)$$

The total number of coefficients L is set at 30. $w_i(n)$ is the linear filter coefficients at time n . $e(n)$ is the difference between the linear filter estimation output $\hat{y}(n)$ and the actual system output signal $y(n)$. The input data to the linear filter can be expressed by the matrix as shown in Eq. 5.2, in which $x(n) = 0$ when $n \leq 0$.

$$U = \begin{bmatrix} x(1) & x(2) & \cdots & x(M) \\ x(0) & x(1) & \cdots & x(M-1) \\ \vdots & \vdots & \vdots & \vdots \\ x(-j) & x(-j+1) & & x(M-j) \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ x(-L+3) & x(-L+4) & \vdots & x(M-L+2) \\ x(-L+2) & x(-L+3) & \cdots & x(M-L+1) \end{bmatrix} \quad (5.2)$$

The coefficients vector for linear model is in Eq. 5.3:

$$\hat{\mathbf{w}} = [w_0(n), w_1(n), \dots, w_i(n), \dots, w_{L-1}(n)]^H \quad (5.3)$$

$\mathbf{u}(i)$ is the i th column ($i=1,2,3 \dots M$) in Eq.5.2

$$\mathbf{u}(i) = U(:, i) \quad (5.4)$$

The system desired output vector is:

$$D = [y(1), y(2), \dots, y(i), \dots, y(M)] \quad (5.5)$$

The output signals $y(n)$ at time n would equals to following equation:

$$\hat{y}(n) = \hat{\mathbf{w}}^H(n) \mathbf{u}(n) \quad (5.6)$$

There are two modes in the data transmission: *training mode* and *compensation mode*. The channel estimation is performed in the training mode. At the transmitter, a pre-determined 8192 bits training sequence is encoded into a 16 QAM format and then modulated by the OFDM modulation scheme. The resulting signals are denoted by $x(n)$, which is a complex valued signal representing the I and Q component of the OFDM signal. At the receiver, the received signal after coherent detection can be sampled and saved, and the signals are represented by $y(n)$.

We update the linear model coefficients vector $\hat{\mathbf{w}}$ in the training mode using the procedure as shown in Table 2 in section 4.1. The total number of coefficients is set at 30 ($L=30$). The filter coefficients vector $\hat{\mathbf{w}}$ is initialized to be zero. The variable ε is a small positive number. Once the coefficients vector $\hat{\mathbf{w}}$ is obtained, the system output signal, $y(n)$, can be compensated by the inverse linear system. The inverse of the linear channel model can be represented by the following equation:

$$\hat{x}(n) = [y(n) - \sum_{i=1}^{L-1} w_i \hat{x}(n-i)] / w_0 \quad (5.7)$$

In the compensation mode, we pass the received signals $y(n)$ through the linear inverse filter shown in Eq.5.7. The inverse filter output signal $\hat{x}(n)$ should approximate the actual system input signal $x(n)$.

5.1.2 Compensator Design based on p th-order Volterra Inverse Model

The discrete-time nonlinear optical system can be modeled by a bandpass Volterra model in the form of Eq.5.8 whose coefficients can be updated adaptively. The variables, $x(n)$ and $y(n)$, are the input and output signals indicated in Fig.5.1. The variable, $\hat{y}(n)$, is the Volterra model output and $e(n)$ is the estimation error. The block diagram for the bandpass Volterra model is shown in Fig.5.2. We can use those linear and cubic kernels to characterize the nonlinear system.

$$\begin{aligned}
 y(n) &= \sum_{i=0}^N h_i(n)x(n-i) + \sum_{i=0}^N \sum_{j=0}^N \sum_{k=0}^N h_{i,j,k}(n)x(n-i)x(n-j)x^*(n-k) + e(n) \\
 &= \hat{y}(n) + e(n)
 \end{aligned}
 \tag{5.8}$$

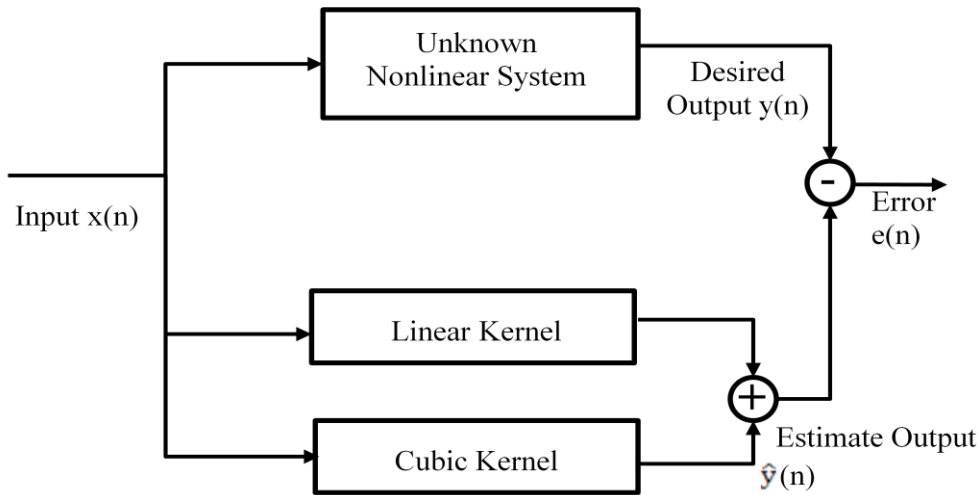


Figure 5.2 Third order Volterra model for optical nonlinear channel.

M is the total estimation sequence length. The numbers of linear and cubic kernels are calculated as following.

$$L_1 = N + 1 \tag{5.9}$$

$$L_3 = (N + 1)^3 \quad (5.10)$$

The total number of coefficients is:

$$L = L_1 + L_3 \quad (5.11)$$

The input data matrix is shown in Eq. 5.12, in which $x(n) = 0$ for $n \leq 0$.

$$U = \begin{bmatrix} x(1) & x(2) & \cdots & x(M) \\ x(0) & x(1) & \cdots & x(M-1) \\ \vdots & \vdots & & \vdots \\ x(1-N) & x(2-N) & & x(M-N) \\ x^2(1)x^*(1) & x^2(2)x^*(2) & \cdots & x^2(M)x^*(M) \\ x^2(1)x^*(0) & x^2(2)x^*(1) & & x^2(M)x^*(M-1) \\ \vdots & \vdots & & \vdots \\ x^2(1-N)x^*(1-N) & x^2(2-N)x^*(2-N) & \cdots & x^2(M-N)x^*(M-N) \end{bmatrix} \quad (5.12)$$

The estimate of model coefficients vector $\hat{\mathbf{w}}$ can be represented by:

$$\hat{\mathbf{w}} = [h_0(n), h_1(n), \dots, h_N(n), h_{000}(n), h_{001}(n), \dots, h_{0N0}(n), h_{0N1}(n), \dots, h_{NNN}(n)]^H \quad (5.13)$$

The system desired output vector is:

$$\mathbf{d} = [y(1), y(2), \dots, y(i), \dots, y(M)] \quad (5.14)$$

The variable $\mathbf{u}(i)$ denotes the i th column of the matrix U :

$$\mathbf{u}(i) = U(:, i) \quad (5.15)$$

The output signals at time n would equals to following equation:

$$\hat{y}(n) = \hat{\mathbf{w}}^H(n) \mathbf{u}(n) \quad (5.16)$$

Similar to the discussion in section 5.1.1, the channel estimation is performed in the training mode, as shown in Fig.5.3.

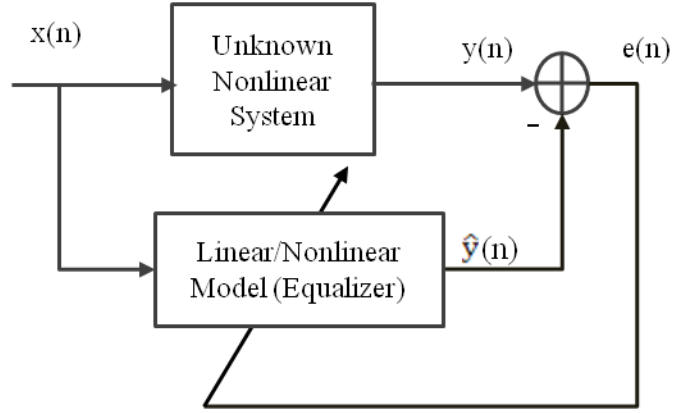


Figure 5.3 Channel estimation process (training mode).

The memory length we choose for the Volterra model is two. As a result, $N=2$ and the total number of coefficients is 30. We update the coefficients vector $\hat{\mathbf{w}}$ using the procedure summarized in Table 3 in the training mode. First, we initialize the coefficients vector to 0 and initialize matrix $\boldsymbol{\vartheta}$. \mathbf{I} is a 30×30 identity matrix and $\delta = 0.007$. We set $\gamma = 1$, after training data set of 8192 bits is sent, the nonlinear channel model is determined and the obtained model is used to design the equalizer with the p th-order inverse method. To get the Sparse Volterra model, we can use the algorithm described in Section 3.3 to obtain the most significant kernels of the previously determined full Volterra channel model. Only the significant kernels are determined with RLS algorithm after the first training model (the training sequence might be sent out periodically). Similarly, a p th-order inverse Sparse Volterra equalizer can be designed base on the p th-order Sparse Volterra model.

The p th-order inverse system based on the Volterra model can be found using the p th-order inverse method described in section 3.4. Since the second order Volterra operator is not presented due to system's bandpass nature, the third order operator of the p th-order inverse system can be simplified as follows.

$$K_3 = -H_1(H_3)H_1 \quad (5.17)$$

The detail procedure for getting the p th-order inverse system based on the Volterra series is summarized in Table 4.

Table 4 *Pth-order* inverse summary for the third order Volterra series.

Stage 1: Linear Inverse (n=1: M):

$$a(n) = [y(n) - \sum_{i=1}^N h_i a(n-i)] / h_0$$

Stage 2: Cubic Operator (n=1: M):

$$b(n) = \sum_{i=0}^N \sum_{j=0}^N \sum_{k=0}^N h_{i,j,k} a(n-i) a(n-j) a^*(n-k)$$

Stage 3: Linear Inverse (n=1: M):

$$c(n) = [b(n) - \sum_{i=1}^N h_i c(n-i)] / h_0$$

State 4: Final output (n=1: M):

$$\hat{x}(n) = a(n) - c(n)$$

To compensate the distorted signal using the *pth-order* Volterra inverse system, we process the received distorted signal $y(n)$ as described in Table 4. The output signal $\hat{x}(n)$ should also approximate the original sending signals $x(n)$. The development of *pth-order* inverse system based on the Sparse Volterra model follows the same procedure as the Full Volterra model.

As indicated in reference 40, the *pth-order* pre-inverse is identical to the *pth-order* post-inverse. Therefore, the procedure summarized in Table 4 can be used to design either the *pth-order* inverse equalizer or the predistorter. Fig.5.4 shows the *pth-order* pre-inverse K in tandem with the nonlinear system H .

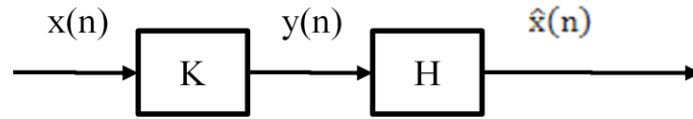


Figure 5.4 Tandem connection of the *pth-order* pre-inverse K and the nonlinear system H .

However, as shown in reference 21, the *pth-order* inverse predistorter changes the nonlinear system input. As a result, the nonlinear model based on which the *pth-order* inverse predistorter is designed might not be accurate after the predistorter is introduced since the nonlinear system

characteristics might depend on the input signal. To alleviate this problem, while designing a *pth-order* inverse predistorter, we need to conduct the channel modeling in several iterations. Once the *pth-order* inverse predistorter is designed and used to change input signal, the system model needs to be re-determined based on *pth-order* predistorter output and resulting system output. The new *pth-order* inverse predistorter is then designed based on the new nonlinear system model. The detailed design procedure of a *pth-order* Volterra predistorter is shown in Table 5:

Table 5 Predistorter design based on the *pth-order* pre-inverse Volterra system.

Stage 1: Initialize the transmission channel coefficients vector $\hat{\mathbf{w}} = [1, 0, \dots, 0, 0]^H$. The length of $\hat{\mathbf{w}}$ is 30. Perform channel estimation as described in section 5.1.2.

Stage 2: Develop *pth-order* inverse system based on the obtained Volterra channel model.

Stage 3: Perform channel estimation as described in section 5.1.2 based on the predistorter output and the corresponding nonlinear channel output. Update the model coefficients vector:

$$\hat{\mathbf{w}} = [h_0(n), h_1(n), \dots, h_N(n), h_{000}(n), h_{001}(n), \dots, h_{0N0}(n), h_{0N1}(n), \dots, h_{NNN}(n)]^H$$

Stage 4: Develop *pth-order* inverse system based on the new Volterra model.

Stage 5: Go back to Stage 3. End if the predistorter performance does not vary a lot.

In the compensation mode, at the transmitter, we pass the input signal $x(n)$ through the *pth-order* inverse predistorter. Then the predistorted signal modulates the optical carrier and is launched into the optical channel.

5.2 Adaptive Equalizer Design Based on Inverse Channel Estimation

Section 5.1 describes the transmission channel estimation and the compensator design based on the transmission channel model. In this section, the equalizer design based on the inverse channel estimation is presented.

The inverse channel estimation/equalizer design is carried out in the training stage. The desired equalizer output is the transmitted signal, while the input to the equalizer is the distorted signal at the receiver. The nonlinear channel is the same as the diagram presented in Fig.5.1. A pre-determined training sequence is sent and the distorted received signals are saved. The linear and nonlinear equalizers coefficients can be obtained by LMS or RLS.

Fig.5.5 shows the equalizer coefficients update process in the training mode for the equalizer design. $\hat{y}(n)$ is the equalizer output signal. $e(n)$ is the difference between the equalizer output and the desired output. After the training stage, the obtained linear model, Volterra model, and Wiener-Hammerstein model based equalizers are used to compensate the distorted output in the compensation mode. Fig.5.6 demonstrates the compensation mode, in which the linear or nonlinear equalizer is used to compensate the nonlinear signal distortions.

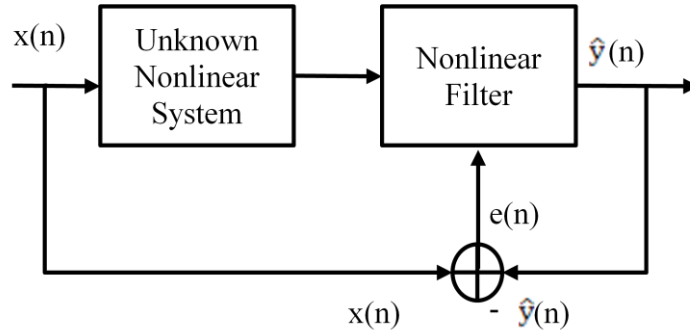


Figure 5.5 Inverse system determination process (training mode).

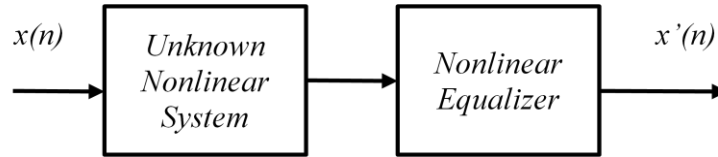


Figure 5.6 Equalizer compensation mode.

The Volterra model based equalizer coefficients can be determined in the training mode. The orthogonal search method discussed in section 3.3 is used to derive the Sparse Volterra model based equalizer from the full Volterra model based equalizer. The adaptive filter algorithm needs to be modified to derive the Wiener-Hammerstein model based equalizer and the rest of this section is dedicated to the determination of a Wiener-Hammerstein model based equalizer.

The input-output relations of an adaptive Wiener-Hammerstein model consisting of the first FIR filter, the nonlinear filter and the second FIR filter is given in Eqs.5.18-5.20.

$$y(n) = \sum_{i=0}^{M_1-1} u_i(n)x(n-i) \quad (5.18)$$

$$z(n) = v_0(n)y(n) + v_1(n)y^2(n)y^*(n) \quad (5.19)$$

$$o(n) = \sum_{i=0}^{M_2-1} w_i(i)z(n-i) \quad (5.20)$$

where $M_1 - 1$ is the first FIR filter memory length, $M_2 - 1$ is the second FIR filter memory length and $x(n)$ is the Wiener-Hammerstein model input.

The difference between desired signal $d(n)$ and the filter output $o(n)$ can be represented by:

$$e(n) = d(n) - o(n) \quad (5.21)$$

The coefficients can be represented in a vector form as shown in Eqs.5.22~5.24:

$$\mathbf{u}(n) = [u_0(n), u_1(n), \dots, u_{M_1-1}(n)]^T \quad (5.22)$$

$$\mathbf{v}(n) = [v_0(n), v_1(n)]^T \quad (5.23)$$

$$\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{M_2-1}(n)]^T \quad (5.24)$$

The input signals of the first FIR filter, the nonlinear filter and the second FIR filter are shown in Eqs.5.25~5.27.

$$\mathbf{x}(n) = [x(1), x(2), \dots, x(n - M_1 + 1)]^T \quad (5.25)$$

$$\mathbf{y}(n) = [y(n), y^2(n)y^*(n)]^T \quad (5.26)$$

$$\mathbf{z}(n) = [z(1), z(2), \dots, z(n - M_2 + 1)]^T \quad (5.27)$$

The output signal after the first FIR filter, the nonlinear filter and the second FIR filter can be calculated by:

$$y(n) = \mathbf{u}^T(n)\mathbf{x}(n) \quad (5.28)$$

$$z(n) = \mathbf{v}^T(n)\mathbf{y}(n)$$

(5.29)

$$o(n) = \mathbf{w}^T(n)\mathbf{z}(n)$$

(5.30)

The memory length for both linear filters is 1. We use the joint normalized LMS (joint NLMS) algorithm to update the linear and nonlinear subsystem coefficients of the Wiener-Hammerstein model jointly [41]. The NLMS method needs to be modified for the complex Wiener-Hammerstein model determination [43]. The detailed procedure is summarized in Table 6 [41, 43]. The difficulty in the Wiener-Hammerstein model coefficients estimation is that the coefficients update is dependent among each subsystem [41]. The estimation is very sensitive to the coefficients initial condition and the step size. To solve this problem, we perform the update procedure in several iterations, making the current coefficients vector as the next estimation starting point, as shown in Table 6.

Table 6 LMS for the Wiener-Hammerstein model coefficients update.

Initialization:

$$\mathbf{u}(0) = \sigma$$

$$\mathbf{v}(0) = \sigma$$

$$\mathbf{w}(0) = \sigma$$

σ is a positive small constant.

Loop: Update procedure (n=1, 2, 3...):

$$\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-M_1+1)]^T$$

$$y(n) = \mathbf{u}^T(n)\mathbf{x}(n)$$

$$\mathbf{y}(n) = [y(n), y^2(n)y^*(n)]^T$$

$$\mathbf{z}(n) = \mathbf{v}^T(n)\mathbf{y}(n)$$

$$\mathbf{z}(n) = [z(n), z(n-1), \dots, z(n-M_2+1)]^T$$

$$o(n) = \mathbf{w}^T(n)\mathbf{z}(n)$$

$$e(n) = d(n) - o(n)$$

$$\mathbf{a}(n) = [1, 2y(n)y^*(n)]^T$$

$$\mathbf{b}(n) = \mathbf{v}^T(n)\mathbf{a}(n)$$

$$\mathbf{x}_n(n) = \mathbf{b}(n)\mathbf{x}(n)$$

$$\mathbf{X}(n) = [\mathbf{x}_n(n), \mathbf{x}_n(n-1), \dots, \mathbf{x}_n(n-M_2+1)]$$

$$\mathbf{p}(n) = \mathbf{X}(n)\mathbf{w}(n)$$

$$\mathbf{Y}(n) = [\mathbf{y}(n), \mathbf{y}(n-1), \dots, \mathbf{y}(n-M_2+1)]$$

$$\mathbf{q}(n) = \mathbf{Y}(n)\mathbf{w}(n)$$

$$\mathbf{u}(n+1) = \mathbf{u}(n) + (\alpha_u/(\|\mathbf{p}(n)\|^2 + \delta))\mathbf{p}^*(n)e(n)$$

$$\mathbf{v}(n+1) = \mathbf{v}(n) + (\alpha_v/(\|\mathbf{q}(n)\|^2 + \delta))\mathbf{q}^*(n)e(n)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + (\alpha_w/(\|\mathbf{z}(n)\|^2 + \delta))\mathbf{z}^*(n)e(n)$$

$\alpha_u, \alpha_v, \alpha_w, \delta$ are positive small constant.

Go back to **Loop** (*end* after several iterations)

Chapter 6.Simulation System

The CO-OFDM system is numerically simulated by a commercial optical simulation tool, OptiSystem 8.0. The simulation diagram is illustrated in Fig.6.1.

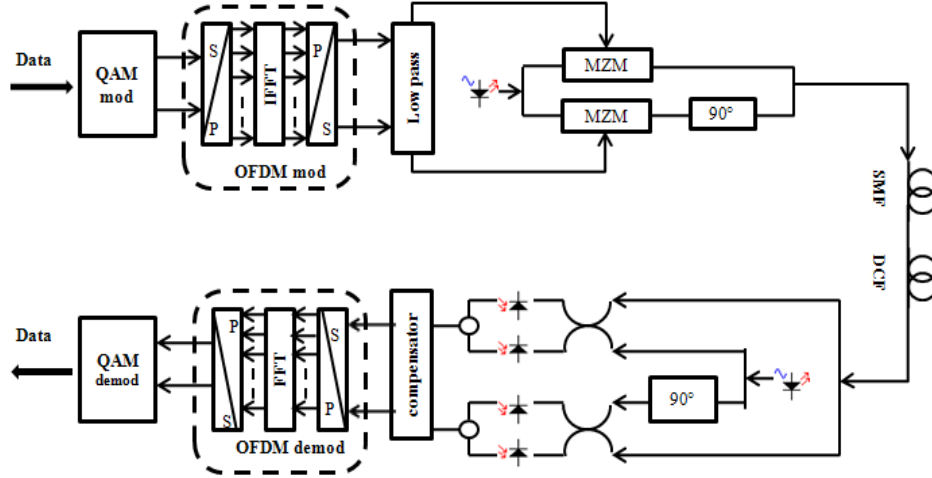


Figure 6.1 Block diagram for numerical simulation of CO-OFDM system.

The transmission bit rate is 100Gbit/s. On the transmitter side, the bit stream is generated using a pseudo random binary sequence generator, and the data is mapped by a 16-QAM encoder. The 16-QAM encoders with Gray coding has three different amplitude levels, as shown in Fig. 6.2.

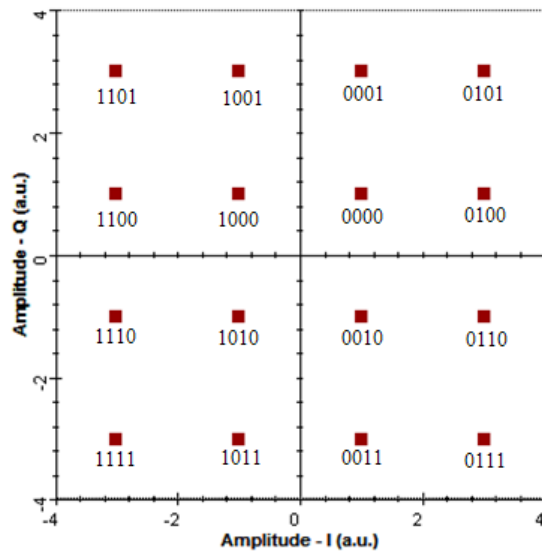


Figure 6.2 16-QAM encoders with Gray coding.

The information stream is further parsed into 128 low speed parallel data subcarriers and present to an Inverse Fast Fourier Transform (IFFT) IFFT processor. Finally, cyclic prefix is added to ensure correct data recovery. A baud rate of 25 Gbaud/s OFDM in-phase and quadrature parts are then filtered by low pass filters, and their spectra are shown in Fig.6.3. The resulting electrical signal then modulates optical carrier using a pair of Mach-Zehnder modulators. The resulting optical signal spectrum is shown in Fig.6.4. The laser line width is set at 1 MHz, with adjustable launch powers and no phase offset.

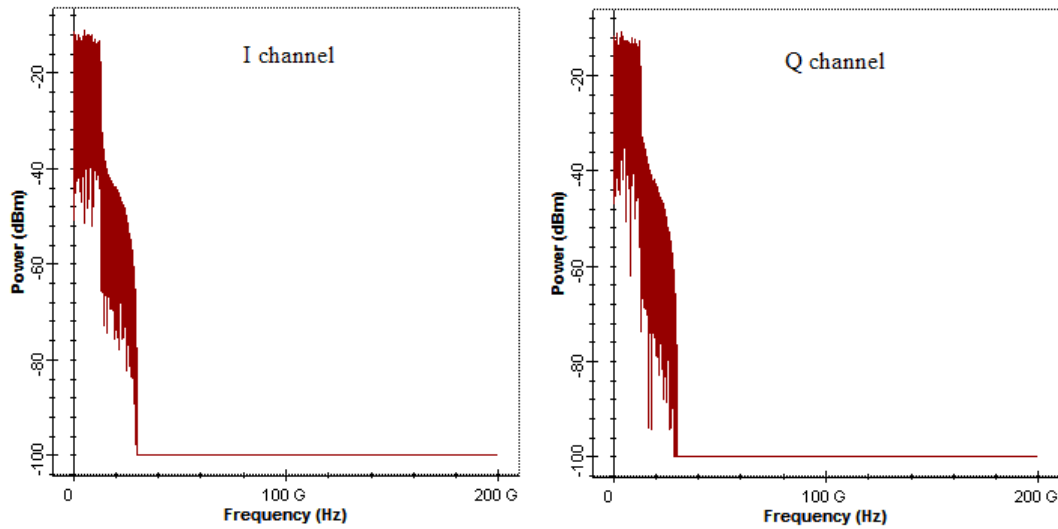


Figure 6.3 I/Q channel RF OFDM spectrum.

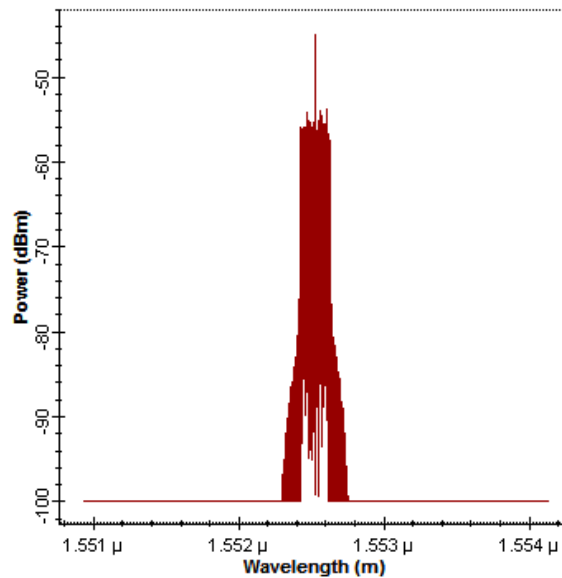


Figure 6.4 Optical OFDM spectrum after a pair of MZMs.

The optical channel consists of 10 spans of 80 km SMF, with attenuation=0.2 dB/km, dispersion =16 ps/nm/km and nonlinearity coefficients =2.09 /w/km. Fiber dispersion is fully compensated by a 16 km of DCF in each span which has 0.6 dB/km attenuation, -80 ps/nm/km dispersion and 6.4 w/km nonlinearity. The combined SMF and DCF attenuation is balanced by a 4 dB noise figure optical amplifier in each loop. As shown in Fig. 6.5, after transmitting through 800km of optical fibers, the signal spectrum becomes very noisy. Optical Signal to Noise Ratio (OSNR) is measured at the end of the transmission channel.

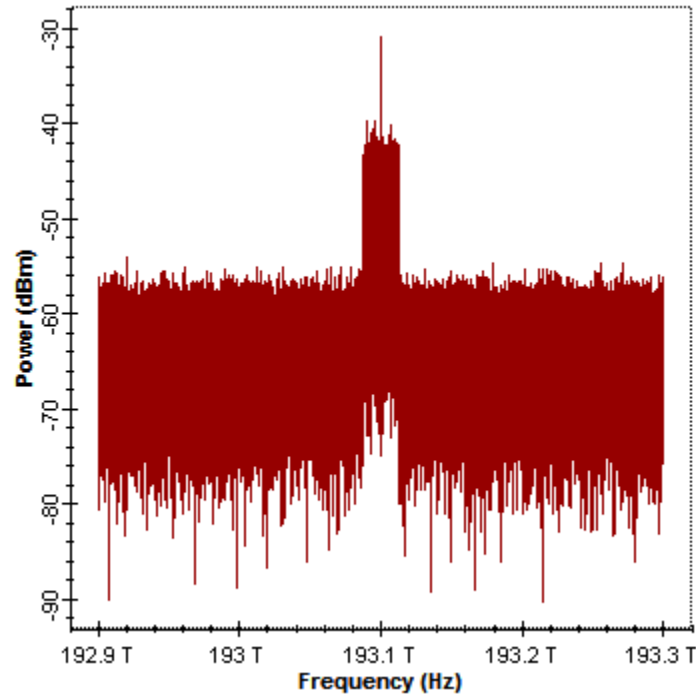


Figure 6.5 Optical OFDM spectrum after 800 km of optical fiber transmission.

Amplified spontaneous emission (ASE) noise is reduced by an optical filter at the receiver. The LO laser is assumed to be perfectly aligned with the power set at -2dBm and line-width equals to 1MHz. The I/Q components of the OFDM signal is recovered by a 2×4 90 degree optical hybrid and a pair of photo-detectors. Photo-detector noise, such as thermal noise, shot noise, dark current and ASE noise are included in the simulation. Electrical amplifiers are included to adjust the received signal intensity. The converted OFDM RF signal is demodulated using FFT processor and the guarding interval is removed. The obtained signals are then fed into a QAM

decoder. The bit error ratio (BER) is calculated at the end of the receiver. The starting and ending OFDM symbols are ignored in the BER calculation.

The equalizer of the optical channel can be in the form of a linear filter, Volterra model, or Wiener-Hammerstein model. The predistorter based on the *pth-order* inverse theory is in the form of a Volterra model. The compensation is done in MatlabTM. For the equalizer, the MatlabTM compensation component is located after the photo-detector and before the OFDM demodulator. The desired output is the data saved after OFDM modulating, and the received output is the data saved before the OFDM demodulation. For the predistorter, the MatlabTM component is located after the OFDM modulator and before the MZM modulation. A training sequence is first sent to obtain the equalizer tap weights. Once the tap-weights are obtained, the coefficients are used in the MatlabTM component to compensate different bit sequences. The modeling and equalizer design tasks are implemented with MatlabTM based on data generated by the OptiSystem.

In the WDM system, 5 channels of 100Gb/s 16-QAM OFDM signals are transmitted, with 100GHz channel spacing. In each channel, the modulation parameter has the same settings as the single channel system. The five optical carrier frequencies are set at 192.9THz, 193.0THz, 193.1THz, 193.2THz and 193.3THz respectively. The bit stream for each channel is different. The transmission length is reduced to 9×80km. The signal spectrum of the five-channel WDM CO-OFDM system at the transmitter is shown in Fig. 6.6.

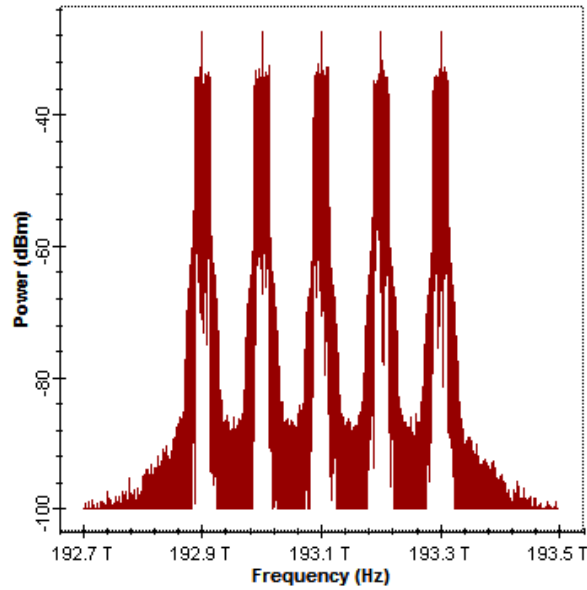


Figure 6.6 Five channel WDM CO-OFDM signal at the transmitter.

At the receiver, each channel is extracted by bandpass optical filters with different center frequencies. Then, the obtained optical signal in each channel is converted back to an electrical signal by photo-detectors separately. BER is calculated at the end of receiver. Only the central channel performance is evaluated.

Chapter 7. Results and Discussions

To evaluate the nonlinear effect of the optical fiber, we assume a dispersion fully-compensated fiber transmission system whose diagram was described in Chapter 6. The signal constellation after 800 km fiber transmission, with different launch power is shown in Fig.7.1. According to the SPM formula, there is no doubt that a higher launch power would lead to a larger nonlinearity distortion. As shown in Fig.7.2, when the fiber transmission distance increases, the distortion increases. This simulation result is consistent with the discussions in section 2.4 that the phase shift caused by fiber nonlinearity is positively related with fiber transmission length.

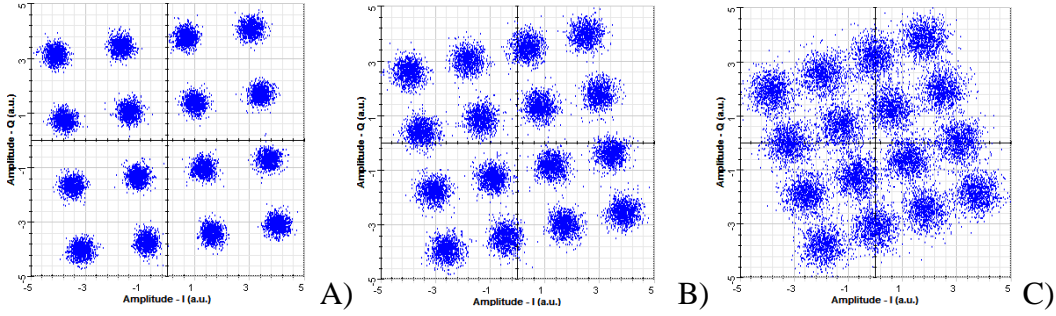


Figure 7.1 Output signal constellation of 16 QAM OFDM system after 800km transmission with different launch power. A) 1dBm, B) 3dBm, C) 5dBm.

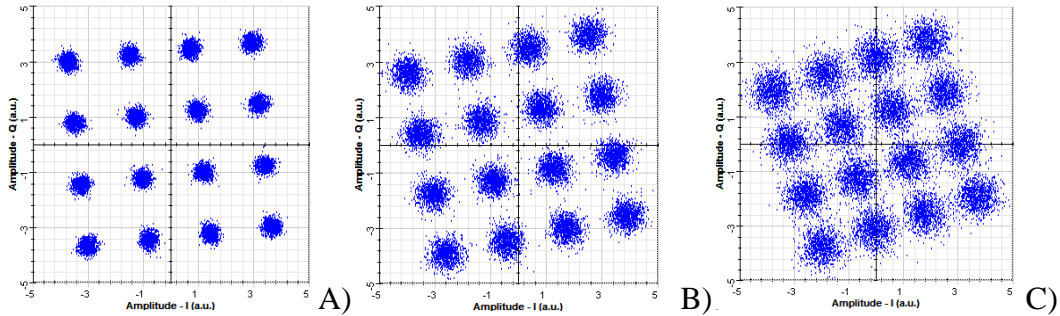


Figure 7.2 Output signal constellation of 16 QAM OFDM system at 3dBm launch power with after different transmission length. A) 400km, B) 800km, C) 1200km.

7.1 Equalization by Full Volterra and Sparse Volterra System

This section describes the system performance with full Volterra equalizer and Sparse Volterra equalizer. The full Volterra equalizer coefficients are determined with a training sequence and the RLS algorithm. The sparse Volterra equalizer is derived based on the full Volterra equalizer.

For comparison purpose, the adaptive linear equalizer is also included in the simulation to

evaluate the performance of the nonlinear equalizers. The received signal constellation diagram after 800 km of fiber transmission, with 2 dBm launch power is shown in Fig.7.3. Due to SPM, ASE noise and photo-detector noise, the constellation diagram has become scattered and has phase and amplitude distortions. The linear kernels account for the attenuation and the dispersion effect of fiber, while the third order kernels can account for the interaction between ASE noise and signal and nonlinear distortion [13]. Since linear equalizer has no nonlinear terms, its capability of removing the phase noise introduced by fiber nonlinearity is restricted. As shown in the constellation diagram in Fig.7.3, there is no doubt that nonlinear equalizers outperform the linear equalizer. The sparse Volterra equalizer with only 8 kernels and full Volterra equalizer with 30 kernels have comparable performance as shown in Fig. 7.3.

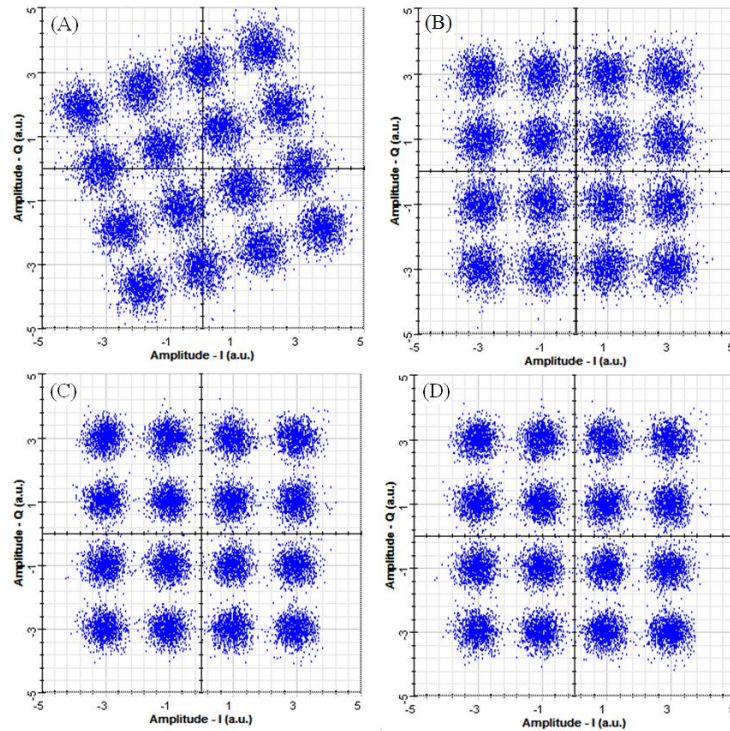


Figure 7.3 Output signal constellation of 16 QAM OFDM system (A) w/o equalizer, (B) with linear equalizer (C) with full Volterra equalizer, (D) with sparse Volterra equalizer.

The Monte Carlo simulations are conducted to evaluate the equalizer effectiveness on the OFDM system after 800 km transmission under different launch power. The resulting BER plot is shown in Fig.7.4. At low launch power, the OFDM system with or without compensation have similar BER, and full Volterra equalizer appears to have similar performance as the linear equalizer. The reason is that, under low input power level, the transmission fiber can be modeled

as a linear filter [13], the linear dispersion is the dominating factor (which is fully compensated by DCF in the simulation diagram) and the fiber nonlinearity effect is weak. The low OSNR at the receiver accompanied with the low launch power also limits the performance of nonlinear equalizer. When launch power increases, the system BER decreases at the first and then increases when launch power is higher than the “optimal” launch power. As illustrated in Fig.7.4, OFDM systems with (or without) different equalizers have different “optimal” launch power and BER values. The OFDM system with the nonlinear equalizer can take higher launch power and reach lower BER.

The BER increase under high launch power is caused by a larger SPM and ASE noise. Peddanarappagari and Brandt-Pearce have shown that at a higher input signal power, for a fixed input pulse width, the detector output pulse width increases with higher peak pulse power and makes the system more sensitive to nonlinearity distortion [9]. As a result, the low order nonlinear equalizer with a short memory span is not able to fully compensate the nonlinear channel. Ideally, a better result can be obtained by increasing the filter order and memory length of the Volterra equalizer. However, the resulting increased equalizer implementation complexity might not be acceptable.

We repeat the Monte Carlo simulation to calculate the BER of OFDM systems with the full Volterra equalizer and the sparse Volterra equalizer and the resulting BER curves are shown in Fig.7.5. As indicated in Fig.7.5, sparse Volterra and full Volterra equalizer have similar performances although the sparse Volterra equalizer has fewer coefficients.

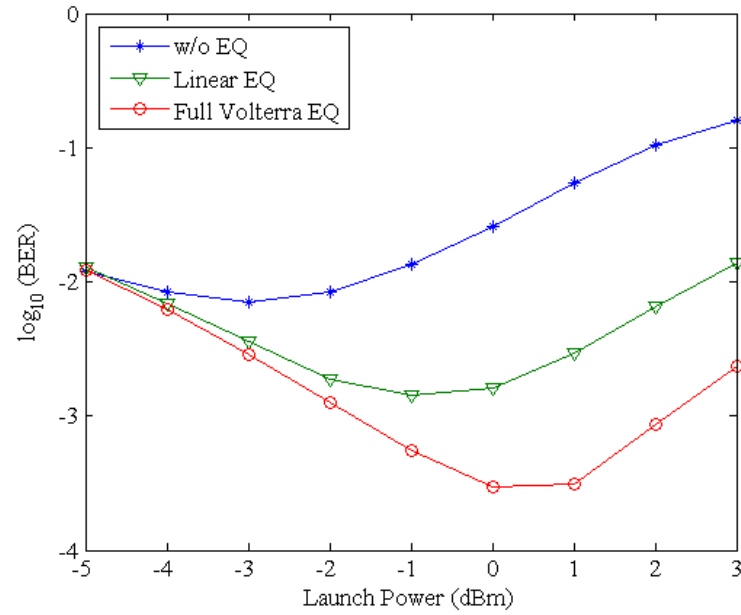


Figure 7.4 BER of 16 QAM OFDM system w/o compensation and with linear/nonlinear compensation as a function of fiber launch power.

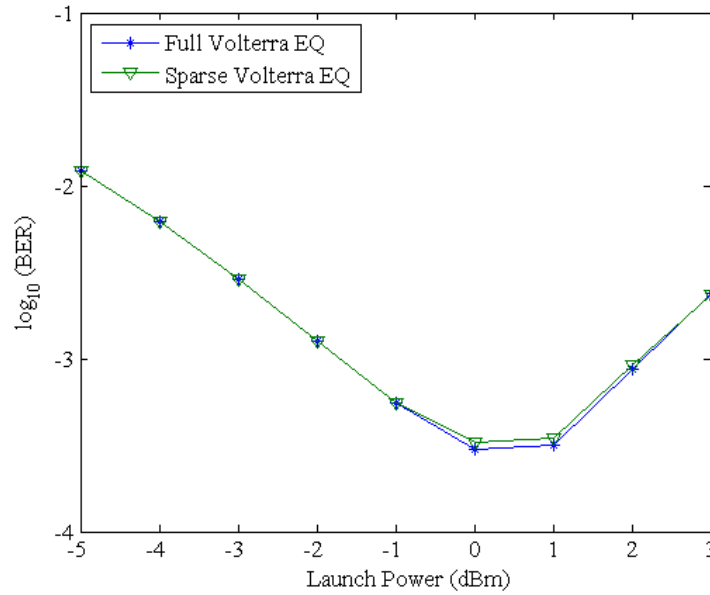


Figure 7.5 BER of 16 QAM OFDM system with full Volterra/sparse Volterra equalization as a function of fiber launch power.

Fig.7.6 shows the BERs of OFDM systems with no compensation and with linear, full Volterra, and sparse Volterra equalizers at different OSNR under 0dBm launch power. Additional ASE

noise is added at the end of transmission to set different OSNR values. It is not surprising that with the increase of OSNR, the system would have a better performance. The outperformance of nonlinear compensators becomes more evident with the increase of OSNR, since the signal becomes less distorted and the estimation becomes more accurate. The sparse Volterra and full Volterra equalizers have similar performance as shown in Fig.7.6 as well.

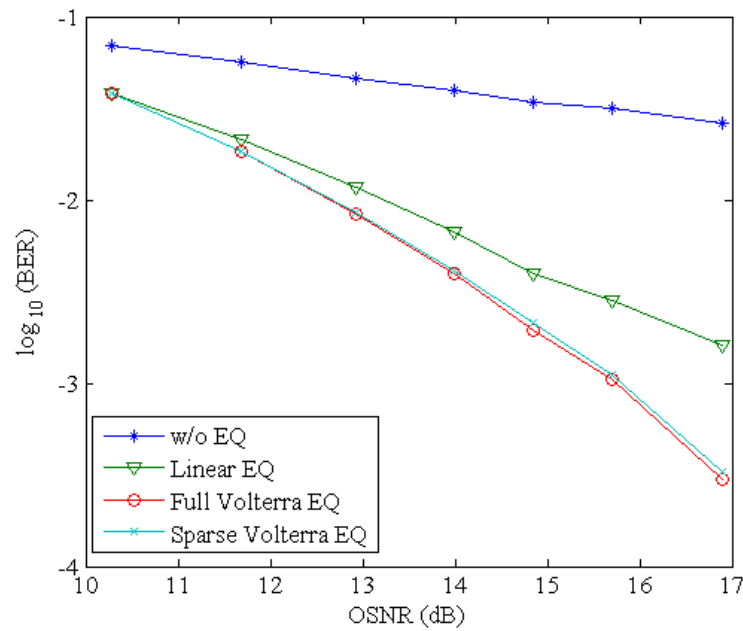


Figure 7.6 BER of 16 QAM OFDM system w/o compensation and with linear/nonlinear equalization as a function of OSNR (fixed launch power).

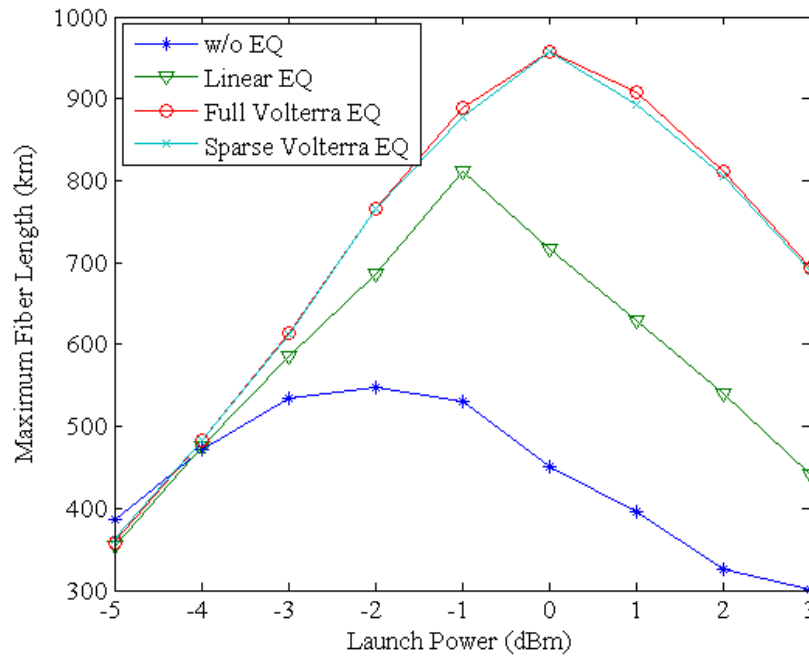


Figure 7.7 Optimal fiber transmission distance at 10^{-3} BER vs. launch power with and w/o compensation.

Simulations are also conducted to investigate reach limits of different OFDM systems under different launch power to guarantee 10^{-3} BER and the results are included in Fig.7.7. At lower launch power, the OFDM system with linear/nonlinear compensator can transmit similar distance as the OFDM system without compensation. The maximum reach of the OFDM system occurs at the same launch power in Figs.7.4 and 7.5 which gives the lowest BER value. With the nonlinear equalization, the OFDM system can approach 1000km transmission distance at 0dBm launch power. As shown in Fig.7.7, the OFDM system with nonlinear compensator can take higher launch power and reach longer distance. At launch power higher than the “optimal” launch power, the linear and nonlinear equalizers can support less transmission length because of high fiber nonlinearity.

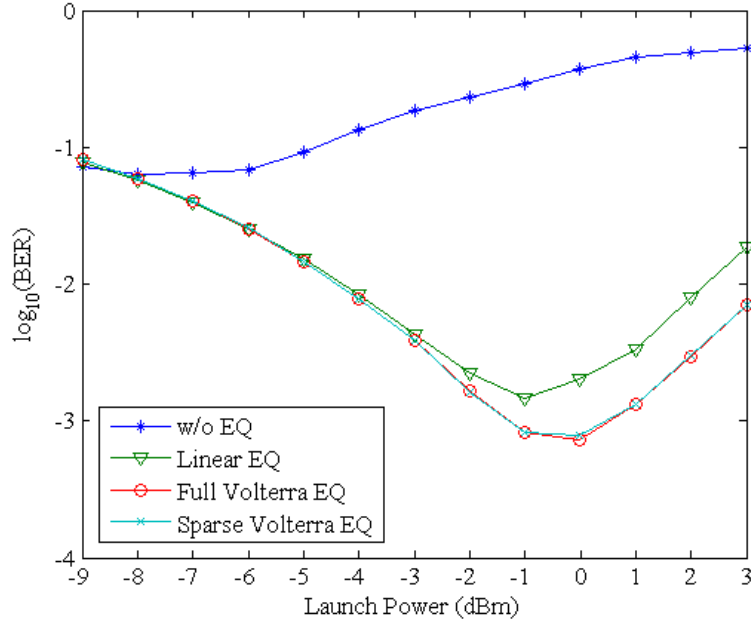


Figure 7.8 BER of a 5 channel WDM 16-QAM OFDM system w/o compensation and with linear/nonlinear compensation as a function of fiber launch power.

The BER vs. (launch power/per channel) simulation is conducted for 5-channel WDM CO-OFDM systems with or without compensation. The transmission distance is reduced to 9×80 km instead of 10×80 km. The BER of the center wavelength is calculated and simulation results are shown in Fig. 7.8. Comparing Fig. 7.8 with Fig. 7.4, one can see that, even with shorter transmission distance, the BER performance of the WDM CO-OFDM system is still worse than the single channel CO-OFDM system. The reason is that, in the WDM system, XPM leads to amplitude distortion and timing jitters [44]. Furthermore, XPM is proportional to the optical power [34]. With increased total launch power, the BER of WDM CO-OFDM system increases due to the higher fiber nonlinearity. The WDM CO-OFDM system with linear or nonlinear compensator show similar tendency as shown in Fig. 7.8. The nonlinear equalizer still shows a better performance, indicating that the Volterra equalizer can be used to compensate intra-channel nonlinearity in a WDM system. The sparse Volterra equalizer also functions well with fewer equalizer taps.

7.2 Equalization by p th-order Volterra Inverse System

To design the p th-order Volterra inverse equalizer, we need to get the p th-order Volterra channel model at first. This can be accomplished by transmitting a predetermined training sequence and estimating the dependence of channel output signals on the training sequence [21]. We used a third-order bandpass Volterra model with memory length of two to model the CO-OFDM system. For comparison purpose, we also use a finite-impulse response (FIR) linear filter to model the same system and then design the equalizer based on the linear inverse of the FIR filter.

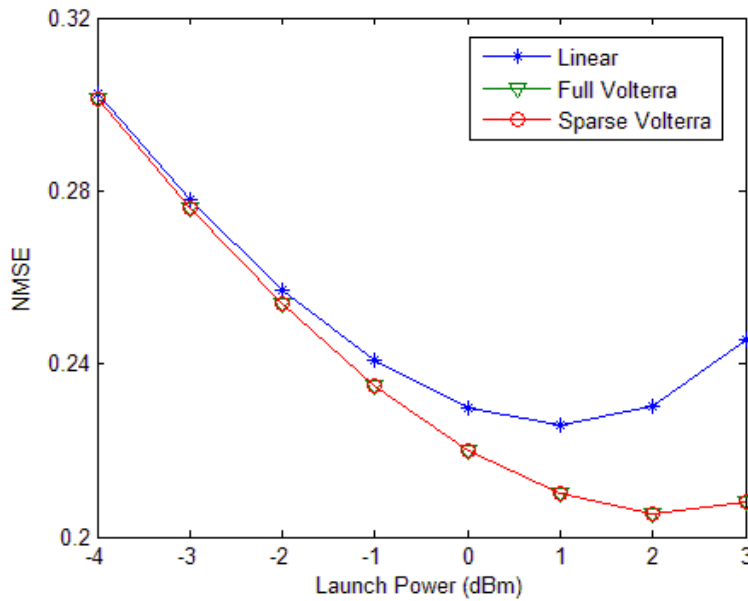


Figure 7.9 NMSE of different channel model of the CO-OFDM system versus launch power after 800km of transmission.

Figure 7.9 shows the normalized mean square error (NMSE) of the different channel models. NMSE quantizes the difference between model output and the actual system output. The smaller the NMSE value is, the more accurate the channel model would be. Overall speaking, the nonlinear Volterra model gives a better NMSE than the linear model. The sparse Volterra model is derived based on the full Volterra model. And the sparse Volterra model which has fewer coefficients has similar NMSE as the full Volterra model. At lower launch power, linear and Volterra models have comparable NMSE because the large signal noise prohibits accuracy channel modeling. When input power increases, the nonlinear models outperform linear model.

However, when the launch power is too high ($>2\text{dBm}$), Volterra model modeling precision deteriorates; since the signal is severely distorted by high fiber nonlinearity under high launch power and a Volterra model with higher order might be necessary. However, the Volterra model still handle the modeling better than the linear model.

After obtaining the pth -order full and sparse Volterra models and the linear model, we can design the pth -order inverse Volterra equalizer based on the Volterra models. The linear equalizer is simply the inverse system of the linear model. The design of pth -order inverse sparse Volterra equalizer follows the same procedure as the full Volterra model. After the equalizers have been designed, the system performance before and after different kinds of equalization are studied and compared.

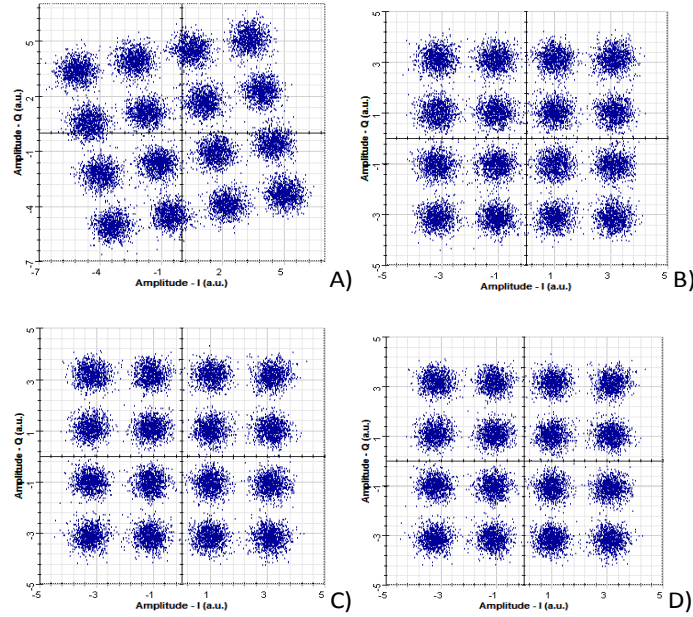


Figure 7.10 Received signal constellation of 16 QAM OFDM systems under 0 dBm launch power. (A) w/o equalizer, (B) with linear inverse equalizer (C) with pth -order full Volterra inverse equalizer, (D) with pth -order sparse Volterra inverse equalizer.

Figure 7.10 shows the received signal constellations of the CO-OFDM system without compensator, with linear compensator, with nonlinear compensator based on full Volterra model, and with nonlinear compensator based on sparse Volterra model. A 2^{16} bit pseudo-random sequence different from the training sequence is used to evaluate the equalizer performance. The transmission distance is 800km and the launch power is 0 dBm. The number of coefficients of the linear model, full Volterra model, and sparse Volterra model are 30, 30, and 18 respectively.

Similar to the constellation diagram shown in section 7.1, the nonlinear compensators can rotate and condense received signal constellations. The p th-order Volterra inverse equalizer gives a better constellation than the linear filter because the nonlinear filter can compensate the nonlinear distortions. P th-order inverse equalizer based on sparse Volterra model delivers performance similar to the p th-order inverse equalizer based on the Full Volterra model with fewer coefficients.

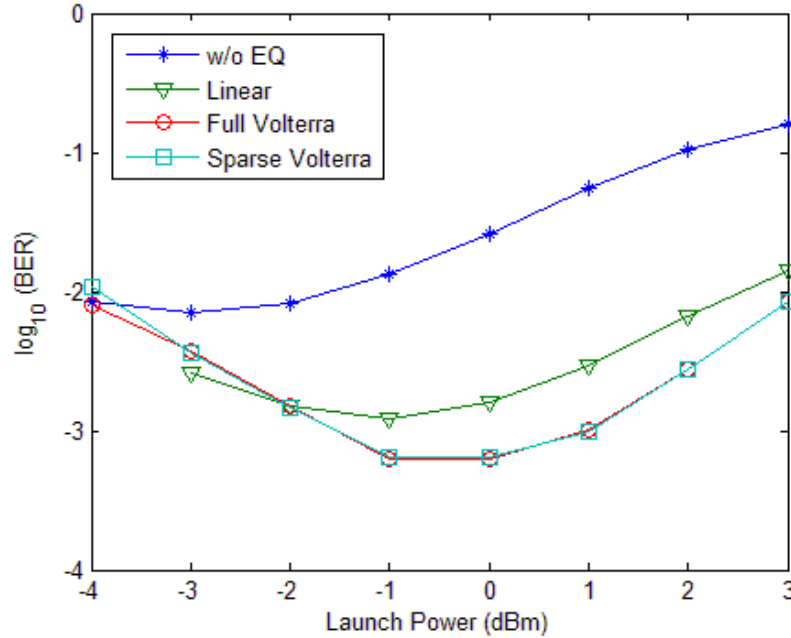


Figure 7.11 BER versus launch power of the CO-OFDM system with or without compensation after 800km transmission.

The bit error rate (BER) of the CO-OFDM system without/with different compensators vs. launch power is illustrated in Fig.7.11. When launch power is low, the linear and nonlinear compensators do not improve the CO-OFDM system performance because the models based on which the equalizers are designed are not accurate. When launch power is higher than -3 dBm, the equalizers can improve the CO-OFDM system performance and the BER of CO-OFDM system with compensation begins to decrease. When launch power is higher than 0 dBm, the system BER begins to increase because the CO-OFDM system becomes highly nonlinear under high input power, a nonlinear compensator with higher order is necessary. The “optimal” launch power for the CO-OFDM system with compensation is between -1 and 0 dBm. Fig.7. 11 also shows that when launch power is higher than -1 dBm, the nonlinear compensators outperform the

linear compensator. This observation is consistent with Fig.7.9 which shows that when launch power is higher than -1dBm , the nonlinear models outperform linear model. Fig.7.11 also demonstrates that the nonlinear compensator based on full Volterra and sparse Volterra models has similar performance.

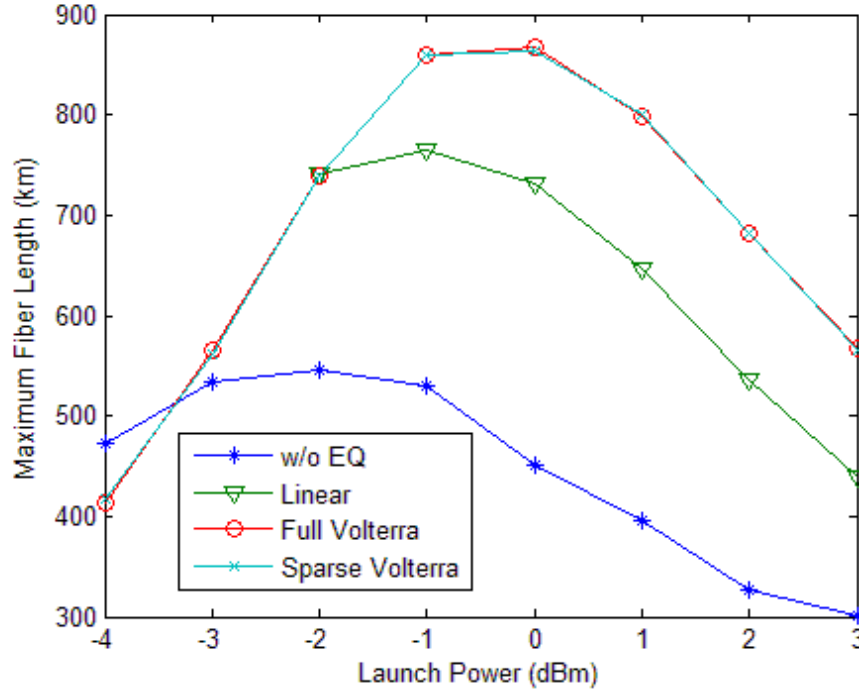


Figure 7.12 Maximum transmission length of the OFDM system with or without compensation as a function of launch power.

Figure 7.12 illustrates the maximum possible transmission length achieved by the CO-OFDM system with/without compensation while maintaining 10^{-3} BER. At lower launch power, the fiber transmission length for the system without compensation is slightly higher than the system with compensation. The explanation is that the linear/nonlinear equalizers based on inaccurate channel models generated under high noise cannot compensate channel distortion. The linear filter and nonlinear filter has similar performance at launch power less than -2 dBm since the nonlinear effect is weak at low launch powers. When launch power increases, the system with a nonlinear compensator has a better performance. However, under high launch powers the nonlinear effect is severe, the nonlinear compensator with higher order is necessary and the maximum reach of the optical system with nonlinear compensator decreases although the nonlinear compensator still outperforms linear compensator.

7.3 Compensation by p th-order Volterra Inverse Equalizer and Predistorter

In this section, we present the results of the p th-order Volterra inverse predistorter and equalizer. The difference between the predistorter and equalizer is studied and discussed. Because the p th-order Volterra inverse predistorter is applied to the signals before entering the nonlinear channel, when the p th-order Volterra inverse is used, the actual input to the nonlinear system is changed.

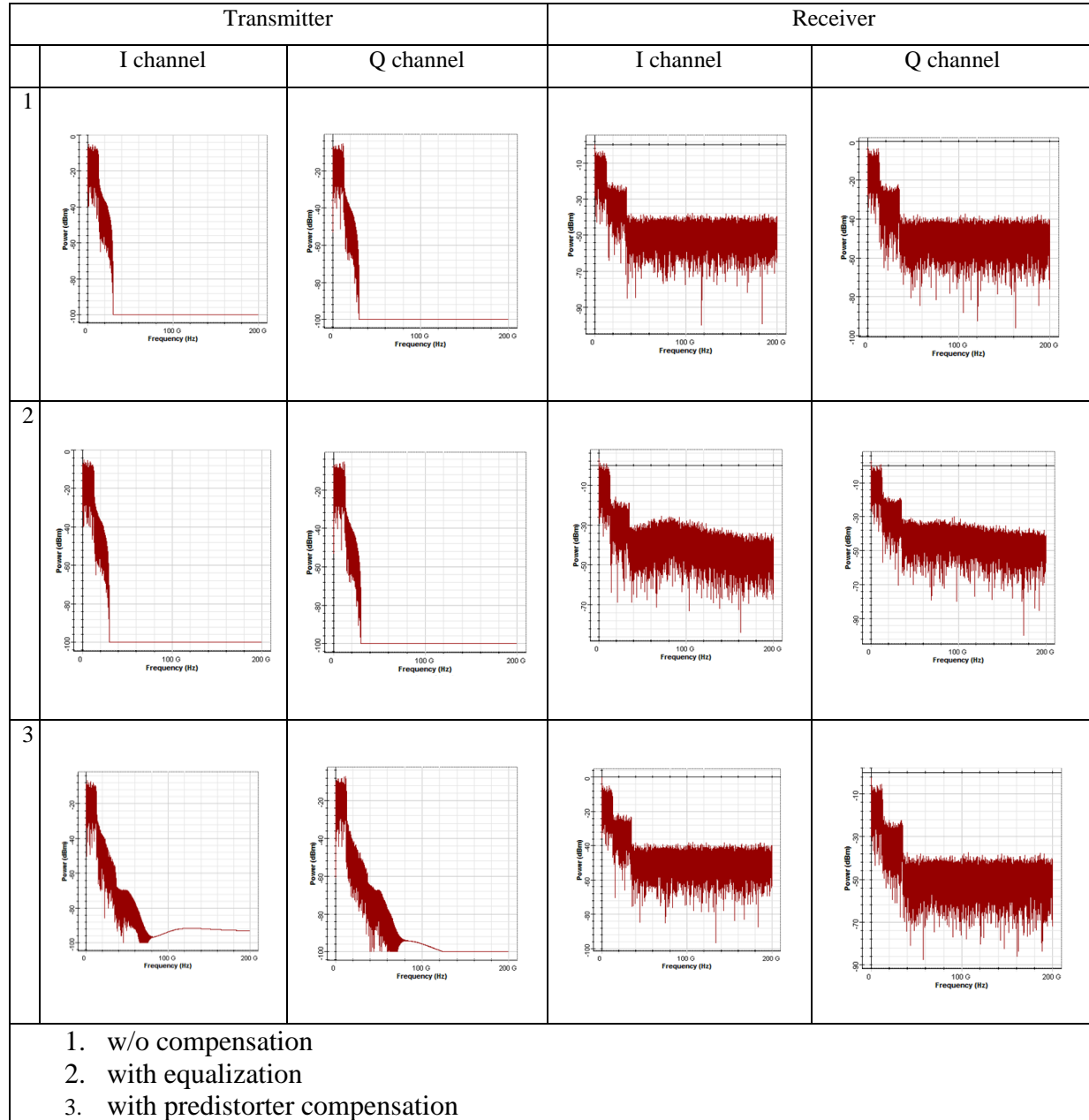


Figure 7.13 Electrical signal spectrum of the OFDM system without compensation, with equalization at the receiver and with predistortion at the transmitter.

Figure 7.13 compares the I/Q channel spectrum of the OFDM system without compensation, with equalizer compensation and with predistorter compensation. At the transmitter, the I/Q channel spectrum without compensation and with p th-order Volterra inverse equalization are identical. After applying the p th-order inverse equalizer, the signal spectrum at the receiver has a noticeable change at around 90GHz. In contrast, the p th-order Volterra inverse predistorter would change the signal spectrum at the transmitter and the signal spectrum at the receiver is similar to the spectrum without compensation. The performance difference between the predistorter and the equalizer is evaluated by the BER results later.

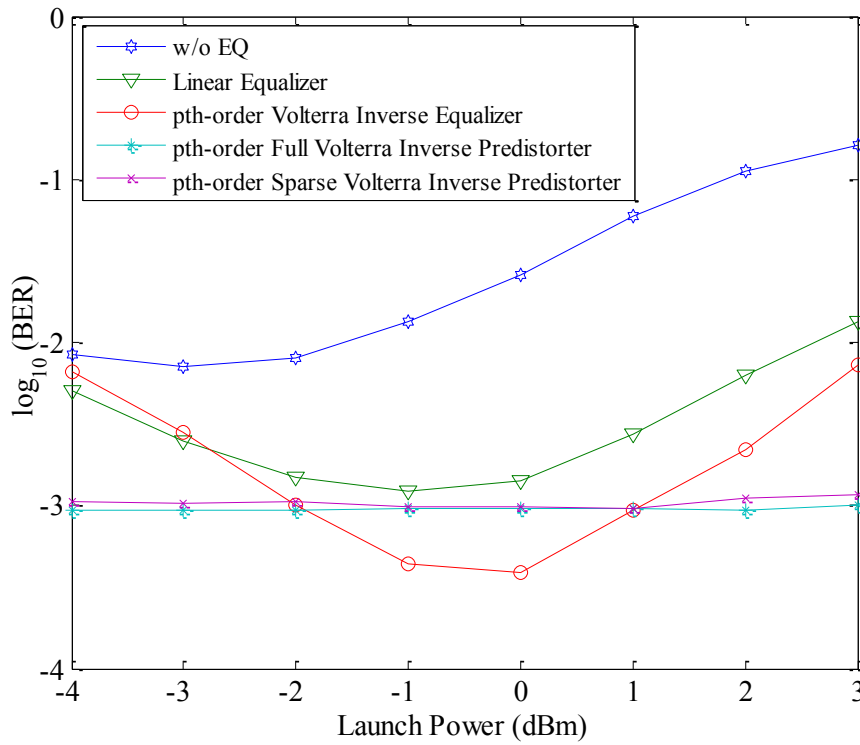


Figure 7.14 BER of the CO-OFDM system without equalization or with linear equalizer, p th-order Volterra inverse equalizer, p th-order full Volterra inverse predistorter, p th-order sparse Volterra inverse predistorter at difference launch power.

The p th-order Volterra inverse equalizer and predistorter effectiveness is evaluated by the Monte Carlo simulations of CO-OFDM system BER performance. The comparison of BER versus launch power of the OFDM system with or without compensation is shown in Fig.7.14. The transmission distance is 800km. For the system with or without compensation, the BER has a parabolic tendency and the BER will reach a minimum point at certain launch power. The

lowest BER occurs at around -3dBm for the system without compensation. With the *pth-order* Volterra inverse equalization, the system's lowest BER occurs around 0dBm. Different from an equalizer, the system with the *pth-order* Volterra inverse predistorter maintains an almost constant BER value at different launch powers. We also design the *pth-order* sparse Volterra inverse predistorter and it has similar performance with the *pth-order* full Volterra inverse predistorter with fewer coefficients.

To explain the reason why the *pth-order* Volterra inverse predistorter can maintain constant BER under different launch power, we compare the actual power of input signal entering the optical fiber under different launch power. For the system with the *pth-order* Volterra inverse equalizer, since the equalization is performed at the receiver, the power of the signals entering the fiber won't be affected by the equalizers and would increase with the increase of launch power. However, the *pth-order* Volterra inverse predistorter changes signals at the transmitter and it acts like a power regulator. As a result, for the system with a *pth-order* inverse predistorter, its input signal power changes slightly, maintaining a nearly constant level when the laser launch power increases from -4dBm to 3dBm.

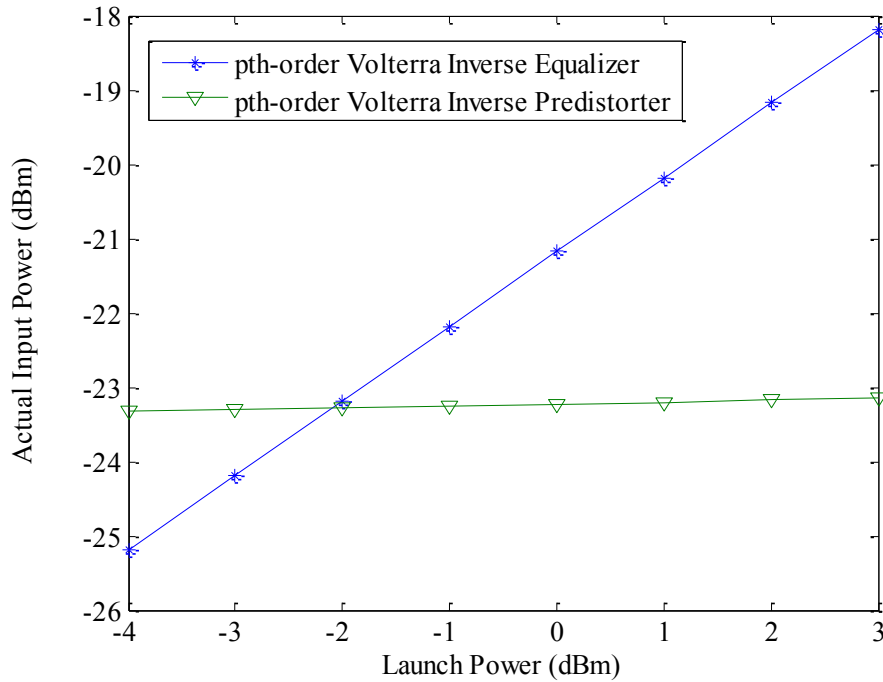


Figure 7.15 Signal power entering the fiber of CO-OFDM systems with *pth-order* Volterra inverse equalizer and with *pth-order* Volterra inverse predistorter under different launch power.

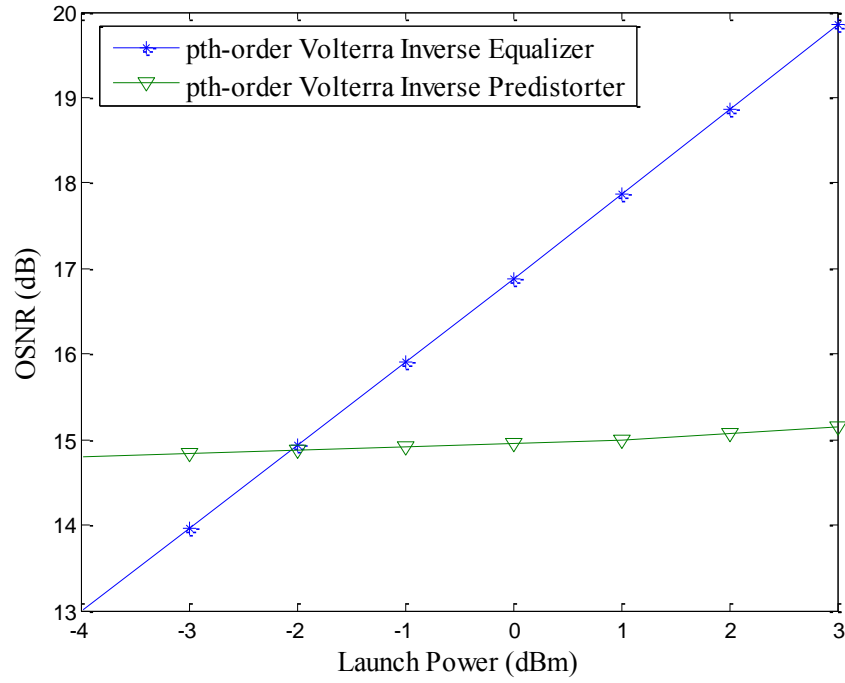


Figure 7.16 OSNR of the CO-OFDM system with *pth-order* Volterra inverse equalizer and with *pth-order* Volterra inverse predistorter under different launch power.

The OSNR at the end of transmission is calculated and shown in Fig. 7.16. The OSNR of the system with the *pth-order* inverse equalizer increases as the launch power increases, while the OSNR of the system with the *pth-order* inverse predistorter increases slightly. This is because the signal power after predistorter will be maintained to around the same level at different launch power, so the OSNR at the end of transmission does not change significantly.

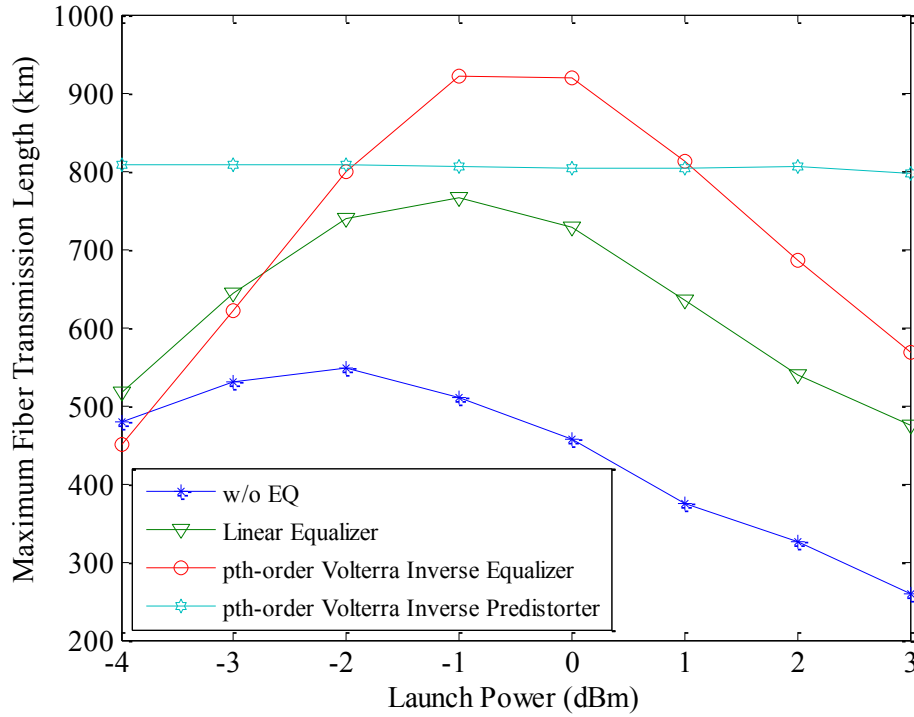


Figure 7.17 Maximum fiber transmission length versus launch power of the OFDM systems without compensation with *pth-order* inverse equalizer and *pth-order* Volterra predistorter.

Fig.7.17 shows the simulation results of CO-OFDM systems' maximum possible transmission length under different launch power to guarantee a 10^{-3} BER. The system without compensation and with *pth-order* Volterra inverse equalizer shows a trend similar to the Fig.7.7. The maximum fiber transmission length of the system with *pth-order* Volterra inverse predistorter remains around 800km under different launch power. This result is in consistent with the results shown in Fig.7.15 and Fig.7.16.

7.4 Equalization by Wiener-Hammerstein Equalizer

Figure.7.18 delineates the received signal constellation of the CO-OFDM system with linear equalizer, Volterra equalizer and Wiener-Hammerstein equalizer and without compensation. The launch power is 0dBm, and the fiber transmission distance is 800km. The number of kernels for linear equalizer, Volterra equalizer and Wiener-Hammerstein equalizer are 30, 30 and 6 (2 for the first FIR filter, 2 for the polynomial filter and 2 for the second FIR filter). Without compensation, the constellation diagram in Fig.7.18 becomes scattered due to SPM, ASE noise and photo-detector noise. The linear equalizer can improve the constellation; however, since the

linear equalizer only consists of linear kernels, its ability in removing amplitude and phase noise is inadequate. Nonlinear Volterra equalizer can improve the constellation quality better than the linear equalizer because its nonlinear kernels can compensate the fiber nonlinear distortions. The difference in the constellation after Volterra equalizer and Wiener-Hammerstein equalizer compensation is marginal; however, the Wiener-Hammerstein equalizer is more preferable since this type of equalizer has fewer numbers of coefficients and is easier to implement.

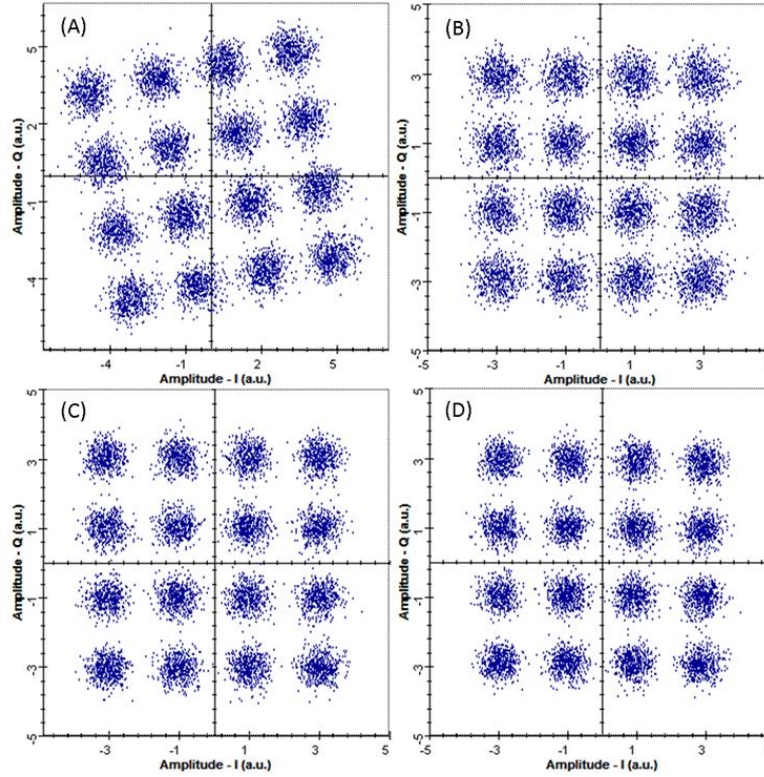


Figure 7.18 Received signal constellation of 16 QAM OFDM system. (A) w/o equalizer, (B) with linear equalizer (C) with Volterra equalizer, (D) with Wiener-Hammerstein equalizer.

The BER results in Fig.7.19 make a comprehensive comparison between the Volterra equalizer and Wiener-Hammerstein equalizer. The number of transmission bit is 2^{17} . In general, with the launch power increasing, the system BER decreases initially until the launch power passes the “optimal” launch power point. Equalization would improve the system BER at different launch powers and the nonlinear equalization outperforms the linear equalization. At lower launch powers (around -4dBm~-2dBm), where the fiber nonlinear effect is weak, the BER difference of system with linear and nonlinear equalization is less significant. At extremely low launch power (-4dBm), the OFDM systems with and without compensation have roughly similar BER values

due to noise enhancement. At high launch powers, the system BER increases due to the increasing in phase and amplitude noise caused by the fiber nonlinearity. The Wiener-Hammerstein equalizer has comparable performance with the Volterra equalizer at lower launch powers. However, the difference between Wiener-Hammerstein equalization and Volterra equalization becomes more noticeable with the increase of launch power, because the Wiener-Hammerstein only has six coefficients and its corresponding Volterra model has fewer terms than the conventional Volterra model in our simulation. An improved Wiener-Hammerstein equalizer should be achieved by increasing the cascading FIR filter memory length or the polynomial nonlinear filter order at the cost of system complexity.

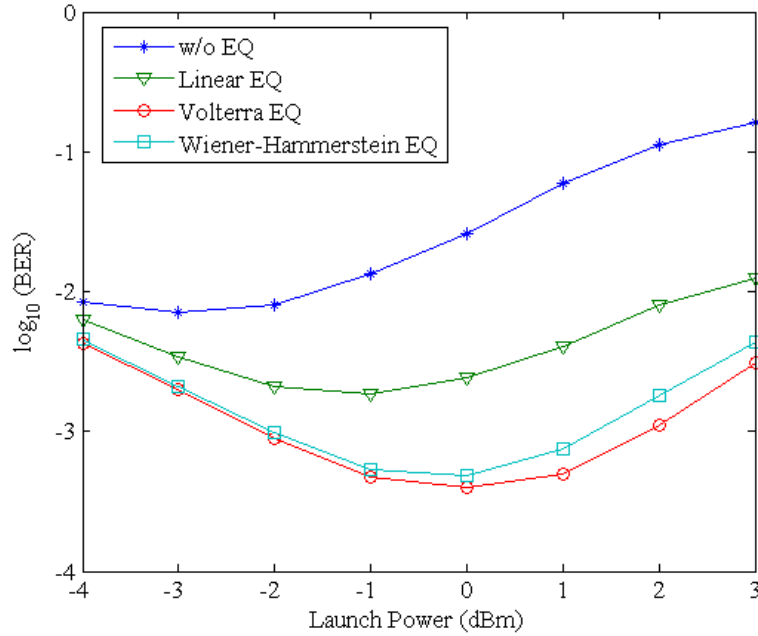


Figure 7.19 BER of 16 QAM OFDM system w/o compensation and with linear/nonlinear compensation as a function of launch power.

The BERs of OFDM system with or without compensation at different OSNR under 0dBm launch power is displayed in Fig.7.20. The OSNR is measured at the end of transmission and the OSNR value is altered by adding ASE noise of various power levels. The increase in ASE noise level would decrease the OSNR, causing the system BER to increase. Nonlinear equalization outperforms the linear equalization and the advantage of using nonlinear equalization becomes more obvious at higher OSNR due to the noise reduction. Wiener-Hammerstein equalizer has comparable performance as the Volterra equalizer at different OSNR with less implementation

complexity.

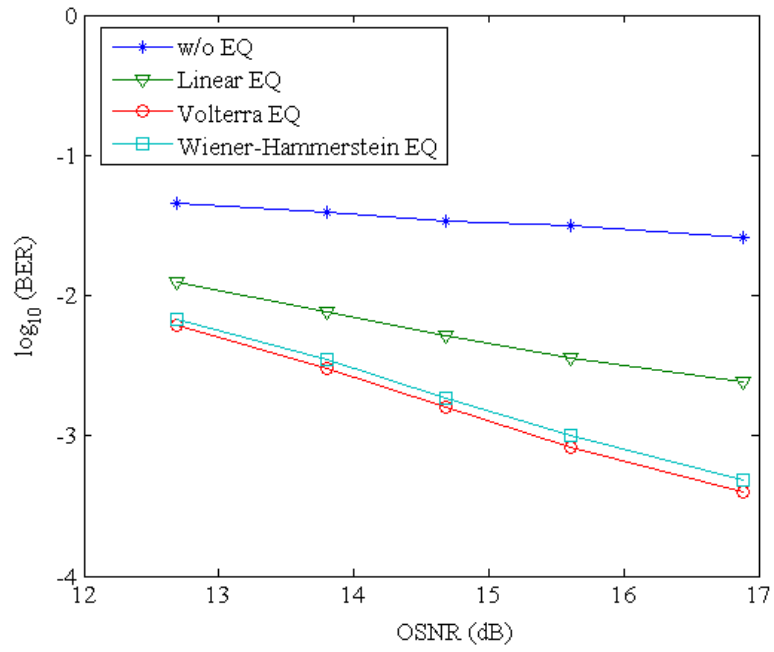


Figure 7.20 BER of 16 QAM OFDM system w/o compensation and with linear/nonlinear compensation as a function of OSNR (fixed launch power).

Chapter 8. Conclusions

Since the OFDM system has a large signal envelope fluctuation, the Optical OFDM system is more vulnerable to the fiber nonlinearity effect than conventional optical communication systems. As a result, nonlinearity compensation is a crucial component of the CO-OFDM system. In this thesis, we concentrated on intra-channel nonlinearity of the CO-OFDM system caused by SPM and FWM among subcarriers. We performed the numerical simulation on the CO-OFDM system and evaluated performances of different nonlinear electrical compensation techniques. The linear compensator was included in our study for comparison purpose.

First, we studied the nonlinear distortions in the single channel and the WDM CO-OFDM systems. We used a third order Volterra model with memory length two to obtain the inverse system model. The tap weights of the Volterra model based equalizer were adaptively updated by the RLS algorithm in the training mode. Then, the nonlinear distortions were compensated by the obtained Volterra model in the data transmission mode. The numerical simulation result shows that the Volterra equalizer has the ability of compensating the intra-channel nonlinear effect and outperforms the linear equalizer. The nonlinear compensator does not perform as well in the WDM system as in the single channel system. The reason is that the WDM system introduces extra nonlinear effect such as XPM, which cannot be easily solved by using a nonlinear electrical domain equalizer only.

Secondly, a conventional Volterra model might have a large number of coefficients, the implementation of such a Volterra system might not be practical and the estimation accuracy deteriorates due to a large number of “unimportant” coefficients. To solve this problem, it is necessary to reduce the number of coefficients of a Volterra system. We used the modified Gram-Schmidt method to identify the most important kernels in the Volterra model and ignored the unimportant ones. The simplified Volterra model is termed as the sparse Volterra model. Our results show that the sparse Volterra system and full Volterra system have comparable performance although the sparse Volterra system usually has fewer coefficients.

Thirdly, we applied the *pth-order* inverse theory to design the equalizer. To design the *pth-order* Volterra inverse equalizer, we need to get the *pth-order* Volterra channel model at first. This can be accomplished by transmitting a known training sequence to derive the channel model. We chose the Volterra channel order to be three, and set its memory length to be two.

The resulting third order Volterra inverse equalizer can compensate nonlinear distortions up to the third order. We also used the *pth-order* sparse Volterra channel model to design the *pth-order* sparse Volterra inverse equalizer. The simulation results show that the equalizer design based on the *pth-order* inverse theory is capable of compensating the intra-channel nonlinear distortion for a signal channel CO-OFDM system. The *pth-order* inverse equalizer based on the sparse Volterra model delivers performance similar to the *pth-order* inverse equalizer based on the Full Volterra model with fewer coefficients.

Fourthly, according to the *pth-order* inverse theory, a *pth-order* pre-inverse is equivalent to the *pth-order* post-inverse. We designed a *pth-order* Volterra inverse predistorter based on the *pth-order* pre-inverse theory. The simulation results show that the predistorter acts as a power regulator and the power level of the signal entering the nonlinear channel is maintained to be the same at different launch powers. The BER and the maximum transmission fiber length of the CO-OFDM system with *pth-order* inverse predistorter remain roughly constant under different launch power.

Finally, we used the Wiener-Hammerstein model to design the equalizer. Compared with the Volterra model, the Wiener-Hammerstein model has simpler structure and fewer coefficients. The coefficient number of the Wiener-Hammerstein model based equalizer used in our simulation is six (2 for the first FIR filter, 2 for the nonlinear filter and 2 for the second FIR filter). The equalizer coefficients were adaptively updated by the joint NLMS algorithm in the training mode. The system BER shows that the equalizer based on the Wiener-Hammerstein model compensate the signal nonlinear distortions and can work better than the linear equalizer. The Wiener-Hammerstein equalizer has comparable performance with the Volterra equalizer, but the Wiener-Hammerstein equalizer has fewer coefficients and is easier to implement.

Reference

- [1] W. Shieh, H. Bao, and Y. Tang, "Coherent Optical OFDM: Theory and Design", *Opt. Express*, Vol.16, pp.842-859, Jan. 2008.
- [2] E. Ip, A. Pak Tao Lau, D. J. F. Barros, and J. M. Kahn, "Coherent Detection in Optical Fiber Systems", *Opt. Express*, Vol.16, pp.753-791, Jan.2008.
- [3] W. Shieh and I. Djordjevic, *OFDM for Optical Communications*, Elsevier, ch.7, 2010.
- [4] I. Kaminow and T. Ye Li, *Optical Fiber Telecommunications IVB*, Academic Press, 2002.
- [5] R. van Nee, "OFDM Codes for Peak-to-Average Power Reduction and Error Correction", *Proc. of IEEE Global Telecomm. Conf.*, pp. 740–744, 1996.
- [6] R. Weidenfeld, M. Nazarathy, R. Noe, and I. Shpantzer, "Volterra Nonlinear Compensation of 112 Gb/s Ultra-long-haul Coherent Optical OFDM Based on Frequency-shaped Decision Feedback", in *Proc. ECOC*, pp. 1- 2, 2009.
- [7] M. Schetzen, *the Volterra and Wiener Theories of Nonlinear Systems*, John Wiley& Sons, Inc., 1980.
- [8] K. V. Peddanarappagari and M. Brandt-Pearce, "Volterra Series Transfer Function of Single-mode Fibers", *J. of Lightwave Technol.*, Vol.15, pp.2232-2241, Dec.1997.
- [9] K. V. Peddanarappagari and M. Brandt-Pearce, "Study of Fiber Nonlinearities in Communication System Using a Volterra Series Transfer Function Approach", in *Proc.31th Annu. Conf. Inform. Sci. Syst.*, pp.752-75, Mar.1997.
- [10] L. Nguyen Binh, "Linear and Nonlinear Transfer Functions of Single Mode Fiber for Optical Transmission Systems", *J. Opt. Soc. Am. A*, Vol. 26, pp.1564-1575, Jul.2009.
- [11] J. D. Reis, L. N. Costa, and A. L. Teixeira, "Nonlinear Effects Prediction in Ultra-Dense WDM Systems Using Volterra Series", in *Optical Fiber Commun. Conf. Collocated National Fiber Optic Engineers Conf.*, pp.1-3, 21-25, Mar.2010.
- [12] B. Xu and M. Brandt-Pearce, "Modified Volterra Series Transfer Function Method", *IEEE Photon. Technol. Letts.*, Vol.14, pp.47-49, Jan. 2002.
- [13] K.V. Peddanarappagari and M. Brandt-Pearce, "Volterra Series Approach for Optimizing Fiber-optic Communications System Designs", *J. of Lightwave Technol.*, Vol.16, pp.2046-2055, Nov.1998.

- [14] Y.Gao, F. Zhang, L. Dou, Zh. Y. Chen, A.Sh. Xu, "Intra-channel Nonlinearities Mitigation in Pseudo-linear Coherent QPSK Transmission System via Nonlinear Electrical Equalizer", *Opt. Commun.*, Vol.282, pp.2421-2425, 2009.
- [15] Ch. M. Xia, and W. Rosenkranz, "Nonlinear Electrical Equalization for Different Modulation Formats With Optical Filtering", *J. of Lightwave Technol.*, Vol.25, pp.996-1001, Apr.2007.
- [16] X. Zhu, S Kumar, S. Raghavan, Y. Mauro, and S. Lobanov, "Nonlinear Electronic Dispersion Compensation Techniques for Fiber-Optic Communication Systems", in *Proc. OFC/NFOEC*, pp.1-3, Feb.2008.
- [17] Ch. H. Tseng and E. J. Powers, "Application of orthogonal-search methods to Volterra modeling of nonlinear systems", *IEEE Int. Conf. on Acoustic. Speech and Signal Process.* , pp.512-515, 1993.
- [18] M. J. Korenberg and L. D. Paarmann, "Orthogonal Approaches to Time series Analysis and System Identification", *IEEE Signal Proc. Mag.*, pp.29-43, Jul.1991.
- [19] L. Yao, W. A. Sethares and Y. H. Hu, "Identification of a Nonlinear System Modeled by Sparse Volterra Series", *IEEE Int. Conf. On Ser. Syst. Eng.*, pp.624-627, 1992.
- [20] F. Y. Wang, J. L. Cheng, Pan J, "MIMO Volterra Filter Equalization Using *Pth-order* Inverse Approach", *IEEE Int. Conf. on Acoustics, Speech, and Signal Process.*, Vol.1, pp.177-180, 2000.
- [21] C. H. Cheng, E. J. Powers, "A Reconsideration of The *Pth-Order* Inverse Predistorter", *1999 IEEE 49th Vehicular Technology Conference*, Vol.2, pp.1501-1504, May, 1999.
- [22] E. Biglieri, S. Barberis, M. Catena, "Analysis and Compensation of Nonlinearities in Digital Transmission Systems", *IEEE J. on Selected Areas in Comm.*, Vol.6, pp.42-51, Jan.1988.
- [23] J. Tsimbinos, K.V.Lever, "Computational Complexity of Volterra Based Nonlinear Compensators", *Electronics Letters*, Vol.32, pp.852-854, Apr. 1996.
- [24] A. Kibangou and G. Favier, "Blind Joint Identification and Equalization of Wiener-Hammerstein Communication Channels Using Paratuck-2 Tensor Decomposition", *15th European Signal Process. Conf. (EUSIPCO 2007)*, Poznan, Poland, pp.1516-1520, Sep.2007.

- [25] M. J. Korenberg, I. W. Hunter, "The Identification of Nonlinear Biological Systems: LNL Cascade Models", *Biol.Cybern.* Vol.55, pp.125-134, 1986.
- [26] P. Cramat, Y. Rolain, "Broad-band Measurement and Identification of a Wiener-Hammerstein Model for an RF Amplifier", *60th ARFTG Conf. Digest*, pp. 49-57, Dec. 2002.
- [27] M. Sano, L. M. Sun, "Identification of Hammerstein-Wiener System With Application to Compensation for Nonlinear Distortion", *Proc. of the 41st SICE Annual Conf.*, vol.3, pp.1521- 1526, Aug. 2002
- [28] M. Zeller and W. Kellermann, "Coefficient Pruning for Higher-order Diagonals of Volterra Filters Representing Wiener-Hammerstein Models", *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, Seattle, USA (WA), Sep. 2008.
- [29] J. P. Costa, A. Lagrange, A. Arliaud, "Acoustic echo cancellation using nonlinear cascade filters", *Proc. 2003 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol.5, pp.389-392, Apr. 2003
- [30] A. S. French, M.J.Korenberg, "Dissection of a Nonlinear Cascade Model for Sensory Encoding", *Ann Biomed Eng.*, Vol.19, pp.473-484, 1991.
- [31] G. P. Agrawal, *Fiber optic Communication Systems*, John Wiley& Sons, Inc., 2002.
- [32] S. V. Kartalopoulos, *Introduction to DWDM Technology*, New York: IEEE press, 2000.
- [33] A. DaSilva, "The Kerr Effect", spring, 2007.
<http://www.phys.psu.edu/~adasilva/miscellaneous/p553kerr.pdf>
- [34] S. Kumar, and D. Yang, "Second-order Theory for Self-phase Modulation and Cross-phase Modulations in Optical Fibers", *J. of Lightwave Technol.*, Vol. 23, pp. 2073-2080, Jun. 2005.
- [35] S. Benedetto, E. Biglieri and V. Castellani, *Digital Transmission Theory*, Englewood cliffs, New Jersey: Prentice-Hall, Inc.1987.
- [36] B. S. In-seung Park, "Design of Equalizers for Nonlinear Digital Communication Systems Using Volterra Filtering Techniques", master thesis, The University of Texas at Austin, 1994.
- [37] A. Gutierrez and W. E. Ryan, "Performance of Adaptive Volterra Equalizers on Nonlinear Satellite Channels", *IEEE Int. Confer. On Communications*, pp.488-492, 1995.
- [38] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd edition, London: The Johns Hopkins Press Ltd. 1989.

- [39] W. Gander, "Algorithms for the QR-Decomposition", Research Report, No.80-82, pp.1-27.,<http://www.inf.ethz.ch/personal/gander/papers/qrneu.pdf>
- [40] M. Schetzen, "Theory of p th-order Inverses of Nonlinear Systems", *IEEE Transact. on Circuits and Syst.*, Vol. CAS-23, pp.285-291, May 1976.
- [41] V. Hegde, C. Radhakrishnan, D. Krusienski, W. K. Jenkins, "Series-Cascade Nonlinear Adaptive Filters", *The 45th Midwest Symposium on Circuits and Syst. 2002 (MWSCAS-2002)*, Vol.3, pp. 219-222, Aug. 2002.
- [42] S. Haykin, *Adaptive Filter Theory*, New Jersey: Prentice-Hall international, INC, 1996.
- [43] D. H. Brandwood, "A Complex Gradient Operator and Its Application in Adaptive Array Theory", *IEEE Proc. Comm., Radar and Signal Processing*, Vol.130, pp.11-16, Feb.1983.
- [44] R. Q. Hui, K. R. Demarest, and C.T. Allen, "Cross-Phase Modulation in Multispan WDM Optical Fiber Systems", *J. of Lightwave Technol.*, Vol. 17, pp.1018-1026, Jun. 1999.

Appendix

Linear equalizer implementation Matlab code (LMS):

```
load -ASCII 'Isd.txt'
load -ASCII 'Irec.txt'
load -ASCII 'Qsd.txt'
load -ASCII 'Qrec.txt'
out=Isd(1:end,2)'+j*Qsd(1:end,2)';
x=Irec(1:end,2)'+j*Qrec(1:end,2)';
order=30;
w=zeros(1,order)';
bufferlen=length(w);
in=[zeros(1,bufferlen-1), x];
beta=0.005;
for k=1:length(in)-bufferlen+1
    buffin=in(k+bufferlen-1:-1:k)';
    filter_out=w'*buffin;
    error(k)=out(k)-filter_out;
    w=w+2*beta*conj(error(k))*(buffin);
end
output = filter(w',1,x);
```

Linear equalizer implementation Matlab code (RLS):

```
function [output,w]=EqualizerLinearFunction(in,desiredout)
%% linear filter
order=30;
w=zeros(order,1);
bufferlen=order;
X=[zeros(1,bufferlen-1), in];
delta=0.0002;
lamda=1;
C=(1/delta)*eye(bufferlen);
for k=1:length(in)-bufferlen+1
    memorybuff=X(k+bufferlen-1:-1:k)';
    error(k)=desiredout(k)-w'*memorybuff;
    p=C*memorybuff/(lamda+memorybuff'*C*memorybuff);
    w=w+p*error(k)';
    C=(1/lamda)*(C-p*memorybuff'*C);
end
output = filter(w',1,in);
```

Full Volterra equalizer implementation Matlab code (RLS):

```
function [output,w,U]=EqualizerFullVolFunction(in,desiredout)
%% Full Volterra
L=length(in);
taps=3;
X=[zeros(1,taps-1) in];
N=taps-1;
```



```

order=3;
coeflen=0;
for l=1:(order+1)/2
    coeflen=coeflen+taps^(2*l-1);
end
w=zeros(coeflen,1);
delta=0.0007;
lamda=1;
C=(1/delta)*eye(coeflen);
U=zeros(coeflen,L);
for i=1:L
    index1=1;
    index3=1;
    index5=1;
    for m1=N:-1:0
        U1(index1)=X(m1+i);
        index1=index1+1;
    end
    for m1=N:-1:0
        for m2=N:-1:0
            for m3=N:-1:0
                U3(index3)=X(m1+i)*X(m2+i)*X(m3+i)';
                index3=index3+1;
            end
        end
    end
    U(:,i)=[U1 U3]';
    error(i)=desiredout(i)-w'*U(:,i);
    k=C*U(:,i)/(lamda+U(:,i)'*C*U(:,i));
    w=w+k*error(i)';
    C=(1/lamda)*(C-k*U(:,i)'*C);
end
output=w'*U;

```

Sparse Volterra equalizer implementation Matlab code part I (RLS):

```

%% Sparse Volterra orthogonal search
u=U.';
coeflen=30;
[v,R]=GramSchmidtReorthogFunction(u);
y=desiredout.';
for i=1:coeflen
    s(i)=v(:,i)'*y;
    delta(i)=s(i)^2/(y'*y);
end
d=y.';
maximum=max(abs(delta));
index=find(abs(delta)==maximum);
S=zeros(size(s));
S(index)=s(index);
ind=1;
NMSE(1)=norm(in.'-y)/norm(y);
recordindex(1)=index;
while ind<4;

```

```

ind=ind+1;
for i=1:coeflen
    if find(recordindex==i)~=0
        delta(i)=0;
    end
end
maximum=max(abs(delta));
index=find(abs(delta)==maximum);
recordindex(ind)=index;
S(index)=s(index);
end
[w]=BackSubstitutionFunction(R,S);
rin=1;
for i=1:coeflen
    if w(i)~=0
        Significantindex(rin)=i;
        rin=rin+1;
    end
end
end

```

Sparse Volterra equalizer implementation Matlab code part II (Gram-Schmidt with reorthogonalization) [39]:

```

function [Q,R] =GramSchmidtReorthogFunction(Q)
%Modified Gram-Schmidt orthogonalization
[m,n] = size(Q);
R = zeros(n);
for j = 1:n,
    t = norm(Q(:,j));
    nach =1;
    while nach,
        for i = 1:j-1,
            x = Q(:,i)'*Q(:,j);
            R(i,j) = R(i,j) + x;
            Q(:,j) = Q(:,j)- x*Q(:,i);
        end
        b = norm(Q(:,j));
        if b>10*eps*t && b<t/10
            nach =1;
            t=b;
        else
            nach = 0;
            if b<10*eps*t
                b=0;
            end
        end
    end
    R(j,j) = b;
    if b*eps~=0
        b =1/b;
    else
        b=0;
    end
end

```

```

    Q(:,j) = Q(:,j)*b;
end

```

Sparse Volterra equalizer implementation Matlab code part III (Back substitution):

```

function [w]=BackSubstitutionFunction(R,s)
% back substitution
n = length( s );
w = zeros( n, 1 );
for i=n:-1:1
    w(i) = s(i);
    for j=(i + 1):n
        w(i) = w(i) - R(i, j)*w(j);
    end
    if R(i,i)~=0
        w(i) = w(i)/R(i, i);
    end
end
end

```

Sparse Volterra equalizer implementation Matlab code part IV(Coefficients update by RLS):

```

function
[output,w]=EqualizerSparseCoefUpdateFunction(in,desiredout,Significantindex)
%% Sparse Volterra Coefficients update
coeflen=30;
L=length(in);
w=zeros(coeflen,1);
taps=3;
X=[zeros(1,taps-1) in];
delta=0.0007;
lamda=1;
C=(1/delta)*eye(coeflen);
U=zeros(coeflen,L);
N=taps-1;
for i=1:L
    index1=1;
    index3=1;
    for m1=N:-1:0
        U1(index1)=X(m1+i);
        index1=index1+1;
    end
    for m1=N:-1:0
        for m2=N:-1:0
            for m3=N:-1:0
                U3(index3)=X(m1+i)*X(m2+i)*X(m3+i)';
                index3=index3+1;
            end
        end
    end
    U(:,i)=[U1 U3].';
    for q=1:coeflen

```

```

        if find( Significantindex==q)~=0;
        else U(q,i)=0;
        end
    end

    error(i)=desiredout(i)-w'*U(:,i);
    k=C*U(:,i)/(lamda+U(:,i)'*C*U(:,i));
    w=w+k*error(i)';
    C=(1/lamda)*(C-k*U(:,i)'*C);
end
output=(w'*U);

```

Linear Inverse Matlab code:

```

function [output]=PthOrderInverseFunctionLinear(w,desiredout)
% Inverse System of the Linear Filter
L=length(desiredout);
filterorder=length(w);
linear(1:filterorder-1)=0;
w=w';
for i=1:L
    sum=0;
    for k=2:filterorder
        sum=sum+w(k)*linear(i+filterorder-k);
    end
    linear(i+filterorder-1)=(desiredout(i)-sum)/w(1);
end
output=linear(filterorder:end);

```

Pth-order Volterra Inverse Matlab code:

```

function [output]=PthOrderInverseFunction(w,desiredout)
% Inverse System of the Nonlinear Filter by pth-Order Inverse
w=w';
linear(1)=0;
linear(2)=0;
L=length(desiredout);
%% Nonlinear Inversion
for i=1:L

    linear(i+2)=(desiredout(i)-w(2)*linear(i+1)-w(3)*linear(i))/w(1);

end
rlinear=linear(3:end);

%% Cubic Inversion
% The first stage of cubic inverse
X=[0 0 rlinear];
U1=zeros(27,L);
% The second stage of cubic inverse
N=2;
for i=1:L

```

```

        index3=1;
        for m1=N:-1:0
            for m2=N:-1:0
                for m3=N:-1:0
                    U3(index3)=X(m1+i)*X(m2+i)*X(m3+i)';
                    index3=index3+1;
                end
            end
        end
        U1(:,i)= U3.';
    end
    out=w(4:end)*U1;
    %The third stage of cubic inverse
    cubicout(1)=0;
    cubicout(2)=0;
    for i=1:L
        cubicout(i+2)=(out(i)-w(2)*cubicout(i+1)-w(3)*cubicout(i))/w(1);
    end
    rcubic=cubicout(3:end);
    %% The total output
    output=rlinear-rcubic;

```

Wiener-Hammerstein model Matlab code:

```

load -ASCII 'Isd.txt'
load -ASCII 'Irec.txt'
load -ASCII 'Qsd.txt'
load -ASCII 'Qrec.txt'
r=Isd(1:4:end,2)'+1i*Qsd(1:4:end,2)';
input=Irec(1:4:end,2)'+1i*Qrec(1:4:end,2)';

M1=2;
u=[
    0.0003
    0.0003
];
in=[zeros(1,M1-1), input];
N=2;
v=[
    0.0003
    0.0003
];
M2=2;
z=zeros(1,M1-1);
w=[
    0.0003
    0.0003];
b=zeros(1,M2-1);
xn=zeros(M1,M2);
yn=zeros(N,M2);
av=0.002;
au=0.002;
aw=0.002;
min=0.4;

```

```

delta=0.0024;
for m=1:14
    for k=1:length(input)
        x=in(k+M1-1:-1:k) .';
        y=u.'*x;
        y1=[y y^2*y' ] .';
        z(k+M2-1)=v.'*y1;
        Z=z(k+M2-1:-1:k) .';
        d=w.'*Z;
        e=r(k)-d;
        a=[1 y'*2*y] .';
        b=v.'*a;
        xn(:,k+1)=b*x;
        X=[ xn(:,k+1) xn(:,k) ];
        p=X*w;
        yn(:,k+1)=y1;
        Y=[ yn(:,k+1) yn(:,k) ];
        q=Y*w;
        u=u+(au/(norm(p)^2+delta))*conj(p)*e;
        v=v+(av/(norm(q)^2+delta))*conj(q)*e;
        w=w+(aw/(norm(z)^2+delta))*conj(Z)*e;
        error(k)=e;
    end
end
for k=1:length(input)
    x=in(k+M1-1:-1:k) .';
    y=u.'*x;
    y1=[y y^2*y' ] .';
    z(k+M2-1)=v.'*y1;
    Z=z(k+M2-1:-1:k) .';
    output(k)=w.'*Z;
end

```