Application of Machine Learning in Optical Communications

Dissertation

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)
der Christian-Albrechts-Universität zu Kiel

vorgelegt von
Rebekka Koch, geb. Weixer
aus
Uelzen

Jahr
2022

1. Gutachter:	Prof. Dr.-Ing. Dipl.-Wirt. Ing. Stephan Pachnicke
2. Gutachter:	Prof. Dr.-Ing. Stephan ten Brink
Tag der mündlichen Prüfung:	13. Januar 2023

Vorwort

Die vorliegende Dissertation ist während meiner Arbeit als wissenschaftliche Mitarbeiterin am Lehrstuhl für Nachrichtenübertragungstechnik an der Technischen Fakultät der Christian-Albrechts-Universität zu Kiel entstanden.

Mein besonderer Dank gilt Herrn Prof. Dr. Stephan Pachnicke, welcher mir die Möglichkeit gegeben hat, mein Promotionsstudium im Anschluss an die Masterarbeit im August 2018 zu absolvieren. Insbesondere möchte ich mich bei ihm für die interessante Aufgabenstellung, seine geduldige Führung, die freundlichen fachlichen und nicht fachlichen Gespräche und die Feuerzangenbowle zur Weihnachtsfeier bedanken.

Ich danke Herrn Prof. Dr. Stephan ten Brink für die Übernahme des Zweitgutachtens und die damit einhergehende investierte Zeit. Außerdem möchte ich mich bei Herrn Prof. Dr. Olaf Landsiedel und Frau Prof. Dr. Martina Gerken für die Bildung einer freundlichen und fairen Prüfungskommission bedanken.

Ein weiterer Dank für die gute Zusammenarbeit gilt Wolfgang Schairer und Bernhard Spinnler von Infinera (Coriant R&D GmbH) für die fruchtbaren Diskussionen und wertvollen Anregungen während meiner Arbeit.

Meinen Kolleg:innen am Lehrstuhl danke ich in besonderem Maße. Danke Petra Usinger, Sandra Robien, Mihail Balanici, Shi Li, Tom Wettlin, Lars Kruse und Olaf Schulz für die gemeinsame Zeit, den stetigen Austausch und nettes Beisammensein. Ein besonderer Dank gilt Sebastian Kühl für die vielen gemeinsamen Programmierstunden am Computer, aber auch Jonas Koch und Simon Ohlendorf für die hervorragende Zusammenarbeit, eure Unterstützung und eure Freundschaft außerhalb der Universität.

Dank gebührt darüber hinaus meiner Familie, meinen Eltern, aber vor allem meiner Tochter Magdalene und meinem Mann. Vielen Dank Jonas, dass du mich immer unterstützt und für mich stark bist. Du sitzt nicht am Rande des Geschehens, sondern direkt an meiner Seite!

Rebekka Koch

Kiel, Januar, 2023

Abstract

Due to increased data traffic, current optical networks are expected to operate at much higher capacity loads than in the past. Advanced transmission techniques, such as coherent transmission or the principle of the elastic optical networks (EON), will be used for this purpose. With the use of coherent transmission, the spectral efficiency can be improved by increasing the modulation depth. However, this requires a higher signal-to-noise ratio. To achieve this, the optical signal power is increased, resulting in more severe degradations due to the highly nonlinear optical channel. The focus of this work is to develop models based on machine learning, that respond or compensate for this nonlinear signal degradation.

In this investigation, the *support vector machine* (SVM) algorithm is implemented and used for the compensation of nonlinear distorted signals as a classifying decision engine of the received symbols. In different transmission systems, the performance of the SVM against nonlinear fiber-optic interference is analyzed in combination with established linear and nonlinear equalizer structures. The results indicate that the flexible decision thresholds of the SVM enable an improved compensation of both, the nonlinear fiber effects as well as distortions of the optical system components, which results in a reduced number of false detections.

The principle of EONs offers a technology to effectively and efficiently use the available spectral resources provided by the optical fiber. A key element of this technology is the bandwidth-variable transponder, which allows the adjustment of the modulation format, the coding scheme, the forward error correction overhead and the symbol rate to the current link conditions. However, this results in a significant increase in the number of configurable parameters in a network, making the choice of parameters for optimal resource utilization a major challenge. In order to reduce the previously increased system cost due to the insufficient utilization of resources, within this work the use of *reinforcement learning* (RL) algorithms for this challenge is investigated. For this purpose, three different problem formulations are tested and compared. The results show that the RL algorithm is able to adapt to unseen link conditions, while comparable heuristic approaches such as the genetic algorithm have to be retrained for each scenario.

Kurzfassung

Aufgrund des zunehmenden Datenverkehrs wird erwartet, dass die optischen Netze zukünftig mit viel höheren Systemkapazitäten betrieben werden als in der Vergangenheit. Zu diesem Zweck werden fortgeschrittene Übertragungstechniken wie die kohärente Übertragung oder das Prinzip der elastischen optischen Netze (EON) eingesetzt. Bei der kohärenten Übertragung kann das Modulationsformat erhöht werden, was die spektrale Effizienz steigert. Dies erfordert jedoch ein größeres Signal-Rausch-Verhältnis. Um dies zu erreichen, wird üblicherweise die optische Signalleistung erhöht, wodurch die Datenübertragung durch die nichtlinearen optischen Kanäle gestört wird. Der Schwerpunkt dieser Arbeit liegt auf der Entwicklung von Modellen auf der Grundlage des maschinellen Lernens, die auf diese nichtlineare Signalverschlechterung reagieren bzw. sie (teilweise) kompensieren.

Im Rahmen dieser Untersuchung wird der Algorithmus einer *Support Vector Machine* (SVM) implementiert und für die Kompensation von nichtlinear verzerrten Signalen als klassifizierende Entscheidungsmaschine der empfangenen Symbole verwendet. In verschiedenen Übertragungssystemen wird die Leistung der SVM gegen nichtlineare faseroptische Störungen in Kombination mit anderen Entzererstrukturen analysiert. Die Ergebnisse zeigen, dass die flexiblen Entscheidungsschwellen der SVM eine verbesserte Kompensation sowohl der nichtlinearen Fasereffekte als auch der Verzerrungen der optischen Systemkomponenten ermöglichen und zu einer geringeren Anzahl von Fehlerkennungen führen.

Das Prinzip von EONs bietet eine Technologie zur effizienten Nutzung der verfügbaren Ressourcen, die von der optischen Faser bereitgestellt werden. Ein Schlüsselelement der Technologie ist der bandbreitenvariable Transponder, der die Anpassung des Modulationsformats, des Codierungsschemas, des Vorwärtsfehlerkorrektur-Overheads und der Symbolrate an die aktuellen Verbindungsbedingungen ermöglicht. Allerdings erhöht sich dadurch auch die Anzahl der konfigurierbaren Parameter in einem Netz deutlich, was die Dimensionierung der Parameter für eine optimale Ressourcenauslastung zu einer großen Herausforderung macht. Um die bisher gestiegenen Systemkosten durch unzureichende Ressourcenauslastung zu reduzieren, wird in dieser Arbeit der Einsatz von Algorithmen des *Reinforcement Learnings* (RL) für diese Problemstellung untersucht. Zu diesem Zweck werden drei verschiedene Problemformulierungen getestet und verglichen. Die Ergebnisse zeigen, dass der RL-Algorithmus in der Lage ist, sich an unbekannte Link-Bedingungen anzupassen, während vergleichbare heuristische Ansätze wie der genetische Algorithmus für jedes Szenario neu trainiert werden müssen.

Table of Contents

Chapter 1

Introduction

The global bandwidth demand is continuously growing exponentially due to the increasing demand for data services (video streaming, Internet-of-Things (IoT), virtual reality, ...). One the significant aspect is that especially video services like Netflix or YouTube account for more than 50% of the total data traffic [1]. Figure 1.1a illustrates that with the introduction of Ultra High Definition (UHD) or 4K video streaming, a growth rate of 47% is still to be expected.

With the use of new technologies, it has been made possible to continuosly serve the growing demand in the recent years. For this purpose, network operators and service providers rely on fiber optic technologies, as these have a large bandwidth and low loss. With 5G being the new mobile communications standard and the steady increase in people and devices accessing the Internet, all indications are that Internet activity can be expected to continue to grow at an annual rate of about 26% (see Fig. 1.1b) [2]. Consequently, the challenges for the optical transmission system that forms

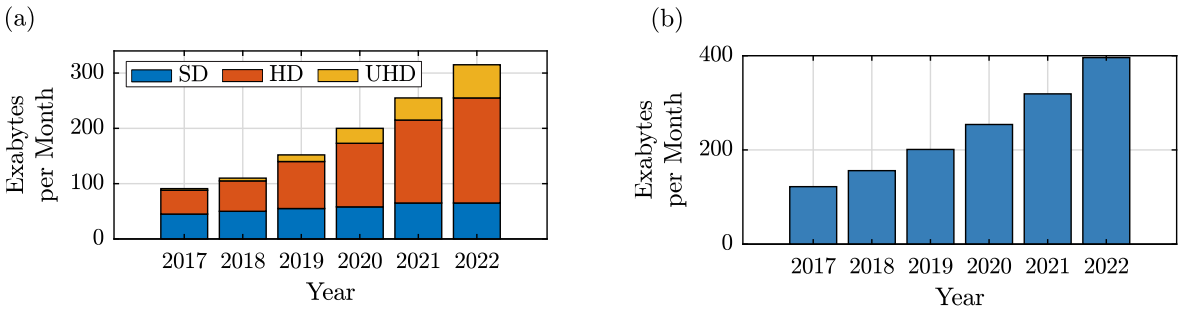


Figure 1.1: Global Internet traffic over the years from 2017 to 2022 [2], showing (a) the global video traffic for standard definition (SD, blue), high definition (HD, orange), and ultra-high definition (UHD, yellow), and (b) the total amount of traffic.

the backbone of today's global communications are increasing as well. An expansion of the infrastructure and further development of the transmission technologies used is therefore inevitable for the future. In this context, the focus lies primarily on increasing the efficiency and scalability of the network.

Wide-area fiber networks connect cities and countries around the world. With the world's first 16-fiber pair repeated submarine cable system, the Hainan-Hong Kong Express, a capacity of 19.2 Tbit/s per fiber pair is achieved a distance of 675 km. This results in a total system capacity of 307.2 Tbit/s [3]. The Dunant submarine cable connects Viginia Beach in the United States with the French Atlantic coast. Over a length of 6,600 km, a total system capacity of 300 Tbit/s is achieved on the 12-fiber pair submarine cable system with a total capacity of 25 Tbit/s per fiber [4]. The networks are based on dense wavelength division multiplexing (DWDM) technology. On shorter distances of 96.5 km, a total capacity of 56.51 Tbit/s can be achieved with this technology, consisting of 34 DWDM channels [5].

While using these high data rates, the quality of transmission is affected by linear effects, as chromatic dispersion, polarization mode dispersion and also by nonlinear effects caused by the optical Kerr-effect. Reliable transmission is therefore always supported by more and more sophisticated signal processing algorithms, such as multilevel modulation and sophisticated equalizers. However, these algorithms come with a high computational complexity. Machine learning (ML) techniques offer promising possibilities for the efficient processing of large amounts of data, which is why they are increasingly becoming the subject of research. For example, the estimation of nonlinear interaction and its equalization forms the basis for a large part of the research in the field of fiber optic communication systems [6]. Furthermore, the increasing complexity of optical networks poses a considerable challenge. The resources of future optical networks should be used optimally by a flexible and dynamic design. Here, the use of ML for modeling and network control is another subject of current research [7]. Moreover, the transmission of information beyond the conventional C-band may further increase the bandwidth requirements. In this context, some additional physical effects have to be particularly considered, such as elastic Rayleigh scattering [8], higher-order fiber dispersion effects, stimulated Raman scattering [9, 10, 11] or stimulated Brillouin scattering [12, 13]. Again, ML approaches offer the possibility to cope with the mentioned impairments, so that in the future the use of broadband systems will be better able to cope with the bandwidth requirements [6].

This thesis investigates the use of two ML techniques in optical transmission systems: the support vector machine (SVM), which is used as a nonlinear decision algorithm to increase system tolerance. And reinforcement learning (RL), which is applied for system optimization with the goal of maximizing the overall throughput.

1.1 Structure

Continuing, the thesis is structured as follows: Chapter 2 presents the architecture of an optical transmission system. First, the components of a single-channel system are explained, providing the communication fundamentals of this thesis. Afterwards, the single-channel system is extended and the principles of EONs are introduced. In contrast to the conventional fixed grid WDM systems, they offer the possibility to adapt to variable transmission conditions by means of a flexible spectrum allocation. Subsequently, the numerical simulation of single-channel and multi-channel transmission is presented.

Chapter 3 introduces the digital signal processing algorithms used in this thesis, and Chapter 4 is devoted to machine learning techniques. It gives an overview of the algorithms and explains how they could be applied in optical transmission technology.

The main part of the work is presented in Chapters 5 and 6, where the algorithms presented in Chapter 4 are implemented, compared and discussed. Chapter 5 focuses on SVM based nonlinear detection and Chapter 6 on parameter optimization for capacity optimization of DWDM systems using RL. After a first analytical evaluation, the respective ML approaches are investigated experimentally.

Finally, a conclusion and outlook are given in Chapter 7.

Chapter 2

The Optical Transmission System

This chapter introduces the fundamental components of an optical transmission system, which are necessary for understanding this thesis. The transmission of an optical signal essentially consists of three components: transmitter (Tx), channel and receiver (Rx). Starting from a digital data signal, the transmitter is responsible for the necessary stages to convert digital (electrical) data to an optical data signal. The most important step there is probably the modulation in which the data signal is modulated on the electrical field of a laser. A booster amplifier, e.g. an Erbium doped fiber amplifier (EDFA), amplifies the signal to the desired power at the beginning of the transmission. Signal propagation takes place in the transmission channel, with the optical fiber as transmission medium and the EDFA as necessary signal inline amplifier. At the output of the fiber, the optical signal is detected and converted back into a digital data signal via the electrical domain. The structure of a multi-span link with N spans, used for long-haul transmission, where each span consists of an optical fiber and an EDFA, is shown in Fig. 2.1.

An increase in channel capacity can be achieved by multiplexing. This technique enables multiple signals to be sent simultaneously as a complex signal through a channel and then recovering the individual signals at the receiver. Common multiplexing techniques are frequency division multiplexing

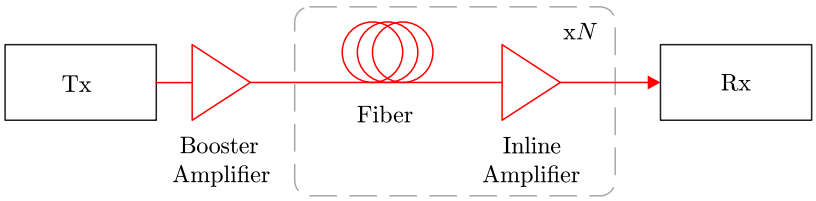


Figure 2.1: Simplified block diagram of a multi-span optical transmission system.

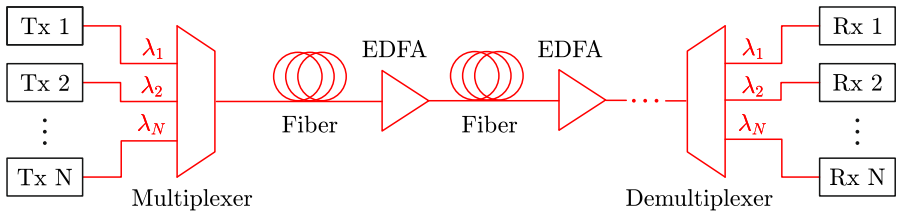


Figure 2.2: Principle of WDM transmission over N channels at the wavelengths $\lambda_1, \lambda_2, ..., \lambda_N$.

(FDM), time division multiplexing (TDM) and wavelength division multiplexing (WDM). In WDM systems, as exemplary shown in Fig. 2.2, different data signals are modulated onto individual lasers (modulated on individual wavelengths). The different carrier wavelengths $\lambda_1, \lambda_2, ...\lambda_N$ are combined by a multiplexer and separated at the receiver by a demultiplexer. Finally, each signal is detected by an individual receiver.

The two main types of WDM used today are coarse wavelength division multiplexing (CWDM) and dense wavelength division multiplexing (DWDM). While traditional WDM systems transmitted only a few widely spaced channels, CWDM systems according to ITU-T recommendations G.694.2 [14] typically use a 20 nm grid within the range 1271 nm to 1611 nm for up to 18 channels. These systems are cost-effective and are used in transport networks in metropolitan areas for a wide range of customers, services and protocols for short distances. In contrast to CWDM, DWDM uses a narrower channel spacing and is therefore able to transport about 100 channels (per band). The requirements for frequency stability are high, so that a control mechanism is used to meet these requirements. Common channel spacings according to ITU-T recommendations G.694.1 [15] are 12.5, 25, 50 or 100 GHz and wider (integer multiples of 100 GHz). Furthermore, it is allowed to choose a flexible grid as long as the respective spacing fulfills the recommendation.

2.1 Components of an Optical Transmission System

After the rough structure of an optical transmission system has been described, the respective components of the transmitter, the optical channel and the receiver are explained in more detail in the following sections. This is done starting with the modulation of the data onto a carrier, followed by the mathematical description of the optical channel with the nonlinear Schrödinger equation (NLSE), and finally the coherent reception of the complex-valued signal is described.

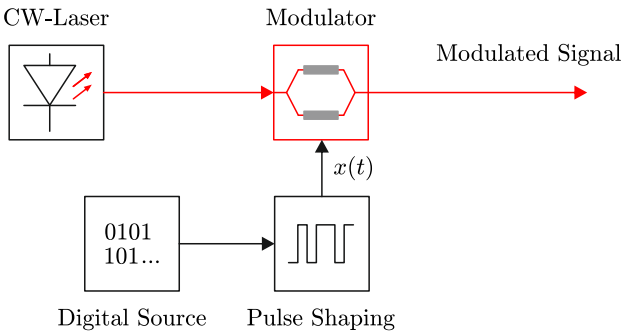


Figure 2.3: Fundamental setup of an optical transmitter with external modulation.

2.1.1 Transmitter

In order to transmit an electrical baseband signal along the optical fiber, it is converted into an optical bandpass signal by the transmitter. For this purpose, the electrical data signal is mapped onto an optical carrier, which is also referred to as modulation. A distinction is made between direct modulation, in which the modulation is performed directly by the laser and external modulation through the use of external modulators such as the Mach-Zehnder modulator (MZM) or the electroabsorption modulator (EAM). Direct modulation is cheaper, but its bandwidth is limited due to chirping and therefore it is generally applied for low data rates. Meanwhile, external modulation is suitable for high data rates and long transmission distances (e.g. 120 GBaud/DP-16QAM over 1000 km [16]). Based on the facts mentioned above, external modulation is used in this work.

Fig. 2.3 illustrates the fundamental setup of an optical transmitter with external modulation. A pulse shaper converts the data signal of the digital source into a continuous time electrical signal $x(t)$ which is described by

$$x(t) = \sum_{k=1}^{N} S_k \cdot (t - kT_{\mathrm{sym}}), \tag{2.1}$$

with the data symbol S_k, the related time duration T_{sym} and the total number of all symbols N to be transmitted. The corresponding symbol rate is specified by the inverse of the symbol duration T_{sym} as

$$R_s = 1/T_{\mathrm{sym}}, \tag{2.2}$$

and measured in Baud.

A continuous wave (CW) laser generates a constant optical carrier signal with a wavelength of λ_c which corresponds to the common carrier frequency i.e. mid of the C-band $f_c \approx 193\,\mathrm{THz}$ ($\lambda_c = 1550\,\mathrm{nm}$). Finally, the continuous time electrical signal is modulated by the external modulator onto this carrier.

Raised-Cosine Pulse Shaping

In transmission technology, it is important to find a suitable choice of pulse shaping. It is necessary to ensure that the detection of a symbol is not affected by neighboring pulses. For illustrative purposes in binary data transmission, the rectangular shaped pulse is often used. But, choosing this pulse shape as a carrier signal is unsuitable for band-limited systems, since in the frequency domain the spectrum approaches an infinitely wide *si*-function, which leads to a degradation of the signal by intersymbol interference (ISI) [17] (if clipping is applied).

A more suitable pulse shape is the raised-cosine (RC) pulse [18]. In the case of RC, the slope in the spectrum is parameterized by the roll-off factor $0 < \alpha_{\mathrm{RRC}} \leq 1$. The factor $\alpha_{\mathrm{RRC}} = 0$ corresponds to an ideal (not realizable) low-pass behavior, the pulses are shaped in the form of an *si*-function and contain relatively high oscillations. The factor $\alpha_{\mathrm{RRC}} = 1$ corresponds to a $\cos^2$-spectrum, similar to a bell shaped pulse, whose bandwidth is doubled compared to $\alpha_{\mathrm{RRC}} = 0$. The normalized transfer function $H(f)$ is given by

$$H(f) = \begin{cases} 1, & |f| \leq \frac{1-\alpha_{\mathrm{RRC}}}{2T_{\mathrm{sym}}} \\ \cos^2\left(\frac{\pi T_{\mathrm{sym}}}{2\alpha_{\mathrm{RRC}}}(|f| - \frac{1-\alpha_{\mathrm{RRC}}}{2T_{\mathrm{sym}}})\right), & \frac{1-\alpha_{\mathrm{RRC}}}{2T_{\mathrm{sym}}} < |f| < \frac{1+\alpha_{\mathrm{RRC}}}{2T_{\mathrm{sym}}} \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

with the corresponding impulse response

$$h(t) = \frac{\sin(\pi t/T_{\mathrm{sym}})}{\pi t/T_{\mathrm{sym}}} \frac{\cos(\alpha_{\mathrm{RRC}}\pi t/T_{\mathrm{sym}})}{1-(\alpha_{\mathrm{RRC}}\pi t/T_{\mathrm{sym}})^2} \tag{2.4}$$

and the symbol duration T_{sym}.

In the literature the RC is also called a *Nyquist pulse* [18], because it fulfills the first and in case of $\alpha_{\mathrm{RRC}} = 1$ the second Nyquist theorem. Fig. 2.4 illustrates the RC for different roll-off factors. If a signal is limited to the symbol duration T_{sym} in the frequency domain ($-1/T_{\mathrm{sym}} \leq f \leq 1/T_{\mathrm{sym}}$), the falling edge occurs point-symmetrically around the Nyquist frequency $f_{\mathrm{Nyq}} = 1/2f_{\mathrm{sym}}$. To optimize the behavior against noise influences, the filtering is implemented „half“ in the transmitter and „half“ in

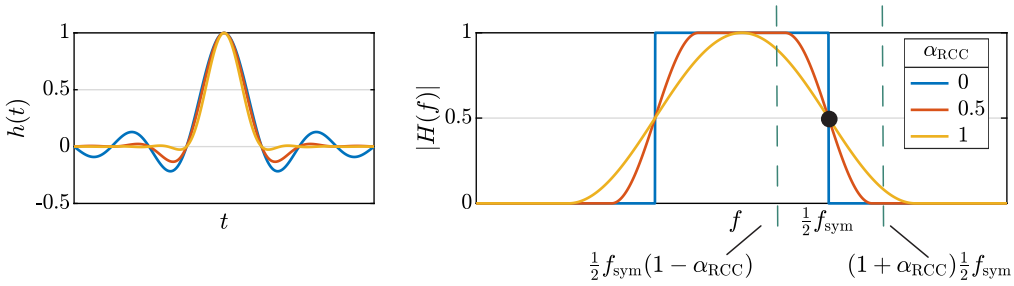


Figure 2.4: RC pulse for the roll-off factors $\alpha_{\mathrm{RRC}} = 0$ (blue), $\alpha_{\mathrm{RRC}} = 0.5$ (red) and $\alpha_{\mathrm{RRC}} = 1$ (yellow), time domain (left) and frequency domain (right).

the receiver, resulting in a root-raised-cosine (RRC) filter and a matched filter (MF) in the transmitter and receiver, respectively.

Modulator

External modulators, including the mentioned MZM, are based on the linear electro-optical effect, the Pockels effect. The Pockels effect produces a change in the refractive index of a waveguide that is implemented on a substrate. Typically, a lithium niobate (LiNbO3) substrate is used because this material exhibits a strong linear electro-optic effect. The change in refractive index alters the propagation constant of the light, causing a relative phase rotation linearly proportional to the applied voltage [19].

The interferometric structure of the MZM is shown in Fig. 2.5a [20]. The incoming optical signal is split by a 3 dB power coupler and fed into two separate waveguides, with a voltage applied to each arm. Subsequently, the branch signals are recombined by a 3 dB combiner and form the output signal E_{out}. Due to the phase shifts in the two waveguides, interference occurs, which modulates the light

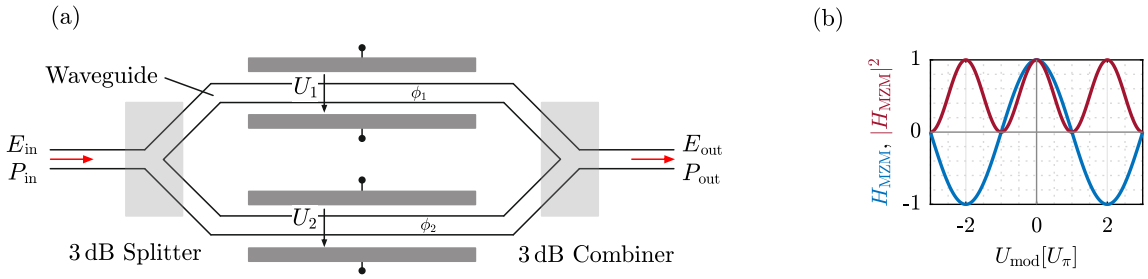


Figure 2.5: Characteristics of a MZM, including (a) the interferometric structure [20] and (b) the transfer curve of the electric field and the corresponding power characteristic.

signal. The relation between the output electric field and the input field is given by

$$E_{\text{out}} = \frac{E_{\text{in}}}{2}\left[\exp(\text{j}\phi_1) + \exp(\text{j}\phi_2)\right] = E_{\text{in}} \cos\left[\frac{\phi_1 - \phi_2}{2}\right] \exp\left(\text{j}\frac{\phi_1 + \phi_2}{2}\right) \tag{2.5}$$

where ϕ_1 and ϕ_2 are the phase shifts of the signals in the two waveguides. Due to a flexible control of the MZM, the modulator can operate in different configurations. At $U_1 = U_2 = 0$ the modulator works as a phase modulator only. In this thesis the modulator operates in a push-pull configuration, which results in an amplitude modulation. For this purpose the following values are applied

$$U_1 = -U_2 = U_{\text{mod}}, \qquad \phi_1 = -\phi_2 = \frac{\pi U_{\text{mod}}(t)}{2U_\pi}, \tag{2.6}$$

where U_π is the voltage that is required to produce a phase shift of 180° or π and $U_{\text{mod}}(t) = u_d(t) + u_{\text{bias}}$ is the applied voltage which consists of the electrical driving signal $u(t)$ and the bias voltage u_{bias} which determines the working point. According to Eq. 2.5, the new transfer function of the MZM in push-pull configuration is obtained as

$$H_{\text{MZM}} = \cos\left(\frac{\pi U_{\text{mod}}(t)}{2U_\pi}\right). \tag{2.7}$$

Fig. 2.5b depicts the transfer characteristics of the electric field used for amplitude modulation and the corresponding power $P_{\text{out}} = |E_{\text{out}}|^2$ used for intensity modulation.

The selection of the optimum working point and its driving characterize the performance of the modulator. The working point voltage u_{bias} determines the direct current (DC) component of the signal. For coherent detection, $u_{\text{bias}} = -U_\pi$ is chosen to suppress the carrier. With a working voltage of $u_{\text{bias}} = -U_\pi/2$, the MZM is driven at the quadrature point, where the working point with highest linearity is located, but containing a high DC component. To reduce the DC component, the working point can be set below the quadrature point ($U_\pi < u_{\text{bias}} \leq -U_\pi/2$). Besides the working point, the driving voltage of the MZM has to be selected optimally as well. Due to the non-linear transfer function of the modulator, nonlinear distortions of the signal occur, if the dynamic range is too high of if a wrong bias point is applied. If, on the other hand, it is too low, the desired signal's amplitude is reduced leading to a low SNR [21].

By combining two MZMs, an I/Q-modulator can be realized, which is able modulate the amplitude and phase of the light. For this purpose, the MZMs are operated in push-pull configuration and one of the two paths is phase-shifted by 90°. The orthogonal signal paths are called inphase (I) and quadrature (Q) and correspond to the real and imaginary parts of the symbols in the constellation diagram (see

Section 2.1.1).

By further combining IQ-MZMs and a polarization rotation, the structure of a polarization multiplexed I/Q-modulator can be obtained. First, the optical signal is divided into x- and y-polarization by a polarization beam splitter. These polarization planes are individually modulated by one I/Q-modulator each. Subsequently, the two modulated polarizations are combined by a polarization beam combiner to form the output signal.

Apart from the nonlinearities of the modulator's transfer characteristic, other effects can arise from the I/Q modulator. These include I/Q-skews or phase and amplitude imbalances due to deviation of the bias voltages [22].

Modulation Techniques

A possibility of enlarging the overall throughput of an optical transmission is the use of higher order modulation techniques, where a certain amount of information carrying bits is assigned to one symbol. The process is called modulation and takes place on the transmitter side. On the receiver side, the information has to be recovered, which is called demodulation. The amount of different symbols $(s_1, s_2, ...s_M)$ is usually a power of two and denoted as M. The higher M, the more information can be carried by any symbol. The symbol duration T_{sym} and its reciprocal quantity is the symbol rate R_S (see Eq. 2.2) [23]. The amount of bits transmitted per symbol is $\log_2(M)$. The bit rate and symbol rate are related by

$$R_B = R_S \log_2(M). \tag{2.8}$$

The mapping of the respective bit sequences to the symbols is determined by the modulation format, whereby a distinction is made between real and complex formats. In the case of real formats, the symbols are located on the real axis, also called the inphase (I-)component. Whereas complex symbols have an imaginary part and consequently contain both an I-component and a quadrature (Q-)component. Each component represents one axis of the complex plane, so that the symbols can be represented in a constellation diagram [24].
In Fig. 2.6 the constellation diagrams of different modulation formats are exemplary shown. The constellation diagram for the pulse amplitude modulation (PAM) of cardinality four (PAM-4) is shown for the bipolar transmission in Fig. 2.6a and for the unipolar transmission in Fig. 2.6b. It can be seen that this modulation format is using only one dimension of the complex plane and therefore belongs to the real modulation formats. This technique has become popular in short-range transmission systems,

such as data centers, because it can be implemented cost-effectively by intensity modulation in the transmitter with direct detection in the receiver (IM/DD) [25].

Fig. 2.6c presents the constellation of a phase modulated system where the symbols are placed at equidistant points from the origin, which is called phase shift keying (PSK). A combination of both approaches is quadrature amplitude modulation (QAM), shown in Fig. 2.6d. This approach is often used for very high data rates and can easily be extended to allow higher modulation levels.

Multidimensional Modulation

Because of the limited flexibility of the presented QAM constellations in terms of spectral efficiency and noise sensitivity [21], it is desirable to realize fractional rational values for the bits per symbol. Multidimensional modulation (MM) provides a way to flexibly map N_b bits to N_{sym} symbols. The N_{sym} symbols in time sequence are combined to one word and characterized by the values of the N_{sym} bits. Consequently, an N_{sym}-dimensional symbol space is spanned from several symbols following each other in time. The time is therefore an additional degree of freedom in the assignment of bits to symbols. This results in fractional rational bits/symbol rates (BpS) for flexible calibration of spectral efficiency or noise sensitivity. The mapping is performed using fixed lookup tables, which must be available in identical form at the receiver for demodulation and the underlying modulation technique is PAM.

Fig. 2.7a exemplary shows the constellation diagrams of an MM with $N_B = \frac{3}{2}$ BpS based on PAM-3 [21]. The axes represent the amplitude values of the two symbols d_1 and d_2. Due to the PAM-3 modulation of the symbols, $M^2 = 3^2$ constellation points are available, but only $2^{N_b} = 2^3$ bit triples have to be assigned to them, therefore one constellation point is not required. The constellation diagram of MM with $N_B = \frac{5}{2}$ based on PAM-6 is shown in Fig. 2.7b, where only $M^2 = 32$ instead of $2^{N_b} = 36$ constellation points are used [22].

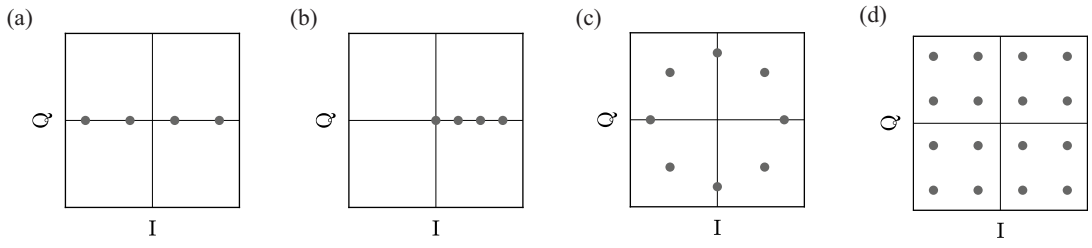


Figure 2.6: Constellation diagrams for different modulation formats. (a) bipolar PAM-4, (b) unipolar PAM-4, (c) 8-PSK, (d) 16-QAM.

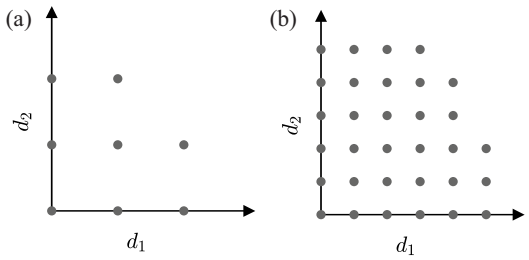


Figure 2.7: Two-dimensional constellations. The x-axis represents the first, the y-axis the second PAM symbol of the word. (a) Unipolar, $N_B = \frac{3}{2}$ BpS, built of two PAM-3 symbols, (b) Unipolar, $N_B = \frac{5}{2}$ BpS, built of two PAM-6 symbols [21].

Table 2.1: Overview of different fractional rational MM-formats.

N_b	N_{sym}	BpS	MM-PAM
3	2	1.5	PAM-3
2	1	2	PAM-4
5	2	2.5	PAM-6
3	1	3	PAM-8

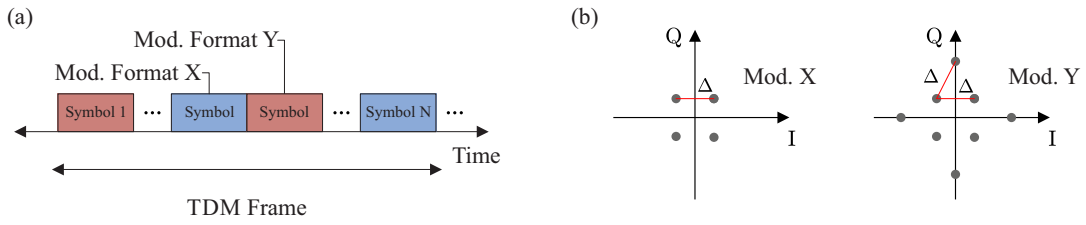


Figure 2.8: Concept of time domain hybrid QAM, (a) in one TDM frame with uniform symbol distribution design and (b) the corresponding constellation with equal Euclidean distance design [26].

Finally, Table 2.1 depicts two possible MM formats and two classic PAM formats with the corresponding BpS values and PAM orders of the individual symbols. Here it quickly becomes apparent that the flexible MM reduces the step sizes between the classic PAM formats.

Time Domain Hybrid QAM

A further possible method to provide flexible mapping of the number of bits to symbols is the use of time domain hybrid QAM (TDHQ). Here, two regular 2^M-ary QAMs are assigned to different time slots within a symbol time (TDM frame). As a result, any fractional rational value can be obtained between two regular integer QAMs. Fig. 2.8 illustrates the concept with the two main design guidelines of TDHQ. The symbols of the two modulation formats X and Y should be evenly distributed in each TDM frame, as indicated in Fig. 2.8a. Furthermore, an identical Euclidean spacing of the two modulation formats (see Fig. 2.8b) helps to increase the noise tolerance of the system. Further possibilities to use several spectral efficiencies are the adjustment of the forward error correction (FEC) rate, probabilistic constellation shaping and coded modulation [26, 27].

2.1.2 Optical Channel

After the signal is modulated onto a carrier, it is transmitted over the optical channel. This contains optical filters, the optical fiber and optical amplifiers, which will be explained in the following in connection with the setup used in this thesis.

Optical Filters

Optical filters can be applied to a wide range of tasks in optical transmission systems. Like optical multiplexers or demultiplexers, they are an essential piece of an optical WDM system. They can also be used for noise suppression, single sideband generation or for routing specific channels in WDM systems. Optical add/drop multiplexers (OADM) allow specific wavelengths to be added to or dropped from the network, reducing the number of unnecessary optoelectronic conversions without affecting

traffic. Today's standard DWDM systems use reconfigurable OADMs (ROADMs). These provide the additional feature of being able to adjust or change the add/drop or pass-through configuration, if the traffic patterns change [28, 29].

Originally, ROADMs used a fixed 50 or 100 GHz grid for channel spacing, which means that each wavelength had to fit into this rigid channel grid in order to pass through the ROADM. Today, more than 90% of ROADM nodes use wavelength selective switch (WSS) technology with a flexible grid. This allows different channel bandwidths and spacing to be used, taking advantage of advances in coherence technology to increase spectral efficiency [30].

Fiber

In optical communications technology, the generally used fiber type are single-mode fibers, made of silica glass according to ITU-T G.652 or G.654. The optical signal propagates there as an electromagnetic wave. Starting from Maxwell's equations of classical electrodynamics, the nonlinear Schrödinger equation (NLSE) serves as a mathematical model for describing the propagation behavior of the optical envelope $A(t, z)$ as a function of the time t and the location z given by

$$\frac{\delta A(z,t)}{\delta t} + \underbrace{\frac{\alpha}{2} A(z,t)}_{\text{attenuation}} - \underbrace{\mathrm{j}\frac{\beta_2}{2}\frac{\delta^2 A(z,t)}{\delta t^2} - \frac{\beta_3}{6}\frac{\delta^3 A(z,t)}{\delta t^3}}_{1^{\text{st}} \text{ and } 2^{\text{nd}} \text{ order dispersion}} = -\mathrm{j}\underbrace{\gamma |A(z,t)|^2 A(z,t)}_{\text{nonlinear Kerr-effect}}. \tag{2.9}$$

The NLSE includes various impairments that limit the practically achievable distances of the transmission. The responsible impairments can be divided into linear (attenuation and dispersion) and nonlinear effects, where nonlinearities arise due to a dependence on the amplitude of the electric field of the light wave (nonlinear Kerr-effect). The following paragraphs outline the linear effects related to the attenuation constant α and the dispersion described by the propagation constant β, and nonlinear effects characterized by the nonlinear coefficient γ of the fiber and their effects on the signal.

Linear Effects

Generally, signal transmission suffers from the attenuation of the signal. The attenuation is mainly dependent on the transmission medium and describes the power losses of the optical signal during transmission. The optical fiber has the advantage that its attenuation is much lower compared to other media like a coaxial cable. The light power P of the optical signal, specified in W or mW, decreases exponentially over the fiber length L and is calculated from the input power P_0 and the attenuation

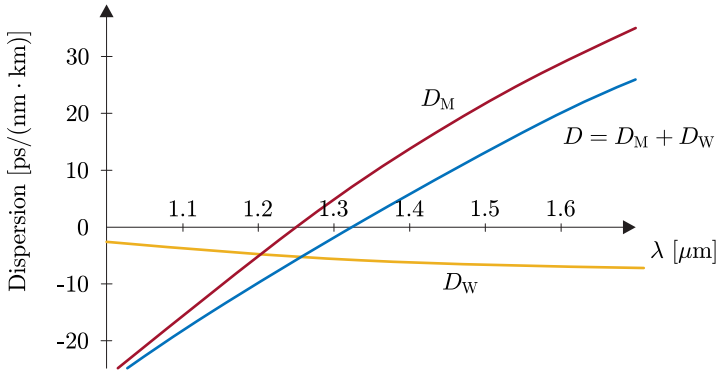


Figure 2.9: Chromatic dispersion D due to material dispersion D_{M} and waveguide dispersion D_{W} for the SSMF [32].

constant α in Neper/m according to

$$P_L = P_0 \exp(-\alpha L). \tag{2.10}$$

The attenuation constant of the fiber is wavelength dependent due to physical causes such as Rayleigh scattering at low wavelengths ($\lambda < 1300\,\mathrm{nm}$), OH and IR absorption [31].
Commonly, the attenuation constant α is expressed in dB/km and has a typical value of $0.2\,\mathrm{dB/km}$ for a standard single-mode fiber at $1550\,\mathrm{nm}$.

If the propagation velocity or phase velocity of an electromagnetic wave depends on the frequency, it is called dispersion. Even in the optical fiber, different spectral signal components propagate at different velocities and thus generate a broadening of the time signal. As a result, the transmitted pulse is broadened, which can lead to ISI resulting in detection errors. In optical communication, a distinction is made between chromatic, multi-mode and polarization mode dispersion (PMD) [31].

The chromatic dispersion dominates among the mentioned dispersion types and is composed of the material dispersion and the waveguide dispersion, as illustrated in Fig. 2.9. The material dispersion D_M is caused by the frequency-dependent refractive index of the materials used in the glass fiber, whereas the waveguide dispersion D_W describes the dependence of the electromagnetic field on the ratio of the core diameter to the wavelength. The propagation velocity is obtained according to

$$v_p(\omega) = \frac{c_0}{n_{\mathrm{eff}(\omega)}} = \frac{\omega}{\beta(\omega)} \tag{2.11}$$

with the speed of light in vacuum c_0 and the effective refractive index $n_{\text{eff}(\omega)}$, which is composed by core and cladding of the fiber, or with the angular frequency ω and the phase constant $\beta(\omega)$. For the definition of a universal dispersion parameter it is helpful to develop the propagation constant $\beta(\omega)$ into a Taylor series around a certain center frequency ω_0 [31]:

$$\beta(\omega) = \beta_0 + \beta_1(\omega - \omega_0) + \frac{1}{2}(\omega - \omega_0)^2 + \frac{1}{6}\beta_3(\omega - \omega_0)^3 + ... \tag{2.12}$$

The coefficient β_0 refers to the absolute phase measure of the carrier frequency ω_0 and β_1 to the inverse of the group velocity. The parameter β_2 describes the change of the group velocity as a function of the distance from the center frequency ω_0, which is also called group velocity dispersion (GVD). β_3 describes the increase of the dispersion. Here, the coefficient β_2, i.e. the GVD, represents the main cause of pulse broadening. With the present dispersion coefficients, the already mentioned general dispersion parameter D can be defined as

$$D = -\beta_2 \frac{2\pi c_0}{\lambda^2} = -\frac{\lambda}{c_0}\frac{d^2 n_{\text{eff}}}{d\lambda^2}, \tag{2.13}$$

and the dispersion slope as

$$S = \frac{\delta D}{\delta t} = \frac{4\pi c_0}{\lambda^3}\beta_2 + \frac{2\pi c_0^2}{\lambda^2}\beta_3. \tag{2.14}$$

For $D > 0$ anomalous dispersion is present, which means that waves with high frequencies propagate faster. For an SSMF, the typical values are $D = 17\,{}^{\text{ps}}/_{\text{nm}\cdot\text{km}}$ and $S = 0.056\,{}^{\text{ps}}/_{\text{nm}^2\cdot\text{km}}$ at $\lambda = 1550\,\text{nm}$.

In addition to chromatic dispersion, PMD occurs due to non-ideal geometrical conditions in the optical fiber, e.g. due to material or production defects also stress or vibrations. Since the refractive index for different polarizations is slightly different the signals propagate at different speeds due to the different propagation time constants $\beta_x \neq \beta_y$. These propagation time distortions are the cause of PMD. In multi-mode fiber, multi-mode dispersion occurs. As soon as different modes have a different propagation constant β, the signal is distorted due to propagation time differences [33].

Nonlinear Effects

Nonlinear effects, which occur during the optical transmission, can be divided into elastic and inelastic scattering processes. An elastic process is the already mentioned Kerr-effect, the intensity-dependent change of the refractive index. Due to the change of the refractive index, the propagation constant

$\beta(\omega)$ and thus the phase of the optical signal changes. This dependency accounts for self-phase modulation (SPM), cross-phase modulation (XPM), and four-wave mixing (FWM) and is defined by the fiber-specific nonlinear coefficient γ given by

$$\gamma = \frac{n_2 \omega_0}{c_0 A_{\text{eff}}}, \tag{2.15}$$

with the nonlinear refractive index n_2 and the effective area A_{eff} of the fiber, where the power is assumed to be uniformly distributed. A typical value of γ for the SSMF is approximately $1.37\,{}^{1}/_{\text{W·km}}$ at 1550 nm. Non-elastic scattering processes include stimulated Brillouin scattering (SBS) and stimulated Raman scattering (SRS).

If only one channel is considered, especially SBS and SPM cause significant effects. SBS describes the interaction of light waves with acoustic waves. In SPM, the intensity of the optical signal modulates its own phase, resulting in a time-varying frequency shift of the carrier wave. This phase modulation is called nonlinear chirp. In fibers with dispersion, this results in amplitude fluctuations. In an SSMF with anomalous dispersion, there is initially a pulse compression and only with further propagation the pulse becomes broader. However, this broadening is smaller than in purely dispersive fibers. If SPM and dispersion cancel each other out in the case of anomalous dispersion, solitons will propagate. These are characterized by the fact that the pulse shape is preserved during the transmission [34].

In multi-channel systems, XPM, FWM and SRS are additional effects that occur. XPM describes the modulation of the signal's phase of one channel, modulated by the power of the other channels. In FWM, a new fourth signal at frequency $\omega_4 = \omega_1 \pm (\omega_2 - \omega_3)$ is generated from three signals at frequencies ω_1, ω_2, ω_3. And SRS results in power transfer between the different WDM channels. The reason for this is the interaction between the photons with vibrations of the silicon molecules, which extract energy from one signal to then amplify another signal. SRS is the basic effect also used in the Raman amplifier, which is an optical amplifier like the EDFA. But the Raman amplifier has less gain, which is why it is only used for special applications [31].

The previously introduced NLSE (see Eq. 2.9) describes the propagation of an optical signal, i.e., the simultaneous interaction of the linear and nonlinear effects. It is a nonlinear partial differential equation having no general analytical solution. For this reason, a numerical solution is preferable. A very common method is the split-step Fourier method (SSFM), which is explained in detail in Section 2.3.1. In this method, the linear and nonlinear effects are considered separately in order to solve in small steps the linear part in the frequency domain and the nonlinear part in the time domain.

Table 2.2: Characteristic values for different fiber types according to [35] in C-band ($f_c \approx 1550\,\text{nm}$).

Type	α [dB/km]	D [ps/nm·km]	S [ps/nm²·km]	γ [1/W·km]
Standard SMF	0.2	17	0.056	1.37
DSF	0.23	0	0.056	1.99
Corning LEAF	0.23	4.2	0.06	1.29
Lycom DK	0.5	-102	-0.23	5.24

Classification of Fiber Types

The previously mentioned SSMF is the most commonly used fiber in optical transmission technology. Its geometrical, mechanical and transmission properties are defined in ITU-T recommendation G.652. Typical values of an SSMF have already been mentioned and are summarized in Table 2.2 for overview. Fig. 2.9 illustrates the dispersion as a function of wavelength for an SSMF, and indicates a dispersion paramter of $D = 0\,\text{ps/nm·km}$ at 1310 nm. By varying the waveguide structure of the fiber, the characteristic wavelength of zero dispersion can be shifted. A fiber which has zero dispersion at 1550 nm is called dispersion shifted fiber (DSF, ITU-T G.653). On the other hand, if the dispersion is very close to zero, nonlinear effects such as FWM will predominate in a WDM transmission. For this reason, there are non-zero dispersion shifted fibers (NZDSF, ITU-T G.655) such as the Corning LEAF in Tab. 2.2, which offer low dispersion. To compensate for the dispersion, there exist dispersion compensating fibers (DCF). These have a negative dispersion coefficient that enables to compensate for accumulated dispersion.

Optical Amplifier and Optical Noise

To compensate for the attenuation of the optical signal, fiber amplifiers are used. These amplify the incoming optical signal instead of converting it into an electrical signal in between. The amplification is based on stimulated emission induced by the signal to be amplified. The amplifiers operate independently of the modulation of the optical signal and have a wide amplifier bandwidth, which is particularly important for WDM techniques. Furthermore, phase, direction and polarization of the input signal are preserved. The invention of the first fiber amplifier, the *Erbium Doped Fiber Amplifier* (EDFA), in 1987 [36] achieved major breakthroughs of optical data transmission over very long distances. Other fiber amplifiers are the *Semiconductor Optical Amplifiers* (SOA) and the *Raman Amplifier*.

The main component of an EDFA is an optical fiber doped with Erbium atoms, in which one or more data signals are guided and amplified by the stimulated emission of the excited Erbium ions. For this purpose, energy is fed to the Erbium atoms via a pump laser, raising them to a higher energy level. When the signal photons pass the stimulated fiber, the electrons fall back to the original energy level, generating photons in the wavelength range of about $\lambda = 1530...1560\,\text{nm}$ that amplify the incoming photons. An optical isolator prevents possible reflections. The gain G of an amplifier, given as the ratio of signal output to signal input power, is

$$G = \frac{P_{\text{out}}}{P_{\text{in}}}. \tag{2.16}$$

The operating wavelength covers 30 to 45 nm in the C- or L-band. In addition to the desired amplification, amplified spontaneous emission (ASE) occurs simultaneously, i.e. photons of arbitrary magnitude and phase in the whole possible wavelength range are generated. Typically, the noise degradation of an amplifier is expressed by the noise figure

$$F_N = \frac{SNR_{\text{in}}}{SNR_{\text{out}}} \approx 2n_{\text{sp}}, \tag{2.17}$$

where n_{sp} is the spontaneous emission factor [31]. Assuming a high gain factor ($G \gg 1$) and complete population inversion ($n_{\text{sp}} = 1$), a minimum noise figure of $F_N = 2\,(3\,\text{dB})$ is achieved. Typically, the noise figure ranges between 4 and 6 dB [36]. The ASE noise is added to the signal and is the main noise source in the optical transmission link. The noise power within the signal bandwidth B_{opt} is

$$\begin{aligned} P_{\text{ASE}} &= 2 \cdot n_{\text{sp}} \cdot h \cdot f_c \cdot (G-1) \cdot B_{\text{opt}} \\ &= 2 \cdot N_{\text{ASE}} \cdot B_{\text{opt}}, \end{aligned} \tag{2.18}$$

with the spectral density N_{ASE} for one polarization, the Planck constant h, and the carrier frequency f_c. Neglecting other noise sources in the system, for example in the transmitter, the performance of the system is expressed by the optical signal-to-noise ratio (OSNR), scaled to $B_{\text{opt}} = 0.1\,\text{nm}$ reference bandwidth

$$\begin{aligned} \text{OSNR} &= \frac{P_{\text{out}}}{P_{\text{ASE}}} \cdot \frac{B_{\text{opt}}}{B_{\text{ref}}} \\ &= \frac{P_{\text{in}}}{F_{\text{N}} \cdot h \cdot f_c} \cdot \frac{1}{B_{\text{ref}}}. \end{aligned} \tag{2.19}$$

When converting the OSNR to dB, Eq. 2.19 can be rewritten to [37]

$$\text{OSNR[dB]} \approx 58\,\text{dBm} + F_N\text{[dB]} + P_{\text{in}}\text{[dBm]}. \tag{2.20}$$

2.1.3 Receiver

In optical communications, a distinction is made between direct detection (DD), which uses a single photodiode for opto-electrical conversion, and the more complex coherent reception, which includes a local oscillator and multiple photodiodes, but preserves the phase information of the signal [38].

Direct Detection

The photodiode converts the instantaneous optical power at a pn-junction into an equivalent electric current $i(t)$. With the optical input power P_{Rx} and R, the wavelength-dependent sensitivity or responsivity of the photodiode ($R \sim 1\,\mathrm{A/W}$ at $\lambda = 1550\,\mathrm{nm}$), the current is defined in terms of

$$i(t) = R \cdot P_{\mathrm{Rx}(t)}. \tag{2.21}$$

If the responsivity is neglected, the detection is modeled by squaring of the complex envelope of the incoming signal $|A(t)|^2$. By detecting the power, the information about the phase is lost, so that only the amplitude information is preserved. If the signal is transmitted via a dispersive fiber, this leads to a broadening of the pulse, i.e. to a change in the amplitude response. In combination with the squared detection of the photodiode this results in nonlinear distortions at the receiver. In addition to this second-order nonlinear impairments, shot noise and thermal noise are added to the signal. However, since the ASE noise of the EDFA is the dominant noise source, these types of noise can be neglected. Usually, DD is applied in conjunction with intensity modulation, so commonly the term intensity modulation with direct detection (IM/DD) is used for this type of systems.

Coherent Detection

To achieve higher spectral efficiency, the more complex coherent detection with polarization multiplexing is used. Here, the amplitude, phase and polarization of the light are obtained so that the information can be encoded in all available degrees of freedom. Coherent detection has its origin in radio communication. Here, a local carrier mixes with the received radio frequency (RF) signal and generates a product term from which the received RF signal can be frequency converted and demodulated. Analogously, in optical transmission technology, the received optical signal is coherently mixed down with a local oscillator (LO) before it is received by the photodetectors. While an DD receiver usually requires more than 1000 photons/bit to operate at a BER of $\leq 10^{-9}$ due to thermal noise, dark current, and many other factors that affect the sensitivity. It can be achieved with coherent receivers, that less than 100 photons/bit are sufficient, because shot noise can be suppressed by increasing the power of the LO. This results in a further advantage of the coherent communication

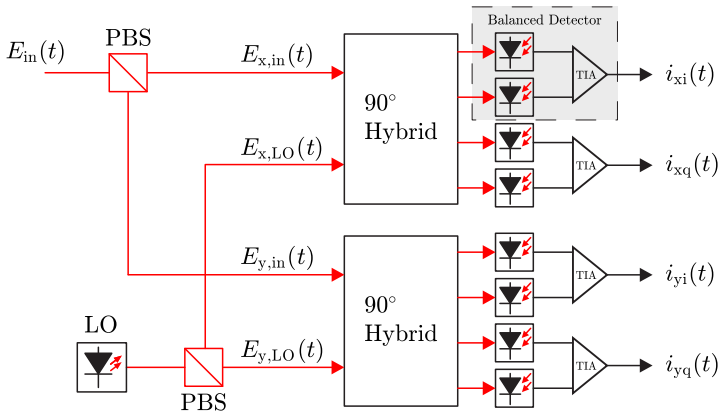


Figure 2.10: Block diagram of an optical coherent PolMux receiver according to [22]. LO: local oscillator; PBS: polarization beam splitter; TIA: transimpedance amplifier.

technique, that the receiver sensitivity is improved compared to IM/DD systems [39].

Fig. 2.10 illustrates the characteristic structure of an optical coherent polarization multiplexed (PolMux) receiver. The input signal $E_{\text{in}}(t)$ is divided into $x-$ and $y-$polarization by a polarization beam splitter (PBS). Another PBS splits the signal of the free-running LO into the respective polarizations. In two 90°- hybrids, one for each polarization, the signals are mixed and split into in-phase (I) and quadrature (Q) components. The in-phase part is obtained by delaying the signal by 0° and 180°. Subsequently, balanced detection is applied by four sets of balanced photodiodes followed by four differential transimpedance amplifiers (TIAs). The objective of balanced detection is to emphasize the difference between similar optical signals by suppressing their common component. In addition, balanced detection usually requires less LO power and significantly improves the dynamic range of the input signal [40, 41]. In unamplified transmissions, the received optical signal is assumed to be free of optical noise, so the noise generated by the coherent receiver defines the final receiver sensitivity. Shot noise and thermal noise are the two basic noise mechanisms responsible for current fluctuations in an optical receiver, even when the incoming optical power is constant. Since in this work each transmission link amplifies the signal with the use of EDFAs, these noise terms are neglected. The resulting signals at the output of the TIAs can be represented as a function of the input polarization for the in-phase and quadrature components [42, 43, 44], according to

$$i_{\text{pol,i}} = R\sqrt{P_{\text{LO}}} \cdot a_{\text{sig,pol}} \cdot \cos(2\pi\Delta f t + \phi_{\text{sig,pol}}(t)), \tag{2.22}$$

$$i_{\text{pol,q}} = R\sqrt{P_{\text{LO}}} \cdot a_{\text{sig,pol}} \cdot \sin(2\pi\Delta f t + \phi_{\text{sig,pol}}(t)), \tag{2.23}$$

where $a_{\text{sig,pol}}$ is the amplitude of the received optical signal of the respective polarization with the corresponding phases $\phi_{\text{sig,pol}}$, and P_{LO} is the power of the LO. In addition, Δf is the difference frequency between the signal carrier frequency and the LO frequency and must be compensated in the DSP by means of e.g. carrier frequency offset (CFO) recovery algorithms.

2.2 Elastic Optical Networks

In order to meet the ever increasing demand for bandwidth, the main goals in the design of modern optical communication systems are higher efficiency, flexibility and scalability. Exactly this potential is provided by so-called elastic optical networks (EONs). In conventional DWDM systems, each optical path is aligned to a fixed grid, independent of path length, bit rate and actual client traffic volume. In EONs, a certain number of transmission parameters, which are fixed in currently deployed networks, such as data rate, modulation format, and inter-channel wavelength spacing, are made tunable. This allows frequency slots to be allocated to the client's data rate requirements [45].

Fig. 2.11 depicts the advantage of spectrally efficient adaptive signal modulation. It can be seen that for shorter optical paths, where a good SNR is to be expected, a higher order modulation format such as 16-QAM can be used. If the entire bandwidth is not utilized, as for example for the second channel, an EON can provide a properly sized intermediate bandwidth. In combination with the flexible channel spacing, a wide range of data rates can be handled in a spectrally efficient manner.
Furthermore, EONs are attractive due to the characteristics of lower signal attenuation, low signal distortion, low power consumption, low material requirements, low space requirements and low costs [45].

One important component of an EON is a bandwidth variable transponder (BVT). BVTs dynamically adjust the bandwidth by changing the bit rate or the modulation format. However, there is always a trade-off between range and capacity. If BVTs have to transmit at low transmission rates, some of the capacity is wasted, which can be avoided by using sliceable BVTs [47, 48, 49]. A comparison between a conventional BVT and an SBVT is shown in Fig. 2.12. An SBVT is able to divide its capacity into one or more optical streams that are transmitted to one or more destinations. Thus, when an SBVT is used to create a low bit rate channel, its unused capacity can be used to transmit other independent streams [45].

A further key task related to EONs is routing, modulation and spectrum allocation (RMSA). The flexible resource allocation mechanisms in EONs make the corresponding service provisioning schemes more complicated, due to the numerous degrees of freedom. To take full advantage of this flexibility, it

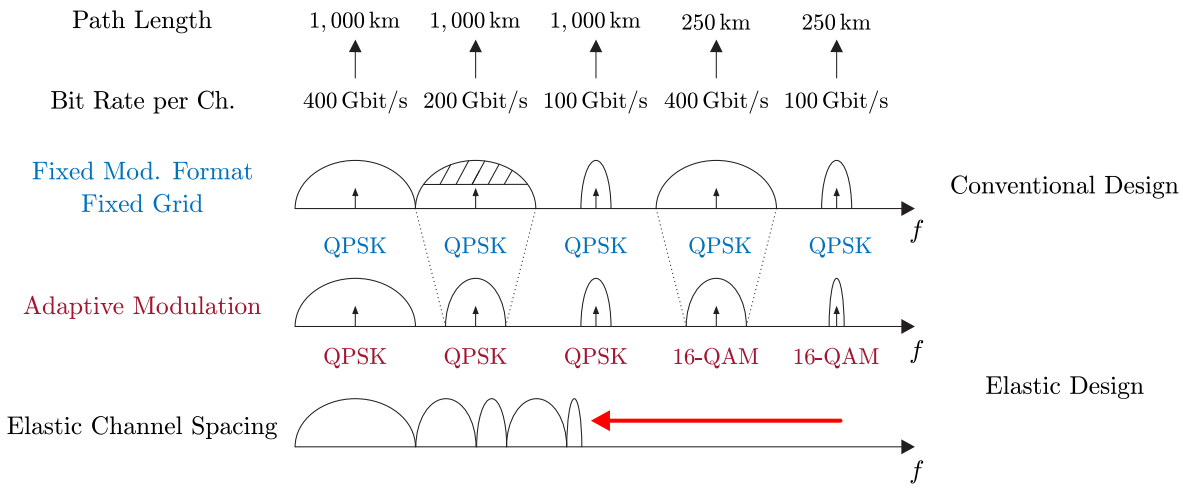


Figure 2.11: Comparison of the spectrum allocation for conventional DWDM system and elastic optical network design [46].

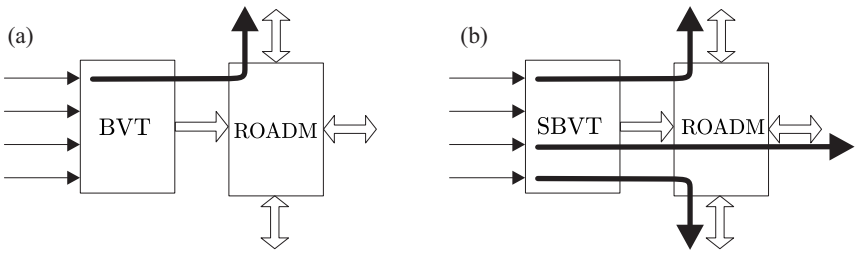


Figure 2.12: Comparison of the BVT and SBVT functionalities according to [45]. (a) depicts the BVT with a single variable bandwidth optical flow and (b) illustrates an SBVT with single or multiple variable bandwidth optical flow.

is necessary to study the RMSA for EONs intensively. Since this is not part of this work, the reader is referred to [50-55].

2.3 Numerical Modeling

This chapter describes the numerical methods, which are used to simulate the transmission system. Commonly, the SSFM is used for single-channel systems, which solves the NLSE (Eq. 2.9) numerically. For multi-channel modeling, the Gaussian noise (GN) model is used, which is based on perturbation theory and provides a sufficiently accurate solution of the NLSE for a system analysis.

2.3.1 Split Step Fourier Method

The NLSE is a nonlinear partial differential equation and can be solved analytically only for a few special cases. Therefore, it is advisable to solve the NLSE numerically. A common method is the SSFM, where nonlinear and linear effects are calculated separately. In small steps the linear part is solved in the frequency domain and the nonlinear part in the time domain. For this purpose Eq. 2.9 is rewritten into

$$\frac{\partial A(z,t)}{\partial z} = (\hat{D} + \hat{N}(z,t))A(z,t) \tag{2.24}$$

with the result that the differential operator

$$\hat{D} = -\frac{\alpha}{2}A - \frac{\mathrm{j}}{2}\beta_2\frac{\partial^2 A}{\partial t^2} + \frac{1}{6}\beta_3\frac{\partial^3 A}{\partial t^3} \tag{2.25}$$

describes the linear and

$$\hat{N} = -\mathrm{j}\gamma|A(z,t)|^2 \tag{2.26}$$

the nonlinear behavior. The propagating signal is jointly affected by dispersion and nonlinearities. The corresponding analytical exact solution of Eq. 2.24 results from the local integration over the interval of length h, according to

$$A(z+h,t) = \left[\exp\left(\int_z^{z+h}[\hat{D} + \hat{N}(z',t)]\,dz'\right)\right]. \tag{2.27}$$

For sufficiently small step sizes, the linear and nonlinear operators can be calculated independently according to

$$A(z+h,t) \approx \exp(h\hat{D})\exp(h\hat{N})A(z,t). \tag{2.28}$$

Through the separate consideration of the differential operators for the linear and nonlinear parts, the calculation of the integral is omitted, which reduces the computational effort. Per step, one Fourier transformation (FT) and one inverse FT (IFT) are needed. If the fast FT (FFT) is further used, the computational effort can be reduced even more [31].

Furthermore, due to the independent consideration of the linear and nonlinear operators, the calculation order can be changed arbitrarily. Modifying this can improve the accuracy of the SSFM. To do this, the differential equation is solved in three steps. The nonlinear part is calculated exactly between two half linear parts. The structure of the Eq. 2.29 leads to the name symmetric SSFM (S-SSFM) [31], defined by

$$A(z+h,t) = \exp\left(\frac{1}{2}h\hat{D}\right)\exp(h\hat{N})\exp\left(\frac{1}{2}h\hat{D}\right)A(z,t). \tag{2.29}$$

The efficiency of the SSFM depends on the respective resolution in time and frequency domains as well as on the selected step size h along the fiber for the numerical approximation. In general, if the step size h is small, the SSFM is sufficiently accurate and the computational effort is correspondingly high. With a suitable choice of the step size, it is possible to adapt the algorithm to the transmission system [37].

2.3.2 Gaussian Noise Model

For modeling the nonlinear propagation of light waves in WDM systems, the Gaussian Noise (GN) model is employed in this work and belongs to the perturbation analysis models. To find an approximate solution for the NLSE, three assumptions are first made here: It is assumed that the nonlinearity is relatively small and therefore only a perturbation compared to the useful signal. A second assumption is that the transmitted signal behaves statistically like stationary Gaussian noise. The last assumption is that the nonlinear interference (NLI), which describes the signal disturbance generated by the nonlinearities, behaves like additive Gaussian noise [56, 57].

The power density spectrum (PSD) of NLI at the end of the link, the $G_{\mathrm{NLI}}(f)$, is calculated using the GN model reference formula (GNRF). This is derived by first inserting a WDM-Tx signal into the Manakov equation, which is modeled as a suitable Gaussian random process and whose spectrum is composed of arbitrarily spectral lines. By using perturbation techniques and ensemble averaging, the GNRF is obtained. The exact derivation steps can be found in [58]. With the assumptions that the compound consists of N_S identical spans and that the loss of each span, including the last one, is

compensated by an EDFA, the definition of the GNRF is as follows

$$G_{\mathrm{NLI}}(f) = \frac{16}{27}\gamma^2 L_{\mathrm{eff}}^2 \cdot \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G_{\mathrm{WDM}}(f_1) G_{\mathrm{WDM}}(f_2) G_{\mathrm{WDM}}(f_1+f_2-f) \cdot \rho(f_1,f_2,f) \cdot \chi(f_1,f_2,f)\; df_2\; df_1 \tag{2.30}$$

with the nonlinear coefficient γ, the effective fiber length L_{eff} and $G_{\mathrm{WDM}}(f)$, the PSD of the transmitted signal. In addition, the factor $\rho(f_1, f_2, f)$ describes the FWM efficiency and takes the closed form in conjunction with an EDFA-like gain, according to

$$\rho(f_1,f_2,f) = \left| \frac{1-\exp(-2\alpha L_s)\exp(\mathrm{j}4\pi^2(f_1-f)(f_2-f)[\beta_2+\pi\beta_3(f_1+f_2)]L_s)}{2\alpha - \mathrm{j}4\pi^2(f_1-f)(f_2-f)[\beta_2+\pi\beta_3(f_1+f_2)]} \right|^2 L_{\mathrm{eff}}^{-2}, \tag{2.31}$$

where α describes the attenuation and β_2 and β_3 GVD and third-order dispersion parameters (see Section 2.1.2). The factor $\chi(f_1, f_2, f)$, often called phased-array factor, takes into account the NLI accumulation in multi-span links and is defined according to

$$\chi(f_1,f_2,f) = \frac{\sin^2(2N_s\pi^2(f_1-f)(f_2-f)[\beta_2+\pi\beta_3(f_1+f_2)]L_s)}{\sin^2(2\pi^2(f_1-f)(f_2-f)[\beta_2+\pi\beta_3(f_1+f_2)]L_s)}. \tag{2.32}$$

In order to use the GN model as a tool for system analysis, the generalized OSNR (GOSNR) is usually taken as a metric, which represents the ratio between the signal power and the power of the combined linear and nonlinear distortions, and is defined as

$$\mathrm{GOSNR} = \frac{P_{\mathrm{ch}}}{P_{\mathrm{ASE}} + P_{\mathrm{NLI}}}, \tag{2.33}$$

with the power of the channel under test (CUT) P_{ch}, the power of the ASE noise P_{ASE} and the power of the nonlinear noise P_{NLI}, the NLI. The associated estimated bit error ratio (BER) is obtained by substituting the conventional OSNR in the usual BER formulas [59]. If $f = 0$ matches the center frequency of the CUT, P_{NLI} results from the PSD of the NLI according to [60]

$$P_{\mathrm{NLI}} = \int_{-R_s/2}^{R_s/2} G_{\mathrm{NLI}}(f)\, df \tag{2.34}$$

where R_s denotes the symbol rate.

The basic model presented so far suffers from the property of overestimating the variance of the nonlinearities. To improve the accuracy, the basic model has been extended. One extension is the extended GN model (EGN), which completely removes the Gaussian approximation from all nonlinear

interference (NLI) components and is thus much more complex than the basic model. Another model, the closed form GN model (CGN), approximates the EGN model starting from the GN model. Using this it is possible to improve the GN model accuracy without increasing the computational complexity [60].

Chapter 3

Digital Signal Processing

Digital signal processing (DSP) is an important part of today's system design. It is commonly divided into a transmitter-side (Tx) DSP and a receiver-side (Rx) DSP. The techniques for higher order modulation and pulse shaping introduced in Section 2.1.1 belong to the Tx DSP. Furthermore, the Tx DSP consists of predistortion, that will be introduced in this chapter. On the receiver side, algorithms for blind compensation of chromatic dispersion [22, 61, 62], time synchronization according to Schmidl & Cox [63, 64, 65], de-rotation of the state of polarization (SOP) by estimating the Jones matrix [17], and carrier frequency offset (CFO) compensation according to [66, 67] are used in this work. Furthermore, a 2x2-MIMO EQ is used to remove the residual dispersion [68]. The main focus of this thesis lies on the equalizer structures of the feed forward equalizer, the nonlinear Volterra equalizer and the 4x4-MIMO equalizer, which is why these will be explained in more detail.

3.1 Transmitter Side

The transmitter of an optical transmission system generally consists of a digital-to-analog converter (DAC), a driver amplifier and a modulator. When using higher baud rates, the system performance is degraded due to bandwidth limitations stemming espacially from DAC and driver, which can be compensated by a precompensation.

3.1.1 Precompensation

The digital precompensation can be used to compensate linear disturbances on the signal already in the transmitter. However, the following conditions must be met: the interference should be time-independent, deterministic and independent of the signal itself. For example, the low-pass characteristic of the transmitter (arbitrary waveform generator (AWG), electrical cables, driver amplifier, MZM),

but also of the photodiode and the analog-to-digital converter (ADC) at the receiver can be taken into account in the equalization. The objective of digital predistortion is to optimize the modulation current or modulation voltage at the transmitter to maximize the signal quality after transmission at the receiver. It is common to measure the frequency response of the system and design a pre-distortion with the use of FIR filters according to this [69, 22].

In the scope of this work, the frequency response is measured with a multi-tone signal in optical back-to-back (B2B) configuration. For the coherent laboratory measurements presented in Section 5.2, the predistortion includes the frequency response and the skew of the AWG only, since no suitable hardware was available to measure the characteristics of the other components in the transmitter setup [22].

3.2 Receiver Side

A modulated signal that is degraded during transmission by the fiber effects presented in Section 2.1.2 requires compensation. Especially linear distortions like ISI contribute to the degradation of the signal. Therefore, this section deals with different structures for the processing of received signals, focusing on signal equalization. First, the linear feed forward equalizer (FFE) is introduced. Then the extension of the FFE, the nonlinear Volterra equalizer (NLVE) is explained and finally the 4x4-MIMO equalizer is presented.

3.2.1 Electronic Dispersion Compensation

In the course of digital signal processing it is common to electronically compensate the chromatic dispersion (EDC), which is done frequently in the frequency domain [68]. The analytical description of the dispersion as an all-pass filter in the frequency domain is given by the transfer function

$$H(L, f_{\mathrm{sym}}) = \exp\left(\mathrm{j} D_{\mathrm{acc}} \frac{\lambda_c^2}{c_0} \pi f_{\mathrm{sym}}^2\right), \tag{3.1}$$

with the length L of the fiber, the symbol frequency f_{sym}, the carrier wavelength λ_c and the dispersion parameter D. By multiplying the signal in the frequency domain with the inverse transfer function $H^{-1}(L, f_{\mathrm{sym}})$ the cumulative dispersion is compensated. This requires the length of the fiber and the dispersion parameter to be known, because $D_{\mathrm{acc}} = D \cdot L$ [70, 71, 72]. If the accumulated dispersion is not known, it can be estimated using the maximum likelihood criterion [73]. In this case, it is called blind dispersion compensation.

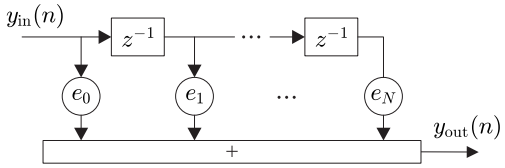


Figure 3.1: Block diagram of a linear FFE.

3.2.2 Feed Forward Equalizer

The probably simplest equalizer is the FFE. This is a linear equalizer constructed as an FIR or delay tap filter. Fig. 3.1 depicts the structure of an FFE of N^{th} order, which consists of N delay taps and $N+1$ coefficients. The discrete (sampled) input signal is passed through the N delay taps, each weighted by a coefficient value. The output signal corresponds to the summed weighted signals according to Eq. 3.3. Depending on the sampling rate of the signal, a distinction is made between a symbol synchronous equalizer with a sampling ratio of one sample per symbol (SpS) and a fractionally spaced equalizer with oversampling (SpS ≥ 2). If no bit synchronous equalizer is used, a sampling reduction to the symbol clock rate should take place at the output [74, 17].

The key for designing an FFE is the channel estimation, i.e. how the coefficients for each delay tap are determined. In contrast to adaptive equalizers, which automatically adapt to time-invariant systems, the FFE is a static equalizer, where the coefficients are calculated in advance and then remain constant. The calculation of the coefficients of the FFE, used in this work, is based on the minimum-mean-square-error (MMSE) algorithm. Its goal is to minimize the mean square error between the received signal and a training sequence at the output of the equalizer. The error is calculated by comparing the output signal of the equalizer $y_{\text{out}}(n)$ with a known training sequence $d(n)$. Due to fiber effects, deviations may occur in this process, increasing the MSE. The MMSE objective function F_{MSE} can be calculated by the MSE according to

$$F_{\text{MSE}} = E\{|y_{\text{out}}(n) - d(n)|^2\} \Rightarrow \min_{\text{error}}. \tag{3.2}$$

The output signal of the equalizer is determined by

$$y_{\text{out}}(n) = \sum_{v=0}^{N-1} e_v \, y_{\text{in}}(n-v) = \boldsymbol{e}^T \boldsymbol{y}_{\text{in}}(n) \tag{3.3}$$

where $\boldsymbol{e} = [e_0, e_1, ..., e_{N-1}]^T$ are the equalizer coefficients and $\boldsymbol{y}_{\text{in}} = [y_{\text{in}}(n), y_{\text{in}}(n-1), ..., y_{\text{in}}(n-N)]$ is the time-dependent input vector. With the autocorrelation matrix $\boldsymbol{R}_{y_{\text{in}}, y_{\text{in}}} = E\{\boldsymbol{y}_{\text{in}} \boldsymbol{y}_{\text{in}}^T\}$ and the

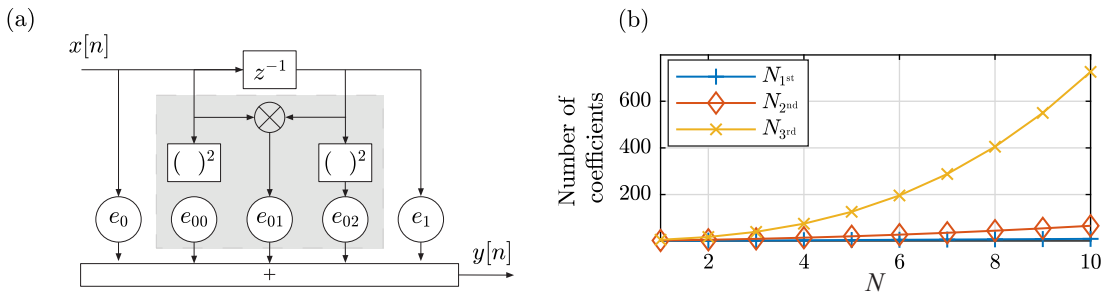


Figure 3.2: (a) Block diagram of a 2nd order NLVE and (b) amount of the total coefficients as a function of the number of delay taps N for the different orders.

cross correlation vector $\boldsymbol{r}_{y_{\text{in}},d} = E\{\boldsymbol{y}_{\text{in}} d(n)\}$, the calculation of the optimal equalizer coefficients can be done according to

$$\boldsymbol{e} = \boldsymbol{R}^{-1}_{y_{\text{in}},y_{\text{in}}} \boldsymbol{r}_{y_{\text{in}},d}. \tag{3.4}$$

For a computationally efficient implementation, filtering is usually done by block processing in the frequency domain using FFT and IFFT. For this purpose, the signals and filter coefficients are specified in vectors of the length of the FFT. The equalized signal is given by

$$\boldsymbol{y}_{\text{out}} = \text{IFFT}\{\text{FFT}\{\boldsymbol{y}_{\text{in}}\} \cdot \text{FFT}\{\boldsymbol{e}\}\}. \tag{3.5}$$

By appending zeros, the operation of a linear convolution is realized from the cyclic convolution. For the processing of the following blocks the *overlapp/save* procedure is used [22, 75, 76].

3.2.3 Nonlinear Volterra Equalizer

An extension of the FFE is the NLVE. Based on the basic structure of the FFE, mixing terms are added in order to compensate for second-order nonlinearities, for example from the modulator or the photodiode. The realization can be either done in the frequency domain or completely in the time domain, as it will be depicted here. Fig. 3.2a illustrates the block diagram of a 2nd order NLVE. By further adding 3rd order mixing terms, where the signal is considered over three delay taps, the 3rd order Volterra is created. The general input-output relationship of a discrete time domain 3rd order NLVE is defined according to

$$y_{\text{out}}(n) = \sum_{v=0}^{N_1-1} e_v \, y_{\text{in}}(n-v) + \sum_{v=0}^{N_2-1} \sum_{l=v}^{N_2-1} e_{vl} \, y_{\text{in}}(n-v) y_{\text{in}}(n-l) + \sum_{v=0}^{N_3-1} \sum_{l=v}^{N_3-1} \sum_{m=0}^{N_3-1} e_{vlm} \, y_{\text{in}}(n-v) y_{\text{in}}(n-l) y_{\text{in}}^*(n-m), \tag{3.6}$$

where y_{in} and y_{out} are the complex-valued in- and outputs of the equalizer at time index n, N_i is the memory length of the i-th order, and e_v, e_{vl}, e_{vlm} are the equalizer coefficients [77].

The 3$^{\text{rd}}$ order Volterra series is strongly related to the NLSE. If the effect of dispersion is neglected, the solution of the NLSE can be expressed as multiplication of two exponential terms [78, 79] (see Section 2.1.2) according to

$$A(z,t) = \exp\left(\frac{-\alpha}{2} z\right) \cdot \underbrace{\exp(-\mathrm{j}\gamma |A(0,t)|^2 L_{\text{eff}} A(0,t))}_{\text{Taylor series expansion}}. \tag{3.7}$$

If the nonlinear part of the NLSE is developed according to the Taylor series, a parallelity of the 3$^{\text{rd}}$ order Volterra series and the NLSE becomes visible. This property allows the NLVE to provide a complete description of the nonlinearity of the channel and is therefore an important tool for the analysis of nonlinear optical transmission systems.

Similarly to the FFE, the coefficients can be estimated using the MMSE criterion. Due to the mixing terms, the number of coefficients and simultaneously the dimension of the model increases rapidly, as as can be seen from the total number of coefficients (as illustrated in in Fig. 3.2b), which is given by

$$N_t = N_1 + N_2(N_2+1)/2 + N_3^2(N_3+1)/2. \tag{3.8}$$

According to Eq. 3.6, the notation NLVE[N1,N2,N3] is used as a complete description of the NLVE. In this case, $N_1 - N_3$ represents the 1$^{\text{st}}$ to 3$^{\text{rd}}$ order memory length of the NLVE. Due to the high computational complexity of Volterra equalizers, attention is generally paid to its complexity reduction e.g. [80, 81].

4x4-MIMO Equalizer

A 4x4-MIMO equalizer is a powerful equalizer that is able to compensate for I/Q imbalances and distortions of the transmitter as well as residual ISI, residual polarization rotations, residual PMD and residual phase noise. It processes the four quadratures of the incoming signal independently, so the

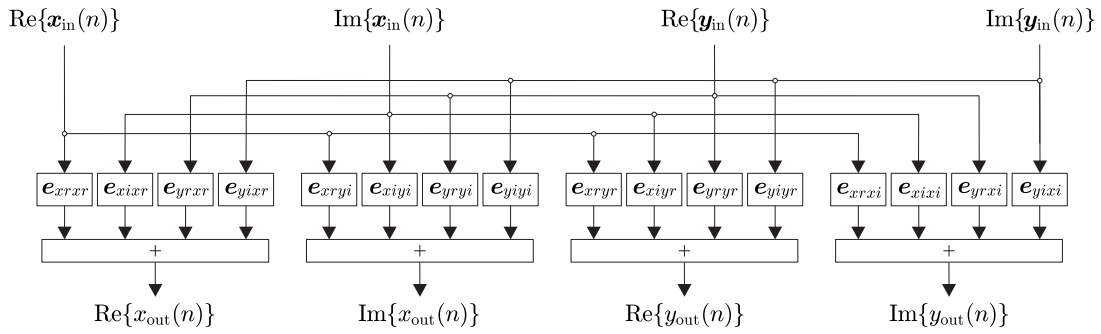


Figure 3.3: Block diagram of a 4x4-MIMO equalizer [22].

imbalances and mixing between the quadrature components can be eliminated [22, 82]. In [22] it is demonstrated that by using the 4x4-MIMO equalizer, the required OSNR remains stable even at high skew values and there is only a small OSNR penalty, if I/Q imbalances are added to the signal.

The structure of the 4x4-MIMO equalizer is depicted in Fig. 3.3. The equalizer consists of 16 real-valued filters, which receive the four quadratures of the received signal in $x-$ and $y-$ polarizations as input signals. The output signals are given by

$$\begin{aligned}
\text{Re}\{x_{\text{out}}\} &= \boldsymbol{e}_{xrxr}^T \,\text{Re}\{\boldsymbol{x}_{\text{in}}\} + \boldsymbol{e}_{xixr}^T \,\text{Im}\{\boldsymbol{x}_{\text{in}}\} \boldsymbol{e}_{yrxr}^T \,\text{Re}\{\boldsymbol{y}_{\text{in}}\} + \boldsymbol{e}_{yixr}^T \,\text{Im}\{\boldsymbol{y}_{\text{in}}\} \\
\text{Im}\{x_{\text{out}}\} &= \boldsymbol{e}_{xrxi}^T \,\text{Re}\{\boldsymbol{x}_{\text{in}}\} + \boldsymbol{e}_{xixi}^T \,\text{Im}\{\boldsymbol{x}_{\text{in}}\} \boldsymbol{e}_{yrxi}^T \,\text{Re}\{\boldsymbol{y}_{\text{in}}\} + \boldsymbol{e}_{yixi}^T \,\text{Im}\{\boldsymbol{y}_{\text{in}}\} \\
\text{Re}\{y_{\text{out}}\} &= \boldsymbol{e}_{xryr}^T \,\text{Re}\{\boldsymbol{x}_{\text{in}}\} + \boldsymbol{e}_{xiyr}^T \,\text{Im}\{\boldsymbol{x}_{\text{in}}\} \boldsymbol{e}_{yryr}^T \,\text{Re}\{\boldsymbol{y}_{\text{in}}\} + \boldsymbol{e}_{yiyr}^T \,\text{Im}\{\boldsymbol{y}_{\text{in}}\} \\
\text{Im}\{y_{\text{out}}\} &= \boldsymbol{e}_{xryi}^T \,\text{Re}\{\boldsymbol{x}_{\text{in}}\} + \boldsymbol{e}_{xiyi}^T \,\text{Im}\{\boldsymbol{x}_{\text{in}}\} \boldsymbol{e}_{yryi}^T \,\text{Re}\{\boldsymbol{y}_{\text{in}}\} + \boldsymbol{e}_{yiyi}^T \,\text{Im}\{\boldsymbol{y}_{\text{in}}\}.
\end{aligned} \tag{3.9}$$

The calculation of the coefficients can be designed statically based on the MMSE criterion, or adaptively using the least mean square (LMS) algorithm [82, 83]. The equalization can be done, analogous to the FFE from Section 3.2.2, with block processing and the overlap-save method. Within this work, the 4x4 MIMO operates as a bit synchronous equalizer, i.e. at 1 SpS and the estimation of the filter coefficients is done using the MMSE criterion.

Chapter 4

Machine Learning

Machine learning (ML) describes a branch of artificial intelligence and deals with the automated development of algorithms based on empirical data or training data. Using the obtained data, patterns are to be recognized in order to find a generalized model that can evaluate unknown input data [84]. The field of machine learning can be divided into three main categories: supervised learning, unsupervised learning and reinforcement learning. In addition, mixed forms such as semi-supervised or a hybrid of supervised and unsupervised learning are possible.

Supervised learning methods require labeled training data, i.e., a data set with the input and output value relations. This allows them to use the existing correlation properties between input and output for the learning process. Supervised learning is mainly used for two types of problems: *classification* of feature groups and *regression* analysis. In classification tasks, the input data is divided into predefined classes. Here, each class is characterized by a certain intensity or expression of the data features. While classification is about dividing data into classes, regression analysis is about fitting a shape that is as close to the data as possible, i.e., looking for an optimal functional relationship between the data characteristics. Therefore, many algorithms set up a hypothesis function $y = f(x)$ to predict the output values that correspond to new input data [7].

In unsupervised learning, no labeled data is provided to the ML algorithm. Instead, the similarities between the different input data are learned to categorize them or to determine a better representation of them. Although unsupervised learning can be used for many tasks, the most common application is clustering. In this process, the entire data set, which was originally interpreted as a homogeneous set, is classified into groups of sub classes (clusters). Frequently used cluster algorithms are k-means [85],

expectation maximization [86, 87].

In reinforcement learning (RL), the algorithm does not have explicit labels, but some form of feedback. The goal of RL is to learn the optimal policy, that is the policy that maximizes the expected cumulative reward. For this, an observer, called agent, is interacting with an unknown environment. Depending on the performed action or the current state, the agent receives either a reward or punishment, in form of a negative reward. Due to the described dynamics, the agent is encouraged to perform a certain action in its current state and develops an optimal strategy of how to act in potentially arising situations in order to maximize the expected cumulative reward [88].

4.1 Artificial Neural Networks

Artificial neural networks (ANNs) are models inspired by the central nervous system of humans or animals [89, 90]. This information processing technique is an ML algorithm, which is based on the model of a human neuron. So it can be used to simulate the behavior of biological systems that consist of neurons. The most popular architecture of an ANN is the feed forward neural network (FNN), whose basic structure corresponds to a directed graph, similar to Fig. 4.1. The FFN predicts an output $\boldsymbol{y}$ for a given input vector $\boldsymbol{x}$ and consists of different layers. The input layer is located at the beginning, where the data $\boldsymbol{x}$ to be processed is fed into the network. The final layer is the output layer. In this layer, the output of the network, an estimate of the output data, is generated. Between the input layer and the output layer is the hidden layer. While the input and output layers consists only ones, the hidden layer can be made up of several layers. If this is the case, it is commonly called a deep neural network (DNN). It should be noted, that as the number of hidden layers increases, the complexity of the neural network also increases [7].

Each layer consists of at least one neuron or node. Neurons serve to transmit the received information in a modified form. Each neuron is provided by a bias parameter, with the function of raising the energy level at the neuron and ensuring the output of the neuron. Within each neuron, an activation function $f(z)$ is applied to control its output or behavior. These are differentiable functions that can be used to map to ranges of values or specific values that allow to learn nonlinear problems [91, 92].

Each connection between neurons is weighted by a set of parameters $\boldsymbol{w}$ and some bias values b. The higher the value of the weight, the greater the influence of one neuron on the other. The knowledge of

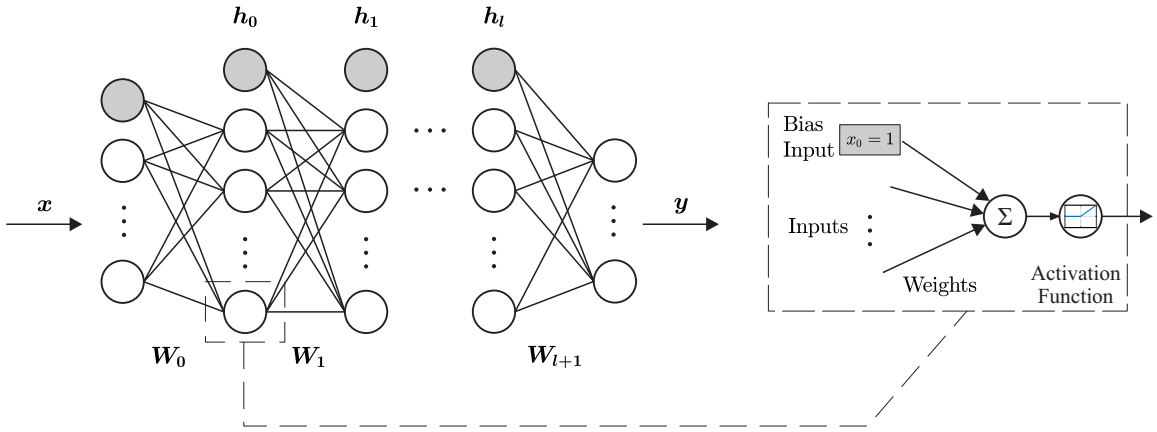


Figure 4.1: Visualization of the different activation functions of the neurons of the NN.

the neural network is stored in these weights. The output $\boldsymbol{y}$ of an FNN is obtained according to

$$\begin{aligned} \boldsymbol{h_0} &= f(\boldsymbol{x}\boldsymbol{W}_0 + b_0) \\ \boldsymbol{h_1} &= f(\boldsymbol{h_0}\boldsymbol{W}_1 + b_1) \\ &... \\ \boldsymbol{h_l} &= f(\boldsymbol{h_{l-1}}\boldsymbol{W}_l + b_l) \\ \boldsymbol{y} &= f(\boldsymbol{h_1}\boldsymbol{W}_{l+1} + b_{l+1}) \end{aligned} \tag{4.1}$$

where the dot product between the input vector and the weight matrix is calculated and then the nonlinear activation function is applied.

Before the correct output is calculated, the FNN needs to be trained. During the training or learning process a change of the weights takes place. For this purpose, the weights of the FNN are initialized randomly at the beginning. After that, training is performed with a large number of known in- and output pairs. In this iterative training process, an optimization problem is solved when it minimizes the difference between the actual output vector $\boldsymbol{y}$ of the network and the desired output vector $\hat{\boldsymbol{y}}$, measured in a loss function

$$\mathcal{L}(\theta) = \frac{1}{2}(\boldsymbol{y}(\theta) - \hat{\boldsymbol{y}})^2, \tag{4.2}$$

including the weights

$$\theta = (\boldsymbol{W}_0, \boldsymbol{W}_1, ...\boldsymbol{W}_{l+1}, b_0, ..., b_{l+1}). \tag{4.3}$$

Since the prediction vector $\boldsymbol{y}(\theta)$ is a function of the neural network weights (abbreviated as θ), the loss is a function of the weights, as well. Exactly this minimization of the loss function leads to the fact that an FNN makes better predictions independent of the characteristics of the respective task, therefore it is important to choose an appropriate loss function depending on the problem. For example, for regression models, the mean square error (MSE) is usually used, while for classification, the cross entropy error function is applied.
The lower the value of a loss function, the closer the weights approximate the real values, which is why the goal is to minimize the loss. The best known general algorithm for minimizing the loss function is *stochastic gradient descent* (SGD) or simply called gradient descent, where the model weights are updated at each iteration using the backpropagation algorithm. In gradient descent the gradient of the loss function is calculated with respect to the weight parameters θ, according to Eq. 4.4. Afterwards, the weights of the network are adjusted in the direction of the negative gradient [93].

$$\theta_j \leftarrow \theta_j - \epsilon \Delta_{\theta_j} \mathcal{L}(\theta) \tag{4.4}$$

Via alternating forward and backward passes, the optimal weights that minimize the loss function are achieved. The learning rate ϵ, is an important and sensitive hyper parameter that has to be set manually. It determines the strength of the gradient decay per step: a too small value of ϵ leads to a slow convergence while a large value can overshoot correct paths. The training takes place over at least one epoch. This describes one iteration over the entire training data set and is commonly divided into batches again. Training in batches offers the advantage that by subdividing the entire training data set and updating the weights frequently, local minimums can be detected and escaped more effectively.

In order to achieve better performance the FNN can be adapted via so-called hyper parameters [94]. For instance, the number and size of the layers but also parameters such as the learning rate, the batch size or the optimizer can be adjusted .

ANNs can assume a variety of different structures. So far, the structure of a feed forward network (FNN) has been discussed. As the name indicates, the flow of information is exclusively forward. Another widely used structure is the recurrent neural network (RNN), which is characterized by the fact that the neurons of a layer have a feedback to other neurons of the same or a previous layer. This allows temporally encoded information to be detected in the data. An RNN is called memory-bound because it gives less weight to data fed in first in continuous time steps than to data fed in later. Therefore, the order in which the data is fed in can be quite relevant, for example in text recognition.

This is the only way to obtain the context of a sentence. A very well-known modification of the RNN is the so-called long-short-term-memory network (LSTM) [95], which is able to remember long-term dependencies and previous experiences. LSTM can be found in numerous applications. In particular, in speech recognition of smartphones, e.g. Amazon Alexa assistant or the intelligent translation program Google Translate [96].

4.1.1 Activation Function

The choice of a suitable activation function depends on the optimization problem [97]. The most common activation functions are the linear or rectified linear unit (ReLU), the step and the tanh-function, which are illustrated in Fig. 4.2 and will be described in the following.

The linear activation function describes a linear dependence between the activity level z of the neuron and its activation, as illustrated in Fig. 4.2a . The one of the most popular linear activation functions is the rectified linear unit (ReLU) and is defined by [97]

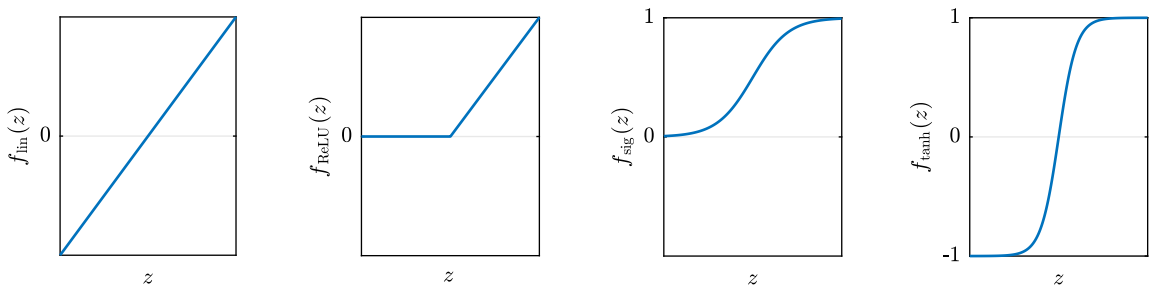
$$f(z) = \begin{cases} 0, & \text{if } z < 0 \\ z, & \text{if } z \geq 0. \end{cases} \tag{4.5}$$

The ReLU function is a linear activation function with a threshold, see Fig. 4.2b. This means that a fixed threshold must first be exceeded before a linear dependence occurs, in this case the threshold is at $z = 0$. If the activation of the neuron is negative, the neuron is deactivated $f(z) = 0$. The network considers the value of the neuron as unusable. If $z > 0$, the activity level of the neuron corresponds to the activation level. This activation function can be useful, if a low input signal, such as noise, should not be forwarded.

The step function has only two states of the activity level: 0 (or -1) and 1 and is also known as binary activation function. The step function is defined according to

$$f(z) = \begin{cases} 0, & \text{if } z \leq 0 \\ 1, & \text{if } z > 0. \end{cases} \tag{4.6}$$

This function is quite easy to evaluate and implement but contains a discontinuity at $z = 0$.



(a) Linear (b) ReLU (c) Sigmoid (d) Tanh

Figure 4.2: Visualization of the different activation functions of the neurons of the NN.

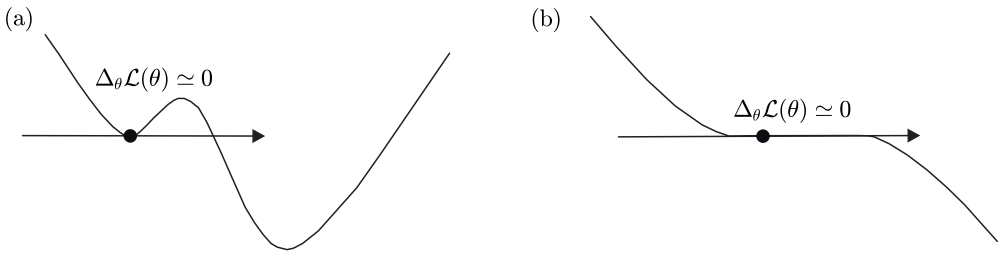


Figure 4.3: Visualization of a saddle point and a local minimum [99].

The sigmoid activation function is a continuously differentiable variant of the step function and is defined by

$$f(z) = \frac{1}{1 + e^{-z}}. \tag{4.7}$$

This type of activation functions is often used. It represents the range of values from 0 to 1 and is therefore used in models where a probability has to be predicted. The tanh-function also belongs to the sigmoid functions. It can be used to map the range of values to the interval from -1 to 1.

Other intervals and limits can be made feasible by shifting them for each of the activation functions mentioned. For more abstract problems that cannot be described visually, the *trial-and-error* principle, i.e. trying different functions, can help in the selection process [98].

4.1.2 Optimizer

If saddle points or local minima occur during the training, as shown in Fig. 4.3, the gradient decreases to zero although the minimum is not reached. As a result, the training slows down. To avoid this

problem, it is common to extend the gradient descent procedure with so-called optimizer algorithms [99, 100].

Stochastic Gradient Descent with Momentum

The stochastic gradient descent (SGD) with momentum differs from the SGD presented in Eq. 4.4 in the fact that an additional term is added, with which the gradient builds up a velocity v during the training, which is controlled by the parameter ρ [92, 99].

$$\begin{aligned} v_{t+1} &\leftarrow \rho v_t + \nabla_\theta \mathcal{L}(\theta) \\ \theta_j &\leftarrow \theta_j - \epsilon v_{t+1} \end{aligned} \tag{4.8}$$

The introduction of the momentum term causes that the step size in direction of the global minimum no longer depends on the slope of the loss function at the current point, but also on the velocity that has built up over time. Therefore, saddle points and local minima are less dangerous for the gradient.

Adaptive Gradient Algorithm

While the SGD with momentum uses a constant learning rate, the AdaGrad optimization strategy sets the learning rate adaptively. For this purpose, the quadratic gradient is summed up during the optimization by whose root the current gradient is divided [92, 99]. Mathematically it is expressed according to

$$\begin{aligned} g_0 &= 0 \\ g_{t+1} &\leftarrow g_t + \nabla_\theta \mathcal{L}(\theta)^2 \\ \theta_j &\leftarrow \theta_j - \frac{\nabla_\theta \mathcal{L}(\theta)}{\sqrt{g_{t+1}} + 10^{-5}} \end{aligned} \tag{4.9}$$

where g_{t+1} represents the sum of squared gradients up to the time $(t+1)$ and the term 10^{-5} is used to avoid dividing by zero.

For example, if only low gradients are calculated, the learning rate increases because it is divided by a lower value. If, conversely, only higher gradients are calculated, they are divided by a high value, which reduces the learning rate. Consequently, the updates of the weights according to AdaGrad are made in equal proportions.

However, if the training takes too long, the sum of squared gradients becomes larger and larger, resulting in a small learning rate, which becomes even smaller the longer the training continues. As a result, training may not stop because the algorithm is stuck [92].

Root Mean Squared Propagation

To solve the problem that occurs when using AdaGrad algorithm, there is a modification that can be found in the Root Mean Squared Propagation (RMSProp) algorithm. Instead of letting the sum increase continuously over the training period, it becomes smaller. For this purpose, the sum of the squared gradients is weighted with α_{RMSProp} and the current gradient is weighted with the term $(1 - \alpha_{\text{RMSProp}})$, so that the algorithm is described with [92, 99]

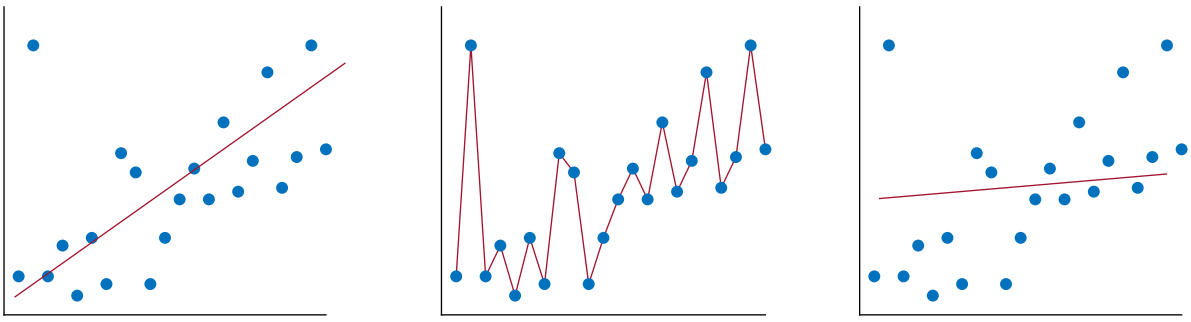
$$\begin{aligned} g_0 &= 0,\ \alpha_{\text{RMSProp}} \simeq 0.9 \\ g_{t+1} &\leftarrow \alpha_{\text{RMSProp}} g_t + (1 - \alpha_{\text{RMSProp}}) \nabla_\theta \mathcal{L}(\theta)^2 \\ \theta_j &\leftarrow \theta_j - \frac{\nabla_\theta \mathcal{L}(\theta)}{\sqrt{g_{t+1}} + 10^{-5}} \end{aligned} \tag{4.10}$$

Adaptive Moment Estimation

The Adaptive Moment Estimation (ADAM) algorithm combines the idea of RMSProp and the SGD with momentum and is characterized by [92, 99]

$$\begin{aligned} m_0 &= 0,\ v_0 = 0 \\ m_{t+1} &\leftarrow \nu_1 m_t + (1 - \nu_1) \nabla_\theta \mathcal{L}(\theta) \\ v_{t+1} &\leftarrow \nu_2 v_t + (1 - \nu_2) \nabla_\theta \mathcal{L}(\theta)^2 \\ \Theta_j &\leftarrow \theta_j - \frac{\nabla_\theta \mathcal{L}(\theta)}{\sqrt{g_{t+1}} + 10^{-5}} m_{t+1}, \end{aligned} \tag{4.11}$$

where the first term describes the first impulse with the hyper parameter ν_1. Unlike the SGD with impulse, here the current gradient is multiplied by the factor $(1 - \nu_1)$. The second term contains the running sum of squared gradients and the hyper parameter ν_2. And the third term updates the weights and represents a combination of RMSProp and SGD with momentum.
At the beginning of the training, the algorithm presented so far would make a large update step due to the initialization of m_0 and v_0. For this reason, a bias correction is inserted with which the ADAM optimizer is fully described [101, 92].



(a) Accurate (b) Overfitting (c) Underfitting

Figure 4.4: Difference between an accurate regression model, under fitting and over fitting according to [7].

4.1.3 Over- and Underfitting

A very common problem in machine learning is the appearance of overfitting or underfitting [102]. A statistical model or ML algorithm is termed *overfitted*, if it memorizes the set of training data. The model is specialized to the training data and does not recognize the underlying pattern or system. If noise or random variations in the training data are picked up and learned during training, it is not able to generalize new data sets. This occurs when the model is too complex for the problem, for example a higher order polynomial model for a linear regression task. In the case of *underfitting*, the statistical model is too simple for describing the data, as is the case with a linear model for quadratic data. Figure 4.4 illustrates the difference between an overfitted and underfitted model compared to an accurate model for the example of a regression task.

To verify that a model accurately captures the underlying pattern, it should be tested on data that was not used during training. For this purpose, the existing data set is divided into training and test data. The model is trained by using the training data and then evaluated by using the test data. A typical split is that 70% of the data is used for training and 30% for testing. Another effective preventive measure against misfitting is cross-validation, which is used for classification algorithms [103]. The basic principle consists of dividing the entire training set into k subsets of equal size and is thus also referred to as k-fold cross-validation. Therefore, the model is trained k times, where in turn each time a different subset is used as validation set, and the remaining $(k-1)$ subsets are training sets. After k trainings, all training results are averaged and returned as the result of the cross-validation. In the case of overfitting, the error in the test set is high while the error in the training set is small. In the case of underfitting, the error in both the training set and the test set is usually high. This technique produces a more reliable estimate of the accuracy of the network. The larger the number k of models,

the more reliable is the validation, but the longer the training takes. For this reason, small datasets are more suitable for cross-validation. Other common ways to avoid over- or underfitting are to train with more data, delete irrelevant input features, insert a regularization method or tune hyper parameters.

4.2 Genetic Algorithms

Genetic algorithms (GA) are searching algorithms for optimizing functions [104]. They are based on the principles of evolutionary theory according to Charles Robert Darvin (1809-1882) [105] and have been developed for the first time by John H. Holland at the beginning of the 1960s. To imitate the principle of nature, an artificial population is created that passes through an evolution. The adaptation to an underlying problem is done by changing the genetic information from generation to generation. Genetic algorithms do not require problem-specific information about the task to be solved and can therefore be used where the problem cannot be well understood or their solution is not solvable due to computational or mathematical reasons. The basis of a genetic algorithm is formed by target objects, also called individuals, which are optimized during training. A set of several individuals represents the entire set of individuals contained in the algorithm and is called a population. The population is created at the beginning with a number of randomly generated individuals. In biological context, an individual is an organism whose genetic information is stored in a set of chromosomes. Within the context of genetic algorithms, the terms individual and chromosome are mostly synonymous. The size of the population should be such that the search space is sufficiently represented. The optimization process takes place over a set of time steps, also called generations, during which the content of the population changes. A specific position in the individual is called a gene. Each individual contains exactly one gene in which the genetic information is contained, that is used for the evaluation of an individual.

Fig. 4.5 schematically illustrates the process of a GA. As already mentioned, the algorithm starts with a randomly generated number of individuals as the initial population. In each generation, each individual is evaluated by a fitness function. After determining the fitness value of the initial population, two individuals are selected from the population based on their fitness value and form a parent pair. Using the genetic information of the parents, the offspring is generated out of a recombination or mutation process and forms the next generation. This process is repeated until a termination condition is met. In the following, the fitness function, selection, recombination and mutation are further explained. The fitness function measures the quality of an individual with respect to the problem to be optimized and has to be formulated at the beginning of the optimization. Once each individual has received its fitness value, they are compared and selected for the next generation depending on the fitness values.

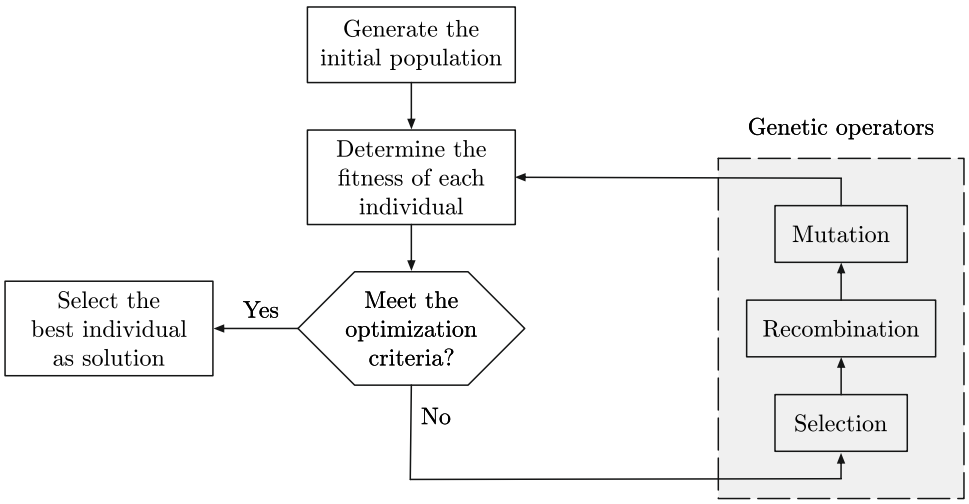


Figure 4.5: Flow-chart of a genetic algorithm.

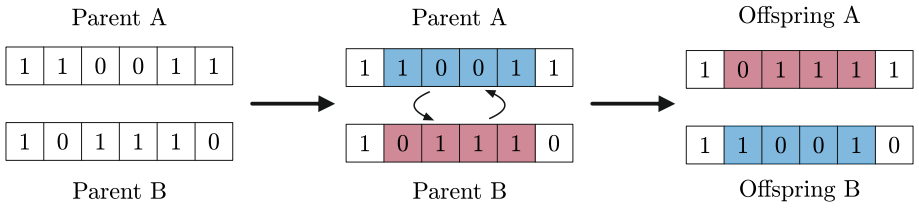


Figure 4.6: Example of a 2-point crossover recombination.

This selection of individuals with the highest fitness value improves the average fitness value of the population. Different methods can be used for selection [106], such as roulette wheel selection [107], tournament selection [108] or rank-based selection [106].

Recombination is the most fundamental part of the evolutionary progress: it changes the genetic information of individuals. Specifically, partial sequences from at least two individuals are being exchanged or recombined with each other. The recombination points are located between the genes of an individual. There are different methods for recombination [109]: the 1-point crossover, 2-point crossover or the uniform crossover method. The example of the 2-point crossover method is shown in Fig. 4.6. The recombination is done between two randomly chosen crossover points. However, it may happen that a subsequence important for fitness is exchanged by a weak subsequence. The success of recombination also depends on the diversity of the population. If the population is homogeneous, the exchange of genetic information has a minor or even no effect.

To ensure that the algorithm is not oriented to a local optimum, it is important to provide an impulse to guide the individuals back to a global optimum. For this purpose, the mutation occurs with probability p and changes an individual slightly. This results in a new individual which is moving away from the local optimum

4.2.1 Genetic Algorithm based Capacity Optimization

For the system optimization discussed in Chapter 6, bit rate, power, GOSNR and margin form an individual for all channels, where bit rate and power are the parameters to be optimized and GOSNR and margin are calculated from bit rate and power using the GN model. Then, the individuals are sorted by total bit rate and finally by margin. The new population of size 64 or 50 of the next generation is generated by replacing the $N/2$ worst individuals with mutated and recombined individuals for 100 generations.

Recombination: $N/4$ new individuals are created by combining two randomly selected individuals from the good N/2 individuals by combining one half of the channels of one individual with the second half of the second individual.

Mutation: $N/4$ new configurations are created by applying the following changes to the power or bit rate to $N/4$ randomly selected individuals from the $N/2$ good individuals. First one channel is selected to be modified with the following probabilities:

- With probability 0.5 select a random channel.
- With probability 0.25 select the channel with the lowest bit rate.
- With probability 0.25 select the channel with the highest margin.

Then the power and bit rate of the selected channel are modified by an offset drawn from a normal distribution $\mathcal{N}(0, \sigma)$ with $\sigma_{\text{bit rate}} = 100\,\text{Gbit/s}$ for the bit rate and $\sigma_{\text{power}} = -3\,\text{dBm}$ for the power.

4.3 Support Vector Machines

The term support vector machine (SVM) describes an ML algorithm that is suited for regression and classification tasks [110]. The method derived by Vladimir Vapnik, Bernhard Boser and Isabelle Guyon in [111], which is mainly based on the foundations in statistical learning theory developed by Vapnik, uses quadratic optimization problems to distinguish two classes in a feature space. The SVM is most commonly used for classification tasks, and therefore only this type of task will be discussed in this work.

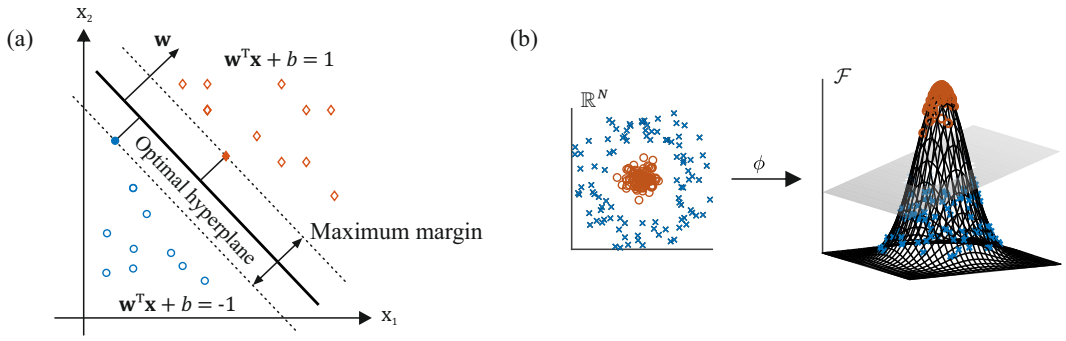


Figure 4.7: (a) Separation of two data classes in the input space by an optimal hyperplane according to [112]; (b) Exemplary mapping to a higher dimensional feature space using a nonlinear kernel function according to [113].

4.3.1 Linear Classification

Based on binary classification, the SVM determines from a set of training data $S = (x_1, y_1), ..., (x_n, y_n), x_i \in \mathbb{R}^N, y_i \in \{\pm 1\}$ a hyperplane $h(x)$ that separates the input data in two classes. The classification function for the separation can be expressed as

$$y = \text{sign}(\mathbf{w}^T\mathbf{x} + b), \tag{4.12}$$

where $\mathbf{w}$ and b are an orthogonal vector to $h(\mathbf{x})$ and a bias term, which are determined during the training process. With regard to the use of new and unknown data sets, the optimal hyperplane should have a high degree of reliability. This is achieved by placing the hyperplane at the greatest possible and preferably equal distance from the boundary points of the data classes. This results in an area, the maximum margin, that remains free of features, as shown in Fig. 4.7.
The optimal hyperplane depends only on the nearest data points, the boundary points, represented as solid dots. These are the so-called support vectors to which the SVM owes its name, since only these are necessary to mathematically describe the hyperplane [114].

The data of the feature vectors do not always occur in an ideally separable area. For example, if a data point of the training set lies in the opposite class, it is not possible to separate them successfully. For this reason, the optimization problem is extended by an error term including a weighting coefficient C and the slack variables ξ [115]

$$\min_{\boldsymbol{w},b,\xi} \|\boldsymbol{w}\|^2 + C\sum_{i=0}^{N} \xi_i \tag{4.13}$$

under constraints

$$y(\boldsymbol{w}^T\boldsymbol{x} + b) \geq 1 - \xi_i \tag{4.14}$$

$$\xi_i \geq 0, i = 1, ..., n. \tag{4.15}$$

To solve the optimization problem, the method of Lagrange multipliers, optimization under constraints, is used [116]. This method introduces a new unknown scalar variable for each constraint, a Lagrange multiplier. For this purpose, the optimization problem is converted into a dual form, a QP-problem, where the objective function Ψ depends purely on a set of Lagrange multipliers α

$$\min_{\boldsymbol{\alpha}} \Psi(\boldsymbol{\alpha}) = \min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j (\boldsymbol{x}_i^T \boldsymbol{x}_j) \alpha_i \alpha_j - \sum_{i=1}^{N} \alpha_i \tag{4.16}$$

subject to the inequality constraints

$$0 \leq \alpha_i \leq C, \forall i, \tag{4.17}$$

and one linear equality constraint

$$\sum_{i=1}^{N} y_i \alpha_i = 0. \tag{4.18}$$

To find the optimal values of the Lagrange multipliers, special solution algorithms are used, which are presented in Section 4.3.4 [116].

4.3.2 Nonlinear Classification

If highly distorted patterns occur, the SVM transforms the input data into a higher dimensional space $\phi : \mathbb{R}^n \to \mathcal{F}$, in which the data can be separated linearly. According to Mercer's theorem, the intensive computation of the scalar product in feature space can be significantly reduced by a suitable kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ [116]. By using different kernels, different problems can be solved, opening up a wide variety of SVM learning machines. The linear kernel denotes the classical linear calculation based on the scalar product of two feature vectors and can be used in case of no or very low correlation between the data points of both classes. The polynomial kernel function is an extension of the linear kernel, the order of which can be set with a parameter k. The kernel function here provides an efficient way to compute polynomials in the extended space, since a computation in the higher dimensional space on the polynomial basis would be very costly due to the numerous mixing terms. The third kernel approach is called radial basis function (RBF) and resembles a Gaussian distribution,

which is why the term Gaussian kernel is often used. The parameter $\gamma > 0$ specifies the width of the kernel and is thus the central variable in an optimization process [117]. The typical kernel functions are summarized in Tab. 4.1 [112]. Fig. 4.7 shows an exemplary use of the RBF kernel: the classes mapped in the input space can only be linearly separated in the feature space.

Table 4.1: Kernel Functions.

Kernel	Function
Linear:	$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
Polynomials:	$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^k$
RBF:	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$

4.3.3 Multi-class Classification

Although the SVM is a binary classifier, the classification of multiple classes is still possible. For this purpose, several binary SVMs can be combined [118, 119, 120, 121]. There are different methods for combining SVMs, such as the *one-vs-one* (OVO), *one-vs-all* (OVA) or the *binary coding SVM* (BCSVM), which are discussed in the following.

The OVO method is based on comparing two different classes from the total set of classes until every possible pairing of classes has been run through. If the dataset contains N different classes, $N(N-1)/2$ different SVMs are trained. For the final decision, a voting scheme is used and the class with the highest vote is determined [119]. In contrast to the OVO method, the OVA method separates one class from the set of all $N-1$ remaining classes. Thus, SVMs are trained for N different classes. At the end, the class with the highest vote is detected [118]. The BCSVM has been developed especially for communication systems. It takes advantage of the fact that the symbols of a modulation format already occur in binary form, so that an SVM is trained for each bit. For a modulation format with N symbols $\log_2(N)$ SVMs are needed [122].

Fig. 4.8 shows, exemplary for 16-QAM, the principle of the BCSVM. The corresponding SVMs are color-coded with the associated bits. The processing structure is illustrated in Fig. 4.8 (b). The complex input vector $\boldsymbol{x}_{\mathrm{rx}}$ contains the received symbols, divided into real and imaginary parts. Training is performed based on the transmitted bit sequence $\boldsymbol{y}$. The separating hyperplane of each SVM is determined during training by quadratic programming. The output of the binary SVM array is an

estimate from the transmitted bit sequence $\hat{\boldsymbol{y}} = [\hat{y}_1, \hat{y}_2, \hat{y}_3, \hat{y}_4]$. Consequently, the received signal is classified and demodulated simultaneously.

Conclusively, Fig. 4.9 illustrates exemplary a comparison of the three methods for one iteration step during training. The data points of the competing classes are highlighted in different colors. It can be seen that in contrast to the OVO method, the OVA and BCSVM methods take all data points into account, which can be very computationally intensive for large training sets. Nevertheless, the OVO method requires more calculation steps than the other methods.

4.3.4 Quadratic Programming

The standard quadratic form of the optimization problem as introduced in Eq. 4.13 can be solved using common quadratic programming (QP) [116]. Due to the often large amounts of data, QP problems quickly become highly complex structures and thus quickly become too computationally intensive. Instead, a decomposition method such as the sequential minimial optimization (SMO) algorithm [124, 125] can be used, which speeds up the computation and avoids running out of memory. In this process the QP-problem is decomposed into several QP-subproblems, consisting of two Lagrange multipliers. In each iteration, the algorithm selects two pairs of the training data based on a selection rule and solves the Lagrange multipliers using the approach described in [124] to optimize the entire SVM. The analytical calculation of the QP-subproblems, avoids the need for computationally intensive numerical methods. Moreover, SMO does not require additional matrix storage and is thus suitable for very large SVM training problems.

A further algorithm for solving the optimization problem is the iterative single data algorithm (ISDA) [126]. In contrast to the SMO, where at least two pairs of training data are required, the ISDA uses an iteration routine based on single training data. ISDA is often performed without the bias term by adding a small positive constant to the kernel function. Omitting the bias term removes the sum constraint in the dual equation (Eq. 4.18). This allows a Lagrange multiplier to be updated at each iteration, a feature that makes removing outliers easier than with the SMO. A simulative comparison of the QP solution algorithms in combination with the different multi-classes SVM methods is presented in Section 5.1.1.

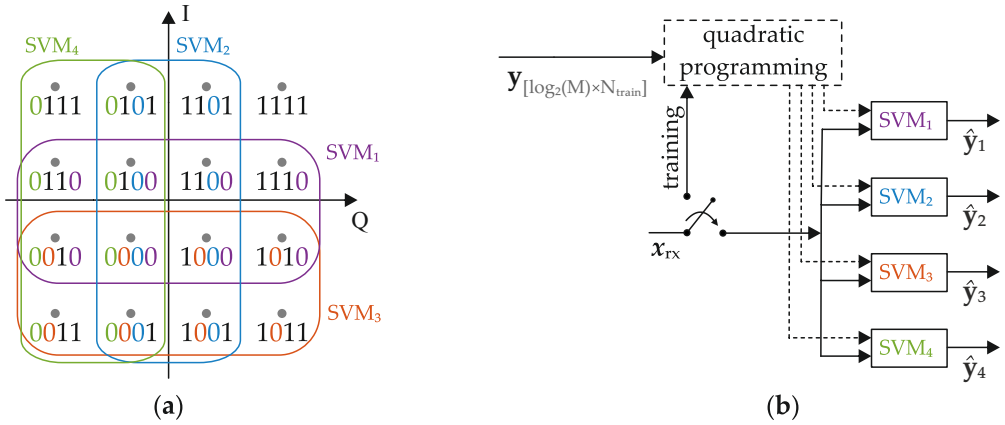


Figure 4.8: BCSVM for a nonlinear classification of a 16-QAM signal [123].

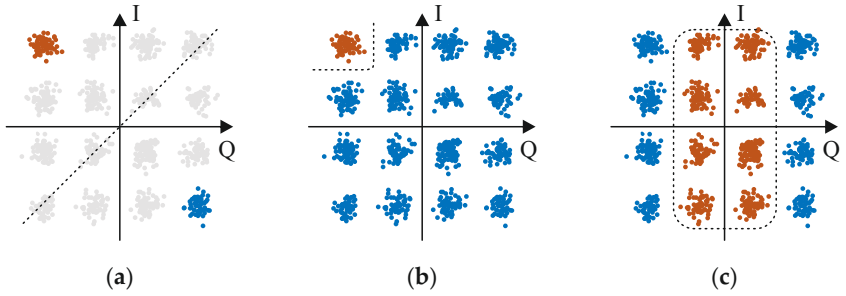


Figure 4.9: Illustration of one iteration during training for (a) OVO, (b) OVA and (c) BCSVM methods in case of 16-QAM transmission. The opposite classes are marked in red and blue and the corresponding hyperplane is indicated by the dashed line [123].

4.4 Reinforcement Learning

Reinforcement learning is a learning technique based on the interaction with an environment, from the consequences of actions, rather than from explicit instructions [127]. Specifically, an agent interacts with an environment as shown in Fig. 4.10. The environment describes the world in which the agent is located and changes its state based on the agent's behavior. After each action, the agent receives feedback from the environment, depending on the action. Based on the experience, the agent can learn a behavior that leads to a certain goal, for example by repeating actions that lead to good rewards.

4.4.1 Markov Decision Process

An RL environment can be described by a Markov decision process (MDP) [129, 127]. This consists of a set of states, a set of rewards, and a set of actions. The agent's goal is to maximize the sum of cumulative rewards. The set of states, the state space, models the observations of the environment and is defined as $S = \{s_0, ...s\}$. If the agent is able to see the full state, the environment is fully observable. If the agent sees only a part of the complete state, the environment is partially observable. The size of the state space can be finite or infinite. The agent chooses from a finite or infinite set of possible actions $A = \{a_0, ..., a\}$ to change the state. If the state/action set is finite, the state/action space is called discrete. If the set of possible state/actions is infinite, the variables of the states and actions correspond to continuous values and the state and action space is called continuous.

The agent interacts with its environment at different time steps. At time step t, the agent is located in state s_t and chooses an action a_t based on the current state. This decision process is based on the agent's policy π. Formally, a distinction is made between a deterministic policy $\pi(s)$, meaning that the state s always leads to the same action a, and a stochastic policy. A stochastic policy is described as $\pi(a|s) = Pr(a_t = a|s_t = s)$ and describes the probability that action a will be performed in state s at time t. If the environment is stochastic, the new state is obtained based on the transition probability distribution: $s_{t+1} \sim P(.|s_t, a_t)$. It gives probability of ending in state s_{t+1} at time $t + 1$, if the agent

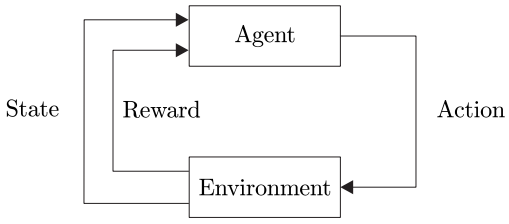


Figure 4.10: Interaction between an agent and the environment according to [128].

was previously in state s_t and performed action a_t. The transition function depends only on the current state and not on the states the agent has visited before. Thus, the Markov property is satisfied. This states that the future depends only on the present and not on the past. If the environment is deterministic, this distribution can be simplified to a transition function. After each action, the agent receives a reward r_t. This is based on the reward function $R(s_t, a_t, s_{t+1})$ and quantifies the quality of the immediate decision made. The value for this can be positive or negative. The above parameters formulate the tuple (S, A, P, R) and thus form the mathematical basis of reinforcement learning: the Markov decision process.

The agent's high level goal is to maximize the expected cumulative reward (often called return) over an episode or trajectory $\tau = (s_0, a_0, s_1, a_1, ...s_T, a_T)$. An episode describes the behavior of the agent over time starting with the state-action pair (s_0, a_0) and ending at $t = T$. Here T is the time step in which the environment is reset to the initial state. The time step T is reached when this either reaches a predefined number of time steps, or when the agent ends up in a final state. For episodic tasks, the horizon T is finite, while for continuous tasks T is infinite [127].

The performance of a strategy is determined by an estimate of the expected (finite-horizon) discounted rewards, which is given by

$$\mathcal{J}(\pi) = E_\pi[G] = E_\pi\left[\sum_{t=0}^{T} \gamma^t r_t\right] \tag{4.19}$$

using the *discount factor* $\gamma \in (0, 1]$, which provides a trade off between short-term (immediate) and long-term rewards, and prevents the sum of rewards received from converging to infinity in continuous tasks [129]. Typically, a value close to 1 is chosen for γ so that future rewards still have an impact on the agent's decision making. If the agent finds a strategy that maximizes $\mathcal{J}(\pi)$, it is called the optimal strategy π^* and is a solution of the MDP.

4.4.2 Model-Free vs. Model-Based RL

RL algorithms can be categorized principally into two categories: model-based and model-free RL [129, 127]. Thereby, model refers to a function that models the environment and predicts state transitions and rewards.

As the name implies, in model-free RL the model of the environment is not present. The agent has no insight into how the environment looks like or may change after a performed action. So the agent learns a value function for each state of the state space with which the agent constructs

a strategy. Or the agent learns the strategy directly from experience only, i.e. by observing the rewards and next states obtained by interacting with its environment. Model-free algorithms can successfully solve various tasks playing video games and solving robot tasks. The disadvantage is that they need many samples to realize good performance. Model-free algorithms include for example A2C/A3C [130], PPO [131] or TRPO [132], which will be further explained in the following sections.

In model-based RL, the agent attempts to understand its environment and creates a model for it based on its interactions with that environment. By using this model, the algorithm can predict the state transitions and rewards. The agent calculates a kind of value function and obtains a strategy from it. The learned model allows the agent to see in advance what will occur based on a set of possible decisions. Thus, the agent is able to incorporate the results of the advance planning into its learning strategy. A well-known model-based algorithm is AlphaZero, which achieves capabilities far beyond those of humans [133].

4.4.3 Value-Based vs. Policy-Based RL

Model-free RL can further be divided into Q-learning and policy optimization, which belongs to value-based and policy-based algorithms, respectively [129, 127].

The goal of value-based methods is to estimate the value function, which is a valuation scheme for states. Depending on the strategy with which the agent chooses the next action, it estimates the expected discounted return for an infinite time horizon and thereby measures the value of the state or the state-action pair. In other words, it is an estimate of how good it is for the agent to be in a particular state, or to perform a particular action in that state. The On-Policy Value Function is defined by [88]

$$V^{\pi}(s) = E_{\pi}\left\{\sum_{t=0}^{\infty}\gamma^{t}r_{t}|s_{0}=s\right\} \tag{4.20}$$

and determines the value of a given state s, i.e., the total reward the agent can expect, if it starts in state s and follows strategy π. The action-value function, also called Q-function, is defined by [88]

$$Q^{\pi}(s,a) = E_{\pi}\left\{\sum_{t=0}^{\infty}\gamma^{t}r_{t}|s_{0}=s,a_{0}=a\right\} \tag{4.21}$$

which determines the value of a given action in a given state. This is given by the total reward the agent can expect if it starts in state s, performs action a, and follows policy π [129].

With $V^{\pi}(s = E_{a\sim\pi}[Q^{\pi}(s,a)])$ a relation can be drawn between the value function and the Q-function. Here $a \sim \pi$ means that the action a is chosen according to the policy π. Maximizing over all states yields the optimal value function $V^*(s) = \max_{a\in A} V^{\pi}(s)$ or the Q-function $Q^*(s,a) = \max_{a\in A} Q^{\pi}(s,a)$, from which the optimal policy $\pi^*(s)$ is derived.

The relationship between two successive states for the value or Q-function is expressed by the Bellmann equation according to [88]

$$V^{\pi}(s) = E_{a\sim\pi, s'\sim P(.|s,a)}[r(s,a) + \gamma V^{\pi}(s')] \tag{4.22}$$

$$Q^{\pi}(s,a) = E_{s'\sim P(.|s,a)}[r(s,a) + \gamma E_{a'\sim\pi}[Q^{\pi}(s',a')]] \tag{4.23}$$

with $s' \sim P(.|s,a)$, meaning that the next state s' is sampled according to the transition probability distribution P of the environment. The Bellman equation is a basic principle of optimization and states that a strategy for a given problem is optimal if all subproblems are optimally solved.

The advantage of a state-action pair is the relative measurement for the benefit of an action in a given state for the given policy and is defined by [88]

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s,a). \tag{4.24}$$

One of the best known value-based algorithms is Q-learning [134]. In Q-learning, Q-values are estimated step by step and stored in a table. Each Q-value corresponds to a state-action pair and represents an action performed in that state. A better Q-value means better chances of greater rewards. The algorithm converges to the optimal strategy assuming that each state-action pair has been visited infinitely often [134, 135]. Specifically, a learned action value function, $Q_{\phi}(s,a)$, is used to approximate the optimal action value function $Q^*(s,a)$. Using the Bellmann equation of the Q-function, the estimate of the current Q-value is updated according to

$$\hat{Q}(s,a) \leftarrow \hat{Q}(s,a) + \epsilon \left[r(s,a) + \gamma \max_{a'} \hat{Q}(s',a') - \hat{s,a} \right] \tag{4.25}$$

with the learning rate $\epsilon \in [0,1]$, which indicates how fast the agent learns. The lower ϵ, the more short-term rewards are weighted compared to long-term rewards. The agent chooses its action according to [88]

$$a(s) = \arg\max_{a\in A} Q_{\phi}(s,a). \tag{4.26}$$

The dimensions of the Q-table are defined by the number of possible states and actions. With high dimensions, the following problems arise: on the one hand, storing and updating the Q-table requires a lot of memory, and on the other hand, a lot of time is required until a Q-value has been determined for each state. For the reasons mentioned above, a *function-approximator* is used to approximate the Q-function and thus interpolate Q-values. Usually, however, deep neural networks are used (deep Q-learning), although any other kind of regressor is theoretically suitable (linear algorithm, radial-basis function network, SVR, ...). A function approximator depends on a set of parameters θ, which can be, for example, the weights and biases of an NN. The approximated Q-value is therefore also denoted by $Q(s, a; \theta)$ or $Q_\theta(s, a)$. An extension of deep-Q learning is the deep-Q network, first proposed in [136]. Here, the input data for the NN is stored in a huge buffer, also called replay buffer, from which randomly generated minibatches are then formed to train the NN. The replay buffer prevents the correlation of input and output values.

A major disadvantage of value-based algorithms is the so-called bootstrappping [137]. This means that the estimation of the value function is based on previous estimates, which can lead to distortions and also to the fact that the algorithm does not converge.

In policy-based methods, the policy is estimated directly. A common method is policy optimization, a policy gradient method in which the parameterized policy $\pi_\Theta(a|s)$ is learned directly. Using the learned policy, the action can be selected without needing the value function. For the policy updates, the gradient descent method on the performance objective $\mathcal{J}$ or a local approximation of it is used according to

$$\theta t + a = \theta t + a_t + \epsilon \nabla \hat{\mathcal{J}}(\theta_t). \tag{4.27}$$

The policy gradient can be simplified to the general form

$$\nabla_\theta \mathcal{J}(\pi_\theta) = E_{\pi_\theta} \left[\sum_{t=0} T \nabla_\theta \log \pi_\Theta(a_t|s_t) \Phi_t \right], \tag{4.28}$$

where Φ_t can appear in several valid forms. One of the first established forms is

$$\Phi_t = \sum_{t'=t}^{T} r'_t - b(s_t) \tag{4.29}$$

with the baseline function $b(s_t)$, which serves as a comparison for the quality of the chosen action value. The baseline function was introduced because it reduces the variance of the update without

changing its expected value. One of the first policy-based algorithms was REINFORCE [138] with $\Phi_t = Q^{\pi_\theta} - b(s_t)$. With policy-based algorithms, the environment is naturally explored by the agent using stochastic policies and continuous action spaces.

4.4.4 On-Policy vs. Off-Policy

The methods presented so far further differ in how the data, i.e., the collection of the agent's experience, is generated. Typically, experience is gained with the last strategy learned and then used to improve the strategy. This is a type of online interaction. The agent uses the same policy that it optimizes, which is called on-policy method. These methods do not learn the action values for the optimal strategy, but for a near optimal strategy since this is still being explored. The advantage of on-policy methods lies in the fact that the data for the policy update come from the same state visit distribution as that of the currently adopted best strategy, which makes them generally more stable.

In classical off-policy methods such as deep deterministic policy gradient (DDPG) [139, 140], the agent's experience is attached to a data buffer (also called a replay buffer), and each new policy collects additional data so that the data buffer consists of samples of the policies. To train a new policy, all of this data is then used. The agent interacts with the environment to collect the samples. The ability to reuse data from other policies generally makes off-policy methods much more efficient.

Striking a good balance between choosing optimal policies and exploring the environment to possibly find better options is not trivial, neither for off-policy nor for on-policy methods. This problem is often referred to as the exploration-exploitation dilemma [129].

4.4.5 Algorithms

Presented in the following is an overview of the model-free algorithms used in this work.

Actor-Critic

With the basis function, introduced in Eq. 4.29, the variance of the gradient can be reduced. Nevertheless, it is still high and can lead to slower convergence. To counteract this, *Actor-Critic* (AC) algorithms are used [141]. For this method, the state value function $V_\pi(s,t)$ (critic) is learned in parallel to the policy (actor) and used as a baseline. The biased baseline reduces the variance of the estimate of the political gradient, making the training process faster and more stable. Instead of the Q-value, also the advantage value can be used which leads to the so-calles advantage actor-critic

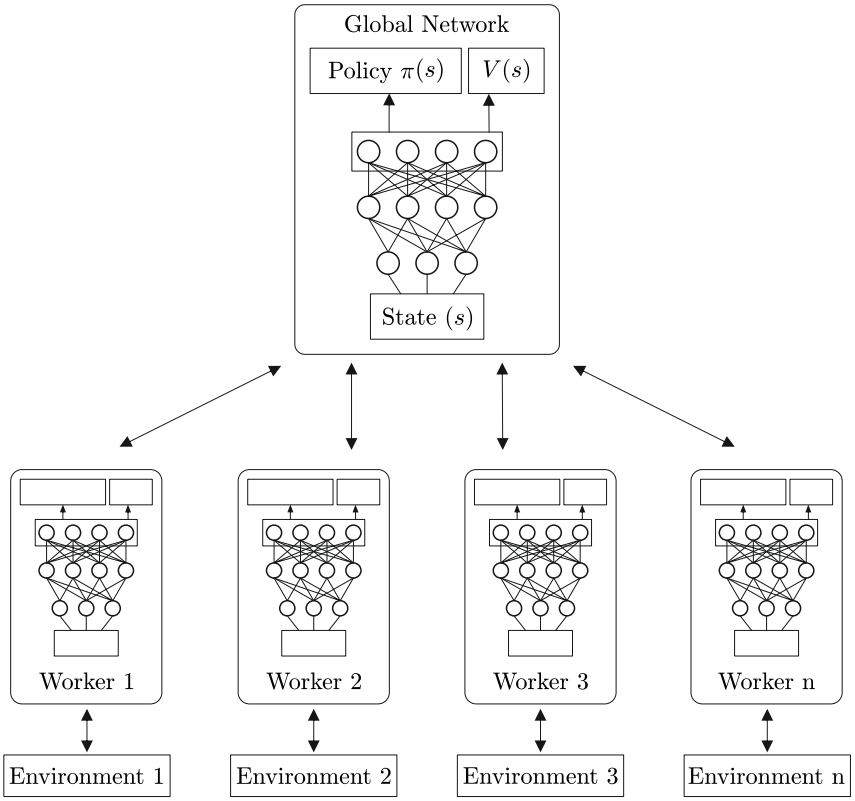


Figure 4.11: Architecture of an A2C/A3C algorithm based on [128].

(A2C). The advantage is calculated according to [130]

$$\begin{aligned} A^{\pi}(s_t, a_t) &= Q^{\pi}(s_t, a_t) - V_{\phi}^{\pi}(s_t) \\ &= r_t + \gamma V_{\phi}^{\pi}(s_{t+1}) - V_{\phi}^{\pi}(s_t). \end{aligned} \tag{4.30}$$

The update of the stochastic policy π (actor) is done by stochastic gradient increase with gradient estimator according to [130]

$$\nabla_{\theta} \mathcal{J}(\pi_{\theta}) = E_{\pi_{\theta}} \left[\sum_{t=0} T \nabla_{\theta} \log \pi_{\theta}(a_t|s_t) A^{\pi}(s_t, a_t) \right] \tag{4.31}$$

$$\theta_{k+1} = \theta_k + \epsilon \nabla_{\theta} \mathcal{J}(\theta). \tag{4.32}$$

The state of the art algorithm is introduced in [130], where parallel training is implemented by having multiple workers in parallel environments independently update a global value function. The idea is

shown in Fig. 4.11. The actor and the critic are stored in a global network. Multiple instances of the environment are then created in different parallel agents (threads). At the beginning of each episode, each agent receives a copy of the actor's and critic's weights from the global network and a random start state. From this, each agent runs through an episode, calculating the accumulated gradients which it sends back to the global network. The global net summarizes the gradients and uses them to adjust the parameters of the actor's and critic's nets. Then the adjusted parameters are sent back to the agents until convergence is achieved. Each agent takes a different amount of time to calculate the parameters. Therefore there are two ways to update the global net: the global net waits until all workers are done (A2C) or each worker reads and writes to the global net when it wants, called asynchronous AC2 (A3C).

Trust Region Policy Optimization

Trust region policy optimization (TRPO) [132] is an on-policy method that updates policies in the largest possible increments to improve performance. At the same time, a special constraint is imposed on how close the new and old policies are permitted to be to each other. The constraint, also called the trust region constraint, is expressed in terms of Kullback Leibler (KL) divergence [142], a measure of the distance between probability distributions. The KL divergence between two subsequent policies is [132]

$$\overline{D}_{\mathrm{KL}}(\pi_{\phi_{\mathrm{old}}}||\pi_{\theta}) \leq \delta. \tag{4.33}$$

This results in the following new objective function, which is maximized taking into account the policy parameters ϕ and respecting the KL constraint.

$$\mathcal{J}(\phi) = E_{s\sim\rho^{\theta_{\mathrm{old}}},s\sim\pi_{\theta_{\mathrm{old}}}}\left[\frac{\pi_{\theta}(a|s)}{\pi_{\theta_{\mathrm{old}}}(a|s)}A^{\pi_{\mathrm{old}}}(s,a)\right] \tag{4.34}$$

with $\rho^{\theta_{\mathrm{old}}}$ the state visitation distribution for policy $\pi_{\theta_{\mathrm{old}}}$. TRPO is applicable to environments with either discrete or continuous action spaces [132].

Proximal Policy Optimization Algorithm

Proximal policy optimization (PPO) [131] combines the idea of multiple workers from the A2C method with the concept of trust regions from the TRPO algorithm and is also an on-policy algorithm. Instead of the hard constraint for KL divergence between two subsequent policies, the PPO uses clipping and penalty to ensure the trust regions [131].

The clipping variant of PPO indicates how far the new policy may deviate from the old policy by the probability ratio $r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}(a|s)}}$ between the old and new policies in the range $[1-e, 1+e]$. This yields the following objective function without additional constraints according to [131]

$$\mathcal{J}^{\text{CLIPPING}}(\theta) = E_{s\sim\rho^{\theta_{\text{old}}}, s\sim\pi_{\theta_{\text{old}}}}\left[\min r(\theta) A^{\pi_{\text{old}}}(s,a) \text{clip}(r(\theta), 1-e, 1+e) A^{\pi_{\text{old}}}(s,a)\right]. \tag{4.35}$$

The penalty version of PPO adds the KL divergence between the old and new policies as a penalty by subtracting it from the target [131]

$$\mathcal{J}^{\text{penalty}}(\theta) = E_{s\sim\rho^{\theta_{\text{old}}}, s\sim\pi_{\theta_{\text{old}}}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} A^{\pi_{\text{old}}}(s,a)\right] - \beta_{\text{RL}} \overline{D}_{\text{KL}}(\pi_{\phi_{\text{old}}} || \pi_\theta). \tag{4.36}$$

The hyper parameter β_{RL} controls the influence of the KL penalty and can be adjusted during the training process.

Chapter 5

Support Vector Machine in Optical Transmission Systems

High-order quadrature amplitude modulation (QAM) formats is a promising technique for achieving high spectral efficiency. Meanwhile, higher order modulation is susceptible to nonlinear distortion due to the dense constellation and required high SNR. Such distortions include e.g. modulation nonlinearities in the short-range and nonlinear Kerr distortion in the long-haul transmission. One of the main impairments is nonlinear phase noise, which is caused by the interaction between the signal and the ASE noise of the inline optical amplifiers [143, 144]. Furthermore, the I- and Q-components of the signal should be ideally orthogonal to each other. Implementation imperfections as incorrect settings of the bias points of the I/Q-MZM, insufficient splitting ratio of the couplers, incorrect sensitivity of the photodiodes or the incorrect setting of the polarization controllers lead to quadrature imbalances in a communication system [145].

The aforementioned limitations complicate the correct detection of high-order QAM at the receiver side, so it is important to compensate for them. Basically, two approaches can be considered for this. The first one involves modifying the signal in such a way that the impairments are ideally reversed. These include, for example, the Gram-Schmidt orthogonalization procedure (GSOP) [146, 147] used to compensate for I/Q imbalances, blind moment estimation (BME) [148], and also the 4x4 MIMO equalizer presented in Section 3.2.3. To compensate for the fiber nonlinearities, digital back propagation (DBP) has already been extensively investigated [149, 150, 151]. However, this method requires fiber information as well as a high computational effort due to the FFT and IFFT. The NLVE presented in Section 3.2.3 is able to compensate modulation nonlinearities as well as fiber nonlinearities. But its complexity increases strongly with an increasing number of coefficients. Recently, attention has been

focused on the techniques of ML, for example, the application of ANNs has gained much attention in this context. Here, the algorithm tries to find a functional description of the given training data with the goal of predicting values for new inputs, which is called *regression* problem [112]. For example, using an ANN in a long-haul scenario demonstrated a similar improvement in BER as digital backpropagation (DBP) with 2 steps per span (StPS) and 2 SpS [152, 153]. In addition, a bidirectional LSTM simulated over metro and long-haul links achieved better performance than DBP with 6 StPS [154, 153].

Another method is to adaptively and nonlinearly adjust the previously static linear decision boundaries to the received signal constellation, which is called *classification* or *clustering* problem. Suitable algorithms for this purpose are unsupervised algorithms such as Expectation Maximization (EM) or k-means. The EM algorithm assumes that each cluster is Gaussian distributed with different mean and covariance matrices. It iteratively optimizes the parameters of the mean and covariance matrices to fit the observed signal distribution as best as possible in terms of maximum likelihood [112, 87]. In contrast, the k-means algorithm assumes that the clusters have the same covariance [87, 155, 156, 7]. In the case that training data is available, the SVM or an ANN are potential ML algorithms for nonlinear classification. They can be used to generate nonlinear decision thresholds as a basis for the symbol decision. Compared to the ANN, the SVM possesses several advantages. For example, the SVM requires only a small amount of a training dataset because it reliably determines the decision boundary based on the sole support vectors [157]. Whereas the ANN, in contrast, requires the entire training dataset. It follows that the training of the SVM is completed much faster than the training of the ANN. Furthermore, it has been shown in [158] that the SVM with its asymmetric soft margins achieves a better trade-off between false positive and false negative decisions. The use of a SVM in optical transmission systems has already been studied in [159, 122]. It has been shown to be a powerful tool for compensation of nonlinear phase noise. For this reason, it is natural to explore the SVM further. Therefore, this chapter investigates and discusses the use of SVM based detection to compensate for modulation nonlinearities, I/Q imbalances, and fiber nonlinearities in an optical transmission system.

In this chapter the various SVM based classification methods presented in Section 4.3.3, for a 16-QAM transmission, are examined. This is followed by an increase of the modulation format to 64-QAM and a comparison with the k-means algorithm (KMA). In addition to the KMA, the NLVE serves as a benchmark algorithm. The compensation of the 2^{nd} order nonlinearities by using the SVM are investigated for a single sideband transmission with DD. Finally, the results are validated experimentally.

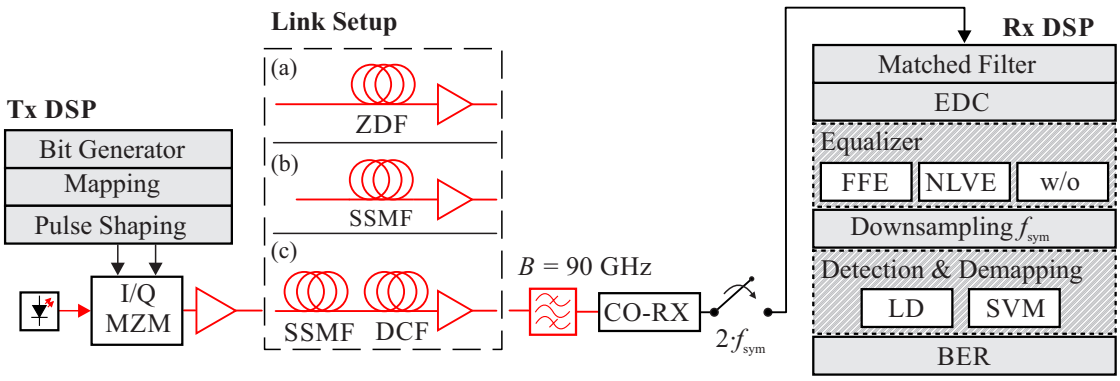


Figure 5.1: Simulation setup of the 64 GBaud 16-QAM single-polarization coherent optical simulation system including three different setups for the link: (a) zero-dispersion link, (b) dispersion uncompensated link and (c) dispersion compensated link.

5.1 Simulation Results

In order to investigate the use of an SVM in optical transmission technology, the SVM is examined by means of simulations first. For this purpose, a coherent 64 GBaud 16-QAM long-haul transmission system is simulated and a comparison of the performance is made. To evaluate the influence of the various transmission effects on the signal propagation and thus equalizability at the receiver, three types of links are under consideration. The transmission links to be presented are the *zero-dispersion link* (ZDL) with an idealized fiber without dispersion, the *dispersion uncompensated link* (DUL) with a receiver-side dispersion compensation and the *dispersion compensating fiber link* (DCL) with a dispersion compensating fiber after each SSMF. Furthermore, the ML based detection is compared with algorithms from digital signal processing: the FFE and the NLVE. Some results have been published in [123].

5.1.1 16-QAM Transmission over 1200 km

The simulation setup used for the numerical investigations is given in Fig. 5.1. All simulations are done using MATLAB (R2014a). A 2^{16} randomly generated bit sequence, using the MATLAB rand function, is mapped to the 16-QAM symbols. The baud rate is kept at 64 GBaud resulting in the total gross bit rate of 256 Gbit/s for the system under consideration. Using an RRC filter with the roll-off factor $\alpha_{\mathrm{RRC}} = 0.3$ the pulse shaping and a digital-to-analog conversion with 32-fold oversampling is applied. The symbols are modulated onto the carrier ($\lambda_c = 1550\,\mathrm{nm}$) via an I/Q MZM. The linewidth of the laser is set to zero. The modulated optical bandpass signal is coupled

Table 5.1: Characteristic values for the different types of fiber according to the links under investigation ($\lambda_c = 1550\,\text{nm}$).

Parameter			Value		
			ZDF	SSMF	DCF
Length	L	[km]	80	80	17
Attenuation	α	$[^{\text{dB}}/_{\text{km}}]$	0.2	0.2	0.5
Dispersion	D	$[^{\text{ps}}/_{\text{nm·km}}]$	0	17	-80
Dispersion slope	S	$[^{\text{ps}}/_{\text{nm}^2\text{·km}}]$	0	0.06	0
Nonlinear coefficient	γ	$[^{1}/_{\text{W·km}}]$	1.3	1.3	5.3

into the fiber after it is amplified by the EDFA with a noise figure of $F_N = 5\,\text{dB}$. In order to investigate the performance of the enhanced detection algorithms, three types of link sub systems are modeled.

Fig.5.1a illustrates the ZDL setup. It is used for the simulative analysis of signal propagation or signal detection in a completely dispersion-free transmission system in order to investigate the effects of nonlinear phase noise only. The dispersion coefficient D and the dispersion slope S are deliberately set equal to zero and thus have no influence on the signal. The advantage of the ZDL is the exclusive study of the nonlinear effects of the fiber. In particular, the nonlinear phase noise induced by the interaction between the ASE noise of the amplifiers and the nonlinear Kerr effect can be investigated without dispersive interference. Fig. 5.1b illustrates the DUL setup, which is a commonly used transmission link including digital dispersion compensation at the receiver. The signal is transmitted via an SSMF and amplified with an EDFA after each span. In the receiver-side DSP, electrical dispersion compensation (EDC) takes place after matched filtering, which neutralizes the dispersion as a linear filter in the frequency domain. Both dispersive and nonlinear effects act simultaneously on the signal in this link. The third type is the DCL setup, see Fig. 5.1c. In contrast to the DUL setup, the compensation of the dispersion takes place in the optical domain, iteratively after each span of the transmission link (which is typical for legacy fiber links).

Table 5.1 compares the fiber parameters of the ZDF, SSMF, and DCF setups. The values of the DCF are assumed to be semi-idealized, which means that the dispersion parameter of $D = -80\,^{\text{ps}}/_{\text{nm·km}}$ is chosen to correspond to the inverse value of the fiber length of the SSMF. Since the SSMF has a coefficient value of $D = 17\,^{\text{ps}}/_{\text{nm·km}}$, a DCF of length $L = 17\,\text{km}$ can compensate the dispersion completely.

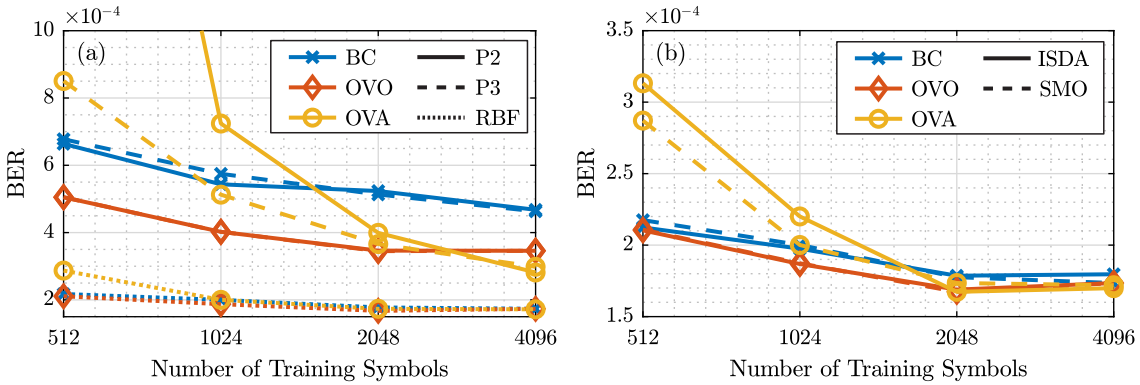


Figure 5.2: BER as a function of the number of training symbols for a) different kernels and b) different QP algorithms for a DUL transmission at a mean launch power of 0 dBm.

For a complete compensation of the span loss an EDFA ($F_N = 5\,\mathrm{dB}$) is applied. After transmission, a Gaussian optical filter with 90 GHz bandwidth is used to reduce the ASE noise. The received signal is detected by a coherent receiver and downsampled to 128 GS/s. After matched filtering an ideal EDC is used to compensate for dispersion, in the case of a DUL. After the equalization stage, which consists of either an FFE, an NLVE or no equalizer at all (w/o), the signal is downsampled to the symbol frequency and detected. Detection and demodulation are performed either linear by using conventional linear decision thresholds and demapping, here called linear detection (LD), or by an ML algorithm such as the SVM. The system performance is evaluated by the BER. The FEC limit is assumed to be $3.7 \cdot 10^{-3}$ corresponding to an overhead of 7 % [160].

Optimization of the SVM

The objective of this section is to find the most suitable multiclass SVM and tooptimize its parameters. The goal of the selection is to find the optimal training length, kernel method and QP solution algorithm. Therefore, the SVMs OVA-, OVO- and BC-SVM are analyzed considering the BER over the training length. Firstly, the nonlinear kernel methods RBF, polynomial second (P2) and third (P3) order are studied, with the consistent SMO method for the QP solution algorithm. Secondly, both QP algorithms are examined using the RBF kernel. Throughout the investigations, the SVM module without combined equalizer is used for demodulation and the number of training symbols is defined by the vector $N_{\mathrm{train}} = [512, 1024, 2048, 4096]$. The simulation results of the SVM configurations for a dispersion uncompensated transmission (DUL) with an input power of $P_{\mathrm{in}} = 0\,\mathrm{dBm}$ are shown in Fig. 5.2 where the BER over the number of training symbols is illustrated.

In Fig. 5.2a it can be observed that the RBF kernel achieves by far the best BER values even for a small amount of training data. Due to the Gaussian distributed transformation, also the Gaussian shaped symbol constellations can be best fitted in the two-dimensional original space. The BER of the polynomial kernels shows significantly higher fluctuations. Apart from the fact that a higher polynomial order requires a longer training, there is no trend that this leads to better results. It is noticeable that the OVA method achieves rather bad values for small training sets and the BC-SVM performs worst for a high number of training data. The former is due to the fact that for low total training lengths, only very few data are assigned to the individual classes, which are trained against a significantly higher residual amount, resulting in an unbalanced training process. In the OVO method, the order of the polynomial does not have any impact on the BER. Comparing the QP algorithms in Fig. 5.2b, the OVA method also performs poorly for a small number of training data. The ISDA algorithm performs less well than the SMO algorithm. In general, however, the difference is very small, so that it can be concluded that the algorithms have a similar quality in solving the QP problem and that the deviations are due to the different types of solutions (numerical, analytical). In summary, the RBF kernel reliably delivers the best results and is therefore used for the following investigations. The OVO method needs at least 1024 training symbols to detect adequately. Regarding the QP solution algorithms, there is only a small difference or no difference at all. In order to ensure comparability, the SMO algorithm is used in the following investigations.

Results

In the following simulations, the equalizer structures of the receiver-side DSP presented in Fig. 5.1 are compared for the three introduced links. The results of the BER are analyzed as a function of the fiber launch power. For comparability, the equalizers and the SVM are trained with 4096 symbols. As mentioned in Section 5.1.1, the equalizer stage consists of either a linear equalizer (FFE), an NLVE, or no equalizer at all. The notation FFE[4] or NLVE[4,2,4] refers to the number of coefficients for the FFE in the first order and the NLVE for the first, second and third order. The demodulation is then performed either linearly via conventional decision thresholds or nonlinearly by means of the SVM.

The simulation results of a ZDL transmission after 1200 km are shown in Fig. 5.3. Due to the lack of dispersion, noise, originating from the EDFA gain along the link, and nonlinear effects are mainly present. Consequently, the demodulated signal is affected by nonlinear phase noise. This results in a more non-Gaussian noise distribution, as visible in the constellation diagrams. Besides a significant phase shift, especially the outer constellation points are distorted due to the nonlinear impairments. The inner constellation points are less affected because of the lower power. The FFE[4] is only able to compensate the mentioned phase shift and therefore performs worst of all equalizers, where the

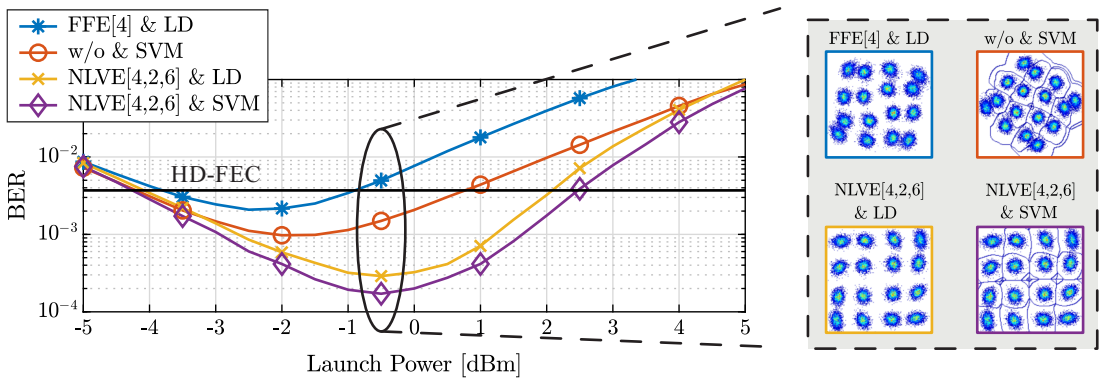
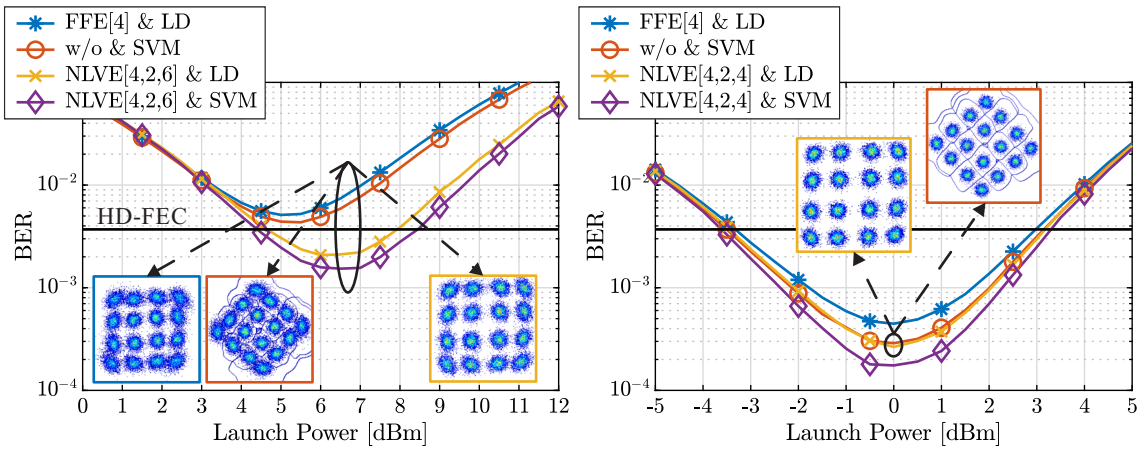


Figure 5.3: BER as a function of the launch power at 1200 km zero dispersion transmission including different structures of equalization and detection stages.

HD-FEC threshold is only barely exceeded between −3.5 dBm and −1 dBm launch power. The SVM increases the nonlinear tolerance significantly by 1.75 dB. The NLVE[4,2,4] improves it even by 3 dB with an optimum at −0.5 dBm and a BER of $2 \cdot 10^{-4}$. The combination of a NLVE[4,2,4] and a nonlinear decision unit, the SVM, allows a further improvement of the BER to $1.7 \cdot 10^{-4}$ and a gain of 0.5 dB at the HD-FEC threshold.

Fig. 5.4 illustrates the simulation results of a forward equalization after 1200 km of (a) dispersion managed transmission and (b) dispersion unmanaged transmission. The launch power of the 16-QAM signal of the DML ranges from 0 dBm to 12 dBm. In contrast to the ZDL, the transmission over a DML is influenced by dispersion. However, since the dispersion is compensated by a DCF after each span, the influence of the nonlinear effects is still high. The self-phase modulated NLPN is present in the constellation diagrams, especially in the outer constellation points. The minimum BER of the FFE at 5.5 dBm is $5.1 \cdot 10^{-3}$. The SVM can increase the nonlinear tolerance by 0.5 so that the minimum BER is $4.3 \cdot 10^{-3}$ at 5.5 dBm. Only by using the NLVE[4,2,6] the HD-FEC can be achieved between 4.5 dBm and 8 dBm. And as for the ZDL, the best performance is achieved with a combination of NLVE[4,2,6] & SVM.

In the case of dispersion uncompensated transmission, the launch power ranges from −5 dBm to 5 dBm. The BER characteristics of the SVM, the FFE[4], and the NLVE[4,2,4] are very close to each other. Between −3.5 dBm and 3 dBm the BER remains below the HD-FEC threshold for all equalization structures. The interaction between dispersion and nonlinearities result in more circularly symmetric Gaussian-like noise. The combination of NLVE[4,2,4] and SVM improves the optimal BER and the nonlinear tolerance slightly by 0.2 dB compared to the FFE[4].



(a) Dispersion Managed Link

(b) Dispersion Unmanaged Link

Figure 5.4: BER as a function of the launch power at 1200 km (a) dispersion managed transmission and (b) dispersion uncompensated transmission including different structures of equalization and detection stages.

5.1.2 Coherent 64 GBaud Transmission over 100 km SSMF

When increasing the order of the modulation format, the tolerance to I/Q-imbalances (a mismatch of phase and amplitude of the relative and imaginary parts of the signal) decreases. This is due to the different amplitude responses of the DAC channels or the deviations from the optimal bias voltages of the I/Q-modulator [22]. Also the signal is influenced by nonlinearities stronger. For these reasons, the use of SVM in combination with various equalizers is investigated for a data center interconnect link with 64 GBaud 64-QAM single polarization transmission with coherent detection over 100 km SSMF [123].

Therefore, the simulation setup shown in Fig. 5.5 is used. In contrast to the previous 16-QAM investigations, only the dispersion uncompensated link is considered here. In addition, the signal is examined for a B2B scenario, according to 5.5a, meaning that no transmission link is simulated. This is done by applying noise loading using an optical attenuator and an EDFA to vary the OSNR. Furthermore, the detection and demapping stage involves the *k-means* algorithm (KMA) for a competitive comparison. The KMA belongs to the class of unsupervised machine learning and is an iterative cluster algorithm, which determines the clusters via the minimum euclidean distance criterion [85]. Investigations of the NLVE determined that a training length of 2048 symbols and a memory length of NLVE[4,2,5] achieve optimal results, for which reason these settings are used in the following investigations.

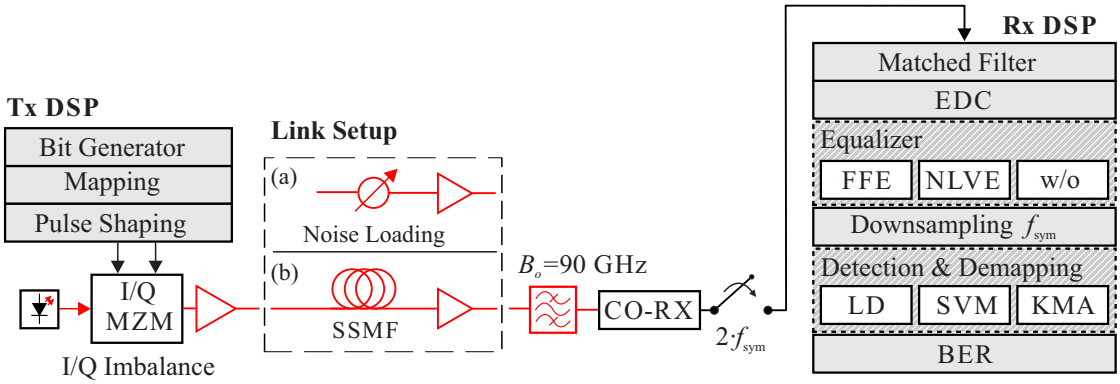


Figure 5.5: Simulation setup of 64 GBaud 64-QAM single-polarization transmission with coherent detection including two link setups [123]: (a) B2B and (b) dispersion uncompensated transmission over SSMF.

To evaluate the ability of SVM to compensate for nonlinear impairments of a 64-QAM signal in the 100 km setup, the nonlinear detection by SVM in combination with an FFE or an NLVE is compared, similarly to Chapter 5.1.1.
Therefore, the 64-QAM signal constellation is disturbed by setting the modulation depth of the MZM to $m = V_{\mathrm{pp}}/V_{\pi} = 2.2$ and an I/Q imbalance of 5° phase deviation of 90° degrees. The training of the SVM is done using 1024 symbols.

Fig. 5.6 depicts the effect of the launch power on the BER in the range of −6 to +12 dBm for 100 km dispersion uncompensated transmission. The results of a conventional equalization and detection, a FFE[1] and a NLVE[4,2,5] in combination with LD are depicted in Fig. 5.6a. In addition, the red curve illustrates the performance of a plain SVM based detection without any previously integrated equalizer. The results point out the capability of the ML based detection. The SVM achieves a minimum BER at 3 dBm launch power, which is 6 times smaller than the BER of the FFE[1] with LD. But, above 4 dBm launch power the SVM based detection is no longer able to compensate the dominating nonlinear effects resulting in a degradation of the BER. Here, the optimally configured NLVE gives the best performance. Nevertheless, the SVM exceeds the FFE[1]. The NLVE[4,2,5] increases the launch power to stay below the HD-FEC threshold by 2 dB compared to FFE[1]. If an equalizer combination composed of FFE[1] or NLVE[4,2,5] and the SVM is used, the overall system performance can be significantly improved, as shown in Figs. 5.6b and 5.6c. In particular, the combination of NLVE[4,2,5] and SVM further minimizes the BER significantly, but does not improve the nonlinear tolerance at the HD-FEC.

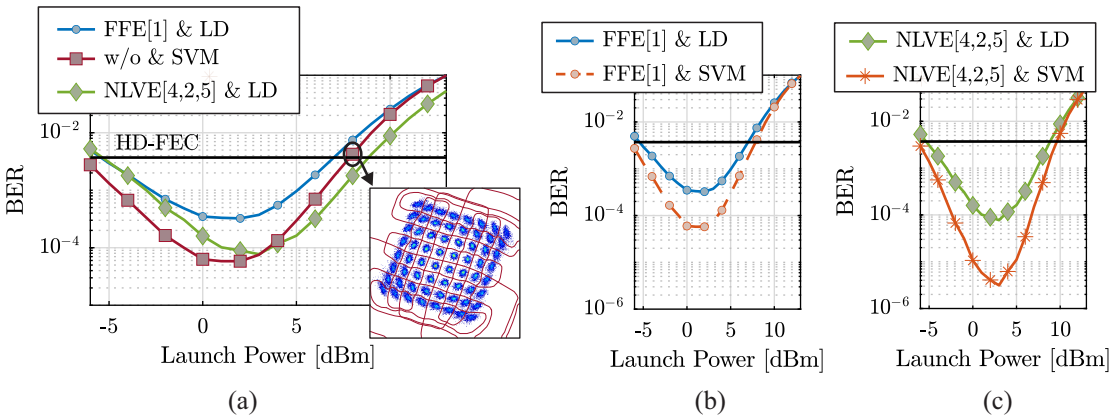


Figure 5.6: Influence of the launch power on the BER at 100 km dispersion uncompensated transmission [123]: (a) compares the equalization of the FFE[1] and the NLVE[4,2,5] combined with LD with only SVM detection. (b) compares the combination of FFE[1] and SVM and (c) the combined structure of NLVE[4,2,5] and SVM.

As mentioned before, the optimal configuration of the NLVE is given by NLVE[4,2,5]. According to Eq. 3.8, the total number of coefficients is $N_t = 82$, with the largest part arising from the third order. Therefore, in the following, the reduction of the number of delay elements of the third order is investigated. For this purpose, $N_3 = 3$ is chosen, which reduces the number of third order coefficients from 75 to 18 (75%). The results of the coefficient reduction are shown in Fig. 5.7. This shows the BER obtained as a function of the launch power for the optimal and reduced NLVE. As expected, the BER for the reduced NLVE deteriorates and is constantly below the curve of a plain SVM based detection. If the NLVE is combined with the SVM, this impairment can be compensated and even better results can be achieved than using the optimally adapted NLVE[4,2,5].

In addition to the number of NLVE coefficients, the number of training symbols is optimized for the NLVE as well as for the SVM. In this regard, Fig. 5.8 illustrates the influence of different training lengths on the BER, covering the observations of the NLVE with optimal and reduced coefficients in combination with LD and SVM based detection, as well as a plain SVM based detection. At a launch power of 5 dBm, the training length is increased from 512 to 3584 symbols in steps of 512 symbols. The results show that most improvement for all configurations occurs by increasing the number of training symbols from 512 to 1024.

The training of the NLVE is based on the amount of inter-symbol interference which is independent of the training data itself, therefore a certain number of symbols is necessary to estimate the coefficients correctly. In the case of the NLVE[4,2,5], the training length of 512 is not sufficient to correctly

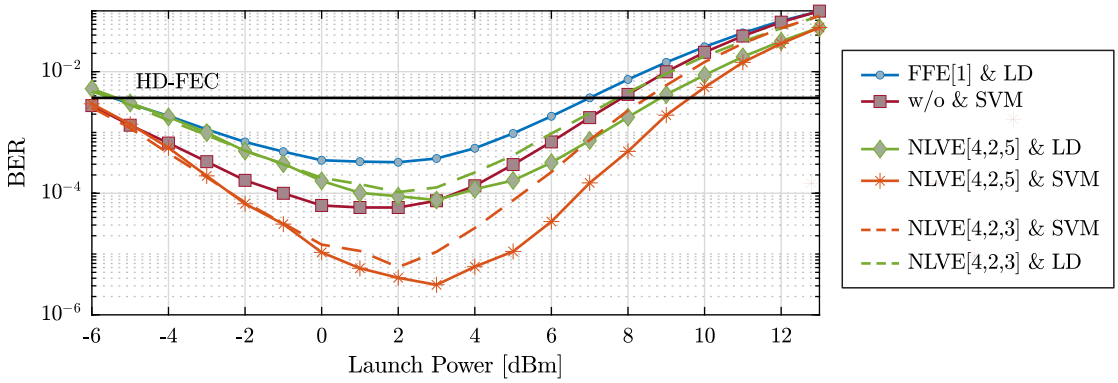


Figure 5.7: Influence of the launch power on the BER after 100 km 64 GBaud 64-QAM transmission for NLVE equalization with optimal and reduced number of coefficients in combination with SVM based detection [123].

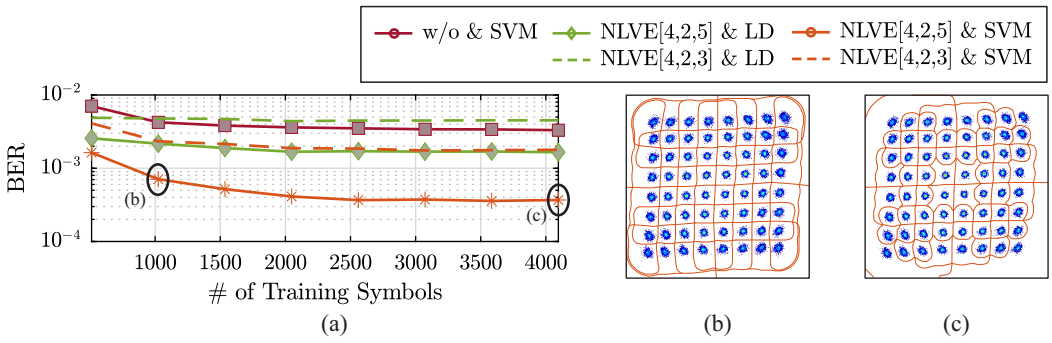


Figure 5.8: Influence of the number of training symbols on the BER at 5 dBm launch power. Constellation diagrams for b) NLVE[4,2,5] & SVM trained with 1024 symbols and c) NLVE[4,2,5] & SVM trained with 3072 symbols [123].

estimate the coefficients. If the number of coefficients is reduced, the channel estimation of the NLVE is still not sufficient to improve its performance, even if more training data is used, as seen in the case of the reduced NLVE[4,2,3]. The training of the SVM is based on the classes contained in the classification task. For this reason, it is important that, in addition to a sufficient number of training symbols, all classes are distributed uniformly in the training set. Only this ensures that the SVM can learn and capture the link properties from a small amount of training data [157]. Therefore, a training length of 512 symbols is also insufficient here. While the basic NLVE and SVM structures saturate quickly, the results of combined NLVE and SVM-based detection are fairly notable.

As previously mentioned, the I/Q-imbalances' impact is increasing with an increasing modulation format. For this reason, the behavior of the SVM against modulator nonlinearities in form of I/Q-imbalances will be emphasized at this point. Ideally, the phase shift between the I- and Q-branches of the I/Q modulator is $90°(\pi/2)$. Imperfect matching of the $\pi/2$ phase shift due to the physical imperfections of the system components can result in amplitude and phase mismatches. These I/Q mismatches can significantly disturb the signal constellation and consequently the signal transmission. The investigations are carried out according to Fig. 5.5a where the signal of a B2B transmission is disturbed at an OSNR of 28 dB at the transmitter side. For a constant amplitude mismatch of 0.125, the phase mismatch is varied between 0° and 30°.
The comparison of the different detection techniques is shown in Fig. 5.9. The performance of the extended detection techniques, two SVMs (OVA SVM, BCSVM) and the KMA, is compared with the linear detection. The training length of the SVMs amounts 1024 symbols and the number of iterations of the KMA is set to 5.

By expectation, the ML techniques are considerably more robust against the I/Q imbalances than LD. Up to a phase shift of 12°, all three ML techniques show a similar constant behavior at a BER of $3.4 \cdot 10^{-4}$. Whereas the BER of the LD deteriorates already from 2.4° phase mismatch. From 12° phase mismatch, the KMA is no longer able to form correct clusters, so that the performance deteriorates rapidly. In contrast, the SVM, the OVA SVM as well as the BCSVM, show a continuous behavior at low BER. The various multiclass classification methods of the SVM differ only slightly. The BCSVM is slightly better than the OVA method, resulting in a difference of about $1 \cdot 10^{-4}$ in BER.

The abrupt degradation of the KMA compared to the SVM could be expected. The SVM and the KMA are two completely different techniques of cluster-based detection. While the SVM belongs to the supervised learning methods, the KMA belongs to the unsupervised learning methods requiring only the knowledge about the number and central position of the clusters. The centers of the clusters form

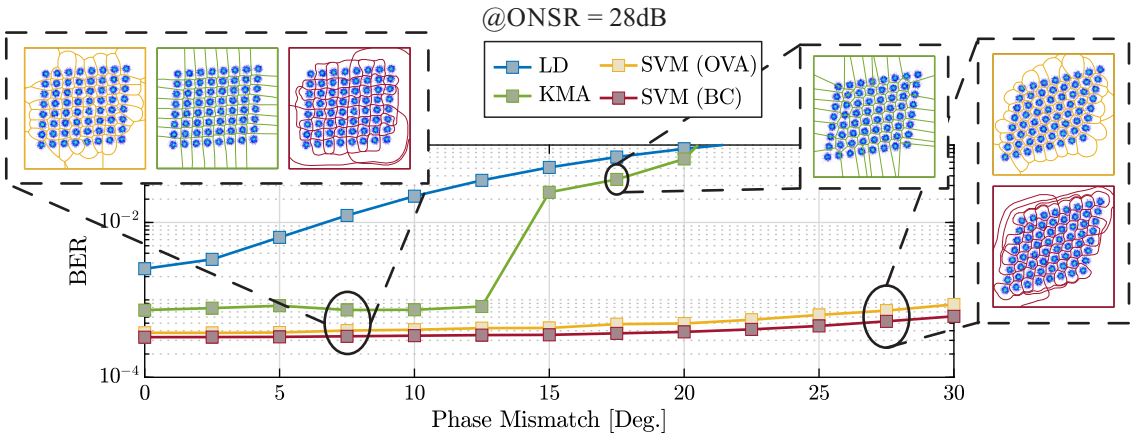


Figure 5.9: Simulation results in case of transmitter I/Q imbalances. BER vs. phase mismatch at 28 dB OSNR and constant amplitude mismatch of 0.125 [123].

the initialization of the KMA and are updated after each iteration. However, if during cluster-based detection the actual cluster is too far away from the expected cluster, the KMA is not able to separate the clusters correctly. This case can be seen in the signal constellation of the KMA at 17.5° phase mismatch. This effect can also occur in case of phase rotation caused by SPM. Moreover, the KMA is basically only a linear algorithm, while the SVM is a nonlinear classifier due to its use of kernels. Accordingly, the KMA is unsuitable for highly complex and nonlinear data distributions.

If the received 64-QAM sequence is represented on the complex plane, the different operating modes of the algorithms are visible in the signal constellation for 7.5° of phase mismatch. While the KMA, as linear algorithm, tries to compensate the I/Q imbalances by shifting the cluster centers and still applies linear decision thresholds, the SVM calculates much rounder and softer decision thresholds based on the RBF kernel. The signal constellation also represents the difference between the multiclass methods of the SVM, which is known from Fig. 4.9.

5.1.3 Compensation of the 2nd Order Nonlinearities and SSBI

This section examines the use of SVM in the presence of second-order nonlinearities that arise due to nonlinear control of the modulator. For this purpose, a double sideband transmission with direct detection for $f_{\mathrm{sym}} = 56$ GBaud symbol rate is investigated. The SSBI effect of the photodiode occurring in this configuration can be compensated here as well. The nonlinear decision thresholds of the SVM are applied to the constellation of the multidimensional modulation format, unipolar PAM with BpS ratios of $M_{\mathrm{BpS}} = 1.5$ and $M_{\mathrm{BpS}} = 2.5$. In the case of $M_{\mathrm{BpS}} = 1.5$, three bits are mapped onto two PAM-3 symbols resulting in eight constellation points. And for $M_{\mathrm{BpS}} = 2.5$, two PAM-6 symbols are

used resulting in a constellation containing 32 points. During the investigation, three bias voltages ν_{bias} of the MZM and three RRC pulse shapes are examined with regard to their influence on the signal distortion. The RRC pulse shape is changed over the roll-off factor α_{RRC}, which has an impact on the slope of the signal spectrum and consequently of the pulse shape. This may result in a different transmission behavior of the modulator.

The block diagram of the simulation setup is shown in Fig. 5.10. A lookup table is used to assign the bits of the PRBS with length $2^{17} - 1$ to the multidimensional symbols. An RRC filter is used for pulse shaping and a digital-to-analog conversion with 32-fold oversampling is applied. Using a laser operating at a wavelength of 1550 nm and an MZM, the analog signal is converted to the optical domain. The value for V_π is about 4.6 V and the linewidth of the laser is set to zero. The location of the MZM bias voltage can be seen in Fig. 5.10a and is located on the power characteristic curve introduced in Section 2.1.1. In order not to degrade the signal by further effects such as from the fiber, a B2B transmission with noise loading is applied using an optical attenuator and an EDFA to vary the OSNR. After an optical filter with $B_o = 50\,\text{GHz}$, the signal is directly detected by a photodiode. Before the demodulation, which is done by look-up tables, an FFE[4] is used to compensate possible linear effects. The demodulation is then performed either with the SVM or a conventional linear detection (LD).

The investigated bias voltages are $\nu_{\text{bias}} = -0.5V_\pi$ (1), $-0.7V_\pi$ (2) and $-0.9V_\pi$ (3) and characterize the different linearity behavior of the MZM. The driving signal $V_{\text{pp}}(t)$, the peak-to-peak input voltage of the MZM, of the input signal $s(t) \in [-1, 1]$ is normalized to the bias voltage and increased stepwise with the help of the factor d_{sig}.

$$V_{\text{pp}}(t) = d_{\text{sig}} \cdot s(t) \cdot 2(\nu_{\text{bias}} + V_\pi) \qquad \text{for} \quad \nu_{\text{bias}} < 0. \tag{5.1}$$

A value of d_{sig} means an overdrive of the MZM which is usually avoided due to possible false detections, but is used here to investigate the nonlinear behavior. A small driving voltage at a fixed bias point leads to a smaller ratio of the useful signal to the DC component. Therefore, the variation of the driving voltage can be considered as a proportional variation of the *optical signal to carrier ratio* (OSCR). For low driving voltages, the OSCR is small, which means that the noise influence is large and the effective OSNR is too low to avoid bit errors. At larger driving voltages, the OSCR also becomes larger, which, in addition to the increasing MZM nonlinearities, increases the influence of the SSBI of the photodiode.

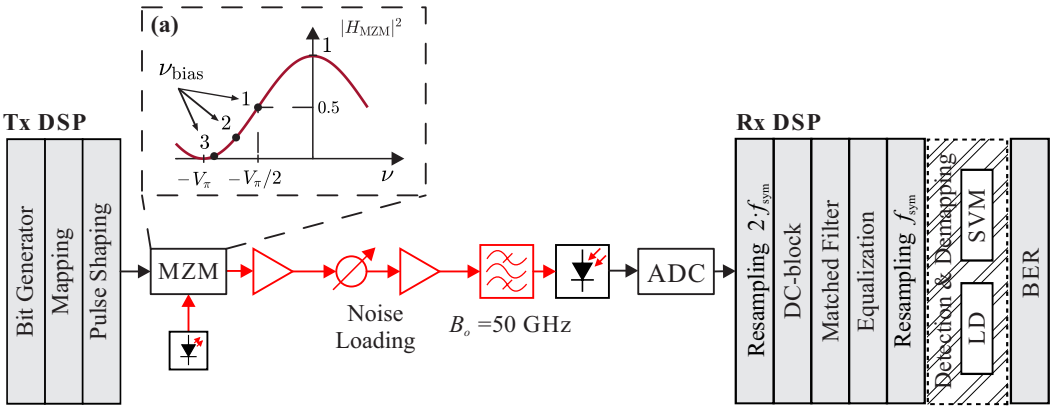


Figure 5.10: Simulation setup of $f_{\mathrm{sym}} = 56\,\mathrm{GBaud}$ double sideband transmission with direct detection.

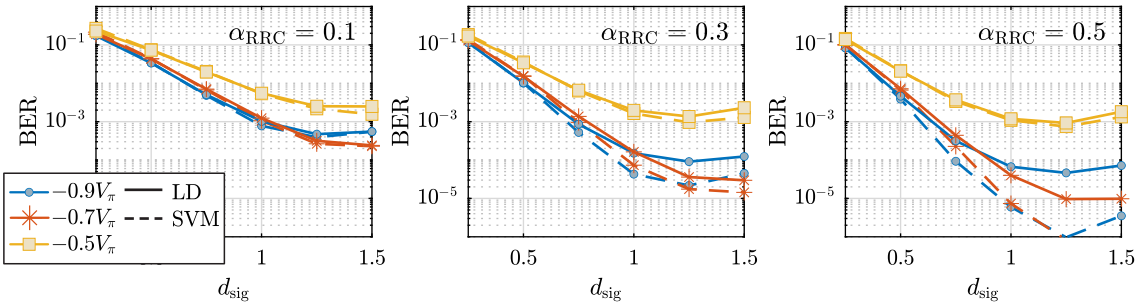


Figure 5.11: BER as a function of d_{sig}, the normalized signal voltage $V_{\mathrm{pp}}(t)$, for different RRC pulse shapes and working point voltages for $M_{\mathrm{BpS}} = 1.5$.

Fig. 5.11 shows, for an OSNR of 26 dB, the BER as a function of the scaling factor d_{sig}, for scaling the peak-to-peak driving voltage, for different RRC pulse shapes. Reducing the roll-off factor results in more oscillations in time domain (see Fig. 2.4), which causes the output signal to be increasingly distorted by ISI. This distortion can be improved by using SVM based detection. A working point of $-0.7V_\pi$ in combination with SVM based detection is the most suitable choice, since the best BER values can be obtained. For a working point of $\nu_{\mathrm{bias}} = -0.5V_\pi$, the SVM achieves no advantage over linear detection, which is due to the high linearity behavior at the quadrature point. Furthermore, for this operating point the BER is almost identical over all roll-off factors, whereas the influence of the different RRC pulse shapes is visible in the characteristics of the operating point voltages below the quadrature point ($\nu_{\mathrm{bias}} = -0.7V_\pi$ and $\nu_{\mathrm{bias}} = -0.9V_\pi$). For a roll-off factor of $\alpha_{\mathrm{RRC}} = 0.3$ and $\alpha_{\mathrm{RRC}} = 0.5$, the SVM already achieves better BER values for $d_{\mathrm{sig}} < 1$ and is thus able to compensate for the noise that dominates due to the low OSCR.

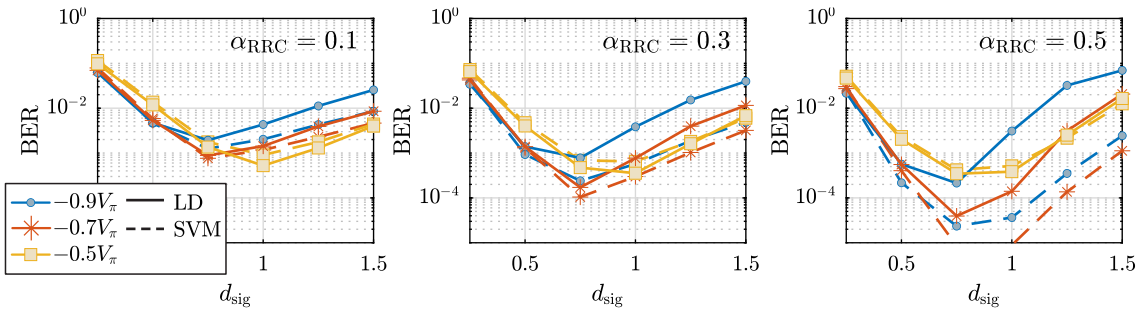


Figure 5.12: BER as a function of d_{sig}, the normalized signal voltage $V_{\mathrm{pp}}(t)$, for different RRC pulse shapes and working point voltages for $M_{\mathrm{BpS}} = 2.5$.

Following from the previous investigations, an identical analysis of the different working points and pulse shaping is performed, but here the MM format $M_{\mathrm{BpS}} = 2.5$ based on PAM-6 is used. Due to the higher number of symbols, the format has a higher bandwidth efficiency, but the sensitivity increases considerably due to the smaller Euclidean distance between the symbols. As a result, nonlinear distortions have a larger impact on the signals and the SVM could produce a more significant gain than in the first observation. Fig. 5.12 presents the results of the three different RRC pulse shapes and the MZM working points at an OSNR of 36 dB.

As for MM format $M_{\mathrm{BpS}} = 1.5$, the influence of RRC pulse shaping on the signal using a working point voltage at the quadrature point is only small. Whereas the operating points $\nu_{\mathrm{bias}} = -0.7V_\pi$ and $\nu_{\mathrm{bias}} - 0.9V_\pi$ for $d_{\mathrm{sig}} > 0.75$ benefit significantly from SVM based detection. For $d_{\mathrm{sig}} < 0.75$, the SVM achieves no advantage due to the low OSCR, since the noise influence is too large. Due to the higher noise sensitivity in this modulation format, the choice of a working point below the quadrature point in combination with LD is not suitable. However, if an SVM is used instead of the LD, this mismatch can be compensated and for $\alpha_{\mathrm{RRC}} = 0.3$ and $\alpha_{\mathrm{RRC}} = 0.5$ even a better BER can be achieved than for $\nu_{\mathrm{bias}} = -0.5V_\pi$.

Fig. 5.13 exemplary shows the two-dimensional symbol constellations at the receiver for $M_{\mathrm{BpS}} = 1.5$ (top) and $M_{\mathrm{BpS}} = 2.5$ (bottom) with the nonlinear decision thresholds of the SVM for $d_{\mathrm{sig}} = 1$. The influence of the different working points can be seen well in the symbol constellations. While the constellation points are more circular for $\nu_{\mathrm{bias}} = -0.5$, they are much flatter for $\nu_{\mathrm{bias}} = -0.9V_\pi$. Especially for $M_{\mathrm{BpS}} = 2.5$ it can be seen that the flexible thresholds of the SVM produce a lower number of false decisions due to a better fit to the distorted symbols.

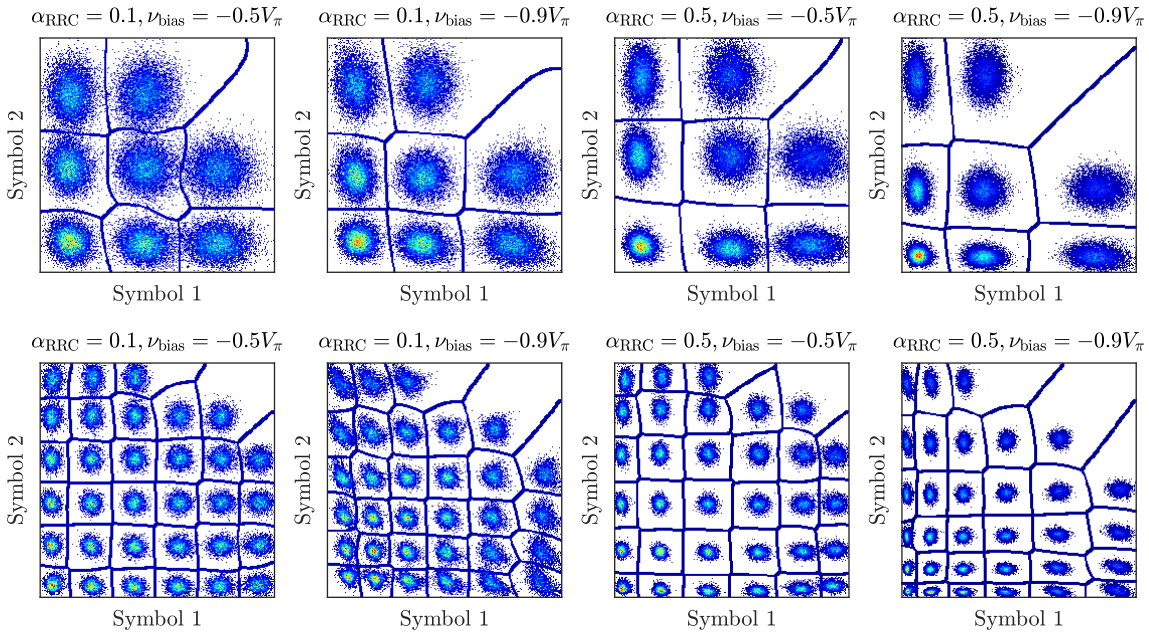


Figure 5.13: Symbol constellations at $d_{\mathrm{sig}} = 1$ for $M_{\mathrm{BpS}} = 1.5$ (top) at 26 dB OSNR and $M_{\mathrm{BpS}} = 2.5$ (bottom) at 36 dB OSNR containing the SVM nonlinear decision thresholds.

5.2 Experimental Results

Lastly, the use of SVM is thoroughly evaluated in an experimental setup for an 80 GBaud DP-16-QAM system as shown in Fig. 5.14 and results have been published in [161]. Similar to the simulation setups, a randomly generated data sequence of 322, 560 symbols is generated and mapped to the 16-QAM symbols. The signal frame consists of 192 training symbols (TS) for synchronization, 64 TS for a rough estimate of the state of polarization (SOP) and 8192 TS for channel estimation. The DAC's memory limits the data payload to 322,560 symbols, so the resulting overhead for TS is 2.63 %. A root-raised cosine filter is used for pulse shaping with a roll-off factor of $\alpha_{\mathrm{RRC}} = 0.1$, and for upsampling with a factor of 1.5 to the 120 GSample/s of the arbitrary waveform generator (AWG). To compensate for the low-pass characteristics of the AWG, cables and oscilloscope, linear precompensation filters are used in frequency domain. The applied AWG (Keysight M8194A) converts the digital signal into an analog signal with a resolution of 8 bits (ENOB $\sim$ 5.5). To meet the required voltage swing for the dual polarization I/Q modulator (PM-IQ-MOD, 21 GHz bandwidth), $V_\pi = 8.3$ V), the electrical signals are amplified and an automatic bias control (ABC) adjusts the bias voltages. The following receiver side DSP, as well as the transmitter side, consists of proven algorithms for coherent transmission, which are executed offline [162]. At first, the signal is re-sampled from the ADC rate to 2 samples per symbol (SpS), followed by a blind compensation of the chromatic dispersion. After a time synchronization

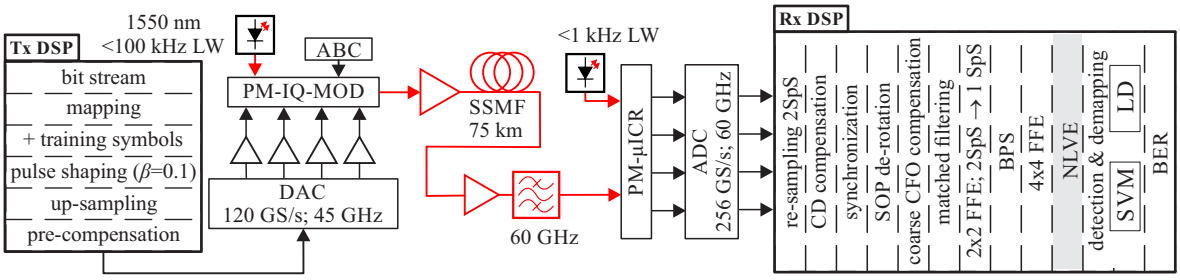


Figure 5.14: Experimental setup of 80 GBaud transmission with coherent detection and offline DSP [161].

according to Schmidl & Cox, a de-rotation of the SOP is performed by estimating the Jones matrix. Then, the carrier frequency offset (CFO) is compensated. After matched filtering, a 2x2 MIMO equalizer (FFE with 14 coefficients, channel estimation by MMSE criterion) is used to remove the residual dispersion. After further downsampling to 1 SPS, the phase noise is compensated by a blind phase search (BPS) with 64 test angles. I/Q imbalances and I/Q shifts from the transmitter are removed using a 4x4 MIMO-EQ with 54 coefficients [162]. If an NLVE is used, it follows the 4x4 MIMO-EQ, and its coefficients were set to 8, 5, and 7 for the first, second, and third orders, respectively, after optimization. Analogous to the simulation setups, the detection and decoding is either linear (LD) or nonlinear by using the SVM (BCSVM, RBF kernel). The NVLE[8,5,7] and SVM are each applied to one polarization. Finally, the BER is calculated.

Fig. 5.15 illustrates the BER as a function of input power in the range from −3 to 15 dBm after 75 km of 16-QAM transmission. For a better understanding of the system performance, the BER is examined for different equalizer stages and combinations, after compensation of the phase noise using BPS. The BER obtained for the linear detection case without second stage EQ, is presented in green. The red curve illustrates the BER of the 4x4 MIMO-EQ in the second stage and the yellow curve illustrates the BER of the 4x4 MIMO-EQ in the second stage compensated by an NLVE[8,5,7] in the third stage, each obtained with LD. The yellow dashed line, which overlays the yellow solid line, represents the SVM based detection after the NLVE.

Unlike the simulation results, the system performance, without the use of the 4x4 MIMO-EQ, is very poor. This is due to the fact that various impairments were not taken into account during the simulation. These include the laser phase noise of the transmitter and receiver, I/Q skews of the transmitter and receiver and the CFO of the receiver. In addition, only one polarization was simulated, which means that effects such as PMD, rotation of the SOP and x/y-skews were also not taken into account. Furthermore, a complete CD compensation was simulated, which means that also residual dispersion was not taken in consideration for simulation. Due to the listed, not simulated, impairments,

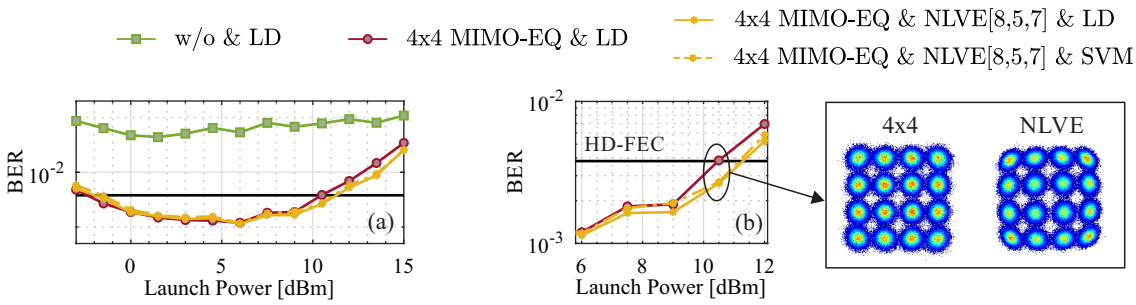


Figure 5.15: BER as a function of the launch power after 75 km 16-QAM transmission. (b) represents a more detailed view of (a) at the HD-FEC including the constellation diagrams of the 4x4 MIMO-EQ and the NLVE[8,5,7] [161].

it is not possible for the SVM to achieve the HD-FEC threshold without a preceding 4x4 MIMO-EQ. The BER of the 4x4 MIMO-EQ is shown in red and reaches a minimum of $1.19 \cdot 10^{-3}$ at 6 dBm launch power. To compensate for nonlinearities in the receiver-side DSP as well, an NLVE is used as a third stage equalizer (yellow). As shown in Fig. 5.15b, this improves the nonlinear tolerance at the HD-FEC threshold by 1 dB. However, with an additional SVM based detection no improvement can be observed afterwards. The constellations prove that after the preceding equalizers, which linearize the signal, linear thresholds are sufficient.

Similar to the simulative investigations from Chapter 5.1.2, the use of SVM based detection in the presence of I/Q imbalances is investigated and presented in Figs. 5.16 and 5.17. The phase offset varies between $0° - 75°$ at 5 dBm launch power (OSNR = 38 dB). Again, the use of the 4x4 MIMO-EQ is mandatory to reach the HD-FEC. With an NLVE added as a third stage equalizer, the tolerance to I/Q imbalances remains unchanged. This was to be expected, since the NLVE is a complex equalizer in which the I- and Q-components are not considered separately, as indicated by Eq. 3.6. The use of SVM based detection with increasing phase offset is shown in Fig. 5.17. It can be seen that already for a small data set of 4096 training symbols, the SVM increases the tolerance to I/Q imbalances by 7° at the HD-FEC threshold. And, any further increase of the training data does not result in any further improvement of the BER.

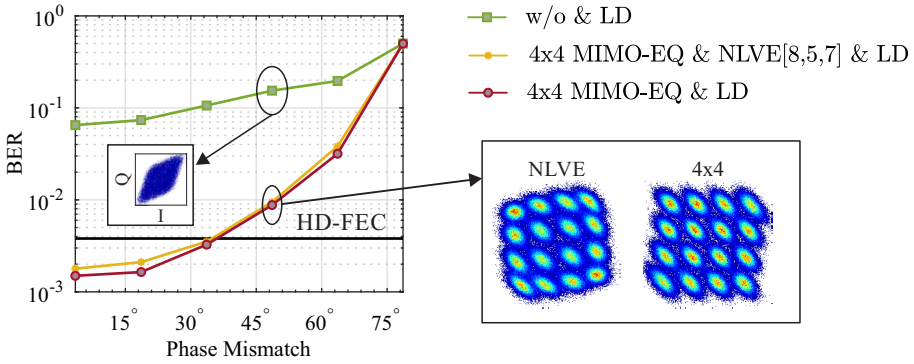


Figure 5.16: BER as a function of the phase mismatch of the transmitter [161].

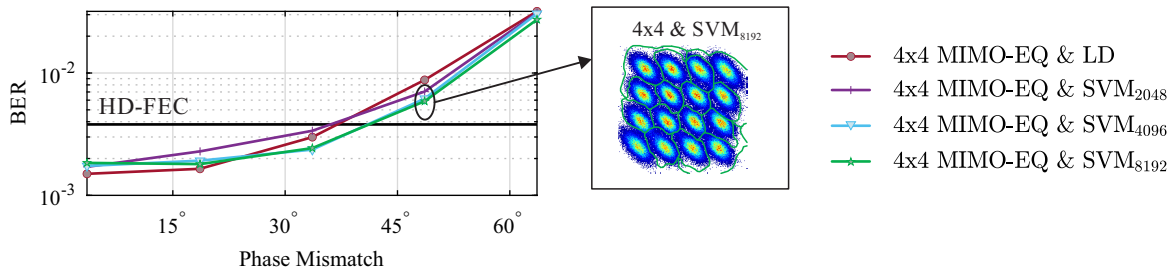


Figure 5.17: BER as a function of the phase mismatch of the transmitter using SVM based detection [161].

5.3 Summary

In this chapter, the use of SVM based detection was investigated simulatively and experimentally. First, an optimization of the SVM was performed by competitively comparing the multiclass classification algorithms, the use of different kernels and the optimizer algorithms. It was shown that the BC classification in combination with the RBF kernel and the SMO algorithm achieved the best results. Using the mentioned settings, investigations for the use of the SVM within different links, the DUL, DML and ZDL, were performed. The transmission via the ZDL showed the strongest nonlinear distortions in the signal constellation, which is why the SVM achieved the best result there by increasing the nonlinear tolerance by 1.75 dB. But even in a more common transmission using a DUL, the SVM is able to increase the nonlinear tolerance compared to a linear equalizer, as well for the 16-QAM transmission over 1200 km, so also for the higher level modulation of 64-QAM over 100 km. In combination with a non-linear Volterra equalizer, the SVM improves the overall system performance. Also an I/Q imbalance, caused by an amplitude or phase offset at the modulator, can be easily compensated by the SVM. In the experimental investigations, it has been shown that SVM-based detection is not possible without the use of the 4x4 MIMO equalizer due to the prevailing skews and polarization impairments. However, when using a 4x4 MIMO equalizer, the signal is linearized - that no improvement in BER can be achieved with nonlinear SVM detection.

Chapter 6

Reinforcement Learning in Optical Transmission Systems

To meet the growing demand for bandwidth, Elastic Optical Networking (EON) uses reconfigurable optical system components that enable efficient use of spectral resources in an optical fiber. Flexible transponders and ROADM architectures are used for this purpose. A key element of such an EON is a bandwidth-variable transponder (BVT), which allows the adaptation of transmission parameters such as modulation format, symbol rate or forward error correction overhead to the current link conditions. This functional flexibility in adapting transmission parameters for efficient use of available spectral resources is accompanied by a significant increase in configuration complexity. The high number of reconfigurable parameters poses the greatest challenge in selecting the optimal system configuration. In addition to the high number of reconfigurable parameters, the selection of parameters has become a major challenge due to physical layer impairments (PLI), which include amplified spontaneous emission (ASE) noise and nonlinear interference (NLI) [163].

Moreover, modeling the transmission system has been a major challenge in network planning so far. Only with sophisticated analytical PLI models, such as the GN model presented in Section 2.3.2, it has become possible to capture the complex system behavior in reasonable computational time. Based on this, a better use of the available and limited network resources by optimizing the transmit power, constellation and DWDM channel allocation has been shown in [164, 165].

Also in this context, the techniques of ML are a promising approach to enable fast network self-configuration and decision making. To this end, a comprehensive comparison between ML approaches

and analytical models based on the GN model was made in [166].

In order to maximize the transmission capacity considering the NLI, heuristic optimization algorithms seem to be a good approach. Therefore, the use of a variety of heuristic approaches for the optimization of self-learning autonomous EONs have been investigated [167, 168, 45, 51, 52]. These are capable of selecting near-optimal parameters, but require a long computation time for densely packed DWDM signals as the search space scales with the number of channels. In principle, these algorithms could be used directly to optimize a real system with BVTs, but due to the many thousands of iterations required to find an optimum, it is not practical to apply these algorithms in a reasonable time frame. In addition, the solutions would have to be recalculated each time the link conditions (e.g. range or number of channels) change. The application of ML techniques to increase the capacity of an SDN-EON in a field trial is also demonstrated in [169] by estimating link performance and transmission quality for non-established light paths, but only based on linear assumptions.

To solve the optimal resource allocation problem, reinforcement learning is currently gaining increased research attention [54, 170, 171, 172]. The main focus lies on the dynamic RMSA problem, where path requests arrive "on the fly" and need to be served immediately. This optimization aims to improve the long-term blocking probability of requests. However, in addition to minimizing the blocking probability, it is also important to ensure that the capacity of the transmission channel is used as efficiently as possible. For this reason, the following chapter focuses on the selection of parameters for multiple transmission channels to maximize the total capacity of a point-to-point link while maintaining a threshold. This problem is similar to the $\mathcal{NP}$-complete Knapsack or bin packing problem. Especially due to the nonlinear interactions, such as intra- and inter-channel effects, a trade-off between the optical power of the signal and the strength of the nonlinear interactions must always be considered, which is why this challenging problem cannot be solved with analytical models in an acceptable time frame.

6.1 RL based System Optimization

The main objective in this optimization problem is to increase the overall throughput of the transmission link. Increasing the bit rate contributes to higher capacity, but reduces the GOSNR, which can be improved again by increasing the power. However, this can also lead to undesirable nonlinear side effects and impairments, as discussed in Section 2.1.2. Thus, there is an optimal point that takes this trade-off into account. To find this using RL, the transmission system represents the environment with which the agent communicates and is therefore presented here.

Table 6.1: Fiber Parameters at $\lambda = 1550\,\text{nm}$.

Parameter			Value	
			LEAF	TWC
Length	L_S	[km]	80	50
Attenuation	α	$[^{\text{dB}}/_{\text{km}}]$	0.22	0.23
Dispersion	D	$[^{\text{ps}}/_{\text{nm}\cdot\text{km}}]$	4.2	2.8
Slope	S	$[^{\text{ps}}/_{\text{nm}^2\cdot\text{km}}]$	0.086	0.068
Nonlinear coefficient	γ	$[^{1}/_{\text{W}\cdot\text{km}}]$	1.3	2.1

6.1.1 Environment Setup

To keep the optimization problem small, a point-to-point link is observed. The block diagram of the transmission system is shown in Fig. 6.1 and defines the environment. It consists of a set of N_c coherent channels $\mathbf{c} = [c_1, ..., c_{N_c}]$, which are transmitted over a DWDM link. A fixed channel spacing is chosen to reduce the parameter space. Each channel c_i is equipped with a BVT that determines the required bandwidth according to the requested bit rate and the used modulation format. Each channel is defined by a set of channel specific parameters such as bit rate, launch power p_i, central frequency f_i and modulation format. These initial parameters are used to simulate the transmission. There are two types of NZDSFs investigated: the LEAF and the TWC, whose fiber parameters can be found in Tab. 6.1.
The transmission impairments are calculated using the Gaussian noise model presented in Section 2.3.2. It takes the initial parameters as input and returns a GOSNR. With the GOSNR, the GOSNR margin m_i for each channel can be calculated using

$$m_i = \text{GOSNR}_i - \text{required OSNR}_i \tag{6.1}$$

and is used in this work as an indicator for the quality of the transmission. In all subsequent investigations, channels that have a margin above the defined margin threshold of 1.7 dB are considered valid, so that their bit rate contributes to the total bit rate. The bit rate varies from 200 GBit/s to 600 GBit/s, the highest possible throughput of the transponders used in the simulation, as well as in the laboratory. For better comparability, the state of each episode is initialized with 200 GBit/s and -3 dBm for all channels.

To find the optimal system configuration, the agent has the goal of maximizing the total bit rate while remaining below the specified margin threshold. In order to enable the agent to learn a strategy for

this, the environment properties still have to be defined in terms of states, actions and rewards. The exact definition is given separately in the following sections, since different definitions were examined in the course of this work.

6.2 Simulation Results

This section provides the simulation results of different agents. Starting with an agent that can adjust all channels over the entire C-band in one action up to the reduction to an adjustable sliding window. In each case, the PLIs are estimated using the GN model and evaluated using the GOSNR margin m_i for each channel according to Eq. 6.1. Results have been published in [173, 174, 175]

6.2.1 Joined Agent

The approach of the so-called joined agent makes use of the fact that the agent is able to adjust the power and bit rate of different channels of the transmission system by one action [173, 176]. The simulation setup is depicted in Fig. 6.1. The agent has an overview of the current entire system state at any time step. Which means that all parameters of the system, relevant for the optimization, are stored in the state. The state is constant and contains the information about the number of spans and the bit rate, power and margins for all channels. Consequently, the state vector s_t at time step t has a size of $(1 + 3N_c)$.

In response to the state, the agent performs an action. The agent is able to set a certain bit rate and power for selected channels. The selection of the channels is realized via a mask, more specifically the channel mask. Using the flag *done*, the agent signals that no further parameters need to be set. In the following, an action is represented by the vector a_t and has a size of $(3 + N_c)$.

After each action, the agent receives a reward from the environment. Scaling the rewards is challenging in this specific application of RL. In particular, because the solution is unknown, which is atypical for

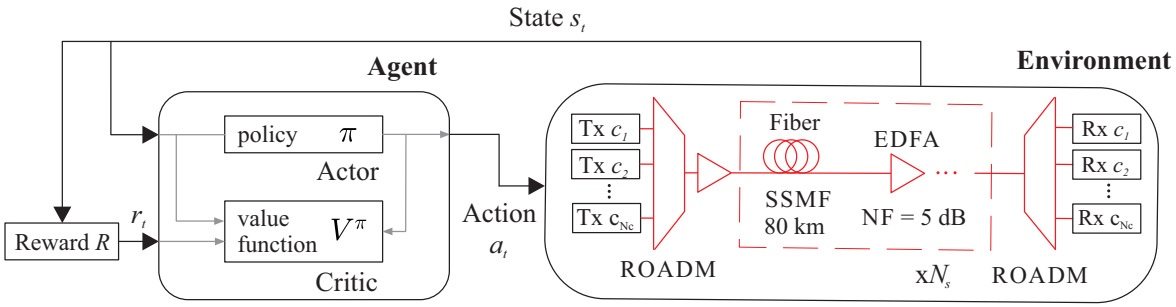


Figure 6.1: Simulation setup of the joined agent.

RL problems. For this reason, it is difficult to scale the reward/punishment ratio properly so that the agent learns as efficiently as possible. Therefore, three different rewards systems are first presented and examined. A consistent feature of all rewards systems is that the agent receives a reward after each episode T. This is finished when either the agent takes the action *done* or when the maximum number of steps $T \leq (2N_c + 1)$ is reached.

The implementation is based on the A2C algorithm from the stable baseline library [177]. Here, each FNN consists of two layers with 64 and 32 neurons each. At the beginning of each episode, the state of the environment is initialized with 200 Gbit/s and -3 dBm for each channel and the number of spans is evenly distributed in the range of 6 to 23. Each agent is trained with a fixed number of episodes, which is set to $N_E = 1000$ using the RMSProp algorithm. After training, results are produced for a set of 1000 test solutions for a 51 channel system with 100 GHz channel spacing. For each solution, the agent starts with the initialized state as in training, but with a fixed number of spans N_S and tries to maximize the bit rate. If the result is a channel below the threshold margin m_{Th}, the solution is considered unfeasible. The associated bit rate, which contributes to the total capacity, is 0 Gbit/s in this case. At the end of this investigation, the solutions of the trained and optimized agent are compared to the solutions of a GA.

Reward Optimization

As stated before, the agent's goal is to maximize the total bit rate while staying below the margin threshold. Therefore, it is obvious to incorporate these two quantities into the rewards system. One possibility is to reward the agent only for the channels, whose margins are above the threshold after optimization. Thus, the agent receives the following normalized reward at the end of the episode:

$$R = \frac{1}{N_c} \sum_{1}^{N_c} r_i \tag{6.2}$$

with

$$r_i = \begin{cases} \frac{b_i - b_{\min}}{b_{\max} - b_{\min}}, & \text{if } m_i \geq m_{\mathrm{Th}} \\ 0 & \text{otherwise} \end{cases} \tag{6.3}$$

with the minimum bit rate $b_{\min}$ and the maximum bit rate $b_{\max}$. Using this strategy, the agent receives a reward $R > 0$ as soon as at least one channel is above the margin threshold and is therefore further referred to as *reward without margin threshold* (w/o m_{Th}).

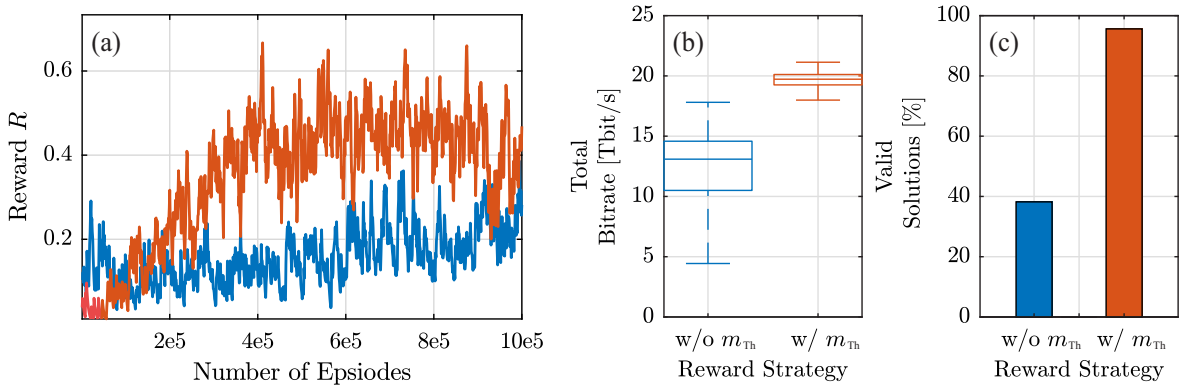


Figure 6.2: Comparison of the different reward strategies (w/o m_{Th}) (blue) and (w/ m_{Th}) (red). (a) represents the average return, (b) shows the total bit rate after 1000 solutions for 16 spans and (c) represents the corresponding valid runs [175].

Another option is to reward the agent only, if all channels have a margin above the threshold, so that the calculation of the reward is done according to

$$R = \begin{cases} \frac{1}{N_c} \sum_1^{N_c} r_i, & \text{if } \min_{\forall m \in \mathbf{m}} m \geq m_{\mathrm{Th}} \\ 0 & \text{otherwise} \end{cases} \tag{6.4}$$

using the reward per channel

$$r_i = \frac{b_i - b_{\mathrm{min}}}{b_{\mathrm{max}} - b_{\mathrm{min}}}. \tag{6.5}$$

This strategy adopts the approach that the agent first produces only valid solutions and only thereafter tries to maximize the bit rate and is continuously referred to as *reward with margin threshold* (w/ m_{Th}).

The comparison between the rewards systems from Eq. 6.3 (blue, w/o m_{Th}) and Eq. 6.4 (red, w/ m_{Th}) is shown in Fig. 6.2. Here, Fig. 6.2a shows the respective learning profiles of the agents during training over 10^5 episodes. Therein, it can be observed that the agent w/ m_{Th} achieves significantly higher rewards. The better performance during training can also be seen in the results of Fig. 6.2b, which shows the distribution of the total bit rate over 1000 solutions. The agent w/o m_{Th} achieves a maximum bit rate of 17.8 Tbit/s and on average 14.5 Tbit/s, while the agent w/ m_{Th} achieves a maximum bit rate of 21.13 Tbit/s with a mean of 19.7 Tbit/s. Fig. 6.2c shows the percentage of valid runs, which means that the agents ended up with a feasible solution in which the margin of all channels is above the margin threshold. Again, this shows the better performance of the agent w/ m_{Th}. This agent produces 95.6 % of valid solutions, while the agent w/o m_{Th} produces only 38.2 % of feasible

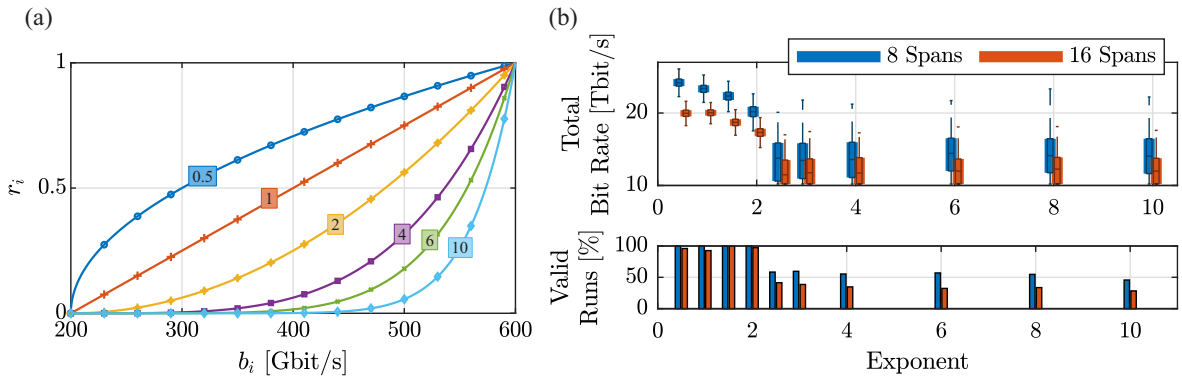


Figure 6.3: Reward exponent optimization for the capacity optimization of a 51-channel WDM-system. (a) Reward r_i per channel for different exponents (acc. to Eq. 6.6) in the range of 0.5 to 10 for bit rates between 200 Gbit/s and 600 Gbit/s and (b) total bit rate for 8 (blue) and 16 (red) spans after training with different reward exponents in the range of 0.5 to 10 (top) and the corresponding number of valid runs (bottom) [175].

solutions.

Another strategy independent of the margin threshold is a nonlinear scaling of the reward. This is realized by an exponent x which is applied to the reward per channel. Eqs. 6.3 and 6.4 are respectively converted to

$$r_i = \left(\frac{b_i - b_{\text{min}}}{b_{\text{max}} - b_{\text{min}}} \right)^x . \tag{6.6}$$

The Fig. 6.3a initially presents the influence of the exponent on the reward over the bit rate for the exponents from 0.5 to 10. It can be seen that for the exponent $x = 1$ the agent receives a reward that is linear to the bit rate. For $x < 1$ differences in the lower bit rate range influence the reward more significantly, and for $x > 1$ the significant range is in the last 20% of the maximum bit rate, i.e. between 500 Gbit/s and 600 Gbit/s. For the proposed problem, the deviation between the rewards and not the value itself is relevant. For example, for a bit rate of 550 Gbit/s at $x = 10$ a small increase in bit rate is rewarded more than at $x = 0.5$. This principle should induce the agent to choose even higher bit rates even at already high bit rates, because exactly this fine tuning is the difficulty in the system optimization problem to be solved.

Using the agent (w/ m_{Th}), the influence of the reward exponent on the total bit rate is shown in Fig. 6.3b, exemplified for 8 and 16 spans. As mentioned earlier, every fourth span was trained between 6 and 23 spans. Accordingly, 8 and 16 spans do not belong to the state space of the agent, so that a generalization capability is already evident here. Higher bit rates are achieved after transmission over

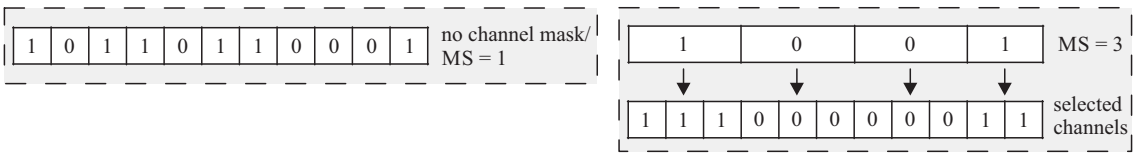


Figure 6.4: Introduced channel mask for $N_c = 11$ in the case that no channel mask is selected, which corresponds to $\psi = 1$ (left) and for $\psi = 3$ (right) including the corresponding assignment to the channels [175].

8 spans than after 16 spans, which was to be expected because of the shorter distance. For exponents $x < 2$, the total bit rate decreases slightly with an increasing exponent. For exponents $x > 2$, the total bit rate dropped significantly, which correlates with the number of valid runs. This also drops in the range from nearly 100% to 50%, so that only half of the agent's output contains solutions whose margin is greater than m_{Th} for each channel. Contrary to expectations, even for 8 spans the total bit rate decreases significantly. Although the variation in reward is among the strongest near the maximum at an exponent of 10 (see Fig. 6.3a). The agent does not reach the region near the maximum possible bit rate because the difference in reward is small at low bit rates, suggesting that an exponent of 1 is ideal because the slope of the reward is always the same and thus the variation per bit rate remains constant.

Action Space Optimization

By using the channel mask $\xi \in [-1, 1]$ of length N_c, the agent specifies the channels to be modified. If $\xi_i \geq 0$, the channel c_i is affected and gets a new value for the power and bit rate. Consequently, the agent can use the action to change the bit rate and power for the masked subset of channels. This results in an action space that is defined by the vector a_t of size $(3 + N_c)$. If the number of channels increases, the number of possible masks also increases and is given by

$$N_{\text{channel mask}} = 2^{N_c}. \tag{6.7}$$

For example, in the case of 51 channels, there are 2^{51} different channel masks that the agent has to explore and learn. One consideration to reduce the parameter space and thus the training time is to introduce a quantized channel mask. For this purpose, the parameter ψ is used, which specifies how many channels are addressed with the mask index and then adjusted identically.
Fig. 6.4 shows exemplarily for eleven channels the principle of the introduced mask size for $\psi = 1$ and $\psi = 3$. For $\psi = 1$, each channel is addressed individually, which effectively corresponds to no channel mask. For $\psi = 3$ three channels are selected in a bundle. Here it should be noted that with eleven

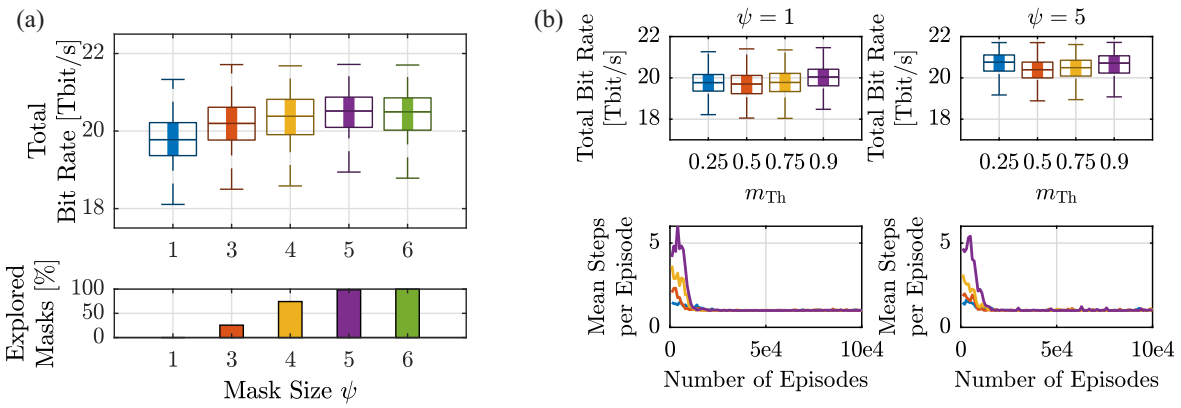


Figure 6.5: Simulation results of the use of the channel mask ψ after 16 spans of 50 GBaud transmission. (a) illustrates the total bit rate for mask sizes 1 to 6 and the corresponding rate of explored masks during training. And (b) shows a comparison of the margin thresholds for mask sizes $\psi = 1$ and $\psi = 5$, where, the total bit rate is evaluated over the margin thresholds as well as the related mean number of steps per episode [175].

channels and the mask size $\psi = 3$, a remainder of two channels remains, so that the last mask is of size 2. With the use of masking, the action space is significantly reduced and has a new size of $(3 + \lceil \frac{1}{\psi} \rceil N_c)$ depending on the mask size.

A comparison of the mask size from 1 to 6 is shown in Fig. 6.5a. There, the total bit rate (top) and the corresponding rate of explored masks (bottom) are compared after 50 GBaud transmission over 16 spans. As expected, the number of explored masks increases with the increase of mask size. For $\psi = 1$ the rate is about 0 %, while for $\psi = 3$ about 25 % and for $\psi = 5$ even 100 % of all masks were seen. But the interesting fact is that the bit rate is only slightly affected by this. The maximum bit rate for $\psi = 1$ is 21.5 Tbit/s and is similar to the maximum bit rates set by agents for the mask sizes $\psi = 3$ to $\psi = 6$, which is 21.7 Tbit/s. Nevertheless, the average total bit rate increases with the mask size. For example, at $\psi = 1$ the mean bit rate is 19.7 Tbit/s and at $\psi = 5$ it is 20.5 Tbit/s. This suggests that with longer mask size, the agent reaches a higher total bit rate more quickly. Not shown here are the valid runs of the different mask sizes. These all show similar behavior, as they are all between 88.3% and 96.8%.

In addition to the channel masking, the threshold value of the done decision can also be varied. This can be used to influence the behavior of the agent during training. Namely, a higher threshold leads to longer episodes for the originally random strategy. Accordingly, it is an indicator for the reliability of the agent. In the previous investigations m_{Th} was set to 0.5. In Fig. 6.5b the total bit rate after 16 spans for the mask-size $\psi = 1$ (left) and $\psi = 5$ (right) is shown for the threshold values of 0.25 to 9, respectively. The bottom row represents the average number of steps per episode during training

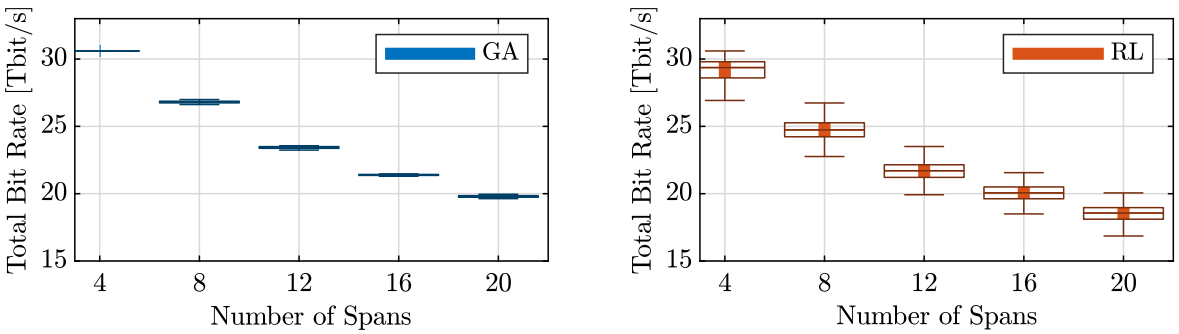


Figure 6.6: Overall bit rate achieved by the agent in 1000 solutions compared to 53 solutions found with a GA for span counts {4, 8, 12, 16, 20} [175].

averaged over 1000 episodes. Analogous to the previous investigations of the mask size, a higher total bit rate is obtained with $\psi = 5$, which is also independent of the done threshold value set. The effect of the threshold can only be seen at the beginning of the training by the average number of steps per episode. This is because the agent requires more steps as the threshold increases with initial random start strategies.

Comparison and Benchmarking

Followed by RL algorithm optimization, the agent is compared with the solutions generated by optimizing the system parameters with a GA trained for 500 generations and a population size of 64. For this purpose, 1000 solutions were generated in each case. In Fig. 6.6 it is shown that the total bit rate achieved by the RL agent definitely competes with the results of the GA and can even exceeds them. It is noticeable that the variance of the total bit rate of the RL solutions is significantly higher. This could be reduced after optimizing the hyperparameters of the FNN and increasing the number of training episodes N_E. A very important fact not shown here is that the computation time differs significantly between the two approaches. The computation time for the GA depends on the length of the link and ranges from a few minutes for short links to several hours for longer links, whereas the RL agent takes on average 1s, after 12h of initial training on our state-of-the-art PC.

While the GA has to be retrained for each link length, the RL offers generalization properties. This means that the agent is able to produce solutions for states which the agent has never seen before, e.g. a new link length. This principle is demonstrated in more detail in Fig. 6.7. There, the total bit rate is compared over span counts from 6 to 23 for four different agents. The agents are trained for every, every 4^{th}, every 8^{th}, and for 10 & 20 spans. It is noticeable that there is no significant difference or decrease in the solutions obtained. Even the agent trained only for 10 & 20 spans achieves quite good

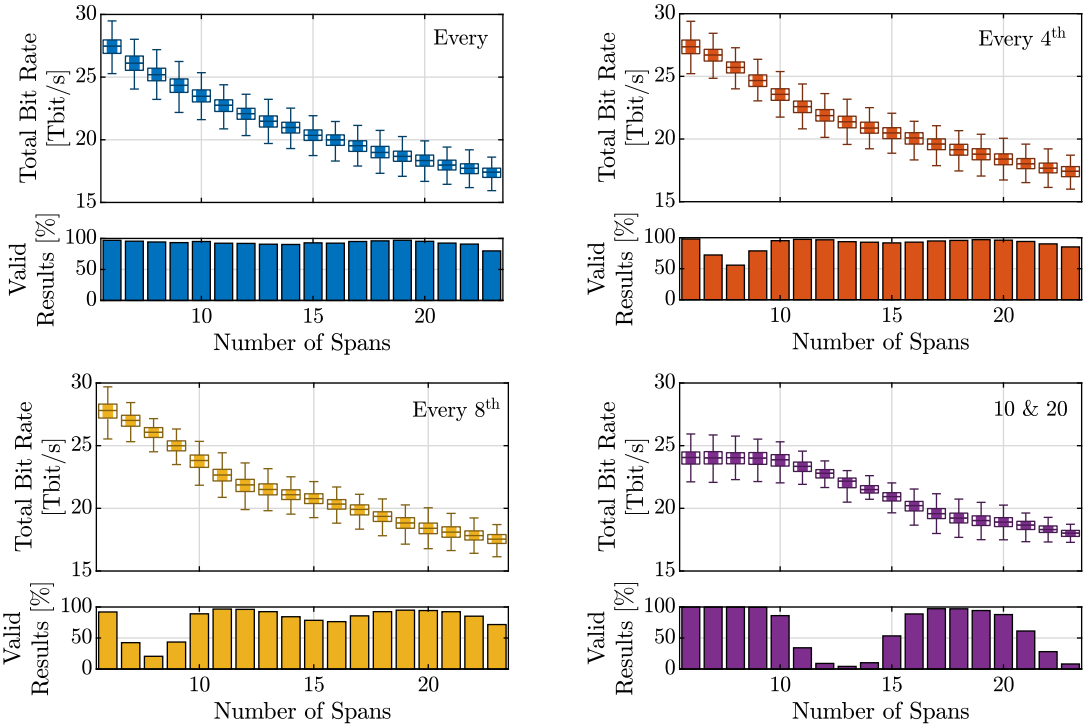


Figure 6.7: Overall bit rate and percentage of valid results achieved by the RL agent which is trained for every span, every 4th span, every 8th span, 10 & 20 spans with $\psi = 5$, $m_{\mathrm{Th}} = 0.5$ and $x = 1$ [175].

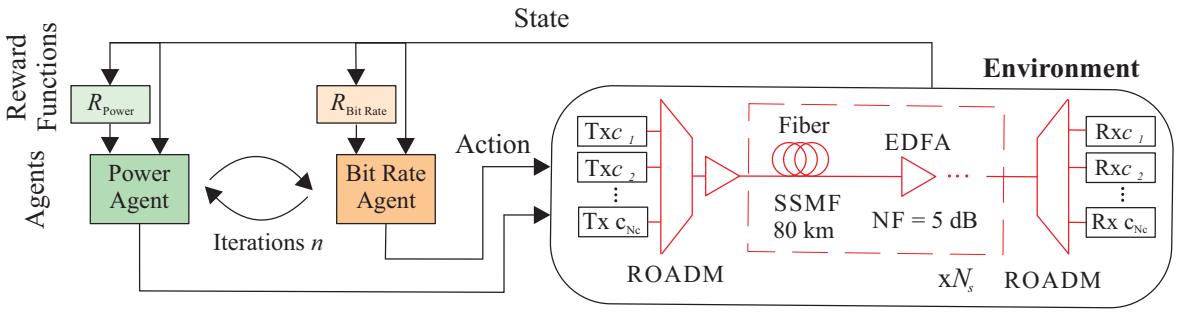


Figure 6.8: Simulation setup.

results. But it is apparent that the number of invalid solutions varies between the agents. The less spans are seen in the training, the more risky solutions are chosen by the agent during optimization. Therefore, the number of valid solutions decreases significantly, if there are too many unknown states between the training states.

6.2.2 Two independent Agents

One way to reduce the parameter space for keeping the task of the RL agent as simple as possible is to divide the optimization problem into two sub problems. Results have been published in [174]. For this purpose, two independent agents are trained. Specifically, one agent is responsible for optimizing the power (*Power Agent*), at a given bit rate configuration, with the goal of increasing the margin of each channel. And the other agent (*Bit Rate Agent*) aims, for a given power configuration, to optimize the bit rate and maximize the total capacity of the transmission. Further, the overall intent is to find the optimal BVT parameters so that the total capacity of a DWDM point-to-point link is maximized.

Fig. 6.8 depicts the simulation setup of the transmission system, consisting of a variable number of identical spans. The same environment as in Section 6.2.1 is used here.
The state of each agent contains the margin, bit rate and power of all channels, resulting in a state vector of the size $3N_c$. At the beginning of each episode, the state of the bit rate agent is initialized with the minimum bit rate of 200 Gbit/s and a power for each channel from a discrete random uniform distribution $\mathcal{U}\{-3, 3\}$ dBm. The bit rate is increased by 50 Gbit/s so that $n = 13$ discrete values are possible in each episode. This allows the power agent to find a bit rate policy π_b that generalizes to previously unknown power configurations, since there are 13^{N_c} different permutations of power. Similarly, the state of the power agent is initialized with the minimum power of -3 dBm and the bit rate drawn from a discrete random uniform distribution $\mathcal{U}\{200, 600\}$ Gbit/s for each channel. The power is increased by 0.5 dB, resulting in $n = 9$ discrete values.

The following action at time t is to select a channel i for which either the power p_i or the bit rate b_i is increased. Since each episode is initialized with the respective minimum values, decreasing the bit rate or power is excluded as an action in order to reduce the action space.
After each action, the agent receives a reward. With the goal of increasing the total bit rate of the system for the given power configuration, the reward function of the bit rate agent can be defined as

$$R_{\text{bit rate}} = \left(10 \cdot \sum_{i=1}^{N_c} r_i\right)^2 \tag{6.8}$$

with

$$r_i = \begin{cases} \frac{b_i - 200}{400}, & \text{if } m_i \geq m_{\text{Th}} \\ 0 & \text{otherwise.} \end{cases} \tag{6.9}$$

With the goal of maximizing the margins of all channels, the power agent receives its reward according to the function

$$R_{\text{Power}} = \sum_{i=1}^{N_c} m_i. \tag{6.10}$$

As soon as the sum of the margins is smaller than the one before, the episode is finished.
As can be seen in Eqs. 6.9 and 6.10, the expected return of one agent does not depend on the choice of action of the other agent, so it is possible to train the agents independently. For optimization, however, the agents must act together, by alternating bit rate and performance optimization, as shown in Alg. 1. After initializing the power $\boldsymbol{p} = \{p_1, p_2, ..., p_{N_c}\}$ and the bit rate $\boldsymbol{b} = \{b_1, b_2, ..., b_{N_c}\}$ for all channels, a counter is used to count the number of iterations in which no improvement was achieved.
The implementation is based on the A2C algorithm from the stable baselines library [177]. Each FNN contains 64 and 32 neurons in the hidden layers. An action (i.e., selecting a channel index) is performed by drawing from the Gaussian random distribution parameterized by the policy FNN for the current state using a discrete kernel. Each agent is trained once over $N_E = 10^7$ episodes using the RMSProp algorithm [178]. During training and in optimization, the number of channels is $N_c = 11$, and the number of spans is set to $N_S = 10$. After training, a set of 100 solutions is produced according to Alg. 1 with $n_{\max} = 10$. Similar to Section 6.2.1, once $m_i < m_{\text{Th}}$, the channel is considered invalid and the associated bit rate contributing to the total capacity is then 0 Gbit/s. For comparability and better estimation of the generalization capability of the RL, the GA is again used as reference algorithm. For this purpose, 100 solutions are generated over 1000 generations for each number of spans.

Algorithm 1: Power and bit rate optimization [174].

Input: π_p, π_b, $n_{\max}$, $\boldsymbol{b}$, $\boldsymbol{p}$
Output: Bit rate $\boldsymbol{b}_{\text{out}}$ and power $\boldsymbol{p}_{\text{out}}$
$n = 0$, $b_{\max} = \sum_{b \in \boldsymbol{b}} b$, $\boldsymbol{b}_{\text{out}} = \boldsymbol{b}$, $\boldsymbol{p}_{\text{out}} = \boldsymbol{p}$
while $n < n_{max}$ **do**
  Optimize power $\boldsymbol{p}$ for $\boldsymbol{b}$ with π_p for $1 < i < N_c$
  Optimize bit rate $\boldsymbol{b}$ for $\boldsymbol{p}$ with π_b for $1 < i < N_c$
  $b_{\text{cur}} = \sum_{i=1}^{N_c} b_i$
  if $b_{cur} > b_{max}$ **then**
    $b_{\max} = b_{\text{cur}}$
    $\boldsymbol{b}_{\text{out}} = \boldsymbol{b}$
    $\boldsymbol{p}_{\text{out}} = \boldsymbol{p}$
    $n = 0$
  else
    $n = n + 1$
  end
end

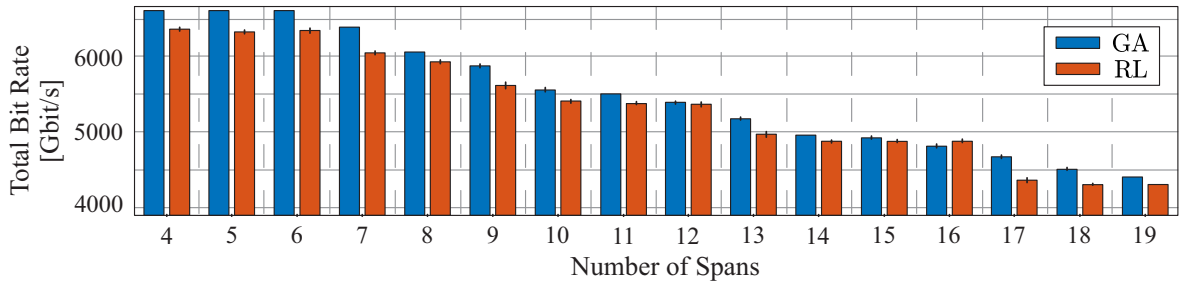


Figure 6.9: Overall bit rate achieved by the GA (blue) and the RL agents, trained for $N_s = 10$ spans, (red) in 100 solutions [174].

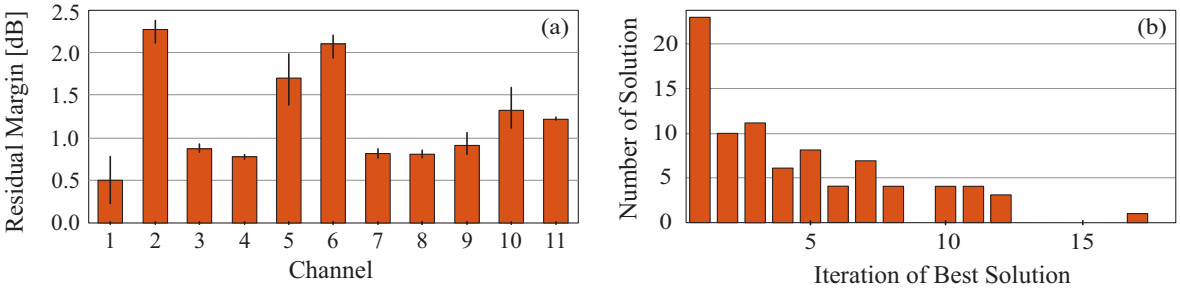


Figure 6.10: (a) Residual margins of each channel for a fixed span count of $N_s = 10$ and (b) the histogram of the number of solutions over the number of iterations for which the highest bit rate was reached during the generation of 100 solutions according to Alg. 1 [174].

A comparison between the RL and GA for different span counts ranging from 4 and 19 spans is depicted in Fig. 6.9. It can be seen that although the RL agents were only trained for a fixed span count of $N_S = 10$, they achieve a similar total bit rate as the GA. For $N_S = 16$, the RL policies even perform better than the GA with a total bit rate of 4750 Gbit/s, which again highlights the generalization capability of RL. Furthermore, a time advantage can be noticed again. While the GA needs about 1-2h per span, the RL policies find the solution, after initial long training, already within 60-180s.

The residual margins, above the threshold $m_{\mathrm{Th}} = 1.7\,\mathrm{dB}$, of the respective channels are shown in Fig. 6.10a for a fixed span count of $N_S = 10$ spans. It can be seen that at least 0.5 dB residual margin is still available on each channel. For channels 2, 4 and 5, the residual margin is even above 1.5 dB. In this way, the difference in the total capacity compared to the GA can be explained. It can be assumed that the agent could have set an even higher bit rate for these channels. The reason for this behavior might be due to a local minimum of the gradients during training and might not occur if different policies are trained with the same hyper parameters. Fig. 6.10b shows the distribution of iterations where the highest total capacity, for $N_S = 10$ spans, was reached during optimization. This suggests that $n_{\max} = 10$ could be reduced, since maximum total capacity is often found during the first few iterations.

6.2.3 Sliding Window Agent

This section examines a further method for implementing RL for maximizing the total capacity. Results have been published in [176]. This approach enables to maximize the overall throughput of a point-to-point link, independent of the number of channels. Analogous to the approach in Section 6.2.2, the parameters of only one channel, the channel of interest (COI) are modified during the action. With the difference that the agent does not choose the channel itself, instead it is predetermined by

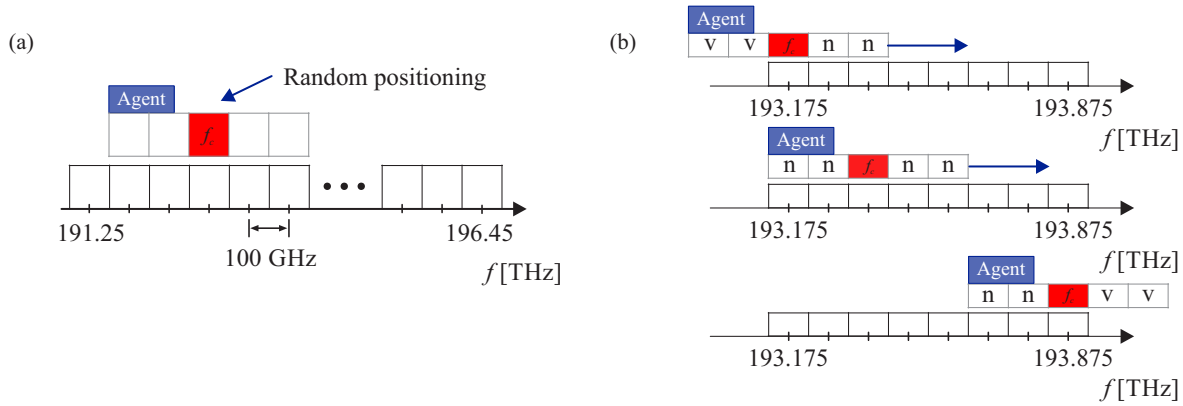


Figure 6.11: Window-based agent during training over the entire C-band in (a) and during optimization for a fixed user defined number of channels and location in the C-band in (b) [176].

the center frequency f_t given in the state. If this is chosen during training from a discrete random uniform distribution $\mathcal{U}\{191.25, 196.45\}$ THz drawn in steps of 100 GHz, it is possible to train the agent over the entire C-band. Furthermore, the number of visible channels is limited, which leads to a reduction of the parameter space. The number of visible channels is parameterized with the variable n, which specifies the number of neighbors of the COI. If $n = 2$, there are two neighbors each to the left and right of f_t, resulting in a window size of 5. The principle of the window-based agent during training over the entire C-band is illustrated in Fig. 6.11a. At time step t, the agent observes the state $s_t = \{N_S, f_t, p_t, b_t, \boldsymbol{m}\}$ with $\boldsymbol{m} = [m_{t-n}, ..., m_{t+n}]$. The agent observes the current number of spans, the position of the channel to be adjusted with the associated bit rate b_t and power p_t, and the margins of all channels located in the window. The symbol rate remains constant at 69 GBaud by using different hybrid modulation formats in the time domain from QPSK to 64-QAM.

The action following the state is limited to 5 actions selected from $\{\text{stay}, b_{\text{inc}}, b_{\text{dec}}, p_{\text{inc}}, p_{\text{dec}}\}$. If a policy selects the first action, the state remains unchanged. Analogous to Section 6.2.2, the bit rate (per channel) can be set between 200 and 600 Gbit/s in steps of ±50 Gbit/s, and the power can be adjusted between −3 and 3 dBm in steps of ±0.5 dB. For each episode, the initial state is drawn from a discrete uniform random distribution. After each action, the agent receives a reward, which is defined as

$$R(t) = \begin{cases} \sum_1^{N_c} b_i, & \text{if } t = t_{\max} \\ 0 & \text{otherwise,} \end{cases} \tag{6.11}$$

so $R(t) = 0$ for all steps in the episode, except for the last one, where the reward is the sum of all bit rates that exceed the margin threshold.
The implementation is based on the A2C algorithm of the stable baselines library [177]. The policy and value function are parameterized again by FNNs, which consists of two layers with 64 and 32 neurons and uses the hyperbolic tangents as activation function. The learning rate during training with the RMSProp algorithm [178] ranges from $1 \cdot 10^{-3}$ to $5 \cdot 10^{-5}$. Two agents were trained over 100,000 episodes each with $i = 3$ steps per episode and $n = 2$ neighbors. While one was trained on every span, the other was trained on every fourth span, i.e. the subset $N_s \in \{12, 16, 20, 24\}$.

After training, the agent optimizes the parameters of a 51 channel system over different span counts with the goal of increasing the overall capacity. For this purpose, the learned policy is repeatedly applied to each channel using the sliding window, shown in Fig. 6.11b, beginning from the leftmost and ending at the rightmost channel. The state s is changed according to the current COI and its neighbors. After a fixed number of steps $i \in N^+$, or when the *stay* action is selected, the next channel is optimized. This procedure is performed over several iterations to take into account the nonlinear influence of the neighboring channels. It is important to clarify that no left or right neighbors are available for optimizing the channel at the edge. Therefore, the optimization is performed by using virtual channels (v) with a bit rate of 200 Gbit/s, a power of −3 dBm and 2 dB margin. For training, this procedure is not necessary, since the center frequency of the left edge $f_{c,\min}$ and the right edge $f_{c,\max}$ is defined by

$$f_{c,\min} = f_{\min} + (n+1) \cdot \Delta f \tag{6.12}$$

$$f_{c,\max} = f_{\max} - (n+1) \cdot \Delta f \tag{6.13}$$

using the channel spacing $\Delta f = 100\,\text{GHz}$ and the number of neighbors n. Consequently, the edge channel of the channels to be optimized is identical to the edge channel of the window. In other words, the agent never optimizes the outermost channels of the C-band during training. For optimization, it is conceivable to randomly adjust the position of the window once over all channels, from right to left, or alternately to either side of the center channel.

The results of the optimization after 12 iterations are shown in Fig. 6.12. It can be seen that both RL policies achieve a similar overall capacity. Thereby, even the agent which was trained only for every fourth span outperforms the one which was trained for every span. This again demonstrates the interpolation capabilities of RL. Even in comparison with the reference algorithm (the GA) the total bit rates are all in a similar range. Nevertheless, the capacities of the links parameterized by

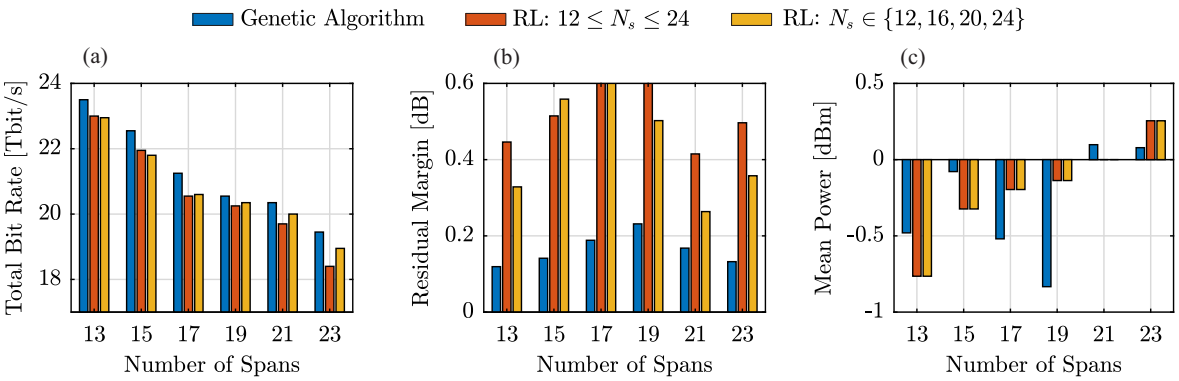


Figure 6.12: Simulation results after 12 iterations of RL optimization and compared with the measurements of the simulation-based GA results. (a) Achieved total bit rate, (b) residual margin and (c) mean launch power.

GA outperform the RL policies by 3.1% on average (12.7 Gbit/s/channel), but require 5×104 PLI estimates to optimize a single span. Here we consider 1000 generations with a population size of 50, which is about an hour of computation time depending on the number of spans. The associated excess margin averaged over all channels (see Fig 6.12b) indicates that the RL policies are able to minimize the residual margins, i.e. the margin above the threshold of $m_{\mathrm{Th}} = 1.7\,\mathrm{dB}$. Nevertheless, the GA residual margins are significantly lower, even for a very similar total bit rate, such as for 19 spans. An explanation can be found in the averaged adjusted power over all channels (see Fig. 6.12c). Here, the set values of the RL agents differ significantly from the GA.

Comparison with the Joined Agent Approach

A direct comparison between the joined agent approach and the sliding window-based agent is shown in Fig. 6.13. For the visualization, the mean value was calculated from the 1000 solutions of the joined agent. It is obvious that the sliding window-based agent achieves higher bit rates than the joined agent after optimizing a 51-channel system, independent of whether they were trained (for every or every fourth span). In addition to the higher overall bit rate, the comparison also reveals that the window-based agent requires significantly less training time. Furthermore, it is an immense advantage that any user defined number of channels can be optimized after a single training with the sliding window agent.

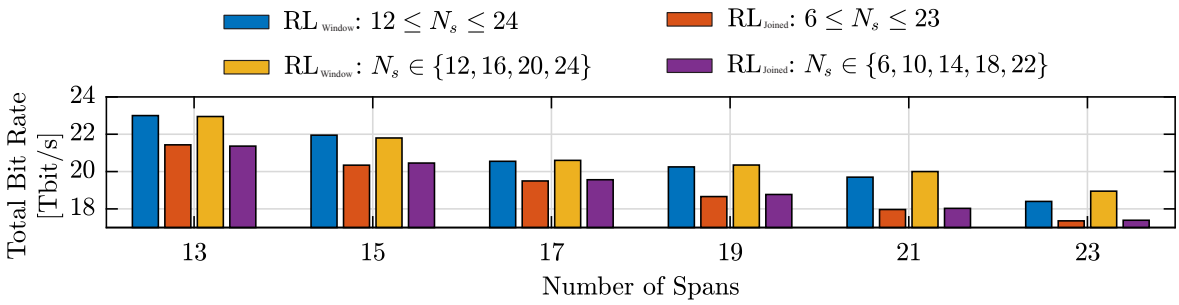


Figure 6.13: Total comparison of joined agent approach and the sliding window based agent for an optimization of an 51-channel system, evaluated for 1000 solutions.

6.3 Experimental Results

Using the sliding window approach, the throughput of a real transmission system is maximized [175]. For this purpose, the agent is simulatively trained with idealized parameters and applied in the laboratory to 8 DWDM channels over 9 spans of True Wave Classic (NZDSF) (see Tab. 6.1). The experimental setup is shown in Fig. 6.14. It contains 4 Infinera CHM2T transponders (in an Infinera Groove G30 housing), each with 2 channels. After each 50 km TWC NZDSF segment, the span loss was also varied with a VOA to emulate longer fiber lengths.

Following the previously presented approach from Section 6.2.3, the state and reward of the agent were slightly changed for these measurements. At each time step $t \leq t_{\max}$, the agent observes the state $s_t = [N_s, f_c, \boldsymbol{p}, \boldsymbol{b}, \boldsymbol{m}]$, which includes the number of spans N_s and a center frequency f_c, indicating the position of the COI. Also, the bit rates $\boldsymbol{b}$ and powers $\boldsymbol{p}$ of all channels located in the window were included in the state. With the associated (GSNR) margins $\boldsymbol{m}$, the state vector is of size $(2 + 3 \cdot n)$. The action space corresponds to the one in Section 6.2.3, where an action at time t can be drawn

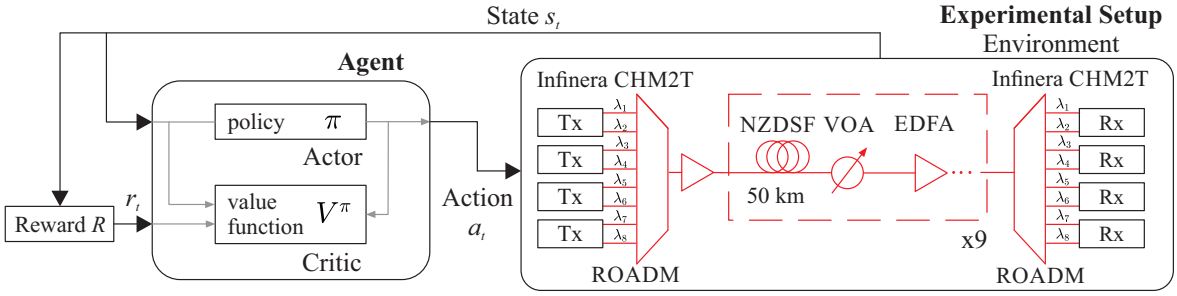


Figure 6.14: Experimental setup made up of the RL agent (left) and the transmission link as environment (right) which includes a variable optical attenuator (VOA) that is used to vary the span loss to investigate different emulated span lengths.

from the actions $a_t \in \{\text{stay}, b_{\text{inc}}, b_{\text{dec}}, p_{\text{inc}}, p_{\text{dec}}\}$. Compared to Section 6.2.3, the reward is marginally different, defined below as

$$R(t) = \begin{cases} \sum_1^{N_c} b_{i,t=t_{\max}} - b_{i,t=0}, & \text{if } t = t_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{6.14}$$

and is the difference of the sums of all bit rates b_i for the channels exceeding the predefined threshold for the margin between the first and the last step, and describes the change in link capacity achieved by the choice of actions during this episode.

The implementation is based on the Proximial Policy Optimization (PPO) algorithm [131]. The FNNs parameterizing the policy and value function consist of two layers with 16 and 8 nodes, respectively. During training, the learning rate of the Adam algorithm is reduced from 10^{-3} to 5^{-5}. To further support generalizability, the RL agent is provided with additional e.g. path-related parameters, that are uniformly varied in a certain range. Specifically, the span length is varied between 60 and 70 km, the nonlinear coefficient between 1.8 and 2.4 $^1/_{\text{W·km}}$, and the noise figure of the EDFAs between 4 and 5 dB. In addition, the agent is trained on a transponder simulation model with idealized characteristics. The agent is trained for 9 spans and over the entire C-band, i.e. in the frequency range from 191.25 to 196.45 THz with a frequency spacing of 100 GHz. The training is done over 100,000 episodes with $i = 3$ actions per episode and takes about 39h on a modern PC.

During the optimization of the real system, the agent starts with the leftmost channel. From there, the agent adjusts all channels with 3 steps per episode using the sliding window shown in Fig. 6.11. This process is repeated for a fixed number of iterations. The initial values for the bit rate are 200 Gbit/s and for the power −6 dBm, which is lower than for the SSMF studied so far and due to the NZDSF.

In addition to the usual comparison with the GA, a power sweep is also performed for comparison. This has been calculated for each span loss by exhaustive search with the restriction that the starting power and bit rate are identical for all channels. The optimization for the aforementioned benchmark algorithms is performed for a fixed noise figure of 5 dB, a nonlinear coefficient of 1.99 $^1/_{\text{W·km}}$ and a span loss of 12, 15, 18, and 21 dB. Unlike the RL agent, a realistic transponder model was used to train the benchmark algorithms, taking into account, for example, implementation losses. The runtime of the PS was about 5 min. In comparison the runtime of the GA was $\sim$ 12 h for the optimization of eleven channels over a maximum of 24 spans. The immense time requirement explains why the reference algorithms and RL training are based on simulations.

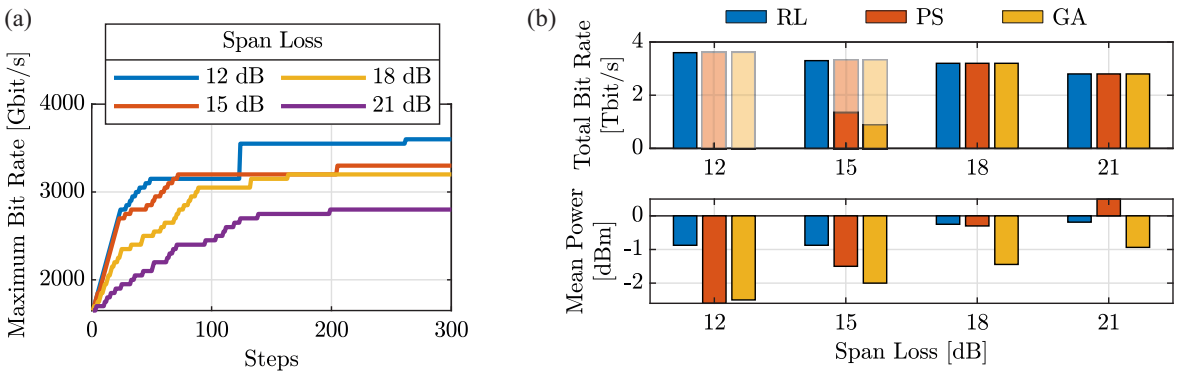


Figure 6.15: Experimental results of the RL window-based optimization for 8 WDM channels over 9 spans of TWC (NZDSF). (a) depicts the RL progress during optimization towards the maximum bit rate for different span losses and (b) illustrates the achieved total bit rate and mean launch power compared with the simulation-based measurements of the GA/PS. These benchmark algorithms achieve the same total bit rate as the RL agent only after adjusting the simulation model (transparent) [176].

The comparison of the total bit rate of the system over each step of the optimization of the BVT parameters by the RL agent for different chip losses is shown in Fig. 6.15a. It can be seen that the agent finds a configuration with a high bit rate after only a few steps, but then needs many more steps for further increases.

The total bit rate achieved for different span losses and the various algorithms is shown in Fig. 6.15b. For the benchmark algorithms, the simulated channel power and bit rate were set in the transmission testbed. Only channels with a measured (GOSNR) margin greater than the margin threshold $m_{\mathrm{Th}} = 1.7\,\mathrm{dB}$ contribute to the total bit rate. Contrarily, the RL agent communicates directly with the system and sets the maximum achievable bit rate based on its learned policy. It can be observed that at higher span losses of 18 and 21 dB, all algorithms find a valid solution for all channels and achieve the same aggregated bit rate. At low span loss, only the RL agent is able to provide valid results, while the GA and the PS lead to settings from which margins below the threshold are measured. The reason for this is that in the PS and GA simulations the NF of the EDFAs was set to 5 dB, which is too optimistic for the low span loss values. To verify this statement, a further parameter set of GA and PS was generated with an NF of 6 dB. After re-optimization with the adjusted simulation parameters, the benchmark algorithms achieve the same total bit rates as the RL agent, shown by transparent bars in Fig. 6.15b. Since the RL agent acts on the real system, it can account for the increased NF. This clearly shows the advantage of RL for optical transmission system optimization. While GA and PS require the solution to be recalculated for each specific system setting (e.g. voltage loss, channel count), RL can be applied to different link conditions and channel counts

after a one-time training. Although the time required to run the benchmark algorithms is less, the time invested in RL training is still beneficial.

6.4 Summary

This chapter investigated RL learning for capacity maximization of a DWDM system simulatively and experimentally. For this purpose, three approaches of the MDP problem formulation were presented: the joined agent, two independent agents and the sliding window agent. The joined agent is able to continuously adjust the bit rates and powers of the BVTs using TDHQ. The agent is able to select the parameters to maximize the total bit rate of a 51-channel point-to-point link within seconds. The formulation of the reward function and the use of a channel mask have a significant impact on the achieved transmission rates. With the introduction of channel masking, the computational effort in training could be reduced. In order to reduce the parameter space, the power and bit rate of the two independent agents were set discretely. This approach is also able to increase the total link capacity and achieve similar results as the GA. While the joined agent and the two independent agents have to be trained for the number of channels they are supposed to optimize, the sliding window agent is trained over the whole C-band. After training, the agent can optimize any number of channels specified by the user. Compared to the joined agent, the window-based agent achieves even higher overall bit rates. Even in the laboratory, the window-based agent is able to adapt to the unseen conditions during training and increase the total bit rate, while the precomputed solutions of the heuristic optimization algorithms, as PS/GA, cannot meet the margin requirement. Only the RL algorithm is able to compensate the discrepancy between simulation and reality here.

Chapter 7

Conclusions and Outlook

In this work, machine learning methods for use in optical transmission systems were investigated. Due to fiber nonlinearity, signal constellations occur at the receiver, where the clusters are smeared and rotated, when high launch powers are used. For this reason, SVM based detection was investigated to show that the tolerance of the system to these nonlinear effects can be improved. Compared to linear decision thresholds, SVM has been shown to be a suitable model for signal detection in modern optical transmission systems. The nonlinear distortions of a 64-QAM signal after 100 km SSMF can be correctly detected using the nonlinear decision thresholds of the SVM as well as the modulator nonlinearities for a 56 GBaud dual band transmission with DD. However, in a long haul simulation over 1200 km, the nonlinear impairments caused by the Kerr effect have been shown to also interact with the dispersion. It follows that the constellation points at the receiver appear as linearly Gaussian distributed points, so linear detection is sufficient. It has further been shown that the SVM in combination with an NLVE improves the BER. Therefore, the number of NLVE coefficients can be reduced to reach a BER level, if the detection is done with an SVM. This reduces the computational effort considerably. The experimental results of an 80 GBaud 16-QAM transmission over 75 km SSMF have shown that SVM-based detection is not possible without the use of a 4x4-LMS-MIMO equalizer due to the prevailing skews and polarization impairments. However, when using a 4x4-LMS-MIMO EQ, the signal is linearized to the point that no improvement in BER can be achieved with nonlinear SVM detection. Due to noise in the receiver in the digital domain, increasing the input power also does not result in lower BERs, even if the NLVE could compensate nonlinearities.

From the results of SVM investigations, it can be concluded that it is an efficient ML algorithm for increasing the data rate only under certain conditions. Theoretically, the signal affected by memoryless nonlinearities, such as nonlinear phase noise, I/Q modulator penalties and nonlinearity of the drive

electronics, no longer appears circular at the receiver, which is why the idea of nonlinear symbol detection appears to be useful. Practically, however, an interplay of chromatic dispersion and Kerr-effect takes place during transmission, resulting in Gaussian-distributed constellation points, which is why the nonlinear decision thresholds of SVM do not improve the BER. Additionally, in high bit rate systems, other effects such as I/Q skews dominate, leading to ISI. Due to the correlation of the data, which is also present due to the remaining dispersion, the SVM has difficulty in separating the data. For this reason, SVM based detection would be more suitable for lower symbol rate systems. Furthermore, the effects of WDM effects should still be investigated further.

The results of the RL algorithm investigations demonstrated that with suitable problem formulation, RL can be used in optical transmission technology for system parameter optimization with the goal of maximizing the system capacity. First, RL requires initial training with a large number of configurations. After that, RL is able to generalize to different conditions without requiring further training for each possible link condition, while the benchmark algorithms such as the GA/GS need to be re-run for each scenario. To keep the initial training low, it is advisable to keep the parameter space as small as possible. This can be done, for example, with a window-based agent, which observes only a part of the whole transmission system. It has been shown that a neighbor number of two is sufficient to optimize the total capacity of 51 channels, while accounting for interchannel effects.

So far, only the performance objective of maximizing throughput has been considered. However, in a real network design it is necessary to understand the interdependencies of multiple performance goals and consider them in the design, for which RL is quite a suitable tool. Future research could initially incorporate additional parameters into the training to create an agent that is even more adaptable to other systems. An agent could be trained for a sufficient number of different spans and could then also be able to interpolate between these spans and optimize for arbitrary distances. In addition, the spectrum assignment could be made flexible. After that, the point-to-point link could be extended into a network topology and other goals could be included in the MDP problem formulation, such as minimizing latency or maximizing resilience.

Nevertheless, it should be noted that the problem is not a typical RL problem, which makes the MDP problem formulation challenging. In particular, reward allocation is a major challenge to achieve the desired goals. This has a great impact on the achievable transmission rates.

List of Figures

List of Tables

Acronyms

ASE amplified spontaneous emission

ANN Artificial neural network

AC actor critic

A2C advantage actor critic

A3C asynchronous advantage actor critic

ABC automatic bias control

AWG arbitrary waveform generator

BC binary coding

BCSVM binary coding support vector machine

B2B back-to-back

BER bit error ratio

BPS blind phase search

CFO carrier frequency offset

CW continuous wave

CGN closed-form Gaussian noise

CUT channel under test

CWDM coarse wavelenght division multiplexing

CD chromatic dispersion

COI channel of interest

DBP digital back propagation

DD direct detection

DWDM dense wavelength division multiplexing

DUL dispersion unmanged link

DML dispersion managed link

DP dual polarisation

DC direct current

EDFA Erbium doped fiber amplifier

EAM electroabsorption modulator

EGN extended Gaussian noise

EDC electronic dispersion compensation

EM expectation maximization

EQ equalizer

FT Fourier transformation

FFT fast Fourier transformation

FDM frequency division multiplexing

FFE feed forward equalizer

FWM four wave mixing

GN Gaussian noise

GVD group velocity dispersion

GNRF Gaussian noise reference formula

GOSNR generalized optical signal to noise ratio

GA genetic algorithm

HD high definition

ISDA iterative single data algorithm

IFT inverse Fourier transformation

IFFT inverse fast Fourier transformation

ISI inter-symbol interference

IM intensity modulation

IoT Internet of things

KL Kullback Leibler

LO local oscillator

LD linear detection

LSTM long short-term memory

ML machine learning

MF matched filter

MZM Mach-Zehnder modulator

MSE mean squared error

MMSE minimum mean squared error

NLSE nonlinear Schrödinger equation

NLI nonlinear interference

NLP nonlinear phase noise

NLVE nonlinear Volterra equalizer

OADM optical add drop multiplexer

OSNR optical signal to noise ratio

OVO one vs. one

OVA one vs. all

OSCR optical signal to carrier ratio

PAM pulse amplitude modulation

PSD power spectral density

PMD polarization mode dispersion

PBS polarization beam splitter

PS power sweep

PPO procimal policy optimization

QAM quadratur amplitude modulation

QP quadratic programming

RC raised cosine

RRC root raised cosine

ROADM reconfigurable optical add drop multiplexer

RF radio frequency

RL reinforcement learning

Rx receiver

RBF radial basis function

SBVT sliceable bandwidth variable transponder

SGD stochastic gradient descent

SNR signal-to-noise ratio

SSFM split step Fourier method

SSMF standard single-mode fiber

SOP state of polarization

SPM self-phase modulation

SVM support vector machine

SOA semiconductor optical amplifier

SRS stimulated Raman scattering

SBS stimulated Brillouin scattering

SpS samples per symbol

SMO sequential minimal optimization

SD standard definition

SSBI signal-signal beat interference

StpS steps per span

SpS samples per symbol

TDHQ time domain hybrid QAM

TDM time division multiplexing

Tx transmitter

TIA transimpedance amplifier

TS training symbols

TRPO trust region policy optimization

UHD ultra high-definition

VOA variable optical attenuator

WSS wavelength selective switch

WDM wavelength division multiplexing

XPM cross-phase modulation

ZDL zero dispersion link

Bibliography

[1] Sandvine, *The Global Internet Phenomena Report*, 2022.

[2] CISCO, *Cisco annual internet report (2018–2023) white paper*, 2020.

[3] Submarine, "H2HE - Submarine Networks," 2022-06-29. [Online]. Available: https://www.submarinenetworks.com/en/systems/intra-asia/hhscs

[4] ——, "Dunant," 2022-09-16. [Online]. Available: https://www.submarinenetworks.com/en/systems/trans-atlantic/dunant

[5] F. Pittala, R.-P. Braun, G. Bocherer, P. Schulte, M. Schaedler, S. Bettelli, S. Calabro, M. Kuschnerov, A. Gladisch, F.-J. Westphal, C. Xie, R. Chen, Q. Wang, and B. Zheng, "1.71 Tb/s Single-Channel and 56.51 Tb/s DWDM Transmission Over 96.5 km Field-Deployed SSMF," *IEEE Photonics Technology Letters*, vol. 34, no. 3, pp. 157–160, 2022.

[6] J. W. Nevin, S. Nallaperuma, N. A. Shevchenko, X. Li, M. S. Faruk, and S. J. Savory, "Machine Learning for Optical Fiber Communication Systems: An Introduction and Overview," *APL Photonics*, vol. 6, no. 12, p. 121101, 2021.

[7] F. Musumeci, C. Rottondi, A. Nag, I. Macaluso, D. Zibar, M. Ruffini, and M. Tornatore, "An Overview on Application of Machine Learning Techniques in Optical Networks," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1383–1408, 2019.

[8] T. Zhu, X. Bao, L. Chen, H. Liang, and Y. Dong, "Experimental Study on Stimulated Rayleigh Scattering in Optical Fibers," *Optics Express*, vol. 18, no. 22, p. 22958, 2010.

[9] A. R. Chraplyvy, "Limitations on Lightwave Communications imposed by Optical-Fiber Nonlinearities," *Journal of Lightwave Technology*, vol. 8, no. 10, pp. 1548–1557, 1990.

[10] I. Roberts, J. M. Kahn, J. Harley, and D. W. Boertjes, "Channel Power Optimization of WDM Systems Following Gaussian Noise Nonlinearity Model in Presence of Stimulated Raman Scattering," *Journal of Lightwave Technology*, vol. 35, no. 23, pp. 5237–5249, 2017.

[11] M. Cantono, D. Pilori, A. Ferrari, C. Catanese, J. Thouras, J.-L. Auge, and V. Curri, "On the Interplay of Nonlinear Interference Generation With Stimulated Raman Scattering for QoT Estimation," *Journal of Lightwave Technology*, vol. 36, no. 15, pp. 3131–3141, 2018.

[12] R. M. Shelby, M. D. Levenson, and P. W. Bayer, "Guided Acoustic-Wave Brillouin Scattering," *Physical Review B*, vol. 31, no. 8, pp. 5244–5252, 1985.

[13] P. Serena, F. Poli, A. Bononi, and J.-C. Antona, "Scattering Efficiency of Thermally Excited GAWBS in Fibres for Optical Communications," in *45th European Conference on Optical Communication (ECOC 2019)*, IEEE, Ed., Dublin, 2019.

[14] ITU-T, "G.694.2," 2003.

[15] ——, "G.694.1," 2020.

[16] Y.-W. Chen, K. Kuzmin, R. Zhang, M. Poirier, T. Tomimoto, G. Zarris, R. Moore, C. Chen, W. Wu, J. Huang *et al.*, "InP CDM and ICR Enabled 128Gbaud/DP-16QAM-PS and 120Gbaud/DP-QPSK Long-Haul Transmission," *IEEE Photonics Technology Letters*, vol. 34, no. 9, pp. 471–474, 2022.

[17] K.-D. Kammeyer, *Nachrichtenübertragung*. Springer-Verlag, 2013.

[18] P. A. Höher, *Grundlagen der digitalen Informationsübertragung*. Wiesbaden: Vieweg+Teubner, 2011.

[19] G. P. Agrawal, *Lightwave Technology: Components and Devices*. John Wiley & Sons, 2004.

[20] S. Pachnicke, *Lecture Notes Optical Communication*, 2021.

[21] J. Leibrich, *Digitale Modulationsverfahren und Signalverarbeitungsalgorithmen für flexible optische Datenübertragung*. Shaker Verlag, 2018.

[22] S. Ohlendorf, *Experimental Demonstration of Flexible Modulation Formats for Optical Data Center Interconnects*. Shaker Verlag, 2021.

[23] I. P. Kaminow, T. Li, and A. E. Willner, *Optical Fiber Telecommunications VB: Systems and Networks*. Elsevier, 2010.

[24] S. Kumar and M. J. Deen, *Fiber Optic Communications: Fundamentals and Applications*. John Wiley & Sons, 2014.

[25] K. Zhong, X. Zhou, J. Huo, C. Yu, C. Lu, and A. P. T. Lau, "Digital Signal Processing for Short-Reach Optical Communications: A Review of Current Technologies and Future Trends," *Journal of Lightwave Technology*, vol. 36, no. 2, pp. 377–400, 2018.

[26] X. Zhou and L. Nelson, "Advanced DSP for 400 Gb/s and Beyond Optical Networks," *Journal of Lightwave Technology*, vol. 32, no. 16, pp. 2716–2725, 2014.

[27] A. Willner, *Optical Fiber Telecommunications*. Elsevier, 2019.

[28] J. Hoyningen-Huene, *Experimental Investigations of Digital Signal Processing in Optical Access Networks*. Shaker Verlag, 2015.

[29] C. Ruprecht, *Orthogonales Frequenzmultiplexverfahren für passive optische Netze*. Shaker Verlag, 2018.

[30] Ciena Corporation, "What is ROADM?" 2022-06-29T09:57:36.000Z. [Online]. Available: https://www.ciena.com/insights/what-is/what-Is-roadm.html

[31] G. P. Agrawal, "Nonlinear fiber optics," in *Nonlinear Science at the Dawn of the 21st Century*. Springer, Berlin, Heidelberg, 2000, pp. 195–211.

[32] M. M. Shaker, M. S. Majeed, and R. W. Daoud, "Functioning the Intelligent Programming to Find Minimum Dispersion Wavelengths," *WSEAS Transactions on Communications*, vol. 8, no. 2, pp. 237–248, 2009.

[33] R. Noe, *Essentials of modern optical fiber communication*. Berlin: Springer, 2010.

[34] Akira Hasegawa and Frederick Tappert, "Transmission of stationary nonlinear optical pulses in dispersive dielectric fibers. I. Anomalous dispersion," *Applied Physics Letters*, vol. 23, no. 3, p. 142, 1973.

[35] S. Pachnicke, *Fiber-Optic Transmission Networks: Efficient Design and Dynamic Operation*. Springer Science & Business Media, 2011.

[36] E. Desurvire, J. R. Simpson, and P. C. Becker, "High-gain erbium-doped traveling-wave fiber amplifier," *Optics Letters*, vol. 12, no. 11, pp. 888–890, 1987.

[37] J. Leibrich, *Modeling and Simulation of Limiting Impairments on Next Generation's Transparent Optical WDM Transmission Systems with Advanced Modulation ...*. Shaker Verlag, 2007.

[38] R. Rath, D. Clausen, S. Ohlendorf, S. Pachnicke, and W. Rosenkranz, "Tomlinson–Harashima Precoding For Dispersion Uncompensated PAM-4 Transmission With Direct-Detection," *Journal of Lightwave Technology*, vol. 35, no. 18, pp. 3909–3917, 2017.

[39] G. P. Agrawal, *Fiber-Optic Communication Systems*. John Wiley & Sons, 2012.

[40] A. Carena, V. Curri, P. Poggiolini, and F. Forghieri, "Dynamic Range of Single-Ended Detection Receivers for 100GE Coherent PM-QPSK," *IEEE Photonics Technology Letters*, vol. 20, no. 15, pp. 1281–1283, 2008.

[41] Y. Painchaud, M. Poulin, M. Morin, and M. Têtu, "Performance of Balanced Detection in a Coherent Receiver," *Optics Express*, vol. 17, no. 5, pp. 3659–3672, 2009.

[42] K. Kikuchi, "Coherent Optical Communications: Historical Perspectives and Future Directions," *High Spectral Density Optical Communication Technologies*, pp. 11–49, 2010.

[43] ——, "Fundamentals of Coherent Optical Fiber Communications," *Journal of Lightwave Technology*, vol. 34, no. 1, pp. 157–179, 2016.

[44] E. Ip, A. P. T. Lau, D. J. F. Barros, and J. M. Kahn, "Coherent Detection in Optical Fiber Systems," *Optics Express*, vol. 16, no. 2, pp. 753–791, 2008.

[45] B. C. Chatterjee, N. Sarma, and E. Oki, "Routing and Spectrum Allocation in Elastic Optical Networks: A Tutorial," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1776–1800, 2015.

[46] T. Morioka, M. Jinno, H. Takara, and H. Kubota, "Innovative future optical transport network technologies," *NTT Technical Review*, vol. 9, 08 2011.

[47] M. Jinno, H. Takara, Y. Sone, K. Yonenaga, and A. Hirano, "Multiflow Optical Transponder for Efficient Multilayer Optical Networking," *IEEE Communications Magazine*, vol. 50, no. 5, pp. 56–65, 2012.

[48] N. Sambo, A. D'Errico, C. Porzi, V. Vercesi, M. Imran, F. Cugini, A. Bogoni, L. Potì, and P. Castoldi, "Sliceable Transponder Architecture Including Multiwavelength Source," *Journal of Optical Communications and Networking*, vol. 6, no. 7, pp. 590–600, 2014.

[49] V. López, B. de la Cruz, O. G. de Dios, O. Gerstel, N. Amaya, G. Zervas, D. Simeonidou, and J. P. Fernandez-Palacios, "Finding the Target Cost for Sliceable Bandwidth Variable Transponders," *Journal of Optical Communications and Networking*, vol. 6, no. 5, pp. 476–485, 2014.

[50] Y. Wang, X. Cao, and Y. Pan, "A Study of the Routing and Spectrum Allocation in Spectrum-Sliced Elastic Optical Path Networks," in *2011 Proceedings IEEE INFOCOM*. IEEE, 2011.

[51] K. Christodoulopoulos, I. Tomkos, and E. A. Varvarigos, "Elastic Bandwidth Allocation in Flexible OFDM-Based Optical Networks," *Journal of Lightwave Technology*, vol. 29, no. 9, pp. 1354–1366, 2011.

[52] M. Klinkowski, M. Ruiz, L. Velasco, D. Careglio, V. Lopez, and J. Comellas, "Elastic Spectrum Allocation for Time-Varying Traffic in FlexGrid Optical Networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 1, pp. 26–38, 2013.

[53] L. Gong, X. Zhou, W. Lu, and Z. Zhu, "A Two-Population Based Evolutionary Approach for Optimizing Routing, Modulation and Spectrum Assignments (RMSA) in O-OFDM Networks," *IEEE Communications Letters*, vol. 16, no. 9, pp. 1520–1523, 2012.

[54] X. Chen, B. Li, R. Proietti, H. Lu, Z. Zhu, and S. J. B. Yoo, "DeepRMSA: A Deep Reinforcement Learning Framework for Routing, Modulation and Spectrum Assignment in Elastic Optical Networks," *Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4155–4163, 2019.

[55] H. Wu, F. Zhou, Z. Zhu, and Y. Chen, "On the Distance Spectrum Assignment in Elastic Optical Networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2391–2404, 2017.

[56] P. Poggiolini, G. Bosco, A. Carena, V. Curri, Y. Jiang, and F. Forghieri, "The GN-Model of Fiber Non-Linear Propagation and its Applications," *Journal of Lightwave Technology*, vol. 32, no. 4, pp. 694–721, 2014.

[57] P. Poggiolini, "The GN Model of Non-Linear Propagation in Uncompensated Coherent Optical Systems," *Journal of Lightwave Technology*, vol. 30, no. 24, pp. 3857–3879, 2012.

[58] A. Carena, V. Curri, G. Bosco, P. Poggiolini, and F. Forghieri, "Modeling of the Impact of Nonlinear Propagation Effects in Uncompensated Optical Coherent Transmission Links," *Journal of Lightwave Technology*, vol. 30, no. 10, pp. 1524–1539, 2012. [Online]. Available: http://dx.doi.org/10.1109/jlt.2012.2189198

[59] P. Poggiolini and Y. Jiang, "Recent Advances in the Modeling of the Impact of Nonlinear Fiber Propagation Effects on Uncompensated Coherent Transmission Systems," *Journal of Lightwave Technology*, vol. 35, no. 3, pp. 458–480, 2017.

[60] P. Poggiolini, Y. Jiang, A. Carena, and F. Forghieri, "A Simple and Accurate Closed-Form EGN Model Formula," 2015.

[61] D. Wang, C. Lu, A. P. T. Lau, and S. He, "Adaptive Chromatic Dispersion Compensation for Coherent Communication Systems Using Delay-Tap Sampling Technique," *IEEE Photonics Technology Letters*, vol. 23, no. 14, pp. 1016–1018, 2011.

[62] J. C. M. Diniz, E. P. da Silva, M. Piels, and D. Zibar, "Joint IQ Skew and Chromatic Dispersion Estimation for Coherent Optical Communication Receivers," *Signal Processing in Photonic Communications*, p. SpTu2F.2, 2016.

[63] T. M. Schmidl and D. C. Cox, "Robust Frequency and Timing Synchronization for OFDM," *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1613–1621, 1997.

[64] M. Kuschnerov, M. Chouayakh, K. Piyawanno, B. Spinnler, E. de Man, P. Kainzmaier, M. S. Alfiad, A. Napoli, and B. Lankl, "Data-Aided Versus Blind Single-Carrier Coherent Receivers," *IEEE Photonics Journal*, vol. 2, no. 3, pp. 387–403, 2010.

[65] D. Chu, "Polyphase codes with good periodic correlation properties (corresp.)," *IEEE Transactions on Information Theory*, vol. 18, no. 4, pp. 531–532, 1972.

[66] M. Selmi, Y. Jaouen, and P. Ciblat, "Accurate Digital Frequency Offset Estimator for Coherent PolMux QAM Transmission Systems," in *35th European Conference on Optical Communication*, IEEE, Ed., Vienna, 2009, p. P3.08.

[67] M. S. Faruk and S. J. Savory, "Digital Signal Processing for Coherent Transceivers Employing Multilevel Formats," *Journal of Lightwave Technology*, vol. 35, no. 5, pp. 1125–1141, 2017.

[68] E. Ip and J. M. Kahn, "Digital Equalization of Chromatic Dispersion and Polarization Mode Dispersion," *Journal of Lightwave Technology*, vol. 25, no. 8, pp. 2033–2043, 2007.

[69] D. Clausen, *Experimentelle Untersuchung adaptiver flexibler digitaler Signalverarbeitung für hybride Modulationsformate*. tredition, 2021.

[70] H. Bulow, F. Buchali, and A. Klekamp, "Electronic Dispersion Compensation," *Journal of Lightwave Technology*, vol. 26, no. 1, pp. 158–167, 2008.

[71] R. Kudo, T. Kobayashi, K. Ishihara, Y. Takatori, A. Sano, and Y. Miyamoto, "Coherent Optical Single Carrier Transmission Using Overlap Frequency Domain Equalization for Long-Haul Optical Systems," *Journal of Lightwave Technology*, vol. 27, no. 16, pp. 3721–3728, 2009.

[72] R. Andres Soriano, F. N. Hauske, N. Guerrero Gonzalez, Z. Zhang, Y. Ye, and I. Tafur Monroy, "Chromatic Dispersion Estimation in Digital Coherent Receivers," *Journal of Lightwave Technology*, vol. 29, no. 11, pp. 1627–1637, 2011.

[73] H. Wymeersch and P. Johannisson, "Maximum-Likelihood-Based Blind Dispersion Estimation for Coherent Optical Communication," *Journal of lightwave technology*, vol. 30, no. 18, pp. 2976–2982, 2012.

[74] C. Xia, *Advanced Electronic Distortion Equalization for High Speed Optical SMF and MMF Communications*. Shaker Verlag, 2008.

[75] R. Rath, *Investigation of Digital Signal Processing Techniques for Compensation of Linear and Nonlinear Impairments in Fiber-optic Communication Systems*. Shaker Verlag, 2020.

[76] S. Warm, C.-A. Bunge, T. Wuth, and K. Petermann, "Electronic Dispersion Precompensation With a 10-Gb/s Directly Modulated Laser," *IEEE Photonics Technology Letters*, vol. 21, no. 15, pp. 1090–1092, 2009.

[77] Y. Gao, F. Zhang, L. Dou, Z. Chen, and A. Xu, "Intra-channel nonlinearities mitigation in pseudo-linear coherent QPSK transmission systems via nonlinear electrical equalizer," *Optics Communications*, vol. 282, no. 12, pp. 2421–2425, 2009.

[78] K. V. Peddanarappagari and M. Brandt-Pearce, "Volterra Series Transfer Function of Single-Mode Fibers," *Journal of Lightwave Technology*, vol. 15, no. 12, pp. 2232–2241, 1997.

[79] R. Rath, J. Hoyningen-Huene, and W. Rosenkranz, "Combined Precoding and Volterra Equalization for the Mitigation of Fiber-Optic Nonlinear Channel Impairments," *Combined In Photonic Networks; 15. ITG Symposium*, pp. 1–5, 2014.

[80] N.-P. Diamantopoulos, H. Nishi, W. Kobayashi, K. Takeda, T. Kakitsuka, and S. Matsuo, "On the Complexity Reduction of the Second-Order Volterra Nonlinear Equalizer for IM/DD Systems," *Journal of Lightwave Technology*, vol. 37, no. 4, pp. 1214–1224, 2019.

[81] T. Wettlin, S. Pachnicke, T. Rahman, J. Wei, S. Calabro, and S. Nebojsa, "Complexity Reduction of Volterra Nonlinear Equalization for Optical Short-Reach IM/DD Systems," *Photonic Networks*, vol. 21th ITG-Symposium, 2020.

[82] M. S. Faruk and K. Kikuchi, "Compensation for In-Phase/Quadrature Imbalance in Coherent-Receiver Front End for Optical Quadrature Amplitude Modulation," *IEEE Photonics Journal*, vol. 5, no. 2, p. 7800110, 2013.

[83] O. Zia-Chahabi, R. Le Bidan, M. Morvan, and C. Laot, "Efficient Frequency-Domain Implementation of Block-LMS/CMA Fractionally Spaced Equalization for Coherent Optical Communications," *IEEE Photonics Technology Letters*, vol. 23, no. 22, pp. 1697–1699, 2011.

[84] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[85] J. Dong and M. Qi, "K-means Optimization Algorithm for Solving Clustering Problem," in *Second International Workshop on Knowledge Discovery and Data Mining*, IEEE Computer Society, Ed. Moscow: IEEE, 2009, pp. 52–55.

[86] T. K. Moon, "The Expectation-Maximization Algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[87] D. Zibar, O. Winther, N. Franceschi, R. Borkowski, A. Caballero, V. Arlunno, M. N. Schmidt, N. G. Gonzales, B. Mao, Y. Ye, K. J. Larsen, and I. T. Monroy, "Nonlinear impairment compensation using expectation maximization for dispersion managed and unmanaged pdm 16-qam transmission," *Optics Express*, vol. 20, no. 26, pp. B181–B196, 2012.

[88] J. Achiam, "Spinning Up in Deep Reinforcement Learning," 2018.

[89] D. M. D'Addona, *Neural Network*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[90] A. Ng, "Cs229 lecture notes," *CS229 Lecture notes*, vol. 1, no. 1, pp. 1–3, 2000.

[91] D. Rafique and L. Velasco, "Machine Learning for Network Automation: Overview, Architecture, and Applications," *Journal of Optical Communications and Networking*, vol. 10, no. 10, p. D126, 2018.

[92] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ser. Adaptive computation and machine learning. Cambridge, Massachusetts: The MIT Press, 2016.

[93] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai, "Gradient Descent Finds Global Minima of Deep Neural Networks," *International Conference on Machine Learning*, pp. 1675–1685, 2019.

[94] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020.

[95] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[96] G. Van Houdt, C. Mosquera, and G. Nápoles, "A Review on the Long Short-Term Memory Model," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5929–5955, 2020.

[97] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, 2018.

[98] Stanford, *Cs231n: Convolutional neural networks for visual recognition*, 2019. [Online]. Available: https://cs231n.github.io/neural-networks-1/.[Accessed07Januar2021]

[99] A. Oppermann, *Optimization in Deep Learning: AdaGrad, RMSProp, ADAM*, 2021. [Online]. Available: https://artemoppermann.com/optimization-in-deep-learning-adagrad-rmsprop-adam/ .[Accessed14September2022]

[100] I. J. Goodfellow, O. Vinyals, and A. M. Saxe, "Qualitatively Characterizing Neural Network Optimization Problems," *arXiv preprint arXiv:1412.6544*, 2014.

[101] S. J. Reddi, S. Kale, and S. Kumar, "On the Convergence of Adam and Beyond," *arXiv preprint arXiv:1904.09237*, 2019.

[102] H. Li, J. Li, X. Guan, B. Liang, Y. Lai, and X. Luo, "Research on Overfitting of Deep Learning," in *2019 15th International Conference on Computational Intelligence and Security (CIS)*. IEEE, 2019, pp. 78–81.

[103] T.-T. Wong and P.-Y. Yeh, "Reliable accuracy estimates from k-fold cross validation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1586–1594, 2019.

[104] M. Mitchell, *An Introduction to Genetic Algorithms*. MIT press, 1998.

[105] D. E. Goldberg. New York: Addison-Wesley, 1989.

[106] A. Shukla, H. M. Pandey, and D. Mehrotra, "Comparative Review of Selection Techniques in Genetic Algorithm," in *2015 international conference on futuristic trends on computational analysis and knowledge management (ABLAZE)*. IEEE, 2015, pp. 515–519.

[107] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT press, 1992.

[108] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *Foundations of genetic algorithms*. Elsevier, 1991, vol. 1, pp. 69–93.

[109] K. A. De Jong and W. M. Spears, "A Formal Analysis of the Role of Multi-point Crossover in Genetic Algorithms," *Annals of mathematics and Artificial intelligence*, vol. 5, no. 1, pp. 1–26, 1992.

[110] L. Wang, *Support Vector Machines: Theory and Applications*. Springer, Berlin, Heidelberg, 2005.

[111] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, 1992.

[112] F. N. Khan, Q. Fan, C. Lu, and A. P. T. Lau, "An Optical Communication's Perspective on Machine Learning and Its Applications," *Journal of Lightwave Technology*, vol. 37, no. 2, pp. 493–516, 2019.

[113] G. Wu, E. Y. Chang, and N. Panda, "Formulating Distance Functions via the Kernel Trick," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2005, pp. 703–709.

[114] Schölkopf, *Support vector learning.* München, Germany: Oldenbourg, 1997.

[115] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[116] B. Schölkopf, A. J. Smola, and F. Bach, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2002.

[117] Bishop, *Pattern recognition and machine learning.* Springer, 2006.

[118] A. Rocha and S. K. Goldenstein, "Multiclass from binary: Expanding one-versus-all, one-versus-one and ecoc-based Approaches," *Ieee Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 289–302, 2013.

[119] M. Aly, "Survey on multiclass classification methods," *Neural Netw*, vol. 19, no. 1, 2005.

[120] D. J. Sebald and J. A. Bucklew, "Support Vector Machines and the Multiple Hypothesis Test Problem," *IEEE Transactions on Signal Processing*, vol. 49, no. 11, pp. 2865–2872, 2001.

[121] L. Zhibin and j. Lianwen, "LATTICESVM—A new Method for Multi-Class Support Vector Machines," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, IEEE, Ed., Hong Kong, 2008, pp. 727–733.

[122] M. Li, S. Yu, J. Yang, Z. Chen, Y. Han, and W. Gu, "Nonparameter Nonlinear Phase Noise Mitigation by Using M-ary Support Vector Machine for Coherent Optical Systems," *IEEE Photonics Journal*, vol. 5, no. 6, p. 7800312, 2013.

[123] R. Weixer, J. Koch, P. Plany, S. Ohlendorf, and S. Pachnicke, "Mitigation of Nonlinear Impairments by using Support Vector Machine and Nonlinear Volterra Equalizer," *Applied Sciences*, vol. 9, no. 18, p. 3800, 2019.

[124] J. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines," 1998.

[125] V. Kecman, Michael Vogt, and Te Ming Huang, "On the equality of kernel adatron and sequential minimal optimization in classification and regression tasks and alike algorithms for kernel machines," in *The European Symposium on Artificial Neural Networks (ESANN)*, vol. 3, Bruges, 2003, pp. 215–222.

[126] V. Kecman, T.-M. Huang, and M. Vogt, "Iterative single data algorithm for training kernel machines from huge data sets: Theory and performance," in *Support Vector Machines: Theory and Applications*. Springer, Berlin, Heidelberg, pp. 255–274.

[127] R. S. Sutton and A. G. Barto, *Reinforcement Learning, second edition: An Introduction*. MIT Press, 2018.

[128] T. T. Nguyen and V. J. Reddi, "Deep Reinforcement Learning for Cyber Security," *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[129] B. Sutton, *Introduction to Reinforcement Learning*, 1998.

[130] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning," *International Conference on Machine Learning*, pp. 1928–1937, 2016.

[131] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[132] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust Region Policy Optimization," *International Conference on Machine Learning*, pp. 1889–1897, 2015.

[133] Q. Huang, "Model-Based or Model-Free, a Review of Approaches in Reinforcement Learning," in *International Conference on Computing and Data Science (CDS)*. Stanford: IEEE, 2020, pp. 219–221.

[134] Dayan and Watkins, "Q-Learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[135] D. P. Bertsekas and J. N. Tsitsiklis, "Neuro-dynamic programming: an overview," in *Proceedings of 1995 34th IEEE Conference on Decision and Control*. IEEE, null.

[136] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-Level Control through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[137] A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine, "Stabilizing Off-Policy Q-learning via Bootstrapping Error Reduction," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[138] R. J. Williams, "Simple Statistical Gradient-following Algorithms for Connectionist Reinforcement Learning," *Machine learning*, vol. 8, no. 3, pp. 229–256, 1992.

[139] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," *International Conference on Machine Learning*, pp. 387–395, 2014.

[140] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, *Continuous Control with Deep Reinforcement Learning.* arXiv preprint arXiv:1509.02971, 2015.

[141] I. Grondman, L. Busoniu, G. A. Lopes, and R. Babuska, "A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1291–1307, 2012.

[142] S. Kullback, *Information Theory and Statistics.* Courier Corporation, 1997.

[143] S. Kumar, "Analysis of Nonlinear Phase Noise in Coherent Fiber-Optic Systems Based on Phase Shift Keying," *Journal of Lightwave Technology*, vol. 27, no. 21, pp. 4722–4733, 2009.

[144] N. Ekanayake and H. V. R. Herath, "Effect of Nonlinear Phase Noise on the Performance of -Ary PSK Signals in Optical Fiber Links," *Journal of Lightwave Technology*, vol. 31, no. 3, pp. 447–454, 2012.

[145] I. Roudas, M. Sauer, J. Hurley, Y. Mauro, and S. Raghavan, "Compensation of Coherent DQPSK Receiver Imperfections," in *2007 Digest of the IEEE/LEOS Summer Topical Meetings.* IEEE, 2007.

[146] I. Fatadin, S. J. Savory, and D. Ives, "Compensation of Quadrature Imbalance in an Optical QPSK Coherent Receiver," *IEEE Photonics Technology Letters*, vol. 20, no. 20, pp. 1733–1735, 2008.

[147] S. Haykin, *Digital communications.* Wiley, 1988.

[148] L. Anttila, M. Valkama, and M. Renfors, "Blind Moment Estimation Techniques for I/Q Imbalance Compensation in Quadrature Receivers," in *2006 IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications.* IEEE, 2006.

[149] G. Gao, J. Zhang, and W. Gu, "Analytical Evaluation of Practical DBP-Based Intra-Channel Nonlinearity Compensators," *IEEE Photonics Technology Letters*, vol. 25, no. 8, pp. 717–720, 2013.

[150] E. Ip and J. M. Kahn, "Compensation of Dispersion and Nonlinear Impairments Using Digital Backpropagation," *Journal of Lightwave Technology*, vol. 26, no. 20, pp. 3416–3425, 2008.

[151] I. Sackey, F. Da Ros, J. K. Fischer, T. Richter, M. Jazayerifar, C. Peucheret, K. Petermann, and C. Schubert, "Kerr Nonlinearity Mitigation: Mid-Link Spectral Inversion Versus Digital

Backpropagation in 5×28-GBd PDM 16-QAM Signal Transmission," *Journal of Lightwave Technology*, vol. 33, no. 9, pp. 1821–1827, 2015.

[152] O. Sidelnikov, A. Redyuk, and S. Sygletos, "Equalization Performance and Complexity Analysis of Dynamic Deep Neural Networks in Long Haul Transmission Systems," *Optics Express*, vol. 26, no. 25, pp. 32 765–32 776, 2018.

[153] P. J. Freire, Y. Osadchuk, B. Spinnler, W. Schairer, A. Napoli, N. Costa, J. E. Prilepsky, and S. K. Turitsyn, "Experimental Study of Deep Neural Network Equalizers Performance in Optical Links," in *Optical Fiber Communication Conference (OFC) 2021*. Washington, D.C.: OSA, 2021.

[154] S. Deligiannidis, A. Bogris, C. Mesaritakis, and Y. Kopsinis, "Compensation of Fiber Nonlinearities in Digital Coherent Systems Leveraging Long Short-Term Memory Neural Networks," *Journal of Lightwave Technology*, vol. 38, no. 21, pp. 5991–5999, 2020.

[155] E. Alpaydin, *Introduction to Machine Learning, fourth edition*. MIT Press, 2020.

[156] D. Wang, M. Zhang, M. Fu, Z. Cai, Z. Li, H. Han, Y. Cui, and B. Luo, "Nonlinearity Mitigation Using a Machine Learning Detector Based on -Nearest Neighbors," *IEEE Photonics Technology Letters*, vol. 28, no. 19, pp. 2102–2105, 2016.

[157] D. Wang, M. Zhang, Z. Cai, Y. Cui, Z. Li, H. Han, M. Fu, and B. Luo, "Combatting Nonlinear Phase Noise in Coherent Optical Systems with an Optimized Decision Processor based on Machine Learning," *Optics Communications*, vol. 369, pp. 199–208, 2016.

[158] A. Osareh, M. Mirmehdi, B. Thomas, and R. Markham, "Comparative Exudate Classification Using Support Vector Machines and Neural Networks," in *International Conference on Medical Image Computing and Computer-assisted Intervention*, Springer, Ed., Heidelberg, 2002, pp. 413–420.

[159] D. Wang, M. Zhang, Z. Li, Y. Cui, J. Liu, Y. Yang, and H. Wang, "Nonlinear Decision Boundary Created by a Machine Learning-Based Classifier to Mitigate Nonlinear Phase Noise," in *41st European Conference on Optical Communication*, IEEE, Ed., Valencia, 2015.

[160] ITU-T, "G.975," 2000.

[161] R. Weixer, J. Koch, S. Ohlendorf, and S. Pachnicke, "Experimental Demonstration of Nonlinearity Compensation by Using SVM and Nonlinear Volterra Equalizer for 80 GBd DP-16-QAM Transmission," in *22nd International Conference on Transparent Optical Networks (ICTON)*, IEEE, Ed., Bari, 2020, pp. 1–4.

[162] S. Ohlendorf, T. Wettlin, S. Pachnicke, and W. Rosenkranz, "Optimized Flexible Mappings with Multidimensional Modulation for Coherent Optical Transport," in *45th European Conference on Optical Communication (ECOC 2019)*, IEEE, Ed., Dublin, 2019.

[163] L. Yan, E. Agrell, H. Wymeersch, P. Johannisson, R. Di Taranto, and M. Brandt-Pearce, "Link-Level Resource Allocation for Flexible-Grid Nonlinear Fiber-Optic Communication Systems," *IEEE Photonics Technology Letters*, vol. 27, no. 12, pp. 1250–1253, 2015.

[164] D. J. Ives and S. J. Savory, "Transmitter Optimized Optical Networks," in *Optical Fiber Communication Conference*. Anaheim, California: OSA The Optical Society, 2013, p. JW2A.64.

[165] David J. Ives, Polina Bayvel, and Seb J. Savory, "Adapting Transmitter Power and Modulation Format to Improve Optical Network Performance Utilizing the Gaussian Noise Model of Nonlinear Impairments," *Journal of Lightwave Technology*, vol. 32, no. 21, pp. 3485–3494, 2014.

[166] J. Lu, G. Zhou, Q. Fan, D. Zeng, C. Guo, L. Lu, J. Li, C. Xie, C. Lu, F. N. Khan, and A. P. T. Lau, "Performance Comparisons between Machine Learning and Analytical Models for Quality of Transmission Estimation in Wavelength-Division-Multiplexed Systems [Invited]," *Journal of Optical Communications and Networking*, vol. 13, no. 4, p. B35, 2021.

[167] X. Zhou, W. Lu, L. Gong, and Z. Zhu, "Dynamic RMSA in Elastic Optical Networks with an Adaptive Genetic Algorithm," in *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2012.

[168] M. Zeng, Y. Li, W. Fang, W. Lu, X. Liu, H. Yu, and Z. Zhu, "Control Plane Innovations to Realize Dynamic Formulation of Multicast Sessions in Inter-DC Software-Defined Elastic Optical Networks," *Optical Switching and Networking*, vol. 23, pp. 259–269, 2017.

[169] S. Yan, F. N. Khan, A. Mavromatis, D. Gkounis, Q. Fan, F. Ntavou, K. Nikolovgenis, F. Meng, E. H. Salas, C. Guo, C. Lu, A. P. T. Lau, R. Nejabati, and D. Simeonidou, "Field Trial of Machine-Learning-Assisted and SDN-based Optical Network Planning with Network-Scale Monitoring Database," in *43rd European Conference on Optical Communication (ECOC)*, IEEE, Ed. Gothenborg: IEEE, 2017.

[170] X. Chen, J. Guo, Z. Zhu, R. Proietti, A. Castro, and S. J. B. Yoo, "Deep-RMSA: A Deep-Reinforcement-Learning Routing, Modulation and Spectrum Assignment Agent for Elastic Optical Networks," *Optical Fiber Communication Conference*, p. W4F.2, 2018.

[171] X. Chen, R. Proietti, C.-Y. Liu, Z. Zhu, and S. J. Ben Yoo, "Exploiting Multi-Task Learning to Achieve Effective Transfer Deep Reinforcement Learning in Elastic Optical Networks," in *Optical Fiber Communication Conference (OFC) 2020*. Washington, D.C.: OSA, 2020.

[172] X. Luo, C. Shi, L. Wang, X. Chen, Y. Li, and T. Yang, "Leveraging Double-Agent-based Deep Reinforcement Learning to Global Optimization of Elastic Optical Networks with Enhanced Survivability," *Optics Express*, vol. 27, no. 6, pp. 7896–7911, 2019.

[173] R. Weixer, S. Kühl, R. Morais, B. Spinnler, W. Schairer, B. Sommerkorn-Krombholz, and S. Pachnicke, "A Reinforcement Learning Framework for Parameter Optimization in Elastic Optical Networks," in *46th European Conference on Optical Communication*, IEEE, Ed., Brussels, 2020.

[174] S. Kühl, R. Koch, W. Schairer, B. Spinnler, and S. Pachnicke, "Optimized Bandwidth Variable Transponder Configuration in Elastic Optical Networks using Reinforcement Learning," *Photonic Networks; 22th ITG Symposium*, pp. 1–4, 2021.

[175] R. Koch, S. Kühl, R. Morais, B. Spinnler, W. Schairer, B. Sommerkorn-Krombholz, and S. Pachnicke, "Reinforcement Learning for Generalized Parameter Optimization in Elastic Optical Networks," *Journal of Lightwave Technology*, vol. 40, no. 3, pp. 567–574, 2022.

[176] R. Koch, S. Kühl, W. Schairer, B. Spinnler, and S. Pachnicke, "High-Generalizability Reinforcement Learning Based Capacity Optimization in WDM Long-Haul Networks," *IEEE Photonics Technology Letters*, vol. 34, no. 14, pp. 891–894, 2022.

[177] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, C. DhariwalPrafulla and Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu, "Stable baselines," *GitHub repository*, vol. 2018.

[178] G. Hinton, *Coursera Neural Networks for Machine Learning, Lecture 6*, 2018.

Veröffentlichungen des Autors mit Angaben über die Höhe des Eigenanteils bei Publikationen/Manuskripten mit mehreren Autoren

Mitigation of Nonlinear Impairments by Using Support Vector Machine and Nonlinear Volterra Equalizer, Applied Sciences vol. 9 no. 18, 2019, DOI: 10.3390/app9183800.

Autoren: R. Weixer, J. Koch, P. Plany, S. Ohlendorf, S. Pachnicke

Konzeptionierung	Planung	Durchführung	Manuskripterstellung
Hoch	Hoch	Hoch	Hoch

Equalization of Soliton Transmission Based on Nonlinear Fourier Transform using Neural Networks, 45th European Conference on Optical Communication (ECOC), Dublin, 2019.

Autoren: J. Koch, R. Weixer, S. Pachnicke

Konzeptionierung	Planung	Durchführung	Manuskripterstellung
Niedrig	Niedrig	Mittel	Niedrig

Impact of DAC Properties on Tomlinson-Harashima Precoding for 200 Gb/s Intra Datacenter Links, IEEE Photonics Conference (IPC), San Antonio, USA, 2019.

Autoren: T. Wettlin, R. Weixer, T. Rahman, J. Wei, S. Calabro, N. Stojanovic, S. Pachnicke

Konzeptionierung	Planung	Durchführung	Manuskripterstellung
Niedrig	Niedrig	Mittel	Niedrig

(Invited) Experimental Demonstration of Nonlinearity Compensation by Using SVM and Nonlinear Volterra Equalizer for 80 GBd DP-16-QAM, International Conference on Transparent Optical Networks (ICTON), Bari, Italy, 2020.

Autoren: R. Weixer, J. Koch, S. Ohlendorf, S. Pachnicke

Konzeptionierung	Planung	Durchführung	Manuskripterstellung
Hoch	Hoch	Mittel	Hoch

A Reinforcement Learning Framework for Parameter Optimization in Elastic Optical Networks, 46th European Conference on Optical Communication (ECOC), Brussels, Belgium, 2020.

Autoren: R. Weixer, S. Kühl, R. M. Morais, B. Spinnler, W. Schairer, B. Sommerkorn-Krombholz, S. Pachnicke

Konzeptionierung	Planung	Durchführung	Manuskripterstellung
Hoch	Hoch	Mittel	Hoch

Optimized Bandwidth Variable Transponder Configuration in Elastic Optical Networks using Reinforcement Learning; 22th ITG-Symposium, Leipzig, 2021.

Autoren: S. Kühl, R. Koch, W. Schairer, B. Spinnler, S. Pachnicke

Konzeptionierung	Planung	Durchführung	Manuskripterstellung
Mittel	Mittel	Niedrig	Niedrig

Reinforcement Learning for Generalized Parameter Optimization in Elastic Optical Networks, IEEE Journal of Lightwave Technology. (JLT), vol. 40 no. 3, 2021.

Autoren: R. Koch, S. Kühl, R. Morais, B. Spinnler, W. Schairer, B. Sommerkorn-Krombholz, S. Pachnicke

Konzeptionierung	Planung	Durchführung	Manuskripterstellung
Hoch	Mittel	Mittel	Hoch

High-Generalizability Reinforcement Learning Based Capacity Optimization in WDM Long-Haul Networks, IEEE Photonics Technology Letters (PTL), vol. 34 no. 17, 2022.

Autoren: R. Koch, S. Kühl, W. Schairer, B. Spinnler, S. Pachnicke

Konzeptionierung	Planung	Durchführung	Manuskripterstellung
Hoch	Mittel	Mittel	Hoch