MICAS 913 Project
Equalization in Fiber-Optic Communication using Neural Networks

Mansoor Yousefi
Télécom Paris, Palaiseau, France

Abstract

This is a guide on a semester simulation project on the deep learning of a communication system. The project consists of implementing a point-to-point fiber transmission system, including a generative neural network representing the channel, and a predictive neural network for equalization at the receiver. The bit error rate (BER) is computed as a function of the signal-to-noise ratio for different modulation formats, and compared with the BER of a zero-forcing receiver. The end-to-end learning of the transmitter and receiver is implemented in an autoencoder optionally.

Contents

1 Introduction

We implement the deep learning of a point-to-point optical fiber transmission system. The project consists of two parts. In the first part in Section 2, we describe a generative model for producing the data sets. This part consists of simulating a transmitter (TX), a receiver (RX), and a generative neural network (NN) representing the channel. In the second part in Section 4, we will design a NN for equalization at the RX. A bonus mark is provided to groups that implement joint end-to-end learning of the TX and RX in an auto-encoder model, by back-propagation of the training error across the channel in the gradient descent.
The software may be written in Python, MATLAB (or GNU Octave) or Julia[1]. You should write a report in which you will present the plots on the performance and complexity of the deep learning receiver, and answer the questions in this guide. The report is written in LaTeX.
The instructor will provide office hours and hands-on help, however, the project must be completed mostly at home during the semester. The part on the communications is explained in detail in this guide together with the pseudo code, to help students who lack background in communications.

2 A simplified point-to-point communication system

Layered architecture. The *separation theorem* for point-to-point data communication implies that the source and channel can be designed separately, suggesting a layered communication architecture [1]. The data (*e.g.*, a camera image) is initially in the analogue domain. The analogue signal is converted to a sequence of bits by the source encoder (consisting of sampling, quantization and data compression). This digital representation has a number desirable advantages [1, Chap. 4.1.1]. After applying digital signal processing, the bits sequence is transformed back to a continuous-time signal by the channel encoder.
The channel encoder performs channel coding, modulation (bits-to-symbols and symbols-to-signal mapping) and baseband-to-passband frequency translation. The inverse of these layers is implemented in the channel decoder. In this project, we only implement the part from the input of the channel encoder to the output of the channel decoder. Furthermore, we neglect forward error-correction coding, and consider a baseband model. The resulting simplified diagram is shown in Fig. 1 The foundations of the digital communication are reviewed in [1–3].

Setting up a simulation environment. Create a simulation environment as follows.

a. SSH into the school server, issuing `ssh username@ssh.enst.fr`

[1]MATLAB is used for demonstration in this guide due to its compact syntax.

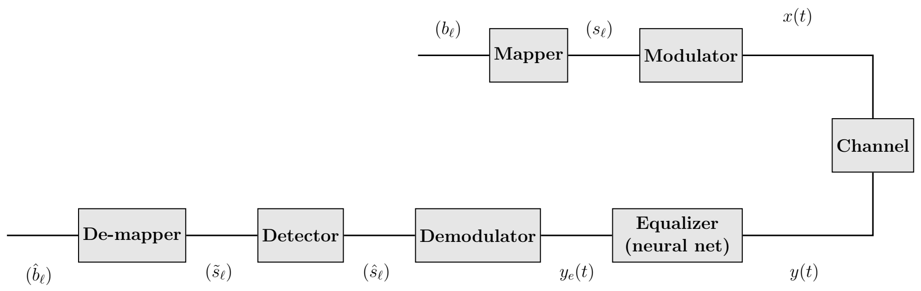


Fig. 1: Block diagram of a simple communication system.

b. Create a folder “micas913-project-name” with sub-folders “generative” and “inference” in your home directory, where “name” is the last name of the group representative. You will save all files here.

c. Create the following Matlab files (m-files): parameters.m, main.m, source.m, mod.m, channel.m, nnet_gen.m, demod.m, nn_pred.m, detector.m, and ber.m.

   In Python, the functions and classed are organized in modules. For instance, you may create a module `generative.py` that contains all functions required to produce the data set, and a module `predictive.py` for functions that implement the NN receiver.

In what follows, we describe and implement each block in the system diagram Fig. 1.

2.1 Transmitter

2.1.1 Binary source

A discrete source is a discrete-time stochastic process.

Question 1. Consider a Bernoulli stochastic process, *i.e.*, a sequence

$$b^N = (b_0, b_1, \cdots, b_{N-1}),$$

where b_i are random variables drawn independently from a Bernoulli probability distribution function (PDF)

$$p(b_i) = \begin{cases} p, & b_i = 0, \\ 1-p, & b_i = 1. \end{cases}$$

Write an m-function `b = source(N, p)` that generates b^N, and test it with $N = 1024$ and $p = 1/2$.

□

Remark 1. In practice, the input bit stream is a pseudo-random binary sequence (PRBS). □

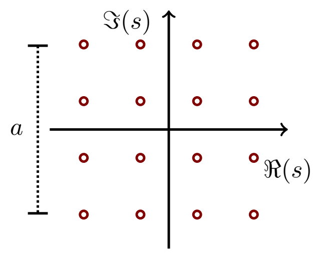


Fig. 2: A 16-QAM constellation.

2.1.2 Modulation

The modulation consists of bits-to-symbols and symbols-to-signal mapping.

2.1.3 Bits-to-symbol mapping

A constellation $\mathcal{C}$ of order M is a set of M real or complex numbers. The energy of the constellation with respect to a probability distribution $p_S(s)$ on $\mathcal{C}$ is the variance of $p_S(s)$. If the symbols in the constellation are drawn from a uniform distribution, the energy of the constellation is

$$\mathcal{P} = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} |s_i|^2, \tag{1}$$

where $|\mathcal{C}|$ is the cardinality of $\mathcal{C}$.

Question 2. We consider a quadrature amplitude modulation (QAM) constellation with $M = 16$ points, shown in Fig. 2. Write a function that takes M and $\mathcal{P}$ and outputs a vector of points in the M-QAM constellation. You could write this for any M, or $M = 16$. Consider the average energy $\mathcal{P} = 1$. Obtain, verify the energy, and plot the constellation.

```
function cnt = constellation(M, P)

...

end
```

Remark 2. The code in this guide should be viewed as pseudo-code. You have to complete the code, add missing variables, arguments, etc. You may have to change the parameters as well in order to obtain meaningful results, especially in places where this is indicated. □

There are M points in the constellation. Each point is mapped to a binary sequence of length $\log_2 M$. We consider a *Gray mapping*, in which the Hamming distance between the binary sub-sequences associated with the neighbor points is 1.

Question 3. Find a Gray mapping, and illustrate it on the constellation. □

The sequence of bits b^N is divided into $n_s = N/\log_2 M$ sub-sequences, each of which with $\log_2(M)$ bits. Each sub-sequence is mapped to a point in the constellation, to obtain a sequence of complex numbers $s^{n_s} = (s_0, s_1, \cdots, s_{n_s-1})$, where $s_i \in \mathcal{C}$ are *symbols*. It can be shown that if $p = 1/2$ then s^{n_s} is a sequence of complex random variables drawn i.i.d. from the uniform distribution over $\mathcal{C}$.

Question 4. Write an m-function `s = bit_to_symb(b, cnt)`, where b is the input bit stream and cnt is the (vector or matrix of the) constellation points, and s is the sequence of symbols. □

2.1.4 Symbols-to-signal mapping

Linear modulation consists of expanding a signal $q(t, 0)$ onto an orthonormal basis. The *Nyquist-Shannon sampling theorem*, applied to the space of functions with finite energy and bandlimited to B Hz, provides a suitable representation

$$x(t) = \sum_{\ell=\ell_1}^{\ell_2} s_\ell \operatorname{sinc}(Bt - \ell), \tag{2}$$

where $(s_k)_{k\in\mathbb{Z}}$ is a sequence of zero-mean independent identically distributed (i.i.d.) random variables drawn uniformly from $\mathcal{C}$, and $\ell_1 = -\lfloor n_s/2 \rfloor$, $\ell_2 = \lceil n_s/2 \rceil - 1$.
In practice, an arbitrary wave generator (AWG) takes the sequence of symbols $\Re(s^{n_s})$ (the in-phase component I) and $\Im(s^{n_s})$ (the quadrature component Q) and generates the analogue signal $q(t, 0)$. This step is performed using the digital-to-analogue converters (DACs) in the AWG.

Question 5. The time interval between two consecutive sinc functions is $1/B$. Thus, the symbol rate is $R_s = 1/B$ symbols/s. Express the bit rate R_b in bits/s in terms of R_s.

□

Constraints on input. It is assumed that $x(t)$ is bandlimited to B Hz. Further, the transmitter is subject to the average power constraint

$$\lim_{T\to\infty} \frac{1}{T} \mathsf{E}\left\{ \int_{-\frac{T}{2}}^{\frac{T}{2}} |x(\tau)|^2 \mathrm{d}\tau \right\} = \mathcal{P}. \tag{3}$$

In Appendix B, it is shown that the average power of $x(t)$ is $\mathsf{E}|s_k|^2 = \mathcal{P}$.

Question 6. We write one m-function main.m for each section, in which you set up and run the project. In this section, you will gradually complete the first main.m file.
Complete the following main.m file.

```
% time mesh
T  = 400; % you have to choose this
```

```
N  = 2^10; % you have to choose this
dt = T/N;
t  = ... % should be from -T/2 to T/2 with length N

% frequency mesh
F  = 1/dt;
df = 1/T;
f  = ... % should be from -F/2 to F/2 with length N

% bits to signal
M = 16; % size of the constellation
ns = 3; % number of symbols (or sinc functions); test with s=1
nb = ... ; % number of bits
p =1/2; % probability of zero
b = source(nb, p); % Bernoulli source, random bits sequence
s = bit_to_symb(b, cnt); % symbol sequence; could be inside modulator.m
q0 = mod(t, s, B); % transmitted signal
```

Write an m-function for modulator `mod(t, s, B)`, implementing (2).

```
function q0t = mod(t, s, B)

Ns = len(s)  % number of symbols
l1 = ...
l2 = ...

q0t = ...

end
```

Plot $x(t)$ in time and in frequency. Make sure that the basis elements have unit energy: $\|\mathrm{sinc}(x)\|^2_{L^2(\mathbb{R})} = \left\|\sqrt{B}\,\mathrm{sinc}(Bx)\right\|^2_{L^2(\mathbb{R})} = 1$.

□

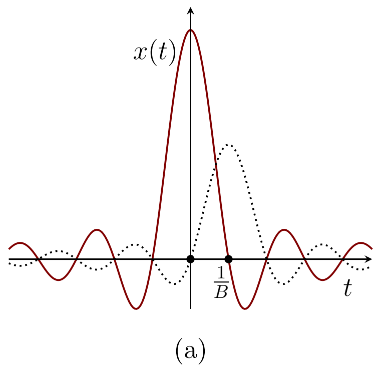
2.2 Channel Model

2.2.1 Dispersive Additive White Gaussian Noise Channel

Communication media are frequently modeled by the additive white Gaussian noise (AWGN) channel [1–3]. The baseband model is

$$y(t) = h(t) * x(t) + n(t), \tag{4}$$

where $x(t)$ is input, $y(t)$ is output, $h(t)$ is the channel impulse response, $n(t)$ is band-limited white circular symmetric Gaussian noise, and $*$ denotes convolution. The functions are all from $\mathbb{R}$ to $\mathbb{C}$. The channel filter $h(t)$ introduces a distortion called the inter-symbol interference (ISI), that must be cancelled via equalization.



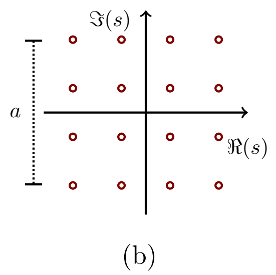


Fig. 3: (a) Modulation using sinc functions; (b) a 16-QAM constellation.

The communication theory of the AWGN channels, such as the design of the optimal receiver, is well developed [2]. As a result, we consider the optical fiber channel, for which the model is nonlinear and rather intractable. The application of deep learning may help find a good TX and RX for this complex channel.

2.2.2 Fiber-Optic Channel

Let $Q(\tau,\ell) : \mathbb{R}\times\mathbb{R}^+ \mapsto \mathbb{C}$ be the complex envelope of the signal propagating in fiber as a function of time τ and distance ℓ. Pulse propagation in optical fiber is modeled by the stochastic nonlinear Schrödinger (NLS) equation [4, Eq. 7]

$$\frac{\partial Q(\tau,\ell)}{\partial \ell} = -\underbrace{\frac{j\beta_2}{2}\frac{\partial^2 Q(\tau,\ell)}{\partial \tau^2}}_{\text{dispersion}} + \underbrace{j\gamma|Q(\tau,\ell)|^2 Q(\tau,\ell)}_{\text{nonlinearity}} + \underbrace{N(\tau,\ell)}_{\text{noise}}, \qquad (5)$$

where β_2 is the dispersion coefficient, γ is the nonlinearity parameter and $j = \sqrt{-1}$. Further, $N(\tau,\ell)$ is bandlimited circular symmetric complex Gaussian noise with auto-correlation

$$\mathsf{E}\left\{N(\tau,\ell)N^*(\tau',\ell')\right\} = \sigma_0^2\delta_B(\tau-\tau')\delta(\ell-\ell'),$$

where $\delta(x)$ is the Dirac Delta function, $\delta_B(x) = B\,\text{sinc}(Bx)$, $\text{sinc}(x) = \sin(\pi x)/(\pi x)$, B is the noise bandwidth, σ_0^2 is the in-band noise power spectral density (PSD) and E is the expected value. The transmitter is located at $\ell = 0$ and the receiver at $\ell = \mathcal{L}$.
The stochastic NLS equation (5) models the chromatic dispersion (responsible for temporal broadening), Kerr nonlinearity (responsible for spectral broadening) and noise. It is assumed that the fiber loss is perfectly compensated by distributed amplification.

Question 7. The variables Q, τ and ℓ are measured in $\sqrt{\text{Watt}}$ ($\sqrt{W}$), second (s) and meter (or kilometer km), respectively. Find out the units of β_2, γ and σ_0^2. □

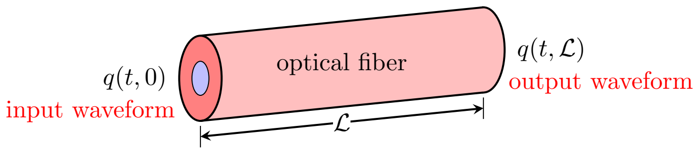


Fig. 4: Fiber channel.

The normalized NLS equation. We will find it convenient to normalize the NLS equation.

Question 8. Consider the change of variables

$$q = \frac{Q}{\sqrt{P_0}}, \qquad t = \frac{\tau}{T_0}, \qquad z = \frac{l}{\mathcal{L}}.$$

Show that if we choose

$$P_0 = \frac{2}{\gamma\mathcal{L}}, \qquad T_0 = \sqrt{\frac{|\beta_2|\mathcal{L}}{2}}, \qquad L = \mathcal{L}, \tag{6}$$

we obtain from (5)

$$\boxed{j\frac{\partial q(t,z)}{\partial z} = \frac{\partial^2 q(t,z)}{\partial t^2} + 2|q(t,z)|^2 q(t,z) + n(t,z), \quad (7)}$$

where $0 \leqslant z \leqslant 1$, and we assumed $\beta_2 < 0$. Here, $n(t,z)$ is circular symmetric complex Gaussian noise with auto-correlation

$$\mathsf{E}\left\{n(t,z)n^*(t',z')\right\} = \sigma^2 \delta_{B_n}(t-t')\delta(z-z'), \tag{8}$$

in which

$$\boxed{\sigma^2 = \frac{\sigma_0^2\mathcal{L}}{P_0T_0}}, \tag{9}$$

and $B_n = BT_0$ is the normalized bandwidth. □

In what follows, we consider the normalized NLS equation (7) with noise PSD (9). The normalized model (7) contains no parameter other than σ^2 in (9) which is computed in the following question.

Signal and system parameters We consider a fiber with parameters in Tab. 1, $\mathcal{L} = 1000$ km and $B = 1$ to 10 GHz. We assume a quadrature amplitude modulation (QAM) constellation $\mathcal{C}$ with $M = 16$ points; see Fig. 3(b).

Question 9. Enter the constants in Tab. 1 in `parameters.m`
Convert the dispersion parameter D with unit ps/(nm − km) to dispersion coefficient β_2, using $\beta_2 = -\frac{\lambda_0^2}{2\pi c}D$, where $c = 3\times10^8$ m/s is the speed of light. Convert

the loss coefficient α from unit dB/km to m^{-1}, using $\alpha = 10^{-4}\log(10)a_{\mathrm{dB}}$ in m^{-1}; see Appendix A.
Calculate the scale parameters in (6).
Calculate the noise PSD $\sigma_0^2 = n_{sp}h\alpha f_0$, where $f_0 = c/\lambda_0$ is the carrier frequency, and its normalized value σ^2 (9).
Enter the constants, scale factors, physical and normalized variables in `parameters.m` as follows. The values should be in the International System of Units.

```
% variables
L = ...       % distance
B = ...       % bandwidth

% physical constants
a_dB = ...    % power loss in dB
D = ...       % dispersion ps/(nm-km)
gama = ...    % nonlinearity coefficient
nsp = ...     % a constant factor
h = ...       % Planck constant
lambda0 = ... % center wavelength
f0 = ...      % center frequency

alpha =       % loss coefficient
beta2 = ...   % dispersion coefficient

% scale factors
L0 = ...
T0 = ...
P0 = ...

% noise PSD
sigma02 =     % physical
sigma2 =      % normalized
```

□

Table 1: Fiber and Noise Parameters

a_{dB}	0.2 dB/km	fiber loss
D	17 ps/(nm-km)	chromatic dispersion
γ	1.27 $\mathrm{W}^{-1}\mathrm{km}^{-1}$	nonlinearity parameter
n_{sp}	1	spontaneous emission factor
h	$6.626 \times 10^{-34}\mathrm{J}\cdot\mathrm{s}$	Planck's constant
λ_0	1.55 μm	carrier wavelength

2.2.3 Fiber-optic Channel in the Absence of the Nonlinearity

Consider first the NLS equation (7) in the absence of nonlinearity

$$j\partial_z q(t,z) = \partial_{tt} q(t,z) + n(t,z). \tag{10}$$

Let us simplify the partial differential equation channel (10) to the more familiar form (4).
Define the Fourier transform with respect to t with convention:

$$\hat{q}(\omega,z) = \mathcal{F}(q)(\omega) = \int\limits_{-\infty}^{\infty} q(t,z)e^{-j\omega t}\mathrm{d}t. \tag{11}$$

Solve (10) in the presence of noise and show that (10) is reduced to the AWGN channel

$$\hat{q}(\omega,z) = \hat{h}(\omega,z)\hat{q}(\omega,0) + \hat{z}(\omega), \tag{12}$$

where $\hat{q}(\omega,z)$ is the Fourier transform of $q(t,z)$, and the channel transfer function is

$$\hat{h}(\omega,z) = e^{j\omega^2 z}.$$

Question 10. Characterize the stochastic process $\hat{z}(\omega)$, *i.e.*, determine the PDF of $\hat{z}(\omega)$, its mean $\mathsf{E}\hat{z}(\omega)$ and the correlation function $\mathsf{E}(\hat{z}(\omega)\hat{z}^*(\omega'))$. □

In the time domain we have

$$q(t,z) = h(t,z) * q(t,0) + z(t), \tag{13}$$

where $*$ is convolution.

Question 11. Determine the channel impulse response $h(t,z)$. Determine the PDF of $z(t)$, and its mean and correlation. □

2.2.4 Continuous- and Discrete-time Models

The continuous-time model (5) should be discretized to a discrete-time model for information-theoretic analysis, and implementation. We need to discretize the signal, channel (time and space) and the constraints. To illustrate the process, in this subsection, we discretize and implement the linear channel (13) and (12).
The signal $q(t,z)$ can be sampled in time at the Nyquist rate $1/B$ to obtain a vector

$$\mathbf{q}(z) = \begin{pmatrix} q_1(z) & q_2(z) & \ldots & q_N(z) \end{pmatrix}.$$

However, to discretize the derivative operator ∂_{tt} in (10)) (or the integral that underlies the convolution in (13)), we sample the signal at a sufficiently small simulation step size Δt, which can be much less than the Nyquist rate.

Question 12. Discretize the signal power (3), by discretizing the integral into a Riemann sum, proving that

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \mathsf{E}|q_i(0)|^2 = \mathcal{P}. \tag{14}$$

□

Question 13. The channel can be discretized in the frequency domain by sampling (12) on a frequency mesh obtained from the time mesh.
Discretize the Fourier integrals

$$\hat{x}(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\mathrm{d}t, \quad x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{x}(\omega)e^{j\omega t}\mathrm{d}\omega,$$

on the time-frequency mesh introduced above in the main.m file, and express these Fourier integrals in terms of the discrete Fourier transforms (DFTs)

$$\hat{x}_k = \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi nk}{N}t}, \quad x_n = \frac{1}{N}\sum_{n=0}^{N-1} \hat{x}_k e^{j\frac{2\pi nk}{N}t}.$$

Use this result to approximate $\hat{q}(\omega, z)$ in terms of $\hat{\mathbf{q}}(z) = \mathrm{DFT}(\mathbf{q}(z))$.
To summarize, write down the channel model in the frequency $\hat{\mathbf{q}}(0) \mapsto \hat{\mathbf{q}}(z)$ and in time $\mathbf{q}(0) \mapsto \mathbf{q}(z)$, and their input constraints.

□

Question 14. Write an m-function `channel(t, q0t, z, sigma2, B)` that outputs a realization of the stochastic process $q(t, z)$ given $q(t, 0)$.
You need to discretize noise $n(t, z)$ to a vector $\mathbf{n}(z)$, and (8) to a discrete autocorrelation function $\mathsf{E}(\mathbf{n}(z)\mathbf{n}^H(z))$. Hint: equate the total noise power in the continuous and discrete models.

```
function [qzt, qzf] = channel(t, q0t, z, sigma2, B)

a = sigma2*B*z % total noise power in B Hz and distance [0, z]
f = ...        % get the f vector from t vector

q0f = ...       % input in frequency
qzf = q0f.* ... % output in frequency

% add Guassian noise in frequency, with correct variance
qzf = qzf + ...

qzt = ... % back to time

end
```

□

2.3 Receiver

At the receiver, first the I and Q components of the analogue optical signal are converted to two digital signals using the analogue-to-digital converters (ADCs). The resulting two discrete-time signals are converted to a complex signal, and passed to a digital signal processing (DSP) chain described below.

2.3.1 Linear Equalization

Equalization is the task of inverting the channel transformation in the absence of noise. For the dispersive channel (13), this can be done in the frequency domain

$$\begin{aligned} \hat{q}_e(\omega, \mathcal{L}) &\triangleq \hat{h}^{-1}(\omega, \mathcal{L})\hat{q}(\omega, \mathcal{L}) \\ &= \hat{q}(\omega, 0) + \hat{h}^{-1}(\omega, \mathcal{L})\hat{z}(\omega) \\ &\triangleq \hat{q}(\omega, 0) + \hat{w}(\omega), \end{aligned} \tag{15}$$

where $\hat{w}(\omega) = \hat{h}^{-1}(\omega, \mathcal{L})\hat{z}(\omega)$. In the time domain

$$q_e(t, \mathcal{L}) = q(t, 0) + w(t), \tag{16}$$

where, *e.g.*, $q_e(t, \mathcal{L}) = \mathcal{F}^{-1}(\hat{q}_e(\omega, \mathcal{L}))$. Linear equalization is performed in the frequency domain, applying (15) upon discretization.

Question 15. Characterize the stochastic processes $w(t)$ and $\hat{w}(\omega)$, namely, state their PDFs and the statistical properties (the mean and the auto-correlation function).

□

Question 16. Write an m-function `equalize(t, qzt, z)` that outputs $q(t, 0)$ given $q(t, z)$ in the absence of noise.

```
function [qzte, qzfe] = equalize(t, qzt, z)

f = ...        % get the f vector from t vector

qzf = ...      % input in frequency
qzfe = qzf.* ... % output in frequency


qzte = ... % back to time

end
```

Let us test the code so far. Add the following to the main.m file.

```
T = ...
N = ...
dt = ...
t = ...
```

```
A   = 1;
q0t = A*exp(-t.^2); % Gaussian input, for testing
q0f = ... % input in frequency
% plot below the input in t & f. You must tune T and N!
...
...

% propagation
[qzt, qzf] = channel(t, q0t, z, 0, 0); % output in t,f. Zero noise.
plot(t, q0t, ...)        % plot input output in t; tune T and N
    accordingly
plot(f, abs(q0f), ...)   % plot input output in f; tune T and N
    accordingly

% equalization and comparison
[qzte, qzfe] = equalize(t, qzt, z);     % equalized output
plot(t, q0t, ...) % plot input & equalized output in t
compare(q0t, qzte) % you should write the function compare
```

□

The equalization for the nonlinear channel (5) will be performed in Section 4 using a NN.

2.3.2 Demodulation

The demodulation consists of obtaining the received symbols at the output via projection

$$\begin{aligned}\hat{s}_\ell &= \frac{\langle q_e(t,\mathcal{L}), \operatorname{sinc}(Bt-\ell)\rangle}{\|\operatorname{sinc}(Bt-\ell)\|^2} \\ &= B\int_{-\infty}^{\infty} q_e(t,\mathcal{L})\operatorname{sinc}(Bt-\ell)\mathrm{d}t. \end{aligned} \tag{17}$$

Question 17. Write an m-function `shat = demod(t, qzte, B, Ns)` to compute $\hat{s}$ from $q_e(t,\mathcal{L})$, by implementing (17). Make sure that the basis elements have unit energy: $\|\operatorname{sinc}(x)\|^2_{L^2(\mathbb{R})} = \left\|\sqrt{B}\operatorname{sinc}(Bx)\right\|^2_{L^2(\mathbb{R})} = 1$.

```
function shat = demod(t, qzte, B, Ns)

shat = ...

end
```

□

2.3.3 Detection.

The optimal receiver is the *maximum likelihood (ML) detection* applied to the joint conditional PDF $p(\hat{s}^{n_s}|s^{n_s})$, *i.e.*,

$$\tilde{s}^{n_s} = \arg\max_{s^{n_s}\in\mathcal{C}^{n_s}} \quad p(\hat{s}^{n_s}|s^{n_s}). \tag{18}$$

Question 18. Substitute (16) into (17) and show that the continuous-time model $q(t,0) \mapsto q(t,z)$ is discretized to the discrete-time model $s^{n_s} \mapsto \hat{s}^{n_s}$, described by

$$\hat{s}_\ell = s_\ell + Z_\ell, \quad \ell = 1, 2, \dots,$$

where Z_ℓ i.i.d. $\sim \mathcal{N}_{\mathbb{C}}(0, c)$. Determine c.
Conclude that the channel $s^{n_s} \mapsto \hat{s}^{n_s}$ is memoryless, *i.e.*,

$$p(\hat{s}^{n_s}|s^{n_s}) = \prod_{\ell=1}^{n_s} p(\hat{s}_\ell|s_\ell). \tag{19}$$

Therefore the maximization in (20) can be performed separately per component $p(\hat{s}_\ell|s_\ell)$

$$\begin{aligned}\tilde{s}_\ell &= \arg\max_{s_\ell\in\mathcal{C}} \quad p(\hat{s}_\ell|s_\ell)\\ &= \arg\min_{s_\ell\in\mathcal{C}} \quad |\hat{s}_\ell - s_\ell|^2,\end{aligned} \tag{20}$$

where we used

$$p(\hat{s}_\ell|s_\ell) = \frac{1}{2\pi c}\exp\left(-\frac{1}{2c}|\hat{s}_\ell - s_\ell|^2\right). \tag{21}$$

We thus obtain that, for the AWGN channel, the ML detector is given by the minimum distance detector (20).
Find decision regions and simplify the optimal receiver.
Write an m-function `stilde = detector(shat, cnt)` that implements the ML detection. The function takes the received-equalized symbols $\hat{s}^{n_s}$ and outputs the ML estimates $\tilde{s}^{n_s}$.

```
function stilde = detector(shat, cnt)

stilde = ...

end
```

□

2.3.4 Demapping.

This step is symbols-to-bits mapping, that you should implement in `bhat = symb_to_bit(stilde, cnt)` where cnt is the constellation. Use a look-up table for mapping and demapping.

Question 19. Check first the end-to-end implementation when noise is zero. In this case, you should obtain $\hat{\mathbf{s}} \approx \mathbf{s}$ with high accuracy.
Complete the main.m file.

```
% modulation
nb = 64; % number of bits
M = 16; % order of modulation
ns = ... % number of symbols
b= ... % random bit sequence
s = bit_to_symb(b, cnt);
q0t = mod(t, s, B);

% propagation & equalization. Set the noise to zero for now
[qzt, qzf] = channel(t, q0t, z, sigma2, B); % output in t,f
[qzte, qzfe] = equalize(t, qzt, z);     % equalized output

% demodulation
shat = demod(t, qzte, B);

% detection
stilde = detector(shat, cnt);
bhat = symb_to_bit(stilde, cnt);
```

Do you obtain $\tilde{\mathbf{s}} = \mathbf{s}$ and $\hat{\mathbf{b}} = \mathbf{b}$ in the absence of noise?

□

Inter-symbol interference. One type of distortion in communication is *inter-symbol interference (ISI)*, explained in the class.

Question 20. Simulate the output for the input

$$q(t,0) = A_1 e^{-\frac{(t+t_0)^2}{2D^2}} + A_2 e^{-\frac{(t-t_0)^2}{2D^2}},$$

where $A_1 = 1$, $A_2 = 2$, $D = 1$ and $t_0 = 4$. Plot $q(t,0)$ and $q(t,1)$ (without equalization) in the same graph.
Does dispersion give rise to ISI in time (*i.e.*, the two parts in $q(t,0)$ that are non-overlapping at $z = 0$ overlap at $z = 1$)? Repeat for $t_0 = 7$. Is ISI completely canceled with equalization?
Does dispersion give rise to interaction between frequency components? Which domain is preferred for modulation and equalization?

□

2.3.5 Performance evaluation

The performance of the receiver is measured in terms of bit error rate (BER)

$$\text{BER} = \text{Prob}(\hat{b}_i \neq b_i) = \frac{1}{N_b}\Big|\Big\{i : \hat{b}_i \neq b_i\Big\}\Big|.$$

Equivalently, the Q-factor may be used

$$\text{Q-factor} = 20\log_{10}\left(\sqrt{2}\text{InvErfc}(2\text{BER})\right) \quad \text{dB},$$

where InvErfc is the inverse complementary error function.
The BER is computed as a function of the signal-to-noise ratio (SNR) at TX

$$\begin{aligned}\text{SNR} &= \frac{\mathcal{P}}{P_n}\\ &= \frac{\mathcal{P}}{\sigma_0^2 B\mathcal{L}},\end{aligned}$$

where $\mathcal{P}$ is the signal power (1) and $P_n = \sigma_0^2 B\mathcal{L}$ is the total noise power (from $z = 0$ to $z = \mathcal{L}$). The SNR can vary by scaling the constellation radius a. The numerator and denominator in SNR can be calculated with the normalized or non-normalized variables.

Back-to-back setup. Make sure that you obtain $\hat{s}^{n_s} = s^{n_s}$ and $\hat{b}^N = b^N$ in the back-to-back setup where the channel is $q(t,\mathcal{L}) = q(t,0)$. Also, make sure that you obtain zero error in the noise-free channel with equalization.

Noisy channel. We now turn to the noisy channel.

Question 21. *(Simulated BER).* Add noise to the channel and compute the BER (or the symbol error rate). Complete the main.m file.

```
a = 1; % constellation radius
M = 2; % the number of points in the constellation
ser = ser(s,shat); % symbol error rate
ber = ber(b, bhat); % bit error rate

% plot the constellation at TX and RX. Change the annotation
plot(real(cnt), imag(cnt), '*', real(shat), imag(shat), '.');
% plot the BER
plot(snr, ber)
```

You can choose the number of symbols $n = 10000$, run the code $N_r = 1$ time, and count the errors. It is, however, easier to choose, say, $n_s = 32$ symbols and run the same code $N_r = 312$ times in parallel over 32 CPU cores. You can change n_s and N_r as long as the graphs are insensitive to the values of these parameters.
The final results of this section are: (i) constellation plot at TX and RX; (ii) the plot of the BER as a function of SNR for 16-QAM.

□

3 Generative neural network

We now consider the NLS equation (7) with dispersion, nonlinearity and noise. We solve (7) numerically using an algorithm called *split-step Fourier method*

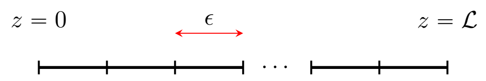


Fig. 5: Discretization of the channel in distance.

(SSFM) [4]. The SSFM is a generative deep NN described below [5].
Consider a model described by the partial differential equation (PDE):

$$\frac{\partial q(t,z)}{\partial z} = \underbrace{L_L(q)}_{\text{linear}} + \underbrace{L_N(q)}_{\text{nonlinear}} + \underbrace{n(t,z)}_{\text{noise}}, \tag{22}$$

where L_L and L_N are linear and nonlinear operators. In the stochastic NLS equation (7),

$$L_L(q) = -j\frac{\partial^2 q}{\partial t^2}, \quad L_N(q) = -2j|q|^2 q. \tag{23}$$

Discretize the distance $\mathcal{L}$ into a large number $M \to \infty$ of segments of length $\epsilon = \mathcal{L}/M$. Discretize the time interval $[-T/2, T/2]$ into a large number $n \to \infty$ of intervals of step size $\mu = T/n$.
Divide the PDE (22) into three parts shown in (22), and perform successively the following linear, nonlinear and noise transformations in distance.

3.1 Linear transformation

The linear part of PDE (22) with (23) is

$$\frac{\partial q(t,z)}{\partial z} = -j\frac{\partial^2 q(t,z)}{\partial t^2}. \tag{24}$$

Taking the Fourier transform of both sides of (24), we obtain

$$\frac{\partial \hat{q}(\omega,z)}{\partial z} = j\omega^2 \hat{q}(\omega,z).$$

The solution is

$$\hat{q}(\omega,z) = e^{j\omega^2 z}\hat{q}(\omega,0). \tag{25}$$

In the time domain

$$q(t,z) = h(t,z) * q(t,0),$$

where

$$h(t,z) = \frac{1}{\sqrt{-j4\pi z}} e^{-j\frac{t^2}{4z}t^2}, \tag{26}$$

in which we used $e^{-\frac{t^2}{2\lambda}} \leftrightarrow \sqrt{2\pi\lambda}e^{-\frac{\lambda\omega^2}{2}}$.

It follows that dispersion is a simple phase change (all-pass filter) in the frequency domain. In the time domain, dispersion is a convolution, giving rise to memory and pulse broadening.
Let the input and output of the linear transformation in a given layer be $\mathbf{X} \in \mathbb{C}^n$ and $\mathbf{V} \in \mathbb{C}^n$, respectively. Discretizing (25), we obtain

$$\boxed{\mathbf{V} = W\mathbf{X}. \quad (27)}$$

Here,

$$W = D^H \Gamma D, \tag{28}$$

where D is the discrete Fourier transform (DFT) matrix at n frequency points $(\omega_i)_{i=1}^n$, and $\Gamma = \operatorname{diag}(\hat{\mathbf{h}})$, in which $\hat{\mathbf{h}} = [\hat{h}_1, \cdots, \hat{h}_n]$, $\hat{h}_i = e^{j\epsilon\omega_i^2}$. It follows that linear transformation is multiplication by a (convolutional) *weight matrix* $W \in \mathbb{C}^{n\times n}$.
Let

$$\mathbf{h} = \operatorname{IDFT}(\hat{\mathbf{h}}). \tag{29}$$

Linear step is convolution of the input with the dispersion filter $\mathbf{h}$. The weight matrix is W, which is Toeplitz with the first row $\mathbf{h}$.

Remark 3. In the generative NN W and $\hat{\mathbf{h}}$ are fixed and not learned (given by (28) and (29)). In inference NN $\mathbf{h}$ must be learned. □

3.2 Nonlinear transformation

The nonlinear part of PDE (22) with (23) is

$$\frac{\partial q(t,z)}{\partial z} = -2j|q(t,z)|^2 q(t,z).$$

It can be verified that the solution in the time domain is

$$q(t,z) = e^{-2j|q(t,0)|^2 z} q(t,0). \tag{30}$$

It follows that nonlinearity is a signal-dependent phase change in the time domain. In the frequency domain, nonlinearity gives rise to a convolution and spectral broadening.
Let the input and output of the nonlinear step in a given layer be $\mathbf{V} \in \mathbb{C}^n$ and $\mathbf{U} \in \mathbb{C}^n$, respectively. Discretizing (30)

$$\boxed{U_i = \sigma(V_i),}$$

```
input  : $\mathbf{X} \in \mathbb{C}^n$, number of layers $M$, and parameters $\epsilon$, $P_n$.
output: $\mathbf{Y} \in \mathbb{C}^n$
for $l = 1$ to $M$ do
    $\mathbf{V} = W\mathbf{X}$, where the weight matrix $W$ is given by (28).
    $\mathbf{U} = \sigma(\mathbf{V})$, where the activation function is given by (31).
    $\mathbf{Y} = \mathbf{V} + \mathbf{N}$, where $\mathbf{N} \sim \mathcal{N}_{\mathbb{C}}(0, P_n I_n)$
    $\mathbf{X} = \mathbf{Y}$
end
return $\mathbf{Y}$
```

Algorithm 1: Generative fully-connected feedforward neural net.

where $i = 1, 2, \ldots, n$, and $\sigma : \mathbb{C} \mapsto \mathbb{C}$ is the activation function

$$\sigma(x) = xe^{-2j\epsilon|x|^2}. \tag{31}$$

Note that the nonlinearity is memoryless, *i.e.*, it acts component wise.

Noise addition. This is a simple noise addition

$$\mathbf{Y} = \mathbf{U} + \mathbf{N}, \quad \mathbf{N} \sim N(0, P_n I_n),$$

where $P_n = \sigma^2 B_n \epsilon$ is the noise power.

Question 22. Write a function that generates a realization of a random vector in $\mathbf{Z} \in \mathbb{C}^n$ drawn from $\mathcal{N}_{\mathbb{C}}\big(0, \sigma^2 I_n\big)$.

```
function Z = noise(n, sigma2)

Z = ...       % you need real and imaginary components
end
```

□

Remark 4. In NNs, one performs successive linear and nonlinear transformations, and there is no noise step. In the generative neural net there is a noise step, so that the data set comes from a probability distribution. There will not be a noise step in the neural net designed for inference. □

Question 23. Write an m-function `nnet_gen(x, nz, params)` that outputs **y** given **x**, where nz is the number of layers and `params` is a structure or class containing the required parameters.
The code is an implementation of the Algorithm (1). We first write a code for a feed-forward neural net that may not be very efficient. Later we shall discuss efficient numerical implementation in the context of the deep learning.

```
function y = nnet_gen(x, nz=500, params)

% pre-compute the fixed weight matrix W
```

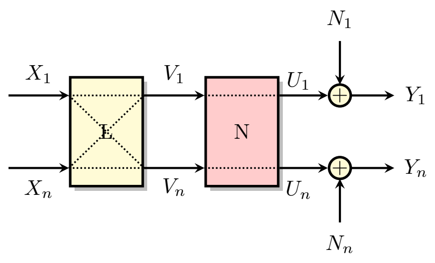


Fig. 6: One layer of the generative NN consisting of a linear, a nonlinear and a noise transformation.

```
n = len(x)
f = ...                 % frequency vector
w = 2*pi*f              % angular frequency vector

z = params. ...
dz = z/nz               % epsilon
h = exp(j*w.^2*dz);     % all-pass filter
D =                     % DFT matrix of size n
W = conj(D') * ...

% Loop over the nnet layers
for k=1:nz

    % linear transformation -- multiplication by the weight matrix W
    x = W*x

    % activation function
    x = sigma(x, dz)

    % noise addition
    x = x + ...

end

y = x

end
```

4 Neural Network Equalizer

4.1 Data set generation

The predictive NN takes samples of the signal at the output of the channel, and outputs either a prediction of the transmitted signal or symbols (regression

problem), or bits (classification problem).
Consider a signal-to-symbol NN. Let $\mathbf{s}^{(i)} = (s_1^{(i)}, s_2^{(i)}, \cdots, s_{n_s}^{(i)})$ denote the sequence of symbols at TX in the transmission block i, and $\mathbf{y}^{(i)}$ the corresponding received vector. The generative NN can be used to obtain a training data set

$$S_n = \big\{(\mathbf{y}^{(i)}, \mathbf{s}^{(i)}) : \quad i = 1, 2, \cdots, n\big\}.$$

Question 24. Generate a training data set of total $50,000$–$400,000$ symbols. □

The NN equalizers in communications can be classified into two categories. In *model-based equalizers*, the architecture is based on the parameterization of the channel model. An example is a parameterization of the SSFM, and learning the parameters with the gradient-based optimization.
On the other hand, in *model-agnostic equalizers*, the architecture is independent of the channel model. The model-agnostic schemes do not require the channel state information, such as the fiber parameters. Here, the NNs can be placed at the end of the linear equalizer for the nonlinearity mitigation, or right after the channel for compensating both the linear and nonlinear distortions (thereby constituting most of the receiver DSP).
A number of NN architectures have been proposed for the nonlinearity mitigation. Fully-connected (FC) or dense NNs with 2 or 3 layers have been applied, few hundred neurons per layer, and tanh activation. The overfitting and complexity become problems when the models get bigger. The convolutional NNs can model the linear time-invariant (LTI) systems with a finite impulse response. The application of the convolutional networks for compensating the nonlinear distortions is investigated in the literature, showing that one-dimensional convolution can well compensate the dispersion. The bi-directional recurrent and long-short term memory networks (LSTM) receivers are shown to perform well in fiber-optic equalization. Compared to the convolutional and dense networks, BiLSTM networks better model LTI systems with infinite impulse response, such as the response of the dispersion (26).

4.2 Convolutional-FC Equalizer

The communication signals in baseband are complex-valued. To apply a NN with real parameters, the vector $\mathbf{y}$ is decomposed into two sequences $\Re(\mathbf{y})$ and $\Im(\mathbf{y})$, which are passed to the NN. We consider a many-to-many architecture, where the NN equalizes all complex symbols given n_i input samples. The inputs of the network are two vectors, each containing a window of $n_i = M + 1$ consecutive elements from each of the two input sequences, where M is the channel memory. The network outputs a vector of $n_o = 2n_s$ real numbers, corresponding to the real and imaginary parts of the symbols.
The Conv-FC model is a cascade of a complex-valued convolutional layer, a FC hidden layer, and a FC output layer. The first layer implements the discrete convolution of $\mathbf{y} \in \mathbb{C}^K$ with a kernel $\mathbf{h} \in \mathbb{C}^K$, to compensate the dispersion, where K is the number of kernel taps. The discrete complex convolution $\mathbf{y} * \mathbf{h}$ is

implemented using two real convolutions in terms of two filters $\Re(\mathbf{h})$ and $\Im(\mathbf{h})$, according to

$$\begin{aligned}\mathbf{y} * \mathbf{h} &= \Re(\mathbf{y}) * \Re(\mathbf{h}) - \Im(\mathbf{y}) * \Im(\mathbf{h}) \\ &\quad + j\Big\{\Re(\mathbf{y}) * \Im(\mathbf{h}) + \Im(\mathbf{y}) * \Re(\mathbf{h})\Big\}. \qquad (32)\end{aligned}$$

The first layer thus contains four parallel real-valued one-dimensional convolutions, with the stride one, "same padding," and no activation. There are total $2K$ trainable real filter taps. The four real convolutions are combined, obtaining $\Re(\mathbf{y}*\mathbf{h})$ and $\Im(\mathbf{y}*\mathbf{h})$, which are then concatenated. The resulting vector is fed to a FC hidden layer with n_h neurons, and tangent hyperbolic (tanh) activation. Finally, there is an output FC layer with 2 neurons for each complex-valued symbol, and no activation.

4.3 BiLSTM-FC Equalizer

4.4 Training

A Units

The conversion of the unit for the loss coefficient α is as follows. Let α denote the loss coefficient for signal in m^{-1}, and $a_{\mathrm{dB}}(z)$ the power loss in the dimensionless unit dB at distance z. Let $P(z)$ denote the signal power at distance z.
The power decays 0.2 dB every 1 km, *i.e.*, $a_{\mathrm{dB}}(1\mathrm{km}) = 0.2$ dB. We have

$$10\log_{10}\left(\frac{P(0)}{P(z)}\right) = a_{\mathrm{dB}}(z), \quad \text{at } z = 1\mathrm{km},$$

or $P(0)/P(z) = 10^{a_{\mathrm{dB}}(z)/10}$ at $z = 1$km. Since $P(0)/P(z) = e^{\alpha z}$, we have

$$\begin{aligned}\alpha &= \frac{\log(10^{\frac{a_{\mathrm{dB}}(z)}{10}})}{z} \\ &= \frac{\log(10)}{10z} a_{\mathrm{dB}}(z) \\ &= \frac{\log(10)}{10} a_{\mathrm{dB}} \quad \mathrm{km}^{-1} \\ &= 10^{-4}\log(10) a_{\mathrm{dB}} \quad \mathrm{m}^{-1}.\end{aligned}$$

We write $a_{\mathrm{dB}} = 0.2$ dB/km to mean $a_{\mathrm{dB}}(z = 1\mathrm{km}) = 0.2$ dB. The notation dB/km should not suggest that a_{dB} has unit m^{-1}.

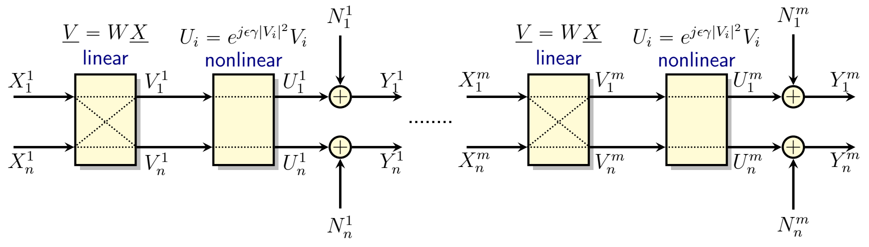


Fig. 7: The generative NN with several layers.

B The average power of a QAM signal

Consider the QAM signal

$$\begin{aligned} q(t) &= \sum_{k=-\infty}^{\infty} s_k p(t-kT_s) \\ &= \sum_{k=-\infty}^{\infty} s_k \operatorname{sinc}(Bt-k), \end{aligned} \tag{33}$$

where $p(t) = \operatorname{sinc}(Bt)$ is pulse shape, $T_s = 1/B$, $(s_k)_{k\in\mathbb{Z}}$ are i.i.d. complex random variables. We show that the average power of PAM signal (33) is the average energy of the symbols, *i.e.*, $\mathcal{P} = \mathsf{E}|s_k|^2$.
The power is energy divided by time duration. In the time duration $[-\frac{T}{2}, \frac{T}{2}]$, there are $N = T/(1/B) = TB$ pulses; let $S \triangleq \{-N/2, \cdots, N/2-1\}$ be their index set. The average power is

$$\begin{aligned} \mathcal{P} &= \lim_{T\to\infty} \frac{1}{T}\mathsf{E}\left\{\int_{-\frac{T}{2}}^{\frac{T}{2}} |q(t)|^2 \mathrm{d}\tau\right\} \\ &\overset{(a)}{=} \lim_{N\to\infty} \frac{1}{N/B} \sum_{k\in S} \mathsf{E}|s_k|^2 \left\{\lim_{T\to\infty} \int\limits_{-T/2}^{T/2} |\operatorname{sinc}(Bt-k)|^2 \mathrm{d}t\right\} \\ &\overset{(b)}{=} \lim_{N\to\infty} \frac{1}{N/B} \sum_{k\in S} \mathsf{E}|s_k|^2 \times \frac{1}{B} \\ &\overset{(c)}{=} \lim_{|S|\to\infty} \frac{1}{|S|} \sum_{k\in S} \mathsf{E}|s_k|^2 \\ &= \mathsf{E}|s_k|^2. \end{aligned}$$

Step (a) follows from $T = N/B$, and the orthogonality of $\big(\operatorname{sinc}(Bt-k)\big)_{k\in\mathbb{Z}}$. Step (b) follows from $\int\limits_{-\infty}^{\infty} |\operatorname{sinc}(Bt-k)|^2 = 1/B$ which can be computed in the frequency domain. Step (c) follows because s_k are i.i.d. random variables.

Remark 5. Instead of orthogonality, we may require s_k to be i.i.d. zero mean.

Remark 6. The basis $\big(\text{sinc}(Bt-k)\big)_{k\in\mathbb{Z}}$ used above is unit power. The version $\big(\sqrt{B}\,\text{sinc}(Bt-k)\big)_{k\in\mathbb{Z}}$ is unit energy. $\square$

References

[1] R. G. Gallager, *Principles of Digital Communication.* Cambridge, UK: Cambridge University Press, 2008.

[2] A. Lapidoth, *A Foundation in Digital Communication.* Cambridge University Press, 2017.

[3] G. D. Forney, Jr., “Principles of digital communication II.” Massachusetts Institute of Technology, Mar. 2005, MIT OpenCourseWare, Course no. 6.451, Lecture Notes.

[4] G. Kramer, M. I. Yousefi, and F. Kschischang, “Upper bound on the capacity of a cascade of nonlinear and noisy channels,” in *IEEE Info. Theory Workshop*, Jerusalem, Israel, Apr. 2015, pp. 1–4.

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.* MIT Press Cambridge, 2016.