ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

School of Computer and Communication Sciences

Handout 22 Principles of Digital Communications
Project May. 4, 2022

Channel Model

The channel takes an input vector $\mathbf{X} \in \mathbb{C}^n_{[-3/2,3/2]}$ and returns an output vector $\mathbf{Y} \in \mathbb{C}^n$, where $\mathbb{C}_{[-3/2,3/2]}$ is the set of complex number such that the real part and imaginary part is restricted to the interval $[-3/2, 3/2]$. The input and output are related as,

$$\mathbf{Y} = \exp(\Theta i)\mathbf{X} + \mathbf{Z}$$

where Θ is uniformly distributed on $[0, 2\pi]$. Note that a same Θ affects all the components of the vector $\mathbf{X}$, but Θ is re-generated each time a vector $\mathbf{X}$ is sent. The noise $\mathbf{Z}$ has i.i.d. circularly symmetric Gaussian components, namely, real and imaginary parts of Z are independent i.i.d. Gaussians with zero mean and variance $\sigma^2/2$, where $\sigma^2 = 10^{-2.65}$.

To promote efficient communication, we insist on $n \leq 100$. We also enforce a constraint on the transmitted average energy : with $s = n^{-1}\sum_{i=1}^{n} |X_i|^2$, we check if $s \leq 1$. If $s > 1$, then $\mathbf{Z}$ will be scaled by $\sqrt{s}$ to maintain the signal to noise ratio.

Assignment

a) Solve the theoretical part of the project. Please prepare one solution per team, and submit it through Moodle. The deadline for this submission is on **Wednesday June 1**.

b) Develop a system capable of reliably transmitting text messages over this channel. Specifically:

- Design a transmitter that reads a text message and returns complex-valued samples of an information-bearing signal $\mathbf{X}$.
- You send $\mathbf{X}$ to a server that applies channel effects, returning $\mathbf{Y}$.
- Having received $\mathbf{Y}$, your designed receiver must reconstruct the text message.

Submission and Evaluation Rules

- You work in teams of *three or four*.

  Please choose your teammates at latest by **Friday, May 13** and send an email to `reka.inovan@epfl.ch` in order to register your team.

- You can use any programming language, as long as all the code pertaining to the transmitter and receiver is produced by your team.

- During the last session (June 3), each group presents their project in 5–10 minutes and gives a demonstration by transmitting a file that we provide.

  (i) You will run the transmitter and the receiver on your own laptop.

(ii) You must submit your team code to Moodle before **Friday, June 4, 10am**. For each team, it is sufficient to submit through one of the member's Moodle account.

(iii) The message which you will be asked to transmit is a text which contains *exactly 78 characters*. The text is guaranteed to be ASCII-encodable, i.e., each character is a 7-bit symbol.

(iv) Your presentation should contain a brief explanation of your signaling scheme, followed by the transmission and decoding of the chosen text (that will be given to you on the spot).

(v) You will be given *two* chances for transmission. I.e., if you fail to transmit the message during your first transmission, you can repeat the transmission once more.

- The grade is based on the solution that you submitted for the theoretical part, and the reliability of your scheme during the demonstration. The theoretical part counts for 5 points of the grade and the demonstration part counts for 15 points. The demonstration part will be graded as follows,

  (a) If you manage to transmit the text message without error during one of the two transmissions, you will get the full mark (15/15 pts).

  (b) Otherwise your mark will be $\max\{7-\varepsilon, 0\}$ where ε is the number of incorrect characters in the reproduced text at the receiver. We will grade based on the best out of two transmission.

  (c) On top of that, the team which uses smallest n will get 5 additional (bonus) points.

  (d) You will be given 0 if you do not attend the presentation and your team mates cannot specify your contribution during the presentation.

Python Client

To simplify communication with the channel server, we provide you a Python script `client.py` that you can download on the course webpage. Please read the associated docstrings for more information. You can only connect to the server if you are inside EPFL network. If you are working outside of EPFL, you can use EPFL's VPN (please visit the following link[1] for more information). Please use the following connection parameters

- `--srv_hostname=iscsrv72.epfl.ch`
- `--srv_port=443`

Tips and Practical Considerations

- To avoid overloading the server, we only allow each client to connect once every 30 seconds.
- The input file for python client should be organized such that the first n lines of the input file contains the real part of the signal, and the next n lines contains the imaginary part of the signal. You are not required to use Python, but the input

[1] https://www.epfl.ch/campus/services/en/it-services/network-services/remote-intranet-access/

and the output of your code need to adhere to this format. As a reference, your transmitted signal will be serialized (i.e., saved to the file) and deserialized as follows,

```
def serialize_complex(complex_vector,file_name):
  complex_vector = complex_vector.reshape(-1)
  np.savetxt(file_name,np.concatenate([np.real(complex_vector),
             np.imag(complex_vector)]))

def deserialize_complex(file_name):
  tx_data = np.loadtxt(file_name)
  N_sample = tx_data.size
  N_sample = N_sample//2

  tx_data = np.clip(tx_data, -1.5,1.5)
  tx_data = tx_data[0:N_sample] + 1j*tx_data[N_sample:(2*N_sample)]
  return tx_data
```

- The implementation of channel models on the server is equivalent to the following function:

```
def channel(sent_signal):
  s = np.mean(sent_signal**2)
  if s <= 1:
    s = 1
  noise_power = (10**(-2.65))*s
  shift = np.exp(-2j*np.pi*np.random.rand())
  sent_signal = sent_signal*shift
  noise_std = np.sqrt(noise_power/2)
  rcv_signal = sent_signal + noise_std*np.random.randn(len(sent_signal))
               + 1j*noise_std*np.random.randn(len(sent_signal))
  return rcv_signal
```

You can use this function to test your transmission scheme locally.