ETH zürich

Exercise 9: Circulant Systems and OFDM

Christoph Studer (studer@ethz.ch)
Stefan M. Moser (moser@isi.ee.ethz.ch)

- **Exercise: Thursday, 5 December 2024**
- This exercise will be assisted in CAB G11 from 8:15 h to 10:00 h
- Problems 3 and 4 require a computer with MATLAB (or Octave) installed
- **Due date:** Monday, 16 December 2024, 08:00 h
- **Submission instructions:** Upload your solutions (answers and MATLAB code) to Moodle as a *single* zip-file with a file name `⟨last-name⟩-⟨first-name⟩-e9.zip`. Your homework should consist of a single pdf file for the written answers and one MATLAB script per problem part. The MATLAB file must be named according to the syntax `E9_P3_4.m` (for "Exercise 9 Problem 3 Part 4").
- **Rules:** You may collaborate with other students on this exercise, but each student must turn in their own assignment. You must specify the names of your collaborators! Copying answers from previous years' exercises is strictly forbidden.

Problem	**Maximum Points**	**Points Received**
1	12	
2	30	
3	25	
4	33	
Total	**100**	

Problem 1: Compute the IDFT with Input Reversal (12pts)

In class, you have seen that one can compute the inverse DFT to the vector $\hat{\boldsymbol{s}}$ with the conjugate trick as follows $\boldsymbol{s} = (\boldsymbol{F}\hat{\boldsymbol{s}}^*)^*$. We now show an alternative approach to compute the IDFT with the DFT. Let

$$\bar{s}_k = \hat{s}_{(N-k) \bmod N}, \quad k = 0, 1, \ldots, N-1, \tag{1}$$

be the DFT-domain sequence reversed. Show that $\boldsymbol{s} = \boldsymbol{F}\bar{\boldsymbol{s}}$ implements the inverse DFT $\boldsymbol{s} = \boldsymbol{F}^{\mathsf{H}}\hat{\boldsymbol{s}}$.

Problem 2: Circulant Communication System (30pts)

Consider the following circulant communication system:

$$\mathbf{y} = \boldsymbol{C}\mathbf{s} + \mathbf{z}, \tag{2}$$

where $\mathbf{y} \in \mathbb{C}^N$ is the received vector, $\boldsymbol{C} \in \mathbb{C}^{N \times N}$ is a circulant matrix that is perfectly known at the receiver, $\mathbf{s} \in \mathcal{X}^N$ is the transmit data vector where the entries are taken uniformly from the constellation set $\mathcal{X}$, and $\mathbf{z} \sim \mathcal{CN}(\mathbf{0}, N_0\boldsymbol{I}_N)$ models IID circularly symmetric complex Gaussian noise with variance N_0 per entry.

For this problem, assume that the vector $\boldsymbol{c} \in \mathbb{C}^N$ appears in the first row of the circulant matrix $\boldsymbol{C}$ and the discrete Fourier transform (DFT) matrix $\boldsymbol{F}$ is normalized so that $\boldsymbol{F}^{\mathsf{H}}\boldsymbol{F} = \boldsymbol{I}_N$.

Part 1 (10pts). Derive a compact mathematical expression for the SNR of this communication system that depends on the vector $\boldsymbol{c}$. Assume that each transmitted symbol $\mathsf{s}_k \in \mathcal{X}$, $k = 0, 1, \ldots, N-1$, is normalized so that its energy satisfies $E_\mathrm{s} = \mathbb{E}\left[|\mathsf{s}_k|^2\right]$, $k = 0, 1, \ldots, N-1$.

Part 2 (10pts). The zero-forcing (ZF) detector for this circulant communication system is given by

$$\hat{\mathbf{s}} = \boldsymbol{C}^{-1}\mathbf{y}. \tag{3}$$

Design an algorithm with complexity $\mathcal{O}(N\log(N))$ that implements the above ZF detector. The DFT with $\boldsymbol{F}$ and IDFT with $\boldsymbol{F}^{\mathsf{H}}$ can be carried out with an FFT and IFFT that require $N\log(N)$ operations, respectively. Describe the algorithm and show that the algorithm requires $\mathcal{O}(N\log(N))$ operations. *Important: The transmitter is* not *allowed to do any preprocessing.*

Part 3 (10pts). For the ZF detector, derive the statistics of the post-ZF-equalization noise vector. To this end, consider the system *after* ZF equalization

$$\boldsymbol{C}^{-1}\mathbf{y} = \mathbf{s} + \boldsymbol{C}^{-1}\mathbf{z} \tag{4}$$

and compute the statistics of the term $\boldsymbol{C}^{-1}\mathbf{z}$. Fully specify its distribution and express your results using the vector $\boldsymbol{c}$.

Hint: You can use some of the quantities from Part 2.

Problem 3: Fast Vector Multiplication and Convolution (25pts)

The diagonalization result of circulant matrices using the DFT also has direct consequences for Toeplitz systems. For example, consider the system $\mathbf{y} = \boldsymbol{H}\mathbf{s}$, where the matrix $\boldsymbol{H}$ is 3×3 and Toeplitz

$$\boldsymbol{H} = \begin{bmatrix} h_0 & h_{-1} & h_{-2} \\ h_1 & h_0 & h_{-1} \\ h_2 & h_1 & h_0 \end{bmatrix}. \tag{5}$$

One can convert this matrix into a larger circulant matrix

$$\boldsymbol{C} = \left[\begin{array}{ccc|cc} h_0 & h_{-1} & h_{-2} & h_2 & h_1 \\ h_1 & h_0 & h_{-1} & h_{-2} & h_2 \\ h_2 & h_1 & h_0 & h_{-1} & h_{-2} \\ \hline h_{-2} & h_2 & h_1 & h_0 & h_{-1} \\ h_{-1} & h_{-2} & h_2 & h_1 & h_0 \end{array}\right], \tag{6}$$

where the upper-left part is the original Toeplitz matrix $\boldsymbol{H}$.

Part 1 (10pts). Fast Vector Multiplication with Toeplitz Matrix.

Show that given a square Toeplitz matrix $\boldsymbol{H} \in \mathbb{C}^{N\times N}$ and a vector $\boldsymbol{s} \in \mathbb{C}^N$ one can compute $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{s}$ using an $\mathcal{O}\big(N\log(N)\big)$ algorithm. Be precise when describing your algorithm.

Part 2 (15pts). Fast Convolution with the I/DFT.

We have shown in class that one can carry out a *cyclic* convolution $\boldsymbol{y} = \boldsymbol{h} \star \boldsymbol{s}$ in the DFT domain in only $\mathcal{O}\big(N\log(N)\big)$ time. Assume that you want to implement a *regular* convolution of an impulse response $\boldsymbol{h} \in \mathbb{C}^L$ of fixed length L with the input signal $\boldsymbol{s} \in \mathbb{C}^N$ so that $\boldsymbol{y} = \boldsymbol{h} * \boldsymbol{s}$, where $\boldsymbol{y} \in \mathbb{C}^{N+L-1}$.

Develop an $\mathcal{O}\big(N\log(N)\big)$ algorithm in MATLAB to carry out this discrete-time convolution operation. Include the MATLAB source code along with an explanation of why your algorithm (i) implements a regular convolution and (ii) why it requires only $\mathcal{O}\big(N\log(N)\big)$ time. ***Do not forget to include proof that your code matches that of MATLAB's convolution function*** `conv` ***up to numerical precision.***

Problem 4: OFDM Channel Estimation (33pts)

Consider OFDM transmission via the ISI AWGN channel

$$\mathbf{y} = \boldsymbol{h} * \mathbf{s} + \mathbf{z}, \tag{7}$$

where $\boldsymbol{h}$ describes the channel impulse response to be estimated at the receiver in the presence of IID complex AWGN $\mathrm{z}[k] \sim \mathcal{CN}(0, N_0)$. To enable channel estimation, the transmitter sends an OFDM modulated pilot signal $\boldsymbol{s}$, which represents predefined unit-energy QPSK symbols on $W_{\text{SC}} = 64$ subcarriers, which can be generated at random, but are known to the receiver. The cyclic prefix length is $L_{\text{CP}} = 16$ symbols.

For this simulation, we assume an 8-tap channel impulse response $\boldsymbol{h}$, given by

$$h[k] = \begin{cases} (-0.6 + 0.2i)^k & \text{if } k \in \{0, \dots, 7\}, \\ 0 & \text{else.} \end{cases} \tag{8}$$

Part 1 (8pts). Compute the SNR for this ISI channel with OFDM transmission, which we define as

$$SNR = \frac{\mathbb{E}\left[\|\boldsymbol{h} \star \bar{\mathbf{s}}\|_2^2\right]}{\mathbb{E}\left[\|\bar{\mathbf{z}}\|_2^2\right]}, \tag{9}$$

with the cyclic prefix-free transmit vector $\bar{\mathbf{s}}$, the corresponding noise samples $\bar{\mathbf{z}}$, and the *cyclic* convolution $\star$. Thereby, this definition does not depend on the length of the cyclic prefix, which is anyways discarded by the receiver.

Part 2 (15pts). Write a MATLAB script that implements this OFDM-based channel estimation scheme. Evaluate the estimation error in the OFDM (frequency) domain and plot the mean squared error (MSE) averaged over all subcarriers and Monte–Carlo trials versus the SNR.

Part 3 (10pts). Also with MATLAB, evaluate the peak-to-average-power ratio (PAR) of $\mathbf{s}$ originating from random QPSK pilots and compare with all-ones (i.e., $\hat{\boldsymbol{s}} = \mathbf{1}$) and one-hot pilots (i.e., $\hat{\boldsymbol{s}}$ is a unit-vector). Here, we include the cyclic prefix in the definition of the PAR (which differs to the conventional PAR definition used in Lecture 12):

$$PAR = \frac{\|\boldsymbol{s}\|_\infty^2}{\frac{1}{W_{\text{SC}}+L_{\text{CP}}}\|\boldsymbol{s}\|_2^2} = \frac{(W_{\text{SC}} + L_{\text{CP}})\|\boldsymbol{s}\|_\infty^2}{\|\boldsymbol{s}\|_2^2}. \tag{10}$$

Which pilot pattern would you implement in a practical communication system?