Technische Universität München

MIMO Systems Programming Project 2

Instructions

Donia Ben Amor, M.Sc.
Dr.-Ing. Michael Joham

WS 2022/23

Fakultät für Elektrotechnik und Informationstechnik
Professur für Methoden der Signalverarbeitung
Univ.-Prof. Dr.-Ing. Wolfgang Utschick

Professur für Methoden der Signalverarbeitung
Technische Universität München
80290 München

http://www.msv.ei.tum.de

Contents

Chapter 2

Multiple Access Channel

In this project, the *multiple access channel* (MAC) is analyzed. We will first consider the achievable rate regions, followed by a discussion on achievable sum rates, and finally the capacity region of the MIMO MAC is computed.

The Gaussian MIMO MAC is depicted in Fig. 2.1. A set of K transmitters simultaneously accesses the physical resource to transmit data to a single receiver.

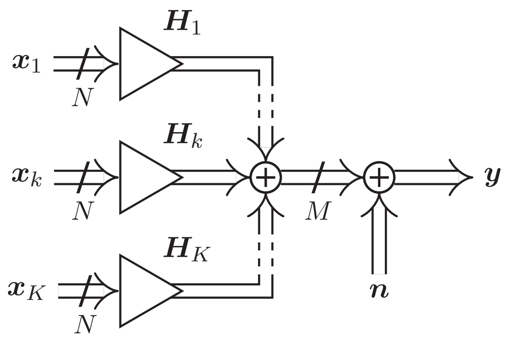

Figure 2.1: MIMO Multiple Access Channel

Here, each of the K transmitters is equipped with N antennas and the single receiver is equipped with M antennas. The received signal vector is given by

$$\boldsymbol{y} = \sum_{k=1}^{K} \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{n}, \tag{2.1}$$

where $\boldsymbol{H}_k \in \mathbb{C}^{M \times N}$ denotes the k-th transmitter's channel to the receiver, $\boldsymbol{x}_k \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}_N, \boldsymbol{Q}_k)$ is the N-dimensional transmit vector, and $\boldsymbol{n} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}_M, \boldsymbol{C}_{\boldsymbol{n}})$ denotes the additive white Gaussian noise with covariance matrix $\boldsymbol{C}_{\boldsymbol{n}} \in \mathbb{C}^{M \times M}$, where, for simplicity, $\boldsymbol{C_n} = \mathbf{I}$ is assumed throughout this Section. The transmit signal $\boldsymbol{x}_k$ must satisfy the per-transmitter power constraint

$$\mathrm{E}\left[\|\boldsymbol{x}_k\|_2^2\right] = \mathrm{tr}(\boldsymbol{Q}_k) \leq P_k, \quad k = 1, \ldots, K. \tag{2.2}$$

The capacity region of the MIMO MAC can be found by a combination of

- the single-user capacity of each user,
- the sum capacity, and
- the weighted sum capacity.

Before we start with optimizing each of the above rate bounds, we discuss the achievable rate region for fixed transmit covariances in the following section. For the remainder of this project, we will consider the two user MIMO MAC.

2.1 Achievable Rate Region for Fixed Transmit Covariances

For given transmit covariance matrices, the achievable user rates in the Gaussian MIMO MAC are limited by

$$\begin{aligned} R_1 &\leq I(\boldsymbol{y}; \boldsymbol{x}_1 | \boldsymbol{x}_2) = \log_2 \det(\mathbf{I}_M + \boldsymbol{C_n}^{-1} \boldsymbol{H}_1 \boldsymbol{Q}_1 \boldsymbol{H}_1^{\mathrm{H}}), \qquad (2.3)\\ R_2 &\leq I(\boldsymbol{y}; \boldsymbol{x}_2 | \boldsymbol{x}_1) = \log_2 \det(\mathbf{I}_M + \boldsymbol{C_n}^{-1} \boldsymbol{H}_2 \boldsymbol{Q}_2 \boldsymbol{H}_2^{\mathrm{H}}),\\ R_1 + R_2 &\leq I(\boldsymbol{y}; \boldsymbol{x}_1, \boldsymbol{x}_2) = \log_2 \det(\mathbf{I}_M + \boldsymbol{C_n}^{-1} \boldsymbol{H}_1 \boldsymbol{Q}_1 \boldsymbol{H}_1^{\mathrm{H}} + \boldsymbol{C_n}^{-1} \boldsymbol{H}_2 \boldsymbol{Q}_2 \boldsymbol{H}_2^{\mathrm{H}}). \end{aligned}$$

These rate bounds completely define the achievable rate region for given $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$. A sketch of an examplary rate region for a two-user MAC is given in Fig. 2.2.

QUESTION 1

Give the coordinates (abscissa and ordinate values) in terms of the mutual information expressions of the points $\boldsymbol{r}_\mathrm{A} = [R_{\mathrm{A},1}, R_{\mathrm{A},2}]^\mathrm{T}$ and $\boldsymbol{r}_\mathrm{B} = [R_{\mathrm{B},1}, R_{\mathrm{B},2}]^\mathrm{T}$ on the rate region boundary in Fig. 2.2.

PROGRAMMING TASK 1

Implement a function to compute the coordinates $\boldsymbol{r}_\mathrm{A} = [R_{\mathrm{A},1}, R_{\mathrm{A},2}]^\mathrm{T}$ and $\boldsymbol{r}_\mathrm{B} = [R_{\mathrm{B},1}, R_{\mathrm{B},2}]^\mathrm{T}$ for the rate region boundary in Fig. 2.2 for given transmit covariance matrices $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$.

1 Deliverables: (Matlab code file): `ratesMAC.m`

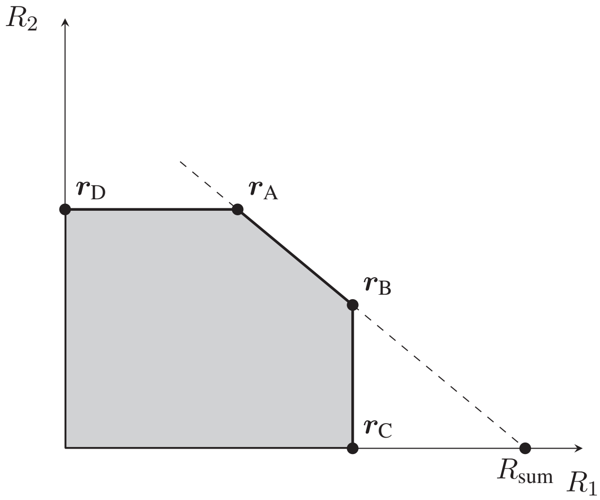

Figure 2.2: Rate region of the two-user MIMO MAC

- Function definition:
  `function [R,Rsum] = ratesMAC(Q,H)`

2 Input Specification:
- `Q`: $N \times N \times 2$ array of transmit covariances $\boldsymbol{Q}_1, \boldsymbol{Q}_2$
- `H`: $M \times N \times 2$ array of channel matrices $\boldsymbol{H}_1, \boldsymbol{H}_2$

3 Output Specification:
- `R`: 2×2 matrix of rate region coordinates $\boldsymbol{R} = [\boldsymbol{r}_\mathrm{A}, \boldsymbol{r}_\mathrm{B}]$
- `Rsum`: achieved sum rate R_sum

4 Hint(s):
- Make sure that the returned values are real numbers by employing the function `real(·)`.

PROGRAMMING TASK 2

Implement a function that plots the rate region boundary with R_1 on the abscissa and R_2 on the ordinate.

1 Deliverables (Matlab code file): `plotRegionMAC.m`
- Function definition:
  `function [fig] = plotRegionMAC(R,fig)`

2 Input Specification:

- R: $2 \times S$ array of boundary points $\boldsymbol{R} = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_S]$
- `fig` (optional): figure handle for plotting the boundary into a given figure

3 Output Specification:

- `fig`: figure handle to the figure of the plotted boundary region

4 Hint(s):

- Make sure that $\boldsymbol{R}$ is sorted increasingly (or decreasingly) in R_1.
- Create a new figure if the optional input `fig` is not used when calling the function.

For plotting an example rate region, the channels and exemplary normalized transmit covariance matrices with $\mathrm{tr}(\boldsymbol{Q}_1) = \mathrm{tr}(\boldsymbol{Q}_2) = 1$ for a two-user MIMO MAC are given in the file `exampleMAC.mat` in the format of Programming Task 1.

QUESTION 2

Determine the rate region boundary coordinates $\boldsymbol{r}_{\mathrm{A}}$ and $\boldsymbol{r}_{\mathrm{B}}$ for the given channel matrices and scaled versions of the given transmit covariance matrices in `exampleMAC.mat`, i.e., $\boldsymbol{Q}_i' = P_{\mathrm{Tx}} \boldsymbol{Q}_i$, $i = 1, 2$, for the three transmit powers $P_{\mathrm{Tx}} = -10\,\mathrm{dB}, 0\,\mathrm{dB}, 10\,\mathrm{dB}$.

QUESTION 3

Use the function `plotRegionMAC.m` to plot the exemplary rate regions for the calculated boundary points of Task 2 within one figure. (Save the figure as `exampleMACregionQ.fig`)

In the following, the transmit covariance matrices are variable and only limited by the transmit power constraints in (2.2). Then, the achievable rate region is bounded by the single user capacities, the sum capacity, and the weighted sum capacity. The optimizations for these capacities are examined next.

2.2 Single User Capacities

From the rate bounds, it is clear that the maximum single-user rates are the single user capacities C_k. The k-th user's mutual information maximization reads as

$$C_k = \max_{\boldsymbol{Q}_k \succeq \mathbf{0}} \log_2 \det \left(\mathbf{I}_N + \boldsymbol{X}_k \boldsymbol{Q}_k \right) \quad \text{s.t.} \quad \mathrm{tr}(\boldsymbol{Q}_k) \leq P_k, \tag{2.4}$$

where $\boldsymbol{X}_k = \boldsymbol{H}_k^{\mathrm{H}} \boldsymbol{C}_{\boldsymbol{n}}^{-1} \boldsymbol{H}_k$. The EVD of the rate maximizing transmit covariance is $\boldsymbol{Q}_k = \boldsymbol{V}_k \boldsymbol{\Psi}_k \boldsymbol{V}_k^{\mathrm{H}}$, where the unitary modal matrix is defined by the EVD $\boldsymbol{X}_k = \boldsymbol{V}_k \boldsymbol{\Phi}_k \boldsymbol{V}_k^{\mathrm{H}}$, and the diagonal elements of $\boldsymbol{\Psi}_k$ result from the waterfilling policy [cf. Section 3.3.3 of the Lecture notes].

PROGRAMMING TASK 3

Write a function that computes the (single-user) rate maximizing $\boldsymbol{Q}_k$ and the maximum rate C_k for given $\boldsymbol{X}_k$ and P_k.

1 Deliverables (Matlab code file): `ratemaxQk.m`

- Function definition:
  ```
  function [Qk,Ck] = ratemaxQk(Xk,Pk)
  ```

2 Input Specification:

- `Xk`: $N \times N$ effective inverse noise covariance matrix $\boldsymbol{X}_k$
- `Pk`: maximum available transmit power P_k

3 Output Specification:

- `Qk`: rate maximizing transmit covariance matrix $\boldsymbol{Q}_k$
- `Ck`: k-th user's maximum rate C_k

4 Hint(s):

- Use the function `waterfilling.m` from Project 1.

QUESTION 4

Use the function `ratemaxQk.m` to calculate the single-user capacities C_1 and C_2 and the corresponding transmit covariance matrices $\boldsymbol{Q}_{1,\text{single}}$ and $\boldsymbol{Q}_{2,\text{single}}$ for the exemplary channels in `exampleMAC.mat` when $P_1 = P_2 = 0$ dB.

If user k transmits with $\boldsymbol{Q}_{k,\text{single}}$ to achieve C_k, the rate of user $j \neq k$ is bounded from above by maximizing the unconditioned mutual information $I(\boldsymbol{x}_j; \boldsymbol{y})$, where the signal of user k is regarded as noise. The optimization reads as

$$R_{j,\text{single}} = \max_{\boldsymbol{Q}_j \succeq \mathbf{0}} \log_2 \det \left(\mathbf{I}_N + \boldsymbol{X}_j \boldsymbol{Q}_j \right) \quad \text{s.t.} \quad \operatorname{tr}(\boldsymbol{Q}_j) \leq P_j. \tag{2.5}$$

QUESTION 5

Give the expression for $\boldsymbol{X}_j$ for the unconditional mutual information in (2.5).

QUESTION 6

Use the function `ratemaxQk.m` to calculate the maximum achievable rate $R_{j,\text{single}}$ and the corresponding transmit covariance matrix $\boldsymbol{Q}_j$ for user $j \neq k$, when user k employs $\boldsymbol{Q}_{k,\text{single}}$ to transmit with its single-user capacity C_k, $j, k = 1, 2$, with $P_1 = P_2 = 0$ dB.

2.3 Sum Capacity via Iterative Waterfilling

Being aware of the joint mutual information in (2.3), we can state the optimization problem for achieving the sum capacity of the considered two-user MIMO MAC as follows:

$$C_{\text{sum}} = \max_{\boldsymbol{Q}_1, \boldsymbol{Q}_2} I(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{y}) \qquad \text{s.t.} \quad \boldsymbol{Q}_i \succeq \mathbf{0}, \quad \text{tr}(\boldsymbol{Q}_i) \leq P_i, \quad i = 1, 2 \tag{2.6}$$

where the mutual information expression is given in (2.3). This optimization problem is convex as detailed in Section 3.3.3 of the lecture notes and Problem 4.2 of the tutorials.

An efficient algorithm for solving (2.6) is the *iterative waterfilling* procedure. It exploits the chain rule for mutual information for alternatingly updating the transmit covariance matrices.

QUESTION 7

Use the chain rule for mutual information to split $I(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{y})$ into the sum of two mutual information expressions. Two forms are possible. Which of the two forms corresponds to an update of $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$, respectively?

The update of the k-th transmit covariance matrix in the $n+1$-th iteration of the iterative waterfilling procedure reads as

$$\boldsymbol{Q}_k^{(n+1)} = \underset{\boldsymbol{Q}_k \succeq \mathbf{0}}{\arg\max} \log_2 \det\left(\mathbf{I}_N + \boldsymbol{X}_k^{(n)} \boldsymbol{Q}_k\right) \quad \text{s.t.} \quad \text{tr}(\boldsymbol{Q}_k) \leq P_k. \tag{2.7}$$

This optimization is equivalent to the single-user capacity optimizations in (2.4), where $\boldsymbol{X}_k^{(n)}$ depends on the currently available transmit covariance matrix of the other user (cf. Task 5).

Alternatingly updating the transmit covariance matrices as in (2.7), the iterative waterfilling algorithm monotonically increases the achieved sum rate in each update until convergence to the sum capacity of the given MIMO MAC. Convergence can be declared when the change in variables falls below a given threshold, i.e., $\sum_{i=1}^{K} \|\boldsymbol{Q}_i^{(n+1)} - \boldsymbol{Q}_i^{(n)}\|_{\mathrm{F}}^2 \leq \epsilon$.

PROGRAMMING TASK 4

Write a function that computes the sum capacity C_{sum} and the maximizing $\boldsymbol{Q}_k$ for given channels $\boldsymbol{H}_k$ and transmit powers, $k = 1, 2$.

1 Deliverables (Matlab code file): `iterWaterfill.m`

- Function definition:
  `function [Q,Csum,Rsum] = iterWaterfill(H,P,epsilon)`

2 Input Specification:

- `H`: $M \times N \times 2$ array of the users' channels $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$
- `P`: 2×1 vector of the users' transmit powers P_1 and P_2
- `epsilon`: stopping threshold ϵ of the iterative waterfilling algorithm

3 Output Specification:

- `Q`: rate maximizing transmit covariance matrices $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$
- `Csum`: sum rate C_{sum} of the iterative waterfilling algorithm after convergence
- `Rsum`: vector of sum rate values for the iterations.

4 Hint(s):

- Initialize the algorithm with $\boldsymbol{Q}_1^{(0)} = \boldsymbol{Q}_2^{(0)} = \boldsymbol{0}$ and use the function `ratemaxQk.m` from Programming Task 3 for the updates of the transmit covariance matrices $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$.

QUESTION 8

Run the function `iterWaterfill.m` for the channels given in `exampleMAC.mat` and $P_1 = P_2 = 0$ dB for calculating C_{sum}, $\boldsymbol{Q}_1$, and $\boldsymbol{Q}_2$. How many iterations are required for an accuracy of $\epsilon = 10^{-4}$?

With the transmit covariance matrices calculated in Tasks 4, 6, and 8, three rate regions can be plotted that have a line segment at the capacity region boundary of the given MAC.

QUESTION 9

First, calculate the boundary points to these rate regions that correspond to the transmit covariance matrices from Questions 4, 6, and 8. Then, plot the three rate regions into one figure with `plotRegionMAC.m` and save the plot under the name `singleAndSumBounds.fig`. Do the three line segments at the capacity region boundary of the MAC adjoin? What is the reason for this behavior?

2.4 Weighted Sum Capacity

To fill the gaps between the line segments that correspond to the sum capacity bound and the single user capacity bounds, a *weighted sum rate* (WSR) maximization problem has to be solved [cf. Section 3.3.3 of the Lecture notes] for different weights. For given weights $w_1, w_2 \in [0,1]$ with $w_1 + w_2 = 1$, the weighted sum rate optimization reads as

$$C_{\text{wsum}} = \max_{\boldsymbol{Q}_1,\boldsymbol{Q}_2} w_1 R_1 + w_2 R_2 \qquad \text{s.t.} \quad \boldsymbol{Q}_i \succeq \boldsymbol{0}, \quad \text{tr}(\boldsymbol{Q}_i) \leq P_i, \quad i = 1,2. \tag{2.8}$$

In contrast to the sum rate maximization in Subsection 2.3, the maximally achievable WSR depends on the decoding order of the transmitted signals. Fortunately, the decoding order only depends on the values for the weights w_i, $i = 1,2$ due to the (polymatroidal) structure of the rate region boundary.

QUESTION 10

Give the decoding order of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ for $w_2 > w_1$ and state the corresponding mutual information expressions $I(\bullet;\bullet)$ that replace R_1 and R_2 in (2.8) for this scenario.

For the other ordering of the weights, i.e. $w_1 > w_2$, the decoding order is opposite. Hence, the resulting WSR maximization can be written in the (general) form

$$C_{\text{wsum}} = \max_{\boldsymbol{Q}_1,\boldsymbol{Q}_2} R_{\text{wsum}}(\boldsymbol{Q}_1,\boldsymbol{Q}_2) \quad \text{s.t.} \quad \boldsymbol{Q}_i \succeq \boldsymbol{0}, \quad \text{tr}(\boldsymbol{Q}_i) \leq P_i, \quad i = 1,2 \tag{2.9}$$

where the WSR function R_{wsum} is defined as

$$\begin{aligned} R_{\text{wsum}} : (\boldsymbol{Q}_1,\boldsymbol{Q}_2) = (w_i - w_j) \log_2 \det\left(\mathbf{I}_M + \boldsymbol{C}_{\boldsymbol{n}}^{-1} \boldsymbol{H}_i \boldsymbol{Q}_i \boldsymbol{H}_i^{\text{H}}\right) \\ + w_j \log_2 \det\left(\mathbf{I}_M + \boldsymbol{C}_{\boldsymbol{n}}^{-1} \boldsymbol{H}_1 \boldsymbol{Q}_1 \boldsymbol{H}_1^{\text{H}} + \boldsymbol{C}_{\boldsymbol{n}}^{-1} \boldsymbol{H}_2 \boldsymbol{Q}_2 \boldsymbol{H}_2^{\text{H}}\right) \end{aligned} \tag{2.10}$$

and the indices $i, j \in \{1, 2\}$ with $i \neq j$ have to be chosen such that $w_i > w_j$.

Above problem is a convex optimization problem and belongs to the important class of *semidefinite programs*. This type of problem can be solved by standard solvers such as SDPT3. As these solvers are often rather cumbersome to use, additional packages like YALMIP or CVX are commonly used to turn Matlab into a modeling language for convex optimization.

An exemplary Matlab code that solves the standard point-to-point MIMO problem

$$\hat{\boldsymbol{Q}} = \underset{\boldsymbol{Q} \succeq 0}{\arg\max} \log\det\left(\boldsymbol{I}_M + \boldsymbol{H}\boldsymbol{Q}\boldsymbol{H}^{\mathrm{H}}\right) \quad \text{s.t.} \quad \operatorname{tr}(\boldsymbol{Q}) \leq P \tag{2.11}$$

with YALMIP as modeling language and SDPT3 as backend solver is given below.

```
1 M = 4;
2 N = 4;
3 H = 1/sqrt(2)*(randn(M,N) + 1i*randn(M,N));
4 P = 10^3;
5
6 % Set YALMIP options
7 options = sdpsettings('solver','sdpt3','verbose',0);
8 % Initialize optimization variables
9 Q = sdpvar(N,N,'hermitian','complex');
10 % Define constraint set
11 Constraints = [Q>=0, trace(Q)<=P];
12 % Define objective function (minimization only!)
13 Objective = -logdet(eye(M) + H*Q*H' );
14 % Solve optimization problem
15 sol = optimize(Constraints, Objective, options);
16 % Retrieve solution
17 C = real(log2(det(eye(M) + H*value(Q)*H')));
```

Listing 2.1: Matlab YALMIP code for solving the point-to-point MIMO problem

In Line 7 the options are set such that SDPT3 is used as back end. With 'verbose' set to zero, the output of the solver status during the optimization is suppressed. In Line 9, the complex Hermitian optimization variable $\boldsymbol{Q}$ with size $N \times N$ is initialized. The definition of multiple optimization variables is possible by calling the function `sdpvar` multiple times for different variables. The constraints are defined in Line 11 in the form of an array. The objective function is defined in Line 13. Note that by definition, optimization problems are always minimization problems in YALMIP. By calling the `optimize` function in Line 15, the solver is started. The optimizing argument can be obtained by calling the function `value` (see Line 17).

Installation instructions for YALMIP and SDPT3 are given below. Please refer to the YALMIP and SDPT3 user guides for further information.

YALMIP and SDPT3 Installation Instructions

Install YALMIP in the folder for your Matlab simulations by the following steps:

- Retrieve the latest version of YALMIP from `https://yalmip.github.io/` and unpack the file in your simulations folder.
- Start Matlab and change to the location of YALMIP
- Add the folder and subfolders to your MATLAB path (right click on the folder and select 'Add to Path')
- To save the path for subsequent Matlab sessions type `savepath`.

Install SDPT3 in the folder for your Matlab simulations by the following steps:

- Retrieve the latest version of SDPT3 from `http://www.math.nus.edu.sg/~mattohkc/sdpt3.html` and unpack the file in your simulations folder.
- Start Matlab and change to the location of SDPT3
- Run the scripts `Installmex.m` and `startup.m`
- To save the path for subsequent Matlab sessions type `savepath`.

PROGRAMMING TASK 5

Modify the function `maxWSRmac.m` that it maximizes the weighted sum-rate as in (2.9) for given weights and power levels P_1 and P_2.

1 Deliverables (Matlab code file): `maxWSRmac.m`

- Function definition:
```
function [Q,Cwsr] = maxWSRmac(H,P,w)
```

2 Input Specification:

- `H`: $M \times N \times 2$ array of the users' channels $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$
- `P`: 2×1 vector of the users' transmit powers P_1 and P_2
- `w`: 2×1 vector of each users' weight w_1 and w_2.

3 Output Specification:

- `Q`: rate maximizing transmit covariance matrices $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$
- `Cwsr`: maximum weighted sum rate C_{wsum}

4 Hint(s):

- Use YALMIP with SDPT3 to maximize the weighted sum rate,
- Distinguish the two cases where user 1 or user 2 is decoded first.

2.5 Capacity Region Boundary

With the single-user rate maximization, the sum rate maximization by iterative waterfilling, and the weighted sum rate maximization, all points on the boundary of the capacity region can be computed. This allows us to analyze the shape of the Pareto boundary of the capacity region for various transmit powers and channel configurations.

For plotting the complete capacity region boundary of the two-user MIMO MAC, $2S$ sample points of the Pareto boundary shall be computed. The weights w_1 and w_2 for these points shall vary between zero and one in steps of $\frac{1}{2(S-1)}$. The boundary point calculation shall depend on the weights, i.e., the single user (capacity) calculations of Subsection 2.2 for either $w_1 = 0$ or $w_2 = 0$, the iterative waterfilling of Subsection 2.3 for $w_1 = w_2$ (two boundary points result from this case), and the WSR maximization of Subsection 2.4 for the other cases.

PROGRAMMING TASK 6

Write a function that calculates $2S$ coordinate points $\boldsymbol{r}_s$, $s = 1, \ldots, 2S$ on the Pareto boundary of the two-user MIMO MAC capacity region $\mathcal{C}$ via varying the weights w_1 and w_2 and for given channels $\boldsymbol{H}_i$ and powers P_i, $i = 1, 2$

1 Deliverables (Matlab code file): `ParetoBound.m`
- Function definition: `function [R] = ParetoBound(H,P,S)`

2 Input Specification:
- `H`: $M \times N \times 2$ array of channel matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$
- `P`: column vector of available transmit powers P_1 and P_2
- `S`: number of Pareto boundary sample points S

3 Output Specification:
- `R`: $2 \times 2S$ matrix of rate region coordinates $\boldsymbol{R} = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{2S}]$

4 Hint(s):
- Distinguish the different cases for the weights w_1 and w_2 as mentioned above.

This function shall be used in what follows to analyze the (convex) shape of the capacity region for low, medium, and high transmit power.

QUESTION 11

Calculate $S = 20$ Pareto boundary points of the exemplary two user MIMO MAC in `exampleMAC.mat` for $P_1 = P_2 = -10$ dB, 0 dB, 10 dB and plot the boundaries of the corresponding capacity regions $\mathcal{C}$ with `plotRegionMAC.m` into one figure. Save the figure as `exampleMACregion.fig`.

- Which of the rate regions is closest to a rectangular shape that is spanned by the origin and the single user capacities C_1 and C_2?
- Which of the rate regions is closest to the triangle that is spanned by the origin and the rate pairs $(C_1, 0)$ and $(0, C_2)$?