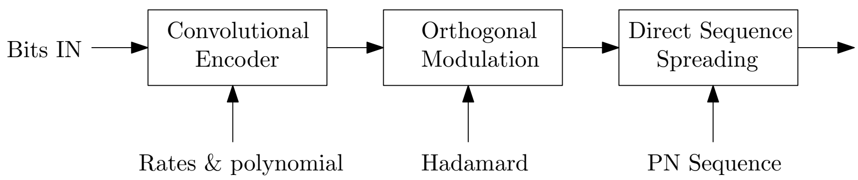
1 Introduction

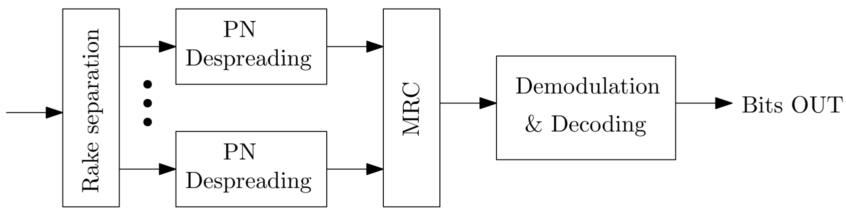
In this project you will implement the digital baseband portion of a MIMO-CDMA system conforming to the IS-95 CDMA standard in MATLAB. IS-95 CDMA systems were first developed in the 1990s and are still widely used in North American 2G cellular networks (often as a fall-back mode and in rural areas).
You are expected to write a suitable structure and framework for the simulator and to justify the design decisions. Measurements of the performance (e.g., BER and capacity) should be included for various channel scenarios, so design your system accordingly.

2 Implementation of the IS-95 Standard

In this part of the project you will implement the forward-link (i.e., basestation to mobile) of a single antenna IS-95 system. The implementation is similar to the laboratory exercises on CDMA, but you will need to include convolutional encoding and design your system (e.g., spreading rates and PN sequences) to conform to the IS-95 standard.



(a) Transmitter



(b) Recevier

Figure 1: High-level block diagram of an IS-95 forward-link.

A high-level block diagram of the IS-95 forward-link is shown in Fig. 1. There is a lot of information contained within the Standard (which is available on Moodle) and you will need to extract the relevant information about the:

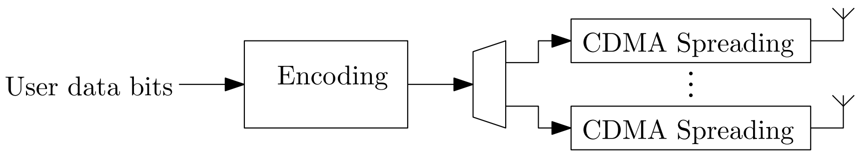
- size of the frames;

- encoding schemes and the appropriate rates and polynomials; and
- Hadamard orthogonalisation and PN spreading sequences.

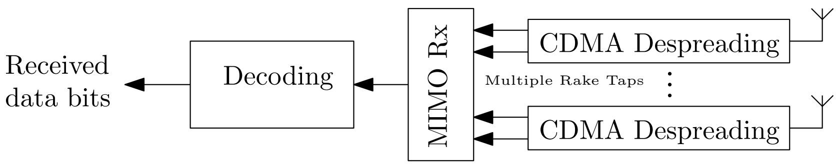
3 Implementation of MIMO-CDMA

MIMO is used to increase the channel capacity by exploiting multi-path propagation, however, in the exercises we have focused on the case when there is a single user in the system. In this part of the project, we will consider a multi-user MIMO system where the users are separated using CDMA, specifically the IS-95 standard. Fig. 2 shows a very high-level block diagram of how the MIMO transmitter and receiver blocks should fit into your existing implementation. Test your implementation with a 2×2 MIMO system (once this works you can try other configurations). Measurements of the performance (e.g., BER) should be included for various channel scenarios and you are required to implement the various MIMO receivers discussed during the lectures.

There are some specific considerations that need to be taken into account when designing the receiver architecture for MIMO-CDMA. In particular, combining the Rake-taps *before* the MIMO receiver will lead to a very poor result (bonus point: explain why). We can get around this problem by assuming the Rake-taps are 'virtual antennas' and performing the MIMO signal processing on the entire set of Rake-taps as shown in Fig. 2.



(a) Transmitter



(b) Receiver

Figure 2: High-level block diagram of CDMA-MIMO system.

4 Channel Models

For both parts of the project you are required to implement various channel models. Initially, to confirm your code is working correctly it is often a good idea to use a pass-through channel, i.e., no noise or distortion. Once you confirm the basic functionality of your code (i.e., that you get back what you put in), you can start adding in the channel effects.

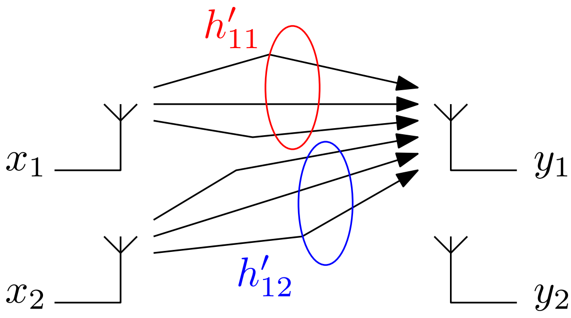


Figure 3: MIMO channel

- The simplest is an AWGN channel (don't use the MATLAB built-in function for this, rather, use `randn` and scale your noise signal appropriately depending on the SNR).
- However, for wide-band signals, such as CDMA, we need to consider the impact of fading, which in turn will require the implementation of a Rake receiver to compensate for the multi-path effects.

MIMO Channel with Delay Paths

This section shows how the multi-path effects can be captured in the MIMO channel matrix, which will lead to simpler signal processing at the receiver side. For a 2×2 MIMO-CDMA system with three Rake receiver fingers, the overall H-matrix is 6×2 (where we assume the Rake fingers act as 'virtual antennas'),

$$H = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ h_{31} & h_{32} \\ h_{41} & h_{42} \\ h_{51} & h_{52} \\ h_{61} & h_{62} \end{bmatrix}. \tag{1}$$

However, we only have 2 physical antennas, as shown in Fig. 3. In this case, the (physical) signals y_1 and y_2 can be found by considering the channel matrix as,

$$H = \begin{bmatrix} h'_{11} & h'_{12} \\ h'_{21} & h'_{22} \end{bmatrix}, \tag{2}$$

where each h'_{nm} consists of 3 delay taps,

$$h'_{nm} = \left[g^1_{nm}, g^2_{nm}, g^3_{nm}\right]^T. \tag{3}$$

Giving the overall 6×2 H-matrix,

$$H = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ h_{31} & h_{32} \\ h_{41} & h_{42} \\ h_{51} & h_{52} \\ h_{61} & h_{62} \end{bmatrix} = \begin{bmatrix} g^1_{11} & g^1_{12} \\ g^2_{11} & g^2_{12} \\ g^3_{11} & g^3_{12} \\ g^1_{21} & g^1_{22} \\ g^2_{21} & g^2_{22} \\ g^3_{21} & g^3_{22} \end{bmatrix}. \tag{4}$$

The signals on the two physical receiving antennas are thus,

$$y_1 = h'_{11} * x_1 + h'_{12} * x_2 \tag{5}$$

$$y_2 = h'_{21} * x_1 + h'_{22} * x_2, \tag{6}$$

where $*$ represents convolution. MIMO signal processing can then be performed on y_1 and y_2 to recover the bit-streams.

5 Helpful Guidelines

- Write general Matlab code, i.e., don't hard-code variables such as (but not limited to!) the coding rates, MIMO configuration or the number of users in the system.
- Write meaningful comments in your code so that anyone reading/grading it afterwards will understand what you are doing.
- Make sure you simulate a sufficient number of bits to ensure your BER calculations are valid.

6 Deliverables

The deliverables for this project are:

- a 10–15 page report outlining the system and explaining the results;
- a 15 minute presentation (in the last week of semester); and
- the MATLAB code.

Please upload all the files in a single compressed directory to Moodle before the due date. You are permitted to work in groups of two.