

ELEC3204 Wireless and Optical Communications

Space-Time Block Code Coursework

Rob Maunder - rm@ecs.soton.ac.uk

Please note that the **marking scheme** for this coursework is discussed in Section 5.

1 Introduction

In this coursework, your task is to design the transmitters and the receiver of the multi-user Space-Time Block Code (STBC) scheme shown in Figure 1.

In this scheme, $K = 6$ users are transmitting to a single receiver. The input to the k^{th} user's transmitter is a vector of $L = 4$ bits $\mathbf{b}^{(k)}$, as shown in Figure 1. The output of the k^{th} user's transmitter should be an STBC signal $\mathbf{X}^{(k)}$ comprising $T \times P = 16$ complex symbols, which are transmitted during $T = 4$ timeslots using $P = 4$ transmit antennas, as shown in Figure 1. Each user's transmitter is required to have an average transmission energy per bit $E_b^{(k)}$ of no more than 1, where

$$E_b^{(k)} = \text{E} \left[\frac{1}{L} \sum_{t=1}^T \sum_{p=1}^P |x_{t,p}^{(k)}|^2 \right] \quad (1)$$

and $\text{E}[\cdot]$ is the expected value of its operand, while $|\cdot|$ is the magnitude of its operand.

The input to the receiver is the signal \mathbf{Y} comprising $T \times Q = 24$ complex symbols, which are received during the $T = 4$ timeslots using $Q = 6$ receive antennas, as shown in Figure 1. Here, \mathbf{Y} is obtained according to

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{H} + \mathbf{N} \quad (2)$$

where

$$\mathbf{X} = [\mathbf{X}^{(1)} \quad \mathbf{X}^{(2)} \quad \mathbf{X}^{(3)} \quad \mathbf{X}^{(4)} \quad \mathbf{X}^{(5)} \quad \mathbf{X}^{(6)}] \quad (3)$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}^{(1)} \\ \mathbf{H}^{(2)} \\ \mathbf{H}^{(3)} \\ \mathbf{H}^{(4)} \\ \mathbf{H}^{(5)} \\ \mathbf{H}^{(6)} \end{bmatrix}. \quad (4)$$

More specifically, $\mathbf{H}^{(k)}$ are the $P \times Q = 24$ Rayleigh-distributed complex channel gains, which impair the signal transmitted between each combination of the k^{th} user's $P = 4$ transmit antennas and the receiver's $Q = 6$ receive antennas. Note that these channel gains remain constant throughout all $T = 4$ timeslots. Furthermore, \mathbf{N} comprises the $T \times Q = 24$ complex Additive White Gaussian Noise (AWGN) signals, which have the noise power spectral density N_0 and are received during the $T = 4$ timeslots using the $Q = 6$ receive antennas, as shown in Figure 1. The receiver may exploit perfect knowledge of all $KP \times Q = 144$ complex channel gains in \mathbf{H} , in order to perform multi-user detection. The output of the receiver should be the decoded bit matrix $\hat{\mathbf{B}}$ comprising $K \times L = 24$ bits, which correspond to each of the $K = 6$ user's $L = 4$ bits.

2 Matlab

You are required to implement your transmitter and receiver as the Matlab functions `transmitter.m` and `receiver.m`, respectively. Your design can be tested using the following Matlab files, which can be downloaded from

<https://secure.ecs.soton.ac.uk/notes/elec3204/rm/coursework.zip>

`main.m` - This script can be run to generate the bit vectors $\mathbf{b}^{(k)}$ and provide these to `transmitter.m`. The script will check that the resultant STBC matrices $\mathbf{X}^{(k)}$ have the correct dimensions and it will measure the average transmission energies per bit $E_b^{(k)}$, giving an error message if any of these are greater than

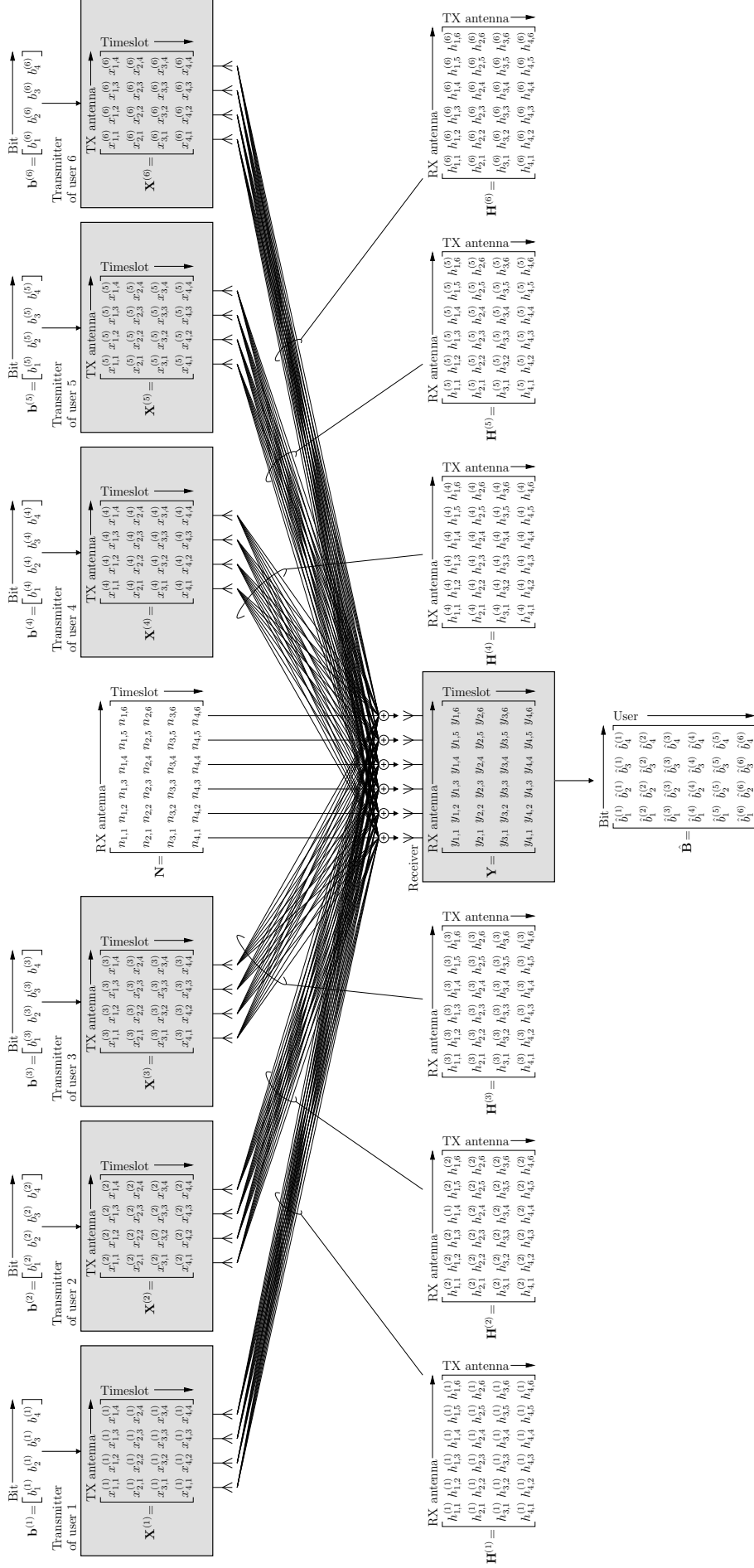


Figure 1: Schematic of the multi-user STBC scheme.

1. The script will generate the channel matrix \mathbf{H} and the noise matrix \mathbf{N} , before combining these with the STBC matrices $\mathbf{X}^{(k)}$ to give the received matrix \mathbf{Y} . This is provided to `receiver.m`, together with perfect knowledge of the channel matrix \mathbf{H} . The dimensions of the resultant bit matrix $\hat{\mathbf{B}}$ are checked by the script and the number of bit errors observed for each user is measured. The script will plot the Bit Error Ratio (BER) for each user as a function of $1/N_0$.

`transmitter.m` - This function is listed in Figure 2. It implements an example transmitter, in which a Quaternary Phase Shift Keying (QPSK) Alamouti STBC signal is transmitted by only one user using two transmit antennas and two timeslots. You need to replace this function with your own `transmitter.m`, which must use the line `function [X_k, tx_title] = transmitter(b_k, k)` to define the inputs and outputs of the function. Here, `b_k` provides the k^{th} user's vector of $L = 4$ bits $\mathbf{b}^{(k)}$, while `k` is the index of the user k . Furthermore, `X_k` should provide the k^{th} user's STBC signal $\mathbf{X}^{(k)}$ comprising $T \times P = 16$ complex symbols, having $T = 4$ rows corresponding to the timeslots and $P = 4$ columns corresponding to the transmit antennas. Finally, `tx_title` should provide a string that describes the operation of the transmitter.

`receiver.m` - This function is listed in Figure 3. It implements an example receiver, in which a QPSK Alamouti STBC signal is received from only one user using one receive antenna and two timeslots. You should replace this function with your own `receiver.m`, which must define the inputs and outputs of the function using the line `function [B_hat, rx_title] = receiver(Y, H)`. Here, `Y` provides the received signal \mathbf{Y} comprising $T \times Q = 24$ complex symbols, having $T = 4$ rows corresponding to the timeslots and $Q = 6$ columns corresponding to the receive antennas. Furthermore, `H` provides the receiver's perfect knowledge of the channel matrix \mathbf{H} comprising $KP \times Q = 144$ complex channel gains, having $KP = 24$ rows corresponding to the transmit antennas of the users, as well as $Q = 6$ columns corresponding to the receive antennas. Also, `B_hat` should provide the bit matrix $\hat{\mathbf{B}}$ comprising $K \times L = 24$ bits, having $K = 6$ rows corresponding to the users and $L = 4$ columns corresponding to the bits of each user. Finally, `rx_title` should provide a string that describes the operation of the receiver.

3 Example

The example `transmitter.m` and `receiver.m` of Figures 2 and 3 implement a simple transmitter and receiver. When employing the transmitter of Figure 2, only the first of the $K = 6$ users will transmit anything. More specifically, the transmitter of the first user splits the vector of $L = 4$ bits $\mathbf{b}^{(1)} = [b_1^{(1)} \ b_2^{(1)} \ b_3^{(1)} \ b_4^{(1)}]$ into two pairs of bits $[b_1^{(1)} \ b_2^{(1)}]$ and $[b_3^{(1)} \ b_4^{(1)}]$. Using the constellation diagrams of Figure 4, the pair of bits $[b_1^{(1)} \ b_2^{(1)}]$ is converted into the complex QPSK symbol $x_{1,1}^{(1)}$, while $[b_3^{(1)} \ b_4^{(1)}]$ is converted into the symbol $x_{1,2}^{(1)}$. For example, if $\mathbf{b}^{(1)} = 0001$, then the complex QPSK symbols $x_{1,1}^{(1)} = +\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i$ and $x_{1,2}^{(1)} = -\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i$ are generated.

The first user transmits the complex QPSK symbols $x_{1,1}^{(1)}$ and $x_{1,2}^{(1)}$ using only the first two of its $P = 4$ transmit antennas and only in the first two of the $L = 4$ timeslots, using the Alamouti STBC according to

$$\mathbf{X}^{(1)} = \begin{bmatrix} x_{1,1}^{(1)} & x_{1,2}^{(1)} & 0 & 0 \\ -\left(x_{1,2}^{(1)}\right)^* & \left(x_{1,1}^{(1)}\right)^* & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (5)$$

where $(\cdot)^*$ is the complex conjugate of its operand. Meanwhile, the other five users transmit nothing, giving

$$\mathbf{X}^{(2)} = \mathbf{X}^{(3)} = \mathbf{X}^{(4)} = \mathbf{X}^{(5)} = \mathbf{X}^{(6)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (6)$$

The receiver uses only the first of its $Q = 6$ receive antennas and only in the first two of the $L = 4$ timeslots, resulting in a pair of complex received symbols $y_{1,1}$ and $y_{2,1}$. The receiver exploits its perfect knowledge of the complex channel coefficients between the first receive antenna and the first two transmit antennas of the first user, namely $h_{1,1}^{(1)}$ and $h_{2,1}^{(1)}$. More specifically, the receiver expresses this knowledge in the form $\mathbf{Y}' = \mathbf{H}' \cdot \mathbf{X}' + \mathbf{N}'$, according to

$$\underbrace{\begin{bmatrix} y_{1,1} \\ y_{2,1}^* \end{bmatrix}}_{\mathbf{Y}'} = \underbrace{\begin{bmatrix} h_{1,1}^{(1)} & h_{2,1}^{(1)} \\ \left(h_{2,1}^{(1)}\right)^* & -\left(h_{1,1}^{(1)}\right)^* \end{bmatrix}}_{\mathbf{H}'} \cdot \underbrace{\begin{bmatrix} x_{1,1}^{(1)} \\ x_{1,2}^{(1)} \end{bmatrix}}_{\mathbf{X}'} + \underbrace{\begin{bmatrix} n_{1,1} \\ n_{2,1}^* \end{bmatrix}}_{\mathbf{N}'}. \quad (7)$$

```

% The kth user's transmitter, which converts 4 bits into an STBC signal,
% which is transmitted using 4 antennas and 4 timeslots
% b.k - this is the kth user's vector of four bits
% k - this is the index of the user, in the range 1 to 4
% X.k - this is a 4x4 matrix of complex STBC symbols, where the 4 rows correpond
%       to the 4 timeslots and the 4 columns correspond to the 4 transmit antennas
% tx.title - this is a string that describes the operation of the transmitter
function [X.k, tx.title] = transmitter(b.k, k)

% Describe the operation of the transmitter
tx.title = 'A QPSK Alamouti STBC signal is transmitted by 1 user using 2 antennas and 2 timeslots';

% Set the parameters that are assumed by the transmitter
L = 4; % number of bits per user
T = 4; % number of timeslots
P = 4; % number of transmit antennas per user

% Check that the dimensions of the input agree with the parameters above
if length(b.k) ~= L
    error('Soton:argChk','length(b.k) ~= L');
end

% Nothing will be transmitted by most users on most antennas in most timeslots
X.k = zeros(T, P);

% Only user 1 transmits anything
if k == 1
    % Determine the transmissions in the first timeslot using Gray-coded QPSK
    X.k(1,1) = (1-2*b.k(1,2))/sqrt(2) + 1i*(1-2*b.k(1,1))/sqrt(2);
    X.k(1,2) = (1-2*b.k(1,4))/sqrt(2) + 1i*(1-2*b.k(1,3))/sqrt(2);

    % Determine the transmissions in the second timeslot using Alamouti STBC
    X.k(2,1) = -conj(X.k(1,2));
    X.k(2,2) = conj(X.k(1,1));
end
end

```

Figure 2: Listing of example transmitter.m.

```

% A MUD receiver, which recovers 4 bits for each of 6 users from the signal
% received using 6 antennas and 4 timeslots. The 4 users' STBC signals interfere
% with each other and are impaired by quasi-static fading channels and by noise.
% Y - this is a 4x6 matrix of complex received symbols, where the 4 rows correspond
% to the 4 timeslots and the 6 columns correspond to the 6 receive antennas
% H - this is a 16x6 matrix of complex channel coefficients, where the 16
% rows correspond to the 4 transmit antennas of the 4 users, while the 6
% columns correspond to the 6 receive antennas
% B.hat - this is a 6x4 matrix of decoded bits, where the 6 rows correspond
% to the 6 users and the 4 columns correspond to the 4 bits of each user
% rx_title - this is a string that describes the operation of the receiver
function [B.hat, rx_title] = receiver(Y, H)

% Describe the operation of the receiver
rx_title = 'A QPSK Alamouti STBC signal is received using 1 antenna and 2 timeslots';

% Set the parameters that are assumed by the receiver
K = 6; % number of users
L = 4; % number of bits per user
T = 4; % number of timeslots
P = 4; % number of transmit antennas per user
Q = 6; % number of receive antennas

% Check that the dimensions of the inputs agree with the parameters above
if ~isequal(size(Y), [T, Q])
    error('Soton:argchk', '~isequal(size(Y), [T, Q])');
end
if ~isequal(size(H), [P*K, Q])
    error('Soton:argchk', '~isequal(size(H), [P*K, Q])');
end

% Build matrices of the form Y2 = H2*X2+N2
Y2 = [Y(1,1); conj(Y(2,1))];
H2 = [H(1,1), H(2,1); conj(H(2,1)), -conj(H(1,1))];

% Perform single-user detection
Z2 = H2'*Y2;

% Zero will be used for the decoded bits of most users
B.hat = zeros(K, L);

% QPSK demodulation for user 1
B.hat(1,1) = imag(Z2(1,1)) < 0;
B.hat(1,2) = real(Z2(1,1)) < 0;
B.hat(1,3) = imag(Z2(2,1)) < 0;
B.hat(1,4) = real(Z2(2,1)) < 0;

end

```

Figure 3: Listing of example receiver.m.

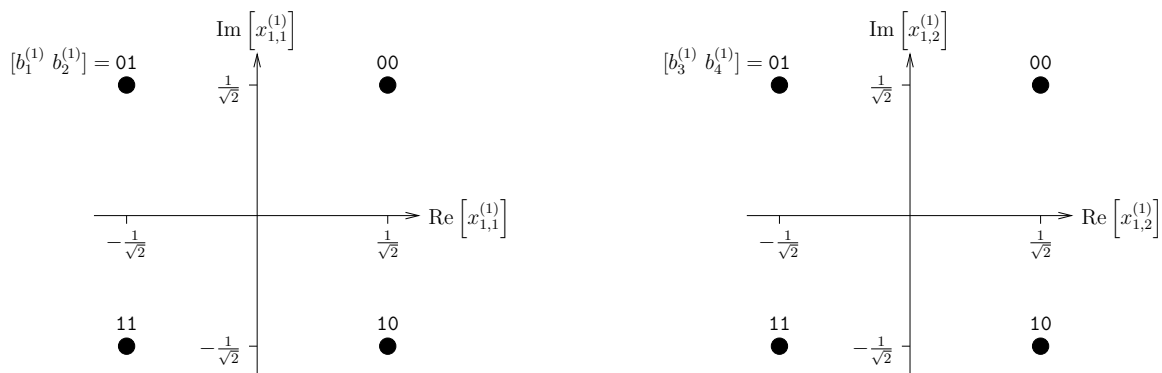


Figure 4: QPSK constellation diagrams.

This allows decision variables to be obtained in the form $\mathbf{Z} = (\mathbf{H}')^H \cdot \mathbf{Y}'$, according to

$$\underbrace{\begin{bmatrix} z_{1,1}^{(1)} \\ z_{1,2}^{(1)} \end{bmatrix}}_{\mathbf{Z}'} = \underbrace{\begin{bmatrix} (h_{1,1}^{(1)})^* & h_{2,1}^{(1)} \\ (h_{2,1}^{(1)})^* & -h_{1,1}^{(1)} \end{bmatrix}}_{(\mathbf{H}')^H} \cdot \underbrace{\begin{bmatrix} y_{1,1} \\ y_{2,1}^* \end{bmatrix}}_{\mathbf{Y}'}, \quad (8)$$

where $(\cdot)^H$ is the Hermitian transpose of its operand.

The receiver obtains an estimate of the first user's bits $[\hat{b}_1^{(1)} \ \hat{b}_2^{(1)}]$ from the label of the constellation point in Figure 4 having the lowest Euclidean distance from the decision variable $z_{1,1}^{(1)}$. Likewise, estimates of the bits $[\hat{b}_3^{(1)} \ \hat{b}_4^{(1)}]$ are obtained from the decision variable $z_{1,2}^{(1)}$. Finally, the recovered bits are generated according to

$$\hat{\mathbf{B}} = \begin{bmatrix} \hat{b}_1^{(1)} & \hat{b}_2^{(1)} & \hat{b}_3^{(1)} & \hat{b}_4^{(1)} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (9)$$

where the bits associated with the second, third, fourth, fifth and sixth users are all set to zero.

Figure 5 plots each user's BER versus $1/N_0$, when employing the example transmitter and receiver of Figures 2 and 3. Note that the BER of the second, third, fourth, fifth and sixth user is equal to 0.5, regardless of the noise power spectral density N_0 . This is because the corresponding transmitters do not transmit anything.

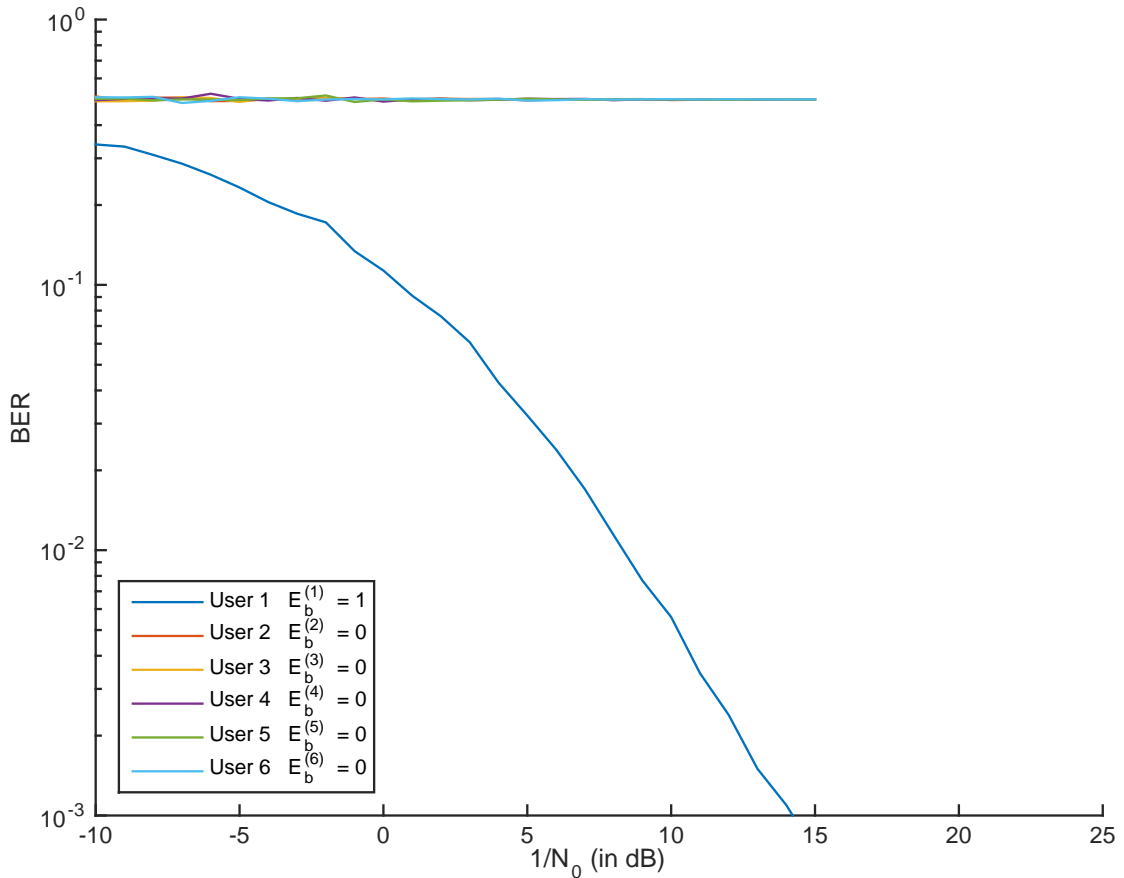


Figure 5: BER plot for each of the $K = 6$ users, when employing the example transmitter and receiver of Figures 2 and 3.

4 Submission

The process for submitting your work is as follows:

1. You should go to <https://handin.ecs.soton.ac.uk/handin/1920/ELEC3204/1/>.
2. There, you should submit the following four things before 4pm on Wednesday 4th March 2020:
 - Your `transmitter.m` function.
 - Your `receiver.m` function.
 - A PDF file containing a plot of the BER achieved by each user as a function of $1/N_0$, as exemplified in Figure 5.
 - A PDF file containing diagrams, equations and discussion, which explain your transmitter and receiver design. This should contain a similar level of detail as is provided in Section 3, so that somebody else could independently reproduce your `transmitter.m` and `receiver.m` functions.

Within four weeks of the submission deadline, I will email some generalised feedback to the class. I will also produce a webpage that anonymises the submissions and ranks them according to the BERs achieved at $1/N_0 = 5$ dB. This will allow you to see how you have performed in relation to the rest of the class. It will also allow you to see the PDF documents submitted by each member of the class, but not their Matlab code. If you want to maintain your anonymity, please don't include your name or email address within your PDF documents. In the revision lecture at the end of the semester, I will discuss my model solution and the best submitted solutions with the class.

5 Tips

Here are some tips that will help you get a high mark in this coursework:

- There are lots of ways that the example transmitter and receiver of Figures 2 and 3 can be improved. In particular, the second, third, fourth, fifth and sixth users can be made to transmit something. You may like to start by implementing a Time Division Multiple Access (TDMA) scheme, although there are not enough timeslots to accommodate all six users in this way. Also, the transmitters can be made to use more than two of their transmit antennas and more than two of the timeslots. Likewise, the receiver can be made to use more than one of its receive antennas and more than two of the timeslots. In particular, using more antennas is the key to achieving diversity gain.
- This coursework contributes 5% of your mark for the 15 credit module ELEC3204 Wireless and Optical Communications. You should therefore aim to spend about 7.5 hours on it. Please do not spend any more time than this unless you are enjoying yourself or you feel that it is helping your revision for the module.
- I would like to encourage you to *verbally* discuss your ideas with your classmates, since this can enhance your learning. However, you should not show each other your Matlab code, diagrams, equations or plots, since this counts as a breach of academic integrity.
- The mark that you receive for this coursework will be directly related to the highest of the six BERs that are achieved for your six users, when $1/N_0 = 5$ dB.
- I will measure your BERs at $1/N_0 = 5$ dB by running `main.m` on my own computer and getting it to call your `transmitter.m` and `receiver.m` functions. So, you should make sure that you submit Matlab functions that will work! Also, make sure that the average transmission energy per bit of each user $E_b^{(1)}$ is no greater than one, to avoid a mark penalty.
- I have tried to design this coursework so that the best BERs will be achieved by the students that invest the most effort. In particular, the students that achieve the best BERs are likely to be those that implement a variety of different solutions and then submit the best one.
- My model solution achieves BERs of 3.85×10^{-2} for all $K = 6$ users at $1/N_0 = 5$ dB. Even better BERs may be possible...