ESD5 – Fall 2024
Lecture Notes – Lecture 11

Department of Electronic Systems
Aalborg University

September 21, 2024

Example 1 – Repetition code and majority voting

When we introduced the Binary Symmetric Channel (BSC), we saw that it affects the communication process by flipping a transmitted bit with probability p_U, resulting in an erroneous bit decoding at the receiver. You might wonder if it is possible to decrease this error probability, especially when the value of p_U is high. The good news is that this is possible by adding redundancy to the symbols to be transmitted. The specific type of redundancy we will study in the following is called Forward Error Correction (FEC) code.

Initially, we consider that Zoya and Yoshi communicate over a BSC with error probability p_U. To send a single bit of information, Zoya adopts symbols that occupy 3 channel uses (c.u.), while each c.u. carries a repetition of the information. Hence, to transmit a bit "`1`", Zoya will send the codeword "`111`" and, to transmit a bit "`0`", she will send the codeword "`000`", as shown in Fig. 1. Yoshi uses majority voting to recover the original message, decoding the bits based on the most frequent bit present in each received codeword.

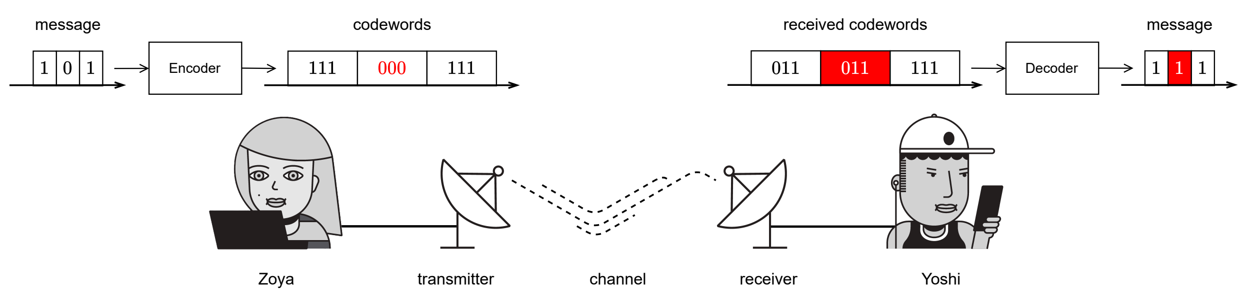

Figure 1: Example of coded transmission where Zoya uses the repetition code to send bits to Yoshi, which uses majority voting to decode the received codewords.

Accordingly, on the one hand, Yoshi will decode a bit "`0`" if the received codeword contains at least two `0`s. On the other hand, he will decode a "`1`" if the received codeword contains at least two `1`s. In the following, you can find the relationship between all potential outcomes of Yoshi's decoder based on the received codeword:

Received codeword	**Decoded bit**
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

As you might notice, combining the repetition code at Zoya and majority voting at Yoshi makes Yoshi capable of decoding the message successfully even when a single bit flip happens. The acquisition of this capacity impacts Yoshi positively by decreasing the Bit Error Rate (BER) he experiences. However, as we will see in the next, this advantage comes at the price of reducing the throughput.

To compute the BER at Yoshi, let us recall that, for each message bit, there are 4 possibilities for the received codeword that result in an erroneous decoding. For instance, if Zoya transmits the bit "`1`", the reception of the codewords "`000`", "`001`", "`010`", or "`100`" would lead to an erroneous decoding. From p_U, that is, the BER of the *uncoded communication*, the probabilities of Yoshi receiving each of these 4 codewords are respectively p_U^3, $p_U^2(1-p_U)$, $p_U(1-p_U)p_U$, and $(1-p_U)p_U^2$. Summing up these probabilities, we get the BER of the *coded communication*:

$$p_C = 3p_U^2(1 - p_U) + p_U^3. \tag{1}$$

Since p_C is always less than p_U, the proposed coding scheme results in fewer bit errors when compared to sending the bits over the BSC without encoding.

Now, to compute the throughput, let us assume that Zoya sends packets containing b information bits and c check bits, used for detecting whether the packet was successfully received. Considering that Zoya gets ideal feedback from Yoshi, the throughput achieved by the coded communication is

$$T_C = \frac{b}{3(b+c)}(1 - p_C)^{b+c}. \tag{2}$$

On the other hand, the throughput achieved by the uncoded communication is defined by

$$T_U = \frac{b}{b+c}(1 - p_U)^{b+c}. \tag{3}$$

Then, we can define T_C in terms of T_U,

$$T_C = T_U \frac{1}{3}\left(\frac{1 - p_C}{1 - p_U}\right)^{b+c}. \tag{4}$$

The last result indicates that the throughput of the coding scheme is higher only when the red term is greater than 1. The results in the following show that the coding scheme can obtain significant throughput gains when both the BER of the BSC and the number of bits to be transmitted are high. On the contrary, when the BSC is very reliable, the coding scheme achieves poor throughput values.

BSC BER (p_U)	**Number of bits** ($b+c$)	**Throughput**
0.495	100	$T_C = 0.549T_U$
0.495	200	$T_C = 0.895T_U$
0.495	300	$T_C = 1.47T_U$
10^{-6}	100	$T_C = 0.333T_U$

Example 2 – Hamming code

In the previous example, we saw how an intuitive scheme like the repetition code is capable of improving the BER at the cost of decreasing the throughput under certain conditions. Now, in this example, we will verify that more sophisticated schemes like the *Hamming code* can bring the same benefits as repetition code but with improved throughput.

First of all, let us start with the definitions. At the transmitter, a block coding scheme works by mapping codewords containing b bits into codewords containing $l > b$ bits. Accordingly, we characterize a block code by the tuple (l, b). For instance, the repetition code presented in Example 1 is a $(3, 1)$ code, as a message block containing a single bit is mapped to codewords containing 3 bits. Here, we will focus on the $(7, 4)$ Hamming code. Another important property of a code is the *code rate*, defined by

$$R = \frac{b}{l} \text{ [bit/c.u.]}. \tag{5}$$

From the code rate, you can see that the $(7, 4)$ Hamming code is superior to the $(3, 1)$ repetition code, as $\frac{4}{7}$ bit/c.u. $> \frac{1}{3}$ bit/c.u. We will consider that the message is denoted by

$$\mathbf{d} = \begin{bmatrix} d_1 & d_2 & \dots & d_b \end{bmatrix}. \tag{6}$$

Moreover, the codeword got from encoding the message is denoted by

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_l \end{bmatrix}. \tag{7}$$

Then, let us see how the encoding and decoding processes work for the Hamming code.

The Hamming code is a *linear block code*, meaning that the codeword can be obtained from the message $\mathbf{d}$ and the *generator matrix* $\mathbf{G}$ by evaluating

$$\mathbf{x} = \mathbf{dG}, \tag{8}$$

where, importantly, we consider **binary addition**. The receiver then obtains the codeword $\mathbf{y} = \mathbf{x} + \mathbf{e}$, where $\mathbf{e}$ is a binary vector denoting the error at the receiver. From $\mathbf{y}$ and the *parity-check matrix* $\mathbf{H}$, the receiver can check if the received codeword contains errors by evaluating the *syndrome vector*

$$\mathbf{s} = \mathbf{y}\mathbf{H}^T. \tag{9}$$

If $\mathbf{s} = \mathbf{0}$, then the received codeword is valid and does not need to be corrected. Otherwise, if $\mathbf{s}$ has at least one nonzero entry, the codeword is invalid and needs to be corrected by the receiver. To understand the error correction process, see Fig. 2. Specifically, Zoya and Yoshi are using the $(7, 4)$ Hamming code defined by the following generator and parity-check matrices:[1]

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

The communication process led to a single-bit flip in the codeword received by Yoshi. Luckily, as they are using a Hamming code, Yoshi can correct the error by flipping each bit of the received codeword at a time and verifying the syndrome vector,

$$\mathbf{s}_i = (\mathbf{y} + \mathbf{n}_i)\mathbf{H}^T, \tag{11}$$

where $\mathbf{n}_i$ is a binary vector filled with zeros, except for the i-th entry which has a 1. When Yoshi finds the i^* value that results in $\mathbf{s}_{i^*} = \mathbf{0}$, the codeword can be corrected by $\hat{\mathbf{x}} = \mathbf{y} + \mathbf{n}_{i^*}$,

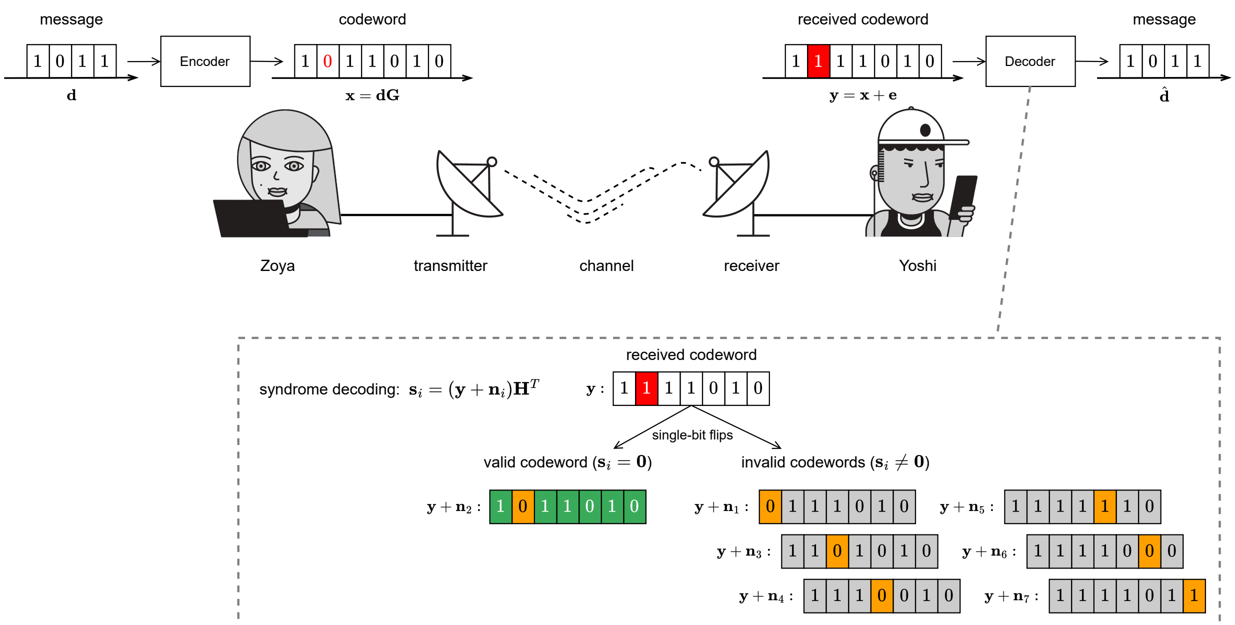

Figure 2: Example of coded transmission with the $(7, 4)$ Hamming code and syndrome decoding. The generator and parity-check matrices used by Zoya and Yoshi are in eq. (10).

[1]There are multiple $\mathbf{G}$ and $\mathbf{H}$ combinations that can be used to implement the $(7, 4)$ Hamming code. To see how these matrices are constructed, check https://en.wikipedia.org/wiki/Hamming_code#[7,4]_Hamming_code

while the message can be obtained by taking the 4 initial bits of the corrected codeword, that is, $\hat{\mathbf{d}} = [\hat{x}_1\ \hat{x}_2\ \hat{x}_3\ \hat{x}_4]$. Such a decoding method is called *syndrome decoding.* Notice in Fig. 2 that, at a Hamming distance equal to 1, there are 7 different codewords and, from all of them, only a single codeword is valid. Any other valid codeword will be at a Hamming distance greater than 1. To be precise, in the Hamming code, the minimum distance between valid codewords is 3. This is why this code is capable of correcting a single erroneous bit.

It is worth mentioning that the Hamming code is not restricted to the $(7,4)$ configuration. For $m \geq 3$ parity bits, we can define a $(m, 2^m - 1)$ Hamming code so that, in theory, we have an infinite number of possible configurations. Accordingly, by choosing $m = 9$, we get a $(511, 502)$ code with rate $\frac{502}{511} = 0.982$, almost twice the rate of the $(7,4)$ code, showing that we can get an even more improvement compared to the repetition code. Despite the gains in terms of code rate, by increasing the number of parity bits, we cannot increase the number of bits to be corrected by the Hamming code. As the minimum Hamming distance between valid codewords remains equal to 3, the code will be always capable of correcting a single bit.

Example 3–Maximum likelihood decoding

General coding idea

In this example, we first highlight how the coding idea leads to reliable communication. Consider the channel between Zoya and Yoshi, which has S identical input and output symbols as shown in Fig. 3, in which we only consider a single channel. An uncoded transmission over that channel is done by taking $\log_2 S$ data bits at a time, mapping them to one of the S symbols and transmitting them. Here, in the example of Fig. 3, the Yoshi can correctly recover the transmitted symbol with probability p and would mistakenly be classified with probability $1-p$. Specifically, the probability that a particular incorrect symbol is received is $\frac{1-p}{S-1}$. In this case, it does not leave us with many options to seek more reliable transmission.

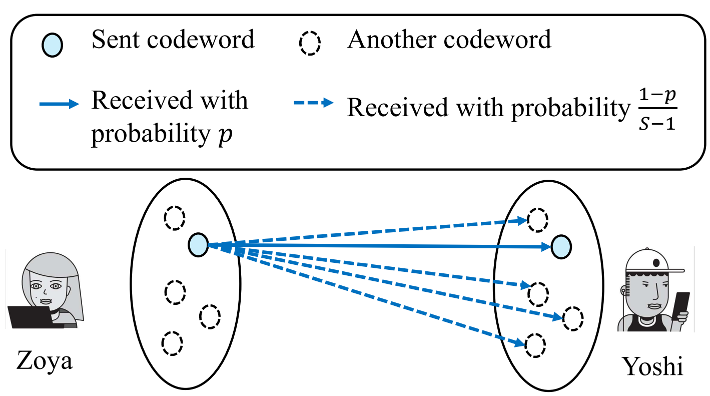

Figure 3: The original communication channel between Zoya and Yoshi with S inputs and identical set of S outputs.

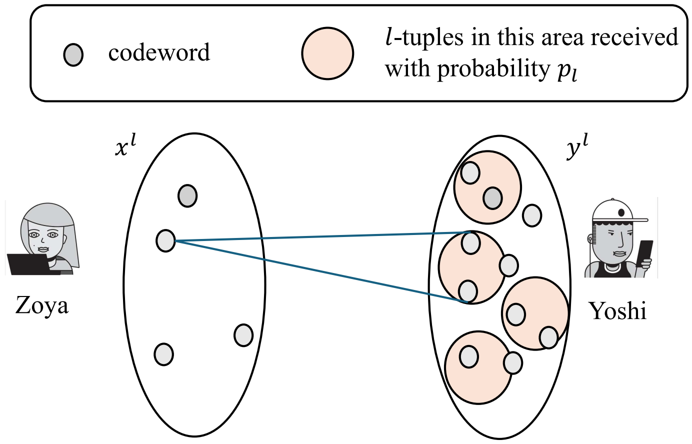

Figure 4: New channel between Zoya and Yoshi created by selection of l-tuplets that are codewords and have non-overlappimg shared areas.

Then, the input symbols that Zoya sends to Yoshi over the l-channel can be described as follows:

$$\mathbf{x} = (x_1, x_2, \ldots, x_l), \tag{12}$$

where each x_i is one of the S possible inputs. In uncoded transmission, any x_i can take any of the S values, such that the total number of possible vectors that can be sent over l channel uses is S^l. This is the best way to transmit data when there is no error. .

Now we consider the multiple channel setting. When a symbol over the l-channel is sent, there is a subset of outputs (l-tuplets) that are more likely to be received compared to the rest of the possible outputs. This increases the probability of successful detection compared with the single channel cases. Let us assume that the Zoya can select M different l-tuples whose shaded areas do not overlap each other as shown in Fig. 4. Each l-tuples is called a codeword and can be considered as a modulation symbol, which can carry $\log_2 M$ symbols. A decision rule that Yoshi can apply is the following. If he receives an l-tuples, then he searches the corresponding codeword $\hat{x}^l$. If he cannot find such a code, then he announces the erasure.

What is important is that for many meaningful channels, as l increases, the probability of error can be made arbitrarily small. In the next subsection, we will closely look at a more precise decision rule, called *Maximum Likelihood* (ML) detection.

Maximum likelihood detection

As we have seen so far, the communication channel could suffer from a variety of impairments, such as noise, attenuation, distortion, fading, and interference. In this example, we closely look at the decision rule at the receiver.

Let us consider a scenario where Zoya sends data to Yoshi through the communication channel, and Zoya and Yoshi have agreed beforehand that only M codewords, denoted as

$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M$ are valid, while there is total S^l possible channel inputs. In other words, there are S^l possible output $\mathbf{y}$ that Yoshi can observe. Yoshi needs to make a decision of the detected signal $\hat{\mathbf{x}}$ based on the received vector $\mathbf{y}$, and he wants to decide $\hat{\mathbf{x}}$ such that it is highly likely candidate that can produce the observed $\mathbf{y}$.

This corresponds to maximize the following probability:

$$\Pr(\mathbf{x}_i|\mathbf{y}) = \frac{\Pr(\mathbf{y}|\mathbf{x}_i)\Pr(\mathbf{x}_i)}{\Pr(\mathbf{y})}. \tag{13}$$

If each codeword is sent with equal probability, then the eq. (13) can be described as follows:

$$\Pr(\mathbf{x}_i|\mathbf{y}) = \frac{\Pr(\mathbf{y}|\mathbf{x}_i)}{M\Pr(\mathbf{y})}. \tag{14}$$

From this equation, we can see the maximization of $\Pr(\mathbf{x}_i|\mathbf{y})$ is equivalent to maximization of $\Pr(\mathbf{y}|\mathbf{x}_i)$. In general, $\Pr(\mathbf{y}|\mathbf{x}_i)$ is called likelihood of observing $\mathbf{y}$ when $\mathbf{x}$ is sent. For simplicity, let us consider the memoryless channel. Then, we can describe the likelihood as

$$\Pr(\mathbf{y}|\mathbf{x}_i) = \prod_{j=1}^{l} \Pr(y_j|x_{i,j}), \tag{15}$$

where $x_{i,j}$ is the input in the j-th channel use of the i-th codeword and the probabilities $\Pr(y_j|x_{i,j})$ are specified in a single use of the original channel. This is equivalent to maximizing the log-likelihood function described as:

$$\log \Pr(y_j|x_{i,j}) = \sum_{j=1}^{l} \Pr(y_j|x_{i,j}). \tag{16}$$

The final decision at Yoshi is made based on the maximum log-likelihood as shown below:

$$\hat{m} = \underset{1\le i\le M}{\arg\max}\ \log \Pr(y_j|x_{i,j}). \tag{17}$$

Example 4 – Error probability

We now analyze the error probability associated with the ML receiver structure. Here, we consider a scenario where Zoya sends a message over the AWGN channel to Yoshi. In the AWGN channel, the received signal can be described as $\mathbf{r}(\mathbf{t}) = \mathbf{s}(\mathbf{t}) + \mathbf{n}(\mathbf{t})$, where $\mathbf{n}(\mathbf{t})$ is white Gaussian noise process with mean 0 and power spectrum density $\frac{N_0}{2}$. The construction $\mathbf{s}_i$ based on the ML receiver can be decided based on the decision boundary Z_i. Here, the decision boundary Z_i can be described as

$$Z_i = (x : \|\mathbf{x} - \mathbf{s}_i\| < \|\mathbf{x} - s_j\|, \forall j = 1, \ldots, M, j \neq i). \tag{18}$$

Mathematically, the error probability can be described as

$$p_e = \sum_{i=1}^{M} \Pr(\mathbf{x} \neq Z_i \mid m_i \text{ sent}) \Pr(m_i \text{ sent}). \tag{19}$$

Let A_{ik} denote the event that $\|\mathbf{x}-\mathbf{s}_k\| < \|\mathbf{x}-\mathbf{s}_i\|$ under the condition that the constellation point $\mathbf{s}_i$ was sent. Here, when the event $A_{i,k}$ happens, then the estimated constellation is wrong, i.e., deemed as an error, as $\mathbf{s}_i$ is not the closest constellation point. Note that the final decision should be conducted considering all constellation points. As we indicated in the previous example, the conditions for the successful decoding are: $\|\mathbf{x}-\mathbf{s}_i\| < \|\mathbf{x}-\mathbf{s}_k\|, \forall k \neq i$. Then, we can describe the error probability as follows:

$$P_e(\mathbf{m}_i) = \Pr\left(\cup_{\substack{i=1 \\ k\neq i}}^{M} A_{ik}\right) \leq \sum_{\substack{i=1 \\ k\neq i}}^{M} \Pr\left(A_{ik}\right), \tag{20}$$

where the inequality follows from the union bound on probability.

Next, let us consider the $\Pr(A_{ik})$ in eq. (20). This probability can be described as

$$\begin{aligned} \Pr(A_{ik}) &= \Pr(\|\mathbf{s}_k - (\mathbf{s}_i + \mathbf{n})\| < \|\mathbf{s}_i - (\mathbf{s}_i + \mathbf{n})\|) \\ &= \Pr(\|\mathbf{s}_k - (\mathbf{s}_i + \mathbf{n})\| < \|\mathbf{n}\|). \end{aligned} \tag{21}$$

The event A_{ik} occurs if n is closer to $\mathbf{s}_i - \mathbf{s}_k$ than to 0, i.e., if $n > \frac{d_{ik}}{2}$. Fig. 5 illustrates the case where the receiver mistakenly classified the s_i symbol as s_k. Here, $d_{ik} = \|\mathbf{s}_i - \mathbf{s}_k\|$ is the distance between the constellation point s_i to s_k. Here Gaussian noise n is zero mean and variance $\frac{N_0}{2}$. Thus, its probability can be described as

$$\Pr(A_{ik}) = \Pr\left(n > \frac{d_{ik}}{2}\right) = \int_{\frac{d_{ik}}{2}}^{\infty} \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{u^2}{N_0}\right) du = Q\left(\frac{d_{ik}}{\sqrt{2N_0}}\right), \tag{22}$$

where $Q(\cdot)$ is the Q-function, defined as $Q(z) = \int_z^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx$.

Then, the eq. (20) can be described as

$$P_e(m_i) \leq \sum_{\substack{i=1 \\ k\neq i}}^{M} Q\left(\frac{d_{ik}}{\sqrt{2N_0}}\right). \tag{23}$$

Using eq. (23), we can derive the expected error probability as follows:

$$P_e = \sum_{i=1}^{M} \Pr(m_i) P_e(m_i) \leq \frac{1}{M} \sum_{i=1}^{M} \sum_{\substack{i=1 \\ k\neq i}}^{M} Q\left(\frac{d_{ik}}{\sqrt{2N_0}}\right). \tag{24}$$

where we assume all messages are equally selected and transmitted with probability $\frac{1}{M}$. This assumption will be held throughout this lecture note.

Let $d_{\min}$ be the minimum distance among all possible constellations, which can be defined as $d_{\min} = \min_{i,k} d_{i,k}$. Then, we could derive the looser bound of error probability as

$$P_e \leq (M-1) Q\left(\frac{d_{\min}}{\sqrt{2N_0}}\right). \tag{25}$$

Bit Error Probability for Binary Phase Shift Keying (BPSK)

BPSK is a modulation method in which one bit is mapped to one baseband symbol. In BPSK, the symbol and bit error rates are the same. Here, the transmitted signal is $s_1(t) = Ag(t)\cos(2\pi f_c t)$ for sending a bit 0, and $s_2(t) = -Ag(t)\cos(2\pi f_c t)$ for sending a bit 1, where $Ag(t)$ is lowpass equivalent form with amplitude A.

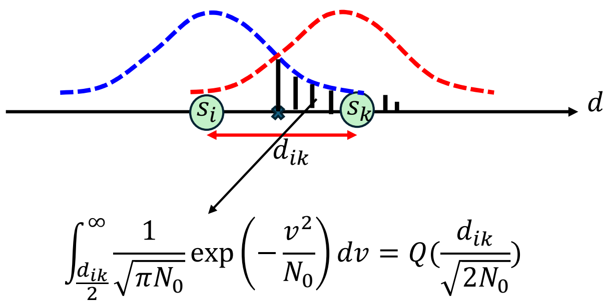

Figure 5: An example of regions where the wrong decision based on ML happens, where Yoshi mistakenly classifies the symbol s_i as s_k.

Based on the eq. (26), the probability error for BPSK can be described as

$$p_b = Q(\frac{d_{\min}}{\sqrt{2N_0}}), \tag{26}$$

where $d_{\min} = \|s_1 - s_0\| = \|A - (-A)\| = 2A$. Then, we want to connect A to energy-per-bit E_b, which can be described as

$$E_b = \int_0^{T_b} s_1(t)^2 dt = \int_0^{T_b} s_2(t)^2 dt = \int_0^{T_b} A^2 g(t)^2 \cos^2(2\pi f_c t) dt = A^2. \tag{27}$$

From this equation, we obtain $d_{\min} = 2A = 2\sqrt{E_b}$. Finally, we can derive the error probability for BPSK as follows:

$$P_b = Q(\frac{2\sqrt{E_b}}{\sqrt{2N_0}}) = Q(\sqrt{\frac{2E_b}{N_0}}) = Q(\sqrt{2\gamma_b}), \tag{28}$$

where $\gamma_b = \frac{E_b}{N_0}$ is a SNR per bit.

Error probability for QPSK

In QPSK, an example of the constellation in $\mathbb{R}^2$ are defined by $s_0 = (A, 0)$, $s_1 = (0, A)$, $s_3 = (-A, 0)$, and $s_4 = (0, -A)$. Note that any rotation of this constellation works equally well. In this case, the minimum distance can be expressed as $d_{\min} = d_{1,2} = d_{2,3} = d_{3,4} = d_{4,1} = \sqrt{A^2 + A^2} = \sqrt{2A^2}$. On the other hand, the distance towards the other constellation points, i.e., distance $d_{1,3}$ and $d_{2,4}$ can be $d_{1,3} = d_{2,4} = 2A$. Then, we can derive the bound of error probability as follows:

$$P_e \leq 2Q(\frac{A}{\sqrt{N_0}}) + Q(A\sqrt{\frac{2}{N_0}}), \tag{29}$$

where we use the symmetric property of each symbol.

As these two components, including the received signal, are orthogonal, the error probability for each branch is the same as that of the BPSK. However, the symbol error rate is different. Using the knowledge we have learned in lecture 5, the symbol error probability can be described as

$$p_{e,symbol} = 1 - (1 - Q(\sqrt{2\gamma_b}))^2 \tag{30}$$

As the symbol energy is spitted between the in-phase and quadrature branches, we have $\gamma_s = 2\gamma_b$, which gives us

$$p_{e,symbol} = 1 - (1 - Q(\sqrt{\gamma_s}))^2 \tag{31}$$

Furthermore, as $\gamma_s = 2\gamma_b = \frac{2A^2}{N_0}$, the union bound in eq. (29) can be

$$P_e \leq 2Q(\sqrt{\frac{\gamma_s}{2}}) + Q(\sqrt{\gamma_s}). \tag{32}$$

Error Probability for ASK/PAM signal

In this scheme, the minimum distance between any two points can be described as

$$d_{\min} = \sqrt{\frac{12\log_2 M}{M^2 - 1} E_{bave}}, \tag{33}$$

where the constellation points are located as $\{\pm\frac{1}{2}d_{\min}, \pm\frac{3}{2}d_{\min}, \ldots, \pm\frac{M-1}{2}d_{\min}\}$.

Here, we can observe that two types of points exist: one is $M - 2$ inner points, and the other is two points located in outer positions. The condition for the error is different:

- [Inner point]: error occurs if $|n| > \frac{1}{2}d_{\min}$, where n is zero-mean Gaussian random variable with variance $\frac{1}{2}N_0$.
- [Outer point]: the probability of error is one-half of the error probability of the inner point, as the outer point only has one nearest point.

Let us denote the error probabilities of inner points and outer points by p_{ei} and p_{eo}, respectively. Based on the above-mentioned condition, the error probability for p_{ei} and p_{eo} can be described as follows:

$$p_{ei} = \Pr\left[|n| > \frac{1}{2}d_{\min}\right] = 2Q(\frac{d_{\min}}{\sqrt{2N_0}}), \tag{34}$$

$$p_{eo} = \frac{1}{2}p_{ei} = Q(\frac{d_{\min}}{\sqrt{2N_0}}). \tag{35}$$

Then, the symbol error probability can be obtained by calculating the expectation as follows:

$$\begin{aligned} p_e &= \frac{1}{M}\sum_{i=1}^{M} \Pr[\text{error}|\text{symbol } m \text{ sent}] \\ &= \frac{1}{M}\left[(M-2)p_{ei} + 2p_{eo}\right] \\ &= \frac{1}{M}\left[(M-2)2Q(\frac{d_{\min}}{\sqrt{2N_0}}) + 2Q(\frac{d_{\min}}{\sqrt{2N_0}})\right] \\ &= \frac{2(M-1)}{M}Q(\frac{d_{\min}}{\sqrt{2N_0}}). \end{aligned} \tag{36}$$

By using eq. (33), it yields:

$$p_e = 2(1-\frac{1}{M})Q\sqrt{\frac{6\log_2 M}{M^2-1}\frac{E_{\text{bavg}}}{N_0}}. \tag{37}$$

Minimum Distance of PAM [2] Some people might wonder how one could obtain the eqs. (33) and (52). To give an idea, we derive the minimum distance for Pulse Amplitude Modulation (PAM) schemes.

In digital PAM, the waveforms may be expressed as follows:

$$s_m(t) = A_m p(t),\ 1 \le m \le M, \tag{38}$$

where A_m is the amplitude of m-th message and $p(t)$ is the pulse of duration T. The symbol amplitude A_m takes discrete values and can be expressed as:

$$A_m = 2m - 1 - M, \tag{39}$$

i.e., $A_m = \{\pm 1, \pm 3, \ldots, \pm(M-1)\}$.

The energy in the signal can be described as

$$E_m = \int_{-\infty}^{\infty} A_m^2 p^2(t)dt = A_m^2 E_p, \tag{40}$$

where E_p is the energy in $p(t)$. Then, the average energy can be calculated as

$$\begin{aligned} E_{\text{ave}} &= \frac{E_p}{M}\sum_{m=1}^{M} A_m^2 \\ &= \frac{2E_p}{M}\frac{M(M^2-1)}{6} \\ &= \frac{(M^2-1)E_p}{3}. \end{aligned} \tag{41}$$

[2]Further detail can be seen in John G. Proakis, "Digital Communications".

The average energy per bit is

$$E_{\text{bavg}} = \frac{(M^2-1)E_p}{3\log_2 M}. \tag{42}$$

The above description is for the baseband PAM signal. In practice, the PAM signal is the bandpass signal with lowpass equivalents of the form $A_m g(t)$. In this case

$$S_m = \text{Re}[A_m g(t) e^{j2\pi f_c t}] = A_m g(t)\cos(2\pi f_c t), \tag{43}$$

where f_c is the career frequency. By combining eq. (38) and (43), we obtain

$$p(t) = g(t)\cos(2\pi f_c t) \tag{44}$$

As the energy in the lowpass equivalent signal is twice the energy in the bandpass signal, energy for bandpass can be:

$$E_m = \frac{A_m^2}{2}E_g, \tag{45}$$

where E_g is the energy for $g(t)$. Likewise eqs. (41) and (42) can be expressed as

$$E_{\text{avg}} = \frac{(M^2-1)E_p}{6}. \tag{46}$$

$$E_{\text{avg}} = \frac{(M^2-1)E_p}{6\log_2 M}. \tag{47}$$

One-dimensional vector representation bandpass PAM signal can be described as

$$s_m = A_m\sqrt{\frac{E_g}{2}}, A_m = \pm 1, \pm 3, \pm(M-1) \tag{48}$$

Then, the Euclidean distance between any pair of signal points is

$$d_{mn} = \sqrt{\|s_m - s_n\|^2} = |A_m - A_n|\sqrt{\frac{E_g}{2}} \tag{49}$$

The minimum distance is the distance of adjacent points, which is 2, thus

$$d_{\text{min}} = \sqrt{2E_g} \tag{50}$$

Finally, by combining the equations above, we get

$$d_{\text{min}} = \sqrt{\frac{12\log_2 M}{M^2-1}E_{bave}}. \tag{51}$$

Due to the lack of space, we omit the details here, but the minimum distance for QAM can also be derived using the same approach mentioned above.

Error probability for QAM

In this scheme, the minimum distance between any points can be described as

$$d_{\min} = \sqrt{\frac{6\log_2 M}{M-1} E_{\mathrm{bavg}}}. \tag{52}$$

In M-ary QAM, there are $\sqrt{M}$ possible amplitude for in-phase and quadrature signals. From this observation, we can consider that M-ary QAM as two $\sqrt{M}$-ary PAM constellations in the in-phase and quadrature directions. Then, in QAM, the error occurs if either n_1 or n_2 is large enough to cause an error. The probability of successful detection, denoted as $P_s^{M-\mathrm{QAM}}$, for this QAM constellation can be calculated by considering the product of correct decision probabilities for constituent PAM systems, denoted as $P_s^{\sqrt{M}-\mathrm{PAM}}$

$$P_s^{M-\mathrm{QAM}} = \left(P_s^{\sqrt{M}-\mathrm{PAM}}\right)^2 = \left(1 - P_e^{\sqrt{M}-\mathrm{PAM}}\right)^2, \tag{53}$$

where $P_e^{\sqrt{M}-\mathrm{PAM}}$ is the symbol error probability for PAM system, in eq. (36). Then, the error probability for QAM, denoted as $P_e^{M-\mathrm{QAM}}$ can be described as

$$\begin{aligned} P_e^{M-\mathrm{QAM}} &= 1 - \left(1 - P_e^{\sqrt{M}-\mathrm{PAM}}\right)^2 \\ &= 2P_e^{\sqrt{M}-\mathrm{PAM}}\left(1 - \frac{1}{2}P_e^{\sqrt{M}-\mathrm{PAM}}\right). \end{aligned} \tag{54}$$

By combining eqs. (36), (52), and (54), we can derive the error probability for QAM as follows:

$$\begin{aligned} P_e^{M-\mathrm{QAM}} &= 4\left(1 - \frac{1}{\sqrt{M}}\right)Q\left(\frac{3\log_2 M}{M-1}\frac{E_{bavg}}{N_0}\right) \\ &\times \left(1 - \left(1 - \frac{1}{\sqrt{M}}\right) Q\left(\frac{3\log_2 M}{M-1}\frac{E_{bavg}}{N_0}\right)\right). \end{aligned} \tag{55}$$