POLITECNICO DI TORINO

Master of Science in Communications Engineering

COMMUNICATION SYSTEMS

REPORT: ASSIGNMENT 1

November 1, 2022

Luca Nepote (s315234)

1 Introduction

In the first assignment of the Master of Science course "Communication Systems" we have dealt with different topics, such as error detection, matrix encoding, generator polynomials, CRC ("Cyclic Redundancy Check") design, code distance profile, retransmission probability and parity check matrix.
The assignments' solutions have been solved in the Matlab environment.

2 Exercise 1: Error Detection by G and H

In the first exercise, we had to detect possible errors generated during a data transmission through a binary symmetric channel. The binary vectors considered had $k = 4$ bits (i.e. $\mathbf{v} \in \mathrm{H}^4$, where H^{k} is the vector space containing all the k-bit vectors), while the codewords have $n = 8$ bits (i.e. $\mathbf{c} \in \mathrm{H}^8$). The two generator matrices were:

$$G_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Notice that the difference between the two correspond to only one bit (the last one in the first row).

The first part consisted in the generation of the Codebook, which is defined as

$$\mathrm{C} = \left\{ \mathbf{c} = \mathbf{v}\mathbf{G}, \mathbf{v} \in \mathrm{H}^{\mathrm{k}} \right\}.$$

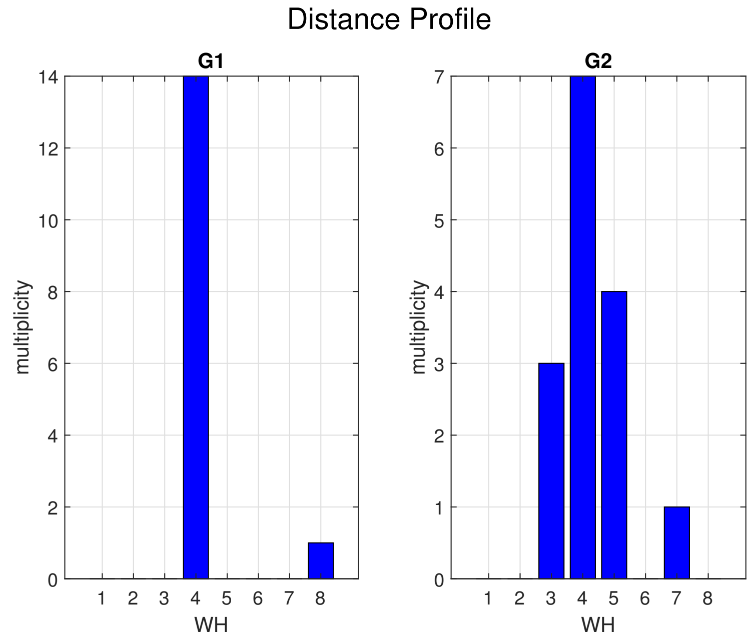
Doing this, we will be able to detect possible errors thanks to the presence of the redundancy bits.
Following the procedure given by the assignment, at first we generated all the $2^k - 1$ non-zero binary vectors and save them into a matrix. We then encoded each vector with the G_1 matrix and computed the "Hamming Weight" $W_H(\mathbf{v})$, which is defined as the number of bits equal to 1 of the codeword. In order to compute the undetected error probability $P(UE)$ with the analytical formula

$$P(UE) = \sum_{i=1}^{n} A_i p^i (1-p)^{(n-i)}, \tag{1}$$

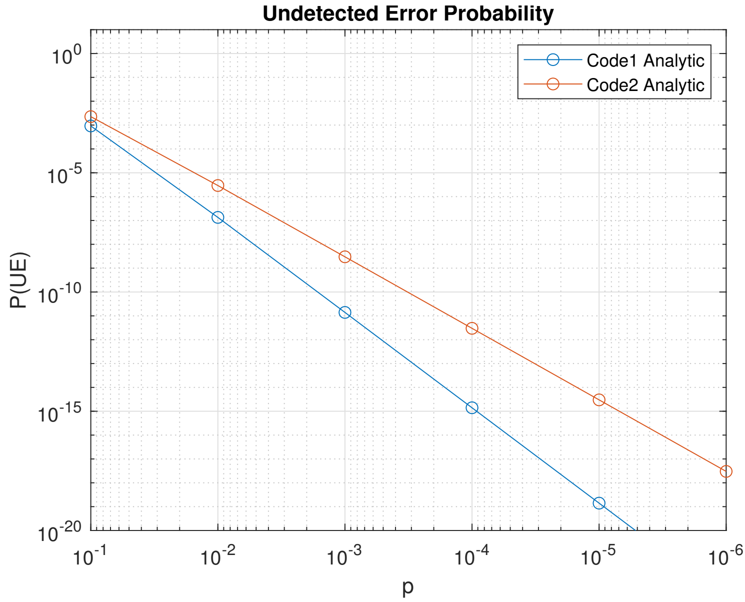
we had first to calculate the A_i, the number of codewords with weight $1 \leq i \leq n$. The probability has been computed for different values of p

$$p \in \left\{ 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} \right\}$$

and then represented on a $p - P(UE)$ chart. The same procedure has been followed for the generator matrix G_2.
As a results, we obtained



where for the first case the most of the codewords has the same Hamming weight equal to 4, whereas for the second they are more distributed.



.

Looking at the picture we can notice that the first generator matrix G_1 returns a better result with respect to G_2: indeed, the probability of undetected error is smaller for every p. This is due to the minimum distance obtained for each codebook: if d_{min} is bigger, in order to obtain an undetected error we must have more errors (i.e. transmitted $'0'$s received as $'1'$s), at least equal to the value of d_{min} , where, in terms of the Hamming weight,

$$d_{min} = \min W_H(\mathbf{c}), \text{ with } \mathbf{c} \in C. \tag{2}$$

In the considered case, the minimum distance achieved with the first generator matrix is bigger with respect to the second one, so the codewords are received correctly more in the first case. More precisely, for G_1 we obtained a $d_{min} = 4$, whereas using G_2 the value is equal to 3.

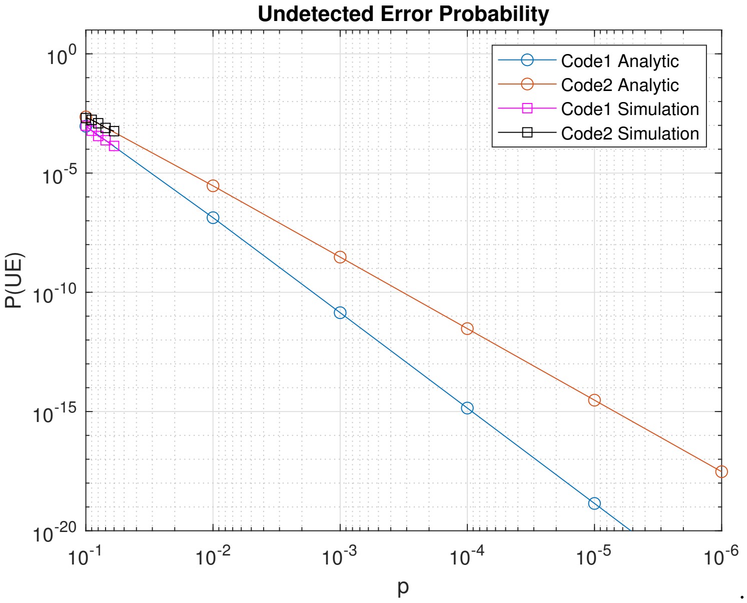
In the next step, we created the two corresponding parity check matrices H_1 and H_2, where

$$\text{H}=\begin{bmatrix} P \\ I \end{bmatrix}.$$

Later, we generated a k-bit binary vector, which has been encoded with G_1 and sent through a binary symmetric channel with probability that an error occurs equal to $p \in p_s$, with $p_s = [0.1, 0.09, 0.08, 0.07, 0.06]$ and $(1 - p)$ to send the correct bit value.

We then compute the syndrome, defined as $\mathbf{s} = \mathbf{y}\text{H}$: if the result was equal to the all zero vector, we checked if an undetected error occurred (i.e. if the codeword and the passed vector are different even if they belong to the same codebook).

Finally, the probability $P(UE)$ has been computed and represented on the previous picture, in order to compare the analytical curve with the simulated one. Repeating also for the second case (i.e. G_2) we obtain the following results.



.

We can note that the simulating results are coherent with the ones predicted by the analytical formula, so the procedure has been followed correctly. Even in the simulation, the first case with G_1 works better than the second one, due to the obtained d_{min} .

In conclusion, the choice of the generator matrix is very important to have the smaller possible undetected error probability. For example, in the considered case, changing only one bit in the whole generator matrix gives results very different: for $p = 10^{-5}$ we pass from $P(UE) \sim 10^{-14}$ to $P(UE) \sim 10^{-19}$.

3 Exercise 2: Generator Polynomial and Code Properties

In the second part of the assignment we had to encode a codebook H^k, with k increasing from 2 to 8, using the generator polynomial $g(D)$, defined as

$$g(D) = p(D)(D + 1). \tag{3}$$

Notice that, in this case, $p(D)$ is a primitive polynomial of degree 3:

$$p(D) = D^3 + D + 1.$$

Indeed, we know that each binary vector can be represented with a polynomial expression. Generally,

$$\mathbf{v} = [v_1\ v_2 ...\ v_n] \rightarrow P(D) = v_1 D^{n-1} + v_2 D^{n-2} + ... + v_n.$$

The same properties of the binary vectors hold for the polynomial representation too, such as the sum or the multiplication. In order to encode a binary vector, we have to multiply it for $g(D)$.

$$c(D) = v(D)g(D),$$

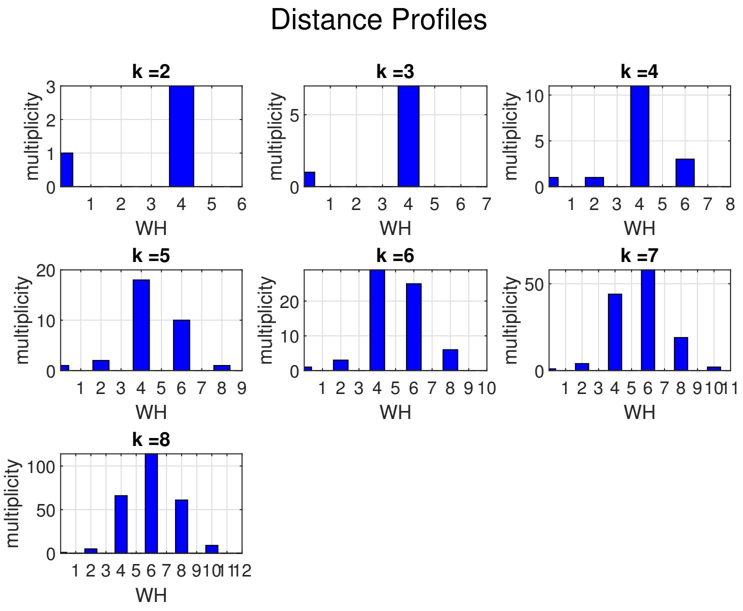
where the three terms have maximum degree respectively equal to $n-1$, $k-1$ and r.
Encoding in this may makes the codebook non systematic. To obtain a systematic one, we have at first to shift the initial binary vectors by r positions to the left. Then, the result is divided by $g(D)$, obtaining that the codeword is still a multiple of the generator polynomial $g(D)$.

$$v'(D) = v(D)D^r$$

$$v'(D) = q(D)g(D) + r(D) \quad \text{q(D): quotient,} \;\; \text{r(D): reminder}$$

$$c(D) = v'(D) + r(D) = v(D)D^r + r(D) = q(D)g(D)$$

The distance profiles have been represented in the next picture, where on the x-axis there are the Hamming weights and on the y one the number of codewords.



Notice that:

- every case includes $A_0 = 1$, corresponding to the all zero vector;

- all the codewords have even weight, due to the presence of the $(D+1)$ term in the generator polynomial. Indeed, $D = 1$ is a root of $g(D)$ and so of $c(D)$ too;
- for $k = 2$ and $k = 3$ there are not codewords with weight smaller than 4 (i.e. there are not $W_H = 2$ vectors). This is due to the presence of the third degree primitive polynomial $p(D) = D^3 + D + 1$: if $p(D)$ has degree x, any polynomial $D^y + 1$, with $y < 2^x - 1$, cannot be divided by $p(D)$. Moreover, any codeword is a multiple of the $g(D)$, but using $p(D)$ as a factor, all the $D^y + 1$, with $y < 2^x - 1$, cannot be codewords. In this case $x = 3$, so $y = 2^3 + 1 = 7$. Considering $k = 2, 3$, the value of $n = k + r$ is respectively $n = 6$ and $n = 7$, where $r = 4$ is the degree of $g(D)$, so the H^k cannot contain weight 2 codewords. For the successive values, such as $k = 4, 5...$ this does not hold, because $n > y$;
- increasing the k value, the distance profile assumes a gaussian distribution, thanks to the available number of permutations for the $'1'$ bit positions.

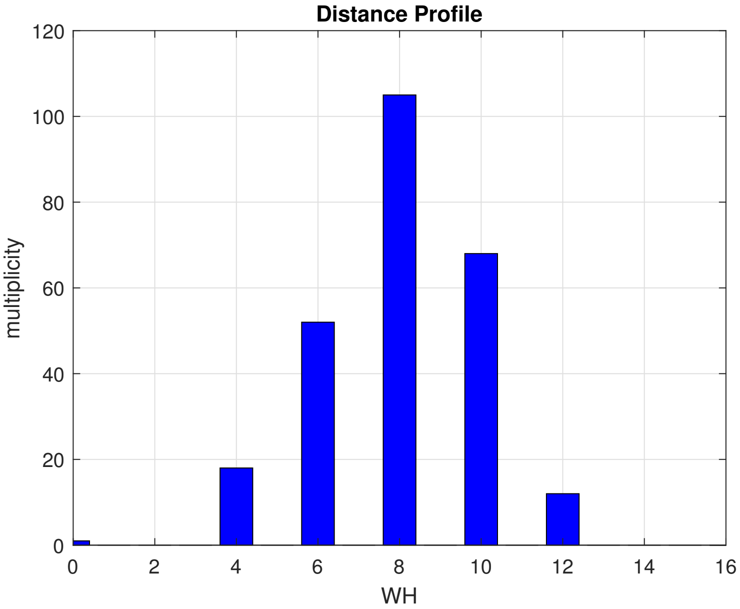
4 Exercise 3: CRC Design and Performance Evaluation

In this section we wanted to design a Cyclic Redundancy Check with 8 redundancy bits. The error vectors that must be corrected have weight 1, 2 and 3, for $k \leq 100$.
In order to do that, we have to correctly choose the generator polynomial: considering what said in the previous section, the presence of $g(D)$ makes the codewords to have an even number of $'1'$s. The goal is then to neutralize the $W_H(e) = 2$ vectors: with $n = k + r \leq 100 + 8 = 108$, we must have y bigger than n. Choosing $x = 7$ as the degree of the primitive polynomial $p(D)$, the result is $y = 2^7 - 1 = 127$, which verifies the condition.
For example, we choose

$$p(D) = D^7 + D^3 + D^2 + D + 1$$

$$g(D) = (D+1)g(D) = D^8 + D^7 + D^4 + 1$$

After fixing the value of $k = 8$, we encoded the whole H_k using $g(D) = P(D)(D+1)$. Then, the distance profile has been computed.



As we can note, the all zero vector is included (i.e. $A_0 = 1$) and the codewords contain only an even number of $'1'$, thanks to the presence of the $(D+1)$ factor in the generator polynomial. Observe also that the profiles assumes a gaussian distribution.

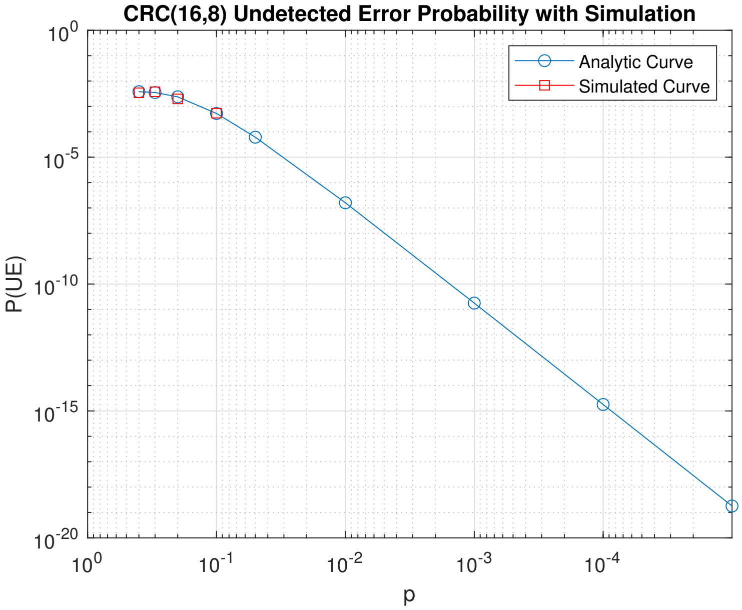
Following the procedure given by the assignment, we plotted the analytical curve representing the undetected error probability and we compared it with the simulated one, for different value of p.
Notice that for the analytical curve we considered

$$p \in \left\{4 \cdot 10^{-1}, 3 \cdot 10^{-1}, 2 \cdot 10^{-1}, 10^{-1}, 5 \cdot 10^{-2}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\right\},$$

whereas for the simulated one

$$p \in \left\{4 \cdot 10^{-1}, 3 \cdot 10^{-1}, 2 \cdot 10^{-1}, 10^{-1}\right\}.$$

Moreover, for the simulation, the same procedure of the first exercise has been followed.
The results obtained are represented in the following picture, where the points represent the analytic result and the squares the simulated ones.



As we can see, the simulation is coherent with the analytical law. Moreover, with smaller value of p, the undetected error probability becomes very small, of the 10^{-19} order.

5 Question 1: Distance Profile

The distance profile is defined as the set of 2^n distances computed with respect to a fixed codeword $\mathbf{c}$, where the "distance" corresponds to the number of bits that are different between the two. For a generic code $C(n, k)$ encoded linearly we would to prove that the distance profile is independent from the choice of the codeword with respect to which we compute the distance.

$$A_i = |\{\mathbf{c} \in C : d_H(\mathbf{c}, \mathbf{c}^*) = i\}| \text{ with } 0 \leq i \leq n$$

In order to do that, we used two properties of the Codebook and the Hamming distance.
The first one is that, whenever we sum to both the vectors another codeword of the Codebook, the Hamming distance between the two does not change:

$$d_H(\mathbf{v_1}, \mathbf{v_2}) = d_H(\mathbf{v_1} + \mathbf{v}, \mathbf{v_2} + \mathbf{v}) \ \forall \mathbf{v_1}, \mathbf{v_2}, \mathbf{v} \in H^k \tag{4}$$

The second property useful for the proof is the following one: considering a codebook C and summing to all its codewords a codeword belonging to C, we obtain the same Codebook but in a different order.
Combining the two properties together we can state that the Hamming distance does not depend on the choice of the first codeword, because if we sum another

one we obtain only the same vectors in a different order, but thanks to the first property, the Hamming distance remains the same.

$$d_H(\mathbf{c},\mathbf{c}^*) = d_H(\mathbf{c}+\mathbf{c}',\mathbf{c}^*+\mathbf{c}') = d_H(\mathbf{c}^{"},\mathbf{c}^{"*})\ \forall \mathbf{c},\mathbf{c}^*,\mathbf{c}',\mathbf{c}^{"*} \in H^k$$

Thanks to this important result, we can always consider the distance with respect to the all zero vector: Indeed, we know that if we sum to a codeword the vector itself, we obtain the all zero vector, but the Hamming distance does not change.

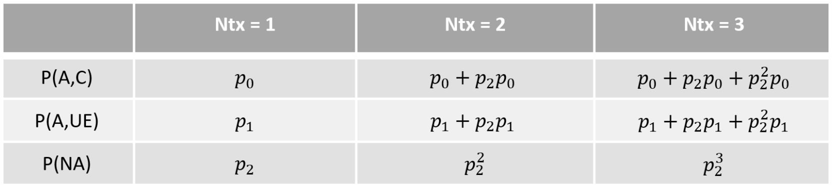
$$d_H(\mathbf{c},\mathbf{c}^*) = d_H(\mathbf{c}+\mathbf{c},\mathbf{c}^*+\mathbf{c}) = d_H(\mathbf{0},\mathbf{c}'^*)\ \forall \mathbf{c},\mathbf{c}^*,\mathbf{c}'^* \in H^k.$$

6 Question 2: Retransmissions Probability

In this section we have to discuss the retransmission probability. We considered $\mathrm{N_{TX}}$ as the maximum number of transmissions and we want to compute, for a generic $\mathrm{N_{TX}} = i$, the $P(A,C)$,$P(A,UE)$ and $P(NA)$, respectively the probability that a codeword is accepted and it's correct, that it's accepted but it's an undetected error and the probability that is not accepted. They cover all the possible cases, so their sum must be equal to 1 for each value of $\mathrm{N_{TX}}$ due to the property of probability distribution.
We also know that, for $\mathrm{N_{TX}} = 1$, we have

$$1 = p_0 + p_1 + p_2, \tag{5}$$

with $p_0 = P(A,C)$, $p_1 = P(A,UE)$ and $p_2 = P(NA)$.
For the cases with $\mathrm{N_{TX}} = 1, 2, 3$ is reported the direct computation:



	Ntx = 1	Ntx = 2	Ntx = 3
P(A,C)	p_0	$p_0 + p_2p_0$	$p_0 + p_2p_0 + p_2^2p_0$
P(A,UE)	p_1	$p_1 + p_2p_1$	$p_1 + p_2p_1 + p_2^2p_1$
P(NA)	p_2	p_2^2	p_2^3

Retransmissions Probabilities for Ntx=1, 2 and 3

The generic expressions for the three probability are:

$$P(A,C) = \sum_{n=1}^{\mathrm{N_{TX}}} p_0 p_2^{n-1}; \tag{6}$$

$$P(A,UE) = \sum_{n=1}^{\mathrm{N_{TX}}} p_1 p_2^{n-1}; \tag{7}$$

$$P(NA) = p_2^{\mathrm{N_{TX}}}. \tag{8}$$

We want now to prove that their sum is equal to 1 for all the $\mathrm{N_{TX}}$s (i.e. $P(A,C)+P(A,UE)+P(NA)=1$). Combining the three expressions computed before we obtain:

$$\begin{aligned}
&\sum_{n=1}^{\mathrm{N_{TX}}} p_0 p_2^{n-1} + \sum_{n=1}^{\mathrm{N_{TX}}} p_1 p_2^{n-1} + p_2^{\mathrm{N_{TX}}} = (p_0+p_1)\sum_{n=1}^{\mathrm{N_{TX}}} p_2^{n-1} + p_2^{\mathrm{N_{TX}}} = \\
&(1-p_2)\sum_{n=0}^{\mathrm{N_{TX}}-1} p_2^{n} + p_2^{\mathrm{N_{TX}}} = \sum_{n=0}^{\mathrm{N_{TX}}-1} p_2^{n} - \sum_{n=0}^{\mathrm{N_{TX}}-1} p_2^{n+1} + p_2^{\mathrm{N_{TX}}} = \\
&\sum_{n=0}^{\mathrm{N_{TX}}} p_2^{n} - \sum_{n=0}^{\mathrm{N_{TX}}-1} p_2^{n+1} = \sum_{n=0}^{\mathrm{N_{TX}}} p_2^{n} - \sum_{n=1}^{\mathrm{N_{TX}}} p_2^{n} = \\
&p_2^0 + \sum_{n=1}^{\mathrm{N_{TX}}} p_2^{n} - \sum_{n=1}^{\mathrm{N_{TX}}} p_2^{n} = p_2^0 = 1,
\end{aligned} \tag{9}$$

where we exploited the property (5) and the changing in the indices summation.

7 Question 3: Parity Check Matrix

Considering a code $C(n,k)$, with $r=n-k$, the generator matrix $G=[I_k\ P]$ and the parity check matrix $H=\begin{bmatrix}P\\I_r\end{bmatrix}$, we wanted to demonstrate that the parity check matrix of C is not unique. We also know that the syndrome $\mathbf{s}=\mathbf{y}\mathrm{H}$ is equal to zero if and only if the codeword $\mathbf{y}$ belongs to the codebook C. For the proof we considered the case with $r=3$ and $H=[col1\ col2\ col3]$, where each col_i is a column vector composed by n bits. To obtain $\mathbf{s}=0$ we must have that each product between the codeword and the column gives zero as result, so we are sure that

$$\mathbf{s}=[\mathbf{y}\cdot\mathbf{col_1}\quad \mathbf{y}\cdot\mathbf{col_2}\quad \mathbf{y}\cdot\mathbf{col_3}]=\mathbf{0} \tag{10}$$

However, by construction the columns are linearly independent, so, in order to obtain a zero vector, each product must be equal to zero.

$$\begin{aligned}
\mathbf{y}\cdot\mathbf{col_1}&=0\\
\mathbf{y}\cdot\mathbf{col_2}&=0\\
\mathbf{y}\cdot\mathbf{col_3}&=0
\end{aligned}$$

Considering three different matrices, we would understand if these are still acceptable parity check matrices for the codebook C. The first one is:

$$H_1=[col_2\ col_3\ col_1]$$

where the columns have been only permutated. The single products do not change and the previous considerations hold also in this case.

$$\mathbf{s} = [\mathbf{y} \cdot \mathbf{col_2} \quad \mathbf{y} \cdot \mathbf{col_3} \quad \mathbf{y} \cdot \mathbf{col_1}] = \mathbf{0}$$

So, the H_1 matrix is still a parity check matrix for the codewords belonging to C.
The second considered matrix is:

$$H_2 = [(col_1 + col_2) \; col_2 \; col_3].$$

Computing the product and applying the distributive property:

$$\mathbf{s} = [\mathbf{y} \cdot (\mathbf{col_1} + \mathbf{col_2}) \quad \mathbf{y} \cdot \mathbf{col_2} \quad \mathbf{y} \cdot \mathbf{col_3}] = \mathbf{0}$$
$$\mathbf{s} = [\mathbf{y} \cdot \mathbf{col_1} + \mathbf{y} \cdot \mathbf{col_2} \quad \mathbf{y} \cdot \mathbf{col_2} \quad \mathbf{y} \cdot \mathbf{col_3}] = \mathbf{0}$$

However, we know form the original matrix H that each product is equal to zero, so H_2 is still admissible.
The last case considered is the following one.

$$H_2 = [(col_1 + col_2) \; (col_1 + col_2) \; col_3].$$

We followed the same procedure of the second case:

$$\mathbf{s} = [\mathbf{y} \cdot (\mathbf{col_1} + \mathbf{col_2}) \quad \mathbf{y} \cdot (\mathbf{col_1} + \mathbf{col_2}) \quad \mathbf{y} \cdot \mathbf{col_3}] = \mathbf{0}$$
$$\mathbf{s} = [\mathbf{y} \cdot \mathbf{col_1} + \mathbf{y} \cdot \mathbf{col_2} \quad \mathbf{y} \cdot \mathbf{col_1} + \mathbf{y} \cdot \mathbf{col_2} \quad \mathbf{y} \cdot \mathbf{col_3}] = \mathbf{0}$$

Each product is equal to zero: H_3 is admissible too.
If we take a vector $\mathbf{y} \notin C$, the result will not be zero and the previous consideration do not hold anymore.

We proved that the parity check matrix is not unique and, for linear combination or permutation of the columns, the matrix is still admissible. This is not true if we permute the rows of H: indeed, for rows permutation, the matrix properties are changed and the new H will be the parity check matrix of a new codebook.

7.1 Example

For example, we consider the $k \times n$ generator matrix

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

In this case $k \neq r$ and the $n \times r$ H matrix is

$$H = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Considering a codeword (for example $\mathbf{y} = [0\ 1\ 0\ 0\ 1\ 0\ 0]$), we compute the syndrome for H, H_1, H_2, H_3.

$$\mathbf{s} = \begin{bmatrix}0\\1\\0\\0\\1\\0\\0\end{bmatrix} \begin{bmatrix}0&1&1\\1&0&0\\0&1&0\\0&0&1\\1&0&0\\0&1&0\\0&0&1\end{bmatrix} = \begin{bmatrix}0&0&0\end{bmatrix}; \qquad \mathbf{s}_1 = \begin{bmatrix}0\\1\\0\\0\\1\\0\\0\end{bmatrix} \begin{bmatrix}1&1&0\\0&0&1\\1&0&0\\0&1&0\\0&0&1\\1&0&0\\0&1&0\end{bmatrix} = \begin{bmatrix}0&0&0\end{bmatrix};$$

$$\mathbf{s}_2 = \begin{bmatrix}0\\1\\0\\0\\1\\0\\0\end{bmatrix} \begin{bmatrix}1&1&1\\1&0&0\\1&1&0\\0&0&1\\1&0&0\\1&1&0\\0&0&1\end{bmatrix} = \begin{bmatrix}0&0&0\end{bmatrix}; \qquad \mathbf{s}_3 = \begin{bmatrix}0\\1\\0\\0\\1\\0\\0\end{bmatrix} \begin{bmatrix}1&1&1\\1&1&0\\1&1&0\\0&0&1\\1&1&0\\1&1&0\\0&0&1\end{bmatrix} = \begin{bmatrix}0&0&0\end{bmatrix}.$$

If instead, a row permutation is considered (for example between the first and the second row):

$$\mathbf{s}_{\mathrm{r}} = \begin{bmatrix}0\\1\\0\\0\\1\\0\\0\end{bmatrix} \begin{bmatrix}1&0&0\\0&1&1\\0&1&0\\0&0&1\\1&0&0\\0&1&0\\0&0&1\end{bmatrix} = \begin{bmatrix}1&1&1\end{bmatrix} \neq \begin{bmatrix}0&0&0\end{bmatrix}.$$