**ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE**
School of Computer and Communication Sciences

**Handout ?**
Introduction to Communication Systems

**Homework 5**
October 16, 2008

**Problem 1.**

1. 
\[
\begin{pmatrix}
1 & 2 & 3 \\
2 & 2 & 2 \\
3 & 3 & 1
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3
\end{pmatrix}
= 
\begin{pmatrix}
2 \\
1 \\
0
\end{pmatrix}
\]

2. Over the integers: \(\text{det}(A) = 1(2 - 6) - 2(2 - 9) + 3(4 - 6) = 4\).
   Over \(\mathbb{F}_7\): \(\text{det}(A) = 1(2 + 1) + 5(2 + 5) + 3(4 + 1) = 3 + 0 + 1 = 4\).

3. We first concatenate the vector \(u\) to the matrix \(A\). Then we perform gaussian elimination:

\[
\begin{pmatrix}
1 & 2 & 3 & 2 \\
2 & 2 & 2 & 1 \\
3 & 3 & 1 & 0
\end{pmatrix}
\rightarrow 
\begin{pmatrix}
1 & 2 & 3 & 2 \\
0 & 5 & 3 & 4 \\
3 & 3 & 1 & 0
\end{pmatrix}
\rightarrow 
\begin{pmatrix}
1 & 2 & 3 & 2 \\
0 & 5 & 3 & 4 \\
0 & 0 & 4 & 3
\end{pmatrix}
\]

Thus:

\[
x_3 = 3 \implies 2 \cdot 4x_3 = 2 \cdot 3 \implies x_3 = 6,
\]

\[
5x_2 = 4 - 4 = 0 \implies x_2 = 0,
\]

\[
x_1 = 2 - 4 = 5.
\]

4. You need \(n^3\) operations.

**Problem 2.**

1. Since we are working over \(\mathbb{F}_2\), \((u_i + v_i) \in \{0, 1\}\), for all \(1 \leq i \leq n\). Thus \(d(u, v) = \sum_{i=1}^{n} (u_i + v_i) \geq 0\). Moreover in order to have \(d(u, v) = 0\), we must have \((u_i + v_i) = 0\), for all \(1 \leq i \leq n\) and vice versa. Which means \(u_i = v_i\), for all \(1 \leq i \leq n\). Thus \(d(u, v) = 0\) if and only if \(u = v\).

2. Since the sum is commutative, \(d(u, v) = d(v, u)\).

3. \(d(u, w) + d(w, v) = \sum_{i=1}^{n} (u_i + w_i) + \sum_{i=1}^{n} (w_i + v_i) = \sum_{i=1}^{n} (2w_i + u_i + v_i) \geq \sum_{i=1}^{n} (u_i + v_i) = d(u, v)\), where the inequality comes from the fact that \(w_i \geq 0\).

**Problem 3.**

1. In order to be a subspace, \(S^\perp\) has to satisfy 3 conditions: \(0 \in S^\perp\), \(S^\perp\) is closed under addition and \(S^\perp\) is closed under scalar multiplication. Let us show that it is indeed the case:

- Let \(w = 0\) and \(s \in S\). We have \(w \cdot s = \sum_i w_is_i = 0\), since \(w_i = 0\), \(\forall i\). Thus, \(w = 0 \in S^\perp\).
- Let \(w \in S^\perp\) and \(s \in S\). \((v + w) \cdot s = \sum_i (v_i + w_i)s_i = \sum_i v_is_i + \sum_i w_is_i = v \cdot s + w \cdot s = 0\). Thus \(v + w \in S^\perp\).
• Let \( c \in F \), \( w \in S^\perp \) and \( s \in S \). \( cw \cdot s = \sum_i cw_i s_i = c \sum_i w_i s_i = c(w \cdot s) = 0. \)

Thus \( cw \in S^\perp \).

2. To belong to \( S^\perp \), \( w \) has to satisfy: \( w_1 + w_2 = 0, w_3 + w_4 = 0 \). Thus \( S^\perp = \{0000, 0011, 1100, 1111\} = S \).

**Problem 4.**

1. No.

2. The code generates \( 2^4 = 16 \) codewords:
   \[
   (0, 0, 0, 0, 0, 0), (1, 0, 0, 0, 1, 1), (0, 1, 0, 0, 1, 1), (1, 1, 0, 0, 1, 1),
   (0, 0, 1, 1, 1, 1), (1, 0, 1, 1, 0, 0), (0, 1, 1, 0, 0, 1), (1, 1, 1, 0, 0, 0),
   (0, 0, 1, 1, 0, 0), (1, 0, 1, 1, 0, 0), (0, 1, 1, 1, 0, 0), (1, 1, 1, 1, 1, 1).
   \]

3. The transmitted codeword was \( (0, 0, 0, 1, 1, 0, 1) \). We can remove up to \( 2 = d_{\text{min}} - 1 \) bits of the codeword and still be able to find what was the transmitted codeword.

4. \( C^\perp = \{\tilde{c} \in \{0, 1\}^7 : \tilde{c} \cdot c = 0, \forall c \in C\} \)
   \[= \{(0, 0, 0, 0, 0, 0, 0), (0, 1, 1, 1, 1, 0, 0), (1, 1, 1, 0, 0, 1, 0), (1, 0, 1, 1, 0, 0, 1),
   (1, 0, 0, 1, 1, 0, 0), (1, 1, 0, 1, 0, 1, 0), (0, 1, 0, 1, 0, 1, 1), (0, 0, 1, 0, 1, 1, 1)\}.
   \]

5. Since \( C \) is systematic, \( G = (I_k, P) \), the parity-check matrix is simply \( H = (P^T, I_{n-k}) \), i.e.,
   \[
   H = \begin{pmatrix}
   0 & 1 & 1 & 1 & 1 & 0 & 0 \\
   1 & 1 & 1 & 0 & 0 & 1 & 0 \\
   1 & 0 & 1 & 1 & 0 & 0 & 1
   \end{pmatrix}.
   \]

6. \( s = (1, 0, 0) \), which corresponds to the 5th column of \( H \). Since we know that there is only one error, the error vector should be \( (0, 0, 0, 0, 1, 0, 0) \). Thus the transmitted codeword should be \( (0, 0, 0, 1, 1, 0, 1) \).

7. Smallest number of errors is \( 3 = d_{\text{min}} \).

**Problem 5.** We are giving two different solutions. The first one follows the hint.

1. Let \( d = d_1 + d_2 \), where \( d \) is the minimum distance between two codewords and \( d_1 \) is the disance among the \( k \) first bits and \( d_2 \) the distance among the \( n - k \) last bits. We consider the binary matrix of size \( (2^k) \times n \) formed by the \( 2^k \) codewords. We remove the \( n - k \) last columns. It remains \( 2^k \) words of length \( k \) and the minimum distance between two of these words is thus \( d_1 \leq 1 \). Moreover \( d_2 \leq n - k \), since it is the distance among \( n - k \) bits. Combining everything, we have \( d = d_1 + d_2 \leq 1 + n - k \).

2. The code contains \( 2^k \) codewords. Assume we remove the \( d - 1 \) last bits of all codewords. We get \( 2^k \) words of length \( n - d + 1 \) which are all distinct, since the minimum distance is \( d \) and we have removed only \( d - 1 \) bits. The maximum number of words of length \( n - d + 1 \) is \( 2^{n-d+1} \). Thus \( 2^k \leq 2^{n-d+1} \), which is equivalent to \( d \leq n - k + 1 \).