09.01.2013

# Linear Algebra - Course 13

# Chapter 4. Introduction to Linear Codes

# Part II

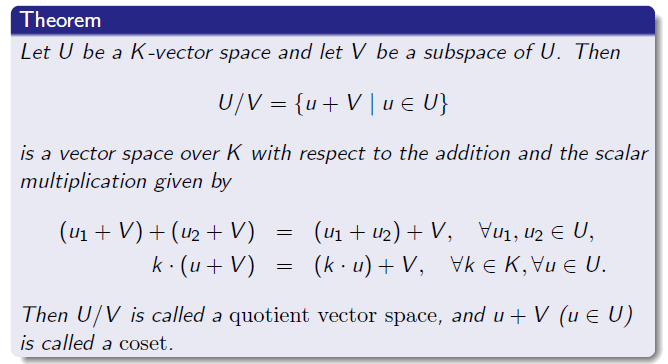
**Error Correcting/Decoding**

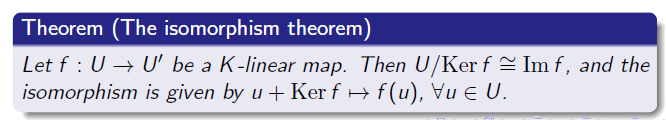
A naive method:

* Given a received word, compute all Hamming distances to the code words. (Recall that the Hamming distance between two words of the same length is the number of positions in which they difer.)
* The code word closest to the received word will be assumed to be the most likely transmitted word.

Not practical!

**Intermezzo: Quotient Vector Spaces**





**Coset Leaders**



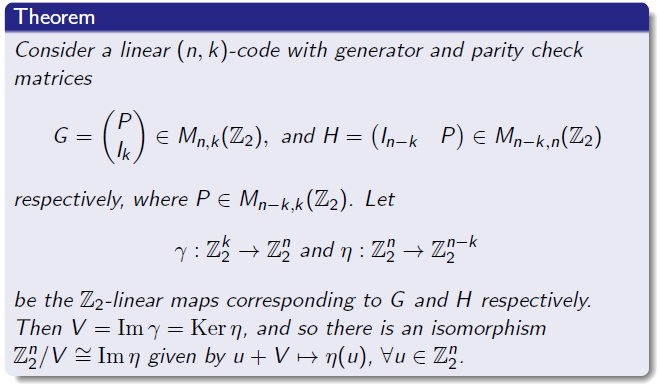
Consider an (n, k) - code with encoding function: and denote V = Imɣ (the subspace of code vectors).

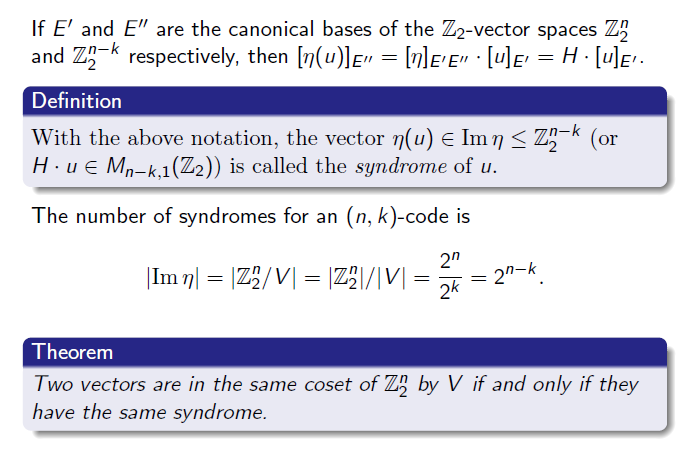
* Start with a code vector v belonging to V = Imɣ ≤ Zn2, and assumet hat an error **e** belonging to Zn2 occurs during transmission.
* Then the received vector is u = v + e belongs to Zn2. The receiver determines the most likely transmitted vector by finding the most likely error pattern (called the coset leader):

e = u - v = u + v belongs to u + V.

* The coset leader will usually be the coset containing the smallest number of 1's. If two or more error patterns are equally likely, the coset leader is chosen such that the 1's in the error pattern are bunched together as much as possible.

**Syndromes**

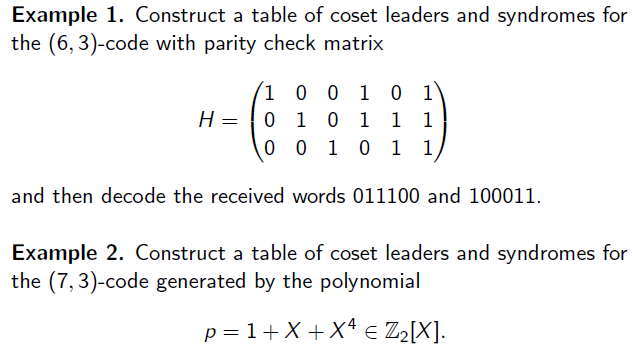


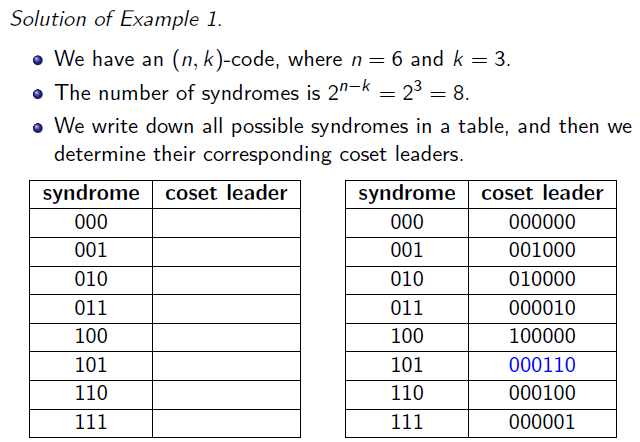


**A general method for decoding:**

1. Calculate the syndrome of the received word.
2. Find the coset leader of the coset corresponding to the syndrome.
3. Subtract the coset leader from the received word to obtain the most likely transmitted word.
4. Drop the check digits to obtain the most likely message.

**Decoding – Examples:**





* The coset leaders (the most likely errors) are chosen such that they contain the smallest number of 1's. If two or more error patterns are equally likely, the coset leader is chosen such that the 1's are bunched together as much as possible.
* We first consider the coset leader with all bits 0, then coset leaders having only one bit 1, then two consecutive bits 1, then two bits 1 not necessarily consecutive etc., until we find all correspondences with the syndromes.
* We use the general matrix equality:

[syndrome] = H·[vector]

