Demodulation and Error Correction

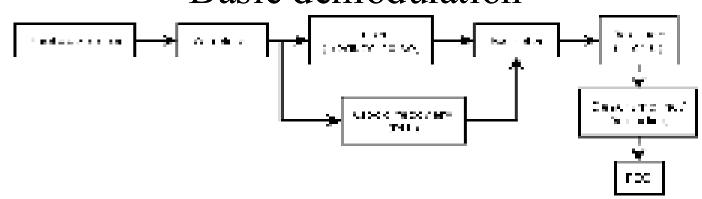
Presentation Outline

- Context of Forward Error Correction
  - From emitter to receiver
  - Basic demodulation
- Signal characteristics
  - Transmission format
  - DC balance
- Error correction
  - Description of a code in general
  - Reed-Solomon codes
- Conclusion

From emitter to receiver...

- Operations on original electrical signal
- Electrical signal modulates a light source
- Transmission over optical network
  - Loss, dispersion, etc.
- Receiver must restore original signal
  - Correct for transmission problems
    - Directly correcting a specific problem
    - Treating the problems as “error during transmission”
  - Restore original signal (unscrambling)

Basic demodulation



- The goal is to take optical power and convert it to an accurate electrical signal
- There is a dependence on signal characteristics in the clock recovery and the decision mechanism
- FEC is an optional module that operates on final electrical signal. It adds overhead

Modulation and formats

- Direct modulation or indirect modulation
- Modulations:
  - OOK
  - Sub carrier
- Influences:
  - transmission quality
  - types of signals
- Formats:
  - Non Return to Zero:
    - Less frequency bandwidth
    - Problem of long sequences
  - Return to Zero:
    - Sampling needs to be more precise

DC balance

- Some signals are more equal than others...
- Problem: receiver needs to separate 1 and 0
- Solution: calculate DC component (constant)
- Implementation:
  - Invert some bits to achieve balance
  - Add balanced code to determine which bits were inverted
- Improvement:
  - Word disparity compensates code disparity

What is error correction?

- Goal: Detect and correct errors in a list of transmitted symbols
  - Symbols can be 0/1 or not (RS uses group of bits)
- Requires encoding and decoding
- Overheads incurred:
  - Time to encode and decode
  - Extra transmitted symbols
- Aimed for gains:
  - More transmission problems allowed
  - Lowered bit error rate

Description of a code in general

- (n, k) code:
  - Takes k symbols as source (data word)
  - Outputs n symbols as code (code word)
- Error correction potential:
  - t represents the number of symbol errors the code can correct
  - For block codes $2t \leq n - k$
- Mathematical representation:
  - Determines implementation
  - Guarantees correctness

Matrix representation of codes

- We have:
  - The data word: $(d_0, d_1, d_2, ..., d_{k-1})$
  - The code word: $(c_0, c_1, c_2, ..., c_{n-1}) = \mathbf{d} \cdot \mathbf{G}$
- The matrix **G** is the representation of the code:

$$\begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,n-k-1} & 1 & 0 & 0 & \ldots & 0 \\ p_{1,0} & p_{1,1} & \cdots & p_{1,n-k-1} & 0 & 1 & 0 & \ldots & 0 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} & 0 & 0 & 0 & \ldots & 1 \end{bmatrix}$$

Matrix representation of codes

- We define **H** as: $\mathbf{G} \cdot \mathbf{H}^T = 0$
- Therefore, we have a characteristics of code words: $\mathbf{c} \cdot \mathbf{H}^T = 0$
- The syndrome, which will allow us to correct errors, is defined as: $\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T = \mathbf{e} \cdot \mathbf{H}^T$

Polynomial representation of codes

- Goal: use polynomial arithmetic to simplify code theory
- Represent words as polynomials where the coefficients are the matrix elements given before
- All calculations are modulo operations !!
- Use a generator polynomial (code word is a multiple of this polynomial)

Link between polynomials and matrix

- Encode basis vectors and form the matrix:

$$c(x) = b(x) + x^{n-k} d(x)$$

- For example for $g(x) = 1 + x + x^3$:

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

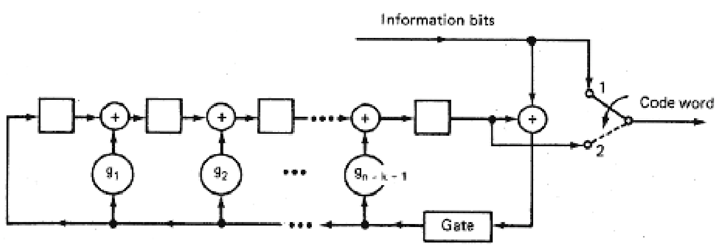
Reed-Solomon codes

- Code that works on blocks:
  - Non recursive code: independent of previous output
  - Uses algebraic properties (field and group theory)
- In detail:
  - Operates on $GF(2^m)$
  - $n = 2^m - 1$ symbols in code word
  - $k = n – 2t$ information symbols (data)
  - $2t = n – k$ check symbols
  - Corrects t symbol errors

Encoding with RS codes

- Choose a generator polynomial (all code words are multiples of this polynomial)
- Divide $x^{n-k}d$ by g and take the remainder
  - Usage of a division circuit
- Concatenate d and the remainder obtained above

RS Encoding: Example



RS Decoding

- Mathematical description complex and long
- Principle:
  - Calculate the “syndrome” or remainder modulo g(x) of the received word
  - Locate the errors (error locater polynomial)
  - Calculate the values of the errors
  - Correct the code
- Operations are easily implemented in hardware.
- Very good for correcting “burst errors”

Conclusion

- Importance of line coding/scrambling
  - Improve chances of correctly interpreting the signal
- FEC advantages:
  - Improves BER
  - Permits more lossy network, etc.
- FEC disadvantages:
  - More overhead
  - Higher costs
  - needs consistency between end nodes
- Need to keep in mind the global picture