Error control schemes in M17

Introduction

- Link degradation and data errors are effects inherent to any real-world RF link
- All digital modes use some form of „data protection”
- A simple repetition code is an example of error control coding

Error control coding

- M17 uses multilevel error control coding

Error control coding

- Input: data to be sent
- That can be the LSF or voice frame
- Catch: the lengths of LSF and voice frames don’t match! Why that’s a problem anyway?

Error control coding - LSF

- Let’s focus on the LSF
- LSF contains 240 bits to be sent
- 40 milliseconds of 9600 bps transmission is 384 bits
- Syncword takes 16 bits…
- That leaves us 368 bits

Error control coding - LSF

- How to encode 240 on 368 bits?

Error control coding - LSF

- Let's convolutionally encode them using ½ rate coder
- That should give us 488 bits (why not 480?)

Error control coding - LSF

- What to do next with those 488 bits?
- Let’s remove some to get 368…
- This process is called code puncturing

Error control coding - LSF

- OK, we ended up with 368 bits, can we send them over RF yet?
- Not really. Imagine that the RF channel can add so much noise that 10 consecutive bits are received wrong. Let’s assume our code can correct up to 5 consecutive errors.

Error control coding - LSF

- If only we, somehow, managed to spread those in time
- Let’s interleave (reorder) our data, so that all those bits are really mixed and consecutive bits get spread out
- This is called interleaving

Error control coding - LSF

- M17 uses a special case of interleaver called Quadratic Permutation Polynomial Interleaver
- The „monster” behind that name is actually a simple algebraic construct

Error control coding - LSF

- The method is robust, interleaver indices can be computed with only integer and modulo arithmetic

Error control coding - LSF

- Now our 368 bits are mixed up, but…

Error control coding - LSF

- There’s a chance that some consecutive bits (or, more precisely, dibits – symbols) are the same and effectively act as a DC component presented to the modulator
- Decoders at the receiving side don’t like that either*

Error control coding - LSF

- That brings the final step to do
- Let’s (pseudo)randomly invert some bits (but always in the same sequence!)
- How? XOR the data with a predefined pseudorandom stream - decorrelate
- Poof, DC component is gone!

Error control coding - LSF

- LSF data bits can now be sent over RF link

What about voice frames?

- The method is similar
- We first encode LICH (48 bits) using Golay code
- Then we convolutionally encode the rest of the frame (frame number and the paylod itself)
- Again – we have to reduce the number of bits to fit in the frame

What about voice frames?

- The last step is interleaving and decorrelating, just like for the LSF

73!