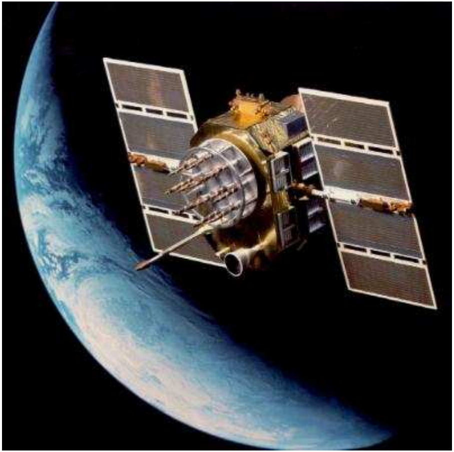
Modulation and Coding for Satellite Communications

H. D. Pfister
Swiss Federal Institute of Technology, Lausanne (EPFL)

SatNEx Summer School
Pisa, Italy
August 22nd, 2005

In the beginning...

- Arthur C. Clarke proposes geostationary satellites for communications (1945)
- Prototypes in space: TELSTAR, RELAY, SYNCOM (1964)
- NavStar satellite shown



Outline

1. Overview
   - Communications Theory
   - Modulation and Coding
2. State of the Art
   - Modulation
   - Coding
3. Status of the Standards
   - DVB-S2

Communications Theory

- The Good News
    - The same tools work for most communications systems
    - Ex: wireline modems, hard drives, cellular phones, etc...
- The Bad News
    - Satellite communications uses almost everything
    - Point-to-point, multiple access, broadcast, etc...

The Big Difference: Costs and Constraints

- Universal costs
  - Bandwidth and Transmit power (usually a spectral mask)
- Other costs and constraints
  - Input alphabet: Magnetic grains have only 2 stable states
  - Battery power: Cellular has 200 mW limit, but prefers less
- Satellite costs and constraints
  - Weight=$$$: Processing and power is expensive in space
  - Ground station: Processing is cheap and upgradeable
  - Doppler shift: Can be significant for LEO and MEO
  - Delay: Geostat. orbit requires 1/4 sec. round trip delay

Goals

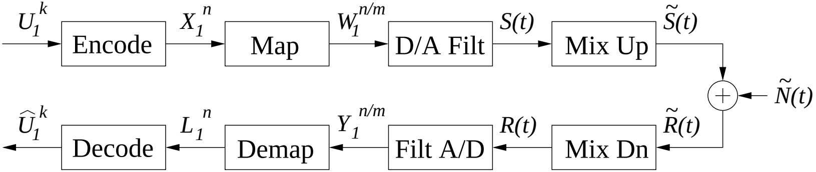
- First Part: What all researchers in satellite communications should know about modulation and coding
  - Pulse shaping and detection theory
  - A bit of information theory
  - Signal constellations
  - A bit of coding theory
- Later, we will cover
  - Advanced modulation techniques
  - Advanced coding systems (Turbo, LDPC, etc...)
  - LDPC codes in the DVB-S2 standard

A Simple as Possible, but...

The wireless telegraph is not difficult to understand. The ordinary telegraph is like a very long cat. You pull the tail in New York, and it meows in Los Angeles. The wireless is the same, only without the cat.

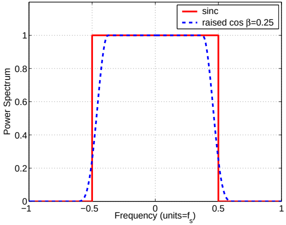
–Albert Einstein

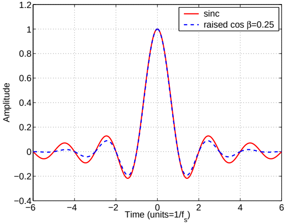
Communications Model



- Encoder: Maps data vector U_1^k to code vector X_1^n
- Mapper: Maps X_1^n (m-bits at a time) to symbols $W_1^{n/m}$
- Demapper: Maps $Y_1^{n/m}$ to bit log-likelihood ratios (LLRs) L_1^n
- Decoder: Estimates information sequence $\widehat{U}_1^k$

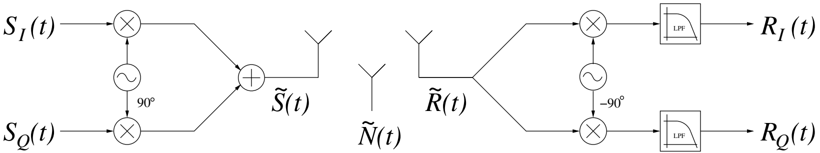
Pulse Shaping





- Continuous time band limited signals: $f_{max} = f_s/2$
  - Spanned by infinite *orthogonal* basis: $\mathrm{sinc}(f_s\, t - i)$ for $i \in \mathbb{Z}$
  - Ideal spectrum and no intersymbol interference (ISI)
  - Slow $O(1/t)$ decay of sinc causes problems though
- Excess bandwidth: $f_{max} = (1+\beta)f_s/2$ (f_s =signalling rate)
  - Raised cosine pulse often used: ISI free with decay $O(1/t^3)$

Mixers: Up and Down Conversion



- Complex Data Symbols (**I**n-phase and **Q**uadrature)
  - Bandwidth *not increased*, but degrees of freedom *doubled*
  - Baseband signal has I/Q components $S(t) = S_I(t) + j\,S_Q(t)$

$$S_I(t) = \sum_{i=1}^{k} \mathrm{Re}\,[X_i]\; p(f_s\,t - i) \qquad S_Q(t) = \sum_{i=1}^{k} \mathrm{Im}\,[X_i]\; p(f_s\,t - i)$$

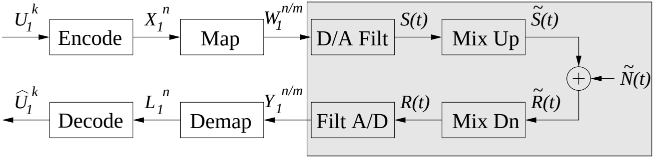
- Ideally: RX baseband = TX baseband + Baseband noise

$$S(t) = R(t) + N(t)$$

Detector

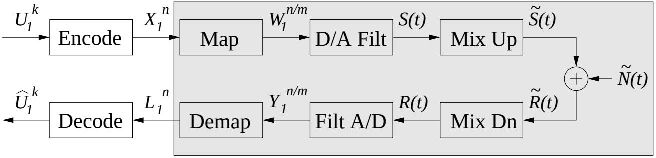
- Baseband model: $R(t) = \sum_{i=1}^{k} X_i \, p(f_s \, t - i) + N(t)$
  - where $p(t)$ is an ISI-free pulse s.t. $p(i) = \delta_i$ for $i \in \mathbb{Z}$
- Matched filter: $h(t) = p^*(-t)$
  - $R(t)$ is filtered with $h(t)$ and sampled at the symbol rate
  - Each output maximizes the symbol SNR in ind. noise
- Square root filtering
  - ISI-free condition on pulse only required at sampling time
  - Real-symmetric pulse: TX pulse and RX filter are identical
  - Sqrt in freq. domain splits filtering evenly between the two

Symbol Effective Channel



- Effective Symbol Channel: $W_1^{n/m} \rightarrow Y_1^{n/m}$
  - Defined by $Pr(Y = y|W = w)$
  - Information lossless

Bit Effective Channel



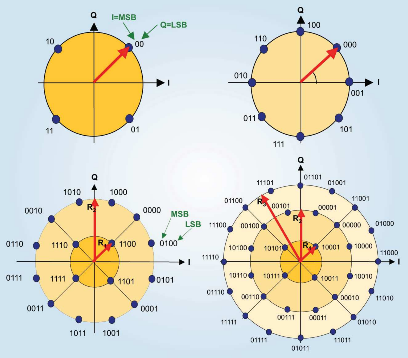
- Effective Bit Channel: $X_1^n \rightarrow L_1^n$
  - Really m channels defined by $Pr(L_i = l | X_i = x)$
  - Suboptimal: information loss in conversion to bit LLRs

Information Theoretic Model

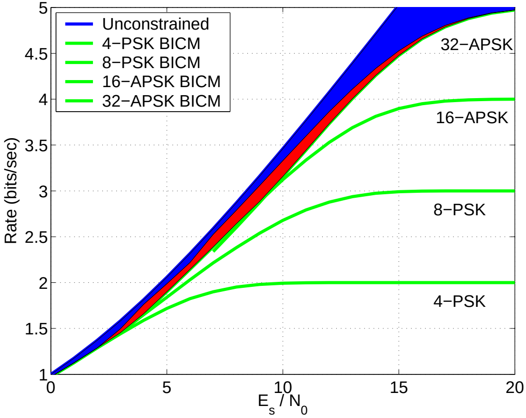
- Information Theory
  - An *information rate* R (bits/channel use) is *achievable* if $R <$ the mutual information between the inputs and outputs
  - *Capacity* C is the max. achievable rate over input dist.
- Effective Symbol Channel: $W_1^{n/m} \rightarrow Y_1^{n/m}$
  - Information rate given by $I_0 = I(W;Y)$
- Effective Bit Channel: $X_1^n \rightarrow L_1^n$
  - Information rate given by $I_{\text{BICM}} = \sum_{i=1}^{m} I(X_i;L_i)$
  - At moderate SNR, small loss for gray-coded modulations

Signal Constellations

- Set of points for transmission
  - PSK = points on a circle
  - QAM = subset of 2D grid
  - APSK = union of scaled PSK
- Labelling
  - Map from bits to symbols
  - Natural binary: 00 01 10 11
  - Gray code: 00 01 11 10
- Constellations from “DVB-S2”
  - A. Morello and V. Mignone EBU Tech. Rev. Oct. 2004



Constrained Capacity



- Information rate of some signal constellations in AWGN

Information Theory and Coding

- Mutual Information: An achievable upper bound on the rate
  - Proof by random construction of error correcting codes
  - Lack of code structure makes decoding infeasible
- In practice: Codes with feasible decoding algorithms used
  - Reed-Solomon (RS) and BCH Codes
  - Convolutional Codes (CC)
  - *Sparse Graph Codes*: Turbo, LDPC, etc...
    - Sparse graph defines codeword constraints
    - Conjectured to achieve capacity on wide variety of channels

Evaluating the Physical Layer

- What’s Important?
- *Average throughput* of a heavily loaded system under max delay and max outage constraints
  - As system load reduces, user experience should improve
- *Efficiency* or Bits per second per Hz
  - Measured relative to the theoretical limit?
  - Losses: Excess BW, code rate, ARQ, packing, etc..

Theoretical Limits vs. Creativity

- Try to separate theoretical limits from past practice
- Ex: Rain fades → large margin for satellite broadcasts
  - 6 dB of margin can reduce rate to 1/4 of original
  - Priority channels: CNN, Weather, SoapNet, etc...
    - Give large margin for rain fades
    - Other channels received only in clear weather
  - Time diversity
    - Correlation time of fading in hours
    - Send popular shows periodically and use soft combining

Non-Linear Amplification

- The Problem
  - Power is a valuable commodity in space
  - Amplifiers must be operated near saturation for efficiency
  - Saturation distorts the signal and expands the bandwidth
- Solutions
  - Linearize via pre-distortion: Reduces back-off roughly 3 dB
  - (almost) Constant envelope modulation
    - Continuous Phase Modulation: works, but too strict?
    - QPSK: 180° phase change causes large variations
    - Offset QPSK: Delays Q by $\frac{1}{2}$ symbol, max phase change 90°
    - Feher/Simon: Merge OQPSK and filter and force continuity
  - QAM → APSK: Not constant, but gains roughly 2 dB

Amplitude Phase Shift Keying

- Multiple rings of PSK modulation
    - Originally proposed by Thomas et al. in 1974
- Recent impact due to optimization (Gaudenzi et al. 2004)
    - Number of points in each ring
    - Relative radii/phase between rings
    - Optimized both for capacity and minimum distance
- Other benefits
    - Easy to pre-compensate phase and amplitude for NLA

Optimal Modulation

- What do we want from pulse shaping and modulation?
  - Narrow bandwidth after the non-linear amplifier (NLA)
- Filtering before the NLA
  - Narrows bandwidth, but NLA can still spread things out
- Combine filter and modulation in a trellis? (Simon 1999)
  - Based on patented Feher QPSK (FQPSK)
  - Set of waveforms chosen based on past inputs
  - Signal and derivative is continuous → narrow bandwidth
  - Optimize for capacity under memory and spectral constraint
  - Extend to 8 PSK (O8PSK?), APSK, etc...

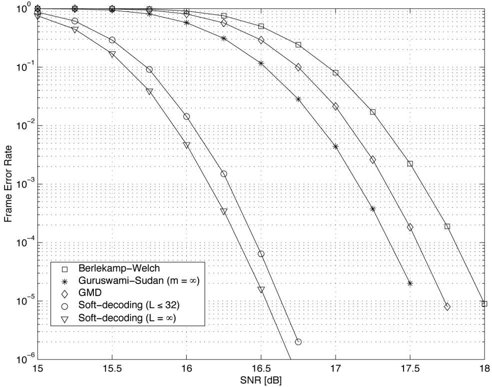
Coding Terminology

- Basic parameters
  - Block length n, information bits k, rate $R = \frac{k}{n}$
  - Minimum Hamming distance between codewords d
- Performance curves versus SNR
  - Bit error rate = Avg. fraction of bits in error after decoding
  - Block/Frame/Packet error rate = Avg. fraction of (*) in error
- Gap to Capacity
  - Excess SNR for reasonable error rate (e.g.,$10^{-3} - 10^{-5}$)

Reed-Solomon and BCH Codes

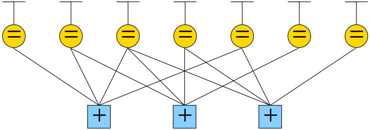
- Can be designed with large minimum distance d
  - RS codes: maximum distance separable $d = n - k + 1$
  - BCH: Good distance for small to moderate block lengths
- Guaranteed decoding radius
  - Classical decoders always correct up to $\left\lfloor \frac{d-1}{2} \right\rfloor$ errors
  - Algebraic List (and soft) decoding extends this radius
- Often used as an outer code (e.g., DVB-S)
  - Standard RS decoders use *hard decision* decoding
  - Typically, use some inner code with *soft decision* decoding

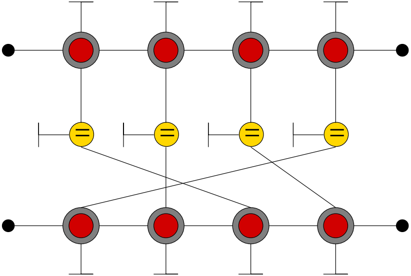
List Decoding of RS Codes



- (n=255,k=144,d=112) RS code for 256-QAM in AWGN
- From “Algebraic Soft Decision Decoding of Reed-Solomon Codes”, Koetter and Vardy, Trans. IT 2001

Sparse Graph Codes



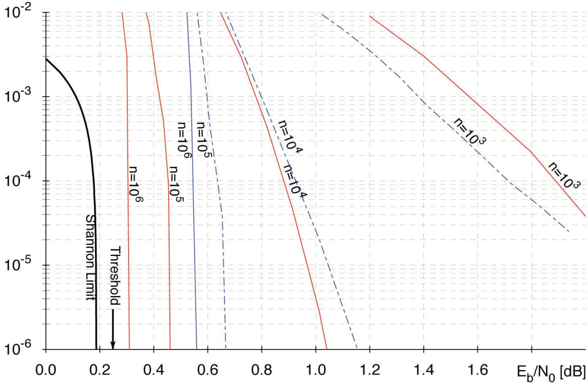


- Codeword constraints defined via sparse factor graph
  - Vertices = constraints
  - Edges = variables, Half-edges = Observations
- Three typical constraints
  - Equality (=): Edges are bits that must have the same value
  - Parity (+): Edges are bits that must sum to zero (mod 2)
  - Trellis: Bit edges must be compatible with state edges

Some Recent History

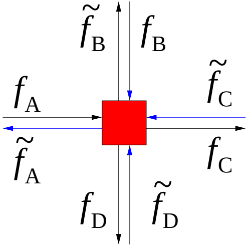
- Turbo Codes
    - Introduced in 1993 by Berrou, Glavieux, and Thitimajshima
    - Revolutionized coding theory with performance
    - McEliece et al.: turbo decoding = belief propagation (1998)
    - Double-binary turbo codes for DVB-RCS standard (2000)
- Low Density Parity Check (LDPC) Codes
    - Introduced in 1960 by Gallager and then forgotten
    - Re-discovered by MacKay in 1995
    - Irregular LDPC achieves capacity on BEC (1997)
    - Density evolution for AWGN: 0.0045 dB from cap. (2001)
- Sparse Graph Codes
    - With the factor graph approach, the possibilities are endless
    - RA, IRA, CA^m, ARA, CT, multi-edge LDPC, protograph, ...

Turbo vs. LDPC Performance



- BER: Standard Turbo (blue) vs Irregular LDPC (red)
- From “The Capacity of LDPC Codes Under Message Passing Decoding”, Richardson & Urbanke, Trans. IT 2001

Message Passing Decoding



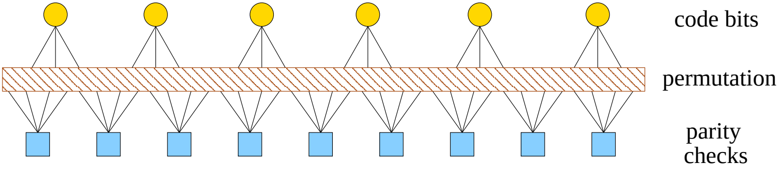
- Example constraint node
  - $\mathcal{C}$ = assignments $\{a, b, c, d\}$ which satisfy the constraint
  - Input msgs are belief functions
    $f_A(a) \sim Pr(A = a)$
  - Output msgs are new belief functions
    $\widetilde{f}_A(a) \sim Pr(A = a)$
- Sum-Product: APP if no cycles

$$\widetilde{f}_A(a) = \sum_{\{a,b,c,d\}\in\mathcal{C}} f_B(b) f_C(c) f_D(d)$$

- Max-Product: ML if no cycles

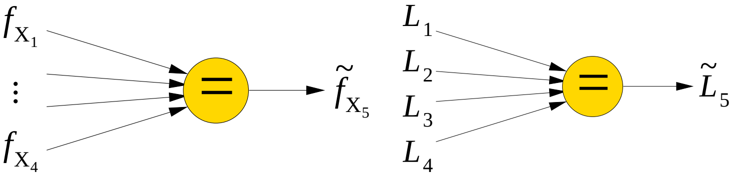
$$\widetilde{f}_A(a) = \max_{\{a,b,c,d\}\in\mathcal{C}} f_B(b) f_C(c) f_D(d)$$

LDPC Codes Defined



- Linear codes defined by $Hx = 0$ for all c.w. $x \in \mathcal{C}$
  - H is the sparse parity-check matrix ($r \times n$) of the code
  - Ensembles defined by bit/check degrees and rand. perm.
- Bipartite graph
  - Bit (check) nodes associated with columns (rows) of H
  - Each check is attached to all bits that must satisfy the check

(Sum/Max)-Product Decoding: Bit Nodes



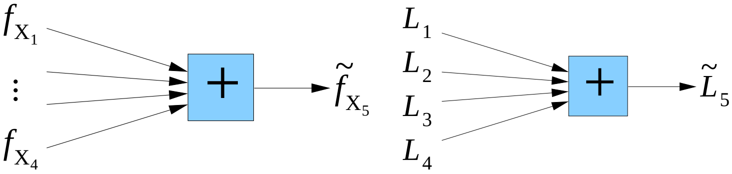
- Binary X: Use log-likelihood ratio (LLR) messages

$$L_i = \log \frac{f_{X_i}(0)}{f_{X_i}(1)} = \log \frac{Pr(X_i = 0)}{Pr(X_i = 1)}$$

- Equality constraint: only $00\ldots00$ and $11\ldots11$ valid

$$\tilde{L}_i = \log \frac{\tilde{f}_{X_i}(0)}{\tilde{f}_{X_i}(1)} = \frac{\prod_{j \neq i} f_{X_j}(0)}{\prod_{j \neq i} f_{X_j}(1)} = \sum_{j \neq i} L_j$$

Max-Product Decoding: Check Nodes



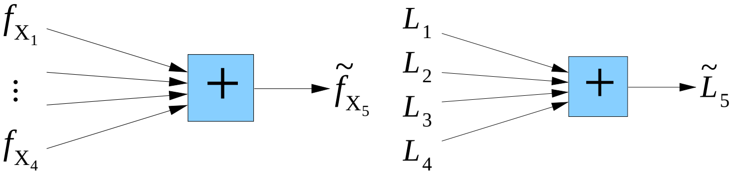
- Parity constraint: only even weight patterns valid

$$\tilde{f}_{X_i}(0) = \max_{\sum_{k\neq i} z_k = 0} \prod_{j\neq i} f_{X_j}(z_j) \qquad \tilde{f}_{X_i}(1) = \max_{\sum_{k\neq i} z_k = 1} \prod_{j\neq i} f_{X_j}(z_j)$$

- Maximizing patterns differ only in least reliable bit, so

$$\tilde{L}_i = \left(\prod_{j\neq i} \operatorname{sgn} L_j \right) \min_{j\neq i} |L_j|$$

Sum-Product Decoding: Check Nodes



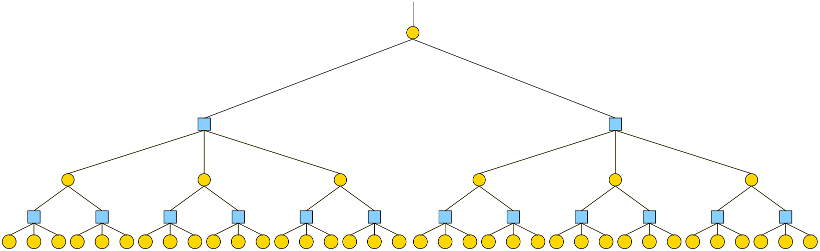
- Parity constraint: only even weight patterns valid

$$\tilde{f}_{X_i}\left(\frac{1}{2} \pm \frac{1}{2}\right) = \frac{1}{2}\prod_{j \neq i}\left(f_{X_j}(0) + f_{X_j}(1)\right) \mp \frac{1}{2}\prod_{j \neq i}\left(f_{X_j}(0) - f_{X_j}(1)\right)$$

- Define $h(x) = -\log\tanh|x/2|$, and some algebra gives

$$\tilde{L}_i = \left(\prod_{j \neq i}\operatorname{sgn} L_j\right) h\left(\sum_{j \neq i} h\left(L_i\right)\right)$$

Why Leave One Out?



- “Leave one out” rule: unrolls the graph as above
- Message passing is exact if unrolled graph is a tree
- If graph had girth g, then depth of first cycle is $g/2$

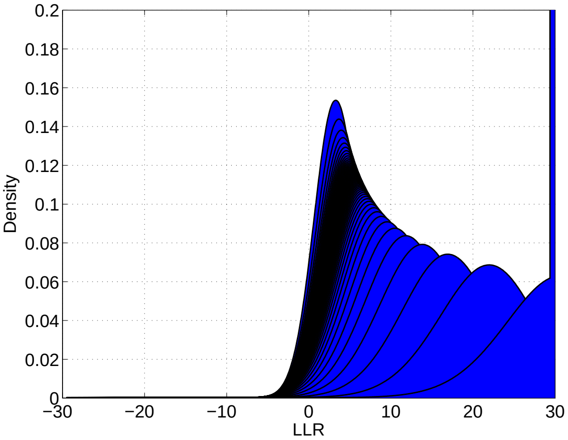
Basic Types of LDPC Codes

- Regular (j, k): All bits degree j and all checks degree k
- Irregular (λ, ρ)
    - λ_i is the fraction of edges attached to degree i bits
        - ρ_i is the fraction of edges attached to degree i checks
        - Optimizing λ, ρ gives very good results for long codes
- Protograph: Generate code from a single small graph
    - Less variation in ensemble → better at short block lengths
        - Large optimization space → still good at large block lengths
        - Similar to multi-edge LDPC but much simpler to visualize

Evolution of the Message Density (1)

- Density Evolution
  - Assume unrolled graph is a tree for $\ln n$ depth
  - Density functions of quantized LLR messages
    - Can be computed recursively based on previous iteration
    - Update rule symmetry: $f(x_1, x_2, x_3) = f(f(x_1, x_2), x_3)$
    - Brute force density of $f(X_1, X_2)$ from density of X_1, X_2
- Concentration Theorem: Threshold effect as $n \to \infty$
  - For *monotonic* channels there is a *maximum noise level*
  - Above this level, decoding is almost surely successful
  - Below this level, decoding almost surely fails

Evolution of the Message Density (2)



- DE for a Regular (3,6) LDPC code in AWGN

Sparse Graph Code Constructions

- Connect component codes via random permutations
  - Turbo codes: connect input bits of two convolutional codes
  - LDPC codes: connect bit node and check node graphs
- Avoid short cycles and/or other bad local configurations
  - Simple heuristics: S-random interleaver, no 4-cycles
  - Progressive Edge Growth: Greedy girth maximization
    - Great for regular codes, subtle problems with irregular
- Complexity constraints
  - Design interleaver to avoid memory conflicts in parallelism
  - Block-π LDPC: parity check matrix is block circulant

Block-Permutation LDPC Codes

$$H = \begin{bmatrix} I_{P(1,1)} & 0 & 0 & I_{P(1,4)} & 0 & I_{P(1,6)} \\ 0 & I_{P(2,2)} & 0 & I_{P(2,4)} & I_{P(2,5)} & 0 \\ 0 & 0 & I_{P(3,3)} & 0 & I_{P(3,5)} & I_{P(3,6)} \end{bmatrix}$$

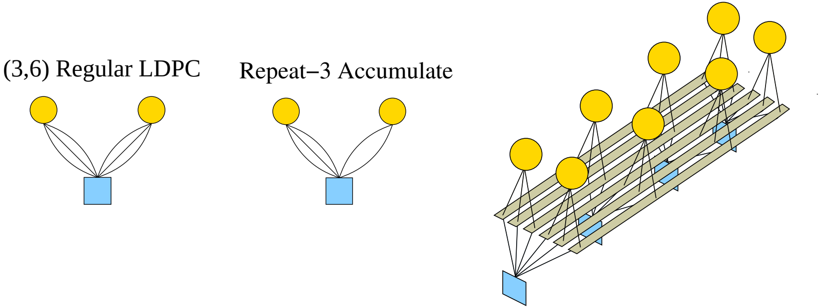
- PC matrix has block form with cyclic permutation blocks
  - I_k denotes a $B \times B$ identity matrix right circ. shifted by k
  - Low descriptive complexity: Block positions and shifts
  - Allows B bits/checks to processed in parallel
  - Quasi-cyclic: Any codeword “shifted” by B is a c.w.
- Drawbacks
  - The girth of the graph becomes small if B/n too large
  - Symmetry: bad configurations occur with large multiplicity
  - One solution: Use an outer code to reduce error floor

A Simple Decoding Example

$$H = \left[\begin{array}{cc} I_0 & I_0 + I_1 \end{array} \right] = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{array} \right]$$

- Simplest non-trivial single error correcting code
  - Parameters $n = 6$, $k = 3$, and $d = 3$
- Easy way to see it's quasi-cyclic ($B = 3$)
  - Shift block right by 1 and down 1 → PC matrix unchanged
  - Therefore, cyclic shifts of c.w. bits in blocks → new c.w.
- Max-product decoding corrects all single errors
  - Also all wt. 2 erasures and all but 1 wt. 3 erasures
  - Try max-product decoding of $\left[\begin{array}{cccccc} 2 & 2 & -1 & 2 & -1 & 2 \end{array} \right]$

Protograph Codes

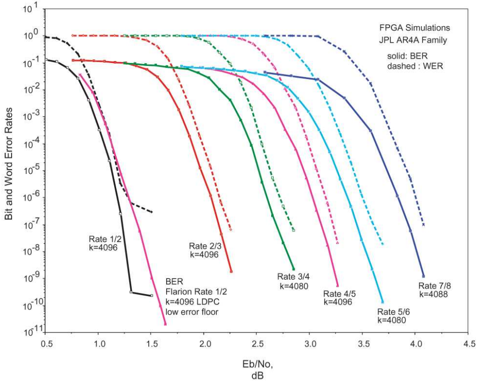


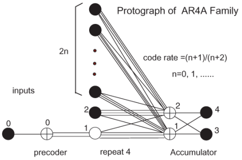
(3,6) Regular LDPC

Repeat–3 Accumulate

- Small graph with edges are labelled $1, \ldots, E$
  - Copy and stack that graph m times
  - Introduce E random permutations of length m
  - Attach edges labelled j to the permutation labelled j
  - For large m, small cycles and double edges are no problem

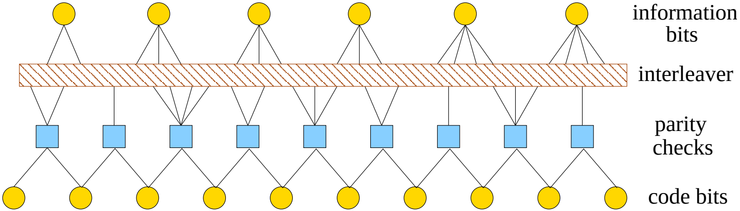
Particularly Promising Protographs





- From “Constructing LDPC Codes From Simple Loop-Free Encoding Modules”, Divsalar et al., ICC 2005

Variations on the LDPC Theme



- Irregular Repeat Accumulate (IRA) Codes
    - Also known as LDPC codes with zig-zag degree 2 bits
    - This is the family of LDPC codes used by DVB-S2
- Best current LDPC codes are variations of IRA and ARA
    - Flarion's multi-edge LDPC similar to punctured IRA
    - Protographs of Divsalar et al. use ARA, ARAA, etc...
    - Shown to achieve BEC capacity with *bounded complexity*

Performance of Iterative Decoding

- Waterfall Region
    - Determined, in general, by large sized failures
    - DE threshold determines performance as $n \to \infty$
    - Finite-length scaling gives convergence rate (Amraoui)
        - Only rigorous and computable for the BEC right now
- Error Floor Region: Pseudo-Codewords
    - Linear programing view of decoding (Feldman et al.)
        - Rigorous definition of pseudo-codewords
        - Maximum likelihood certificate if decoding is successful
    - Local nature of iterative decoding (Koetter-Vardy)
        - Essentially same definition of pseudo-codewords
    - Great for understanding error floor, but not computable yet

DVB-S2 LDPC Coding Standard

- Large block length IRA codes over wide range of rates
    - Block lengths of 16200 and 64800
    - Rates from 1/4 to 9/10
    - Adaptive modulation: spectral efficiency from 0.49 to 4.45
    - BCH inner code with error correction radius from 8 to 12
- Codes are structured for simplified encoding and decoding
    - Repeat-parity portion is quasi-cyclic
        - Information bits used in circular groups of 360
        - Circ. shift of parity bits by $q = \frac{n-k}{360} \Leftrightarrow$ Circ. shift of info by 1
        - Can be rearranged into blocks of permutation matrices
    - Parity bits are accumulated after repeat-parity operation
        - This prevents the overall code from being quasi-cyclic

Generator Matrix

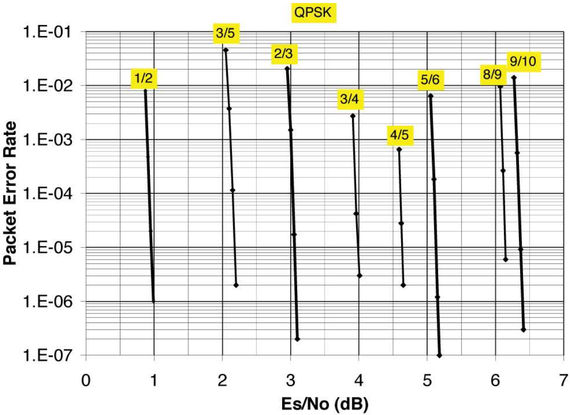
- Built by shifting a small set of prototype row vectors
- Let $v_i(j)$ be the jth element of ith prototype row vector
  - $v_i(j) = 1$ iff j is in the ith row of parity bit address table
  - $i = 0, \ldots, (k/360) - 1$ and $j = 0, \ldots, n-k-1$
- The kth row of G is $v_{\lfloor k/360 \rfloor}$ right circ. shifted by $q\,[k]_{360}$
  - $G_{k,l} = v_{\lfloor k/360 \rfloor}\left(\left[l - q\,[k]_{360}\right]_{n-k}\right)$ where $[a]_b = a \bmod b$
- Final parity formed by accumulating output of repeat-parity
  - Repeat-parity vector is $x = uG$ (u is information vector)
  - Final parity vector is defined by $p_i = \sum_{j=0}^{i} x_i = p_{i-1} + x_i$
  - Also written as $uG = Ax$ for $A_{ij} = 1$ iff $i = j$ or $i = j + 1$

Code Parameters (n=64800)

Code Rate	q	B	R_{avg}	L_3	i	L_i
1/4	135	45	2	2/3	12	1/3
1/3	120	120	3	2/3	12	1/3
2/5	108	36	4	2/3	11	1/3
1/2	90	90	~5	3/5	8	2/5
3/5	72	72	~9	2/3	12	1/3
2/3	60	60	~7	9/10	13	1/10
3/4	45	45	12	8/9	12	1/9
4/5	36	36	15.8	7/8	11	1/8
5/6	30	30	19.6	9/10	13	1/10
8/9	20	20	24.6	7/8	4	1/8
9/10	18	18	27.5	8/9	4	1/9

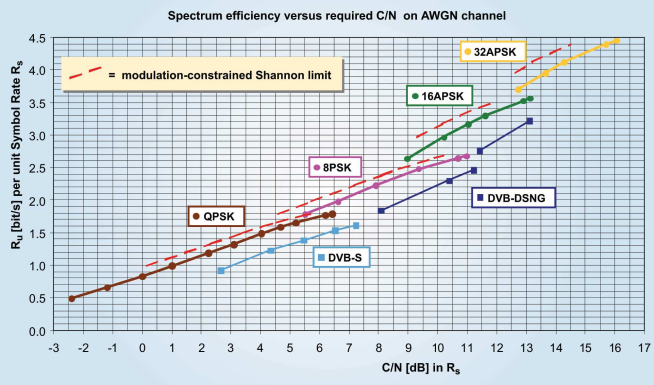
- Fraction L_i bits rep. i times, R_{avg} is the avg. check degree
- PC matrix can be put in block-π form with block size B

Performance



- Block length 64800 with BCH outer in AWGN
- From ETSI Standard TR 102 376 V1.1.1

Spectral Efficiency



- From “DVB-S2”, A. Morello and V. Mignone, EBU Tech. Rev. Oct. 2004

Conclusions

- Coding and modulation for SatComm is now quite good
    - Rapidly approaching the theoretical limits
    - To do this requires a number of advanced techniques
        - Pulse shaping and compensation for non-linear amp
        - Adaptive modulation: Close to capacity for wide SNR range
        - Turbo and LDPC type codes
- Of course, there is always room for good new ideas
    - Make use of previously unknown/ignored phenomenon
        - Ex: multi-user diversity, multiple antennas
    - Reduce HW complexity, simplify descriptions, etc...

Thanks for your attention.