Methods and tools to optimize the trade-off "performance versus complexity" of error control codes architectures.

Emmanuel Boutillon
CNRS, UMR 6285, Lab-STICC
Centre de Recherche - BP 92116
F-56321 Lorient Cedex - FRANCE

Telecom ParisTech
4 octobre 2012

OUTLINE

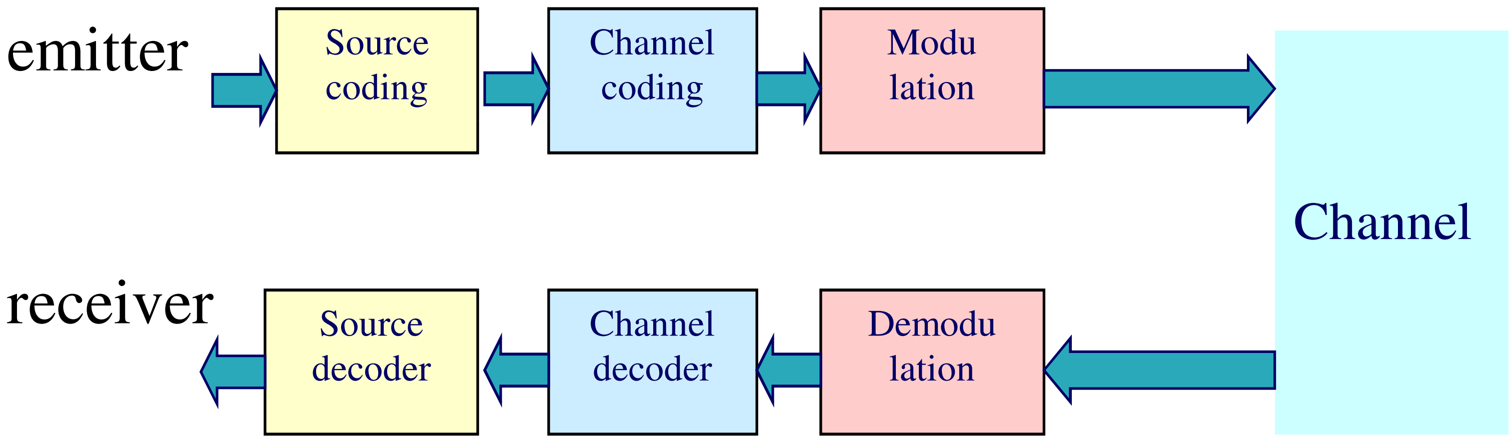
- Motivation
- Reduced Monte-Carlo Simulation
- Entropy Inspired Distance
- Hardware Emulation
- Conclusion

OUTLINE

- Motivation
- Reduced Monte-Carlo Simulation
- Entropy Inspired Distance
- Hardware Emulation
- Conclusion

Motivation

Design of a communication system...

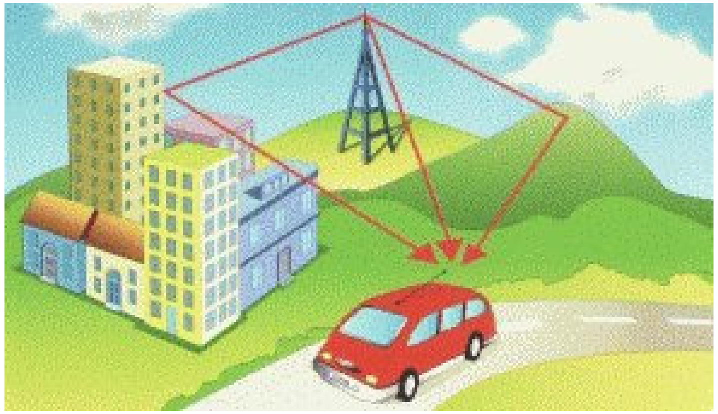


...find the best complexity-performance trade-off

Motivation

Performance:

- BER
- Jammer rejection
- time of synchronization…
- …



Complexity:

- area, power dissipation
- time to market
- …

algorithm
ADC resolution,
sampling frequency,
fixed precision

A very complex problem...

Monte-carlo simulation

- **Formal expression of the BER: refer to Proakis**

- **In practice, estimation of the BER using Monte-Carlo simulation**
  - Software model of emitter, channel, receiver
  - Emulation of the transmission of N bits
  - Estimation of the BER as Nb_errors/N

VERY EFFICIENT… BUT

TIME AND CPU CONSUMING:

Bit Error Rate of 10^{-6} (+-3%) requires 10^{9} bits.

Two separate problems

- **1) The code design problem.**
  - Theoretical bound
  - Weight estimation using impulse method and its derivative
  - EXIT CHART Tools.

Let's start from a C reference model in floating point (ideal decoder).

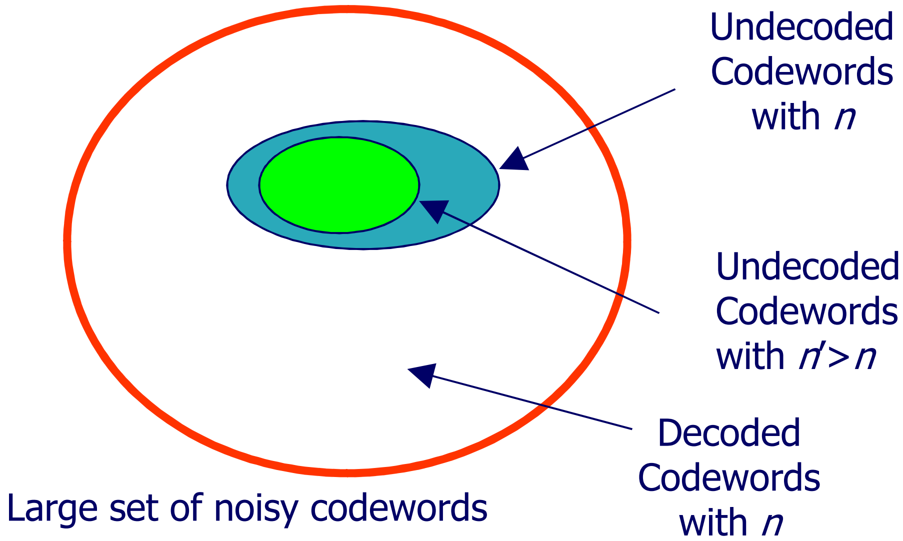
- **2) Evaluate performance of "hardware decoder" vs "ideal decoder"**

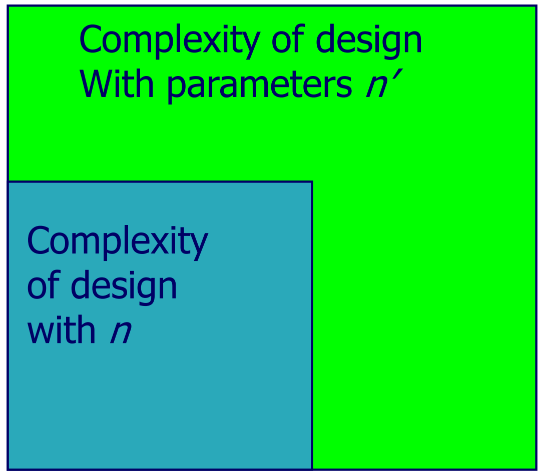
OUTLINE

- Motivation
- Reduced Monte-Carlo Simulation
- Entropy Inspired Distance
- Hardware Emulation
- Conclusion

REDUCED MONTE CARLO SIMULATION

- Let us assume that we want to find the best trade-off of an algorithm parameter *n*. Assuming that:
  - 1) Decoder complexity is an increasing function of *n*.
  - 2) Performance is a strictly increasing function, i.e. if a noisy codeword is decoded with parameter *n*, then the codeword is decoded with parameter *n′* > *n*.





In that case, RMCS can be efficient

REDUCED MONTE CARLO SIMULATION

- Step 1: Run Monte-Carlo simulation with the “worst-case” configuration. Store the undecoded codeword in the set *S* ;

- Step 2: Re-run decoder with better parameters only in the subset *S* to evaluate performances ;

- Step 3: Perform a classical MCS in order to verify the validity of hypotheses 1 for some configurations.

EXAMPLE ON NON-BINARY LDPC GF(64) WITH EMS (parameters n).

10^7 frames
(FER 5.10^{-3})

	it =8	*it* =10	*it* =12	*it* =14	*it* =16
n =8	49693	13535	5404	2993	2105
n =10	12340	2569	815	414	291
n =12	4763	893	270	109	62
n =14	2419	467	163	72	42
n =16	1349	223	~~56~~ **71**	22	~~13~~ **22**
n =18	835	151	25	10	4
n =20	592	116	17	7	6
n =22	455	103	14	4	3
n =24	373	86	~~14~~ **21**	5	~~4~~ **8**
n =26	321	91	16	4	3
n =28	262	54	9	5	3
n =30	235	52	8	4	3
n =32	228	52	~~7~~ **19**	4	~~2~~ **8**

Save 3 months of simulation...

OUTLINE

- Motivation
- Reduced Monte-Carlo Simulation
- Entropy Inspired Distance
- Hardware Emulation
- Conclusion

ENTROPY INSPIRED DISTANCE

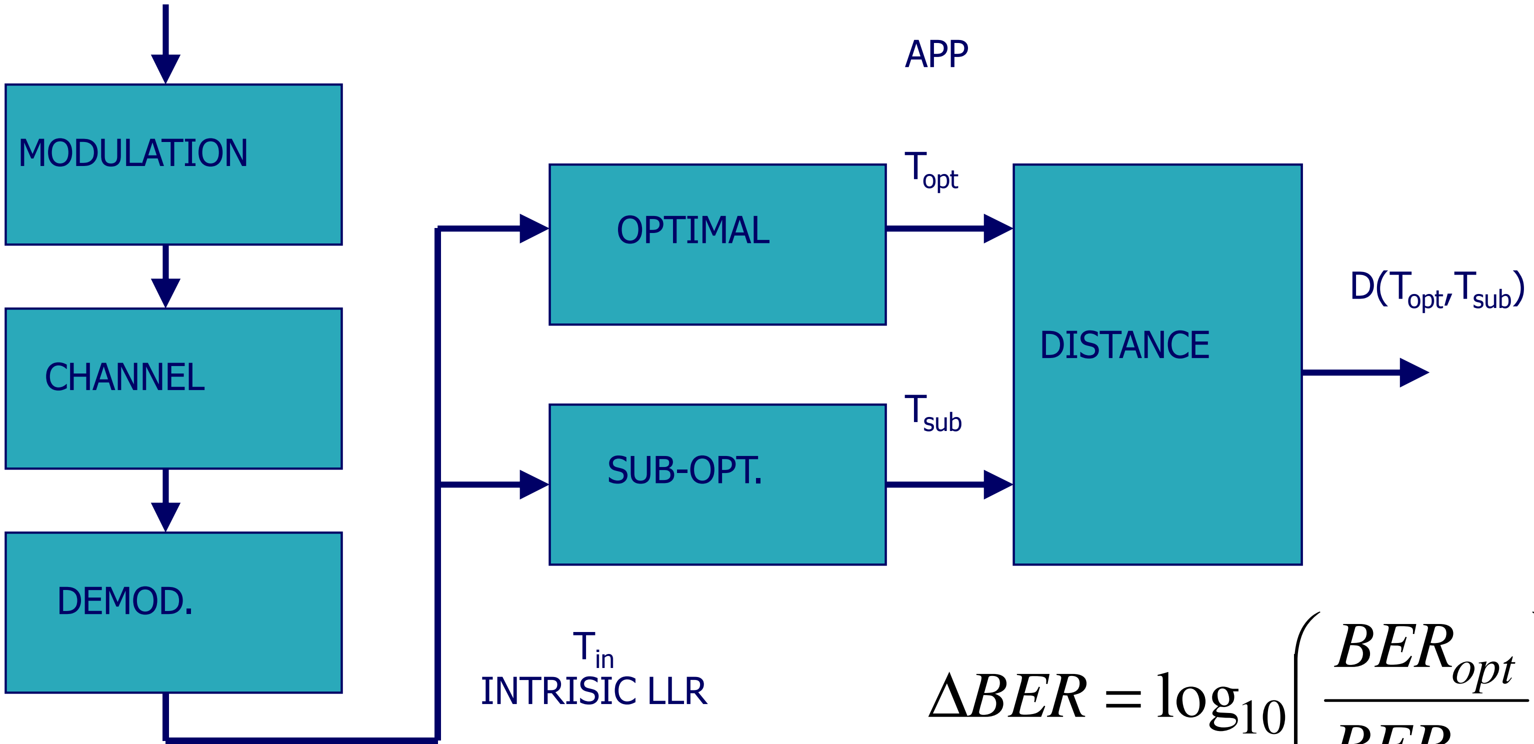
- OBSERVATION : FROM APP TO BER, A HUGE QUANTITY OF INFORMATION IS SUPPRESSED.

QUESTION 1 : CAN WE EXPLOIT THIS INFORMATION ?

QUESTION 2 : HOW ?

DISTANCE BETWEEN OPTIMAL/SUB-OPT.

SET OF *M* CODEWORDS OF SIZE P ($N = M \times P$ *Q*-ary symbols)



$$\Delta BER = \log_{10}\left(\frac{BER_{opt}}{BER_{sub}}\right)$$

FIND D(Topt,Tsub) RELATED TO ΔBER

QUESTION : WHAT DISTANCE TO CHOOSE...

- LET $(p_k^0, p_k^1, \ldots, p_k^{Q-1})$ AND $(\tilde{p}_k^0, \tilde{p}_k^1, \ldots, \tilde{p}_k^{Q-1})$ BE THE APP OF THE k^{th} DECODED SYMBOL FOR OPTIMAL AND SUB-OPTIMAL DECODERS.

$$\sum_{i=0}^{Q-1} p_k^i = 1 \qquad \sum_{i=0}^{Q-1} \tilde{p}_k^i = 1$$

- FIRST ATTEMPT : L1 MORM

$$L_1 = \sum_{k=0}^{N-1} \left(\sum_{i=0}^{Q-1} \left| p_k^i - \tilde{p}_k^i \right| \right)$$

EXPERIMENTAL RESULT: NO CLEAR CORRELATION BETWEEN L1(Tsub, Topt) AND ΔBER.

PROPOSED EXPLANATION: $p = 10^{-6}$ AND $p = 10^{-12}$ ARE REALLY DIFFERENT FROM A DECODING PERSPECTIVE

QUESTION : WHAT DISTANCE TO CHOOSE…

- LET $(p_k^0, p_k^1, \ldots, p_k^{Q-1})$ AND $(\tilde{p}_k^0, \tilde{p}_k^1, \ldots, \tilde{p}_k^{Q-1})$ BE THE APP OF THE *k*th DECODED SYMBOL FOR OPTIMAL AND SUB-OPTIMAL DECODERS.

$$\sum_{i=0}^{Q-1} p_k^i = 1 \qquad \sum_{i=0}^{Q-1} \tilde{p}_k^i = 1$$

- SECOND ATTEMPT : L1-log MORM

$$L_1 = \sum_{k=0}^{N-1} \left(\sum_{i=0}^{Q-1} -\log \left| p_k^i - \tilde{p}_k^i \right| \right)$$

EXPERIMENTAL RESULT: NO CLEAR CORRELATION BETWEEN L1-log(Tsub, Topt) AND ΔBER.

PROPOSED EXPLANATION: *p* =0.7 AND *p*=0.9 ARE REALLY DIFFERENT FROM A DECODING PERSPECTIVE AND -log(0.2) << -log(10^{-5})

QUESTION : WHAT DISTANCE TO CHOOSE...

- LET $(p_k^0, p_k^1, ..., p_k^{Q-1})$ AND $(\tilde{p}_k^0, \tilde{p}_k^1, ..., \tilde{p}_k^{Q-1})$ BE THE APP OF THE k^{th} DECODED SYMBOL FOR OPTIMAL AND SUB-OPTIMAL DECODERS.

$$\sum_{i=0}^{Q-1} p_k^i = 1 \qquad \sum_{i=0}^{Q-1} \tilde{p}_k^i = 1$$

- THIRD ATTEMPT : KULLBACK-LEIBLER DIVERGENCE

$$KL = \sum_{k=0}^{N-1} \left(\sum_{i=0}^{Q-1} - p_k^i \log(p_k^i / \tilde{p}_k^i) \right)$$

EXPERIMENTAL RESULT: NO CLEAR CORRELATION BETWEEN KL(Tsub, Topt) AND ΔBER.

PROPOSED EXPLANATION: *NO EXPLANATION...*

ENTROPY INSPIRED DISTANCE (EID)

- ENTROPY OF A DISTRIBUTION : $H(X) = \sum_{i=0}^{Q-1} - p_k^i \log(p_k^i)$

- PROPERTIES : H(X) ≥ 0, H(X)=log(Q) for $p^i=1/Q$, $i=0..Q-1$.

$$EID = \sum_{k=0}^{N-1} \left(\sum_{i=0}^{Q-1} - \left| p_k^i - \tilde{p}_k^i \right| \log\left(\left| p_k^i - \tilde{p}_k^i \right| \right) \right)$$

EXPERIMENTAL RESULT: USEFULL TOOLS TO PREDICT THE ΔBER !!!

PROPERTIES OF EID

- IF $Arg\max(p^i) = Arg\max(\tilde{p}^i)$

THEN EID IS A DISTANCE, I.E.

EID(X,X')=0 => X=X' (note that EID([0,0,0,1], [1,0,0,0]) = 0)

EID(X,X') = EID(X',X)

EID(X,Y) + EID(Y,Z) ≥ EID(X,Y)

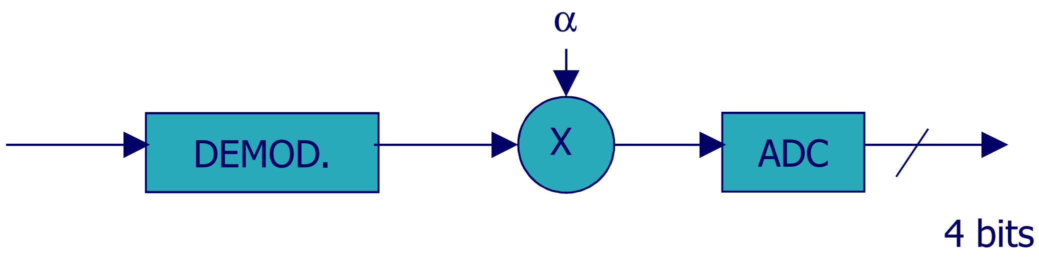
AT THE MOMENT, IT WAS PROVEN FOR Q=2 UP TO 12.

…STILL NEED A PROOF FOR $Q > 12$

SIMULATION RESULT : CASE 1

- DUO BINARY TURBO-CODE

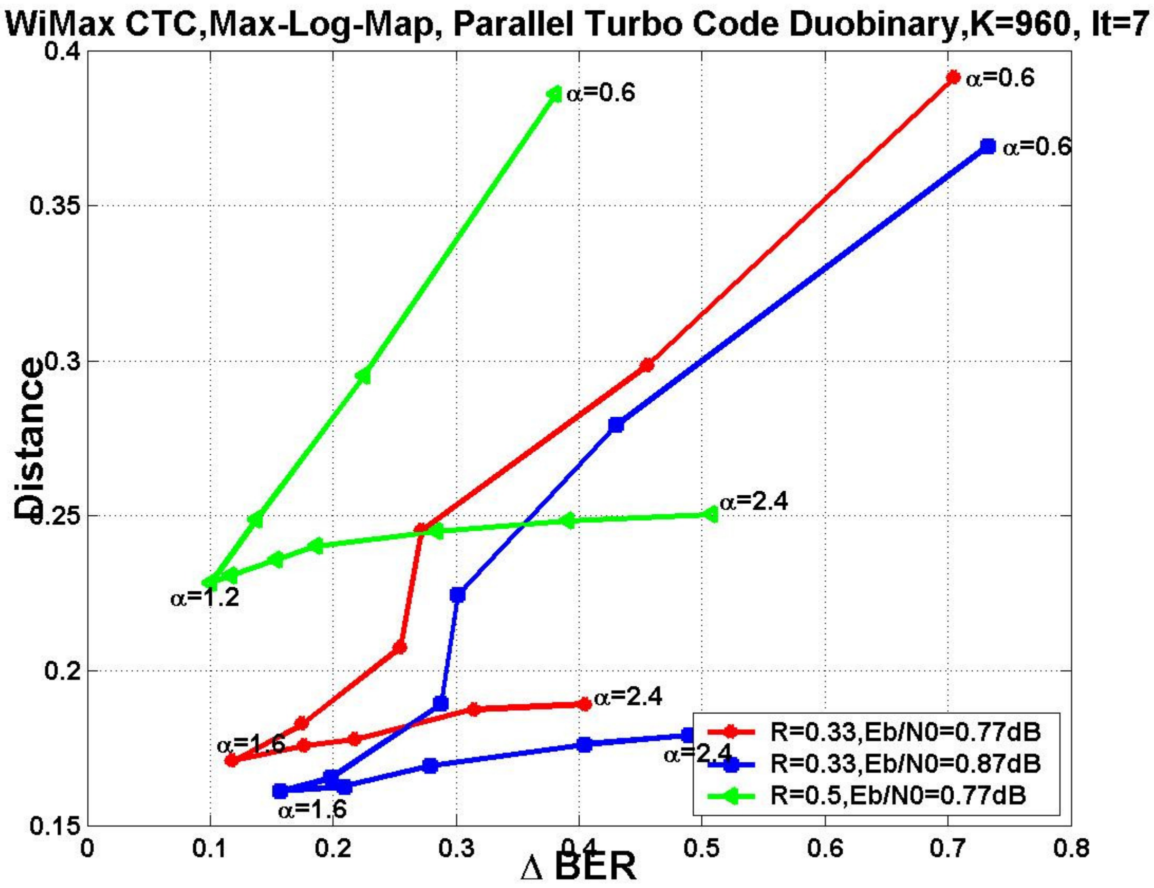
- SEARCH FOR THE OPTIMAL QUANTIZATION FACTOR



- IF α too small => ERASURE
- IF α too big => SATURATION (HARD DECODER)

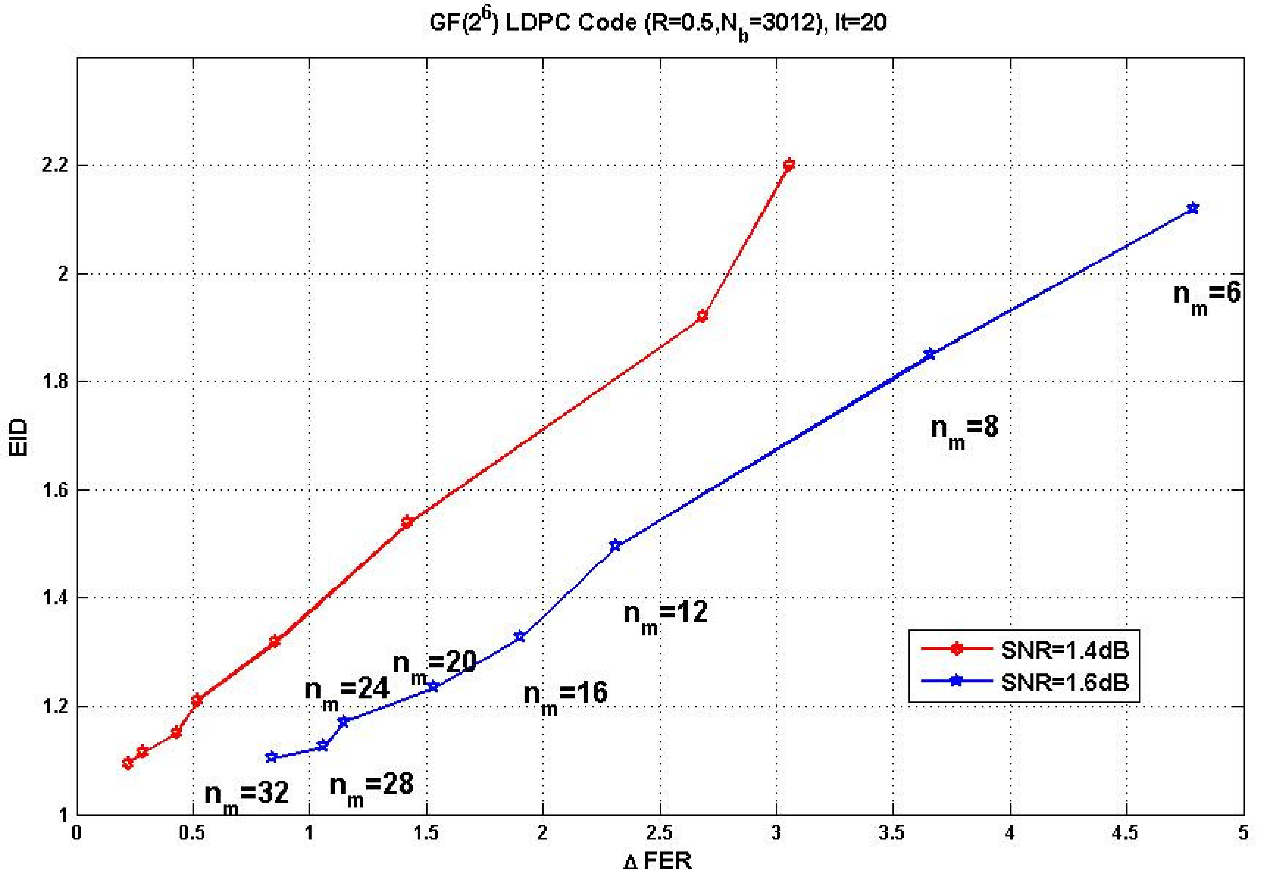
THERE IS AN OPTIMAL VALUE OF α

SIMULATION RESULT : CASE 1



SIMULATION RESULT : CASE 2

- **NON BINARY LDPC DECODER ON GF(64) USING EMS ALGORITM**



Lab-STICC
CNRS

OUTLINE

- Motivation
- Reduced Monte-Carlo Simulation
- Entropy Inspired Distance
- Hardware Emulation
- Conclusion

Software simulation

Three methods to reduce the simulation time:

a) code optimization

b) powerful computers

c) parallel computing
(One Mbps for a turbo-decoder with a cluster of 16 PCs)

also use hardware emulation

Current methodology

Software

Algorithm

C programs

Compilation

Validation/optimization with long simulations

Fix specifications

Hardware

VHDL programs

Synthesis, place and route operations

Validation

Final prototype

Proposed methodology

Software

Algorithm

C programs

Compilation

Validation/optimization

Fix algorithm + Set of non-specified parameters

Hardware

Generic VHDL programs, IP

Synthesis, place and route operations (on FPGA)

Hardware simulation/validation

Final prototype

Channel emulation

Type of communication channel:

- AWGN
- Rice
- Rayleigh

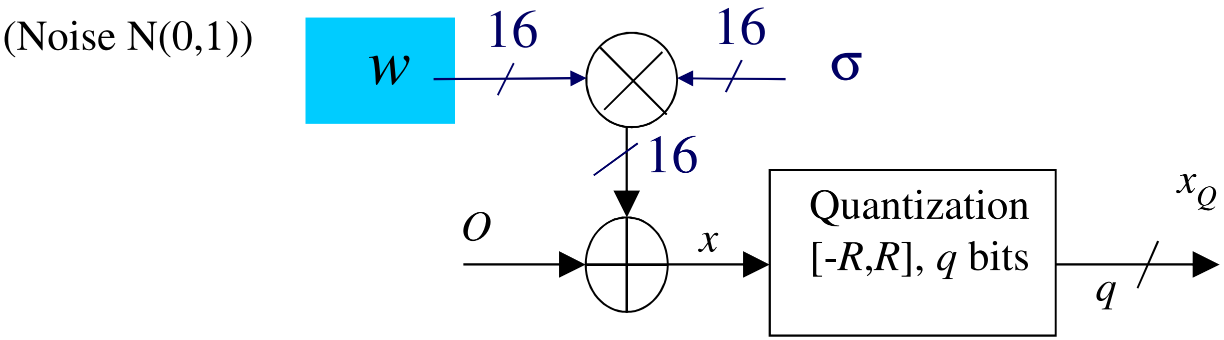
...

All those channels can be derived from Gaussian Noise (with ARMA filter, non-linear operators).

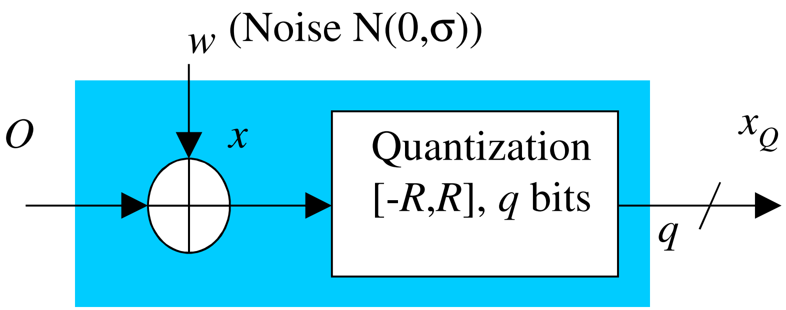
=> Need a White Gaussian Noise Generator (WGNG)

New idea

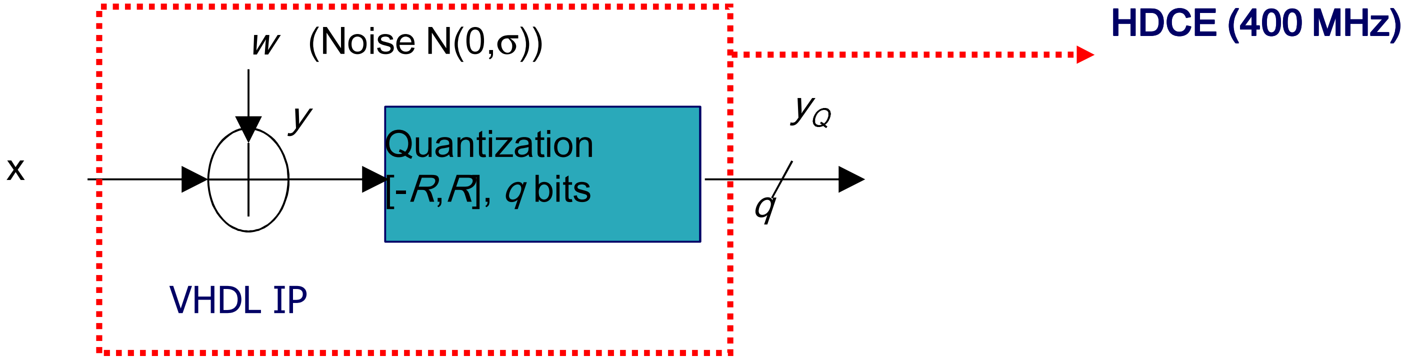
- Straigth method: Direct emulation of the AWGN channel.

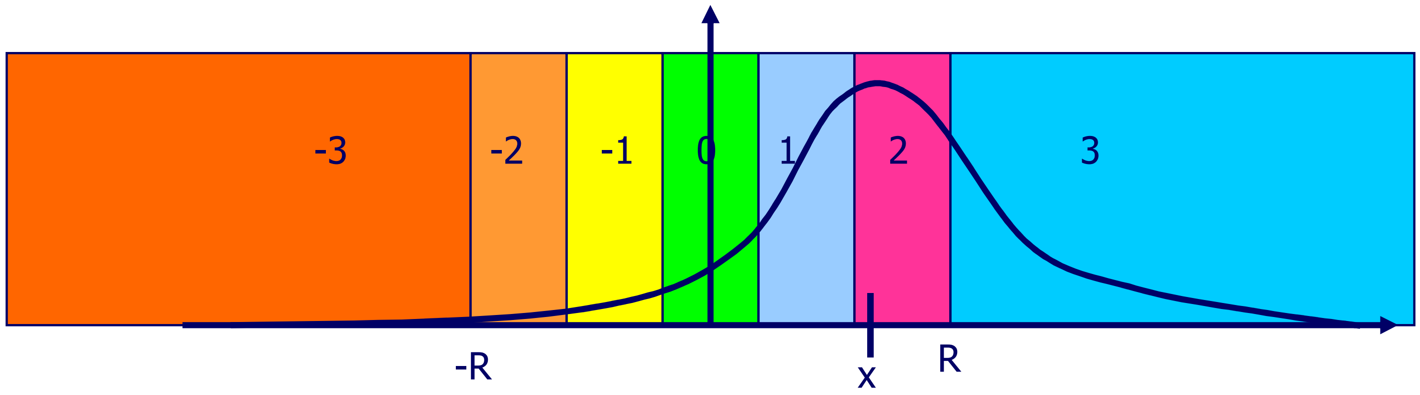


- New method: emulate channel + quantization.



DISCRETE HARDWARE CHANNEL EMULATOR

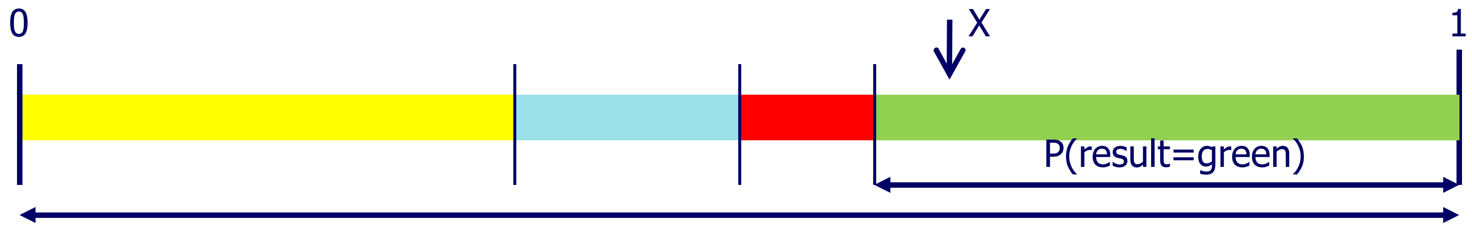




- **FOR A GIVEN *x*, σ and *R*, y_Q is EQUIVALENT TO A DISCRETE R.V.**

Principle of generation of a discrete random variable.

- 1D method
  - Draw a random number between [0,1] and see where it falls



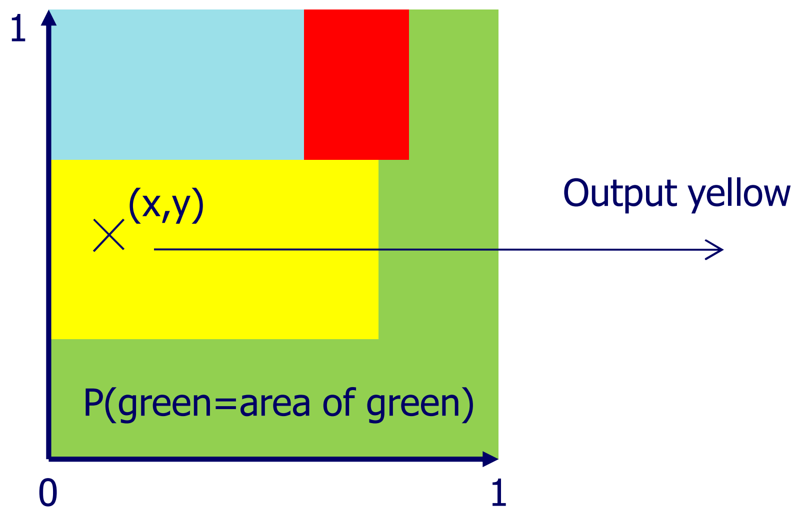
(Discrete random variable with N=4 values, represented by color)

- Complex to implement: the value x needs to be compared to all the thresholds (N-1 comparizons).

- Solution : go towards 2D.

Principle of generation of a discrete random variable.

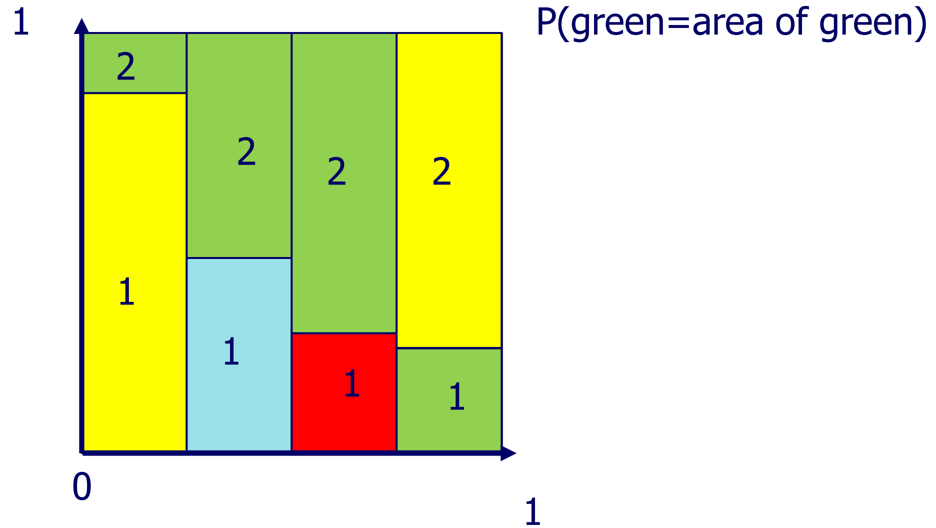
- 2D method
  - Principle: generate two random variables x,y between [0,1] and see where it falls.



  - No simplification => need a structure.

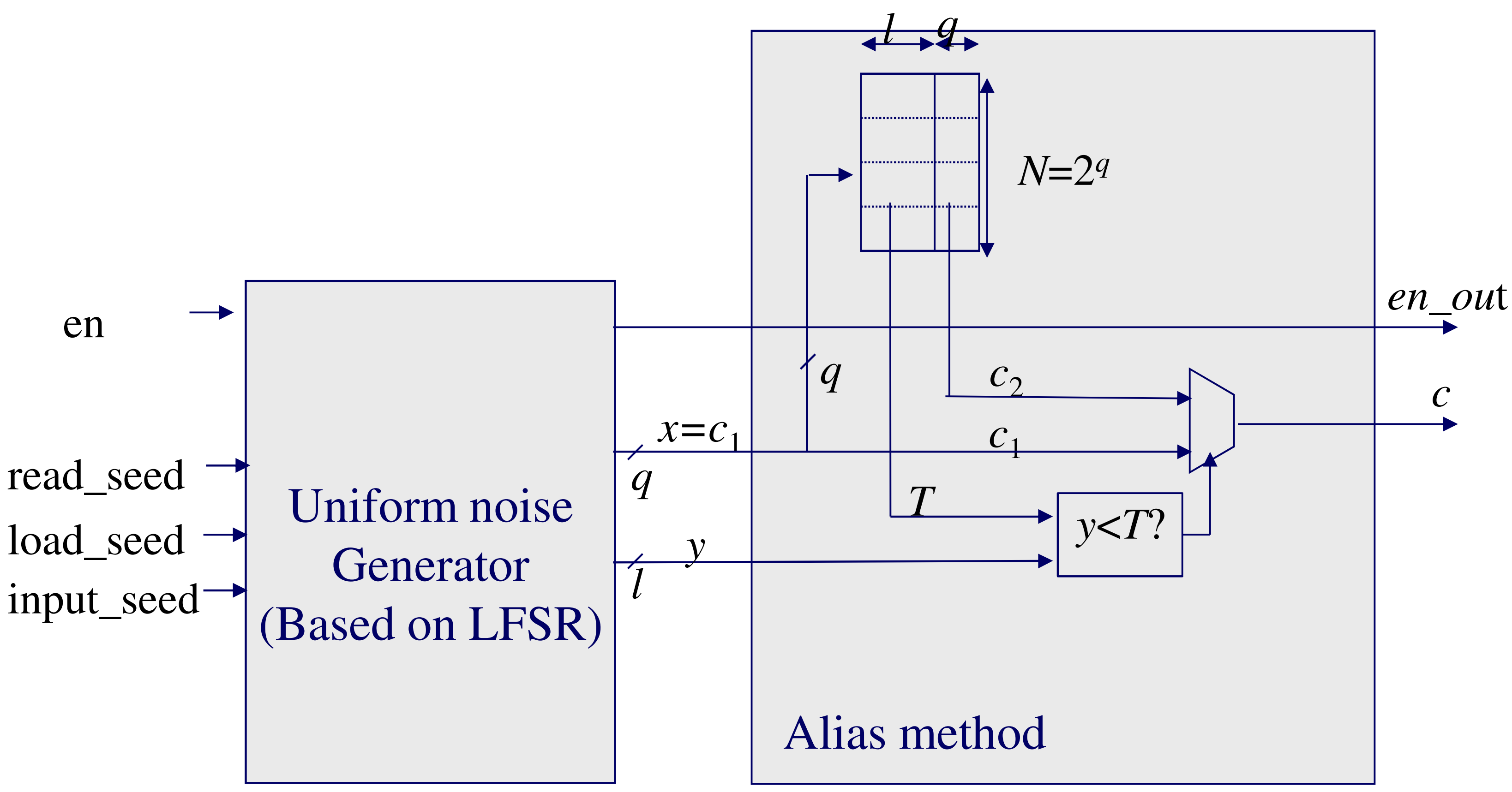
Principe of generation of a discrete random variable.

- 2D method: with x, select a column, with y, select color 1 or 2 of the column.

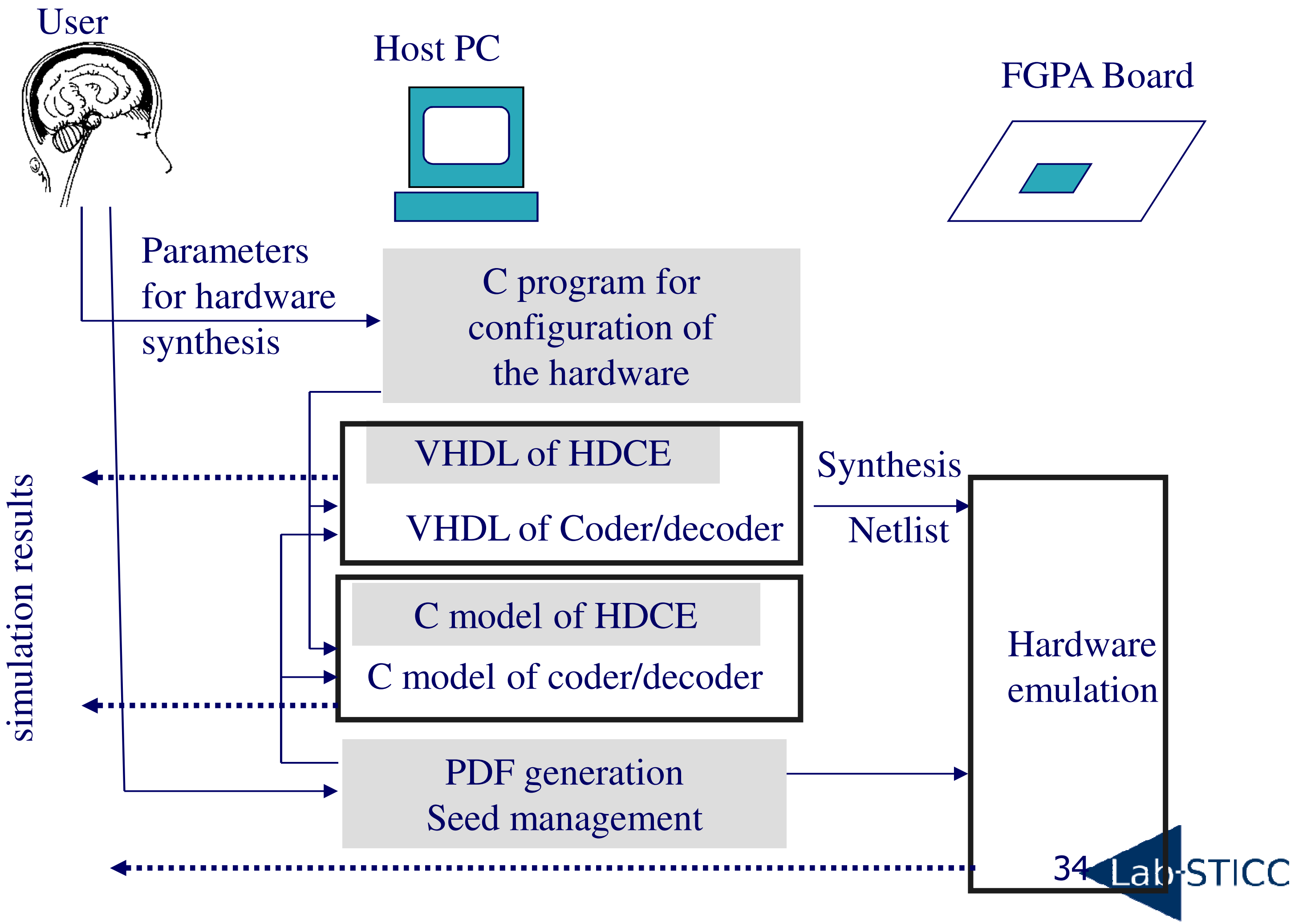


- ◇ X: random number between 1 to N to select color 1 (c_1), then, read color 2 (c_2) and the threshold T in a memory and compare y to t to select c_1 or c_2.

Hardware Discrete Channel Emulator.



Coherent set of simulation/emulation



OUTLINE

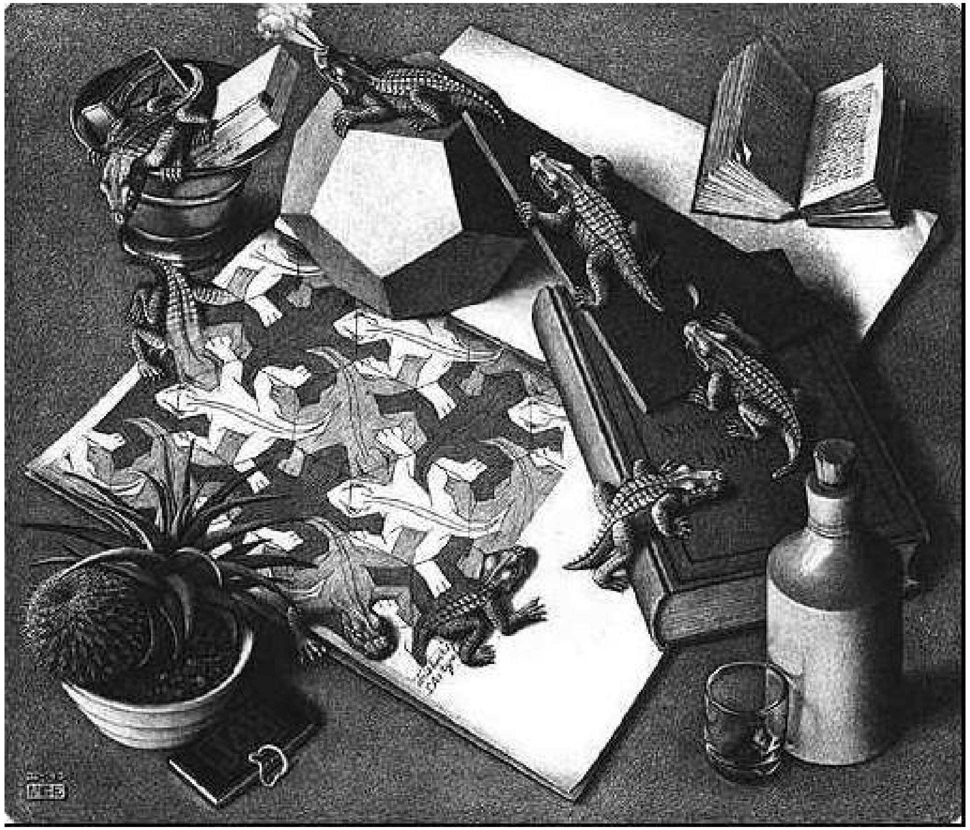
- Motivation
- Reduced Monte-Carlo Simulation
- Entropy Inspired Distance
- Hardware Emulation
- Conclusion

CONCLUSION

- PROPOSITION OF 2 METHODS TO AVOID MONTE-CARLO SIMULATION :
  - Reduced Monte Carlo Simulation
  - Entropy Inspired Distance
- THOSE TWO METHODS ALLOW TO TRADE OFF SPEED VS PRECISION.

  => NEED TO BE VALIDATED BY A DIRECT MONTE-CARLO
  - If you have a generic VHDL, can use HDCE to replace software simulation by hardware simulation (factor of 1000 in speed).

Conclusion



M.C. Escher (1898 - 1972)

Designing an iterative decoder is still an art…

PUBLICATION

- EID : A. Singh, A. Al-Ghouwayel1, G. Masera, E. Boutillon, " New Performance Evaluation Metric for Sub-Optimal Iterative Decoders", IEEE Communications letters, nov. 2008.

- E. Boutillon, C. Douillard, G. Montorsi, " Iterative Decoding of Concatenated Convolutional Codes: Implementation Issues", Transactions of the IEEE, vol. 95, n°6, june 2007.

- E. Boutillon, Y. Tang, C. Marchand, P. Bomel, “Hardware Discrete Channel Emulator”, The 2010 International Conference on High Performance Computing & Simulation (HPCS 2010), pp 452-458, Caen, June 2010.

- http://www.ict-davinci-codes.eu/