IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

EPIC Grant Agreement
No. 760150

Lab-STICC

Design of Next-Generation Tbps Turbo Codes

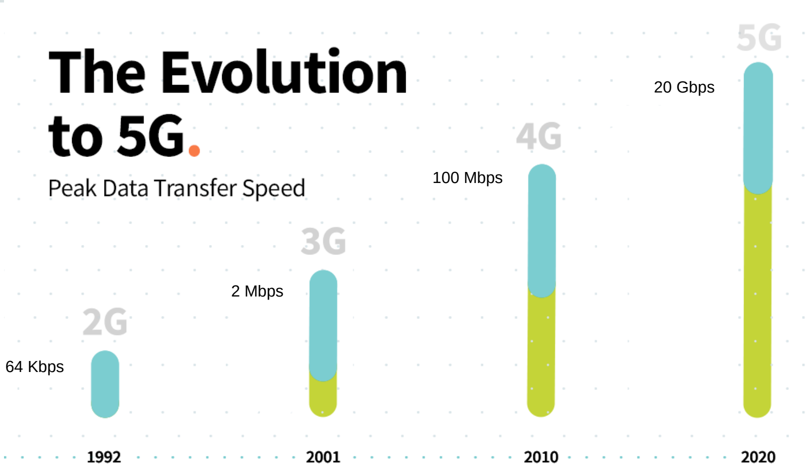
Presented by:	**Vinh Hoang Son Le**
Thesis directors:	**Catherine Douillard** **Emmanuel Boutillon**
Supervisor:	**Charbel Abdel Nour**
Reviewers:	**Charly Poulliat** **Guido Masera**
Examiner:	**Christophe Jego**
Invited Members:	**Jean-François Helard** **David Gnaedig**

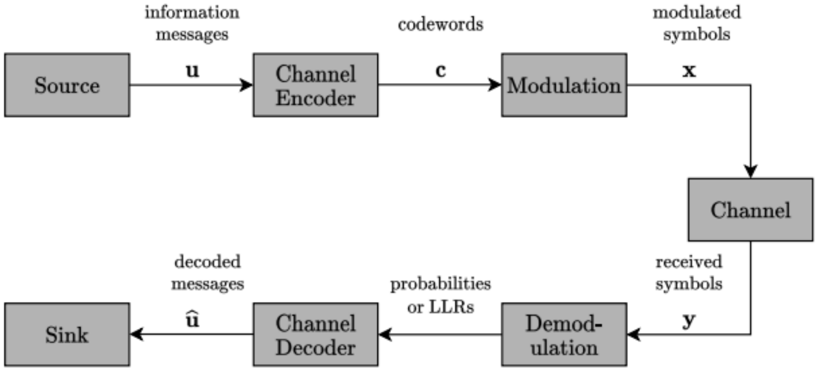
Introduction

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

INTRODUCTION

- The evolution of the data throughput
  - From 2G with 64 Kbps to 5G with 20 Gbps
  - With this trend: beyond 5G?
- With the advent of THz communications
  - Data throughput: hundreds of Gbps, up to Tbps
- Forward Error Correction (FEC)
  - Plays a critical role in enabling the communication link
  - Use redundancy information to correct corrupted information





INTRODUCTION

- The European H2020 project EPIC:
  - Develop FEC technologies for wireless Tbps use cases

« The upgrade to Tbps wireless data rates will not be smooth. The improvement carried by silicon technology progress will significantly fall short of meeting the Tbps FEC challenge »

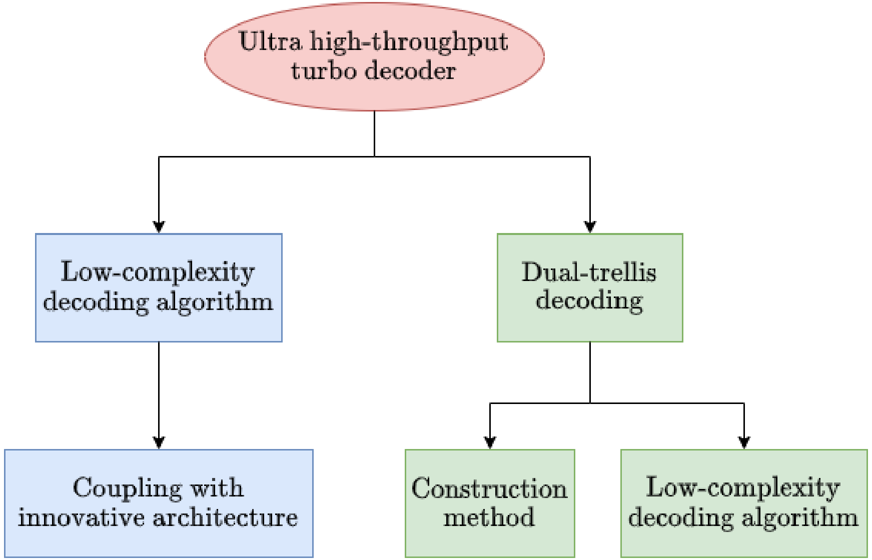
- Major algorithmic and architectural innovations are required
  - Polar codes (5G NR)
  - LDPC codes (5G NR)
  - Turbo codes (LTE Advanced Pro)

INTRODUCTION

Objectives, contributions

- **The thesis focuses on turbo decoders**
  - Allow the decoder to achieve Tbps transmission.
  - Other criteria: complexity, latency, energy, flexibility…

- **Contributions**
  - Novel low-complexity decoding algorithm
  - Coupling with innovative very high-throughput decoder architecture
  - Study of turbo decoding using the dual-trellis.


Ultra high-throughput turbo decoder
Low-complexity decoding algorithm
Dual-trellis decoding
Coupling with innovative architecture
Construction method
Low-complexity decoding algorithm

OUTLINE

1. TURBO CODES
2. NOVEL LOW-COMPLEXITY DECODING ALGORITHM: THE LOCAL SOVA
3. LOCAL-SOVA WITH UXMAP ARCHITECTURE
4. DECODING WITH THE DUAL-TRELLIS
5. CONCLUSION

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

TURBO CODES

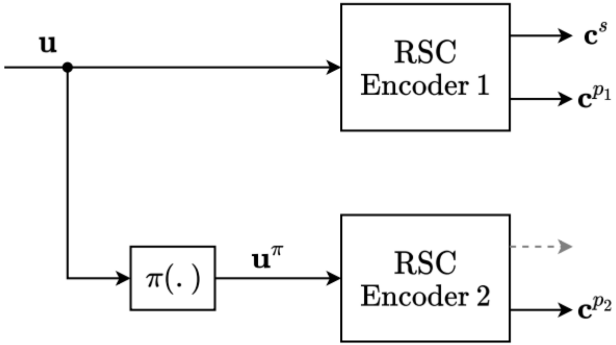
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

TURBO CODES

Encoding & decoding

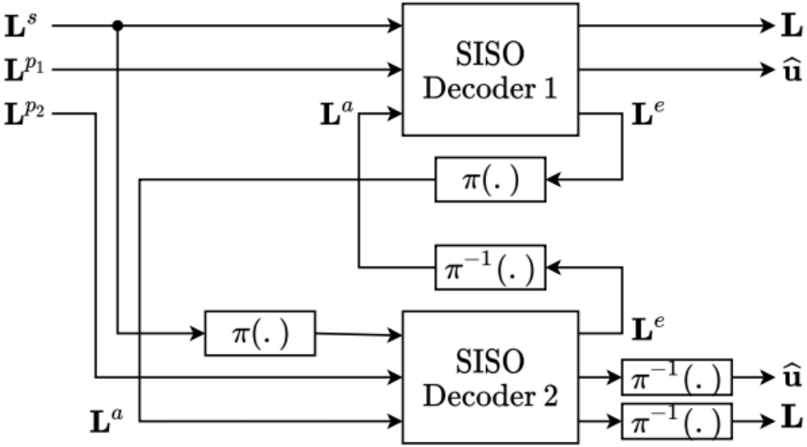
Encoding

- Parallel concatenated convolutional codes
- Recursive systematic convolutional encoders
- Interleaver



Decoding

- Iterative decoding: 1 iteration = 2 half-iterations
- Producing *extrinsic information* each half-iteration
- Soft-input soft-output (SISO) decoders.



TURBO CODES

SISO decoder: Max-Log-MAP algorithm

$$\bar{\gamma}_{k,k+1}(x) = \bar{\Lambda}^{a,x^s}_{u,k} + \sum_{i=0}^{\eta-1} \bar{\lambda}^{(i)}_k x^{(i)}$$

Branch Metrics

$$\bar{\alpha}^m_k = \max *_{\forall m'} \left(\bar{\alpha}^{m'}_{k-1} + \bar{\gamma}^{m',m}_{k-1,k} \right)$$

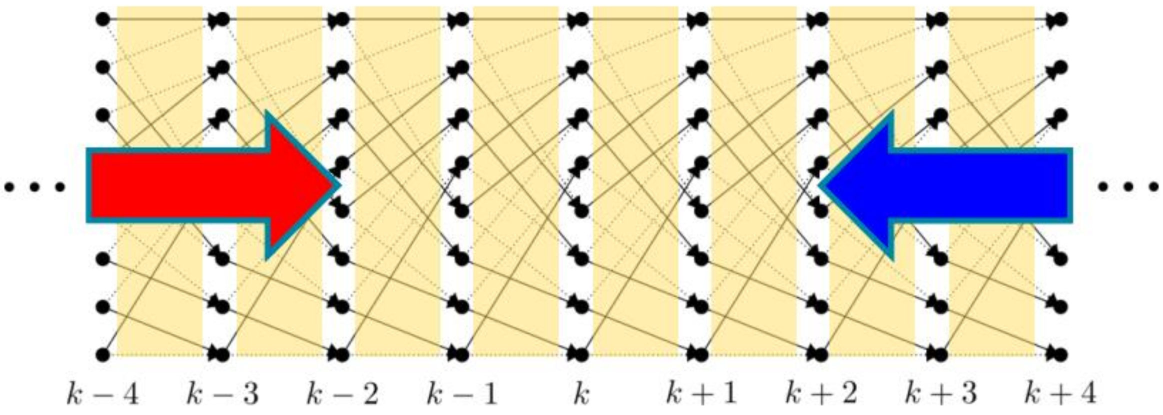
Forward Recursion

$$\bar{\beta}^m_k = \max *_{\forall m'} \left(\bar{\beta}^{m'}_{k+1} + \bar{\gamma}^{m',m}_{k,k+1} \right)$$

Backward Recursion

$$\Lambda^{(i)}_k = \max^*_{\forall m,m' | u^{(i)}_k = 0} \left(\bar{\gamma}^{m,m'}_{k,k+1} + \bar{\alpha}^m_k + \bar{\beta}^{m'}_{k+1} \right) - \max^*_{\forall m,m' | u^{(i)}_k = 1} \left(\bar{\gamma}^{m,m'}_{k,k+1} + \bar{\alpha}^m_k + \bar{\beta}^{m'}_{k+1} \right)$$

Soft Output

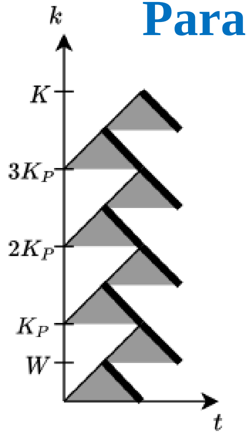


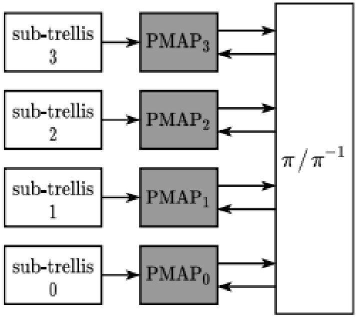
TURBO CODES

High-throughput decoder architectures: several Gbps

- Max-Log-MAP: split into smaller sub-trellises

Parallel MAP (PMAP)

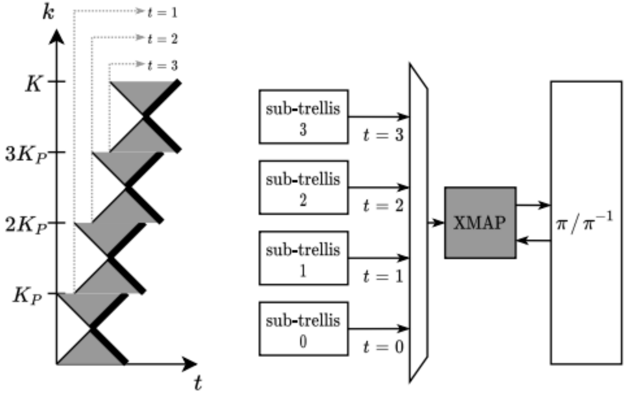




1 – 5 Gbps

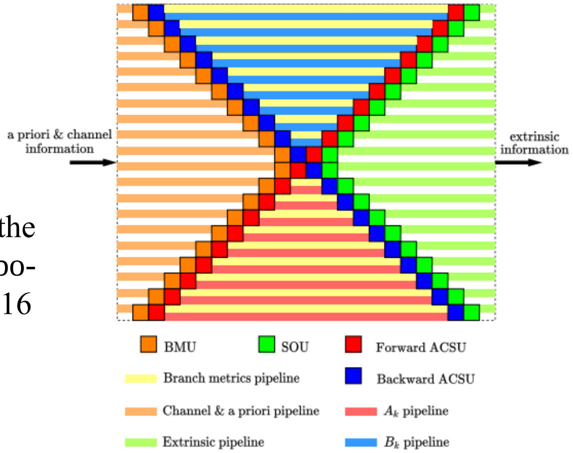
T. Ilnseher, F. Kienle, C. Weis, and N. Wehn, “A 2.12Gbit/s Turbo Code Decoder for LTE Advanced Base Station Applications,” 2012

X-MAP



1 – 5 Gbps

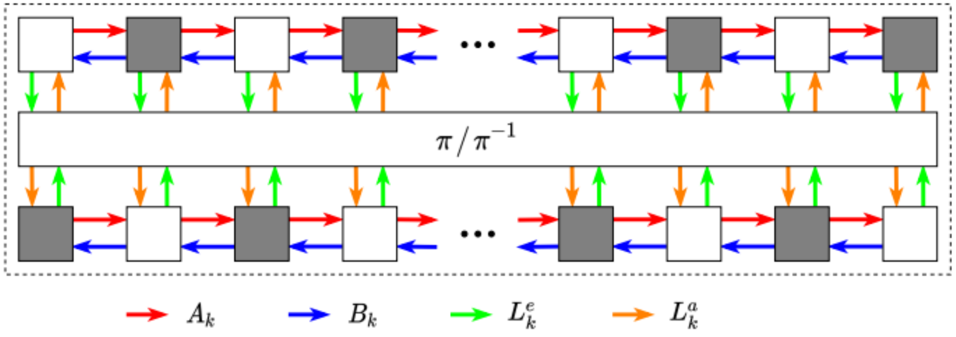
S. Weithoffer, F. Pohl, and N. Wehn, “On the applicability of trellis compression to Turbo-Code decoder hardware architectures,” 2016



TURBO CODES

Very high-throughput turbo decoder architecture: 10 – 100 Gbps

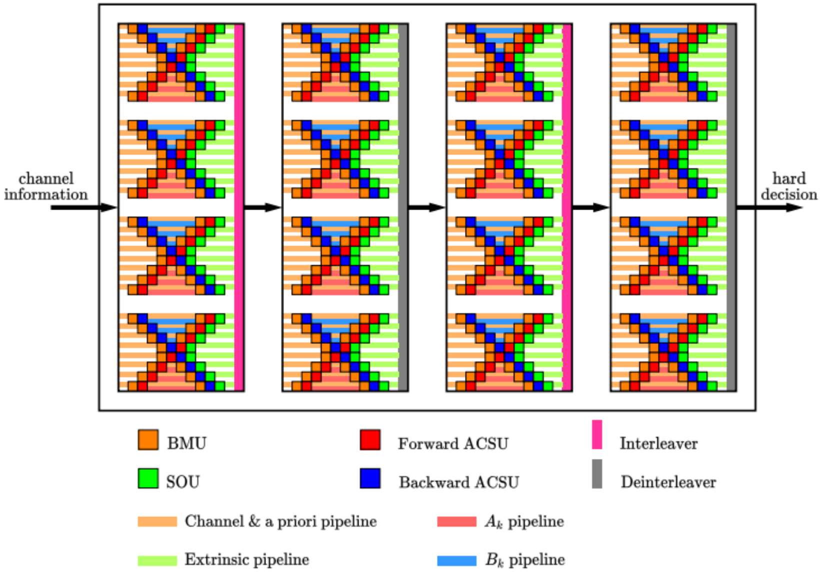
Fully-parallel MAP



10 – 40 Gbps

A. Li, L. Xiang, T. Chen, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "VLSI implementation of fully parallel LTE turbo decoders," 2016

Unrolled XMAP (iteration pipelined)

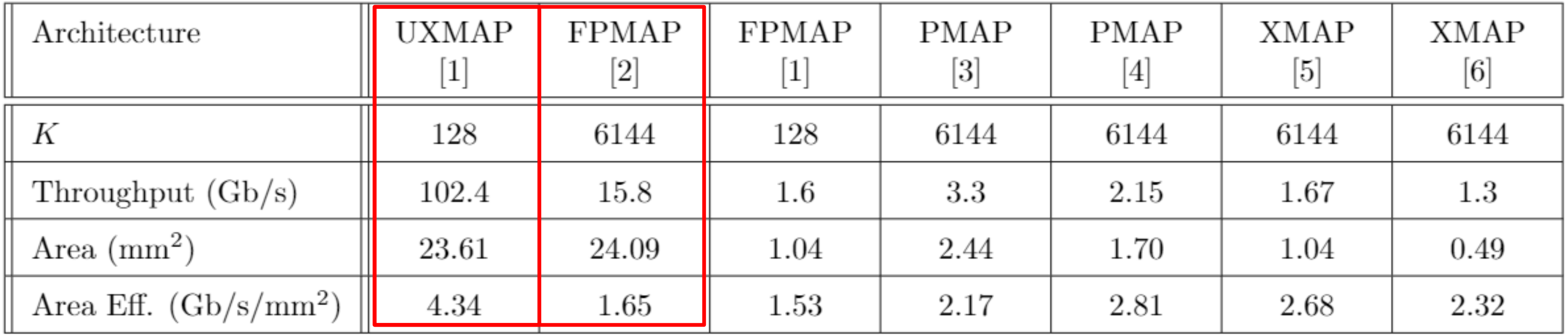


> 100 Gbps

S. Weithoffer, C. A. Nour, N. Wehn, C. Douillard, and C. Berrou, "25 Years of Turbo Codes: From Mb/s to beyond 100 Gb/s," 2018

TURBO CODES

Architecture comparison



Architecture	UXMAP [1]	FPMAP [2]	FPMAP [1]	PMAP [3]	PMAP [4]	XMAP [5]	XMAP [6]
K	128	6144	128	6144	6144	6144	6144
Throughput (Gb/s)	102.4	15.8	1.6	3.3	2.15	1.67	1.3
Area (mm^2)	23.61	24.09	1.04	2.44	1.70	1.04	0.49
Area Eff. (Gb/s/mm^2)	4.34	1.65	1.53	2.17	2.81	2.68	2.32

The UXMAP architecture can deliver ultra high-throughput with high area efficiency

[1] S. Weithoffer, C. A. Nour, N. Wehn, C. Douillard, and C. Berrou, “25 Years of Turbo Codes: From Mb/s to beyond 100 Gb/s,” 2018
[2] A. Li, L. Xiang, T. Chen, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, “VLSI implementation of fully parallel LTE turbo decoders,” 2016
[3] R. Shrestha and R. P. Paily, “High-Throughput Turbo Decoder With Parallel Architecture for LTE Wireless Communication Standards,” 2014
[4] T. Ilnseher, F. Kienle, C. Weis, and N. Wehn, “A 2.12Gbit/s Turbo Code Decoder for LTE Advanced Base Station Applications,” 2012
[5] G. Wang et al. “Parallel Interleaver Design for a High throughput HSPA+/LTE MultiStandard Turbo Decoder,” 2014
[6] S. Weithoffer, F. Pohl, and N. Wehn, “On the applicability of trellis compression to Turbo-Code decoder hardware architectures,” 2016

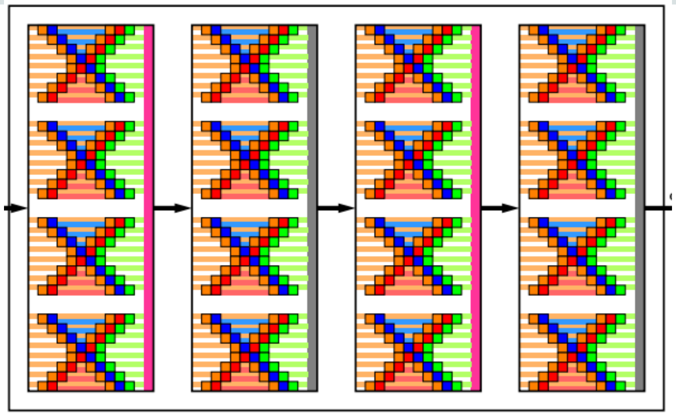
TURBO CODES

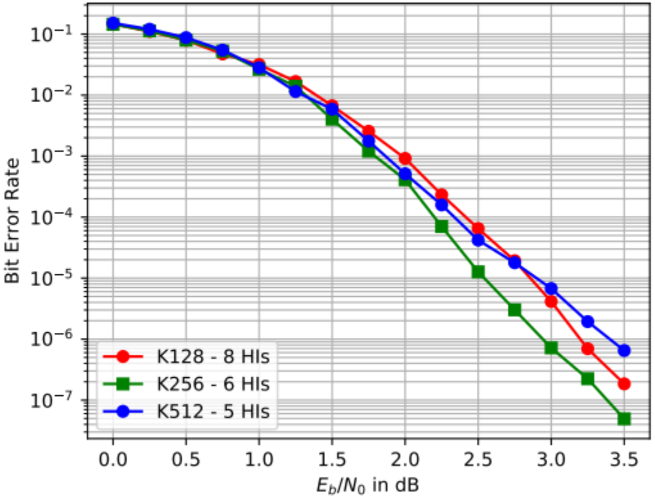
The UXMAP architecture

- Properties:
  - Performance increases with K with same # iterations
  - K bits decoded / clock cycle => throughput increases with K

Increase K and reduce the number of iterations

Configuration (K, #half-iter)	Area (mm^2)	Throughput (Gb/s)	Area efficiency (Gb/s/mm^2)
(128, 8)	12	102.4	8.5
(256, 6)	18	204.8	11.37
(512, 5)	30	409.6	13.65





TURBO CODES

What could be done for the UXMAP architecture?

- Alternative low-complexity decoding algorithm: increase throughput and area efficiency
  - Current algorithm: Max-Log-MAP

Novel algorithm: low-complexity, negligible loss in performance

- The use of high-radix decoding for high throughput
  - Decode multiple bits in parallel
  - Higher radix => lower latency & higher throughput
  - Max-Log-MAP: complexity increases exponentially with number of decoded bits in parallel

Novel algorithm: low-complexity even when use high radix (radix 8, radix 16)

THE LOCAL-SOVA

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

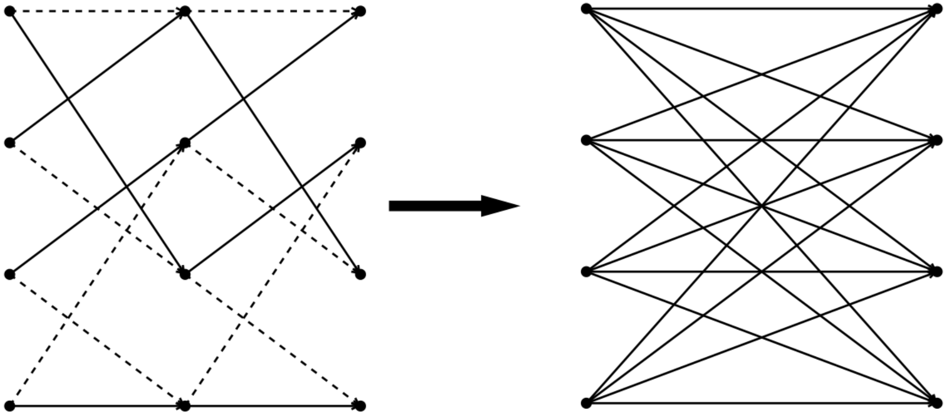
THE LOCAL-SOVA

High-radix decoding

- High-radix trellis
  - Concatenate consecutive radix-2 trellis sections
  - Decode more than 1 bit per section
  - Complexity can be higher

- Going from radix-2 to radix-4 in UXMAP:
  - Reduce pipeline stages => saving area, lower latency
  - Higher throughput

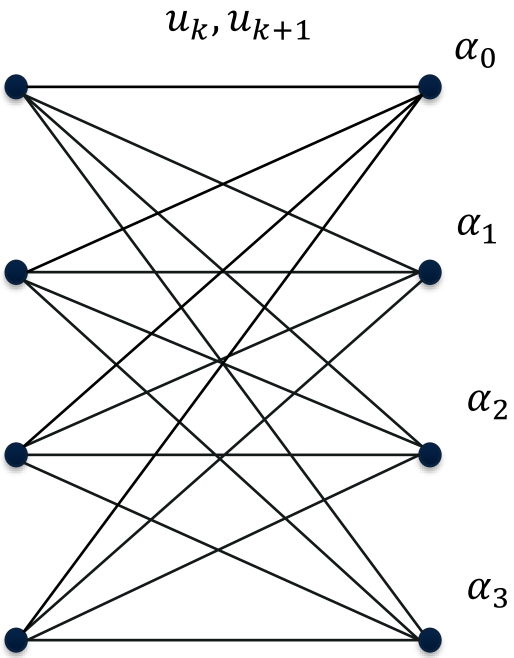
- Radix-8 and radix-16 are not suitable with Max-Log-MAP



THE LOCAL-SOVA

Radix-4 Max-Log-MAP

- **Calculation of alpha (ACSU)**
  - Max 4 branches: α_0
  - Max 4 branches: α_1
  - Max 4 branches: α_2
  - Max 4 branches: α_3
- **Soft-output bit u_k (SOU)**
  - Max 8 branches: $L_k(0)$
  - Max 8 branches: $L_k(1)$
  - $L(u_k) = L_k(1) - L_k(0)$
- **Soft-output bit u_{k+1} (SOU)**
  - Max 8 branches: $L_{k+1}(0)$
  - Max 8 branches: $L_{k+1}(1)$
  - $L(u_{k+1}) = L_{k+1}(1) - L_{k+1}(0)$



THE LOCAL-SOVA

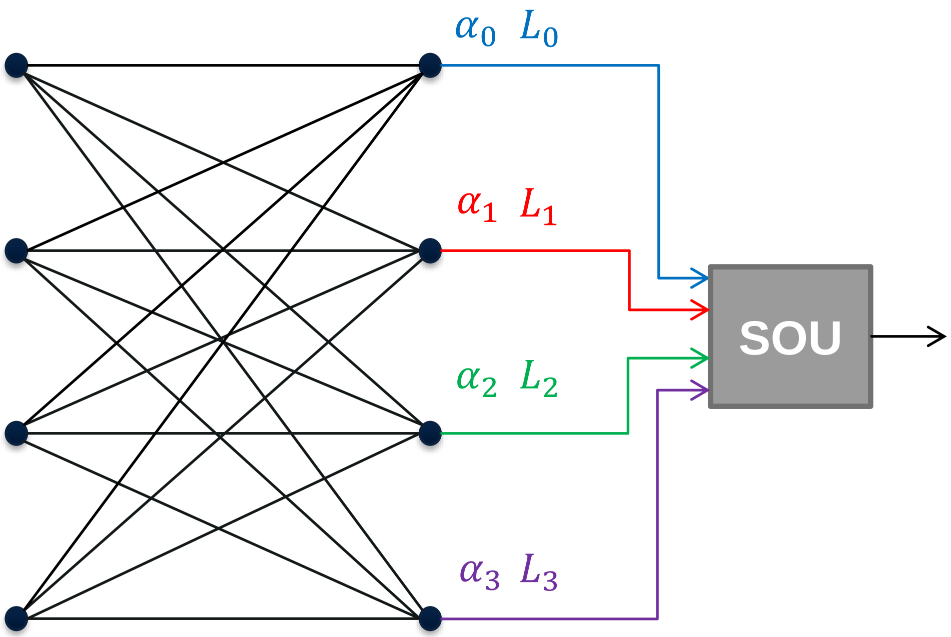
The idea

- Incorporate alpha calculation and soft-output calculation

- Alpha calculation:
  - Carry a soft information for each winning branch

- Soft-output:
  - Process only 4 branches per decoded bit

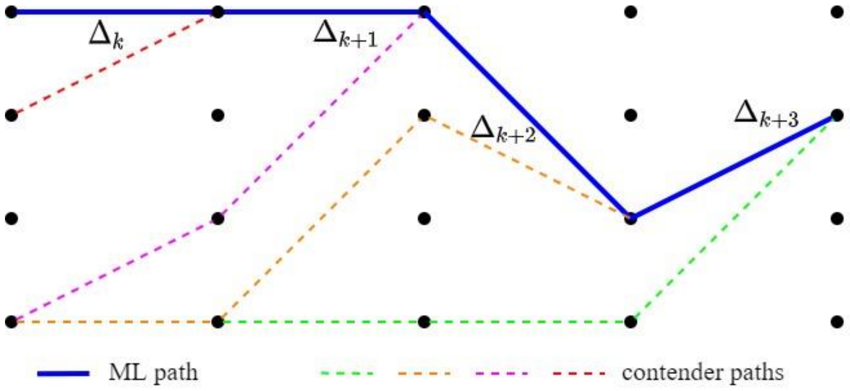
- **Tasks**:
  - Define soft-information
  - How to use them to get soft-output?

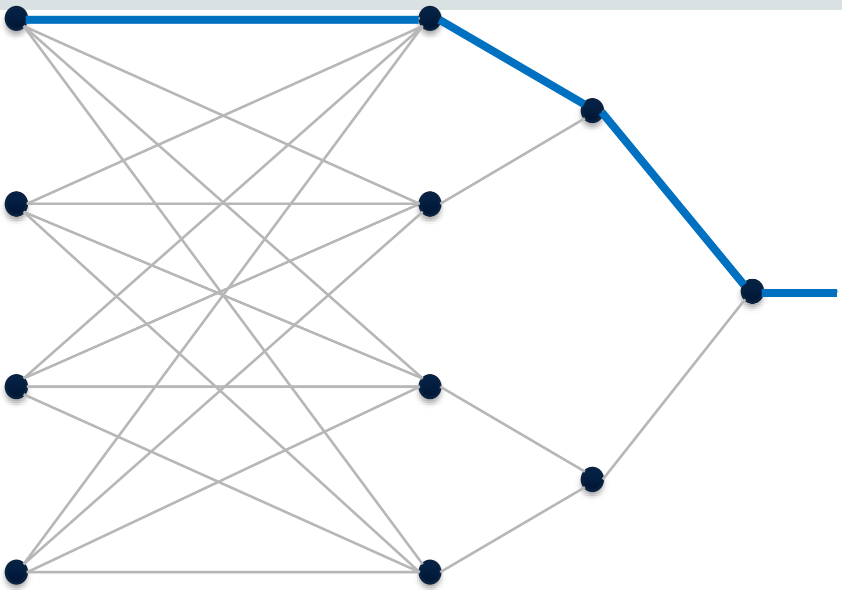


THE LOCAL-SOVA

The Local-SOVA based on SOVA

- Final winning branch:
  - Compared with several branches
  - Similar to the Soft-Output Viterbi Algorithm



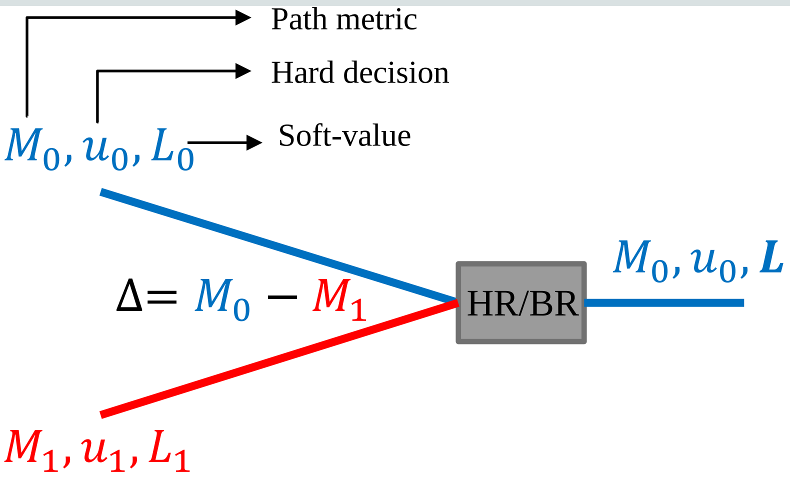


- Update the soft-output by comparing the ML path with contender paths
  - Apply to the Local-SOVA

THE LOCAL-SOVA

The update rules: Hagenauer’s rule and Battail’s rule

- When two branches meet each other:
  - Select the highest path metric & hard decision
  - Update the soft value



- Update rules:
  - Hagenauer’s rule: different hard decisions
  - Battail’s rule: same hard decision

- **Remarks**:
  - Single architecture for both rules: 1 adder, 1 compare-select
  - Employ only HR = lower complexity. Performance?

- **HR:** if $u_0 \neq u_1$

  $$L = \min(L_0, \Delta)$$

- **BR:** if $u_0 = u_1$

  $$L = \min(L_0, L_1 + \Delta)$$

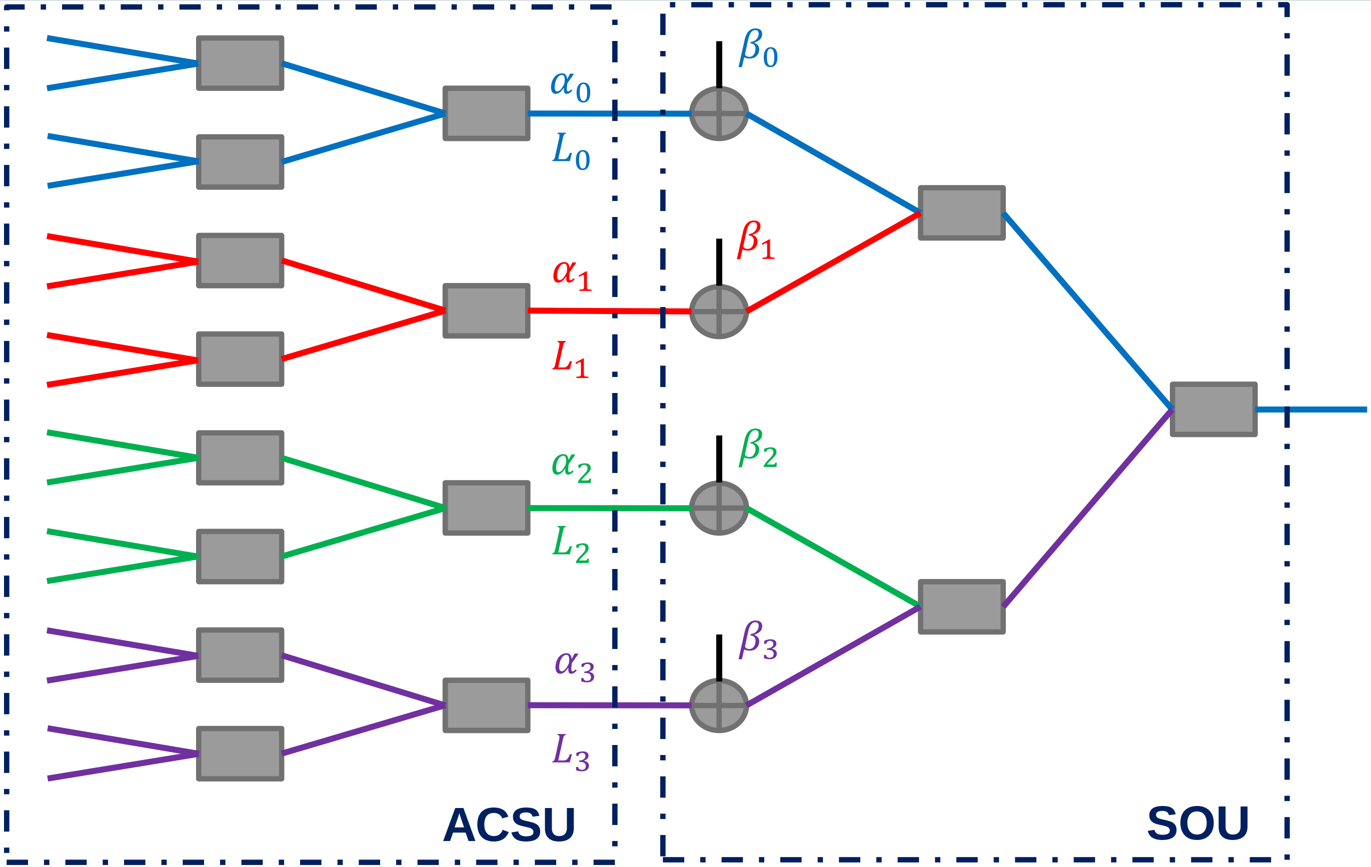
THE LOCAL-SOVA

The Local-SOVA architecture

- Local-SOVA:
  - BMU (idem MLM)
  - ACSU
  - SOU

▨ : HR/BR update

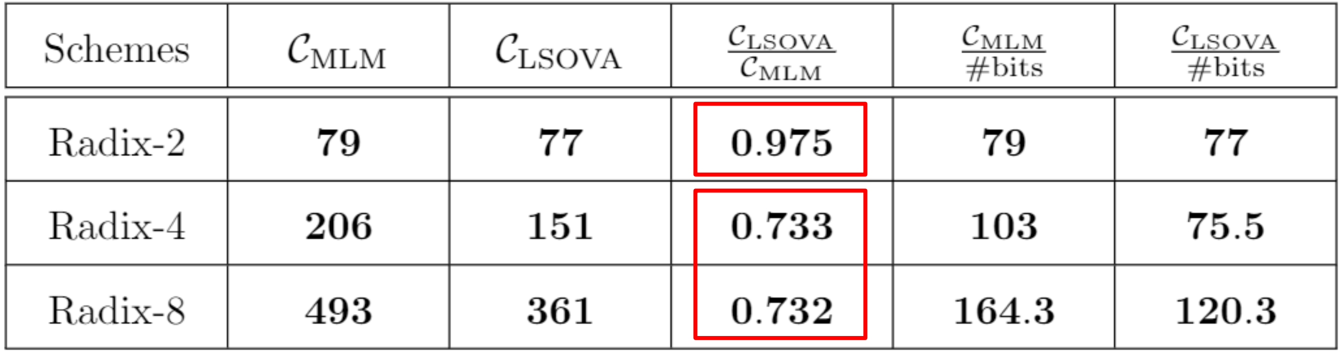
Max-Log-MAP is a particular instance of the Local-SOVA



THE LOCAL-SOVA

Complexity analysis

- Computational complexity for a trellis section:
  - Adder = Compare-Select = 1 unit
  - Same memory consumption
  - Same performance



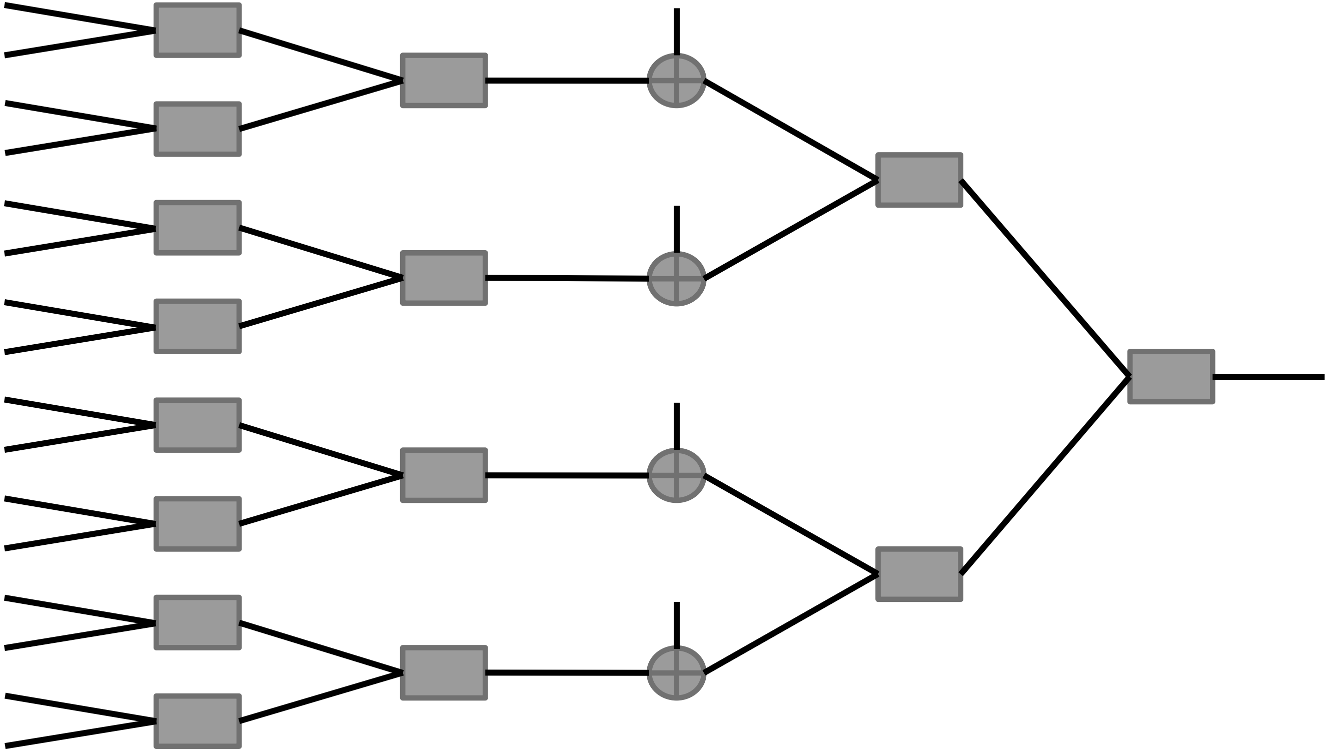
Schemes	$\mathcal{C}_{\text{MLM}}$	$\mathcal{C}_{\text{LSOVA}}$	$\frac{\mathcal{C}_{\text{LSOVA}}}{\mathcal{C}_{\text{MLM}}}$	$\frac{\mathcal{C}_{\text{MLM}}}{\#\text{bits}}$	$\frac{\mathcal{C}_{\text{LSOVA}}}{\#\text{bits}}$
Radix-2	79	77	0.975	79	77
Radix-4	206	151	0.733	103	75.5
Radix-8	493	361	0.732	164.3	120.3

The Local-SOVA is 27% less complex than the Max-Log-MAP for radix 4 & radix 8

THE LOCAL-SOVA

Further simplification for the Local-SOVA

- Simplifications:
  - Use HR only
  - Reduce complexity

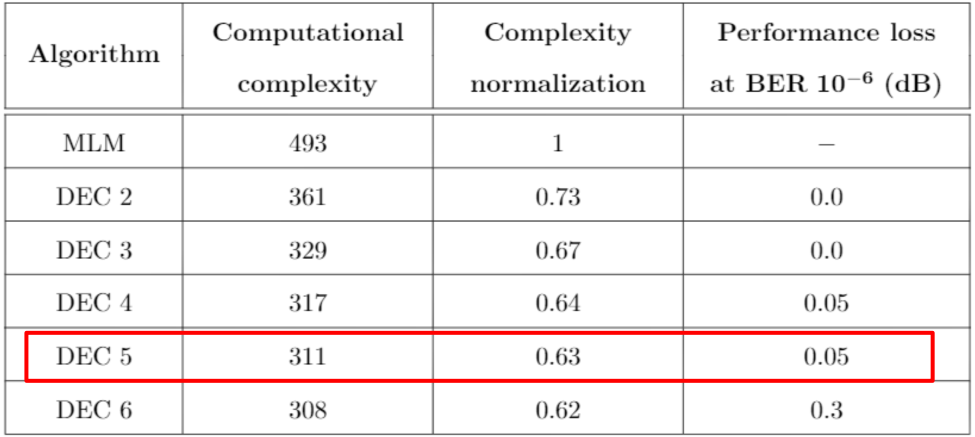


▨ : HR/BR

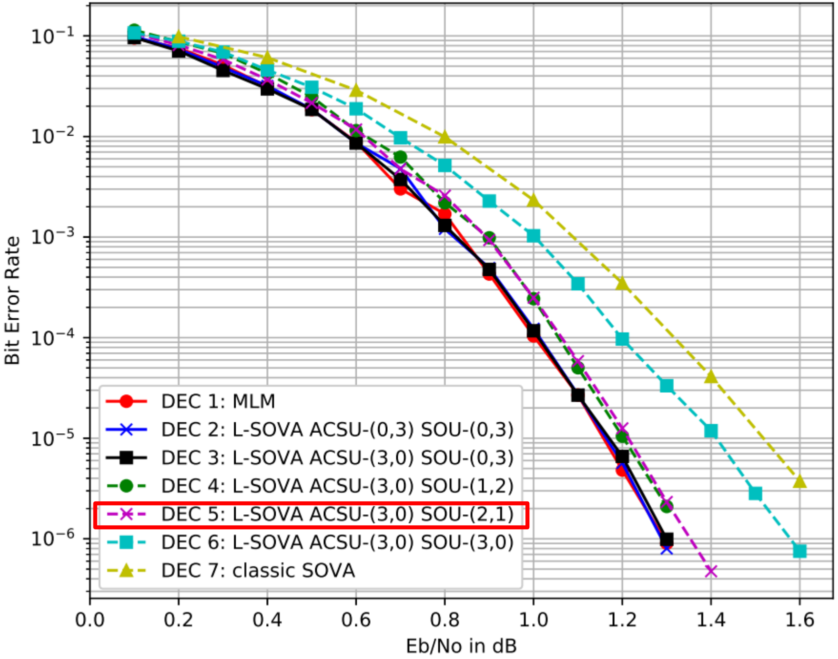
🟥 : HR only

THE LOCAL-SOVA

Applying simplification for radix-8 Local-SOVA



Algorithm	Computational complexity	Complexity normalization	Performance loss at BER 10^{-6} (dB)
MLM	493	1	–
DEC 2	361	0.73	0.0
DEC 3	329	0.67	0.0
DEC 4	317	0.64	0.05
DEC 5	311	0.63	0.05
DEC 6	308	0.62	0.3



Further lower complexity for high radix

Save 37% with negligible loss in performance

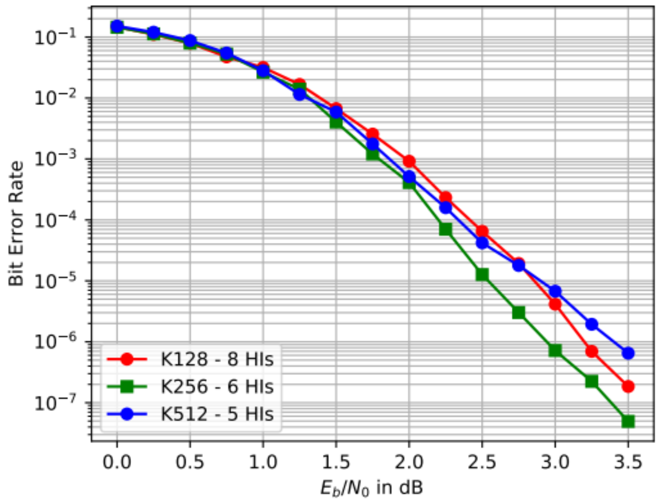
THE LOCAL-SOVA IN THE UXMAP ARCHITECTURE

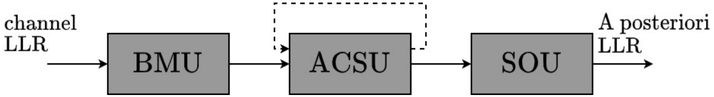
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

LOCAL-SOVA WITH UXMAP

Settings

- UXMAP architecture
  - UXMAP + radix-4 Max-Log-MAP
  - UXMAP + radix-4 Local-SOVA
  - UXMAP + radix-8 Local-SOVA, radix-16 Local-SOVA
- Performance in fixed-point 6-bit input
  - Compare with Max-Log-MAP
  - (K,HIs) = {(128,8), (256,6), (512,5)}
- Comparison conditions:
  - BMU, ACSU, SOU for 12 radix-2 trellis sections
  - 6 radix-4, 4 radix-8, and 3 radix-16 trellis sections.



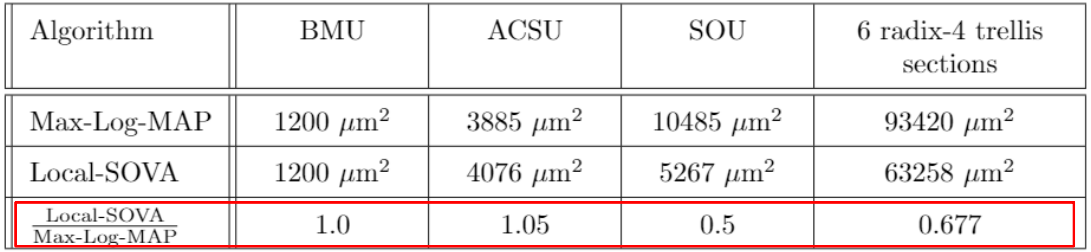


LOCAL-SOVA WITH UXMAP

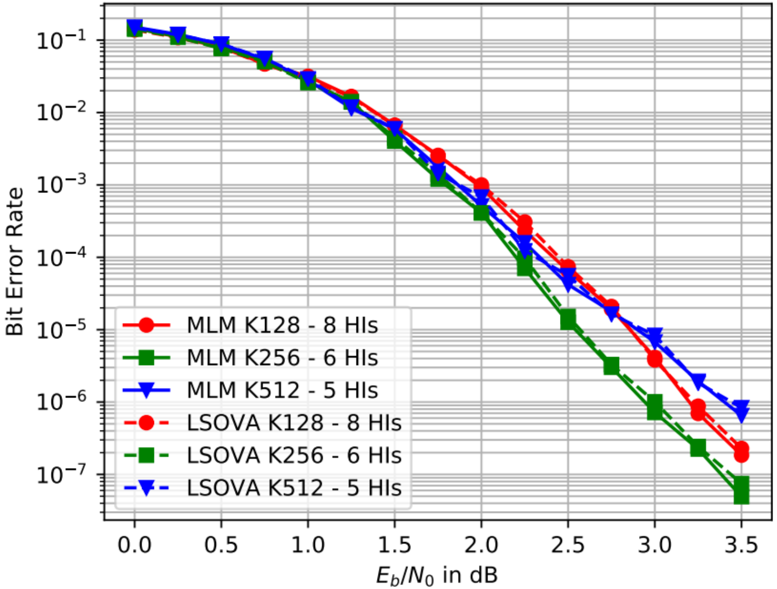
Radix 4

- Radix-4 Local-SOVA:
  - ACSU: HR/BR
  - SOU: HRs, except the last one is HR/BR
  - Same performance

- Area complexity



Algorithm	BMU	ACSU	SOU	6 radix-4 trellis sections
Max-Log-MAP	1200 μm^2	3885 μm^2	10485 μm^2	93420 μm^2
Local-SOVA	1200 μm^2	4076 μm^2	5267 μm^2	63258 μm^2
$\frac{\text{Local-SOVA}}{\text{Max-Log-MAP}}$	1.0	1.05	0.5	0.677



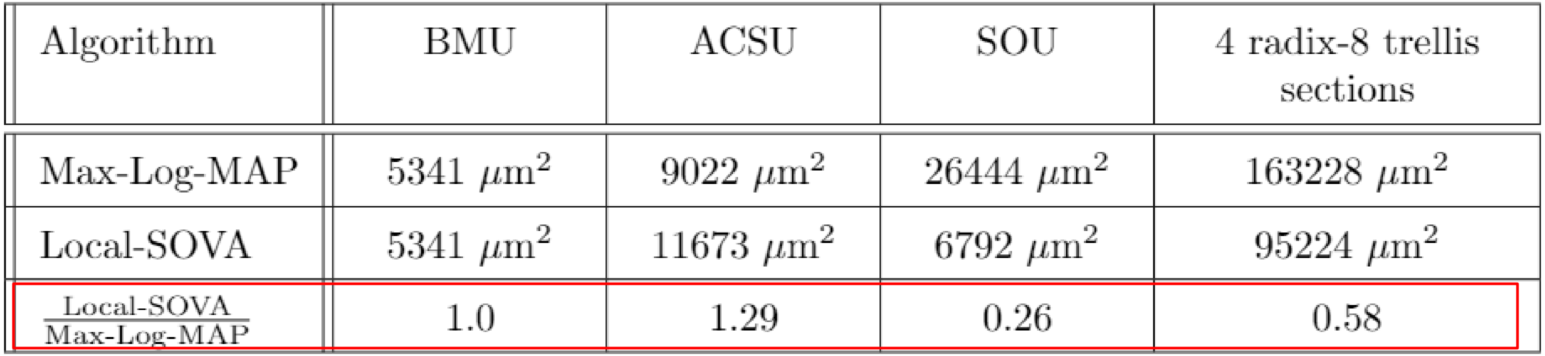
Lower-complexity implementation: save 33% area complexity

LOCAL-SOVA WITH UXMAP

Radix 8

- Generalize the radix-4 Local-SOVA with same structure
  - Same performance
  - Lower latency & higher throughput

- Area complexity

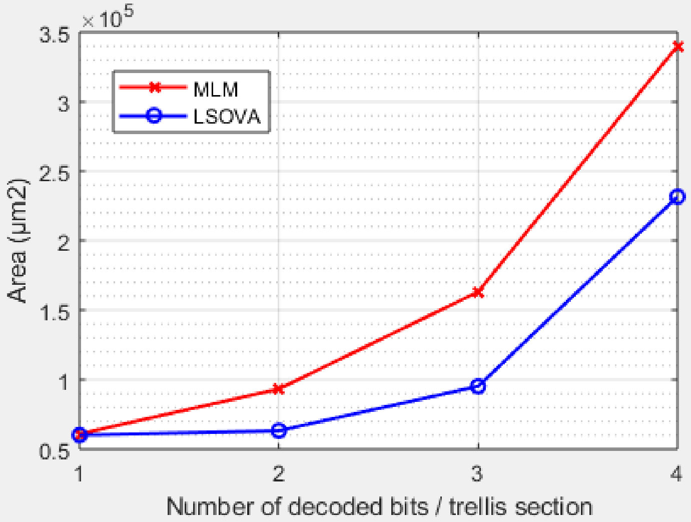


Algorithm	BMU	ACSU	SOU	4 radix-8 trellis sections
Max-Log-MAP	5341 μm^2	9022 μm^2	26444 μm^2	163228 μm^2
Local-SOVA	5341 μm^2	11673 μm^2	6792 μm^2	95224 μm^2
$\frac{\text{Local-SOVA}}{\text{Max-Log-MAP}}$	1.0	1.29	0.26	0.58

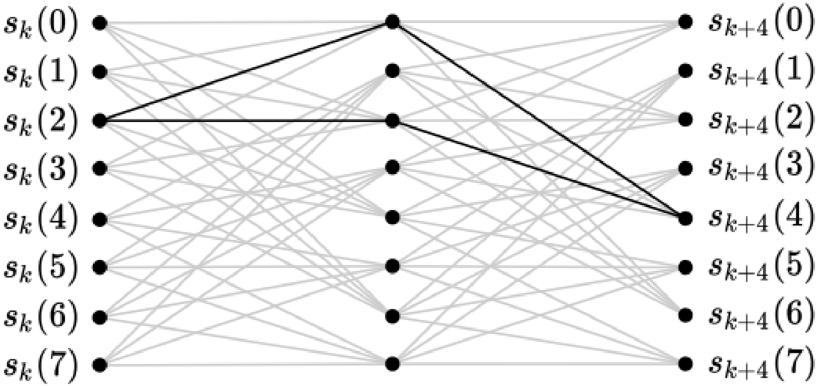
LOCAL-SOVA WITH UXMAP

Radix 16

- Radix-16:
  - The area increases exponentially
  - Due to the radix-16 ACSU



- Solution:
  - Parallel branches between states
  - Eliminate 1 branch in the BMU
  - ACSU is radix 8

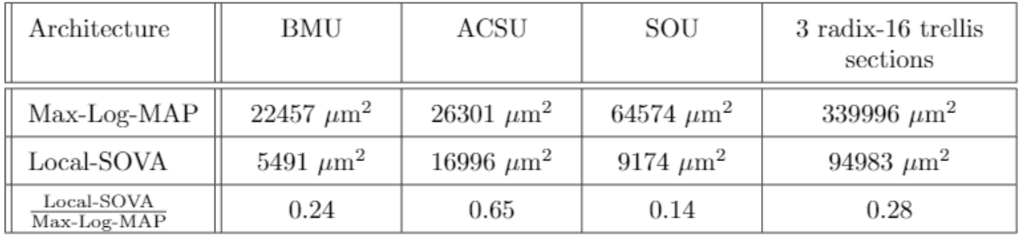


LOCAL-SOVA WITH UXMAP

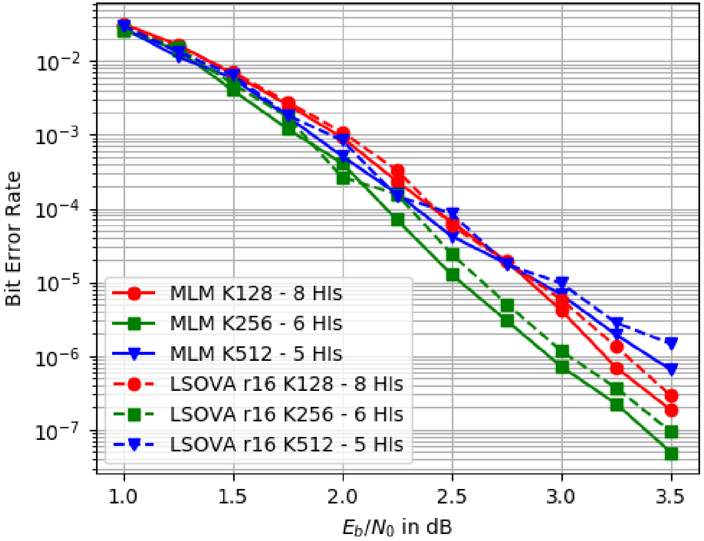
Radix 16

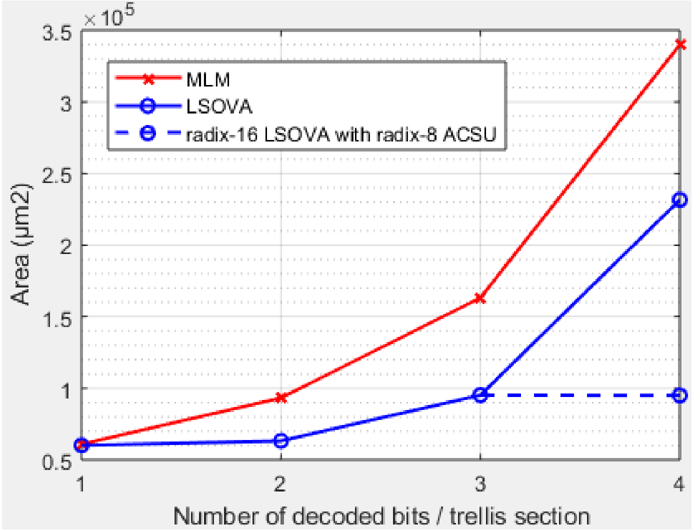
- Radix-16 Local-SOVA:
  - Lower latency & higher throughput
  - Performance: small degradation due to simplifications

- Area complexity (in μm^2)



Architecture	BMU	ACSU	SOU	3 radix-16 trellis sections
Max-Log-MAP	22457 μm^2	26301 μm^2	64574 μm^2	339996 μm^2
Local-SOVA	5491 μm^2	16996 μm^2	9174 μm^2	94983 μm^2
$\frac{\text{Local-SOVA}}{\text{Max-Log-MAP}}$	0.24	0.65	0.14	0.28





LOCAL-SOVA WITH UXMAP

Conclusion

- A new decoding algorithm: Local-SOVA
  - Is a general algorithm: Max-Log-MAP is an instance
  - Decoding high radix more efficiency
- Implementation with UXMAP architecture
  - Radix-4 Local SOVA is more suitable than radix-4 Max-Log-MAP
  - 33 % saving area => increase 33% in throughput (from 400 Gbps to 532 Gbps)
  - Radix 8, radix 16: lower latency by 1.5 and 2 times.
  - TurboLEAP project (*Turbo decoding with Less Energy, Area and more Parallelism for higher throughput*)

DUAL-TRELLIS DECODING

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

DUAL-TRELLIS DECODING

Why high coding rates?

- The demand for Tbps communication wireless link:
  - Wireless intra-device communications [1], mobile virtual/augmented reality [2],…
  - Move the radio frequency to above 100 GHz (Terahertz communications).

- At the PHY layer, the error control coding employ **high-rate schemes**: less redundancy bits
  - The available spectrum is more efficiently used.

- Require **high coding rates high-throughput** (up to Tbps) channel decoders.

[1] V. Petrov, A. Pyattaev, D. Moltchanov, and Y. Koucheryavy, ‘‘Terahertz band communications: Applications, research challenges, and standardization activities,’’ in Proc. 8th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT), Oct. 2016, pp. 183–190.

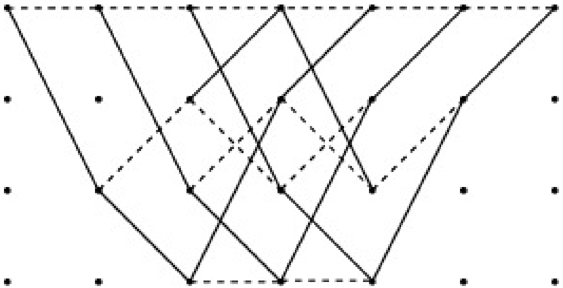
[2] T. S. Rappaport et al., “Wireless Communications and Applications above 100 GHz: Opportunities and Challenges for 6G and Beyond,” IEEE Access, vol. 7, July 2019, pp. 78729–57.

DUAL-TRELLIS DECODING

- **Encoding**: Puncturing rate-1/2 mother code → High-rate code with coding rate *k/n*
- **Decoding**:

Log-MAP or Max-Log MAP

- Decode on the **original trellis**
- Decode **1 bit** per trellis section



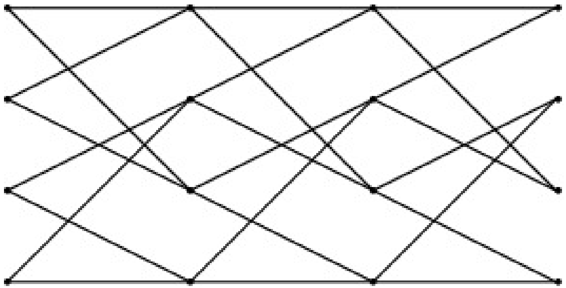
$$\underline{\mathbf{G}}\,\underline{\widetilde{\mathbf{H}}} = 0$$

Generator matrix of punctured CC rate k/n

Reciprocal parity check matrix for dual-code rate $(n-k)/n$

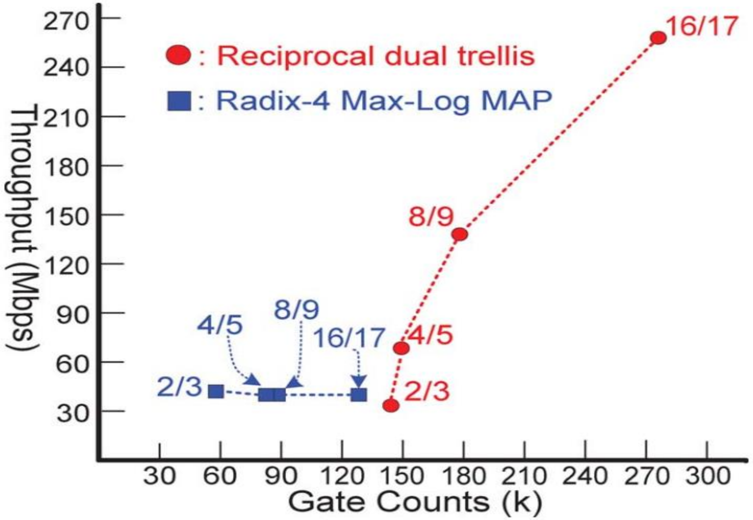
Dual-Log-MAP

- Decode on the **dual trellis**
- Decode ***k* bits** per trellis section



DUAL-TRELLIS DECODING

Advantages of dual-trellis decoding



[3] C. Lin, C. Wong, and H. Chang, “A 40 nm 535 Mbps multiple code-rate turbo decoder chip using reciprocal dual trellis,” IEEE J. Solid-State Circuits, vol. 48, no. 11, pp. 2662–2670, Nov. 2013.

For high code rates, using the dual-log-MAP decoder increases the decoder throughput

DUAL-TRELLIS DECODING

Issues & contributions

- Dual-trellis construction:
  - Only for rate *k/(k+1)* available in the SoA (e.g. 4/5, 8/9 or 16/17)
  - The code rate might be *k/n*, where $n \neq (k+1)$

Propose a novel and generic way to construct the dual trellis

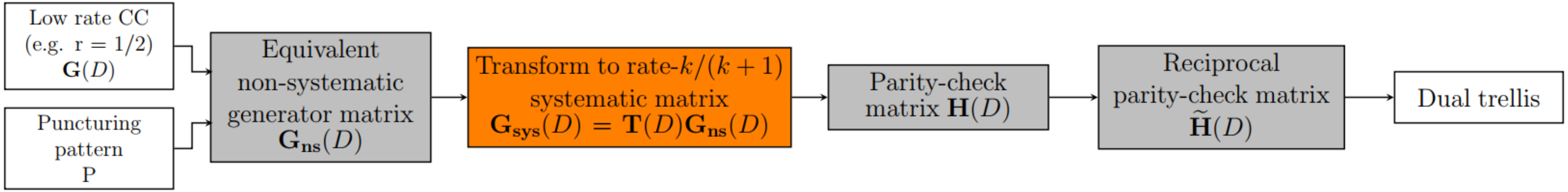
- Dual-Log-MAP decoding algorithm:
  - Produces *k* soft-outputs simultaneously for rate *k/n*
  - Employs a large number of LookUp Tables (LUTs)

Find a sub-optimal, low-complexity decoding algorithm

DUAL-TRELLIS DECODING

Proposed dual-trellis construction method

- The dual-trellis is generated by the *reciprocal parity-check matrix*
- For rate *k/(k+1)* punctured convolutional codes [1]:



Low rate CC (e.g. r $= 1/2$) $\mathbf{G}(D)$

Puncturing pattern P

Equivalent non-systematic generator matrix $\mathbf{G_{ns}}(D)$

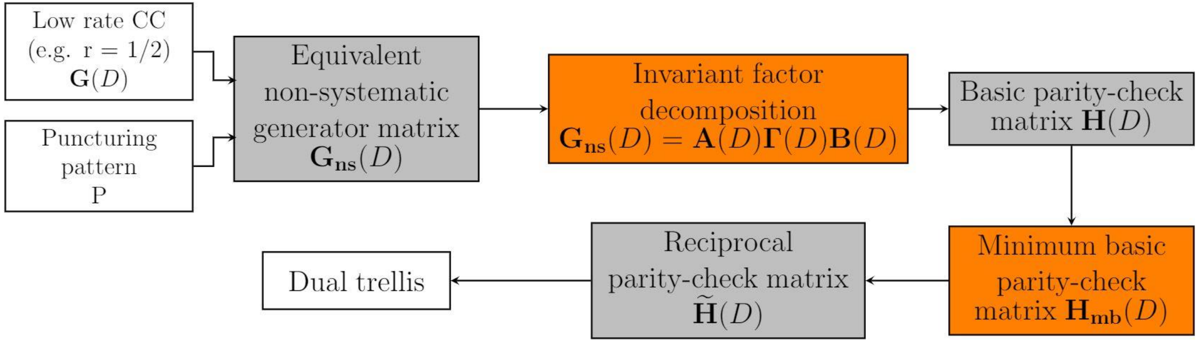
Transform to rate-$k/(k+1)$ systematic matrix $\mathbf{G_{sys}}(D) = \mathbf{T}(D)\mathbf{G_{ns}}(D)$

Parity-check matrix $\mathbf{H}(D)$

Reciprocal parity-check matrix $\widetilde{\mathbf{H}}(D)$

Dual trellis

- Our proposal: using the invariant factor decomposition proposed by Forney [2]:



Low rate CC (e.g. r $= 1/2$) $\mathbf{G}(D)$

Puncturing pattern P

Equivalent non-systematic generator matrix $\mathbf{G_{ns}}(D)$

Invariant factor decomposition $\mathbf{G_{ns}}(D) = \mathbf{A}(D)\mathbf{\Gamma}(D)\mathbf{B}(D)$

Basic parity-check matrix $\mathbf{H}(D)$

Minimum basic parity-check matrix $\mathbf{H_{mb}}(D)$

Reciprocal parity-check matrix $\widetilde{\mathbf{H}}(D)$

Dual trellis

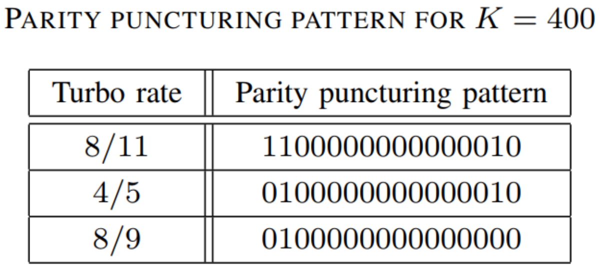
[1] A. Graell i Amat, G. Montorsi, and S. Benedetto, "Design and decoding of optimal high-rate convolutional codes," IEEE Trans. Inf. Theory, vol. 50, no. 5, pp. 867–881, May 2004.

[2] G. Forney, "Convolutional codes I: Algebraic structure," IEEE Trans. Inf. Theory, vol. 16, no. 6, pp. 720–738, November 1970.

DUAL-TRELLIS DECODING

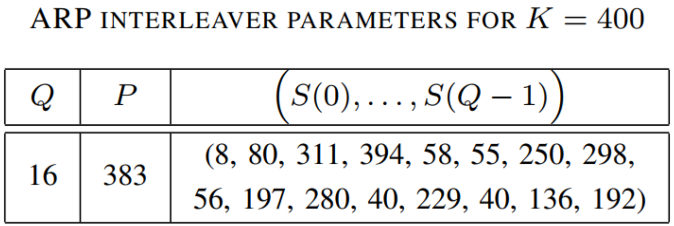
Validate the dual-trellis construction method

- Turbo code: LTE constituent RSC code, 6 decoding iterations
- K = 400, BPSK, AWGN, different code rates



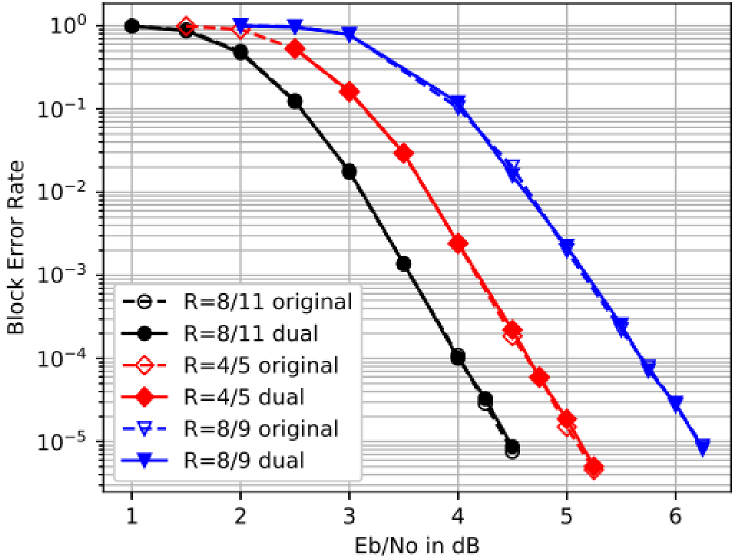
PARITY PUNCTURING PATTERN FOR $K = 400$

Turbo rate	Parity puncturing pattern
8/11	1100000000000010
4/5	0100000000000010
8/9	0100000000000000



ARP INTERLEAVER PARAMETERS FOR $K = 400$

Q	P	$(S(0), \ldots, S(Q-1))$
16	383	(8, 80, 311, 394, 58, 55, 250, 298, 56, 197, 280, 40, 229, 40, 136, 192)

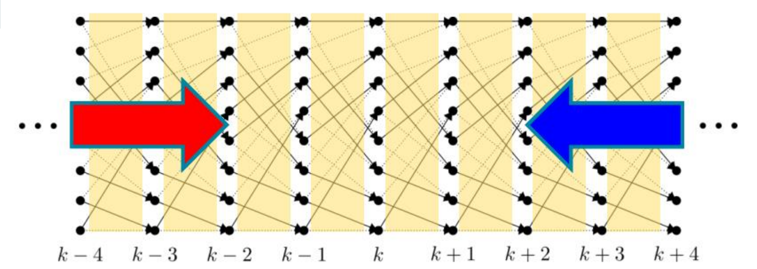


Same performance → confirm the validity of the dual-trellis construction method

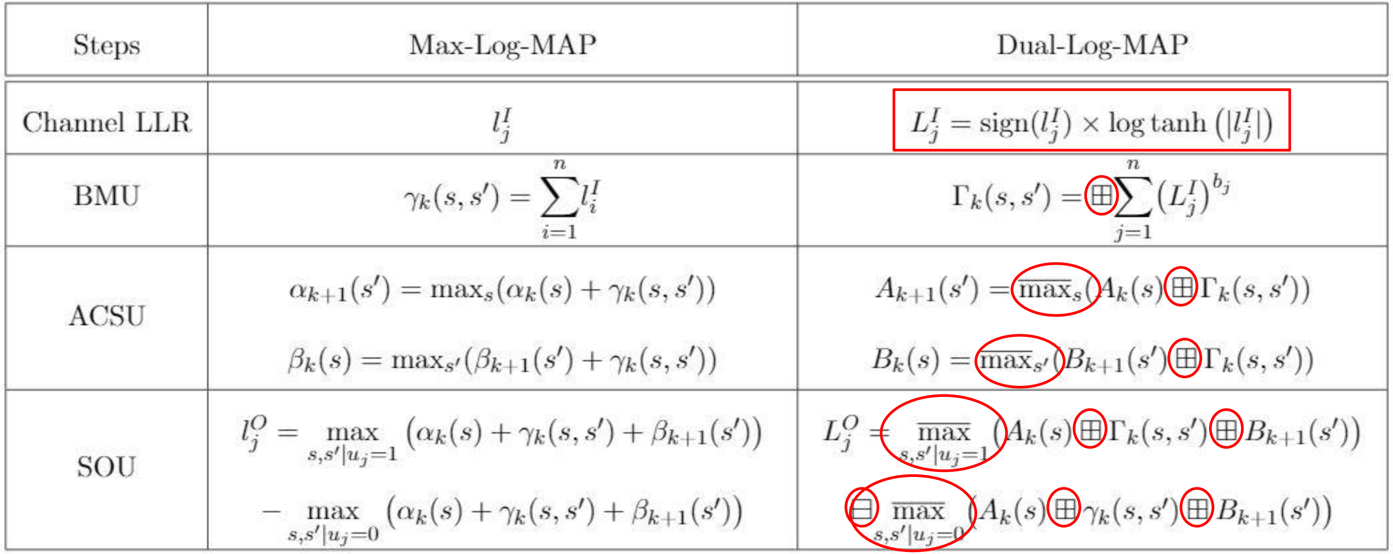
DUAL-TRELLIS DECODING

The dual-Log-MAP decoder

- Basic architecture
  - Branch Metric Unit
  - Add-Compare-Select Unit
  - Soft-Output Unit



- Calculation steps



Steps	Max-Log-MAP	Dual-Log-MAP
Channel LLR	l_j^I	$L_j^I = \text{sign}(l_j^I) \times \log\tanh\left(\lvert l_j^I \rvert\right)$
BMU	$\gamma_k(s,s') = \sum_{i=1}^{n} l_i^I$	$\Gamma_k(s,s') = \boxplus\sum_{j=1}^{n} (L_j^I)^{b_j}$
ACSU	$\alpha_{k+1}(s') = \max_s(\alpha_k(s) + \gamma_k(s,s'))$ $\beta_k(s) = \max_{s'}(\beta_{k+1}(s') + \gamma_k(s,s'))$	$A_{k+1}(s') = \overline{\max}_s(A_k(s) \boxplus \Gamma_k(s,s'))$ $B_k(s) = \overline{\max}_{s'}(B_{k+1}(s') \boxplus \Gamma_k(s,s'))$
SOU	$l_j^O = \max\limits_{s,s'\mid u_j=1}(\alpha_k(s) + \gamma_k(s,s') + \beta_{k+1}(s'))$ $- \max\limits_{s,s'\mid u_j=0}(\alpha_k(s) + \gamma_k(s,s') + \beta_{k+1}(s'))$	$L_j^O = \overline{\max\limits_{s,s'\mid u_j=1}}(A_k(s) \boxplus \Gamma_k(s,s') \boxplus B_{k+1}(s'))$ $\boxminus \overline{\max\limits_{s,s'\mid u_j=0}}(A_k(s) \boxplus \gamma_k(s,s') \boxplus B_{k+1}(s'))$

$$X \boxplus Y = \text{sign}(X)\text{sign}(Y) \times (X+Y)$$
$$X \boxminus Y = \text{sign}(X)\text{sign}(Y) \times (X-Y)$$

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

DUAL-TRELLIS DECODING

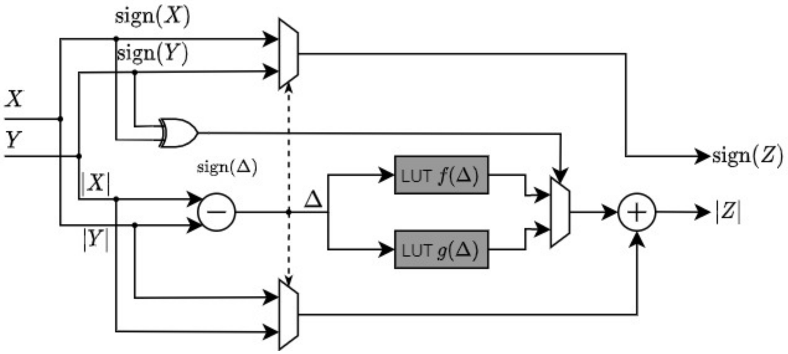
The $\overline{max}$ function

$$\overline{\max}(X,Y)=\begin{cases}\operatorname{sign}(X)\times\left(|X|+\log(1+\operatorname{sign}(X)\operatorname{sign}(Y)e^{|Y|-|X|})\right), & \text{if } |X|>|Y|\\ \operatorname{sign}(Y)\times\left(|Y|+\log(1+\operatorname{sign}(X)\operatorname{sign}(Y)e^{|X|-|Y|})\right), & \text{otherwise.}\end{cases}$$

- Depends on $\operatorname{sign}(X)\operatorname{sign}(Y)$, we have $\begin{cases}f(\Delta)=\log(1+e^{-\Delta}), & \text{if } \operatorname{sign}(X)\neq\operatorname{sign}(Y),\\ g(\Delta)=\log(1-e^{-\Delta}), & \text{if } \operatorname{sign}(X)=\operatorname{sign}(Y),\end{cases}$

  where $\Delta=\big||X|-|Y|\big|$

- Architecture of $\overline{max}$: two LUTs



- The SOU employs extensively $\overline{max}$:
  - Producing k soft-output in parallel
  - 2^{n-k} $\overline{max}$ per soft-output

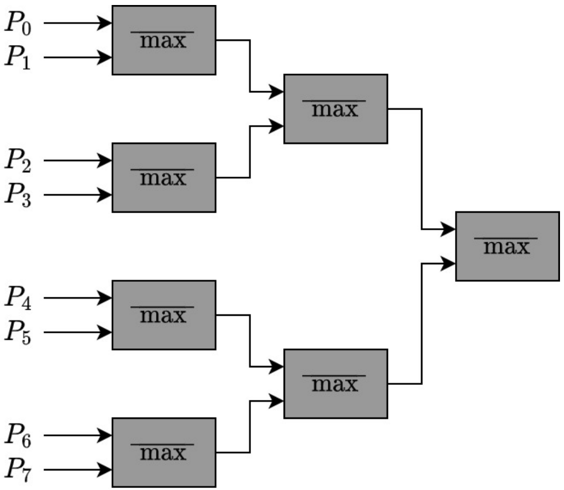
Complexity due to number of LUTs increasing with the code rate

DUAL-TRELLIS DECODING

Proposed simplification

- Given $\mathbf{P} = \{P_0, P_1, \ldots, P_{N-1}\}$ a set of N real numbers

$\overline{\max}(P_0, P_1, \ldots, P_{N-1})$



$\mathbf{S}^+$: Set of postive numbers $\in \mathbf{P}$

$\mathbf{S}^-$: Set of negative numbers $\in \mathbf{P}$

$$\overline{\max}\Big(\overline{\max}(P_i)\big|_{i\in\mathbf{S}^+}, \overline{\max}(P_j)\big|_{j\in\mathbf{S}^-}\Big)$$

$$\Downarrow$$

$$f(\Delta) = \log(1 + e^{-\Delta}) \approx 0$$

$$\Downarrow$$

$$\overline{\max}\Big(\max(|P_i|)\big|_{i\in\mathbf{S}^+}, -\max(|P_j|)\big|_{j\in\mathbf{S}^-}\Big)$$

$$\Downarrow$$

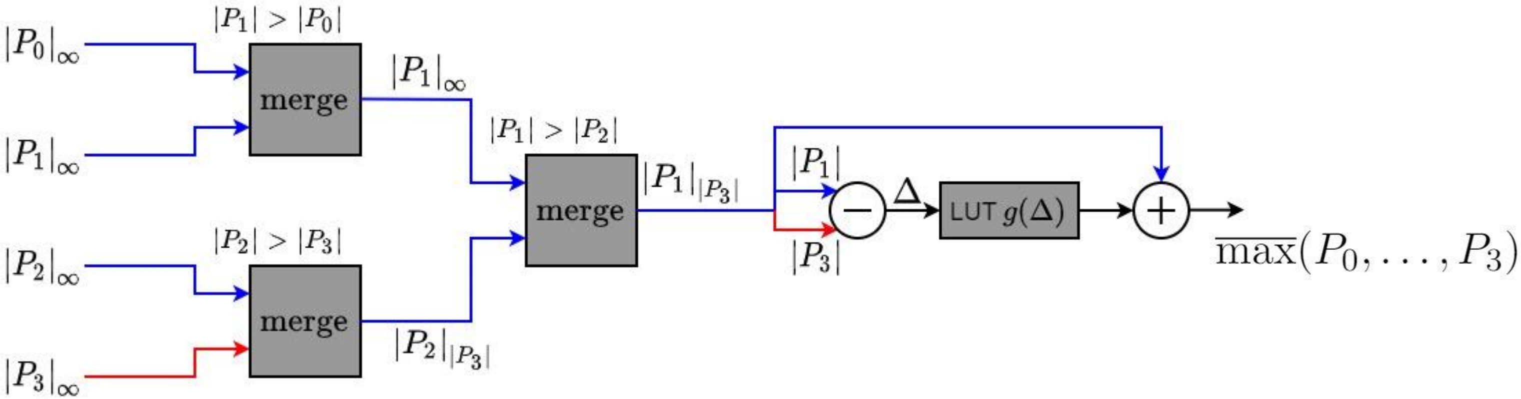
$$\boxed{\max(|P_k|)\big|_{k\in\mathbf{P}} + g\Big(\text{abs}\big(\max(|P_i|)\big|_{i\in\mathbf{S}^+} - \max(|P_j|)\big|_{j\in\mathbf{S}^-}\big)\Big)}$$

Proposal: Finding $\max(|P_k|)|_{k\in\mathbf{P}}$ and $\text{abs}\big(\max(|P_i|)\big|_{i\in\mathbf{S}^+} - \max(|P_j|)\big|_{j\in\mathbf{S}^-}\big)$

DUAL-TRELLIS DECODING

Applying the Local-SOVA

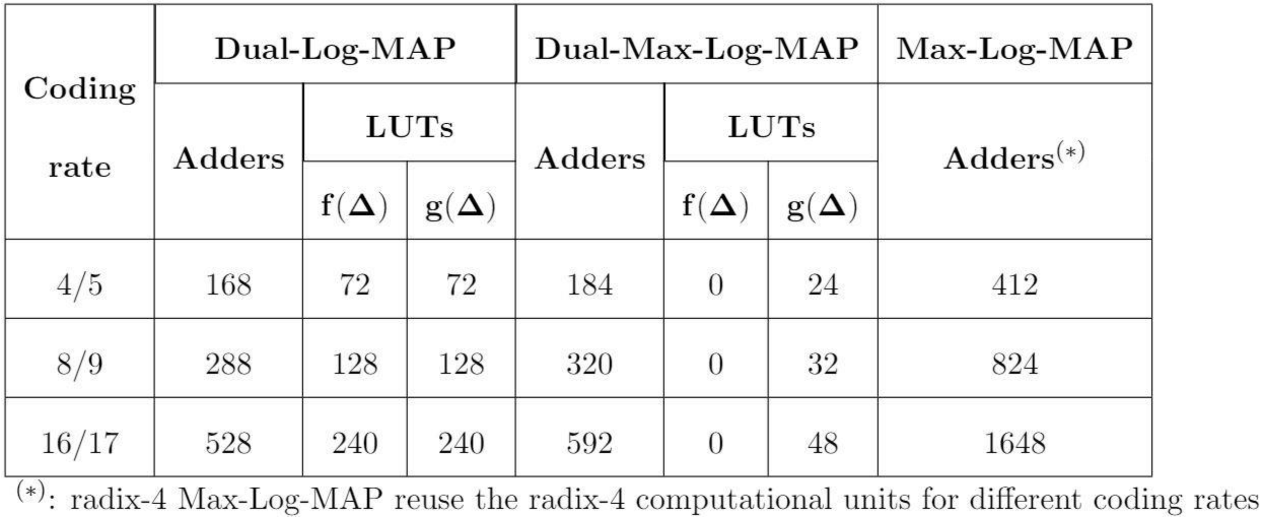
- Each path: 1 **main-value**, 1 **sign**, 1 **sub-value**
- **Merge operation** between two paths:
  - **Main-value** & **sign**: select the higher main-value and sign *(the winner path)*
  - **Sub-value**:
    1. If same sign: select min between two sub-values
    2. If different sign: select min between sub-value of the winner path and the main value of the other path



DUAL-TRELLIS DECODING

Computational complexity analysis

- Compare dual-log-MAP and dual-max-log-MAP for 8-state convolutional codes



Coding rate	Dual-Log-MAP			Dual-Max-Log-MAP			Max-Log-MAP
	Adders	LUTs		Adders	LUTs		Adders$^{(*)}$
		$f(\Delta)$	$g(\Delta)$		$f(\Delta)$	$g(\Delta)$	
4/5	168	72	72	184	0	24	412
8/9	288	128	128	320	0	32	824
16/17	528	240	240	592	0	48	1648

$^{(*)}$: radix-4 Max-Log-MAP reuse the radix-4 computational units for different coding rates

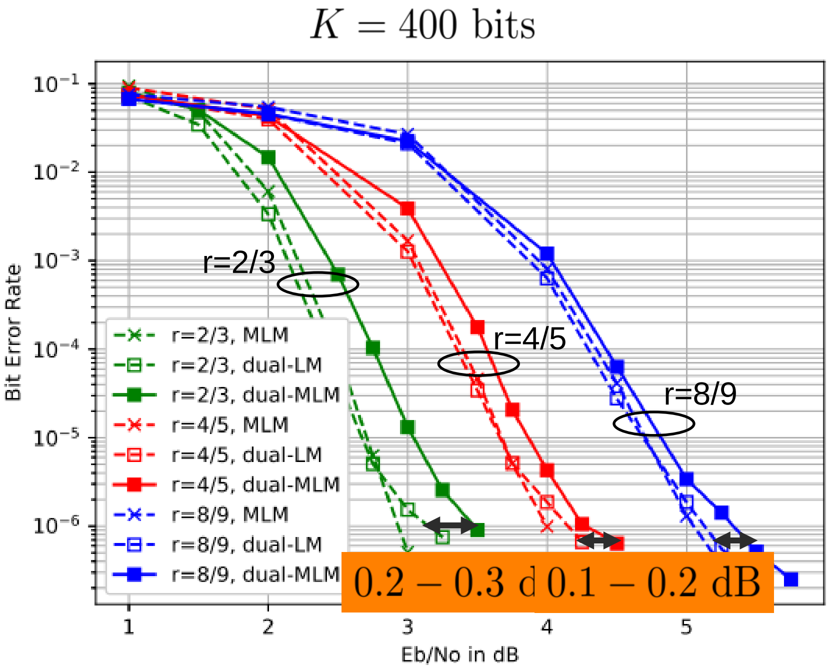
Drastically reduce the number of required LUTs

Inherit the high-throughput for high code rates of the dual-log-MAP

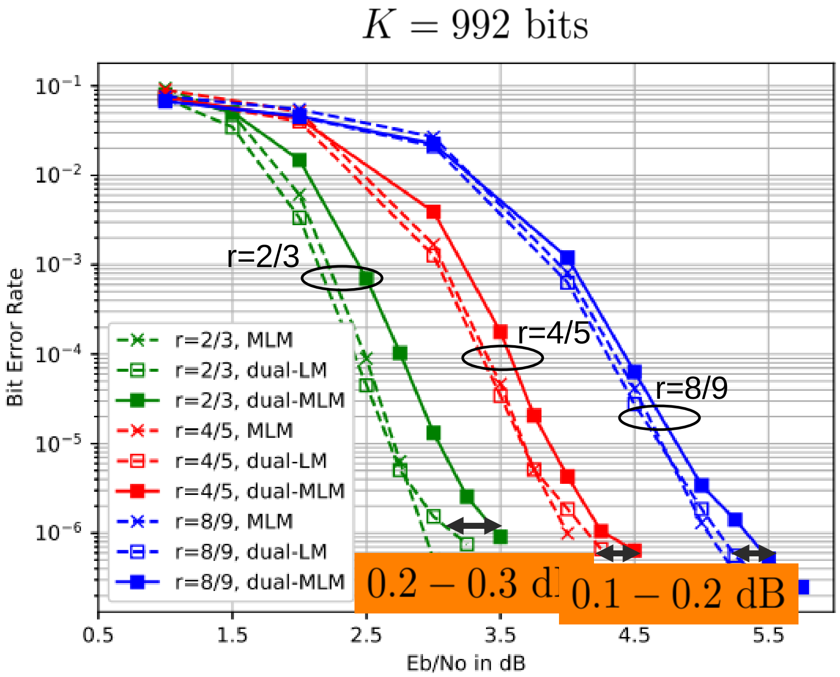
DUAL-TRELLIS DECODING

Performance of the dual-Max-Log-MAP

- LTE turbo codes: BPSK, AWGN channel, 8 iterations, different code rates



$0.2 - 0.3$ d $0.1 - 0.2$ dB



$0.2 - 0.3$ d $0.1 - 0.2$ dB

- **For high coding rates** $(4/5, 8/9)$: **loss** $0.1 - 0.2$ **dB**

DUAL-TRELLIS DECODING

Conclusion

- New dual trellis construction method:
  - General way for arbitrary coding rate *k/n*

- Sub-optimal, low-complexity dual-trellis decoding algorithm: dual-Max-Log-MAP
  - Same throughput as dual-Log-MAP: throughput increase with coding rates
  - Lower the use of the lookup tables down to only 10%
  - Small degradation of 0.1 dB in performance in high coding rate target (4/5, 8/9).
  - Hardware implementation will be investigated to complete the study

CONCLUSION & FUTURE WORKS

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Conclusion

- Very high throughput turbo decoder in EPIC framework
  - Innovative architecture UXMAP: up to 400 Gbps
  - Local-SOVA: novel decoding algorithm
  - Low complexity compared to the Max-Log-MAP with same performance
  - Save 33% area for radix 4, lower latency options with radix 8 and radix 16

- Dual-trellis decoding
  - Decoding throughput increases with coding rate
  - Generalization of the construction method for the dual trellis
  - Novel low-complexity decoding algorithm: dual-Max-Log-MAP

CONCLUSION

Future works

- Local-SOVA in UXMAP architecture
  - A full decoder implementation is necessary for radix 4, radix 8 and radix 16
  - Numerous trade-offs between complexity and performance can be exploited
  - Radix 32 and radix 64 can be investigated for very low latency requirements.

- Local-SOVA in other architectures
  - Consider replacing the Max-Log-MAP in other turbo decoder architectures
  - Use in soft-decoder for convolutional codes with high number of states

- Dual-trellis decoding
  - Implementation of the dual-Max-Log-MAP

LIST OF PUBLICATIONS & PRESENTATIONS

[1] V. H. S. Le, C. A. Nour, E. Boutillon, and C. Douillard, “Revisiting the Max-Log-Map algorithm with SOVA update rules: new simplifications for high-radix SISO decoders,” IEEE Trans. Comm., vol. 68, no. 4, pp. 1991-2004, 2020.

[2] V. H. S. Le, C. A. Nour, E. Boutillon, and C. Douillard, “Dual trellis construction for high-rate punctured convolutional codes,” in IEEE PIMRC Workshops, Istanbul, Turkey, Sept. 2019, pp. 1-7.

[3] V. H. S. Le, C. A. Nour, E. Boutillon, and C. Douillard, “A low-complexity dual trellis decoding algorithm for high-rate convolutional codes,” in IEEE Wireless Communications and Networking Conference (WCNC), Seoul, South Korea, May 2020.

[4] V. H. S. Le, “Dual trellis construction for high-rate punctured convolutional codes (invited talk),” in GdR ISIS Workshop: Enabling Technologies for sub-TeraHertz and TeraHertz communications, Maisons-Alfort, France, Sept. 2019.

[5] V. H. S. Le, “Local-SOVA: Revisiting the Max-Log MAP algorithm with modified SOVA update rules (invited talk),” in EPIC project meeting, Brest, France, Oct. 2019.

THANK YOU FOR LISTENING

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom