Welcome to

DesignCon® 2023
WHERE THE CHIP MEETS THE BOARD

Conference
January 31 – February 2, 2023

Expo
February 1 – 2, 2023

Santa Clara Convention Center

Statistical BER Analysis of Concatenated FEC in Multi-Part Links

Richard Barrie, (University of Toronto)
Ming Yang (University of Toronto/Alphawave SEMI)

Anthony Chan Carusone (University of Toronto/Alphawave SEMI)

SPEAKERS

Richard Barrie

M.A.Sc. Candidate, University of Toronto

richard.barrie@isl.utoronto.ca | https://isl.utoronto.ca/

Richard Barrie received his B.A.Sc. degree in robotics engineering from the University of Toronto in 2022. He is currently a M.A.Sc. candidate in the University of Toronto's Department of Electrical and Computer Engineering. His research interests are in system modeling and design for high-speed communications.

Ming Yang

Ph.D. Candidate, University of Toronto

ming.yang@isl.utoronto.ca | https://isl.utoronto.ca/

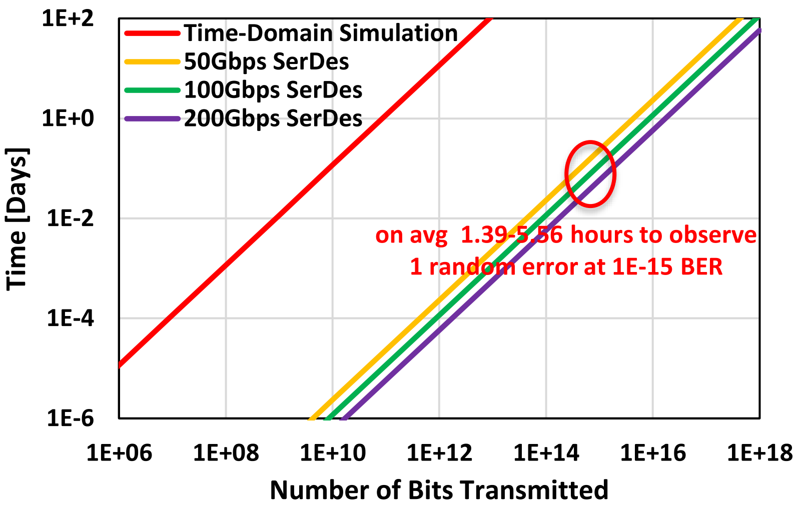
Ming Yang received the B.Eng. degree in aerodynamic engineering from the Department of Aeronautics, Xiamen University, Xiamen, China, in 2012, and the B.Eng. and M.Eng. degree in electrical engineering from the Department of Electrical and Computer Engineering, McGill University, Montreal, Canada, in 2013 and 2016, respectively. He is currently a Ph.D. candidate in the Edward S. Rogers Sr. Department of Electrical & Computer Engineering at University of Toronto. He is the recipient of the Alexander Graham Bell Canada Graduate Scholarships award (NSERC CGS-D). His research interests are in analog integrated circuit design, on-chip analog signal processing and high-performance integrated circuit testing.

Outline

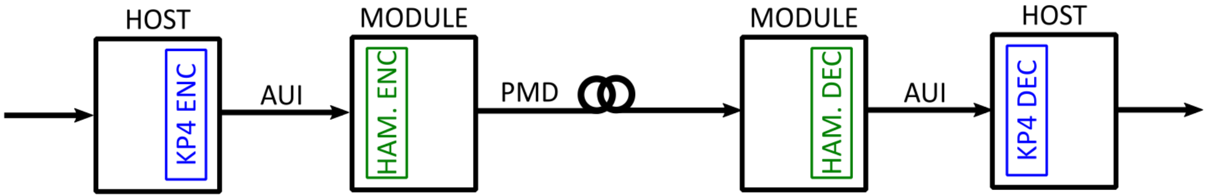
1. Motivation
2. Statistical Analysis of End-to-End RS FEC [1-2]
3. Statistical Analysis of Concatenated FEC
4. Modeling Inner-FEC Interleaving in Concatenated FEC
5. Conclusion

Motivation – Statistical BER Analysis

- We want to confirm post-FEC BERs in simulation down to 10^{-15} – 10^{-21} quickly and accurately
- Verifying the post-FEC BERs at these levels using time-domain simulations can become prohibitively long
- Instead, statistical analysis can be used. To be accurate, the method must capture the statistics of errors
- Bit or symbol error occurrences are correlated; they sometimes occur in bursts due to DFE error propagation, low-frequency clock jitter, supply noise, etc.
- Error statistics strongly affect the performance of FEC



Motivation – Concatenated FEC

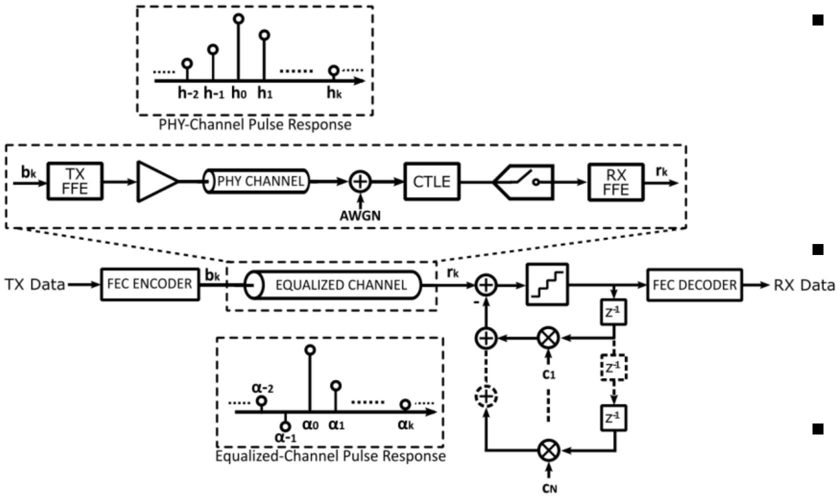


- Concatenated FEC is a popular candidate FEC architecture for 200 Gb/s wireline links
- AUI (Attachment Unit Interface) is a MR or VSR link with a DFE that may introduce burst errors
- PMD (Physical Media Dependent Layer) can be an optical link dominated by random errors
- For 200Gb/s applications, BER in AUI ≤ 10^{-5} and BER in PMD ≤ 10^{-3} [3-4]
- A stronger concatenated FEC code is needed for raw BER at 10^{-3} level!
- No statistical model available for modeling concatenated FEC

Outline

1. Motivation
2. Statistical Analysis of End-to-End RS FEC [1-2]
    a. System Overview
    b. PAM-Symbol Trellis Model
    c. Trellis Dynamic Programming
    d. Time-Aggregation Generating FEC-Symbol Trellis
    e. Post-FEC BER Calculation
    f. Applications
3. Statistical Analysis of Concatenated FEC
4. Modeling Inner-FEC Interleaving in Concatenated FEC
5. Conclusion

Transceiver Model – System Overview

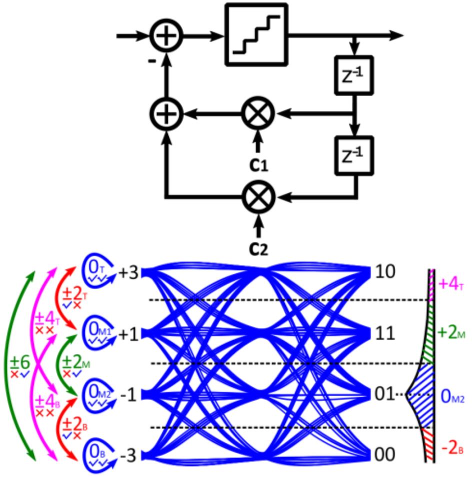


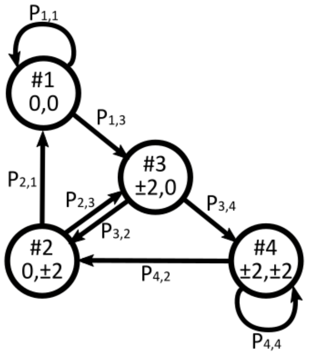
- Equalized pulse response α(z) is generated by convolving the physical channel's pulse with the impulse response of other components in the link, such as the TX FFE, TX driver, CTLE and RX FFE
- Additive white Gaussian noise (AWGN) assumed at CTLE input, creating correlated noise samples after CTLE filtering
- End-to-end RS KP4 (544,514,15) FEC encodes and decodes bit streams in $GF(2^{10})$

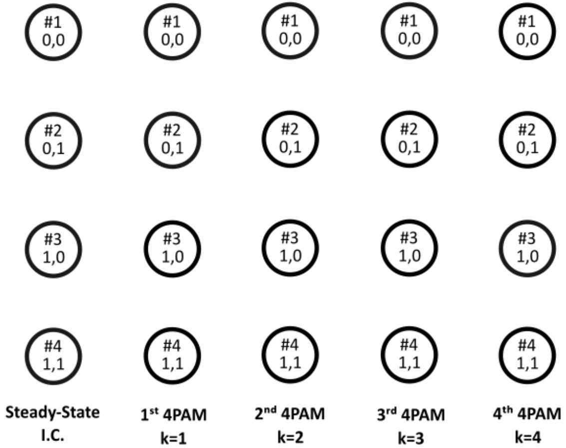
Statistical Model – DFE Error Propagation
[Yang, TCAS-I, 2020]

$Pr_k^j(i)$

≡ probability of all trellis paths visiting state i at the k^{th} stage of the trellis having exactly j bit errors



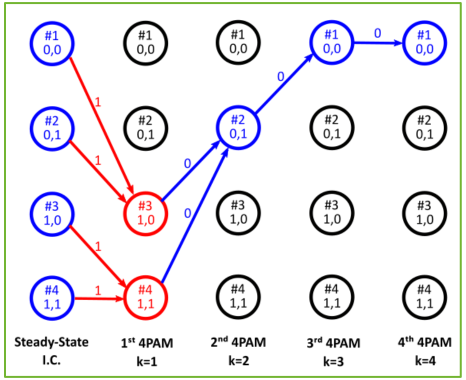




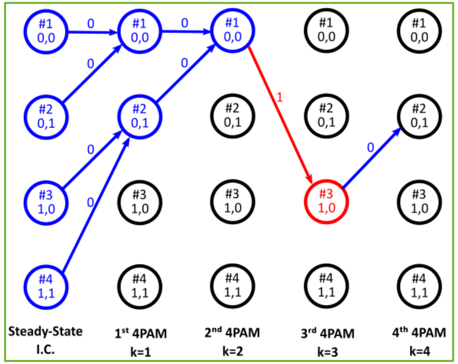
- Example of a 2-tap DFE represented by a simplified 4-state Markov model
- Time-unrolling the Markov DFE model to generate PAM-symbol trellis
- Apply trellis dynamic programming to the PAM-symbol trellis to efficiently collect all error patterns

Probability Model – Finding Pre-FEC BER

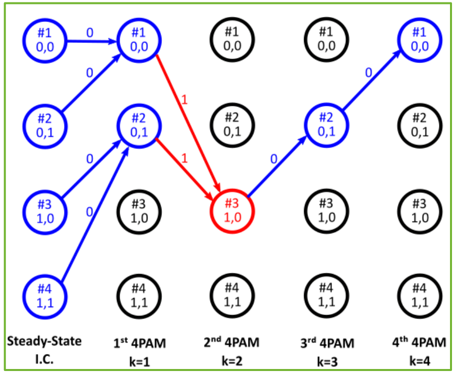
- **Example: 2-tap DFE, 8-bit codeword, 4-PAM, *j* = 1**



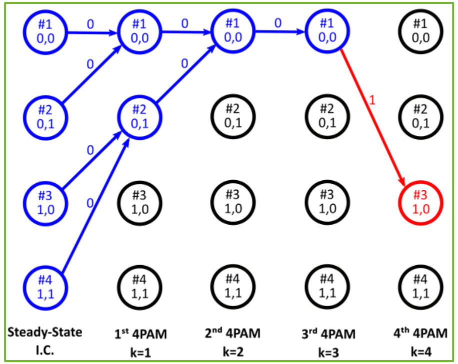
- **Case 1: error at 1st stage**
- $Pr_4^1(1) = \pi_1 p_{13} p_{32} p_{21} p_{11}$
  $+\pi_2 p_{23} p_{32} p_{21} p_{11}$
  $+\pi_3 p_{34} p_{42} p_{21} p_{11}$
  $+\pi_4 p_{44} p_{42} p_{21} p_{11}$



- **Case 3: error at 3rd stage**
- $Pr_4^1(2) = \pi_1 p_{11} p_{11} p_{13} p_{32}$
  $+\pi_2 p_{21} p_{11} p_{13} p_{32}$
  $+\pi_3 p_{32} p_{21} p_{13} p_{32}$
  $+\pi_4 p_{42} p_{21} p_{13} p_{32}$



- **Case 2: error at 2nd stage**
- $Pr_4^1(1) = \pi_1 p_{11} p_{13} p_{32} p_{21}$
  $+\pi_2 p_{21} p_{13} p_{32} p_{21}$
  $+\pi_3 p_{32} p_{23} p_{32} p_{21}$
  $+\pi_4 p_{42} p_{23} p_{32} p_{21}$



- **Case 4: error at 4th stage**
- $Pr_4^1(3) = \pi_1 p_{11} p_{11} p_{11} p_{13}$
  $+\pi_2 p_{21} p_{11} p_{11} p_{13}$
  $+\pi_3 p_{32} p_{21} p_{11} p_{13}$
  $+\pi_4 p_{42} p_{21} p_{11} p_{13}$

Inefficiency of Exhaustive Computations

$\sum_i Pr_4^1(i)$=

Case 1: $\pi_1 p_{13} p_{32} p_{21} p_{11} + \pi_2 p_{23} p_{32} p_{21} p_{11} + \pi_3 p_{34} p_{42} p_{21} p_{11} + \pi_4 p_{44} p_{42} p_{21} p_{11} +$

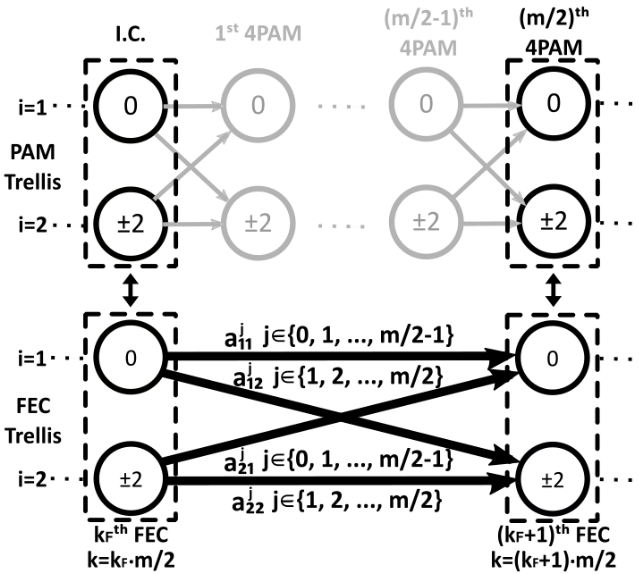
Case 2: $\pi_1 p_{11} p_{13} p_{32} p_{21} + \pi_2 p_{21} p_{13} p_{32} p_{21} + \pi_3 p_{32} p_{23} p_{32} p_{21} + \pi_4 p_{42} p_{23} p_{32} p_{21} +$

Case 3: $\pi_1 p_{11} p_{11} p_{13} p_{32} + \pi_2 p_{21} p_{11} p_{13} p_{32} + \pi_3 p_{32} p_{21} p_{13} p_{32} + \pi_4 p_{42} p_{21} p_{13} p_{32} +$

Case 4: $\pi_1 p_{11} p_{11} p_{11} p_{13} + \pi_2 p_{21} p_{11} p_{11} p_{13} + \pi_3 p_{32} p_{21} p_{11} p_{13} + \pi_4 p_{42} p_{21} p_{11} p_{13}$

- Assuming each erred 4-PAM symbol contain only 1 bit error, computations are required to repeat the analysis for $Pr_4^2(i)$, $Pr_4^3(i)$, $Pr_4^4(i)$
- Pre-FEC BER = $\sum_i (Pr_4^1(i) + 2Pr_4^2(i) + 3Pr_4^3(i) + 4Pr_4^4(i))/8$
- Not practical to enumerate all error patterns for a long codeword
- Some multiplications are performed twice
- Trellis dynamic programming systematically stores these intermediate results so that the same multiplication is only performed once

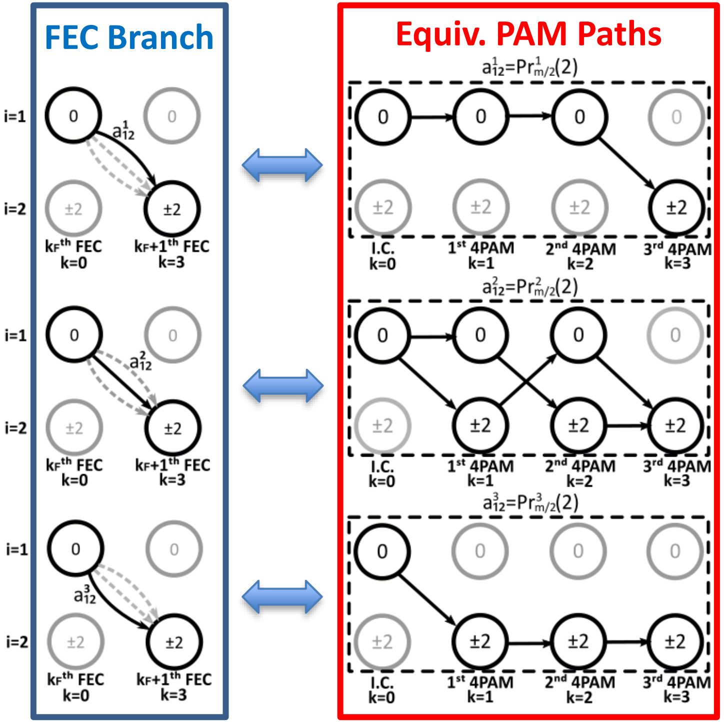
The "FEC-Symbol Trellis" [Yang, TCAS-I, 2020]



Example above: 1-tap DFE

- Construct a new trellis where each stage corresponds to an entire **FEC symbol** over GF(2^m) rather than a PAM symbol
  - "Time aggregation" of a Markov model
  - ✓ Much shorter "FEC-symbol trellis"
- Branch probabilities in the FEC-symbol trellis can be found by analysis of the short length-*m*/2 trellis above

Finding Branch Probabilities in the FEC Trellis



Example above: 1-tap DFE, $m = 6$
Thus, each FEC symbol is 3 4-PAM symbols

- The FEC-symbol trellis has a higher radix if we need to keep track of the number of pre-FEC bit errors

- Example:

$$a_{12}^{1} = \mathrm{Pr}_{m/2}^{1}$$

≡ probability of going from state 1 (no error in DFE) to state 2 (error in DFE) traversing a FEC symbol (duration 3 PAM-4 symbols in this case) experiencing exactly one bit error

Finding Post-FEC BER

- **We wish to find the BER at the output of a FEC decoder operating over GF(2^m), $m > 1$**
  - e.g. many of the standard wireline codes are Reed Solomon codes of this type

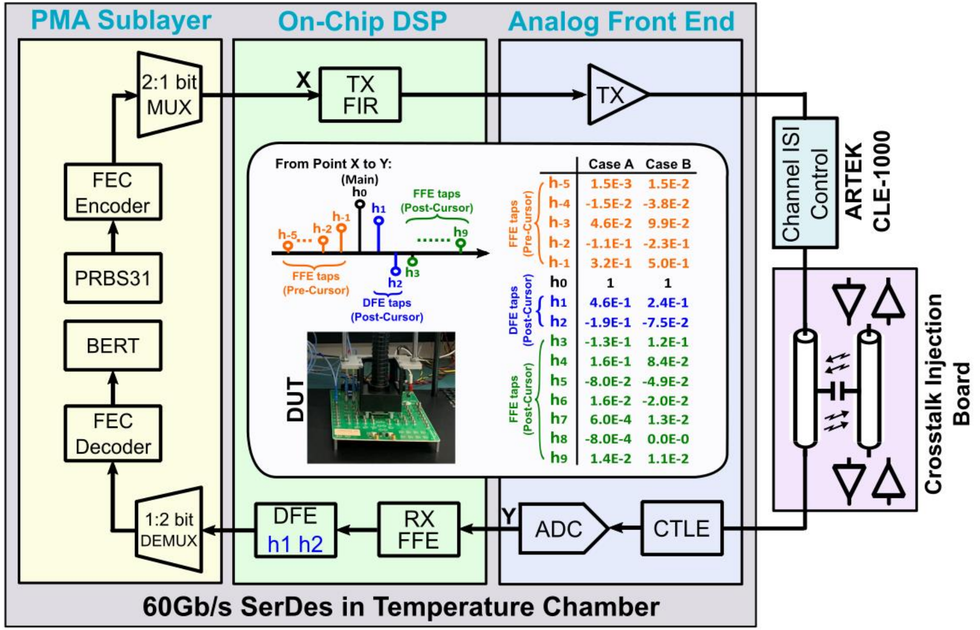
- **Example: RS(544, 514, 15) KP4 FEC over GF(2^{10})**
  - Each block is 5440 bits (544 FEC symbols) long
  - Can correct up to 15 FEC symbol errors

- **In FEC-symbol trellis, we use $Pr\_FEC_k^{j_s,j_b}(i)$ to track the probability of all trellis paths at the k^{th} stage having exactly j_s FEC-symbol errors and j_b bit errors**

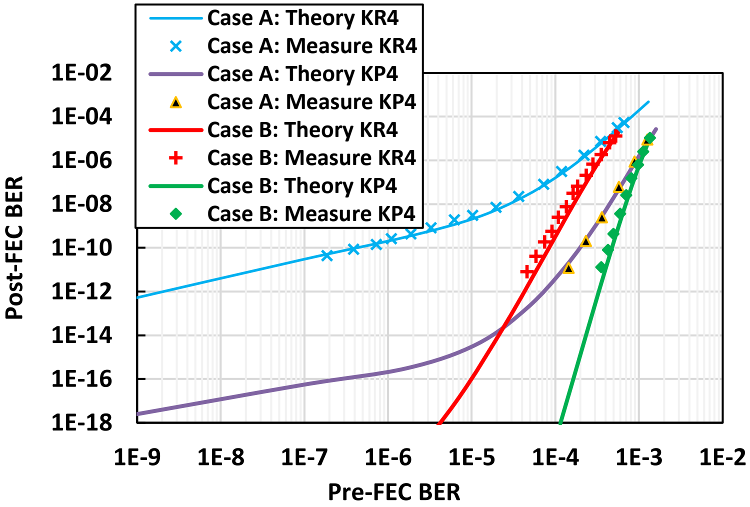
- **Then, we can calculate the post-FEC BER over a *n*-symbol codeword**

$$\text{Post} - \text{FEC BER} = \frac{1}{n \cdot m} \sum_{j_s=t+1}^{n} \sum_{j_b=j_s}^{j_s \cdot m/2} \left(j_b \cdot \sum_i Pr\_FEC_n^{j_s,j_b}(i)\right)$$

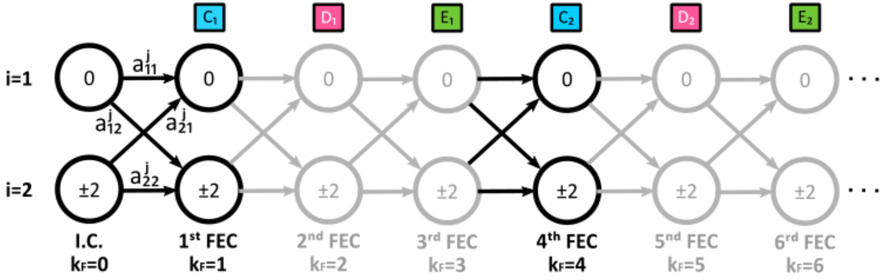
Statistical Model – DFE Error Propagation
[Yang, TCAS-I, 2020]



Measured and statistical pre-FEC vs post-FEC BER plot for RS(528, 514, 7) and RS(544, 514, 15) code

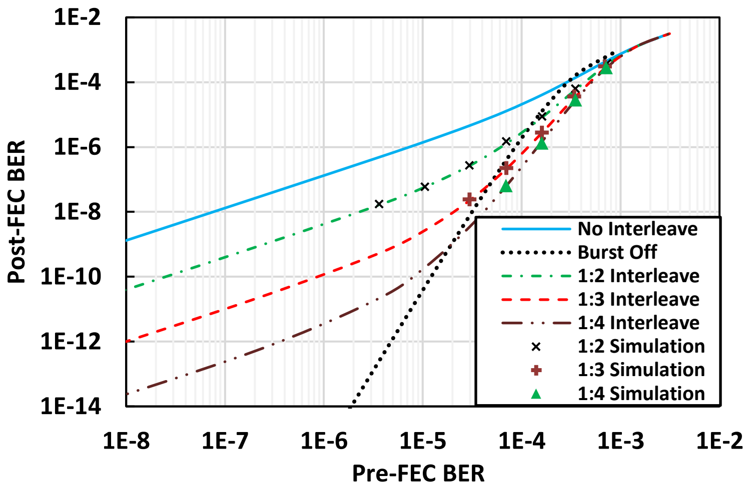


Statistical Analysis of Interleaved Codewords [Yang, DesignCon, 2020]



- **Analysis of a 1:3 interleaved code of length *n* requires analysis of a length-3*n* FEC-symbol trellis**

- **Results confirm the improved burst-error tolerance offered by interleaving**

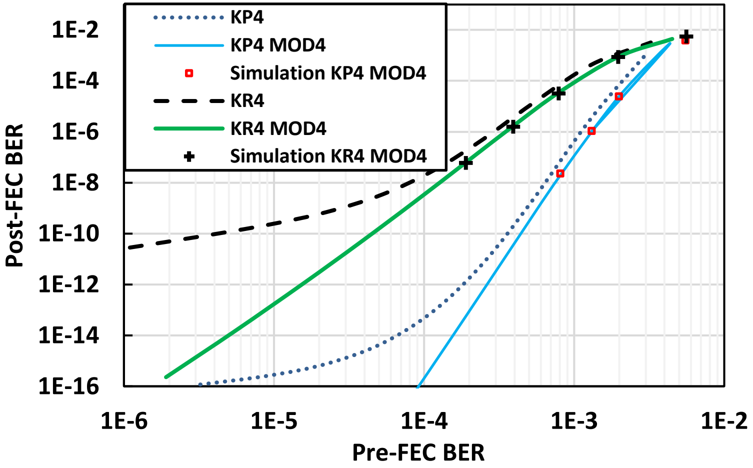


Pre-FEC vs post-FEC BER plot for interleaved RS(1000,992,4) codes with $h = 0.5 + 0.25z^{-1} - 0.25z^{-2}$.

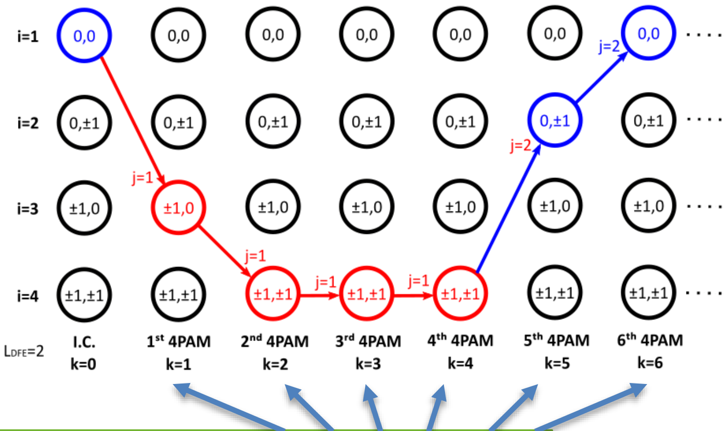
Statistical Analysis of 1/(1+D) Precoding

[Yang, DesignCon, 2020]

- **Statistical analysis method allows us to identify probability of all error patterns**
- **1/(1+D) precoding maps each error pattern to a different error patterns**



Example below corresponds to a 2-tap DFE; hence, 4-state PAM trellis

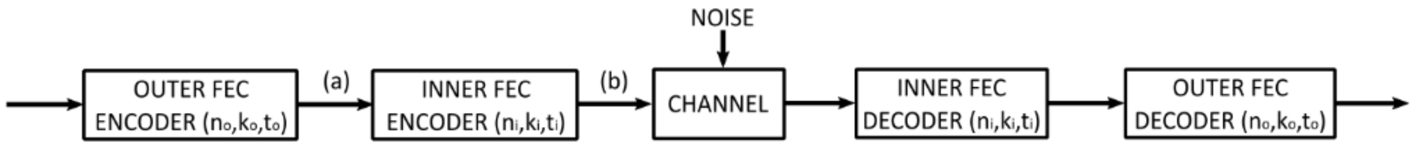


Precoder Input **t_k**	0	2	3	1	1	0	2
Precoder Output **b_k**	0	2	1	0	1	3	3
DFE Output **d_k**	0	3	0	1	0	3	3
Error Value **d_k-b_k**	0	1	-1	1	-1	0	0
Decoder Output **y_k**	0	3	3	1	1	3	2

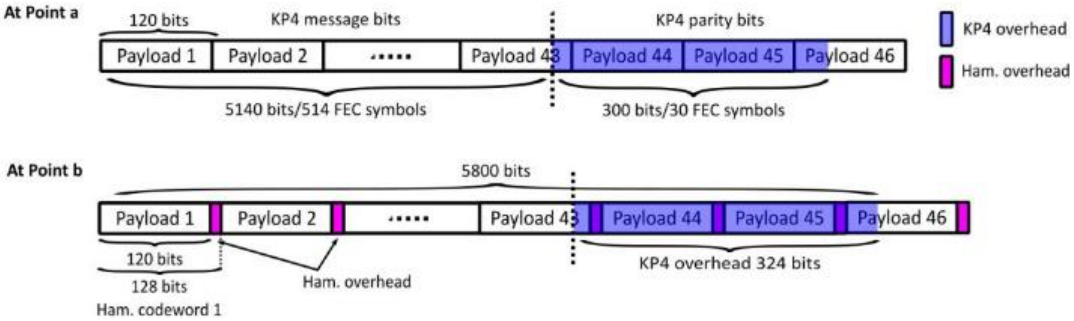
Outline

1. Motivation
2. Statistical Analysis of End-to-End RS FEC
3. Statistical Analysis of Concatenated FEC
   a. System Overview
   b. Trellis Model for Concatenated FEC Codes
   c. Modeling Decoding Errors (Miscorrections)
   d. Simulation Results
4. Modeling Inner-FEC Interleaving in Concatenated FEC
5. Conclusion

Concatenated FEC – System Overview



- **Outer code: strong non-binary linear block code**
  - RS-KP4 (544,514,15)
  - RS-KR4 (528,514,7)
- **Inner code: weaker binary linear block code**
  - Hamming (127,120,1)
  - Extended Hamming (128,120,1)
  - BCH (144,136,1)
- Concatenated FEC results in effectively multiplying inner and outer FEC coding gain



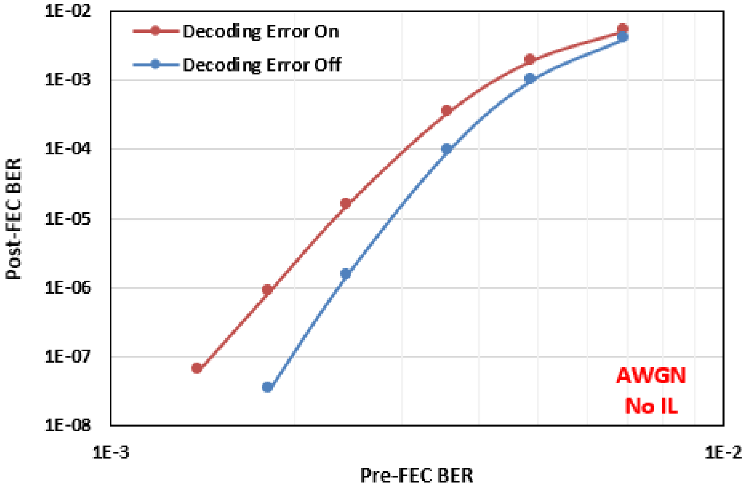
Encoding an outer codeword into inner codewords using a KP4 + Hamming (128,120,1) concatenated FEC

Inner Code Miscorrections

- With more than one error in an inner-FEC codeword that can correct 1 bit error, the inner-FEC decoder may decode a codeword incorrectly, introducing an additional bit error (miscorrection)

- This is a significant source of error for inner-FEC codes having a small Hamming distance

- The Hamming (127,120,1) code can be enhanced by adding one additional parity bit: the extended Hamming (128,120,1)



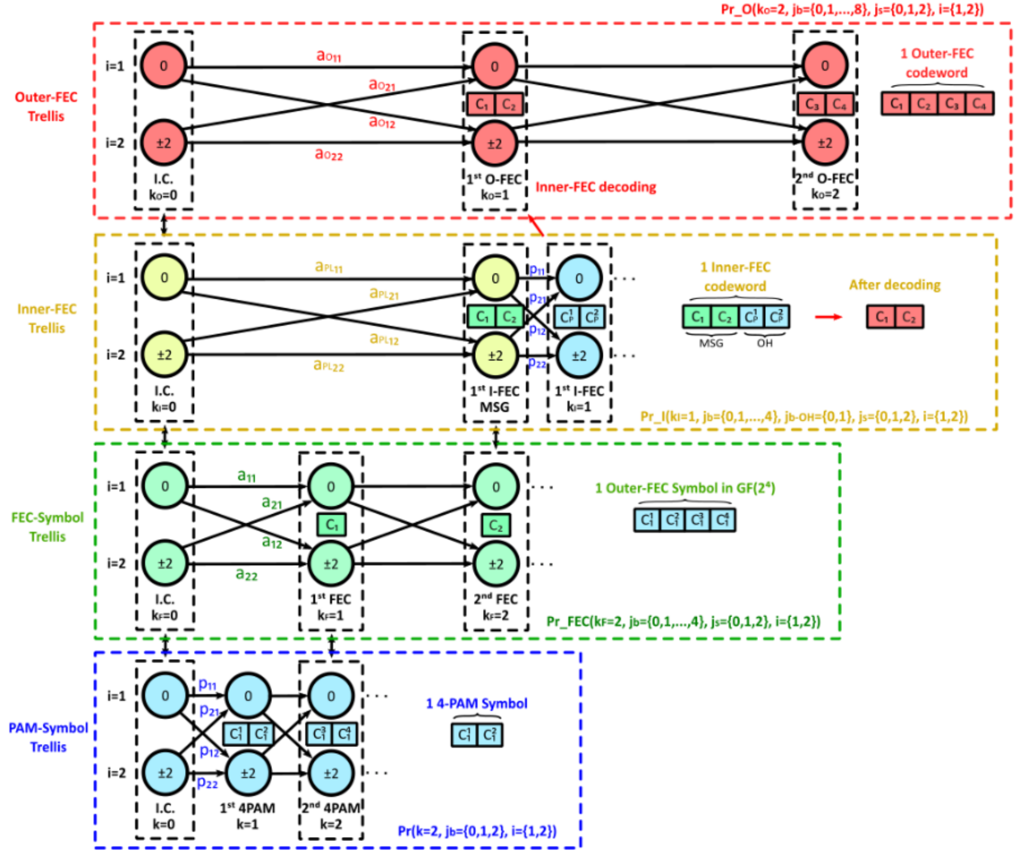
Pre-FEC vs. Post-FEC BER curve for KP4 + BCH (144,136,1) concatenated FEC with and without inner-FEC miscorrections

Concatenated FEC - Statistical Model

- Builds on trellis model for end-to-end FEC by adding two additional layers of abstraction to model inner-FEC and outer-FEC codeword

- Dynamic programming applied at 4 levels of time aggregation:
  - PAM-symbol trellis
  - FEC-symbol trellis
  - Inner-FEC trellis
  - Outer-FEC trellis

- In this example:
  - Inner binary FEC code: (10,8,1)
  - Outer non-binary FEC code in GF(2^4): (2,1,1)

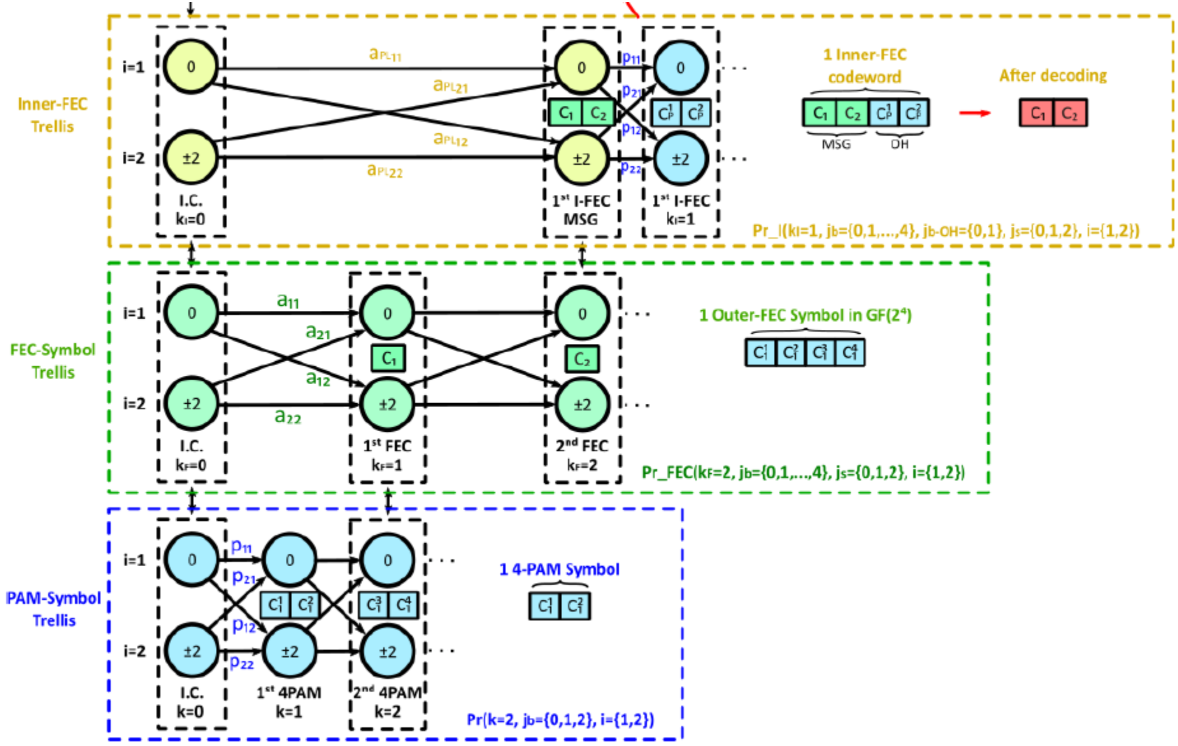


"Inner-FEC Trellis"

- PAM-symbol trellis and FEC-symbol Trellis remain the same as with end-to-end FEC

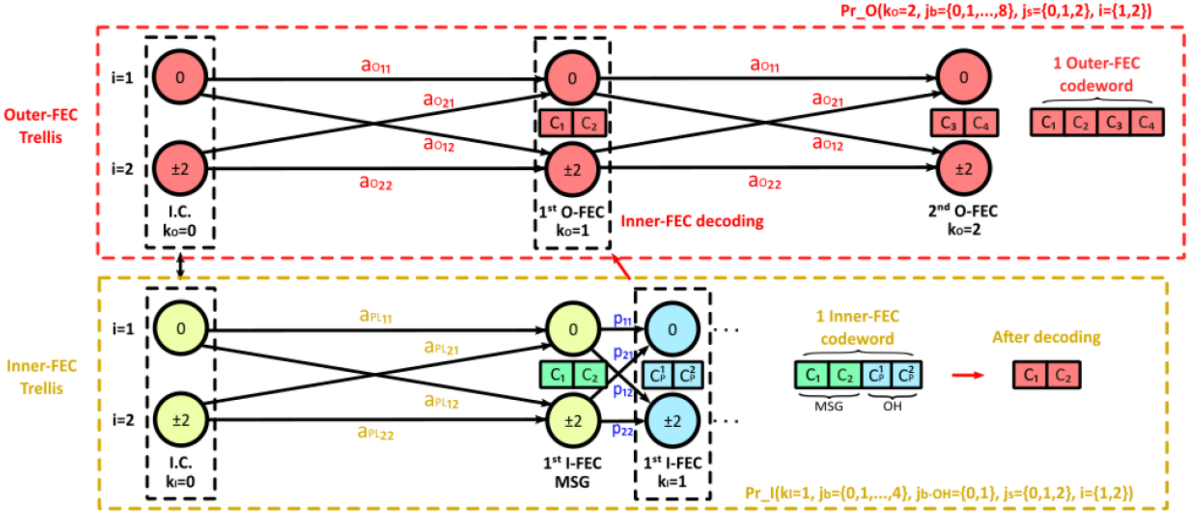
- **FEC symbols** are aggregated to find transition probability '$\boldsymbol{a_{PL}}$' for inner-FEC **payload**
  - Bit errors and FEC-symbol errors are tracked in the payload

- **PAM symbols** are aggregated to find transition probability '$\boldsymbol{a_{OH}}$' for inner-FEC **overhead**
  - Only bit errors are tracked in the overhead



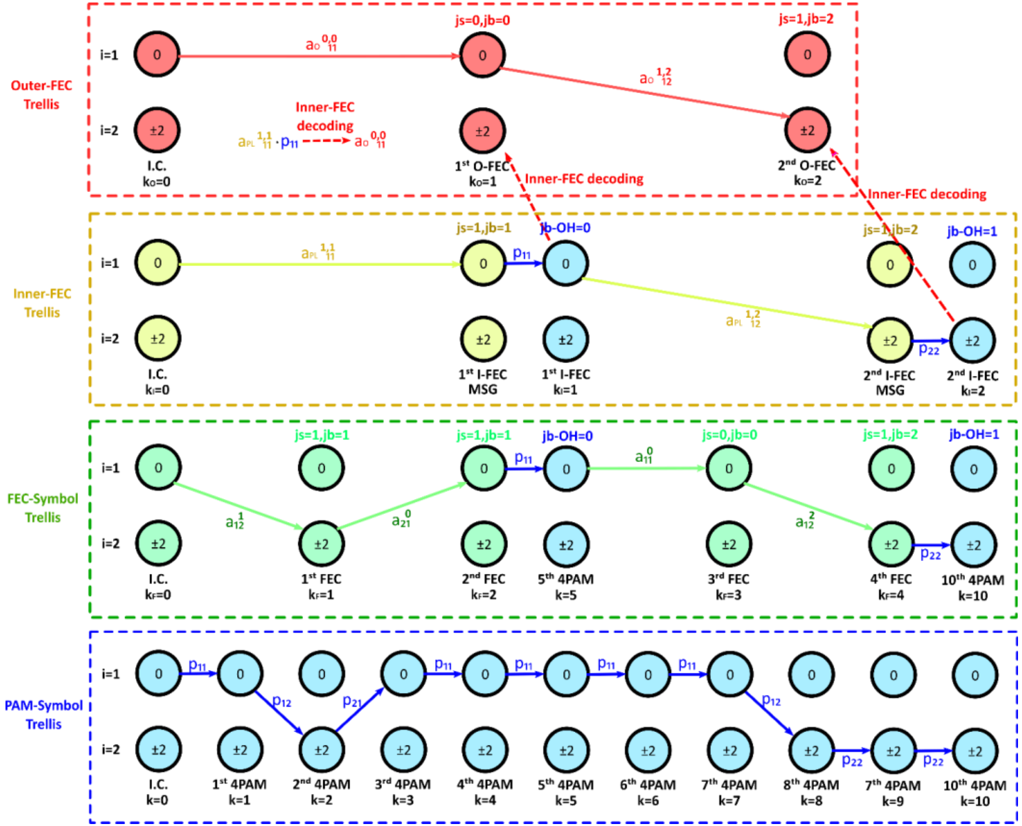
"Outer-FEC Trellis"

- Decoding is applied to each inner-FEC codeword
  - Correctable trellis paths are assigned 0 bit errors and 0 FEC-symbol errors
  - Uncorrectable trellis paths keep their bit errors and FEC-symbol errors
- After decoding, transition probabilities 'a_o' are used in the outer-FEC trellis to reach the end of an outer-FEC codeword
- Post-FEC BER is computed with the same technique used for the end-to-end FEC



Concatenated FEC Trellis Path Example

- **In this example:**
  - Inner binary FEC code: (10,8,1)
  - Outer non-binary FEC code in GF(2^4): (2,1,1)
- **First inner-FEC codeword contains one bit error in the payload**
  - Correctable
- **Second inner-FEC codeword contains one bit error in the payload, and one in the overhead**
  - Not Correctable
- **How many post-FEC bit errors in this trellis path?**



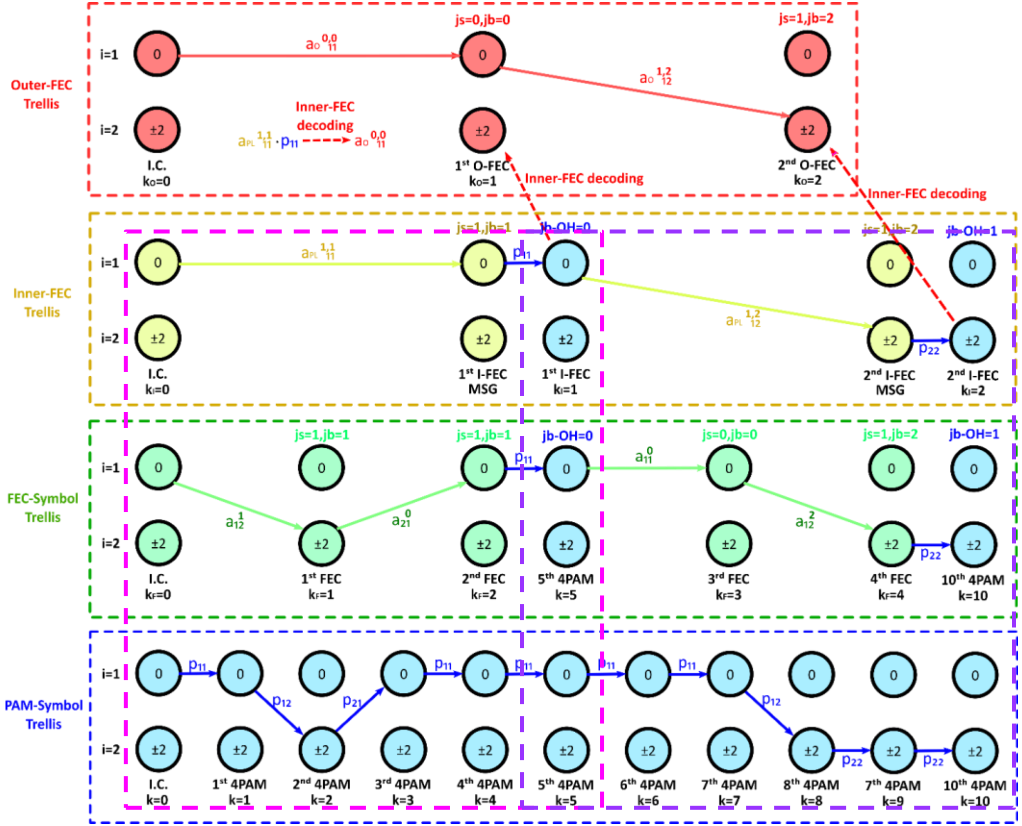
Concatenated FEC Trellis Path Example

- **In this example:**
  - Inner binary FEC code: (10,8,1)
  - Outer non-binary FEC code in GF(2^4): (2,1,1)

- **First inner-FEC codeword contains one bit error in the payload**
  - Correctable

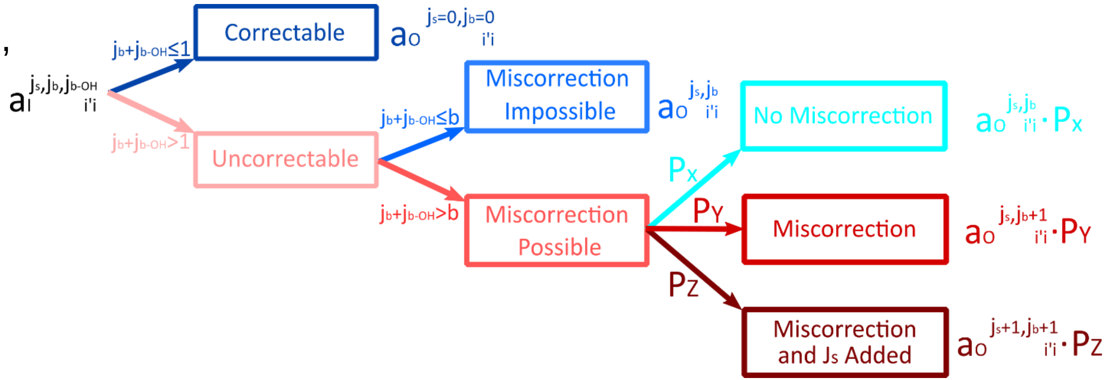
- **Second inner-FEC codeword contains one bit error in the payload, and one in the overhead**
  - Not Correctable

- **How many post-FEC bit errors in this trellis path?**



Inner-FEC Miscorrection – Statistical Model

- If an inner-FEC codeword is not correctable, a miscorrection may occur adding one bit error to the codeword
- If a miscorrection occurs, the additional bit error may appear in an already corrupted FEC symbol with probability P_Y
  - No FEC-symbol error added
- The additional bit error can also appear in an error-free FEC symbol with probability P_Z
  - FEC symbol error added

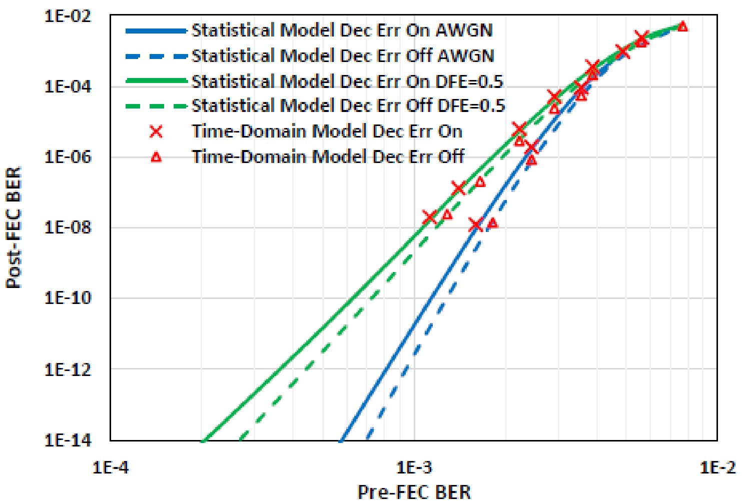


All Possible scenarios for inner-FEC decoding with correctability of 1

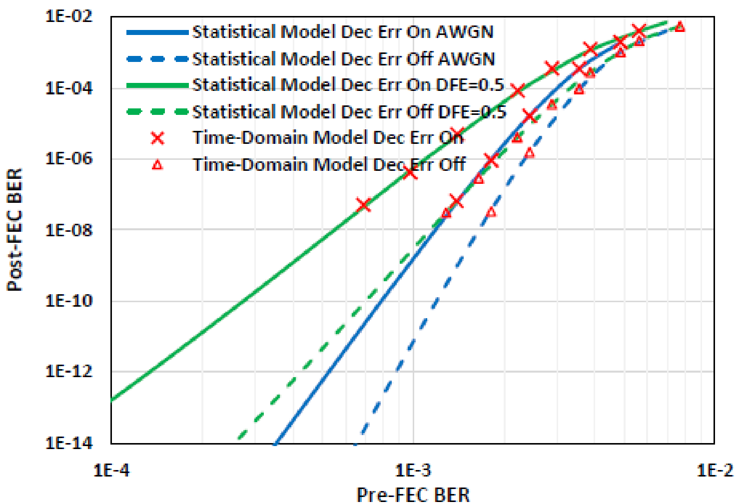
- **Hybrid approach:**
  - Probability of miscorrections determined from a short time-domain simulation
  - Used during correction (inner-FEC decoding) step of statistical model

Concatenated FEC – Simulation Results

- The Hamming (128,120,1) inner-FEC code outperforms BCH (144,136,1)
- Hamming code is also less impacted by decoding errors



Simulation results of a KP4 + Hamming (128,120,1) Concatenated FEC

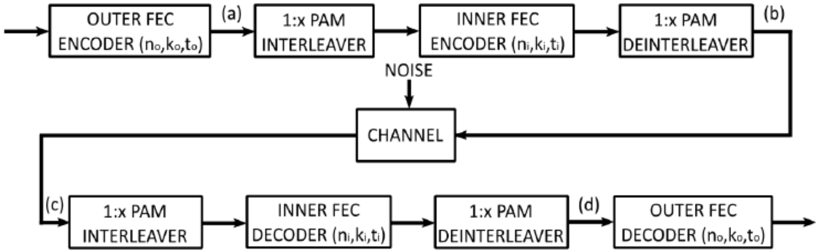


Simulation results of a KP4 + BCH (144,136,1) Concatenated FEC

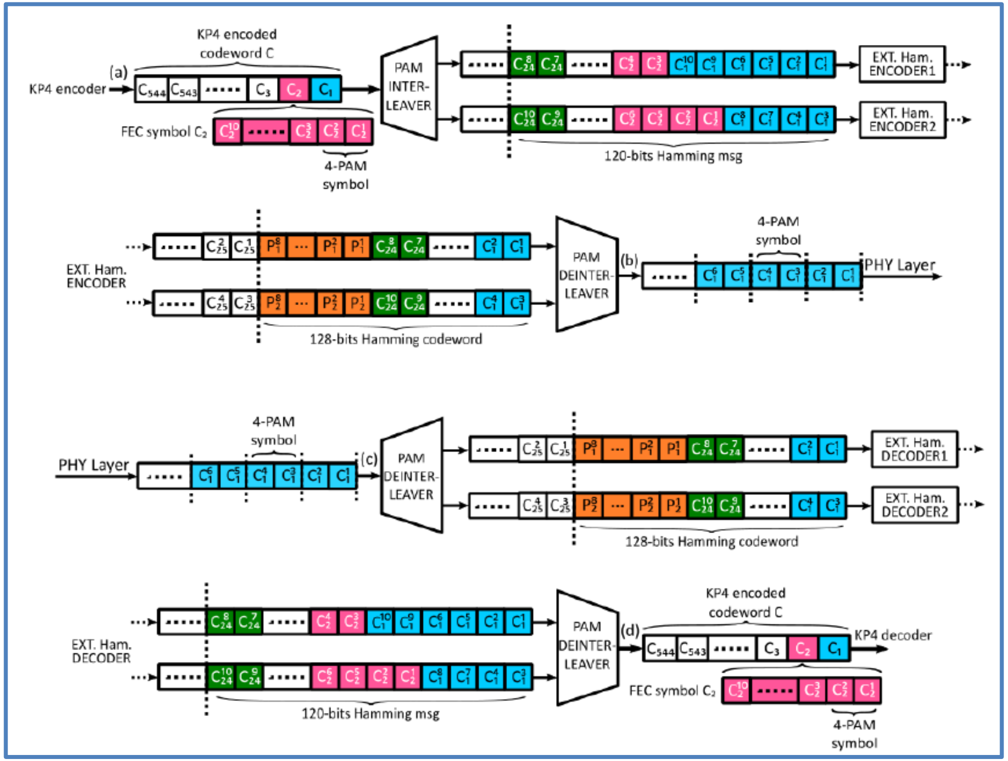
Outline

1. Motivation
2. Statistical Analysis of End-to-End RS FEC
3. Statistical Analysis of Concatenated FEC
4. Modeling Inner-FEC Interleaving in Concatenated FEC
   a. System Overview
   b. Trellis Model for 1:2 interleaving
   c. Trellis Model for 1:4 interleaving
   d. Simulation Results
5. Conclusion

Inner-FEC interleaving – System Overview



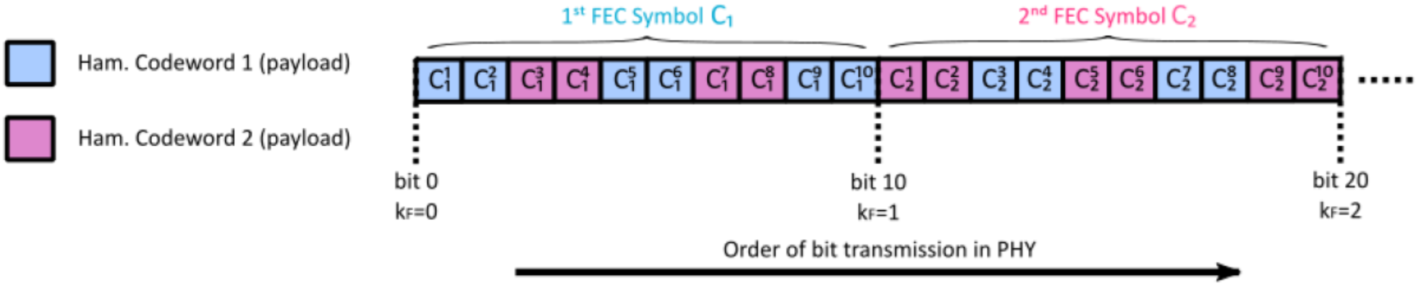
- **Inner-FEC interleaving protects coding gain in the presence of burst errors**
- **PAM symbols are split into different streams and separately encoded and decoded**
- **The order of PAM symbols in the encoded KP4 codeword is the same as the PAM symbols in PHY**



Bit-stream example of a KP4 + Hamming (128,120,1) concatenated FEC with 1:2 inner interleaving

1:2 Interleaving – “FEC-symbol Trellis”

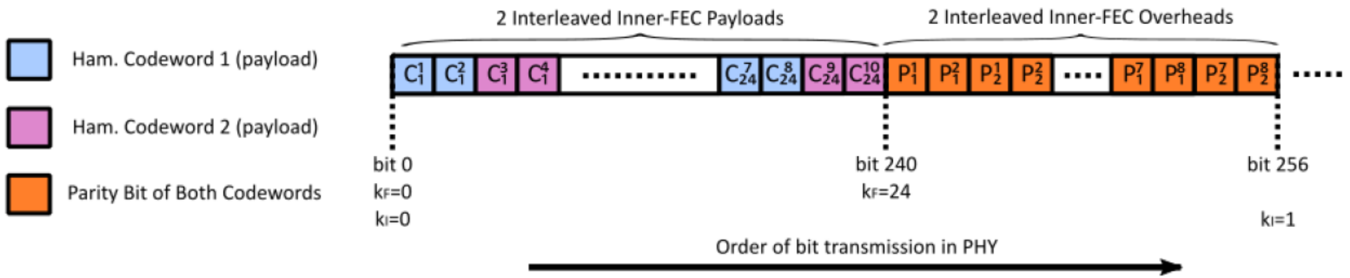
- The same 4-layer trellis model is used for inner-FEC interleaving, with some modifications
- With 1:2 interleaving on inner FEC, consecutive PAM symbols in the PHY layer are distributed to different inner-FEC codewords
  - Probability of miscorrection is minimized in the presence of burst errors
- The FEC-symbol transition probabilities track the number of bit errors in each of the two inner-FEC codewords simultaneously



Example of PAM-symbol distribution between 1:2 interleaved inner codewords

Interleaving – “Inner-FEC Trellis”

- Every 2 interleaved codewords are now traversed in the Inner-FEC trellis
- Tracking the following errors allows us to decode both interleaved codewords
  - Number of bit errors in codeword 1
  - Number of bit errors in codeword 2
  - Number of FEC symbol errors corrupted by only errors in codeword 1
  - Number of FEC symbol errors corrupted by only errors in codeword 2
  - Number of FEC symbol errors corrupted by errors appearing in both codewords

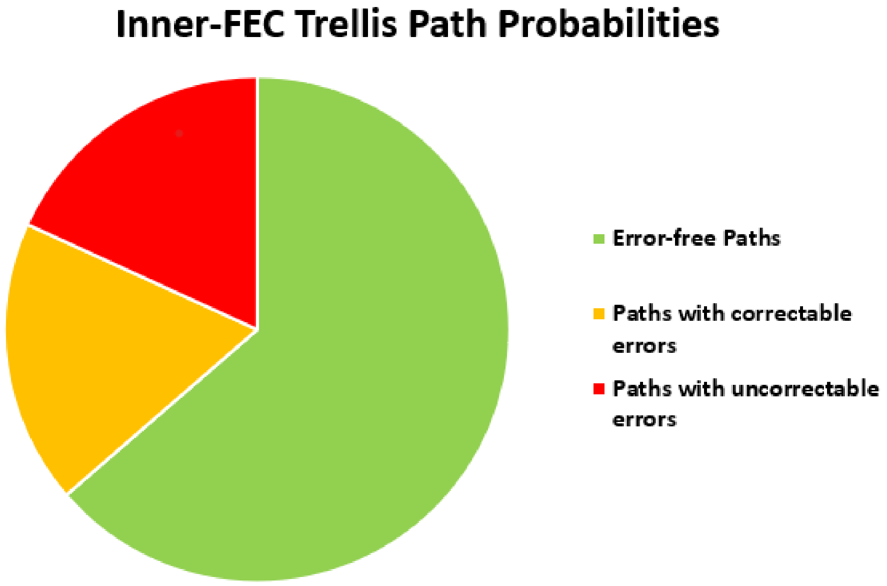


Example of 1:2 interleaved Hamming(128,120,1) codewords

- Outer-FEC trellis and post-FEC BER calculation are the same as with no interleaving

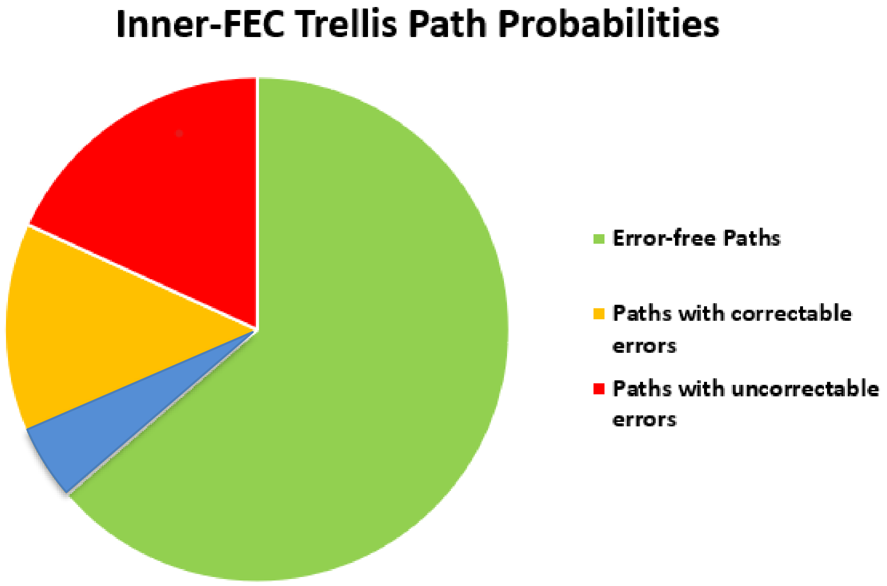
1:4 Interleaving

- Computational complexity of the statistical model quickly grows with higher-order interleaving such as 1:4
  - The number of error patterns that must be tracked for 1:*x* interleaving : $2^x - 1$
- Considering all these error indices jointly produces too many different trellis paths to track
- Correctable inner-FEC trellis paths have exactly one of the following indices that is non-zero:
  - $j_{s1}, j_{s2}, j_{s3}, j_{s4}, j_{s12}, j_{s13}, j_{s14}, j_{s23}, j_{s24}, j_{s34}, j_{s123}, j_{s124}, j_{s134}, j_{s234}, j_{s1234}$
- Instead of tracking all error indices, we partition the correctable trellis paths into mutually exclusive cases resulting in much lower computational complexity



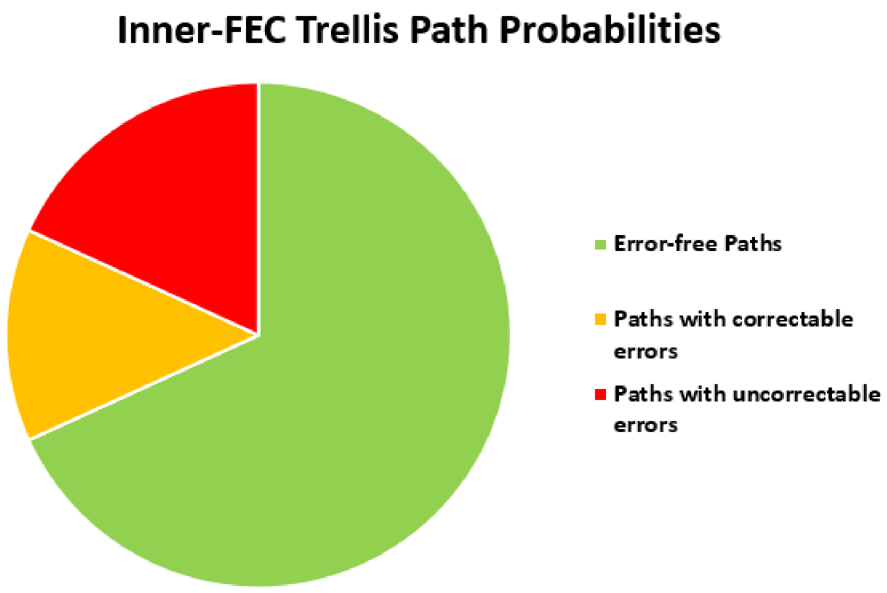
1:4 Interleaving

- Computational complexity of the statistical model quickly grows with higher-order interleaving such as 1:4
  - The number of error patterns that must be tracked for 1:*x* interleaving : $2^x - 1$
- Considering all these error indices jointly produces too many different trellis paths to track
- Correctable inner-FEC trellis paths have exactly one of the following indices that is non-zero:
  - $j_{s1}, j_{s2}, j_{s3}, j_{s4}, j_{s12}, j_{s13}, j_{s14}, j_{s23}, j_{s24}, j_{s34}, j_{s123}, j_{s124}, j_{s134}, j_{s234}, j_{s1234}$
- Instead of tracking all error indices, we partition the correctable trellis paths into mutually exclusive cases resulting in much lower computational complexity



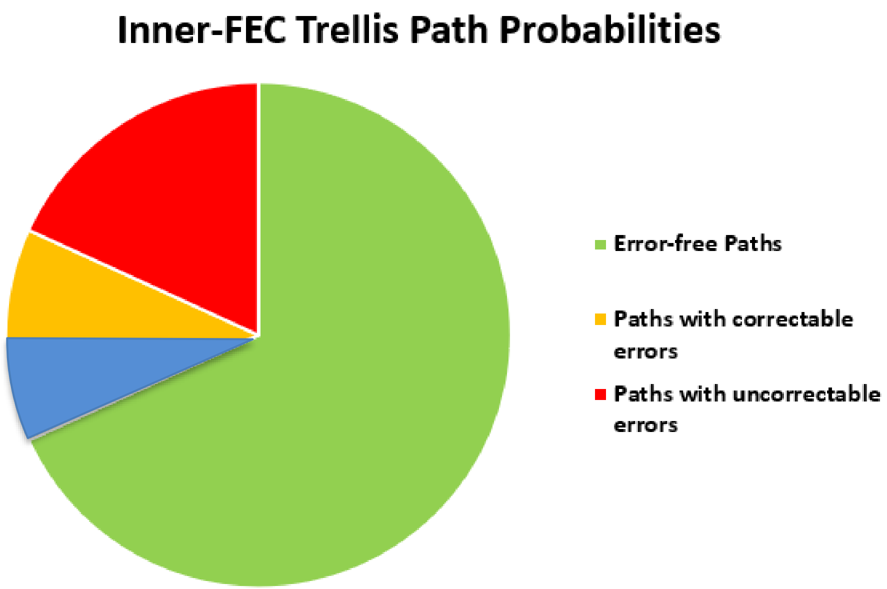
1:4 Interleaving

- Computational complexity of the statistical model quickly grows with higher-order interleaving such as 1:4
  - The number of error patterns that must be tracked for 1:*x* interleaving : $2^x - 1$
- Considering all these error indices jointly produces too many different trellis paths to track
- Correctable inner-FEC trellis paths have exactly one of the following indices that is non-zero:
  - $j_{s1}, j_{s2}, j_{s3}, j_{s4}$, $j_{s12}, j_{s13}, j_{s14}, j_{s23}, j_{s24}, j_{s34}, j_{s123}, j_{s124}, j_{s134}, j_{s234}, j_{s1234}$
- Instead of tracking all error indices, we partition the correctable trellis paths into mutually exclusive cases resulting in much lower computational complexity



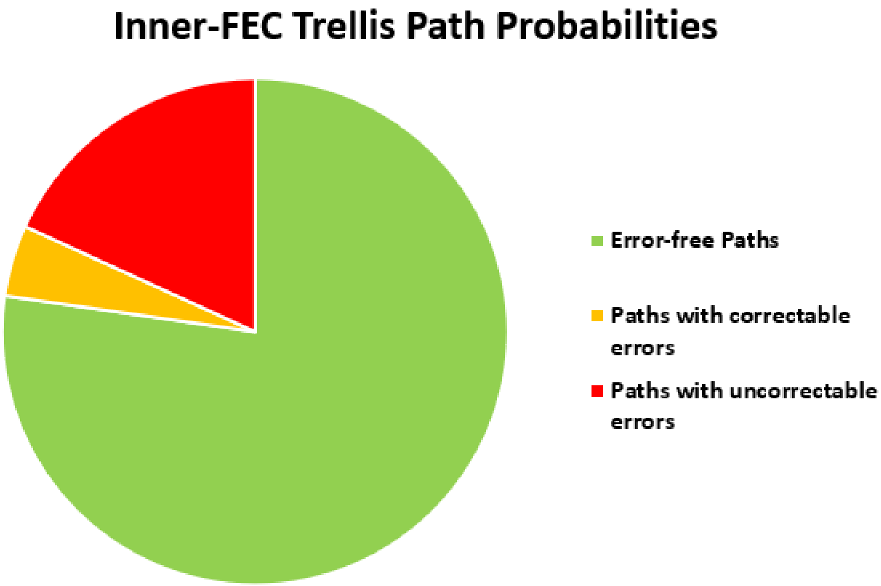
1:4 Interleaving

- Computational complexity of the statistical model quickly grows with higher-order interleaving such as 1:4
  - The number of error patterns that must be tracked for 1:*x* interleaving : $2^x - 1$
- Considering all these error indices jointly produces too many different trellis paths to track
- Correctable inner-FEC trellis paths have exactly one of the following indices that is non-zero:
  - $j_{s1}, j_{s2}, j_{s3}, j_{s4}, j_{s12}, j_{s13}, j_{s14}, j_{s23}, j_{s24}, j_{s34}, j_{s123}, j_{s124}, j_{s134}, j_{s234}, j_{s1234}$
- Instead of tracking all error indices, we partition the correctable trellis paths into mutually exclusive cases resulting in much lower computational complexity



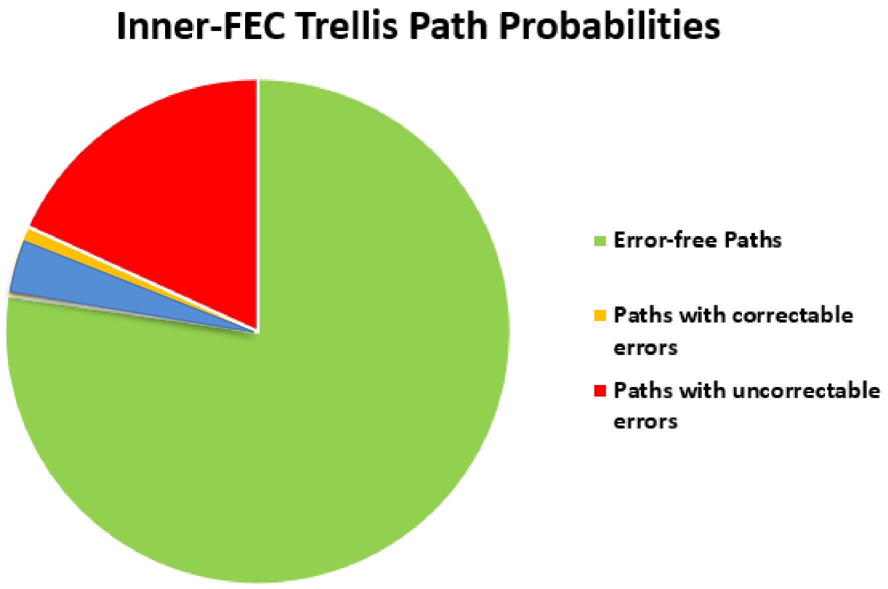
1:4 Interleaving

- Computational complexity of the statistical model quickly grows with higher-order interleaving such as 1:4
  - The number of error patterns that must be tracked for 1:*x* interleaving : 2^x -1
- Considering all these error indices jointly produces too many different trellis paths to track
- Correctable inner-FEC trellis paths have exactly one of the following indices that is non-zero:
  - $j_{s1}, j_{s2}, j_{s3}, j_{s4}, j_{s12}, j_{s13}, j_{s14}, j_{s23}, j_{s24}, j_{s34},$ $j_{s123}, j_{s124}, j_{s134}, j_{s234}, j_{s1234}$
- Instead of tracking all error indices, we partition the correctable trellis paths into mutually exclusive cases resulting in much lower computational complexity



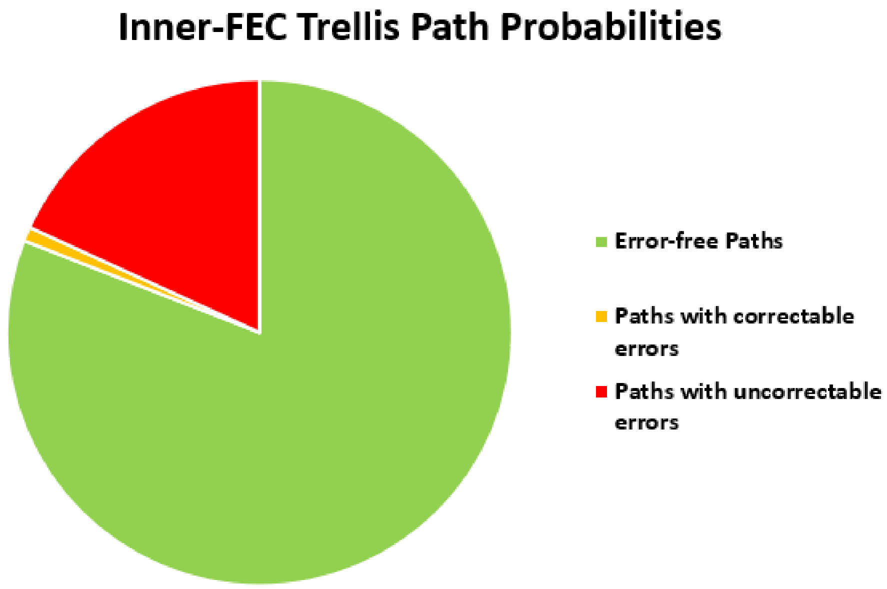
1:4 Interleaving

- Computational complexity of the statistical model quickly grows with higher-order interleaving such as 1:4
  - The number of error patterns that must be tracked for 1:x interleaving : 2^x -1
- Considering all these error indices jointly produces too many different trellis paths to track
- Correctable inner-FEC trellis paths have exactly one of the following indices that is non-zero:
  - $j_{s1}, j_{s2}, j_{s3}, j_{s4}, j_{s12}, j_{s13}, j_{s14}, j_{s23}, j_{s24}, j_{s34}, j_{s123}, j_{s124}, j_{s134}, j_{s234}, j_{s1234}$
- Instead of tracking all error indices, we partition the correctable trellis paths into mutually exclusive cases resulting in much lower computational complexity



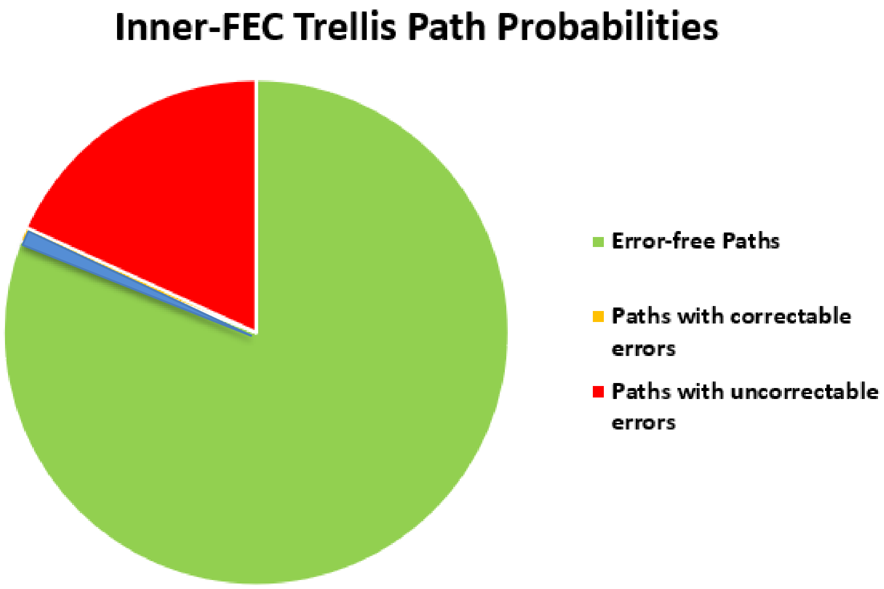
1:4 Interleaving

- Computational complexity of the statistical model quickly grows with higher-order interleaving such as 1:4
  - The number of error patterns that must be tracked for 1:*x* interleaving : $2^x - 1$
- Considering all these error indices jointly produces too many different trellis paths to track
- Correctable inner-FEC trellis paths have exactly one of the following indices that is non-zero:
  - $j_{s1}, j_{s2}, j_{s3}, j_{s4}, j_{s12}, j_{s13}, j_{s14}, j_{s23}, j_{s24}, j_{s34}, j_{s123}, j_{s124}, j_{s134}, j_{s234}, j_{s1234}$
- Instead of tracking all error indices, we partition the correctable trellis paths into mutually exclusive cases resulting in much lower computational complexity



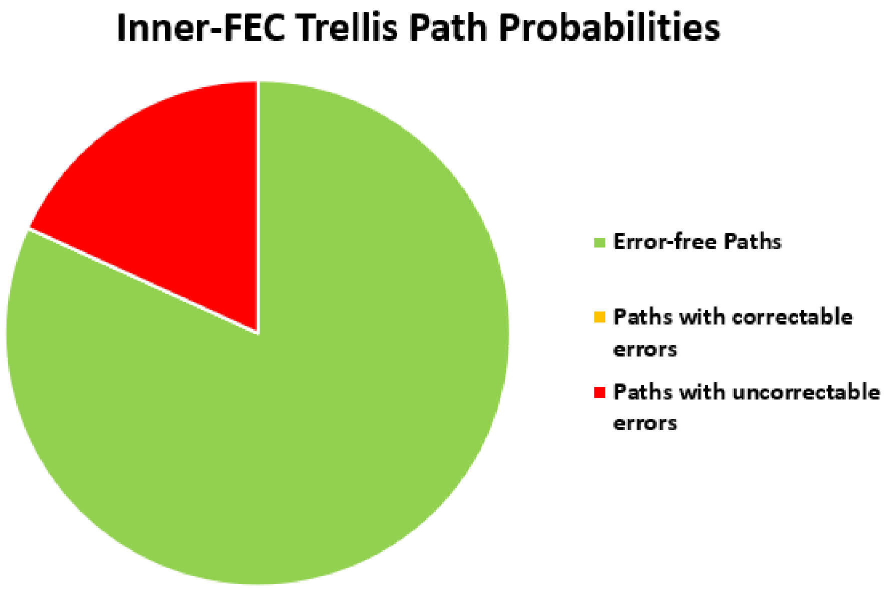
1:4 Interleaving

- Computational complexity of the statistical model quickly grows with higher-order interleaving such as 1:4
  - The number of error patterns that must be tracked for 1:*x* interleaving : $2^x - 1$
- Considering all these error indices jointly produces too many different trellis paths to track
- Correctable inner-FEC trellis paths have exactly one of the following indices that is non-zero:
  - j_{s1}, j_{s2}, j_{s3}, j_{s4}, j_{s12}, j_{s13}, j_{s14}, j_{s23}, j_{s24}, j_{s34}, j_{s123}, j_{s124}, j_{s134}, j_{s234}, j_{s1234}
- Instead of tracking all error indices, we partition the correctable trellis paths into mutually exclusive cases resulting in much lower computational complexity



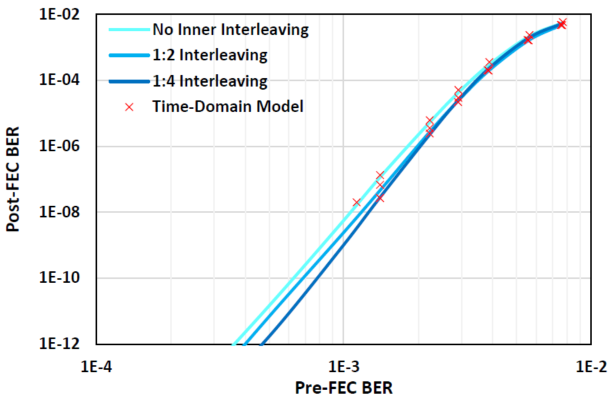
1:4 Interleaving

- Computational complexity of the statistical model quickly grows with higher-order interleaving such as 1:4
  - The number of error patterns that must be tracked for 1:x interleaving : $2^x - 1$
- Considering all these error indices jointly produces too many different trellis paths to track
- Correctable inner-FEC trellis paths have exactly one of the following indices that is non-zero:
  - $j_{s1}, j_{s2}, j_{s3}, j_{s4}, j_{s12}, j_{s13}, j_{s14}, j_{s23}, j_{s24}, j_{s34}, j_{s123}, j_{s124}, j_{s134}, j_{s234}, j_{s1234}$
- Instead of tracking all error indices, we partition the correctable trellis paths into mutually exclusive cases resulting in much lower computational complexity

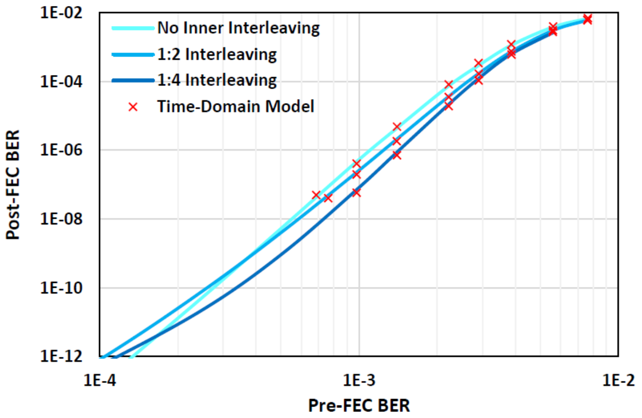


Concatenated FEC with inner interleaving – Simulation Results

- In the presence of burst errors, both Hamming and BCH codes show improvement with interleaving at low BER
    - These plots both have channel with AGWN + 0.5 DFE tap weight
- With the BCH code, higher interleaving results in reaching the error floor at a higher BER due to inner-FEC miscorrections introducing more FEC symbol errors



Simulation results of a KP4 + Hamming (128,120,1) Concatenated FEC with different interleaving schemes



Simulation results of a KP4 + BCH (144,136,1) Concatenated FEC with different interleaving schemes

Outline

Conclusion

- We presented a statistical model for concatenated FEC architectures considered for 200+ Gbps applications

- Using this approach, we can accurately predict post-FEC BER and observe:
  - Good correlation between time-domain and statistical model for both BCH(144,136,1) and Hamming(128,120,1) inner FEC codes
  - The “error floor” imposed by burst errors
  - The impact of inner FEC interleaving on post-FEC BER for 1:2 and 1:4 interleaving schemes.

- The model was validated using a time-domain simulation with DFE error propagation

References

1. M. Yang, S. Shahramian, H. Shakiba, H. Wong, P. Krotnev and A. C. Carusone, “Statistical BER Analysis of Wireline Links With Non-Binary Linear Block Codes Subject to DFE Error Propagation,” in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 1, pp. 284-297, Jan. 2020, doi: 10.1109/TCSI.2019.2943569.
2. M. Yang, S. Shahramian, H. Shakiba, H. Wong, P. Krotnev and A. Carusone, “A Statistical Modeling Approach for FEC-Encoded High-Speed Wireline Links,” *DesignCon 2020*, Santa Clara, CA, 2020.
3. Concatenated FEC Proposal for 200G/Lane PMD, *IEEE Standard 802.3df*, May. 2022.
4. FEC Performance for 200 Gb/s per Lane Optical PHY and Interoperating, *IEEE Standard 802.3df*, May. 2022.
5. S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs, NJ, Prentice Hall, 2004.
6. M. Isaka and M. Fossorier, "High-rate serially concatenated coding with extended Hamming codes," *in IEEE Communications Letters*, vol. 9, no. 2, pp. 160-162, Feb. 2005.
7. Proposal for a specific (128,120) extended inner Hamming Code with lower power and lower latency soft Chase decoding than textbook codes, *IEEE Standard 802.3df*, Oct. 2022.
8. Constructing a BCH/Hamming Inner Code for 200 Gb/s per lane PMDs, *IEEE Standard 802.3df*, Oct. 2022.
9. Concatenated SFEC proposal for 200Gb/s per Lane IM-DD Optical PMD, *IEEE Standard 802.3df*, Nov. 2022.

MORE INFORMATION

- Integrated Systems Laboratory
- Principal Investigator: Prof. Tony Chan Carusone
- isl.utoronto.ca/
- tony.chan.carusone@isl.utoronto.ca
- ming.yang@isl.utoronto.ca
- richard.barrie@isl.utoronto.ca

Thank you for attending !

QUESTIONS?

Thank you for attending this webinar!

Join us again for Advanced Manufacturing Minneapolis

October 10-11, 2023