DESIGNCON® 2023
WHERE THE CHIP MEETS THE BOARD

Statistical BER Analysis of Concatenated FEC in Multi-Part Links

Richard Barrie, University of Toronto
richard.barrie@isl.utoronto.ca

Ming Yang, University of Toronto/Alphawave SEMI
ming.yang@isl.utoronto.ca

Anthony Chan Carusone, University of Toronto/Alphawave SEMI
tony.chan.carusone@isl.utoronto.ca

Abstract

This paper presents a statistical model that accurately estimates post-FEC BER for multi-part links using a concatenated FEC. Both the inner and outer codes in the concatenated FEC are assumed to be linear block codes, and the inner code may include codeword interleaving. A hierarchical approach is adopted to analyze the propagation of PAM-symbol and FEC-symbol errors through a multi-layer Markov model in the presence of DFE error propagation. The proposed model also considers miscorrections introduced by the inner-FEC decoder, a significant source of error in concatenated FEC architectures. A hybrid approach is used to model miscorrections, with the probability of miscorrections determined from a separate time-domain simulation. The proposed statistical model can be combined with our existing approaches to model other noise sources such as residual ISI, crosstalk, transmitter and receiver jitter, and other system-level design choices including precoding and bit-multiplexing.

Authors Biography

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 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. He also works as a system engineer at Alphawave IP Group. His research interests are in analog integrated circuit design, on-chip analog signal processing and high-performance integrated circuit testing.

Anthony Chan Carusone received his Ph.D. from the University of Toronto in 2002 and has since been a professor in the University of Toronto's Department of Electrical and Computer Engineering. He has also been a consultant to industry in the areas of integrated circuit design and digital communication since 1997. He is currently the Chief Technology Officer of Alphawave IP Group in Toronto, Canada.
Prof. Chan Carusone co-authored Best Student Papers at the 2007, 2008, 2011, and 2022 Custom Integrated Circuits Conferences, the Best Invited Paper at the 2010 Custom Integrated Circuits Conference, the Best Paper at the 2005 Compound Semiconductor Integrated Circuits Symposium, the Best Young Scientist Paper at the 2014 European Solid-State Circuits Conference, and a Best Paper at DesignCon 2021. He co-authored the popular textbooks "Analog Integrated Circuit Design" (along with D. Johns and K. Martin) and "Microelectronic Circuits," 8th edition (along with A. Sedra, K.C. Smith and V. Gaudet). He was Editor-in-Chief of the IEEE Transactions on Circuits and Systems II: Express Briefs in 2009, and an Associate Editor for the IEEE Journal of Solid-State Circuits 2010-2017. He was a Distinguished Lecturer for the IEEE Solid-State Circuits Society 2015-2017 and has served on the Technical Program Committee of several IEEE conferences including the International Solid-State Circuits Conference 2016-2021. He is currently the Editor-in-Chief of the IEEE Solid-State Circuits Letters and is an IEEE Fellow.

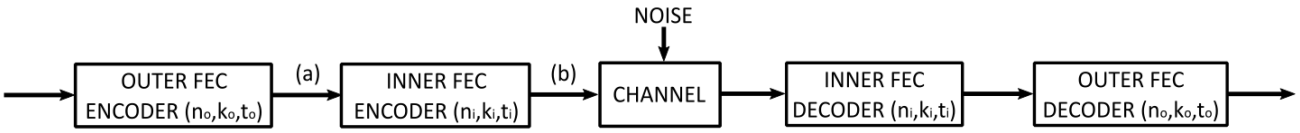


Figure 1. System-level diagram of a serially concatenated FEC with an outer code (n_o, k_o, t_o), and an inner code (n_i, k_i, t_i).

1. Introduction

Forward error correction (FEC) codes are necessary for modern high-speed wireline links to achieve acceptably low bit error rates (BER). The post-FEC BER of a wireline link cannot be accurately estimated from only knowing the pre-FEC BER and the FEC architecture; other factors such as equalization techniques, decision feedback equalizer (DFE) error propagation, inter-symbol interference, crosstalk, and jitter must be considered. Ideally, one may perform a transient simulation to capture the characteristics of all noise sources fully. However, the targeted $<10^{-15}$ BERs make time-domain simulations prohibitively long, especially for exploring architectural design alternatives. Therefore, an efficient statistical model that accurately predicts very low post-FEC BERs serves an essential function in the design of high-speed wireline links.

In our previous work [1-2], we proposed a statistical model for standard non-binary linear block codes, such as the RS(544,514,15) KP4 and RS(528,514,7) KR4 codes. This FEC architecture is suitable for most wireline signaling up to 112 Gb/s. However, in 224 Gb/s per lane applications with 4-PAM signaling, the Nyquist frequency doubles from 28 GHz to 56 GHz. Doubling the Nyquist frequency generally hampers signal integrity resulting in less operating margin and a worse pre-FEC BER. Despite the progress being made on improving channel materials, transceiver architecture designs, and system-level innovations, stronger FECs are needed to loosen pre-FEC BER requirements for 200+ Gb/s applications. Thus the IEEE 802.3dj task force is considering a concatenated FEC architecture.

Figure 1 shows the system-level diagram of a serially concatenated FEC. The outer code is a non-binary linear block code, which can correct up to t_o FEC symbol errors. A typical choice for the outer code in IEEE 802.3dj is the standard RS KP4 (544,514,15) FEC in GF(2^{10}), which can correct up to $t_o = 15$ FEC symbol errors in a KP4 codeword with codeword length $n_o = 544$ FEC symbols, and payload length $k_i = 514$ FEC symbols. This choice of code with large t_o allows it to correct long burst errors caused by DFE error propagation which is crucial for links with the high DFE tap coefficients required for 200+ Gb/s signaling.

Next, an inner code is applied to provide another layer of protection against random bit errors. Due to stringent latency and complexity requirements, the inner code will typically be a simple code that can correct a small number of bit errors. Even codes that can only correct one bit error per codeword are effective in correcting random errors, allowing the outer block code to work more efficiently on long burst errors. Choices for the inner code include the Hamming(127,120,1), which can correct up to $t_i = 1$ bit in an inner codeword with codeword length $n_i = 127$ bits and payload length $k_i = 120$ bits, and BCH (144,136,1), which can correct up to $t_i = 1$ bit in an inner codeword with $n_i = 144$ bits and $k_i = 136$ bits. These codes have a minimum Hamming distance of three between valid codewords [3], which is relatively small. Because of this, inner codewords with more than one bit error are at risk of being decoded to the wrong codeword (miscorrected), adding a bit error to the codeword. These

inner-FEC miscorrections significantly worsen the post-FEC BER, as the additional bit error may corrupt an additional outer-FEC symbol. To improve the performance of the Hamming(127,120,1) code, an additional parity bit can be added to extend the minimum Hamming distance between codewords to 4. This is known as the extended Hamming (128,120,1) code [4]. Although this code can still only correct one bit error, it will not miscorrect codewords with two bit errors, improving the performance of the concatenated FEC.

This paper presents a statistical model that accurately estimates post-FEC BER for multi-part links using a concatenated FEC. We use a simplified link architecture as depicted in Figure 1, focusing on error analysis of the channel connecting the two modules, which may have both random and burst errors. The host-to-module interfaces in the multi-part link are assumed to be benign and do not significantly contribute to the post-FEC BER. Both the inner and outer codes in the concatenated FEC are assumed to be linear block codes, and the inner code may include codeword interleaving. A hierarchical approach is adopted to analyze the propagation of PAM-symbol and FEC-symbol errors through a multi-layer Markov model in the presence of DFE error propagation. Our model also considers inner-FEC miscorrections. The proposed statistical model can be combined with our existing approaches [2] to model noise sources such as residual ISI, crosstalk, transmitter and receiver jitter, and other system-level design choices such as precoding and bit-multiplexing. A series of techniques, including state aggregation, time aggregation, state reduction, and dynamic programming are used, making the time complexity to compute post-FEC BERs below 10^{-15} reasonable. The model's accuracy is verified by comparing pre-FEC vs. post-FEC BER plots with time-domain simulation results. The implementation details of the concatenated FEC for the next generation of high-speed wireline links, such as the choice of inner code and inner-FEC interleaving scheme, are still being discussed by the IEEE 802.3dj task force, and many candidates were proposed [5-7]. The scope of this paper covers some of the most popular options, but the modeling methodologies presented can be easily extended to other candidates.

The main body of this work is divided into two sections. In Section 2, we propose a statistical model for concatenated FEC codes. A hybrid approach is used to model inner-FEC miscorrections, with information on the probability of miscorrections determined by a separate time-domain simulation. In Section 3, we expand the proposed statistical model to consider inner-FEC interleaving. We propose a novel method to partition the correctable trellis paths that significantly reduces the computational complexity for tracking the error information in each interleaved inner codeword. Finally, we conclude our work in Section 4.

2. Modeling Concatenated FEC

2.1 General Approach

In this subsection, we present the trellis model for concatenated FEC shown in Figure 2, which contains four levels of hierarchy: The PAM-symbol trellis and the FEC-symbol trellis were first proposed in [1], where we presented a trellis model for an end-to-end RS FEC. Building on this work, we include two more levels of hierarchy: the inner-FEC trellis and outer-FEC trellis, to model a concatenated FEC architecture.

Figure 2 shows the trellis model of a concatenated FEC with a hypothetical non-binary outer code in GF(2^4) assuming ($n_o = 2$, $k_o = 1$, $t_o = 1$) and a binary inner code assuming ($n_i = 10$, $k_i = 8$, $t_i = 1$). We

describe the four layers in Figure 2 from the bottom to up. The PAM-symbol trellis models state transitions between PAM symbols using the transition probabilities 'p'. The PAM-symbol trellis is time-aggregated to form the FEC-symbol trellis, that models state transitions between FEC symbols with the transition probabilities 'a'. Next, the inner-FEC trellis builds a single-step transition probability for the inner codeword 'a_I', including the inner-FEC payload and overhead. Finally, the outer-FEC trellis models the state transition 'a_O' between decoded inner codewords until the end of the outer codeword, where the post-FEC BER can be calculated. 'a_O' is generated by applying inner-FEC decoding to 'a_I', removing bit errors in correctable trellis paths. Subsections 2.1.1-2.1.5 describe how this trellis model is constructed for general inner and outer codes. Specifically, we focus on calculating the one-step state transition probabilities and the trellis-path probabilities in each trellis layer.

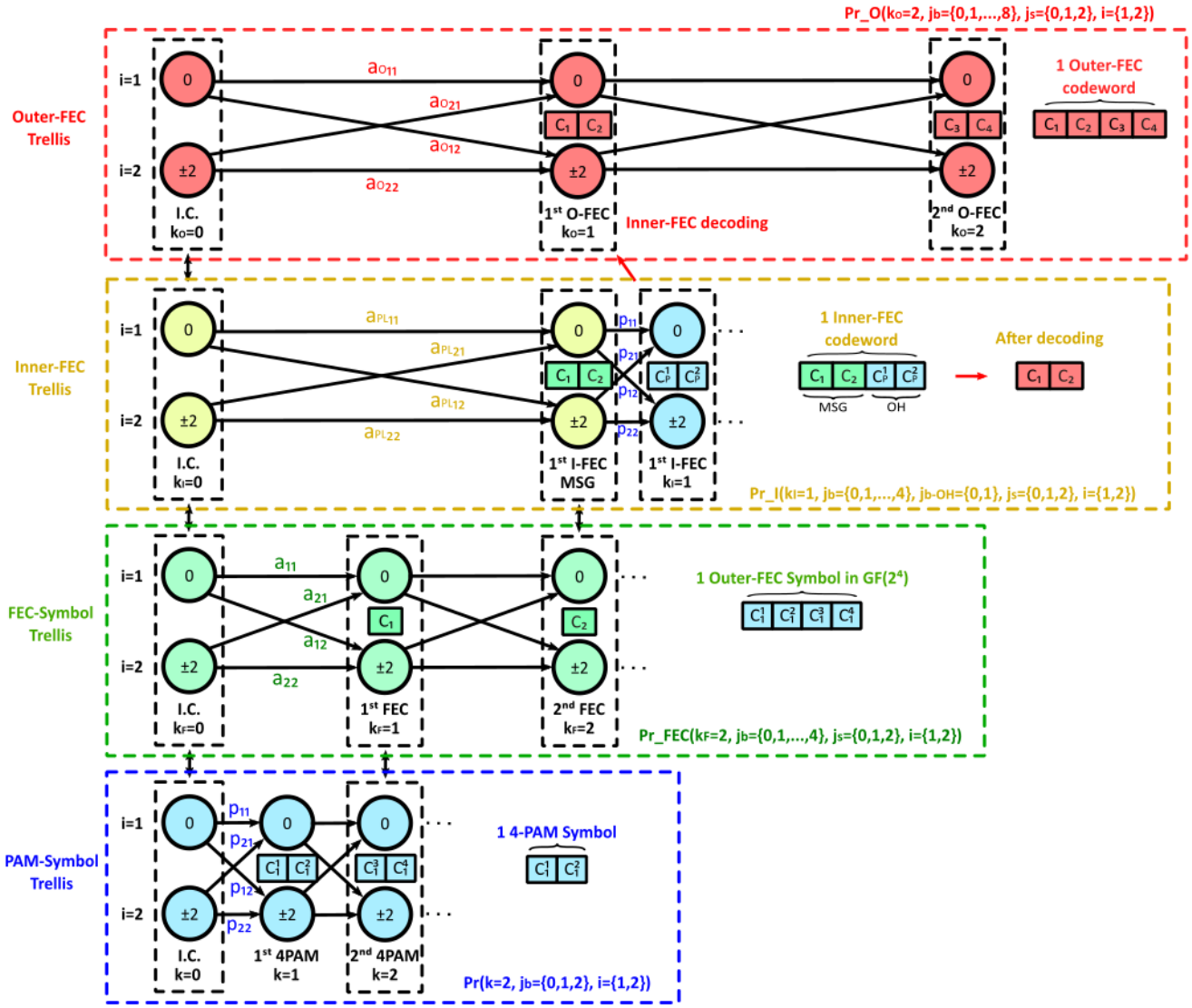


Figure 2. Trellis model of a hypothetical concatenated FEC assuming a non-binary outer code (2,1,1) in GF(2^4) and a binary inner code (10,8,1). The unlikely error states ±4, and ±6 are disregarded for simplicity of the example.

2.1.1 PAM-Symbol Trellis

In [1], we have shown a methodology for modeling an N-tap DFE with 4-PAM signaling using a Markov model containing 16^N states and that this model can be simplified to 4^N states using state aggregation. All examples shown in this paper assume 4-PAM signaling with a 1-tap DFE. At time index k corresponding to the k^{th} transmitted 4-PAM symbol, the aggregated Markov model contains error states $D_k \in \{0, \pm2, \pm4, \pm6\}$, corresponding to the difference in voltage level between transmitted and received 4-PAM symbols. In the Markov model, the state-transition probability from a source state 'i'' to a sink state 'i' in the PAM-symbol trellis is denoted as $p_{i'i}$. We find the steady-state probability, π_i, of any state i in the Markov model by solving the global balance equation [8],

$$\pi_i = \sum_{i'} p_{i'i}\pi_{i'}, \tag{1}$$

subject to

$$\sum_i \pi_i = 1. \tag{2}$$

We next apply trellis-based dynamic programming to the Markov model to efficiently calculate the probability of bit errors in a codeword. Rather than finding the BER by enumerating all possible error patterns in the trellis, dynamic programming solves the problem much faster by grouping the probability of all trellis paths having the same number of bit errors. When traversing a PAM-symbol trellis using dynamic programming, each state transition introduces either 0, 1, or 2 bit errors. We define j_{PAM} as the number of bit errors in a PAM symbol detection. Assuming gray-coding, an error value ±2 or ±6 corresponds to $j_{PAM} = 1$, and an error value ±4 corresponds to $j_{PAM} = 2$. We define $Pr^j_k(i)$ as the probability of arriving at state i at the k^{th} stage of the PAM-symbol trellis after traversing all trellis paths containing exactly j bit errors. In each trellis iteration, for states 'i' where the most recently received 4-PAM symbol has j_{PAM}-bit errors,

$$Pr^j_{k+1}(i) = \sum_{i'} Pr_k^{j-j_{PAM}}(i')p_{i'i}\ . \tag{3}$$

Figure 3 shows the PAM-symbol trellis from the example in Figure 2. The unlikely error states ±4, and ±6 are disregarded for the simplicity of the example. All possible paths from the 0th PAM symbol ($k = 0$) to the 2nd PAM symbol ($k = 2$) are shown with arrows. The probabilities of arriving at the 2nd PAM symbol with $j \in \{0,1,2\}$ bit errors are calculated using the dynamic programming procedure described in Equation 3.

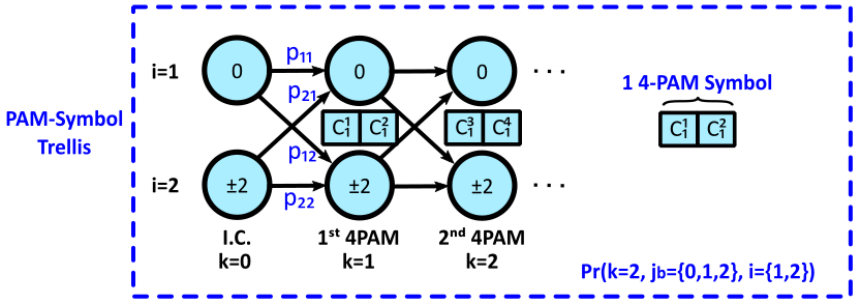


Figure 3. Example of a PAM-symbol trellis.

2.1.2 FEC-Symbol Trellis

Using the methods described so far, every outer-FEC symbol in GF(2^m) can be decomposed into a length-m/2 4-PAM trellis describing the link behavior in the physical layer. We apply Equation 3 to recursively compute $\Pr_k^j(i)$, aggregating the probability of error patterns having exactly j bit errors, where $j \in \{0 \dots m/2\}$.

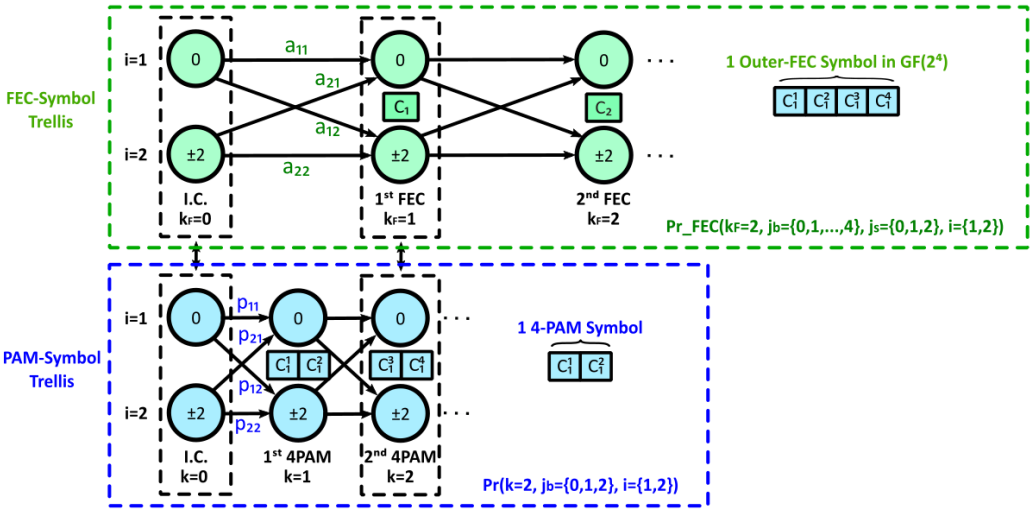


Figure 4. Example of a FEC-symbol trellis in GF(2^4), each state transition in the FEC-symbol trellis is generated by time-aggregating the underlying PAM-symbol trellis paths.

Assuming a generalized N-tap DFE and $D_k \in \{0, \pm2, \pm4, \pm6\}$, all paths in the PAM-symbol trellis representing $Pr^j_k(i)$ can be decomposed into 4^N groups of trellis paths, each starting with one of the 4^N Markov states at $k = 0$. Each of these paths can be considered as a one-step state transition equivalent to traversing m/2 4-PAM stages. As such, we may construct a new trellis model for the entire FEC block, assuming that each state transition from the k_F^{th} to the $(k_F+1)^{\text{th}}$ stage has traversed a group of length-m/2 PAM-symbol-trellis paths. This is known as the time aggregation of a Markov decision process [9]; we group trellis paths over m/2 consecutive 4-PAM symbols while the time-aggregated Markov model preserves both the time-homogeneity and bit-error information. We call this time-aggregated PAM-symbol trellis the FEC-symbol trellis, distinguishing it from the PAM symbol-level trellis considered thus far. For the FEC-symbol trellis example shown in Figure 4, every two 4-PAM symbols are time-aggregated to form a one-step transition in the FEC-symbol trellis.

To analyze the FEC trellis, we must first find all the state-transition probabilities $a^j_{i'i}$ of these 4^N states by analysis of each underlying PAM-symbol trellis. First, we instantiate the expanded PAM-symbol trellis by assuming that the trellis starts at the state 'i" in $a^j_{i'i}$ with a probability of 1,

$$Pr\,{}^0_0(i') = 1. \tag{4}$$

Next, after traversing the expanded PAM-symbol trellis using the dynamic programming procedure described in Equation 3, the transition probability $a^j_{i'i}$ to the next $(k_F+1)^{\text{th}}$ FEC-symbol trellis stage can be calculated by summing the probability of all j-bit-error PAM-symbol-trellis paths ending at state 'i',

$$a\,{}^j_{i'i} = Pr^j_{k=m/2}(i)\Big|_{Pr\,{}^0_0(i')=1}. \tag{5}$$

In the FEC-symbol trellis, dynamic programming is also applied to enumerate the probability of all error patterns having more than t_o FEC symbol errors in a codeword. However, the dynamic programming algorithm described by Equation 3 can only track the total number of bit errors. Therefore, we create another error index allowing us to aggregate all error patterns in terms of both FEC symbol errors and bit errors. In the FEC-symbol trellis, we denote $Pr\_FEC_{k_F}^{j_s,j_b}(i)$ the probability of visiting Markov state i at the k_F^{th} FEC symbol after traversing all trellis paths containing exactly j_s FEC symbol errors and j_b bit errors. Hence, the error probabilities at time k_F +1, $Pr\_FEC_{k_F+1}^{j_s,j_b}(i)$, can be found iteratively from the values of $Pr\_FEC_{k_F}^{j_s,j_b}(i)$ and the transition probabilities $a_{i',i}^{j}$. We calculate $Pr\_FEC_{k_F+1}^{j_s,j_b}(i)$ by iterating over all possible j and i',

$$Pr\_FEC_{k_F+1}^{j_s,j_b}(i) = \sum_j \sum_{i'} Pr\_FEC_{k_F}^{j_s-min(1,j),j_b-j}(i') a_{i',i}^{j}. \quad (6)$$

2.1.3 Inner-FEC Trellis

In [1-2] we have shown the statistical modeling of a single end-to-end FEC where the FEC-symbol trellis is used to traverse the entire length of the outer codeword. When modeling the concatenated FEC architecture, another level of time aggregation is needed to model the inner codeword. In this subsection, we describe the procedure of constructing the inner-FEC trellis using the FEC-symbol trellis. Note that the FEC-symbol trellis refers to outer-FEC symbol transitions, not inner-FEC symbol transitions. As this paper only considers binary inner codes, we simply refer to an inner-FEC symbol as a bit.

Consider an inner-FEC payload containing k_i bits message that can be divided evenly into an integer number of $k_i/(m/2) = 2k_i/m$ outer-FEC symbols. We aggregate all the $2k_i/m$ outer-FEC symbols in the payload into a one-step state transition. We define $a_{PL}{}_{i',i}^{j_s,j_b}$ as the transition probability from source state *i'* at the beginning of the inner-FEC payload to sink state i at the end of the inner-FEC payload with all paths containing exactly j_s FEC symbol errors and j_b bit errors. By traversing a length-$2k_i/m$ FEC-symbol trellis, $a_{PL}{}_{i',i}^{j_s,j_b}$ can be calculated by,

$$a_{PL}{}_{i',i}^{j_s,j_b} = Pr\_FEC_{k_F=2k_i/m}^{j_s,j_b}(i)\Big|_{Pr_{k_F=0}^{0,0}(i')=1}. \quad (7)$$

After traversing the inner-FEC trellis to obtain a_{PL}, we consider the inner-FEC overhead for parity checks. Note that bit errors in the inner-FEC overhead are always discarded after decoding. Nonetheless, it is vital to consider the possibility of bit errors in the inner-FEC overhead because they affect if the inner codeword is correctable. Considering an inner codeword with an even number of n_i - k_i bits overhead that can be represented by a PAM-symbol trellis, we apply time aggregation to the PAM-symbol trellis to find the one-step transition probability $a_{OH}{}_{i',i}^{j_{b-OH}}$ of the inner-FEC overhead having exactly $j_{b\text{-}OH}$ bit errors,

$$a_{OH}{}_{i',i}^{j_{b-OH}} = Pr_{k=(n_i-k_i)/2}^{j_{b-OH}}(i)\Big|_{Pr\,_0^0(i')=1}. \quad (8)$$

We then aggregate the transition probabilities of the inner-FEC payload and inner-FEC overhead to obtain the one-step transition probability of the inner codeword. In the inner-FEC trellis, we denote the time index at the end of each inner codeword k_I. $a_I{}_{i'i}^{j_s,j_b,j_{b-OH}}$ is the state transition probability of going from source state 'i'' at the beginning of an inner-FEC payload to sink state 'i' at the end of an inner-FEC overhead, while having exactly j_s FEC-symbol errors and j_b bit errors in the payload, and

$j_{b\text{-}OH}$ bit errors in the overhead. $a_{I}{}_{i'i}^{j_s,j_b,j_{b-OH}}$ can be computed by time aggregating the payload and overhead of an inner codeword,

$$a_{I}{}_{i'i}^{j_s,j_b,j_{b-OH}} = Pr\_I_{k_I=1}^{j_s,j_b,j_{b-OH}}(i)|_{Pr\_I_{k_I=0}^{0,0,0}(i')=1} = \sum_{i\sim}(a_{PL}{}_{i'i\sim}^{j_s,j_b}\, a_{OH}{}_{i\sim i}^{j_{b-OH}}). \quad (9)$$

In Figure 5, we show an example of two FEC-symbol transitions that are aggregated to form the inner-FEC payload with one-step state transition probability a_{PL}. The inner-FEC overhead consists of only one PAM-symbol with transition probability $a_{OH} = p$. a_{PL} and a_{OH} are combined using Equation 9 to generate a_I.

Next, we apply inner-FEC decoding on a_I to produce a_O, which is the one-step transition probability of a decoded inner codeword in the outer-FEC trellis. a_O does not need to track the information about $j_{b\text{-}OH}$ because the inner-FEC overhead is discarded after decoding. The number of bit errors in the entire inner codeword is $j_b + j_{b\text{-}OH}$. Considering an inner code that can correct t_i bit errors, all trellis paths in the inner-FEC trellis with $j_b + j_{b\text{-}OH} \leq t_i$ are correctable. We iterate over all possible trellis paths in the inner-FEC trellis representing one inner-FEC codeword. If $j_b + j_{b\text{-}OH} \leq t_i$, these correctable paths are accumulated in the outer-FEC transitions a_O containing neither bit errors nor FEC-symbol errors,

$$a_{O}{}_{i'i}^{0,0} = a_{O}{}_{i'i}^{0,0} + a_{I}{}_{i'i}^{j_s,j_b,j_{b-OH}}. \quad (10)$$

For non-correctable paths in the inner-FEC trellis with $j_b + j_{b\text{-}OH} > t_i$, after decoding we add these paths to outer-FEC transitions a_O while preserving the same number of FEC-symbol errors and bit errors that occurred in the inner-FEC payload,

$$a_{O}{}_{i'i}^{j_s,j_b} = a_{O}{}_{i'i}^{j_s,j_b} + a_{I}{}_{i'i}^{j_s,j_b,j_{b-OH}}. \quad (11)$$

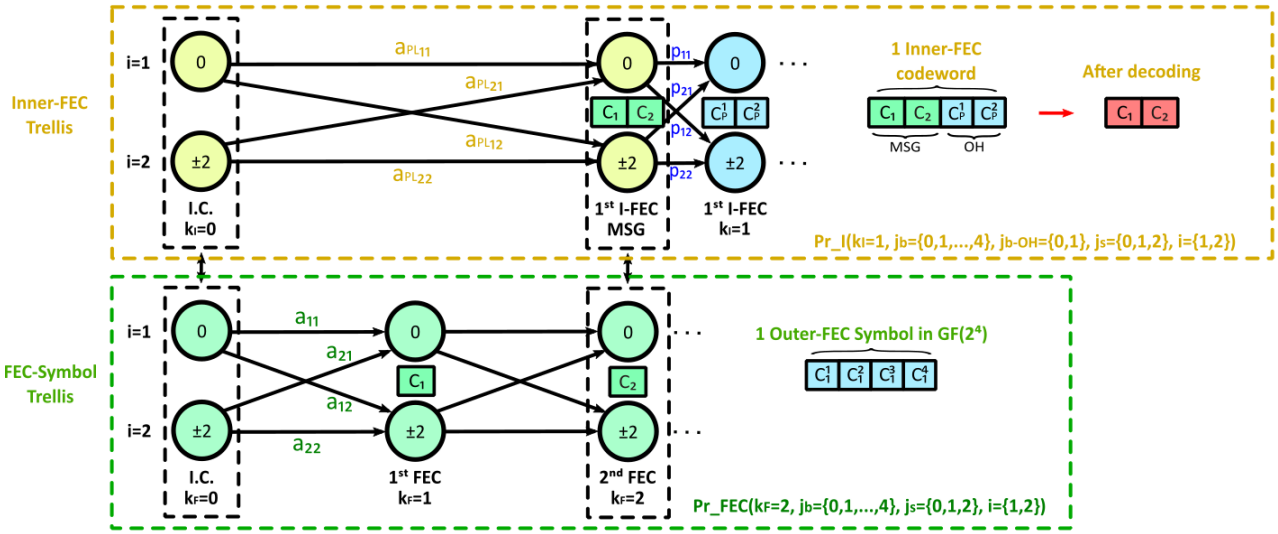


Figure 5. Example of an inner-FEC trellis, each codeword consist of 2 outer-FEC symbols in GF(2^4) and a 2-bit overhead for parity checks.

2.1.4 Outer-FEC Trellis and Post-FEC BER Calculation

We use the decoded inner codeword transition probabilities $a_{O\,i'i}^{j_s,j_b}$ to traverse the outer-FEC trellis. In this section, we consider an outer codeword having $n_o \cdot m$ bits that divides into an integer $n_o \cdot m/k_i$ number of inner-FEC payloads. First, we set the initial conditions at $k_O = 0$ to their steady-state values found by Equation 1 and Equation 2. We define $j_{s\text{-}I}$ and $j_{b\text{-}I}$ as the number of FEC-symbol errors and bit errors that occurred in the decoded inner codeword, respectively. We also denote $Pr\_O_{k_O}^{j_s,j_b}(i)$ the probability of visiting Markov state i at the k_O^{th} stage in the outer-FEC trellis after traversing all trellis paths containing exactly j_s FEC symbol errors and j_b bit errors. Dynamic programming is used to calculate $Pr\_O_{k_O+1}^{j_s,j_b}(i)$ by iterating over all possible $j_{s\text{-}I}$, $j_{b\text{-}I}$ and i'

$$Pr\_O_{k_O+1}^{j_s,j_b}(i) = \sum_{j_{s-I}} \sum_{j_{b-I}} \sum_{i'} Pr\_O_{k_O}^{j_s-j_{s-I},\,j_b-j_{b-I}}(i')\, a_{O\,i'i}^{j_{s-I},\,j_{b-I}}. \tag{12}$$

We traverse the outer-FEC trellis until the end of the outer codeword ($k_O = n_o m/k_i$) using the dynamic programming procedure described in Equation 12 to obtain $Pr\_O_{n_o m/k_i}^{j_s,j_b}(i)$.

The example in Figure 6 shows an outer-FEC trellis example that consists of two decoded inner codewords. At the end of the outer codeword ($k_O = 2$), up to 1 FEC-symbol error can be corrected by the outer-FEC decoder. The post-FEC BER can be approximated by [1],

$$BER_{post-FEC} \approx \sum_{j_s=t_o+1}^{j_s^{max}} \left(\sum_{jb} \left(\frac{Pr\_O_{k_O=n_o m/k}^{j_s,j_b} \cdot j_b}{n_o \cdot m} \right) \right). \tag{13}$$

At low BER, the probability of having an erroneous outer codeword decreases exponentially with increasing j_s. Pruning trellis paths having negligible probabilities can result in a significant reduction in computational complexity. This pruning is achieved by capping the upper summation limit in Equation 13 to j_s^{max}, indicating that only trellis paths having up to j_s^{max} FEC symbol errors contribute to the post-FEC BER.

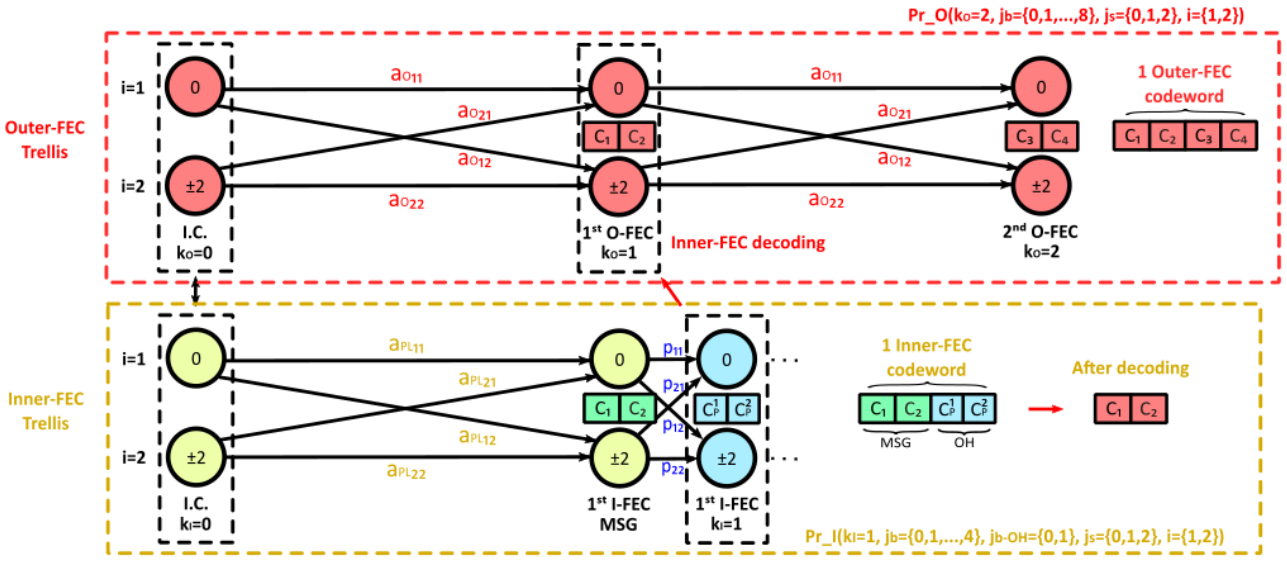


Figure 6. Example of an outer-FEC trellis, state transition a_O in the outer-FEC trellis is generated by applying inner-FEC decoding to a_I.

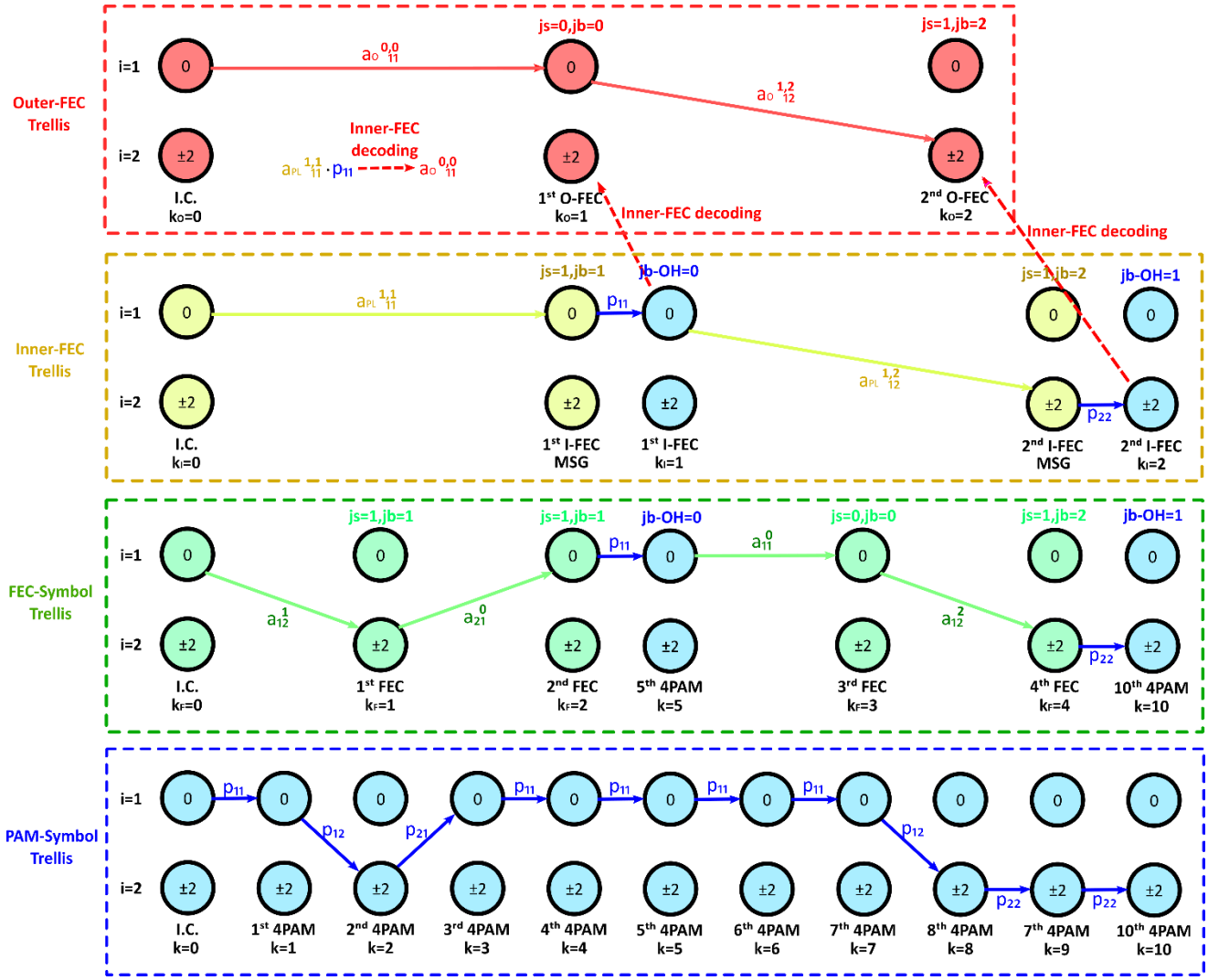


Figure 7. A detailed trellis example of the proposed trellis model described in Figure 2.

2.1.5 Trellis Example of a Concatenated FEC

In Figure 7, we show a detailed trellis example of our proposed trellis model previously described in Figure 2, assuming the same non-binary outer code (2,1,1) in GF(2^4) and the binary inner code (10,8,1). The same trellis path is shown at all four levels of hierarchy: the PAM-symbol trellis, FEC-symbol trellis, inner-FEC trellis and outer-FEC trellis. In the inner-FEC trellis, the first inner codeword has one bit error that appears in the payload ($j_b = 1$), and no bit error in the overhead ($j_{b\text{-}OH} = 0$). The only bit error in the first inner codeword is correctable with $t_i = 1$. After decoding, the first outer-FEC state transition $a_{O}{}_{11}^{0,0}$ is error-free.

In the second inner codeword, two bit errors occur in the payload ($j_b = 2$). Because these two bit errors occur within the same FEC symbol at $k_F = 4$, they only contribute to one FEC symbol error ($j_s = 1$). Also, there is one bit error in the inner-FEC overhead ($j_{b\text{-}OH} = 1$), resulting in a total of three bit errors in the inner codeword, which is uncorrectable by the inner-FEC decoder with $t_i = 1$. After decoding, only the two bit errors from the payload remain in the outer-FEC trellis. Therefore, the second transition $a_{O}{}_{12}^{1,2}$ in the outer-FEC trellis has one FEC-symbol error and two bit errors.

Since all errors in the first inner codeword have been corrected, at $k_O = 2$ we have $j_{s\text{-}I} = 1$ and $j_{b\text{-}I} = 2$. The single FEC symbol error in the outer-FEC codeword is correctable by the outer FEC decoder with $t_o = 1$. As a result, the example trellis path in Figure 7 does not contain any post-FEC bit errors.

2.2 Modeling Inner-FEC Miscorrections

The analysis in Section 2.1 does not consider inner-FEC miscorrections, which significantly impact the accuracy of BER analysis. In this section, we introduce a hybrid approach to accurately model miscorrections, using probabilities extracted from a separate time-domain simulation as part of our proposed statistical model.

The inner codes considered for 200 Gb/s+ wireline transceivers generally have a low Hamming distance. Inner codewords with more than t_i bit errors may be decoded to the wrong codeword. This is known as a miscorrection, which may introduce up to t_i new bit errors and up to t_i new FEC-symbol errors at the inner-FEC decoder output. This paper considers two popular candidate codes in IEEE 802.3dj, the extended Hamming (128,120,1) code and the BCH(144,136,1) codes, both with $t_i = 1$.

Recall that the extended Hamming code uses an additional parity bit to increase the minimum hamming distance between valid codewords, eliminating miscorrections in codewords having exactly 2 bit errors. We define b as the maximum number of bit errors an inner codeword can have before decoding that guarantees no miscorrections. Hence, the Hamming (128,120,1) code has $b = 2$. However, the BCH (144,136,1) decoder may miscorrect any codeword with more than one bit error, so it has $b = t_i = 1$.

Figure 8 shows all possible scenarios for inner-FEC decoding with $t_i = 1$. If the total number of bit errors in the codeword $j_b + j_{b\text{-}OH} \leq t_i$, the decoder always corrects all errors in the codeword. With $t_i < j_b + j_{b\text{-}OH} \leq b$, the codeword is not correctable, but a miscorrection does not occur, so j_s and j_b are preserved after decoding. If $j_b + j_{b\text{-}OH} > b$, the codeword is not correctable and a miscorrection is possible. In this case, the probability that no miscorrection occurs is denoted P_X. The probability that a miscorrection occurs, and the additional bit error corrupts a FEC symbol that already contains a bit error, increasing only j_b by 1, is denoted P_Y. The probability that a miscorrection occurs and the additional bit error corrupts a previously error-free outer-FEC symbol, increasing both j_s and j_b by 1, is denoted P_Z.

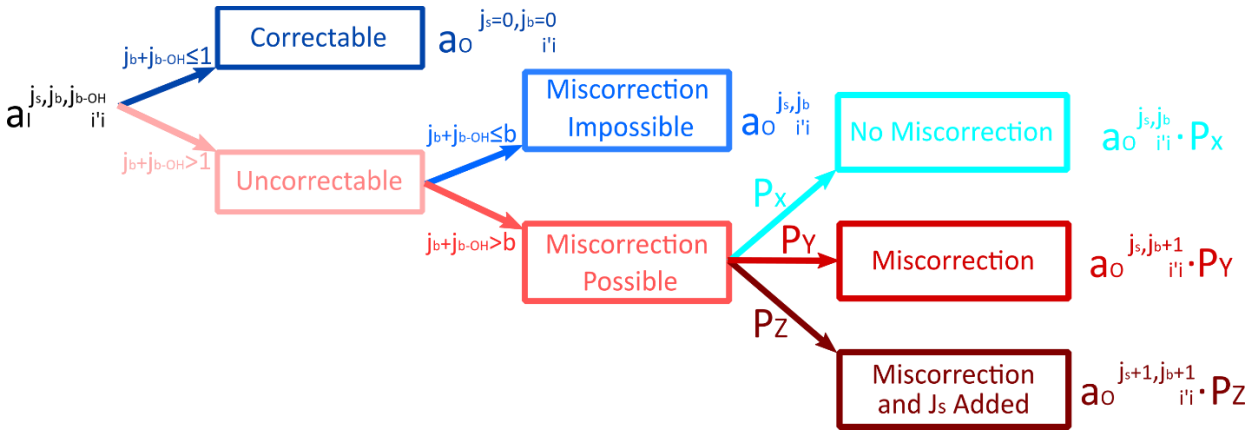


Figure 8. All possible scenarios for inner-FEC decoding with $t_i = 1$.

We extract the probability of P_X, P_Y and P_Z using a time-domain simulation. The time-domain model has the identical channel response, DFE tap weight, and noise sources considered in the statistical model. As P_Z is the probability of an inner codeword miscorrection with a new FEC-symbol error added to the decoded codeword, given that $j_b + j_{b\text{-}OH} > b$ in an inner codeword,

$$P_Z = \frac{Number\ of\ simulated\ inner{-}FEC\ codeword\ miscorrections\ with\ j_s\ added}{Number\ of\ simulated{-}inner\ FEC\ codewords\ with\ j_{b-PL}+j_{b-OH}>b}. \quad (14)$$

Since P_Y is the probability of an inner codeword miscorrection without additional FEC-symbol error added to the decoded codeword, given that $j_b + j_{b\text{-}OH} > d_i$ in an inner codeword,

$$P_Y = \frac{Number\ of\ simulated\ inner{-}FEC\ codeword\ miscorrections\ without\ j_s\ added}{Number\ of\ simulated\ inner{-}FEC\ codewords\ with\ j_{b-PL}+j_{b-OH}>b}. \quad (15)$$

Lastly, as P_X, P_Y and P_Z are mutually exclusive, these probabilities must add up to 1, and

$$\mathrm{P}_X = 1 - P_Y - P_Z. \quad (16)$$

The inner-FEC miscorrection in the trellis can be modeled by modifying the inner-FEC decoding procedure described by Equation 11. With miscorrection, an uncorrectable outer-FEC transition probabilities $a_{I_{i'i}}^{js,jb,jb_{OH}}$ is split into the 3 subcases described by Equation 14-16. For non-correctable transitions with $j_b + j_{b\text{-}OH} > b$, we have

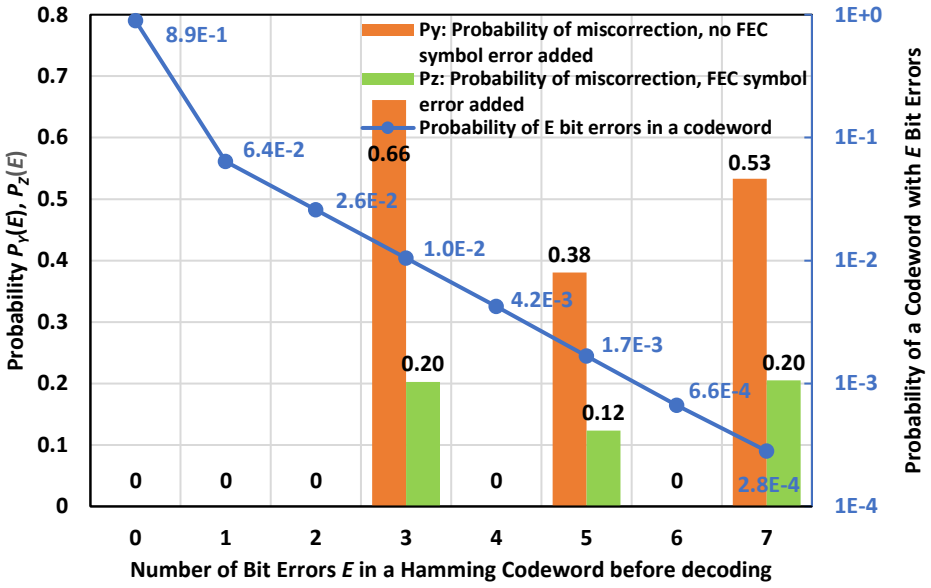
$$\begin{aligned} a_{O_{i'i}}^{j_b,j_s} &= a_{O_{i'i}}^{j_b,j_s} + a_{I_{i'i}}^{js,jb,jb_{OH}} \cdot P_X, \\ a_{O_{i'i}}^{j_b+1,j_s} &= a_{O_{i'i}}^{j_b+1,j_s} + a_{I_{i'i}}^{js,jb,jb_{OH}} \cdot P_Y, \\ a_{O_{i'i}}^{j_b+1,j_s+1} &= a_{O_{i'i}}^{j_b+1,j_s+1} + a_{I_{i'i}}^{js,jb,jb_{OH}} \cdot P_Z. \end{aligned} \quad (17)$$

Our proposed approach assumes that the probabilities P_X, P_Y and P_Z are constant values which are not a function of the number of bit errors in a codeword. For a given inner code, these events can be determined uniquely for every possible error pattern in the trellis, and they do show correlation to the number of existing bit errors in an inner codeword. However, as the number of bit errors per inner codeword increases, their probability of occurring decreases exponentially. As a result, our simplification in modeling P_X, P_Y and P_Z has negligible impact on the post-FEC BER.

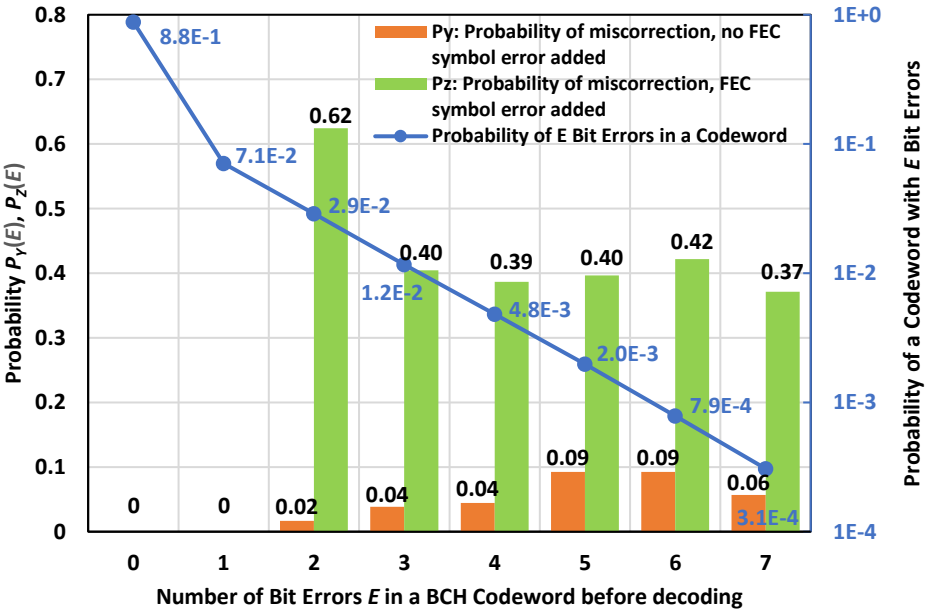
Figure 9 shows the probability $P_Y(E)$ and $P_Z(E)$ as a function of the number of bit errors E in an inner codeword before decoding, and $E = j_b + j_{b\text{-}OH}$. Subplots (a) and (b) in Figure 9 report $P_Y(E)$ and $P_Z(E)$ with the Hamming (128,120,1) code and the BCH (144,136,1) code, respectively. The reported data is generated from a time-domain simulation that observed over one million codewords transmitted with a pre-FEC BER at 10^{-3}, assuming a channel response $h = 1 + 0.5z^{-1}$ and a zero-forcing 1-tap DFE. In each subplot, we also superimpose the probability of having exactly E bit errors in an inner codeword using a solid blue line. Our least-confident data in these figures is for having 7 bit errors in a codeword. In both subplots, we observed over 400 codewords containing 7 bit errors. The miscorrection probabilities depend on the number of bit errors, but as E increases its exponentially decaying probability of occurrence allows us to approximate $P_Y(E)$ and $P_Z(E)$ using a constant value defined by Equation 14 and 15. Note that with the BCH code, miscorrections happen for all codewords having more than 1 bit error and the additional bit error is likely to corrupt an error-free FEC symbol. However, with the extended Hamming code, miscorrections only happen for odd numbers of errors that are greater than 2, and a large portion of the bit errors produced by miscorrections appear in the existing erred FEC symbols. The decoding behavior of the extended

Hamming code can be easily captured by altering the condition for triggering Equation 17. That is, Equation 17 is only updated when $j_b + j_{b\text{-}OH} > b$ and $j_b + j_{b\text{-}OH}$ is an odd number.

Note that a FEC code can have different implementations depending on the parity check matrix used. These implementations affect the probability and location of miscorrections. It was shown that triple bit error miscorrections in extended hamming codes can be minimized by choosing an optimal parity-check matrix [10]. The plots in Figure 9 are only for one choice of BCH(144,136,1) and extended Hamming(128,120,1) codes.



(a) with extended Hamming (128,120,1) code



(b) with BCH (144,136,1) code

Figure 9. P_Y and P_Z as a function of the number of bit errors E in an inner codeword, simulated at 10^{-3} pre-FEC BER with a zero-forcing 1-tap DFE.

2.3 FEC Symbols Divided Between Inner-Codeword Payloads

In Section 2.1.3, we assumed that the inner-FEC payload can be evenly divided into an integer number of outer-FEC symbols. This is true for the Hamming (128,120,1) + RS KP4 concatenated code: the 120-bit payload is divided evenly into 12 RS FEC symbols in GF(2^{10}). However, this is not the case for the BCH(144,136,1) + RS KP4 code. Figure 10 shows the division of outer-FEC symbols in inner-FEC payloads for a concatenated BCH (144,136,1) + RS KP4 FEC. The 14th FEC symbol is divided into 2 inner-FEC payloads. Because of this, we cannot use the same one-step transition probability $a^{j}_{i'i}$ calculated in Equation 5 for the 14th FEC symbol. We modify the approach shown in Section 2.1.3 to create a new transition probability $a_{last}{}^{j}_{i'i}$ representing the last 6 bits in the BCH payload, or equivalently 3 4-PAM symbols. $a_{last}{}^{j}_{i'i}$ can be calculated by

$$a_{last}{}^{j}_{i'i} = Pr_3^j(i)\big|_{Pr_0^0(i'=1)}. \tag{18}$$

When traversing the first inner codeword, $a^{j}_{i'i}$ is used for the first 13 FEC symbols, and $a_{last}{}^{j}_{i'i}$ is used for the 14th FEC symbol.

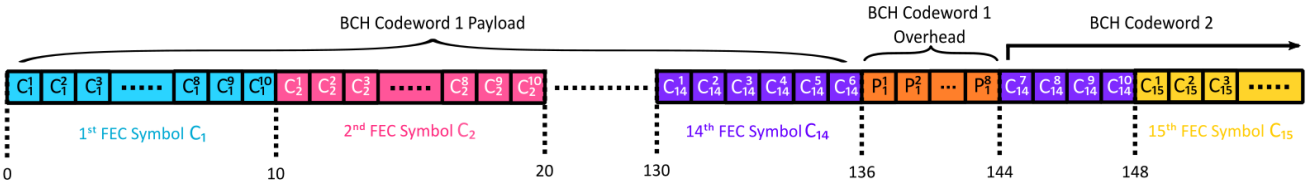


Figure 10. Division of outer-FEC symbols in inner-FEC payloads for a KP4 + BCH (144,136,1) concatenated FEC.

With the BCH (144,136,1) code, the distribution of the FEC symbols in the inner-FEC payloads is not the same for all inner codewords. Only the first 6 bits in the 14th FEC symbol is contained in the first inner-FEC payload, and so the second inner-FEC payload starts with the remaining 4 bits of the 14th FEC symbol. Precisely modeling the division of FEC symbols only results in a very small increase in model accuracy at the cost of higher computational complexity. To simplify the analysis, we assume the same distribution of FEC symbols shown in Figure 10 applies to all inner codewords.

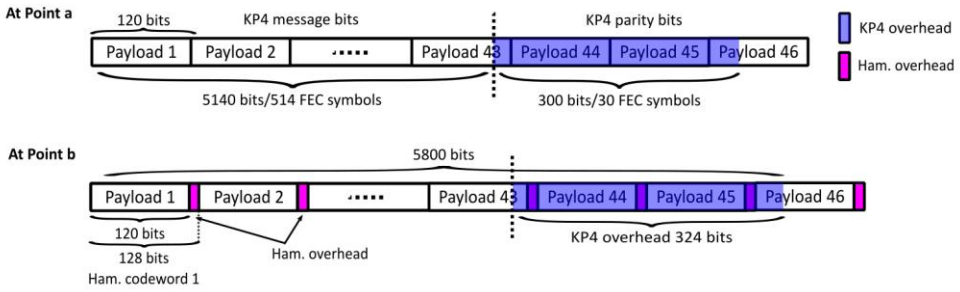


Figure 11. Division of an outer codeword into inner codewords in a KP4 + Hamming (128,120,1) concatenated FEC.

2.4 Inner Codewords Divided Between Outer Codewords

In Section 2.1.4, we assumed an outer codeword having $n_o \cdot m$ bits that divides into an integer $n_o \cdot m/k_i$ number of inner-FEC payloads. Figure 11 shows an example of a concatenated FEC with RS KP4 (544,514,15) outer code and the Hamming (128,120,1) inner code. At the KP4 FEC encoder output, as indicated by point (a) in Figure 1, 30 parity-checking FEC symbols are appended to the 5140-bit message forming one encoded KP4 codeword. The entire KP4-encoded codeword is divided by the 120-bit inner-FEC payloads. Each of the payloads is serially encoded by the extended-Hamming

encoder generating inner codewords as shown in Figure 11 at point (b) of Figure 1. However, in this example, $n_o \cdot m/k_i = 45.33$ is not an integer. In Figure 11, the 46th payload is divided between two neighboring KP4 codewords, with 40 payload bits in the first KP4 codeword and the remaining 80 payload bits in the next KP4 codeword.

When traversing the outer-FEC trellis, as the 46th inner-FEC payload does not fit entirely into the first KP4 codeword, we cannot use the same one-step transition probability $a_{O\,i'i}^{j_s,j_b}$ described in Section 2.1.3 to model the last state transition. We modify the approach to stop accumulating the bit error and FEC symbol errors when the end of the outer codeword is reached. The remaining bits that fall outside the codeword still matter when deciding if the inner codeword is correctable. To capture this, we consider this codeword equivalent to a 40-bit payload with the remaining 80 bits in the payload and 8 overhead bits lumped together to form one 88-bit overhead. As such, we follow the same process described in Section 2.1.3 to model the transition probability of this last codeword as $a_{O,last}$. When traversing the outer-FEC trellis, a_O is used for the first 45 inner codeword transitions, and $a_{O,last}$ is used for the 46th inner codeword.

With the KP4 + Hamming (128,120,1) concatenated FEC, the distribution of the inner codewords in the outer codewords is not the same for all outer codewords. The outer-FEC codeword contains only the first 40 bits of the 46th inner codeword, and so the next outer codeword starts with the last 80 bits of the 46th inner codeword. Modeling these divided inner codewords has little impact on the accuracy of the statistical model while significantly increasing the computational complexity. To simplify the analysis, we assume the same distribution of inner codewords shown in Figure 11 applies to all outer codewords.

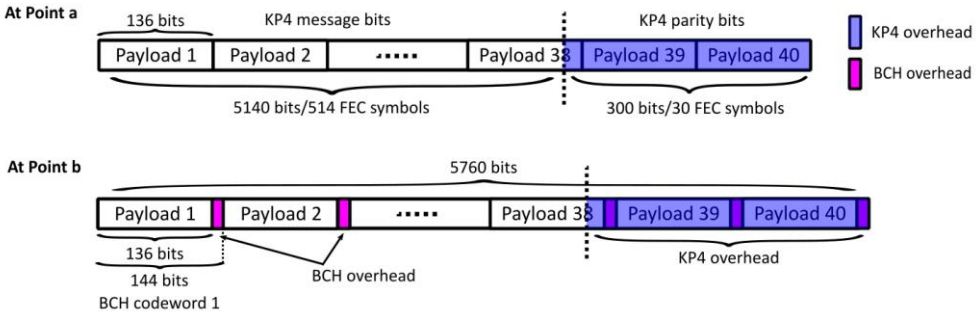


Figure 12. Division of an outer codeword into inner codewords in a KP4 + BCH(144,136,1) concatenated FEC.

Figure 12 shows an example of a concatenated FEC with RS KP4 (544,514,15) outer code and BCH (144,136,1) inner code. The length of the outer codeword is divided evenly into 40 inner-FEC payloads; hence the methodology for divided inner codewords described in this section does not apply to this case.

2.5 Simulation Results

Our 4-PAM statistical model is applied to links with two different concatenated FEC codes. In both cases, the outer code is the standard RS KP4 (544,514,15) FEC. The inner codes are the extended Hamming (128,120,1) reported in Figure 13 and BCH (144,136,1) reported in Figure 14. Each concatenated FEC code is simulated with (1) an all-pass channel response $h = 1$ without DFE and (2) $h = 1 + 0.5z^{-1}$ with a zero-forcing 1-tap DFE. The links are subject to AWGN only. In both figures, we

plot the post-FEC vs. pre-FEC BER curve of the two channels with and without inner-FEC miscorrections using the statistical model. Results generated by a time-domain behavioral model are superimposed in each figure. All time-domain data points in Figure 13 and Figure 14 are simulated down to a post-FEC BER of 10^{-8}. To simulate the post-FEC BERs without miscorrections, we assign P_Y=P_Z=0 in the statistical model, and an ideal inner-FEC decoder is used in the time-domain model. Good consistency is observed between the statistical and time-domain results, both with and without miscorrections. Moreover, the Hamming (128,120,1) code not only outperforms its BCH counterpart in both channel cases but also exhibits a smaller gap between simulations with and without miscorrections. This observation is consistent with the findings reported in Figure 9, that a Hamming decoder is much less likely to add to another FEC-symbol error in a KP4 codeword.

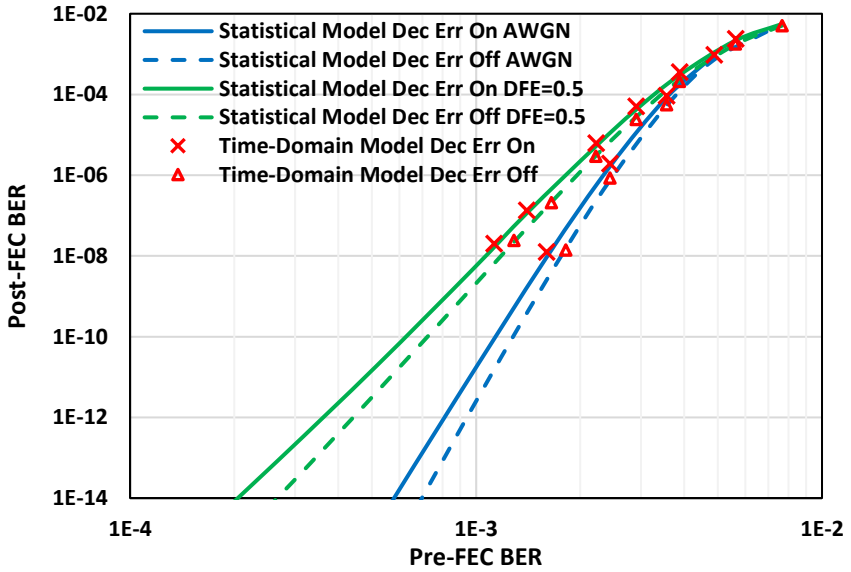


Figure 13. Simulation results of a KP4 + Hamming (128,120,1) concatenated FEC.

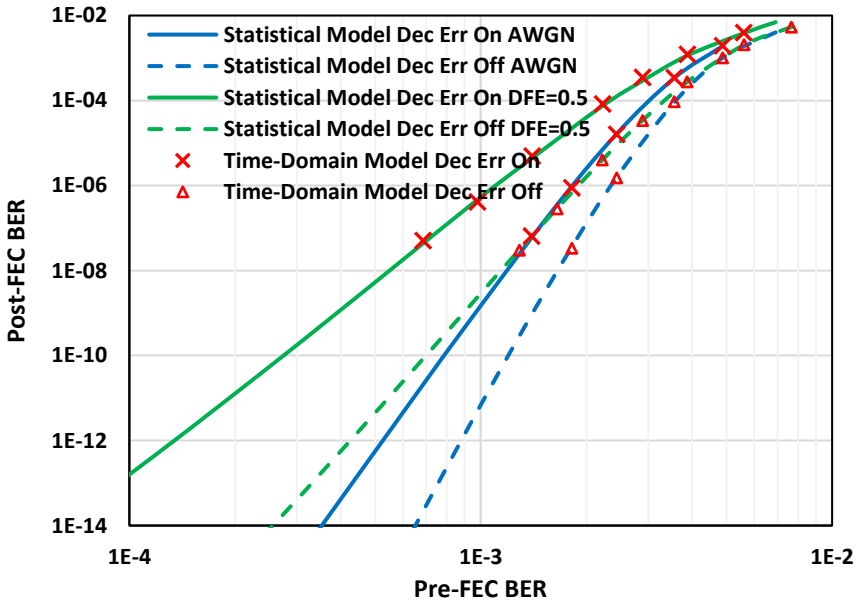


Figure 14. Simulation results of a KP4 + BCH(144,136,1) concatenated FEC.

3. Modeling Inner-FEC Interleaving

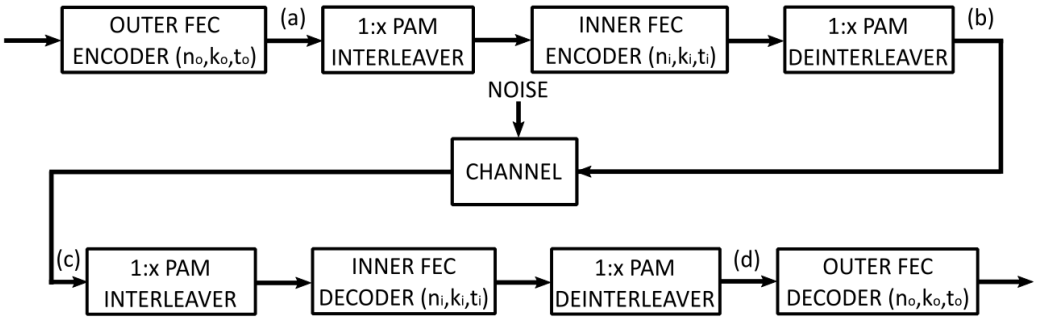


Figure 15. System-level diagram of serially-concatenated FEC with 1:*x* inner-FEC interleaving.

Inner-FEC interleaving is a known technique to protect coding gain in the presence of burst errors. The additional errors introduced by the inner-FEC miscorrection may corrupt an error-free outer-FEC symbol, lowering the coding gain of the concatenated FEC. Figure 15 shows a system-level diagram of a concatenated FEC with 1:*x* inner-FEC interleaving. After the data is encoded by the outer code, the PAM-symbol interleaver divides PAM symbols into *x* streams in a round-robin fashion. Each stream is encoded by a separate inner-FEC encoder before being recombined in the same order by the PAM-symbol deinterleaver. At the receiver shown at point (c), the signal flow is reversed to retrieve and decode the inner codewords.

Inner-FEC interleaving distributes consecutive burst errors to different inner codewords. This allows burst errors that would result in a miscorrection with no inner-FEC interleaving to become correctable or miscorrection-free by satisfying $j_b + j_{b\text{-}OH} \leq b$. Consider a burst of three consecutive 4-PAM errors, each containing 1 bit error, and the extended Hamming inner code with $t_i = 1$ and $b = 2$. Without interleaving, the burst is neither correctable nor decoding-error-free. A likely miscorrection may corrupt an additional outer-FEC symbol, reducing the chance of successfully decoding the outer codeword. With 1:2 inner-FEC interleaving, one of the three errors is distributed to one inner codeword, and the other two are distributed to another inner codeword. Assuming the rest of the decisions in both interleaved codewords are error-free, inner-FEC interleaving avoids a miscorrection in this example.

3.1 Transceiver Architecture of 1:2 Inner-FEC Interleaving

Figure 16 shows the data stream of a concatenated KP4 + Hamming(128,120,1) FEC with 1:2 PAM-symbol interleaving on the inner code. C_n in each block represents the n^{th} outer-FEC symbol. These outer-FEC symbols contain C_n^1- C_n^{10} representing the 10 bits in each outer-FEC symbol over GF(2^{10}). Bits are grouped into pairs to form 4-PAM symbols. For example, C_n^1- C_n^2 make up the 1st PAM symbol in the n^{th} FEC symbol.

After the PAM-symbol interleaver, the data is divided into two streams. Due to a round-robin distribution, the top stream contains the 1st PAM symbol, the bottom stream contains the 2nd, the top stream contains the 3rd, and so on. Figure 16 shows the distribution of the first 24 outer-FEC symbols in an encoded KP4 codeword. These make up the two 120-bit inner-FEC payloads that are encoded by the two extended Hamming encoders to produce two inner codewords. This section refers to these codewords as “inner codeword 1” and “inner codeword 2”. Both codewords have eight parity bits

added to the payload after encoding. The n^{th} parity bit is labelled P_1^n and P_2^n in inner codeword 1 and 2, respectively. The combined data stream appearing at the deinterelaver output is transmitted through the PHY at point (b). This bit stream in the PHY is the same as the order at point (a) in the encoded KP4 codeword, with inner-FEC parity bits added. After transmission through the PHY, the data reaches point (c) at the receiver, and the whole decoding process shown in the bottom-half part of Figure 16 can be seen as the mirrored version of TX encoding. Note that the order of PAM symbols in the encoded KP4 codeword is the same as the PAM symbols transmitted in the PHY. The inner-FEC interleaving scheme is designed to ensure that the burst errors uncorrectable by the inner code corrupt the fewest possible outer-FEC symbols. In other words, the inner-FEC interleaving does not spread out burst errors over more outer-FEC symbols than necessary.

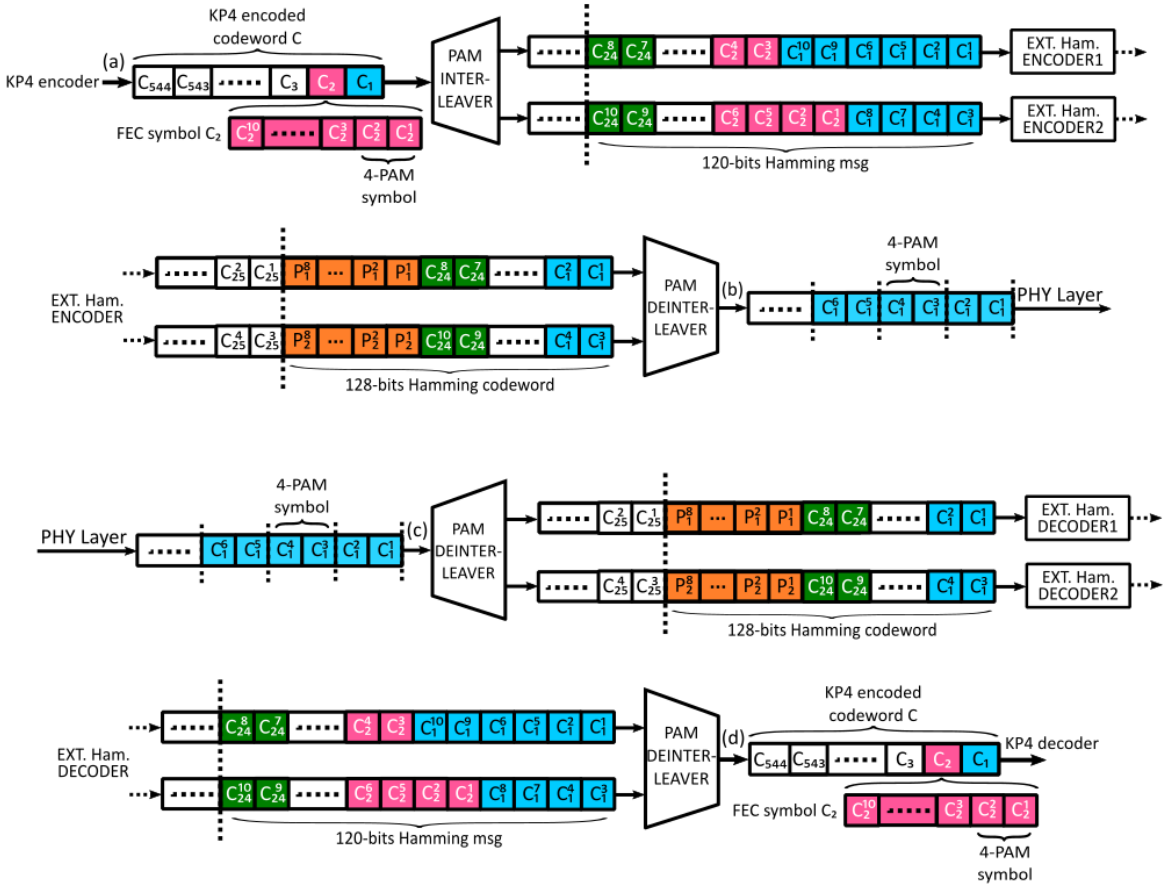


Figure 16. Bit-steam example of a KP4 + Hamming(128,120,1) concatenated FEC with 1:2 inner-FEC interleaving.

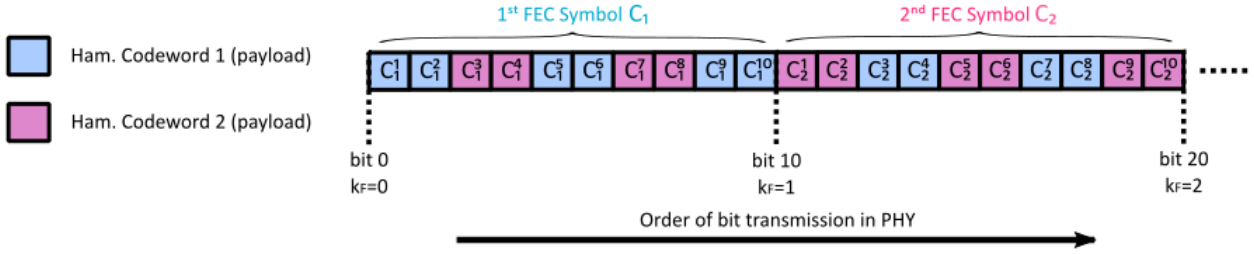


Figure 17. An example of PAM-symbol distribution between 1:2 interleaved inner codewords, resulting in two repeating PAM-symbol patterns in an outer-FEC symbol.

3.2 Trellis Model for Concatenated FEC with 1:2 Inner-FEC Interleaving

In this subsection, we describe how the trellis model for concatenated FEC is modified to model a concatenated FEC with 1:2 inner-FEC interleaving. We continue to use the KP4 + Hamming (128,120,1) code shown in Figure 16 as an example throughout this section.

3.2.1 PAM-symbol and FEC-Symbol Trellis for 1:2 Interleaving

Recall the time-aggregation of PAM symbols described in Section 2.1.2: Every $m/2$ consecutive 4-PAM symbols transmitted in the PHY are grouped into one outer-FEC symbol in GF(2^m). However, the 1:2 interleaving example in Figure 16 shows that the PAM symbols transmitted in the PHY are distributed into different inner codewords. Figure 17 provides a more detailed explanation of the PAM-symbol distribution given in Figure 16. With 1:2 interleaving, the selection of 4-PAM symbols in the PHY alternate between the two encoded inner codewords. We end up with two repeating PAM-symbol patterns in an outer-FEC symbol. The first pattern contains three 4-PAM symbols from inner codeword 1 and two from inner codeword 2. The second pattern contains two 4-PAM symbols from codeword 1 and three from inner codeword 2.

In the PAM-symbol trellis, we introduce two new variables, j_{b1} and j_{b2} to track the number of bit errors in inner codeword 1 and 2, respectively. Dynamic programming is used to find the probability of arriving at state i at timestep k +1 with exactly j_{b1} and j_{b2} bit errors by iterating over all possible j and i',

$$Pr_{k+1}^{j_{b1},j_{b2}}(i) = \begin{cases} \sum_j \sum_{i'} Pr_k^{j_{b1}-j,j_{b2}}(i') p_{i'i}, if\ (k+1)\ odd \\ \sum_j \sum_{i'} Pr_k^{j_{b1},j_{b2}-j}(i') p_{i'i}, if\ (k+1)\ even \end{cases}. \quad (19)$$

We apply time aggregation to find the one-step transition probability in the FEC-symbol trellis. To capture the two repeating PAM-symbol patterns described in Figure 17, we generate two probabilities $a_{odd}{}_{i'i}^{j_{b1},j_{b2}}$ and $a_{even}{}_{i'i}^{j_{b1},j_{b2}}$, the probability of a FEC-symbol transition from state i' to i with j_{b1} and j_{b2} bit errors at a timestep where k_F is odd and even, respectively. We traverse a length-$m/2$ PAM-symbol trellis using Equation 19 to calculate $a_{odd}{}_{i'i}^{j_{b1},j_{b2}}$,

$$a_{odd}{}_{i'i}^{j_{b1},j_{b2}} = Pr_{k=m/2}^{j_{b1},j_{b2}}(i)\Big|_{Pr_0^{0,0}(i')=1}. \quad (20)$$

Next, the one-step transition probability for a_{odd} is generated from a_{even} by swapping j_{b1} and j_{b2},

$$a_{even}{}_{i'i}^{j_{b1},j_{b2}} = a_{odd}{}_{i'i}^{j_{b2},j_{b1}}. \quad (21)$$

In the FEC-symbol trellis, we define variable j_{s1} as the number of outer-FEC-symbol errors containing bit errors only from inner codeword 1. Similarly, j_{s2} is defined as the number of outer-FEC-symbol errors with bit errors occurring only in inner codeword 2. j_{s12} is defined as the number of outer-FEC-symbol errors with bit errors in both inner codewords. Note that each FEC-symbol error is classified into one of three mutually exclusive cases j_{s1}, j_{s2}, or j_{s12}, so the total number of FEC-symbol errors in a trellis path is $j_{s1}+j_{s2}+j_{s12}$. These variables are defined in this manner to easily determine how inner-FEC correction affects the number of outer-FEC symbol errors in a trellis path. For example, if codeword 1 is correctable but codeword 2 is not, j_{s1} is the number of FEC symbol errors that can be corrected by inner codeword 1. The numbers of bit errors in inner codeword 1 and 2 at the $(k_F+1)^{th}$ FEC symbol are denoted as j_1, and j_2, respectively. The probability of arriving at state i at the

$(k_F+1)^{th}$ FEC symbol with exactly j_{s1}, j_{s2}, and j_{s12} FEC-symbol errors, and j_{b1} and j_{b2} bit errors is computed by iterating over all possible j_1, j_2 and i',

$$Pr\_FEC_{k_F+1}^{j_{s1},j_{s2},j_{s12},j_{b1},j_{b2}}(i) = \begin{cases} \sum_{j_1}\sum_{j_2}\sum_{i'} Pr\_FEC_{k_F}^{j_{s1}-1,j_{s2},j_{s12},j_{b1}-j_1,j_{b2}-j_2}(i')\, a_{odd}{}_{i'i}^{j_1,j_2}, if\ j_1>0, j_2=0, (k_F+1)\ odd \\ \sum_{j_1}\sum_{j_2}\sum_{i'} Pr\_FEC_{k_F}^{j_{s1},j_{s2}-1,j_{s12},j_{b1}-j_1,j_{b2}-j_2}(i')\, a_{odd}{}_{i'i}^{j_1,j_2}, if\ j_1=0, j_2>0, (k_F+1)\ odd \\ \sum_{j_1}\sum_{j_2}\sum_{i'} Pr\_FEC_{k_F}^{j_{s1},j_{s2},j_{s12}-1,j_{b1}-j_1,j_{b2}-j_2}(i')\, a_{odd}{}_{i'i}^{j_1,j_2}, if\ j_1>0, j_2>0, (k_F+1)\ odd \\ \sum_{j_1}\sum_{j_2}\sum_{i'} Pr\_FEC_{k_F}^{j_{s1}-1,j_{s2},j_{s12},j_{b1}-j_1,j_{b2}-j_2}(i')\, a_{even}{}_{i'i}^{j_1,j_2}, if\ j_1>0, j_2=0, (k_F+1)\ even \\ \sum_{j_1}\sum_{j_2}\sum_{i'} Pr\_FEC_{k_F}^{j_{s1},j_{s2}-1,j_{s12},j_{b1}-j_1,j_{b2}-j_2}(i')\, a_{even}{}_{i'i}^{j_1,j_2}, if\ j_1=0, j_2>0, (k_F+1)\ even \\ \sum_{j_1}\sum_{j_2}\sum_{i'} Pr\_FEC_{k_F}^{j_{s1},j_{s2},j_{s12}-1,j_{b1}-j_1,j_{b2}-j_2}(i')\, a_{even}{}_{i'i}^{j_1,j_2}, if\ j_1>0, j_2>0, (k_F+1)\ even \end{cases}. \quad (22)$$

3.2.2 Inner-FEC Trellis for 1:2 Interleaving

With 1:2 inner-FEC interleaving, every $2n_i$ bits must be traversed to reach the end of both inner codewords. Figure 18 shows an inner-FEC trellis example of 1:2 interleaved Hamming(128,120,1) codewords where 256 bits are traversed as the basic building block to compute a_I, the one-step transition probability of the two inner codewords before decoding.

In the inner-FEC trellis, $j_{b\text{-}PL1}$ and $j_{b\text{-}PL2}$ are defined as the number of bit errors in the payload of inner codeword 1 and 2, respectively. We traverse the FEC-symbol trellis up to $k_F = k_i/m$ using Equation 22 to find a_{PL},

$$a_{PL}{}_{i'i}^{j_{s1},j_{s2},j_{s12},j_{b-PL1},j_{b-PL2}} = Pr\_FEC_{k_F=k_i/m}^{j_{s1},j_{s2},j_{s12},j_{b-PL1},j_{b-PL2}}(i)\Big|_{Pr_{k_F=0}^{0,0,0,0,0}(i')=1}. \quad (23)$$

We define $j_{b\text{-}OH1}$ and $j_{b\text{-}OH2}$ as the number of bit errors in the overheads of inner codewords 1 and 2, respectively. We traverse the PAM-symbol trellis representing the overhead to $k = 2(n_i\text{-}k_i)$ using Equation 19 to find a_{OH}, which is the one-step transition probability for the overhead in the 2 inner codewords,

$$a_{OH}{}_{i'i}^{j_{b1-OH},j_{b2-OH}} = Pr_{k=2(n_i-k_i)}^{j_{b1-OH},j_{b2-OH}}(i)\Big|_{Pr_0^{0,0}(i')=1}. \quad (24)$$

a_{PL} and a_{OH} are aggregated to produce a_I, the one-step transition probability for the two inner codewords without correction,

$$a_I{}_{i'i}^{j_{s1},j_{s2},j_{s12},j_{b-PL1},j_{b-PL2},j_{b-OH1},j_{b-OH2}} = \sum_{i\sim}(a_{PL}{}_{i'i\sim}^{j_{s1},j_{s2},j_{s12},j_{b-PL1},j_{b-PL2}}\, a_{OH}{}_{i\sim i}^{j_{b1-OH},j_{b2-OH}}). \quad (25)$$

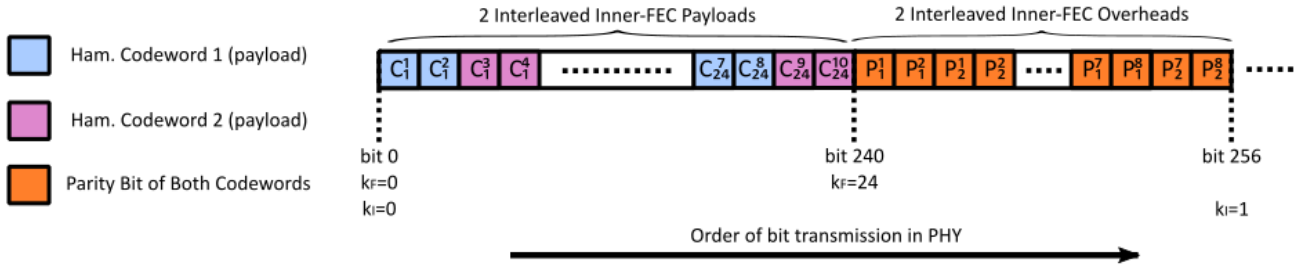


Figure 18. Example of 1:2 interleaved Hamming (128,120,1) codewords.

In the outer-FEC trellis, each state transition a_O corresponds to traversing two decoded inner codewords with 1:2 inner-FEC interleaving. We iterate over all transition probabilities a_I, to compute $a_{O\,i'i}^{j_s,j_b}$, the one-step transition in the outer-FEC trellis for both inner codewords after decoding. Note that a_O does not track the inner-FEC symbol and bit errors. There are 4 cases for inner-FEC error correction:

1. If $(j_{b\text{-}PL1} + j_{b\text{-}OH1}) \leq t_i$ and $(j_{b\text{-}PL2} + j_{b\text{-}OH2}) \leq t_i$, both inner codeword 1 and 2 are correctable, and the transition probability a_I is decoded to the outer-FEC transition a_O with $j_s = 0$ and $j_b = 0$:

$$a_{O\,i'i}^{0,0} = a_{O\,i'i}^{0,0} + a_{I\,i'i}^{j_{s1},j_{s2},j_{s12},j_{b-PL1},j_{b-PL2},j_{b-OH1},j_{b-OH2}}. \quad (26)$$

2. If $(j_{b\text{-}PL1} + j_{b\text{-}OH1}) \leq t_i$ and $(j_{b\text{-}PL2} + j_{b\text{-}OH2}) > t_i$, only inner codeword 1 is correctable. Accordingly, a_O contains only the bit errors $j_{b\text{-}PL2}$ in inner-FEC payload 2. Only the FEC symbol errors j_{s1} are corrected, and so a_O contains $j_{s2} + j_{s12}$ FEC symbol errors:

$$a_{O\,i'i}^{j_{s12}+j_{s2},j_{b-PL2}} = a_{O\,i'i}^{j_{s12}+j_{s2},j_{b-PL2}} + a_{I\,i'i}^{j_{s1},j_{s2},j_{s12},j_{b-PL1},j_{b-PL2},j_{b-OH1},j_{b-OH2}}. \quad (27)$$

3. If $(j_{b\text{-}PL1} + j_{b\text{-}OH1}) > t_i$ and $(j_{b\text{-}PL2} + j_{b\text{-}OH2}) \leq t_i$, only inner codeword 2 is correctable:

$$a_{O\,i'i}^{j_{s12}+j_{s1},j_{b-PL1}} = a_{O\,i'i}^{j_{s12}+j_{s1},j_{b-PL1}} + a_{I\,i'i}^{j_{s1},j_{s2},j_{s12},j_{b-PL1},j_{b-PL2},j_{b-OH1},j_{b-OH2}}. \quad (28)$$

4. If $(j_{b\text{-}PL1} + j_{b\text{-}OH1}) > t_i$ and $(j_{b\text{-}PL2} + j_{b\text{-}OH2}) > t_i$, neither codeword 1 nor 2 is correctable. In this case, a_O contains all FEC-symbol and bit errors from both inner-FEC payloads:

$$a_{O\,i'i}^{j_{s12}+j_{s1}+j_{s2},j_{b-PL1}+j_{b-PL2}} = a_{O\,i'i}^{j_{s12}+j_{s1}+j_{s2},j_{b-PL1}+j_{b-PL2}} + a_{I\,i'i}^{j_{s1},j_{s2},j_{s12},j_{b-PL1},j_{b-PL2},j_{b-OH1},j_{b-OH2}}. \quad (29)$$

After the inner-FEC decoding, the outer-FEC trellis is traversed up to $k_O = n_o m/2k_i$ in the same manner described in Section 2.1.4. The post-FEC BER is obtained in the same way also described in Section 2.1.4. Inner-FEC miscorrections, FEC divided between inner codewords, and inner codewords divided between outer codewords, are modeled as described in Sections 2.3-2.5.

3.3 Modeling 1:4 Interleaving

More complex interleaving schemes, such as 1:4 can be modeled using the same methodology as in 1:2 interleaving. However, modeling higher-order interleaving schemes can quickly become prohibitively complex. For example, consider the three error variables used to track FEC-symbol errors in 1:2 inner-FEC interleaving: j_{s1}, j_{s2}, j_{s12}. Extending this methodology to 1:4 interleaving yields a total number of fifteen error variables: 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}. In general, for 1:x interleaving, 2^x-1 variables are needed.

Considering all these variables jointly produces too many transitions to compute. To simplify the computation, we note that some of these variables do not need to be considered jointly for t_i=1. We can partition the correctable subset of trellis paths into mutually exclusive groups, with each error variable in a group containing no more than 1 bit error. We then find the outer-FEC trellis transition probability a_O by separately calculating and deducting the probability of each correctable group from the total probability space $a_{I\,i'i}^{j_s,j_b,j_{b-OH}}$.

For example, if j_{s1}=1 and j_{s12}=1, there is one erred FEC symbol with bit errors only in codeword 1, and another erred FEC symbol with bit errors only in codeword 1 and 2. Therefore, inner codeword 1 is uncorrectable in this example as it has more than 1 bit error. This suggests j_{s1} and j_{s12} are mutually exclusive events in the correctable-trellis subspace. With t_i=1, only two mutually exclusive error scenarios are possible in this example: j_{s1}=1 and j_{s12}=0, or j_{s1}=0 and j_{s12}=1. We refer to these correctable error scenarios as mutually exclusive because one of j_{s1} and j_{s12} must be 0, so they do not need to be considered jointly. However, certain variables need to be considered jointly, for example j_{s1} and j_{s2}. This is because the correctability of FEC symbols counted with j_{s1} and j_{s2} are independent. The errors in all interleaved codewords may be correctable given j_{s1}=1 and j_{s2}=1. We can group the FEC-symbol error variables into the following nine groups where variables in each group need to be considered jointly:

$j_{s1}, j_{s2}, j_{s3}, j_{s4}$	j_{s12}, j_{s34}	j_{s13}, j_{s24}	j_{s14}, j_{s23}	j_{s123}	j_{s124}	j_{s134}	j_{s234}	j_{s1234}

By traversing the inner-FEC trellis nine times, each time tracking and correcting a different group of outer-FEC symbol error variables, we significantly reduce the computation complexity of the trellis model. The maximum number of variables that must be considered jointly with 1:4 interleaving reduces from fifteen to only four.

3.4 Simulation Results

Our 4-PAM statistical model is applied to links with two concatenated FEC codes. The outer code is the standard RS KP4 (544,514,15) FEC in both cases. The inner codes are the extended Hamming (128,120,1) code reported in Figure 19 and the BCH(144,136,1) code reported in Figure 20. Each concatenated FEC code is simulated with a channel response $h = 1 + 0.5z^{-1}$ and a zero-forcing 1-tap DFE. The links are subject to AWGN only. We plot the post-FEC vs. pre-FEC BER curve with no inner-FEC interleaving, 1:2 interleaving, and 1:4 interleaving in each figure. All simulations include inner-FEC miscorrections. Results generated by a time-domain behavioral model are superimposed in each figure. All time-domain data points reported in Figure 19 and Figure 20 are simulated down to a post-FEC BER of 10^{-8}. Good consistency is observed between the statistical results and time-domain results.

Overall, the extended Hamming inner code reported in Figure 19 outperforms the BCH inner code in Figure 20. Note that in Figure 20, higher-order interleaving not only results in an improved post-FEC BER at high pre-FEC BERs, but it also introduces a higher post-FEC error floor at low pre-FEC BERs. Subject to a large DFE tap weight, long burst errors are the dominant source of post-FEC errors at low pre-FEC BERs [1]. Without interleaving, long bursts are likely contained within one inner codeword. In this case, a miscorrection can possibly add one FEC-symbol error to a KP4 codeword. With interleaving, a long burst will corrupt multiple inner codewords, possibly introducing multiple miscorrections and adding more than one FEC-symbol errors to a KP4 codeword. As reported in Figure 9, the extended Hamming (128,120,1) code exhibits much better decoding behavior and adds much fewer FEC-symbol errors than the BCH (144,136,1) code in the presence of long bursts. This explains why we do not observe a similar error floor at 10^{-12} post-FEC BER with the extended Hamming code in Figure 19.

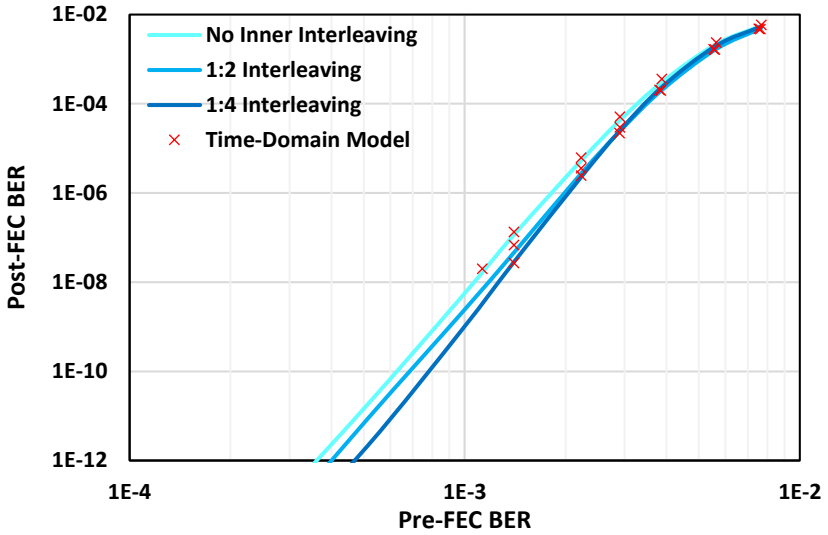


Figure 19. Simulation results of a KP4 + Hamming (128,120,1) concatenated FEC with different interleaving schemes.

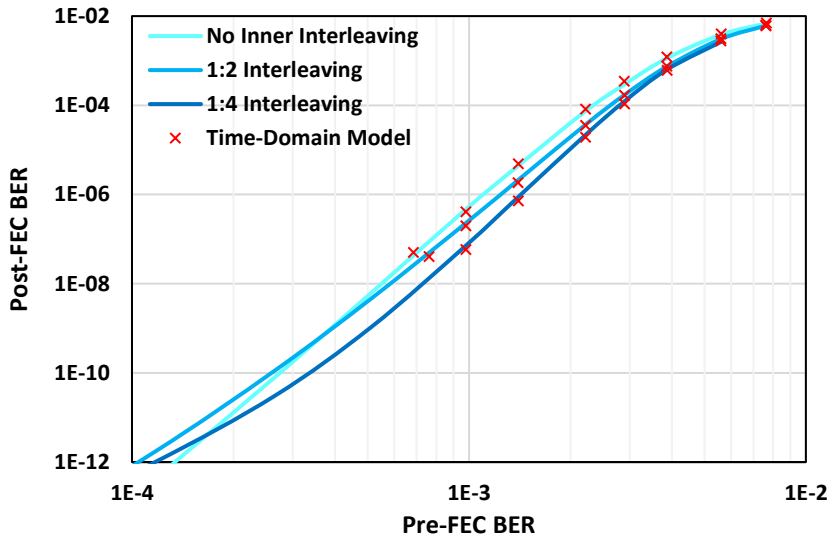


Figure 20. Simulation results of a KP4 + BCH(144,136,1) concatenated FEC with different interleaving schemes.

4. Conclusion

In this paper, we presented a statistical model that accurately estimates post-FEC BER for high-speed wireline links using a concatenated FEC that may include interleaving on the inner code. A hierarchical approach is adopted to analyze the propagation of PAM-symbol and FEC-symbol errors through a 4-layer trellis model in the presence of DFE error propagation. This approach expands on previous work by adding two additional layers of abstraction to model state transitions in inner codewords. A hybrid approach is used to model inner-FEC miscorrections, taking information on the probability of miscorrections determined by a separate time-domain simulation. The proposed statistical model can be combined with our existing approaches to model other noise sources such as

residual ISI, crosstalk, transmitter and receiver jitter, and other system-level design choices, including precoding and bit-multiplexing. One possible future work is to model a concatenated FEC architecture that uses soft decoding.

The proposed model can serve as a tool for evaluating FEC choices for 200+ Gb/s applications. It allows us to compare the effect of different inner/outer codes and inner-FEC interleaving schemes on post-FEC BER. It can also be used as a tool for system-level transceiver design, allowing designers to see the impact of design choices on the post-FEC BER efficiently.

References

[1] M. Yang, S. Shahramian, H. Shakiba, H. Wong, P. Krotnev and A. Chan 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*.

[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] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs, NJ, Prentice Hall, 2004.

[4] 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.

[5] *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.

[6] *Constructing a BCH/Hamming Inner Code for 200 Gb/s per lane PMDs*, IEEE Standard 802.3df, Oct. 2022.

[7] *Concatenated SFEC proposal for 200Gb/s per Lane IM-DD Optical PMD*, IEEE Standard 802.3df, Nov. 2022.

[8] A. Leon-Garcia, *Probability, Statistics, and Random Processes for Electrical Engineering*. Prentice Hall, 2007.

[9] X.-R. Cao, Z. Y. Ren, S. Bhatnagar, M. Fu, and S. Marcus, “A time aggregation approach to Markov decision processes,” *Automatica*, vol. 38, pp. 929–943, 2002.

[10] M. Richter, K. Oberlaender and M. Goessel, "New Linear SEC-DED Codes with Reduced Triple Bit Error Miscorrection Probability," 2008 14th IEEE International On-Line Testing Symposium, Rhodes, Greece, 2008, pp. 37-42, doi: 10.1109/IOLTS.2008.27.