DesignCon 2020

A Statistical Modeling Approach for FEC-Encoded High-Speed Wireline Links

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

Shayan Shahramian, Huawei Canada

Hossein Shakiba, Huawei Canada

Henry Wong, Huawei Canada

Peter Krotnev, Huawei Canada

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

Abstract

This paper presents a statistical modeling approach to accurately estimate post-FEC BER for high-speed wireline links using standard linear block codes, such as the RS(544,514,15) KP4 and RS(528,514,7) KR4 codes. A hierarchical approach is adopted to analyze the propagation of PAM-symbol and FEC-symbol errors through a two-layer Markov model. In this paper, we will turn our proposed BER estimation method into a set of tools to assist in making architectural choices for wireline transceivers, such as co-design of the equalization and FEC in the presence of DFE error propagation and various noise sources including residual ISI, crosstalk, transmitter and receiver jitter.

Authors Biography

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.

Shayan Shahramian received his Ph.D. from the Department of Electrical and Computer Engineering at the University of Toronto, Canada, in 2016. He is the recipient of the NSERC Industrial Postgraduate scholarship in collaboration with Semtech Corporation (Gennum Products). He is the recipient of the best young scientist paper award at ESSCIRC 2014 and received the Analog Devices outstanding designer award for 2014. He joined Huawei Canada in January 2016 and is currently working in system/circuit level design of high-efficiency transceivers for short reach applications.

Hossein Shakiba received his Ph.D. degree in Electrical Engineering from the Department of Electrical and Computer Engineering at the University of Toronto, Canada, in 1997. He has over 30 years of teaching, research, design, and management experience in the area of analog circuit and system design for various applications with focus on wireline communication in both the industry and academia. He is currently working on system and circuit design and development for next generation short reach and high efficiency serial links at Huawei Canada.

Henry Wong received his B.A.Sc. and Ph.D. both in Electrical Engineering and currently he is a Distinguished Engineer in Huawei. His area of R&D interest is in SerDes design, for high-speed interface, optical module and backplane communications. He has worked for Nortel, Cadence, Lucent on high-speed modems, and for Gennum (Semtech) on SerDes, CDR. He joined Huawei in 2013 and currently he is also a manager of SerDes system architecture product development.

Peter Krotnev is a Sr. Principal Engineer at Huawei Technologies, member of the High Speed I/O System Development Team. Peter is responsible for SerDes architecture improvements, electrical specifications, test planning, as well as leading the development of the SerDes tuning and adaptation strategies. As a telecom professional Peter has also worked with STMicroelectronics Inc. on variety of projects and technologies including ADSL, Gigabit Ethernet and High Speed SerDes. As signal integrity expert Peter is also involved in number of patents and papers.

Anthony Chan Carusone received his Ph.D. from the University of Toronto in 2002 and has since been a professor with the Department of Electrical and Computer Engineering at the University of Toronto. He is also an occasional consultant to industry in the areas of integrated circuit design and digital communication. Prof. Chan Carusone co-authored the Best Student Papers at the 2007, 2008 and 2011 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, and the Best Young Scientist Paper at the 2014 European Solid-State Circuits Conference. He also co-authored, along with David Johns and Ken Martin, the 2nd edition of the textbook "Analog Integrated Circuit Design". He was Editor-in-Chief of the IEEE Transactions on Circuits and Systems II: Express Briefs in 2009, an Associate Editor for the IEEE Journal of Solid-State Circuits 2010-2017, and has served on the technical program committees for the Custom Integrated Circuits Conference, and the VLSI Circuits Symposium. He was a Distinguished Lecturer for the IEEE Solid-State Circuits Society 2015-2017 and currently serves as a member of the Technical Program Committee of the International Solid-State Circuits Conference.

1. Introduction

Forward error correction (FEC) codes have become an integral part of many high-speed wireline links at data rates above 25Gb/s. Depending on the equalization techniques used in wireline links, the same pre-FEC BER may result in different post-FEC BER. Decision feedback equalizer (DFE) error propagation and other noise sources such as inter-symbol interference (ISI), crosstalk and jitter can also significantly impact the accuracy of post-FEC BER analysis. Ideally, one may perform a transient simulation to fully capture the characteristics from all noise sources. 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.

This paper presents a statistical modeling approach to accurately estimate post-FEC BER for high-speed wireline links using standard linear block codes, such as the RS(544,514,15) KP4 and RS(528,514,7) KR4 codes. A hierarchical approach is adopted to analyze the propagation of PAM-symbol and FEC-symbol errors through a two-layer Markov model. A series of techniques including state aggregation, time aggregation, state reduction, and dynamic programming are introduced making the time complexity to compute post-FEC BERs below 10^{-15} reasonable. The efficiency of the proposed model allows it to handle a larger state space, more DFE taps, and more sophisticated linear block codes than prior work.

In this paper, we will turn our proposed BER estimation method into a set of tools to assist in making architectural choices for wireline transceivers, such as co-design of the equalization and FEC in the presence of DFE error propagation and various noise sources. First, the impact on FEC performance is investigated using various coding schemes including bit multiplexing, MOD4 precoding and interleaved FEC codes. Behavioral time-domain simulation results are reported along with the statistical results to verify the accuracy of the model. Second, the impact on burst errors and FEC failures is analyzed by considering various noise sources including residual ISI, crosstalk, transmitter and receiver jitter. Specifically, the proposed model will be used to demonstrate how negative residual ISI may significantly impacts DFE error bursts. In addition, a novel statistical ISI analysis method is presented to incorporate transmitter and receiver jitter into the post-FEC BER estimation. The approach can accurately estimate the data-dependent ISI distribution through jittered half-UI pulses that are derived from the standard unit-pulse response. This procedure allows efficient computation of the ISI probability density function in the presence of arbitrary uncorrelated jitter distributions. Lastly, a 4-PAM 60 Gb/s wireline transceiver fabricated in 7 nm FinFET technology is used as a test vehicle to validate our proposed BER estimation methodology. The methodology can accurately predict very low post-FEC BERs ($<10^{-12}$) that are difficult to measure in real-time.

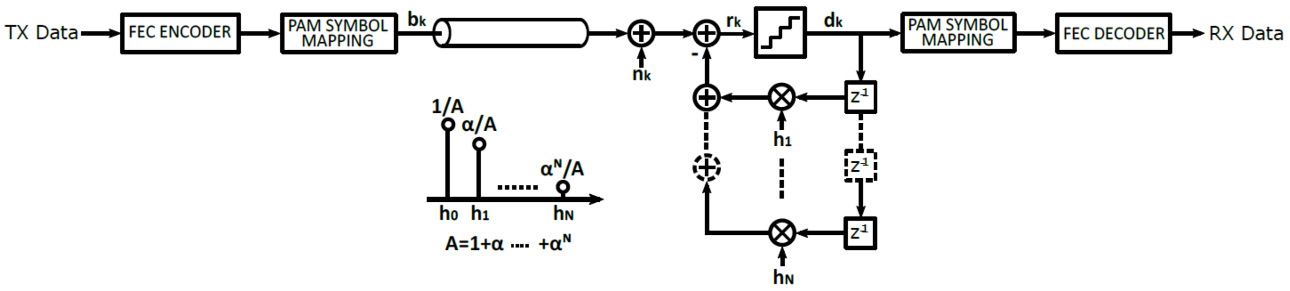


Figure 1. A zero-forcing *N*-tap DFE example for wireline SerDes

2. Modeling DFE Error Propagation in 2-PAM

Consider the link model shown in Figure 1 communicating symbols b_k with time index k. The symbols are filtered by a finite-impulse-response (FIR) channel response h_p with main cursor h_0, and subject to additive noise, n_k. We start by assuming that all pre-cursor and higher-order post-cursor ISIs have been removed by linear equalizers. The detected symbols d_k may differ from the transmitted symbols resulting in the error sequence,

$$D_k = d_k - b_k. \tag{1}$$

This results in an additive error n_k^{dfe} generated by non-zero error terms in the DFE feedback path. Assuming a perfect zero-forcing N-tap DFE,

$$n_k^{dfe} = -\sum_{p=1}^{N} D_{k-p}\, h_p. \tag{2}$$

Then the DFE slicer input r_k becomes

$$r_k = b_k h_0 + n_k + n_k^{dfe}. \tag{3}$$

Error propagation is modeled as a Markov process whose state is specified by the error terms in the DFE feedback, D_{k-1}, $D_{k-2, \ldots}$ Assuming additive white Gaussian noise (AWGN) $n_k \sim \mathrm{N}(0, \sigma^2)$, we have $r_k \sim \mathrm{N}(b_k h_0 + n_k^{dfe}, \sigma^2)$. Hence, the rates at which $d_k \neq b_k$ and $d_k = b_k$ can be determined from the appropriate standard error function. The one-step state-transition probabilities $q_{i'i}$ from a source state ‘i'’ to a sink state ‘i’ can be calculated by applying (3) to each pair of valid transitions $i'i$ in the Markov model, where the term n_k^{dfe} in (3) is exclusively dictated by the source state ‘i'’. With all $q_{i'i}$ calculated, we may find the steady-state probability, π_i, of any state i in the Markov model by solving the global balance equation [1],

$$\pi_i = \sum_{i'} q_{i'i}\pi_{i'}. \tag{4}$$

subject to

$$\sum_i \pi_i = 1. \tag{5}$$

Applying state lumping (sometimes referred as state aggregation) to a Markov process allows the generation of an aggregated chain with a comparatively smaller state space resulting in reduced analytical complexity. The aggregated chain provides a coarser analysis of the state space and can be

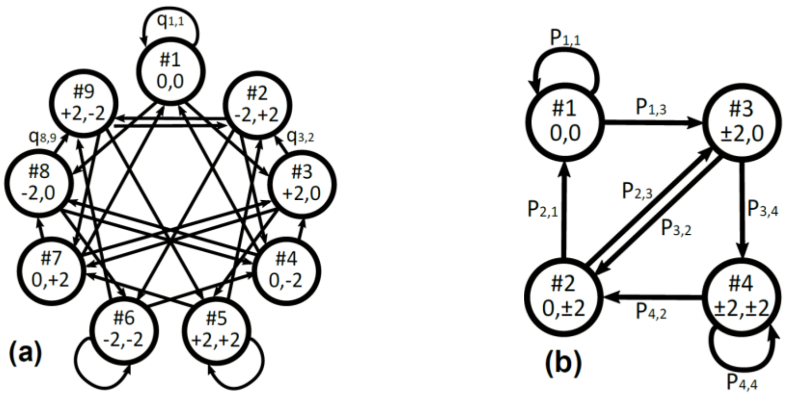


Figure 2. Markov chain model for a 2-tap DFE and 2-PAM symbols $b_k \in \{-1,+1\}$: (a) before lumping [16] (b) after lumping. States are labelled D_{k-1}, D_{k-2}.

used to perform DFE error-rate analysis for the original Markov chain without losing analytical accuracy [2]. For an N-tap DFE with 2-PAM signaling, the original state space $S = \{1,2,\ldots, 3^N\}$ can be reduced to $\bar{S} = \{1,2,\cdots,2^N\}$ using weak lumpability. A 2-tap DFE example is given in Figure 2, and states are labelled according to the errors registered in the DFE: i.e. $<D_{k-1}, D_{k-2}>$. With 2-PAM $b_k = \pm 1$, $D_k \in \{+2, -2, 0\}$ and the DFE may be in $3^2 = 9$ different states as in Figure 2(a). We obtain the $2^2 = 4$ Markov states in Figure 2(b) by lumping all +2 and -2 states at each DFE tap position. The lumped state ±2 preserves the coarser bit-error information by discarding the sign of D_k.

In the scope of this work, we consider the link illustrated in Figure 1 subject to AWGN, having equally spaced DFE slicer thresholds, and an equally probable symbol set b_k that is independent of noise sample n_k. Without the presence of data-dependent residual ISIs, it is proven in [2] that an N-tap DFE Markov process is always lumpable with respect to the partition lumping all states having the same error magnitude $|D_k|$ at each DFE tap.

Denote $p_{i'mim}$ as the one-step state-transition probability from a lumped state 'i'_m' to a lumped state 'i_m'. Transition matrix $P = [p_{i'mim}]$ of the lumped process can be solved using a two-step procedure provided in [3]. First, the aggregated steady-state probabilities Π_{im} can be calculated from the results obtained by (4) and (5),

$$\Pi_{i_m} = \sum\nolimits_{i \epsilon i_m} \pi_i. \tag{6}$$

Next, the aggregated state-transition probabilities $p_{i'mim}$ can be computed by

$$p_{i'_m i_m} = \sum\nolimits_{i' \epsilon i'_m, i \epsilon i_m} \frac{\pi_{i'} q_{i' i}}{\Pi_{i'_m}}. \tag{7}$$

3. 4-PAM Statistical Model for Non-Binary Linear Block Codes

In the previous section, we have reviewed a 2-PAM statistical model to model DFE error propagation. In current long-reach wireline SerDes applications, such as 100GBase-KP4, Gray-coded 4-PAM signaling and RS FEC are standard. For linear FEC codes on GF(2^m), the encoder groups every m bits into one FEC symbol, and correspondingly the decoder can detect and correct up to t erroneous FEC symbols in an n-symbol codeword. All m bit errors in each erred FEC symbol can be corrected so long as the total number of FEC symbol errors does not exceed t. Hence higher-order RS codes provide stronger burst-error correction ability than BCH codes, a measure taken in part to accommodate DFE error propagation. In this section, we extend this statistical model to higher-order M-PAM schemes and linear block FEC codes on GF(2^m), for m being an integer multiple of $\log_2(M)$ including the standardized wireline RS codes.

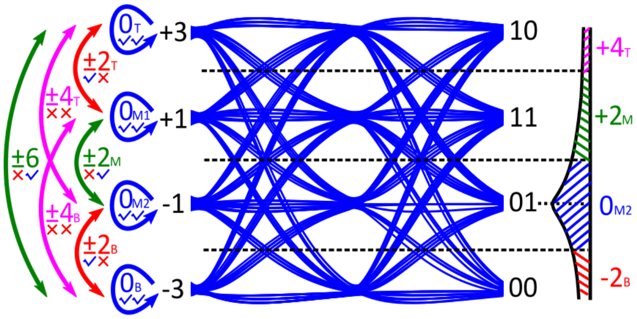


Figure 3. A receiver eye diagram indicating all possible symbol-detection outcomes for a link communicating Gray-coded 4-PAM symbols $b_k \in \{\pm 3, \pm 1\}$.

3.1 4-PAM Markov Model

Figure 3 demonstrates a receiver eye diagram indicating all possible detection outcomes for a link communicating Gray-coded 4-PAM symbols $b_k \in \{\pm3, \pm1\}$. All 16 error values $D_k \in \{0_T, 0_{M1}, 0_{M2}, 0_B, \pm2_T, \pm2_M, \pm2_B, \pm4_T, \pm4_B, \pm6\}$, together with their associated bit-error patterns, are also labeled in the same figure. The subscript of each error value denotes its relative position in the 4-PAM eye from top to bottom. Note that states having the same error value may correspond to different bit-error patterns. For example, subject to an error event $D_k = +2_M$, the 1st bit of the received PAM symbol is in error, which corresponds to the pdf plot superimposed in Figure 3 with $b_k = -1$, $d_k = +1$ and $n_k^{dfe} = 0$. However, the combination of $b_k = +1$ and $d_k = +3$ results in $D_k = +2_T$, which instead makes the 2nd bit erroneous while having the same error value.

Next, in the 16^N-state Markov model, all states having the same error magnitude are aggregated together by applying weak lumpability, resulting in a much smaller 4^N-state state space. Specifically, we can define a new set of $D_k \in \{0, \pm2, \pm4, \pm6\}$ for the 4-PAM example given in Figure 3. Steady-state and state-transition probabilities of the new aggregated chain can be calculated using (6) and (7), similar to what has been done in the 2-PAM case.

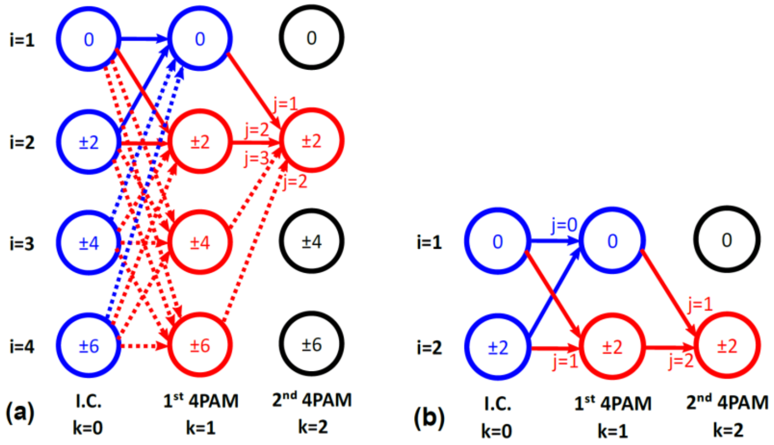


Figure 4. 4-PAM trellis paths for calculating $\sum_j Pr^j{}_2(2)$ with N = 1 and B = 2 using (a) lumped trellis model (b) lumped trellis model ignoring ±4 and ±6 error events.

We next apply trellis-based dynamic programming to the Markov model to efficiently calculate the probability of bit errors in a codeword. The lumped Markov model for an N-tap DFE with M-PAM signaling may be represented by an M^N-state radix-M trellis. 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. The same aggregation procedure is repeated recursively when traversing through each stage in the trellis, resulting in a significant reduction in computational complexity.

When traversing an M-PAM trellis using dynamic programming, each branch decision corresponds to between 0 and at most $\log_2 M$ bit errors. We define j_{PAM} as the number of bit errors in a PAM symbol detection. For example, in a link communicating 4-PAM symbols $b_k \in \{\pm3, \pm1\}$, $j_{PAM} \in \{0, 1, 2\}$ and the receiver error sequence defined in (1) is $D_k \in \{\pm6, \pm4, \pm2, 0\}$. Assuming Gray-coding, an error value ±2 or ±6 corresponds to $j_{PAM} = 1$, whereas an error value ±4 indicates $j_{PAM} = 2$. In each trellis iteration, for states 'i' where the most recently received 4-PAM symbol has j_{PAM}-bit errors,

$$Pr_{k+1}^{j}(i) = \sum_{i'} Pr_{k}^{j-j_{PAM}}(i')p_{i'i}. \tag{8}$$

Figure 4(a) shows an example for a 4-PAM 1-tap-DFE Markov model with $B = 2$, highlighting all possible paths ending in state ±2 ($i = 2$). For example, $Pr_2^j(2)$ represents the probability of arriving at state #2 at the 2nd stage of the trellis having traversed any trellis paths corresponding to exactly j-bit errors, and the highlighted paths in Figure 4(a) indicate all possible error patterns contributing to $\sum_j Pr^j_2(2)$. Hence, from (8) we know $\sum_j Pr^j_2(2)= Pr_1^0(1)p_{12}+Pr_1^1(2)p_{22}+Pr_1^2(3)p_{32}+Pr_1^1(4)p_{42}$, where the only possible node for $k = 1$ and $j = 2$ is #3. Without lumping, the Markov model would have $7^1 = 7$ states for a 4-PAM 1-tap DFE, but it can be reduced to 4 as in Figure 4(a) by lumping the 1-bit errors ±2/±6 and the 2-bit errors ±4. Note that lumping reduces the model's complexity much more as the number of DFE taps increases. Furthermore, the trellis model can be simplified to a 2^N-state radix-2 trellis as demonstrated in Figure 4(b) by ignoring all the dotted paths in Figure 4(a) that have unlikely ±4 and ±6 error events.

3.2 Time-Aggregated FEC Trellis Model

Using the methods described so far, every FEC symbol in GF(2^m) can be decomposed into a length-m/2 4-PAM trellis describing link behavior in the physical layer. Recall the example in Figure 5 that we apply (8) to recursively compute $\mathrm{Pr}_k^j(i)$ in order to aggregate the probability of error patterns having exactly j bit errors, where $j \in \{0 \ldots m/2\}$.

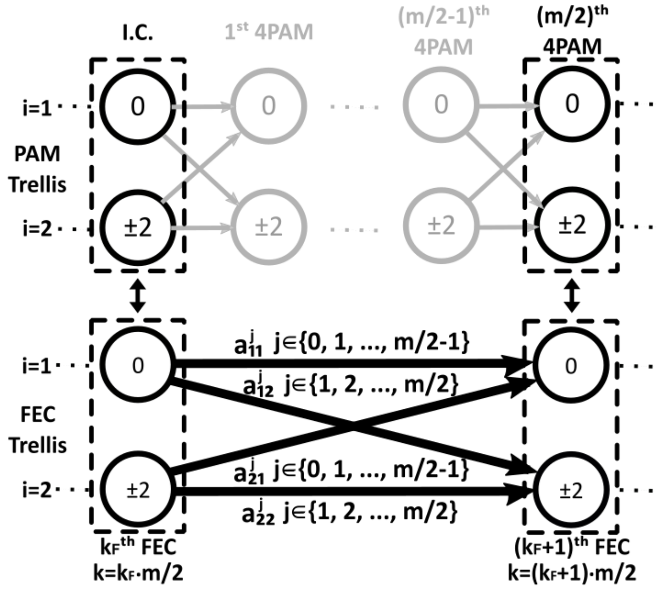


Figure 5. A time-aggregated 4-PAM trellis example with N = 1.

Note that all paths in the trellis representing $Pr^j_k(i)$, the probability of arriving at state i at the kth stage of the trellis after traversing all trellis paths containing exactly j bit errors, can be decomposed into 2^N groups of trellis paths and each starts with one of the 2^N Markov states at $k = 0$. For example, in Figure 4(b) all trellis paths representing $Pr^1_2(2)$ must begin with one of the two DFE states at $k = 0$. As such, we may simplify the entire length-m/2 2^N-state radix-2 trellis to a length-1 2^N-state radix- ($2^N \cdot m/2$) trellis by aggregating all j-bit-error paths within each of the 2^N groups to a one-step direct transition between the two states at $k = 0$ and $k = m/2$. Each one-step transition in the simplified trellis is equivalent to traversing $m/2$ 4-PAM symbols in the fully expanded trellis. Figure 5 shows an example of a time-aggregated 4-PAM trellis with $N = 1$, where we denote $a^j_{i'i}$ as the one-step state-transition probability from source state ‘i'’ to sink state ‘i’ with exactly j bit errors. Depending on the choice of sink state ‘i’ and the number of aggregated PAM-symbol stages, there are in total $m/2$ possible transitions between any of the two states in the simplified trellis. For example, for the transition a^j_{22} in Figure 5, $j \in \{1 \ldots m/2\}$ as all the aggregated paths end at i=2 has at least 1 bit error.

As such, we may construct a new trellis model for the entire FEC block, assuming that each state transition from the k_Fth to the (k_F+1)th stage has traversed a group of length-$m/2$ PAM-trellis paths. This is referred as the time aggregation of a Markov decision processes [4]; 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 trellis the FEC trellis model, distinguishing it from the PAM symbol-level trellis considered thus far.

In order to analyze the FEC trellis, we must first find all the state-transition probabilities of these 2^N states by analysis of each underlying 4-PAM trellis. Figure 6 shows an example illustrating the time-aggregation of a 4-PAM trellis for $N = 1$ and $m = 6$. The FEC trellis is expanded in Figure 6 showing the underlying 4-PAM trellis to illustrate how we may find state-transition probabilities $a^j_{i'i}$ in the FEC trellis. First, we instantiate the expanded PAM trellis by assuming that the PAM trellis starts at the state ‘i'’ in $a^j_{i'i}$ with a probability of 1,

$$Pr\,{}^0_0(i') = 1. \tag{9}$$

Next, after traversing the expanded 4-PAM trellis using the dynamic programming procedure described in (8), the transition probability $a^j_{i'i}$ to the next (k_F+1)th FEC trellis stage can be calculated by summing the probability of all j-bit-error PAM-trellis paths ending at state ‘i’,

$$a\,{}^j_{i'i} = Pr^j_{m/2}(i)\Big|_{Pr\,{}^0_0(i')=1}. \tag{10}$$

For example, in Figure 6, a^2_{12} corresponds to the summed probability of all PAM-trellis paths starting with state i = 1 and ending at i = 2 where 2 bit errors are detected in the fully expanded PAM trellis. For this particular case,

$$a\,{}^2_{12} = Pr^2_3(2)|_{Pr\,{}^0_0(1)=1} = p_{11}p_{12}p_{22} + p_{12}p_{21}p_{12}. \tag{11}$$

To compute the post-FEC BER, we must apply dynamic programming to enumerate the probability of all error patterns having more than t FEC symbol errors in a codeword. However, the dynamic programming algorithm described by (8) 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 trellis, we denote $Pr\_FEC_{kF}^{js,\,jb}(i)$ the probability of visiting Markov state i at time step k_F 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, $Pr\_FEC_{kF+1}^{js,\,jb}(i)$, can be found iteratively from the values of $Pr\_FEC_{kF}^{js,jb}(i)$ and the branch probabilities $a_{i'i}^j$. For a transition to state ‘i’ in the FEC trellis where the traversed $m/2$ PAM symbols have exactly j bit errors,

$$Pr\_FEC^{j_s,j_b}_{k_F+1}(i) = \sum\nolimits_{i'} Pr\_FEC^{j_s-min(1,j),j_b-j}_{k_F}(i')a^j_{i'i}. \tag{12}$$

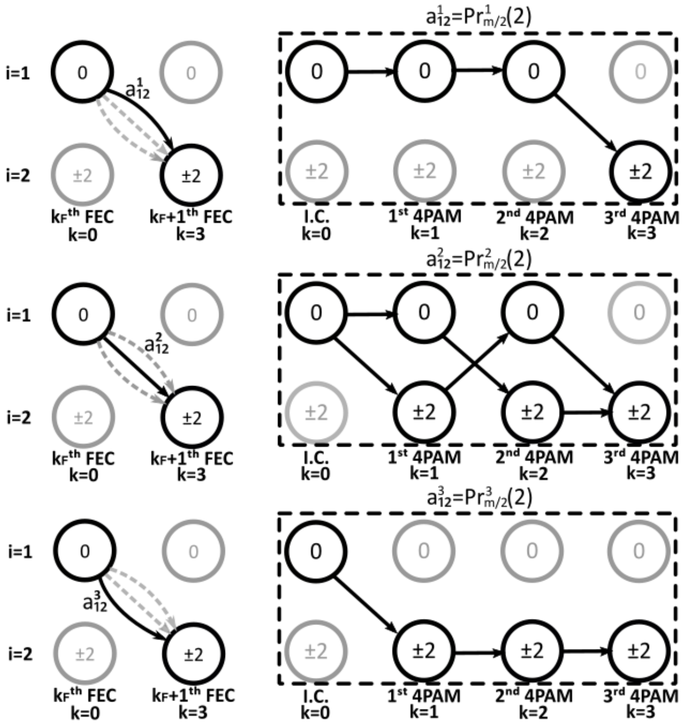


Figure 6. Time aggregating a 4-PAM trellis with *m* = 6 and *N* = 1 showing the time-aggregated PAM trellis and the corresponding aggregated one-step state-transition probability in the fully expanded PAM trellis.

4. Post-FEC BER Estimation and Model Validation

We first define $Pr\_FEC_n^{js,\,jb}$ as the grouped probability of all error patterns having j_s symbol errors and j_b bit errors along with a FEC trellis path of length *n*, computed by

$$Pr\_FEC_n^{j_s,j_b} = \sum_i Pr\_FEC_n^{j_s,j_b}(i). \tag{13}$$

Next, denote $W(j_s)$ the probability of having exactly j_s FEC symbol errors in an *n*-symbol codeword,

$$W(j_s) = \sum_{j_b=j_s}^{j_s\cdot\frac{m}{2}} Pr\_FEC_n^{j_s,j_b}. \tag{14}$$

To calculate BER, we define $E_{avg}(j_s)$ as the average number of bit errors in each erroneous FEC symbol given that exactly j_s symbol errors occurred in an *n*-symbol codeword,

$$E_{avg}(j_s) = \frac{\sum_{j_b=j_s}^{j_s\cdot\frac{m}{2}}\left(Pr\_FEC_n^{j_s,j_b}\cdot j_b\right)}{j_s\cdot W(j_s)}. \tag{15}$$

Then, the pre-FEC BER can be calculated as

$$BER_{pre-FEC} = \sum_{j_s=1}^{n}\left[\frac{W(j_s)\cdot E_{avg}(j_s)\cdot j_s}{n\cdot m}\right]. \tag{16}$$

Finally, to estimate the post-FEC BER for a *t*-error correcting RS code in GF(2^m) of block length *n*,

$$BER_{post-FEC} = \sum_{j_s=t+1}^{n}\left[\frac{W(j_s)\cdot E_{avg}(j_s)\cdot j_s}{n\cdot m}\right]. \tag{17}$$

At low BER, as $W(j_s)$ decreases exponentially with increasing j_s, pruning trellis paths having negligible probabilities can result in a significant reduction in computation. This is achieved by replacing the upper summation limit n in (16) and (17) with j_s^{max}, indicating only trellis paths having up to j_s^{max} FEC symbol errors are preserved.

$$BER_{post-FEC} \approx \sum_{j_s=t+1}^{j_s^{max}} \left[\frac{W(j_s)\cdot E_{avg}(j_s)\cdot j_s}{n\cdot m}\right]. \qquad (18)$$

A 4-PAM statistical model is applied to a link as depicted in Figure 1 with two channel settings $h = 0.6 + 0.2z^{-1} - 0.2z^{-2}$ and $h = 0.6 + 0.2z^{-1} + 0.2z^{-2}$. The negative ISI cursor in the first case may, for example, arise from the combination of a lowpass channel and a continuous time linear equalizer (CTLE) that over-equalizes the channel. The solid lines in Figure 7 reports the pre-FEC vs post-FEC BER with the two channels, calculated using the statistical methods described above for the RS(544,536,4) code on GF(2^{10}). The dotted line reports the results neglecting DFE burst errors. Behavioral simulation results for the negative ISI case are superimposed on the same axes to verify the correctness of our model down to a post-FEC BER of 10^{-8}.

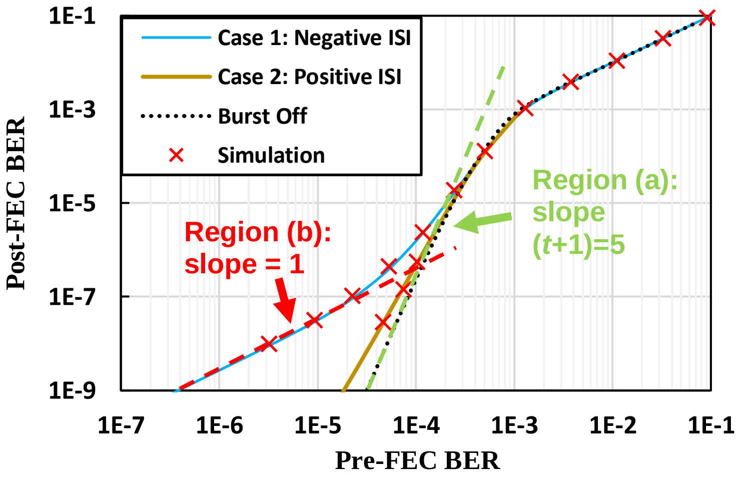


Figure 7. Pre-FEC vs post-FEC BER plot for RS(544,536,4) with $h = 0.6 + 0.2z^{-1} - 0.2z^{-2}$.

In Figure 7, we may identify two regions of interest for the results generated by case 1. First, consider an extreme case where burst errors arise much less frequently than random errors. In such a case, a codeword will be decoded incorrectly only when there are (t+1) random bit errors, each having probability p. Hence, post-FEC BER ~ $p^{(t+1)}$. This case corresponds to the region (a) in Figure 7, where the slope of Post-FEC vs. Pre-FEC BER is (t+1) on a logarithmic scale. Another extreme case can be represented by region (b), where individual random bit errors turning into very long bursts are the dominant source of post-FEC errors. If some small fraction, b, of pre-FEC random errors will generate bursts long enough to create post-FEC errors, post-FEC BER ~ $b\cdot p$. Thus, the slope of post-FEC vs. pre-FEC BER in this region is 1 on a logarithmic scale.

Note that a much higher error floor is observed in case 1 though the two cases differ only in the sign of the second post-cursor ISI. The large but negative second DFE tap in case 1 comparatively increases the probability of error propagation in the DFE feedback loop, resulting in a much higher error floor, region (b) in Figure 7.

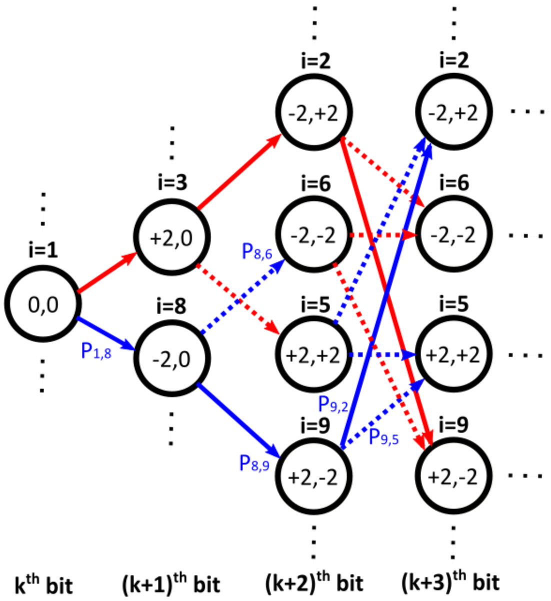


Figure 8. A 2-PAM trellis example showing a burst across 3 bits with *N* = 2; solid lines are the more probable trellis paths if $h_1 > 0$.

Here we will use a 2-tap DFE in 2-PAM as an example to prove that DFE error propagation is enhanced by negative DFE tap weights located at even tap locations, and this proof can be easily extended to a *N*-tap DFE using *M*-PAM signaling. Figure 8 shows the 2-PAM trellis for a burst of errors across 3 bits using the original non-aggregated Markov model where the 3^2 DFE states correspond to those in Figure 2(a). We further simplify Figure 8 by omitting all state transitions and the associated states which do not lead to consecutive decision errors. In the example, the burst is first triggered by a random error in the $(k+1)^{\text{th}}$ bit of a codeword, and is then followed by consecutive errors across subsequent trellis stages. Therefore, the trellis starts at time *k* in state $i = 1$ without any preceding DFE errors, $n_k^{dfe} = 0$. At k+1, two trellis paths are highlighted in red and blue corresponding to errors with $b_{k+1} = -1$ and +1, respectively. According to (2) the first random receiver error D_{k+1} in Figure 8 results in an additive error at the receiver input at $k + 2$

$$n_{k+2}^{dfe} = -D_{k+1} h_1. \tag{19}$$

Thus, with $h_1 > 0$, past bit errors pass through the negative feedback of the DFE tend to cause a subsequent error of opposing sign. The solid and dashed lines in each set of colored paths represent the more probable and less probable burst error paths through the trellis. For example, with $h_1 > 0$, $P_{8,9} >> P_{8,6}$ and $P_{9,2} >> P_{9,5}$ indicating that burst errors are more likely to be in the form "… +2, -2, +2 …".

After two consecutive errors, the additive DFE error at time $k + 3$ is

$$n_{k+3}^{dfe} = -D_{k+1} h_2 - D_{k+2} h_1. \tag{20}$$

Since with $h_1 > 0$, D_{k+1} and D_{k+2} are most likely to have opposing signs, the two terms in (20) will add constructively resulting in the largest possible additive error term only if h_1 and h_2 have opposing signs, implying $h_2 < 0$.

Alternatively, if $h_1 < 0$ the additive error (19) is of the same sign as D_{k+1} increasing the probability of another error D_{k+2} having the same sign. In this case, the additive error (20) is increased when h_2 has the same sign as h_1; that is, when $h_2 < 0$. Thus, in either case the probability of propagating errors two or more time steps is maximized by a negative h_2.

To prove that the probability of having errors with opposing signs is higher if $h_1 > 0$ and vice versa, we assume $D_{k+1} = \pm 2$ and an equal probability of transmitting $b_k \in \{\pm 1\}$. According to (3) the probability of $D_{k+2} = +2$ is

$$P_{+2} = \frac{1}{2} Q\left(\frac{-h_0 \mp 2h_1}{\sigma}\right). \tag{21}$$

Similarly, under the same assumption the probability of $D_{k+2} = -2$ is

$$P_{-2} = \frac{1}{2} Q\left(\frac{-h_0 \pm 2h_1}{\sigma}\right). \tag{22}$$

With a h_0 and h_1 both being positive, $P_{+2} < P_{-2}$ if $D_{k+1} = +2$ and $P_{-2} < P_{+2}$ if $D_{k+1} = -2$. Therefore, it is much more likely that D_{k+1} and D_{k+2} have opposing signs if $h_1 > 0$. Similarly, in (21) and (22) if $h_1 < 0$, since $P_{+2} > P_{-2}$ if $D_{k+1} = +2$ and $P_{-2} > P_{+2}$ if $D_{k+1} = -2$, it can be easily proven that D_{k+1} and D_{k+2} are more likely to have the same sign.

Figure 9 gives an example on finding the transition probability $P_{1,8}$, $P_{8,6}$, $P_{8,9}$, $P_{9,2}$ and $P_{9,5}$ that are highlighted in Figure 8 using the channel setting in case 1 where $h = 0.6 + 0.2z^{-1} - 0.2z^{-2}$. The distribution of the DFE slicer input r_k in (3) is shown in each subplot assuming a 2-tap zero-forcing DFE with AWGN having a standard deviation σ. Knowing $n_k^{dfe} = 0$ at the kth bit, we can arbitrarily assign $b_k = b_{k-1} = -1$ without affecting the results of our analysis. First, following the blue path we start by sending $b_{k+1} = +1$ as shown in Figure 9(a) and the error probability $P_{1,8}$ can be calculated using the standard error function

$$P_{1,8} = \frac{1}{2} Q\left(\frac{-h_0}{\sigma}\right) = \frac{1}{2} Q\left(\frac{-0.6}{\sigma}\right). \tag{23}$$

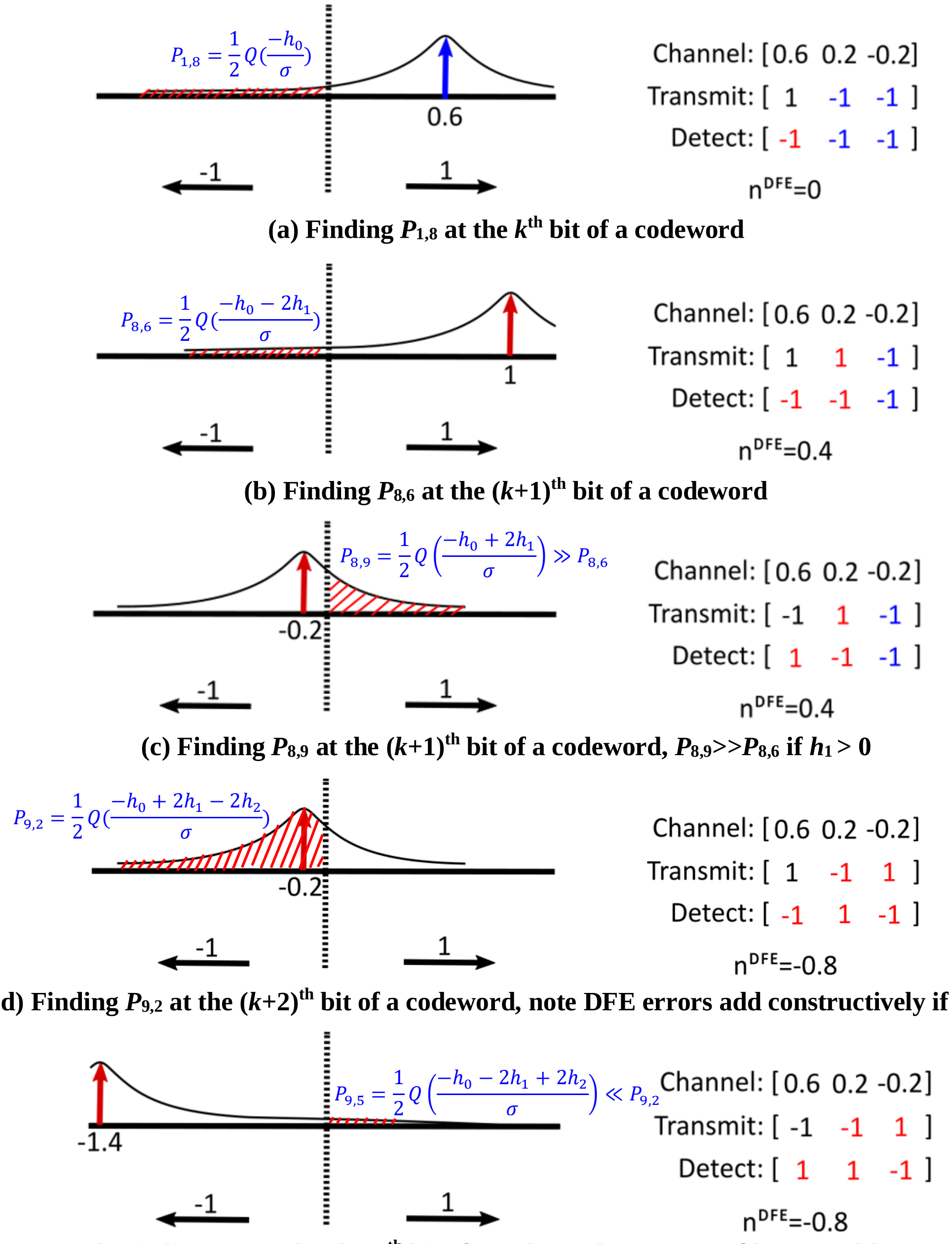
Next, given $D_{k+1} = -2$, the two state transitions $P_{8,6}$ and $P_{8,9}$ leading to $D_{k+2} \neq 0$ are plotted in Figures 9(b) and 9(c), respectively. With $h_0 = 0.6$ and $h_1 = 0.2$, we can directly compare the two transition probabilities using (21) and (22),

$$P_{8,6} = \frac{1}{2} Q\left(\frac{-h_0 - 2h_1}{\sigma}\right) = \frac{1}{2} Q\left(\frac{-1}{\sigma}\right) \ll P_{8,9} = \frac{1}{2} Q\left(\frac{-h_0 + 2h_1}{\sigma}\right) = \frac{1}{2} Q\left(\frac{-0.2}{\sigma}\right). \tag{24}$$

Then, at $k + 3$, with $h_2 < 0$ the two most-probable error terms $D_{k+2} = +2$ and $D_{k+1} = -2$ in (20) add constructively resulting in the largest possible n_{k+3}^{dfe}. State transition probabilities $P_{9,2}$ and $P_{9,5}$ are evaluated in Figures 9(d) and 9(e),

$$P_{9,2} = \frac{1}{2} Q\left(\frac{-h_0 + 2h_1 - 2h_2}{\sigma}\right) = \frac{1}{2} Q\left(\frac{0.2}{\sigma}\right) \gg P_{9,5} = \frac{1}{2} Q\left(\frac{-h_0 - 2h_1 + 2h_2}{\sigma}\right) = \frac{1}{2} Q\left(\frac{-1.4}{\sigma}\right). \tag{25}$$

Therefore, subject to a 3-bit error burst starting at time (k+1) in Figure 8, $P_{8,9} >> P_{8,6}$ and $P_{9,2} >> P_{9,5}$, showing the probability of having errors with opposing signs in between any two neighboring bits is much more likely if $h_1 > 0$. Similarly, this example can be easily extended to the case having $h_1 < 0$, where we would have $P_{8,9} << P_{8,6}$ and $P_{9,2} << P_{9,5}$ hence all subsequent error values in a burst are most likely to have the same sign as the initial random error.



(a) Finding $P_{1,8}$ at the k^{th} bit of a codeword

(b) Finding $P_{8,6}$ at the $(k+1)^{th}$ bit of a codeword

(c) Finding $P_{8,9}$ at the $(k+1)^{th}$ bit of a codeword, $P_{8,9}>>P_{8,6}$ if $h_1 > 0$

(d) Finding $P_{9,2}$ at the $(k+2)^{th}$ bit of a codeword, note DFE errors add constructively if $h_2 < 0$

(d) Finding $P_{9,5}$ at the $(k+2)^{th}$ bit of a codeword, $P_{9,5}<<P_{9,2}$ if $h_1 > 0$ and $h_2 < 0$

Figure 9. Finding the transition probability $P_{1,8}$, $P_{8,6}$, $P_{8,9}$, $P_{9,2}$ and $P_{9,5}$ in each subplot showing the probability distribution of DFE slicer input r_k with $h = 0.6 + 0.2z^{-1} - 0.2z^{-2}$.

5. Common Coding Techniques in Wireline Links

In this section we will discuss three common coding methods that impact burst-error probability and post-FEC BER. First, interleaved FEC codes, which increase latency but spread burst errors, are considered in Section 5.1. Then, MOD4 precoding is presented in Section 5.2 as a simple but effective method to minimize DFE bursts. Lastly, bit multiplexing is often required to combine several slower serial data streams in high-speed wireline links, and it is briefly discussed in Section 5.3.

5.1 Interleaved FEC Code

Figure 10(a) shows an example of 1:3 FEC symbol interleaving at a transmitter using the RS (544, 514, 15) KP4 code. Each transmitted codeword in the PHY layer is generated through a 3:1 FEC symbol multiplexer by taking FEC symbols from the three encoded codewords in a round-robin fashion. At the receiver shown in Figure 10(b), the signal flow is reversed to retrieve the codewords. Burst errors across multiple FEC symbols in the PHY layer are interleaved into 3 codewords, thus significantly reducing the probability of decoder failure in the presence of long DFE bursts.

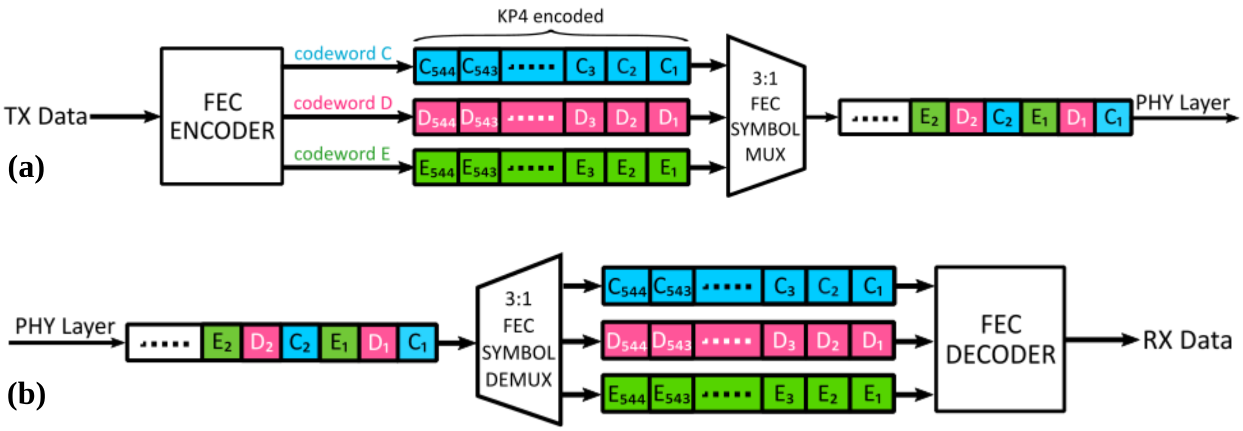

KP4 encoded
TX Data
FEC ENCODER
codeword C
codeword D
codeword E
C_{544} C_{543} ..... C_3 C_2 C_1
D_{544} D_{543} ..... D_3 D_2 D_1
E_{544} E_{543} ..... E_3 E_2 E_1
3:1 FEC SYMBOL MUX
..... E_2 D_2 C_2 E_1 D_1 C_1
PHY Layer
(a)
PHY Layer
..... E_2 D_2 C_2 E_1 D_1 C_1
3:1 FEC SYMBOL DEMUX
C_{544} C_{543} ..... C_3 C_2 C_1
D_{544} D_{543} ..... D_3 D_2 D_1
E_{544} E_{543} ..... E_3 E_2 E_1
FEC DECODER
RX Data
(b)

Figure 10. An example of 1:3 FEC symbol interleaving using the RS (544, 514, 15) KP4 code: (a) at transmitter (b) at receiver, errors across multiple FEC symbols are interleaved into 3 codewords.

To model FEC symbol interleaving, we must consider error propagation in the PHY layer using the FEC trellis model. For a 1:*x* interleaved FEC, the block length of the FEC trellis model must be extended to *x* times longer than codeword length *n*. Figure 11 shows the modeling of 1:3 FEC symbol interleaving using a length-3*n* FEC trellis model with 4-PAM and *N* = 1. In the example we may arbitrarily choose codeword *C* for BER analysis. When iterating each FEC trellis stage in Figure 11 using (12), the error indices j_s and j_b only increase if errors occur in codeword *C*. Without FEC interleaving, random and burst errors may corrupt multiple FEC symbols within the non-interleaved codeword, while in the example of Figure 11 the same errors will instead spread into 3 interleaved codewords making the post-FEC BER better.

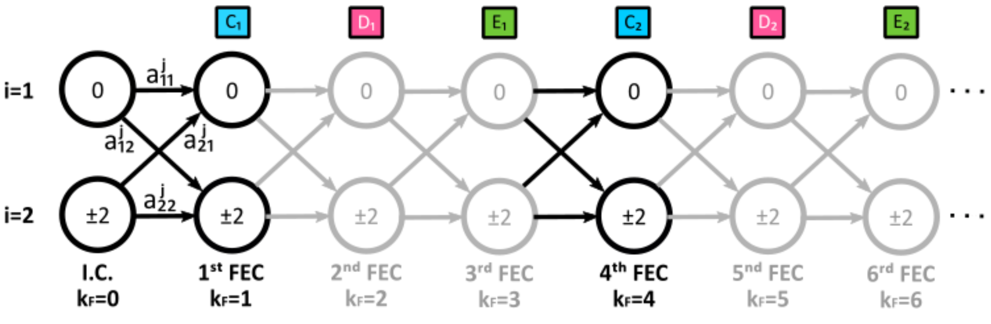


Figure 11. Modeling 1:3 FEC symbol interleaving using a length-3*n* FEC trellis model with 4-PAM and *N* = 1, the error index j_s and j_b only increases if errors are located in codeword *C*.

A 4-PAM statistical model is applied to a link as depicted in Figure 1 with a channel response $h = 0.5 + 0.25z^{-1} - 0.25z^{-2}$. The solid line in Figure 12 reports the pre-FEC vs post-FEC BER

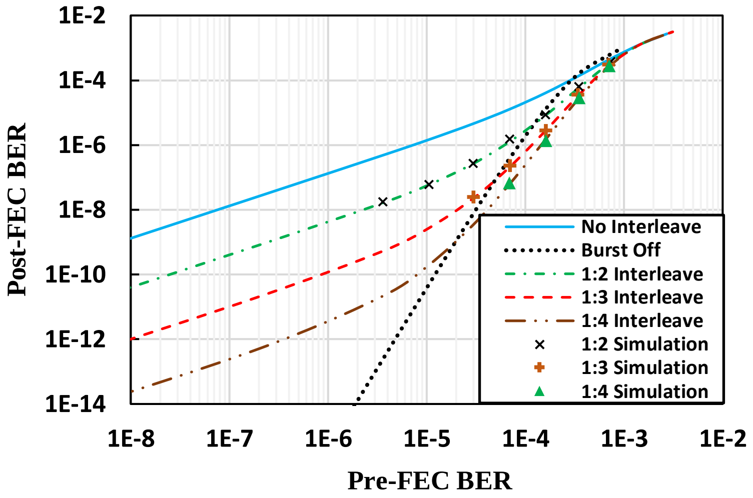


Figure 12. Pre-FEC vs post-FEC BER plot for interleaved RS(1000,992,4) codes with $h = 0.5 + 0.25z^{-1} - 0.25z^{-2}$.

calculated using the methods described above with three interleaved RS(1000,992,4) codes. Behavioral simulation results are superimposed in the same figure. The dotted lines report the results neglecting DFE burst errors. Note that as SNR increases, long burst errors in the PHY layer eventually corrupt the interleaved FEC decoder, hence the burst-free performance described by the dotted line can never be achieved by FEC interleaving. As we increase the number of interleaved codewords, better post-FEC BERs are observed in Figure 12 at the cost of multiplying the overall system latency by the same ratio.

5.2 MOD4 Precoding

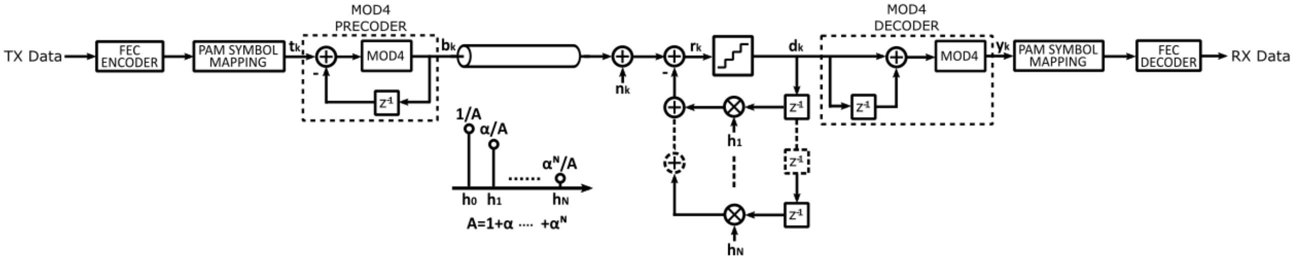
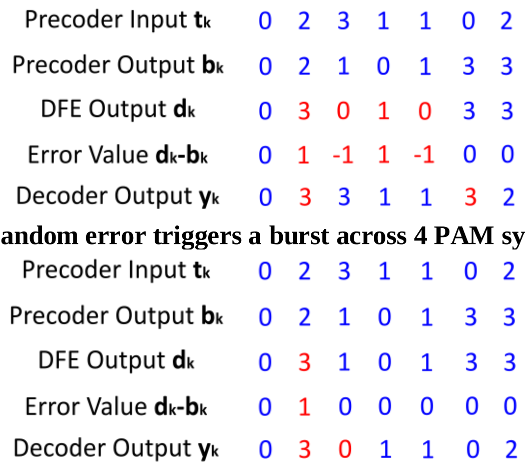


Figure 13. System-level diagram of a 4-PAM wireline SerDes example with a zero-forcing *N*-tap DFE and MOD4 precoding.

MOD4 precoding is a simple technique that can be used to minimize the impact of DFE error propagation. Figure 13 shows a system-level diagram of a 4-PAM wireline SerDes example with a zero-forcing *N*-tap DFE and MOD4 precoding. MOD4 precoding can be considered a multilevel version of the 2-PAM duo-binary precoder: at the transmitter, a MOD4 precoder encodes the 4-PAM signal, t_k, into $b_k = (t_k + b_{k-1}) \bmod 4$; the receiver decodes the DFE output, $y_k = (d_k + d_{k-1}) \bmod 4$.

In Figure 14 we present two examples of MOD4 precoding with $h_1 > 0$ and a 4-PAM symbol set $b_k \in \{0, 1, 2, 3\}$. Both examples use the same data pattern at precoder input t_k and other important node values in Figure 13 are recorded in each subplot of Figure 14. In Figure 14(a) the random error triggers a burst of errors across 4 4-PAM symbols at the DFE output d_k. As described in Section 4, each non-zero error D_k in the burst alternates between opposing signs if $h_1 > 0$. The MOD4 decoder at the receiver minimizes the burst length by summing the current error value D_k with D_{k-1}. This would reduce any DFE error burst into two errors at the decoder output, one at the start and another at the end of the burst. However, as a result we see in Figure 14(b) that a lone random error subject to the MOD4 decoder produces two errors at its output.



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

(a) a random error triggers a burst across 4 PAM symbols

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	1	0	1	3	3
Error Value d_k-b_k	0	1	0	0	0	0	0
Decoder Output y_k	0	3	0	1	1	0	2

(b) a random error without triggering DFE burst

Figure 14. Two numerical examples of MOD4 precoding with $h_1 > 0$ and $b_k \in \{0, 1, 2, 3\}$.

The operation of MOD4 precoding can be divided into four cases based on the current and the previous error value: $D_k = D_{k-1} = 0$; $D_k \neq 0$ and $D_{k-1} = 0$; $D_k \neq 0$ and $D_{k-1} \neq 0$; $D_k = 0$ and $D_{k-1} \neq 0$. The dynamic programming procedure described by (8) should be modified to track both D_k and D_{k-1} at each trellis node. Figure 15 shows a trellis model of 4-PAM MOD4 precoding with $N = 2$. The trellis path highlighted in the example corresponds to the numerical example in Figure 14(a). First, at $k = 0$, the link has no error and $D_k = D_{k-1} = 0$. Correspondingly we have

$$Pr_{k+1}^{j}(i) = \sum_{i'} Pr_k^{j}(i')p_{i'i}. \tag{26}$$

Next, if there is a random error in the PHY layer at $k = 1$ where $D_k \neq 0$ and $D_{k-1} = 0$, an error is produced at the MOD4 decoder output,

$$Pr_{k+1}^{j}(i) = \sum_{i'} Pr_k^{j-1}(i')p_{i'i}. \tag{27}$$

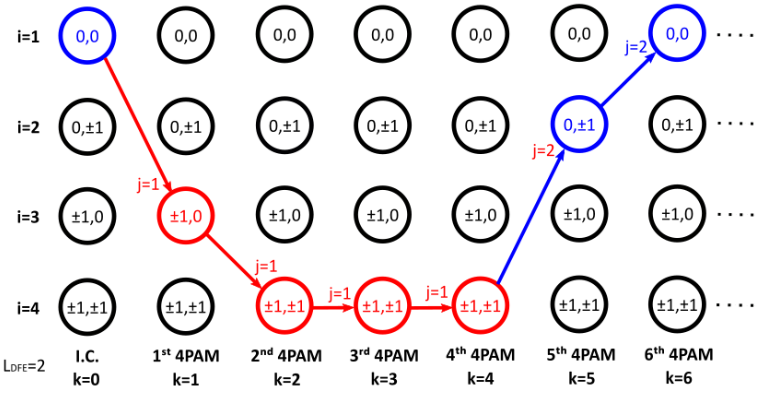


Figure 15. Modeling MOD4 precoding using the 4-PAM trellis model with N = 2; the trellis path in the example corresponds to the error values in Figure 14(a).

Then, assuming the random error at k = 1 triggers an error burst in the DFE, in the next UI $D_k \neq D_{k-1} \neq 0$, the error values with opposing signs cancel each other out and generate an error-free 4-PAM symbol at the decoder output (assuming $h_1 > 0$),

$$Pr_{k+1}^{j}(i) = \sum_{i'} Pr_{k}^{j}(i') p_{i'i}. \tag{28}$$

Lastly, at k = 5, a MOD4 decoder error is generated at the end of the error burst where $D_k = 0$ and $D_{k-1} \neq 0$,

$$Pr_{k+1}^{j}(i) = \sum_{i'} Pr_{k}^{j-1}(i') p_{i'i}. \tag{29}$$

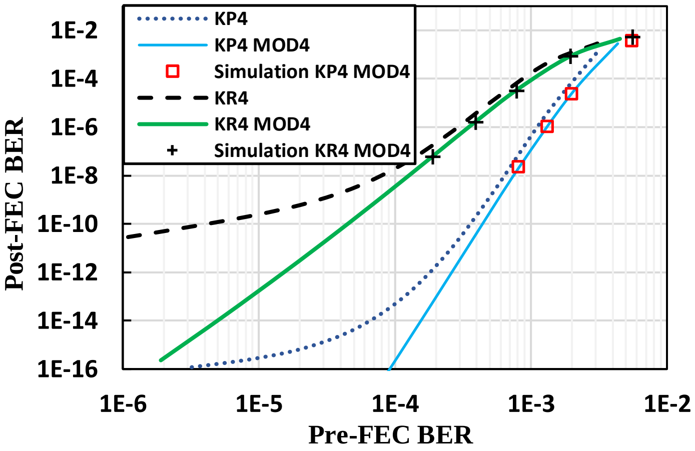


Figure 16. Pre-FEC vs post-FEC BER plot for the RS(544,514,15) KP4 and RS(528,514,7) KR4 code with $h = 0.6 + 0.2z^{-1} - 0.2z^{-2}$.

The 4-PAM link depicted in Figure 13 with a channel response $h = 0.6 + 0.2z^{-1} - 0.2z^{-2}$ is used to validate the proposed statistical model with MOD4 precoding. The solid line in Figure 16 reports the pre-FEC vs post-FEC BER calculated using MOD4 precoding with the RS(544,514,15) KP4 and RS(528,514,7) KR4 codes. Behavioral simulation results are superimposed in the same figure. The dotted and dashed lines report the results without MOD4 precoding, and in both cases the error floor presented at low BER is successfully removed by MOD4 precoding. If we qualitatively compare Figure 16 with the pre-FEC vs post-FEC BER plot in Figure 12, MOD4 precoding appears superior to the FEC interleaving due to its lower latency and burst-error performance at low BERs. However, this is only true if the link operates at low BERs where long DFE bursts are the dominant source of post-FEC errors. At high BERs where random errors are the dominant source, the additional errors created by the MOD4 decoder from individual random errors will make both the pre-FEC and post-FEC BER worse.

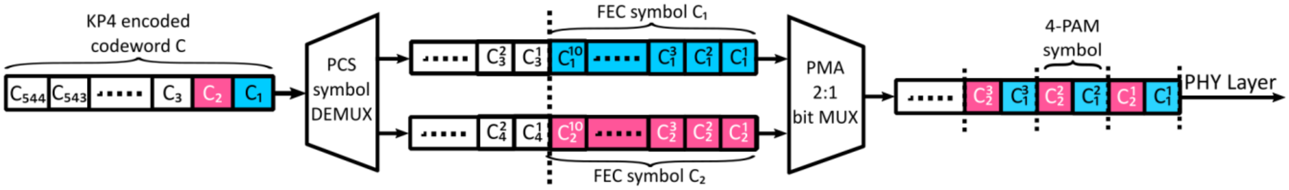


Figure 17. System-level diagram showing FEC symbol distribution and 2:1 bit multiplexing at TX.

5.3 Bit Multiplexing

To comply with IEEE standards, a bit multiplexer is required in high-speed wireline links, which we now show has an impact on post-FEC BER. Figure 17 demonstrates an example showing FEC symbol distribution and 2:1 bit multiplexing at the transmitter. FEC symbols C_1, C_2, ... C_{544} in a KP4-encoded codeword are distributed to two PCS lanes (in a round-robin fashion). Then, a bit multiplexer in the PMA layer groups every two bits from each PCS lane (e.g. the first bit from the two FEC symbols C_1 and C_2 are C_1^1 and C_2^1) to form each physical-layer 4-PAM symbol. At the receiver, the signal flow in Figure 17 is reversed to retrieve the codeword C. As a result, even a short string of burst errors in the physical layer is shuffled across multiple FEC symbols thus making the post-FEC BER worse. A detailed explanation of modeling bit multiplexing using our proposed statistical model appears in [5].

6. Modeling Other Type of Noise Sources

Until now, we have assumed that all pre-cursor and post-cursor ISI has been removed by TX and RX equalizers. In Section 6.1, we will first introduce a method to accurately consider residual ISI in BER estimation using our statistical model. This method serves as the basis of Section 6.2, allowing us to incorporate transmitter and receiver jitter into post-FEC BER estimation.

6.1 Residual ISIs

We start by considering a channel pulse response h_p having both pre-cursor and post-cursor ISI. Assuming the equalizers are not perfectly zero-forcing, we may define h_ISI_p as the equalized overall system pulse response with L_1 residual pre-cursor ISIs, a main cursor h_ISI_0 and L_2 post-cursor ISIs by computing the convolved response for the channel, transmitter and receiver equalizers. Assuming an N-tap DFE with tap coefficients h_DFE_p, the additive DFE error n_k^{dfe} previously defined in (2) becomes

$$n_k^{dfe} = -\sum_{p=1}^{N} D_{k-p}\, h\_DFE_p. \tag{30}$$

If we denote n_k^{ISI} as the aggregated residual ISI amplitude at time index k, the DFE slicer input r_k defined in (3) can be rewritten,

$$r_k = b_k h\_ISI_0 + n_k + n_k^{dfe} + n_k^{ISI}. \tag{31}$$

Without considering the memory effect of the DFE, the aggregated residual ISI amplitude n_k^{ISI} is

$$n_k^{ISI} = \sum_{p=-L_1}^{-1} b_p \cdot h\_ISI_p + \sum_{p=1}^{L_2} b_p \cdot h\_ISI_p. \tag{32}$$

For a 4-PAM symbol set $b_k \in \{\pm 3, \pm 1\}$, the pdf of n_k^{ISI} can be obtained by enumerating all possible data patterns in (32). However, for an N-tap DFE the past N transmitted symbols are dictated by the DFE state $<D_{k-1}, \ldots D_{k-N}>$, where each error value in the DFE state sets the transmitted symbol having the same time index. For example, if the error propagation in 4-PAM is modeled using the 16 error values $D_k \in \{0_T, 0_{M1}, 0_{M2}, 0_B, \pm 2_T, \pm 2_M, \pm 2_B, \pm 4_T, \pm 4_B, \pm 6\}$ as defined in Figure 3, one would be able to identify the past N transmitted symbols from the sign and subscript of each error value. Consequently, when calculating the one-step state-transition probabilities $q_{i'i}$ from a source state 'i'' to a sink state 'i' by applying (31) to each pair of valid transitions in the Markov model, if the first N post-cursor ISIs are equalized by a DFE,

$$n_k^{ISI} = \sum_{p=-L_1}^{-1} b_p \cdot h\_ISI_p + \sum_{p=1}^{N} b_p \cdot h\_ISI_p + \sum_{p=N+1}^{L_2} b_p \cdot h\_ISI_p, \quad (33)$$

where the 4-PAM symbol set b_k in the second term can only take up a single value from {±3, ±1} that is dictated by the source DFE state at each time index $k \in \{1, 2 \ldots N\}$.

Subject to the data-dependent residual ISIs, we can no longer apply state lumping to produce a simplified 4^N-state Markov model with $D_k \in \{0, \pm2, \pm4, \pm6\}$. However, we can still simplify the 4-PAM trellis model by ignoring all the paths that have unlikely ±4 and ±6 error events. This results in a 10^N-state Markov model with $D_k \in \{0_T, 0_{M1}, 0_{M2}, 0_B, \pm2_T, \pm2_M, \pm2_B\}$. The same procedures discussed in Section 3 and 4 are used to compute post-FEC BER using the two-layer Markov model.

6.2 Jitter

RX jitter creates a sampling offset for each sampled point of the unit pulse response. This would make the equalized overall system pulse response h_ISI_p previously defined in Section 6.1 becoming a function of RX sampling jitter Φ_{RX}. The same procedures described by (30-33) can be applied to calculate the aggregated ISI pdf and thus estimate BER. RX jitter can be handled straightforwardly by convolving the BER at each sampling offset Φ_{RX} with the RX jitter distribution. However, TX jitter is more difficult to model since it is filtered and possibly amplified by the channel ISIs. Thus, our primary focus is the modeling of TX jitter in this section.

The ISI analysis in Section 6.1 calculates the total ISI distribution by enumerating all possible symbol values b_k for each non-zero residual ISI cursor. The unit-pulse response is used to generate the ISI pdf for each cursor and the results of each cursor are convolved to obtain the total ISI pdf with uncorrelated transmitted data. However, transmitter jitter modulates the rising/falling edge of each data transition thus creates a correlation between neighboring transmitted symbols. If the link is subject to TX jitters, convolving the pdf of each ISI cursor based on pulse response can no longer be applied to finding the total ISI distribution. Ref [6] proposed a method to account for jittered ISIs using a segment-based analysis as shown in Figure 18. Segments are defined as a jittery transition from the right half-UI of a symbol to the left half-UI of the subsequent symbol. Since every data

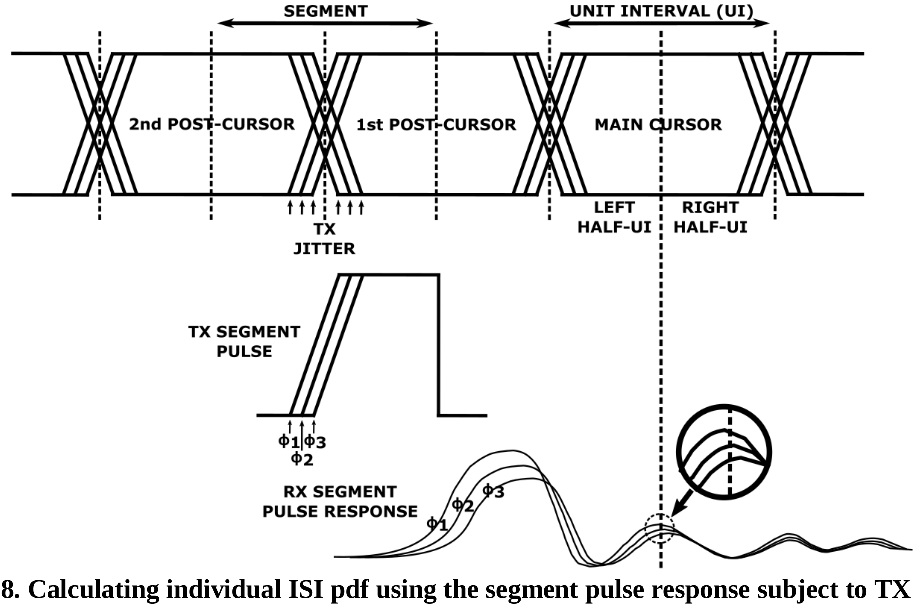


Figure 18. Calculating individual ISI pdf using the segment pulse response subject to TX jitter [6].

transition occurs in the middle of a segment, there is no jitter correlation between neighboring segments, and the pdf of each residual ISI in the presence of jitter derived can be found from segment pulse responses, as shown in Figure 18, then convolved to obtain the overall ISI pdf.

The segment-based analysis can accurately estimate ISI statistics in the presence of jitter by tracking each TX transition, where the step-amplitude of each segment's transition equals the distance between neighboring PAM symbols. However, the method in [6] works with non-return-to-zero segment pulse responses, which makes it difficult to use with 4-PAM where some waveform segments may have DC offset. Moreover, we seek a method compatible with conventional pulse-response-based simulations and measurements. Therefore, in this section we present an alternative statistical method to incorporate transmitter jitter into the post-FEC BER estimation.

In Figure 19, we propose a new methodology to estimate the data-dependent ISI distribution through jittered half-UI pulses that are derived from the standard unit-pulse response. By dividing each segment pulse into two half-UI pulses, we can apply conventional pulse-response-based simulation to efficiently compute the ISI pdf, including the DC component, of each segment by linear superposition of the two half-UI pulse responses. Figure 19 shows the decomposition of three segment pulses that are subject to early, on-time and late TX jitter, respectively. Taking the first subplot as an example where the jitter phase $\Phi_1 < 0$ is early, the jittered segment pulse representing a transition from -1 to 1 can be decomposed into two jittered half-UI pulses that are both subject to a phase shift on the same transition edge by Φ_1.

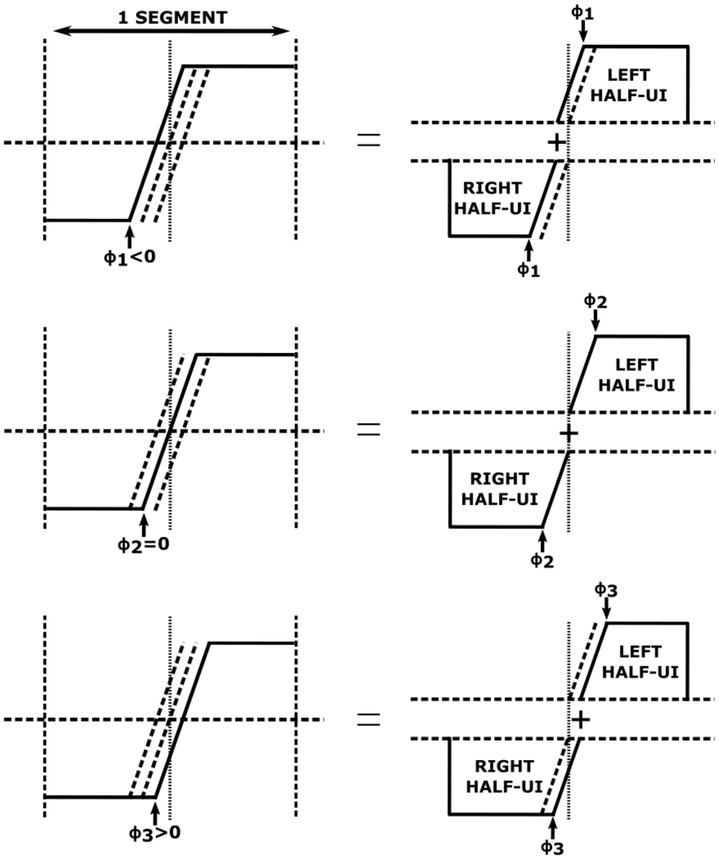


Figure 19. Estimating the segmented ISI pdf through jittered half-UI pulses.

At the receiver we define $H_L(\Phi, N_{seg})$ and $H_R(\Phi, N_{seg})$ being the left half-UI and right half-UI pulse response of the N_{seg}^{th} segment. Both half-UI responses are derived from the standard unit pulse response and are a function of TX jitter Φ and the segment index N_{seg}. Assuming a transition from b_{k-1} to b_k in the N_{seg}^{th} segment, the segment pulse response $J(\Phi, N_{seg}, b_{k-1}, b_k)$ can be represented by

$$J\left(\Phi, N_{seg}, b_{k-1}, b_k\right) = b_{k-1}H_R\left(\Phi, N_{seg}\right) + b_k H_L\left(\Phi, N_{seg}\right). \tag{34}$$

With an arbitrary TX jitter pdf p_Φ (Φ), the jittered transition ISI pdf of the N_{seg}^{th} segment $p_{ISI}(x, N_{seg}, b_{k-1}, b_k)$ can be computed by

$$p_{ISI}\left(x, N_{seg}, b_{k-1}, b_k\right) = \sum\nolimits_{\Phi\epsilon\{J(\Phi, N_{seg}, b_{k-1}, b_k)=x\}}[p_\Phi(\Phi)]. \tag{35}$$

Once the transition ISI pdf $p_{ISI}(x, N_{seg}, b_{k-1}, b_k)$ of each segment is calculated, we can use this information to calculate the aggregated ISI pdf $p_{AGG}(x, N_{seg}, b_{k-1}, b_k)$ in each segment. Figure 20 shows an example on finding the aggregated ISI pdf for a link having 1 pre-cursor ISI and 4 post-cursor ISIs using the 2-PAM symbol set $b_k \in \{0, 1\}$. The link has a 2-tap DFE equalizing the first two post-cursor ISIs, and we assume the current DFE state assigns 0 and 1 to the two post-cursor ISIs, respectively. For any two neighboring segments, the transmitted half symbol within the same UI must be identical. In Figure 20, starting with segment 4, we first average the transition ISI pdfs $p_{ISI}(x, N_{seg}, b_{k-1}, b_k)$ ending with the same b_k,

$$p_{AVG}\left(x, N_{seg}, b_k\right) = \sum\nolimits_{b_{k-1}} p_{ISI}\left(x, N_{seg}, b_{k-1}, b_k\right). \tag{36}$$

Next, in segment 3 where we have a transition from b_k to b_{k+1}, the accumulated ISI pdf $p_{AGG}(x, N_{seg}\text{-}1, b_k, b_{k+1})$ is obtained by convolving the transition ISI pdf $p_{ISI}(x, N_{seg}\text{-}1, b_k, b_{k+1})$ with the averaged ISI pdf ending with the same symbol b_k,

$$p_{AGG}\left(x, N_{seg} - 1, b_k, b_{k+1}\right) = p_{AVG}\left(x, N_{seg}, b_k\right) * p_{ISI}\left(x, N_{seg} - 1, b_k, b_{k+1}\right). \tag{37}$$

We can then average the accumulated ISI pdfs in segment 3 ending with the same b_{k+1},

$$p_{AVG}\left(x, N_{seg} - 1, b_{k+1}\right) = \sum\nolimits_{b_k} p_{AGG}\left(x, N_{seg} - 1, b_k, b_{k+1}\right). \tag{38}$$

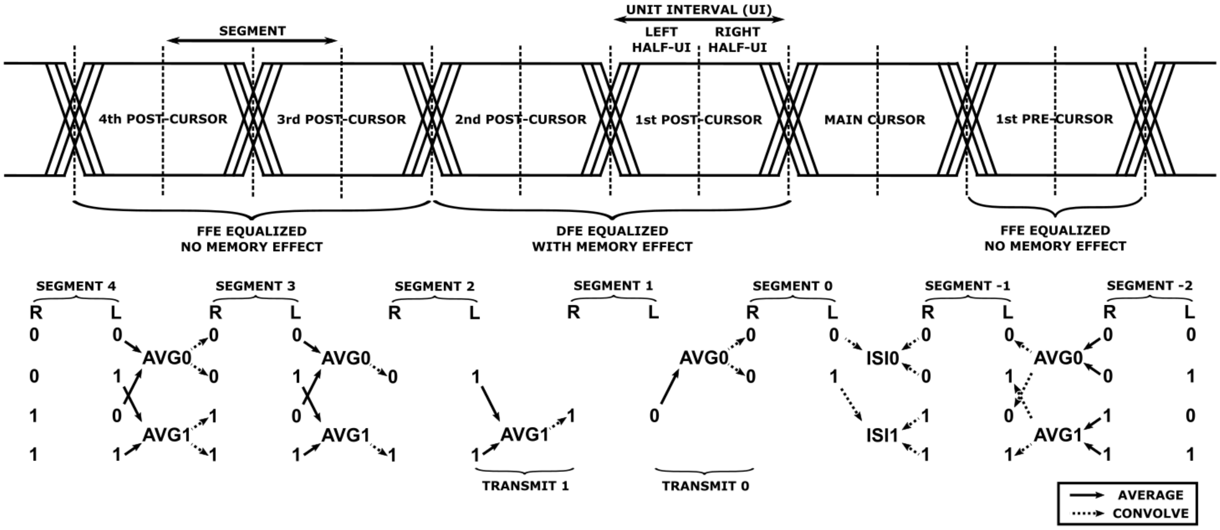


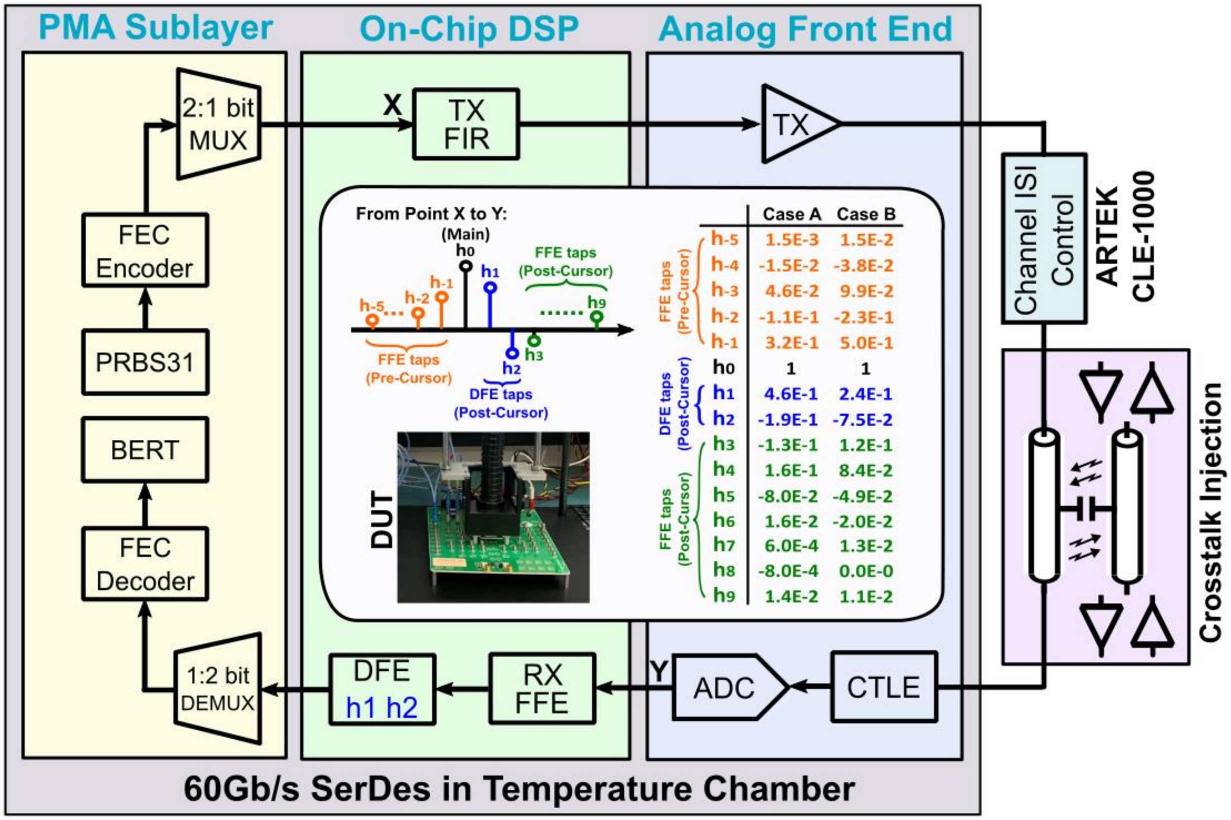
Figure 20. Finding the aggregated ISI pdf for a link having 1 pre-cursor ISI and 4 post-cursor ISIs using 2-PAM $b_k \in \{0, 1\}$; the link has a 2-tap DFE equalizing the first two post-cursor ISIs, and assuming the current DFE state assigns 0 and 1 to the two post-cursor ISIs, respectively.

In Figure 20, the aggregated ISI pdf of each segment can be computed by applying (37) and (38) recursively until reaching segment 0. Note that due to the memory effect of the DFE, in the example certain paths from segment 0 to segment 3 are removed to make sure that 0 and 1 are assigned to the 1st and 2nd post-cursor ISI, respectively.

7. Experimental Verification

7.1 Device Under Test

We have measured a 4-PAM 60 Gb/s SerDes link fabricated in 7 nm FinFET technology. The overall system-level block diagram of the link is plotted in Figure 21. Specifically, subject to a $1V_{ppd}$ maximum output swing, the transmitter has a programmable 3-tap FIR filter to mitigate both pre-cursor and post-cursor ISI. At the receiver, a 13-tap FFE with 5 pre-cursor taps and 7 post-cursor taps is adaptively optimized to cancel ISIs in the channel. A 2-tap DFE equalizes the first two post-cursor ISIs. A statistical unit on-chip monitors and stores BER for PRBS31 data in memory. Both the RS(544, 514, 15) KP4 and RS(528, 514, 7) KR4 codes in $GF(2^{10})$ are implemented in the FEC encoder/decoder.



		Case A	Case B
FFE taps (Pre-Cursor)	h_{-5}	1.5E-3	1.5E-2
	h_{-4}	-1.5E-2	-3.8E-2
	h_{-3}	4.6E-2	9.9E-2
	h_{-2}	-1.1E-1	-2.3E-1
	h_{-1}	3.2E-1	5.0E-1
	h_0	1	1
DFE taps (Post-Cursor)	h_1	4.6E-1	2.4E-1
	h_2	-1.9E-1	-7.5E-2
FFE taps (Post-Cursor)	h_3	-1.3E-1	1.2E-1
	h_4	1.6E-1	8.4E-2
	h_5	-8.0E-2	-4.9E-2
	h_6	1.6E-2	-2.0E-2
	h_7	6.0E-4	1.3E-2
	h_8	-8.0E-4	0.0E-0
	h_9	1.4E-2	1.1E-2

Figure 21. System-level block diagram and test setup of the 60 Gb/s SerDes link [7].

7.2 Test Setup

The test bench setup for the 60 Gb/s SerDes link is also superimposed in Figure 21. A FlexTC temperature forcing system from Mechanical Devices is used to keep the device at room temperature with ±0.2 °C accuracy. Approximately Gaussian-distributed crosstalk noise is coupled to the channel through a crosstalk injection board. Different measurement cases are established by varying the channel insertion loss using an ARTEK CLE1000 variable ISI channel. The corresponding overall pulse responses (including TX FIR, TX driver, channel, RX CTLE and ADC) for two different cases are also tabulated in Figure 21.

In case A, the overall insertion loss is 29 dB. We intentionally configure the CTLE in this case to over-equalize so that the second post-cursor ISI of the overall pulse response becomes large but negative. DFE error propagation is particularly bad in this case compared with all-positive post-cursor ISIs. With large negative DFE tap weights, a measurable floor is expected in the post-FEC BER where burst errors due to error propagation in the DFE dominate. In this region, we expect to see a plot of post- vs. pre-FEC BER exhibit a slope of 1. In case B, the system has a lower overall insertion loss of 24 dB so that the KR4 code can provide adequate coding gain at low BER.

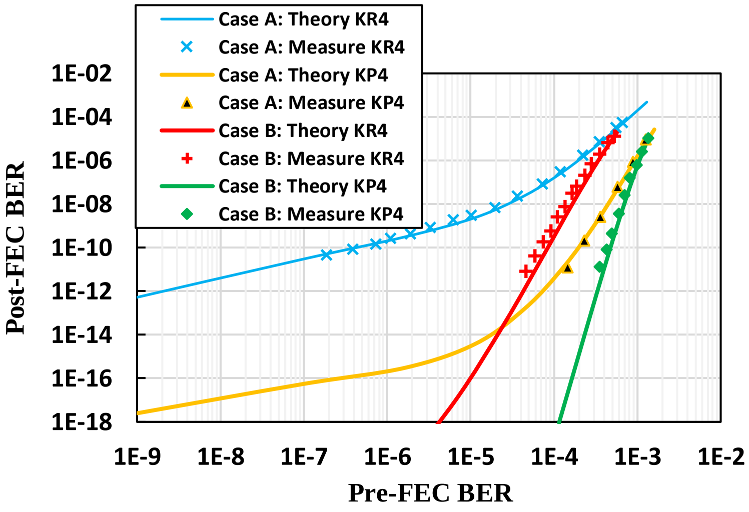


Figure 22. Measured and theoretical pre-FEC vs post-FEC BER plot for RS(528, 514, 7) and RS(544, 514, 15) code.

7.3 Experimental Results

In Figure 22, measured results for both the RS(544, 514, 15) KP4 and RS(528, 514, 7) KR4 codes are reported. Gray encoding is enabled to reduce BER. Different data points are generated by varying the amount of Gaussian-like crosstalk injected to the channel. To minimize the impact of jitter, all data points are measured by locking the CDR phase and DFE tap weights once the DFE tap weights' LMS adaptation has converged. The link is subject to a TX and RX random jitter both being 160fs$_{\text{rms}}$. The curves generated by our statistical model are also superimposed in Figure 22, treating the crosstalk as additive white Gaussian noise.

All data points in Figure 22 are measured down to a post-FEC BER of 10^{-11}. Good consistency is observed between the theoretical curves and measured results. Moreover, for case A where a large amount of error propagation is present, our statistical model can properly predict the error floor with the RS(528, 514, 7) KR4 code. Importantly, our statistical model accurately predicts the measured transition between the two regions for the KR4 and KP4 FEC in case A. The model indicates that for the KP4 FEC, in order to ensure a post-FEC BER of 10^{-18}, a pre-FEC BER of 10^{-4} is adequate for case B, whereas a pre-FEC BER of 10^{-10} is required for case A, conclusions that would have been almost impossible to draw using pre-existing methods. Our statistical model can be used to quantify the precise pre-FEC BER required to achieve very low post-FEC BER depending on the channel and equalizer.

8. Conclusion

We presented a statistical approach that accurately estimates post-FEC BER for high-speed wireline links subject to DFE burst errors and other important noise sources. The proposed statistical approach allows efficient aggregation of PAM-symbol and FEC-symbol errors through a series of techniques with controllable analytical accuracy in post-FEC BER estimation. This approach can simulate wireline links using different equalization techniques/coding schemes and subject to various noise sources. In addition, a novel statistical ISI analysis method is presented to calculate the data-dependent ISI distribution through jittered half-UI pulses derived from the standard unit pulse response. While this paper demonstrates the statistical analysis method in wireline context, the method is general and can be applied to model other communication systems having memory effects.

References

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

[2] R. Kennedy and B. Anderson, "Recovery Times of Decision Feedback Equalizers on Noiseless Channels," in *IEEE Transactions on Communications*, vol. 35, no. 10, pp. 1012-1021, October 1987.

[3] C. D. Meyer, “Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems,” *SIAM Rev.*, vol. 31, no. 2, pp. 240–272, 1989.

[4] 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.

[5] 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*.

[6] B. Casper *et al.*, "Future Microprocessor Interfaces: Analysis, Design and Optimization," *2007 IEEE Custom Integrated Circuits Conference*, San Jose, CA, 2007, pp. 479-486.

[7] M-A. Lacroix *et al.*, "A 60Gb/s PAM-4 ADC-DSP transceiver in 7nm CMOS with SNR-based adaptive power scaling achieving 6.9pJ/b at 32dB loss," *2019 IEEE International Solid - State Circuits Conference - (ISSCC)*, San Francisco, CA, 2019.