Channel Code Analysis and Design using Multiple Variable-Length Codes in Parallel without Feedback

Haobo Wang and Richard D. Wesel
UCLA Communications Systems Laboratory, Electrical and Computer Engineering Department
University of California, Los Angeles (UCLA), Los Angeles, CA 90095, USA
Email: {whb12, wesel}@ucla.edu

***Abstract*—This paper considers a channel coding paradigm that enables high throughput by using many variable-length codes in parallel, where each of the parallel codes has a short average blocklength. The inter-frame coding of Zeineddine and Mansour provides variable-length codes with incremental redundancy from a common pool of redundancy in a way that does not require feedback. A probability-based derivation of a generalized peeling decoder extends the results of Luby *et al.* to the inter-frame scenario. A new expression characterizes the probability that a variable-length decoder in the inter-frame system will fail. Additionally, the three causes for throughput loss as compared to the original feedback system are identified, yielding a new, and far simpler, quasi-regular design methodology for the right degree distribution of the inter-frame code. The inter-frame paradigm can apply to any communication channel, but this paper uses the additive white Gaussian noise channel to demonstrate the concepts.**

I. Introduction

Practical systems and theoretical analysis [1]–[3] show that using a variable-length (VL) code with incremental transmissions controlled by ACK/NACK feedback can approach capacity with short average blocklengths on the order of 200-500 symbols. This paper studies the analysis and design of systems that use many variable-length codes in parallel and ***without feedback*** to approach capacity for point-to-point communication.

As described in [4] a large number of capacity-approaching VL codes can be decoded in parallel without feedback using the inter-frame coding approach of Zeineddine and Mansour [5], where an appropriate number of linear combinations of incremental redundancy, which we will refer to as a ***common pool of redundancy (CPR)***, are transmitted. This can also be considered as an example of a doubly generalized LDPC code [6]. A peeling decoder applied to the CPR provides incremental redundancy to the VL decoders. Peeling decoders have been applied in a similar way for multiple access channels [7]–[10].

The proposed coding system has significant advantages in high-throughput applications providing both parallelism and the complexity advantage of short-blocklength decoders. Optical communications and non-volatile memory data storage are two potential applications for such systems. With an additional high-rate code to correct for the occasional VL code failure, the very low frame error rates required by these applications can be achieved.

The parallel system has as its conceptual building block a VL code, which for the examples in this paper is a tail-biting convolutional code with the reliability output Viterbi decoding algorithm in [4] and pseudo-random puncturing. The expected number of increments required by the VL code (with feedback) determines an upper bound on the throughput of the proposed system (that does not use feedback), but as shown below, practical inter-frame coding systems suffer a small loss in throughput from that bound.

The inter-frame code that generates the CPR is a generalization of one stage of the cascaded erasure correction scheme in [11], which is described by a bipartite graph. The left nodes in the graph represent the VL codes, and right nodes are the linear combinations of incremental redundancy that comprise the common pool of redundancy. Once the VL code has been specified, the main design question becomes the determination of the left and right degree distributions.

Zeineddine and Mansour showed in [5] that using a generalization of the heavy-tail distribution for the left degree distribution and a mixture of Poisson distributions for the right distribution can asymptotically achieve the throughput of the VL code with feedback if the number of increments required by the VL code follows a geometric distribution. The generalized heavy-tail distributions of [5] have (asymptotically) infinite support, but practical systems will have finite support. The system studied in this paper is constrained such that all left nodes have the same degree, which is the number of increments (four in this paper) produced by the VL encoder. Furthermore, the distribution describing the number of increments required to decode is not a geometric distribution according to analysis in [2] and [12].

In the context of a fixed, regular left degree distribution and a non-geometric, empirically-obtained distribution on the number of increments required for successful VL decoding, this paper seeks to find the most suitable right degree distribution in the context of two main performance goals:

1) Maximizing feedback-free throughput $R_t^{(FF)}$, which is upper bounded by the rate $R_t^{(FB)}$ of the feedback system;
2) Guaranteeing that the feedback-free VL decoder failure rate ϵ_{FF} is below a target value ϵ_{FF}^*.

Research supported by National Science Foundation (NSF) grant CCF-1618272. Any opinions, findings, and conclusions or recommendations are those of the author(s) and do not necessarily reflect the views of the NSF.

978-1-5386-4727-1/18/$31.00 ©2018 IEEE

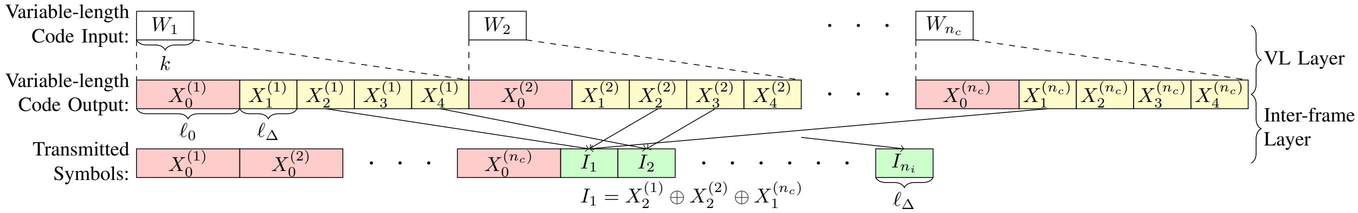


Fig. 1. Encoder structure for an inter-frame coding system

This paper provides a new expression for the probability ϵ_{FF} that a variable-length decoder will fail in the feedback-free system and identifies the three primary causes for throughput loss from the original feedback system. These analytical tools facilitate a new and far simpler design methodology for the right degree distribution of the inter-frame code.

Section II describes inter-frame coding. Section III analyzes inter-frame code convergence, provides a new expression for ϵ_{FF}, and explains mechanisms that lead to throughput loss. Section IV examines potential right degree distributions and studies the trade-off between the number of iterations and the throughput $R_t^{(FF)}$. Section V presents our conclusions.

II. Inter-frame Coding

An inter-frame coding system [5] consists of two layers: a VL code layer and an inter-frame code layer. Fig. 1 shows the encoder structure. At the transmitter, the VL code layer has n_c encoders in parallel. Each VL encoder takes as input a message W_i of length k bits, and produces an initial transmission $X_0^{(i)}$ and $m-1$ incremental transmissions (a.k.a. increments) $X_j^{(i)}$. Transmissions $X_0^{(i)}$ have length ℓ_0 symbols, and transmissions $X_j^{(i)}$ $(j > 0)$ have length ℓ_Δ. Overall, there are m possible transmissions for each VL codeword.

The initial transmissions $X_0^{(i)}$ are sent over the point-to-point channel as usual, but the $m-1$ increments $X_j^{(i)}$ $(j > 0)$ are not. Instead, the inter-frame code sends n_i incremental transmissions I_i, all of length ℓ_Δ. Each I_i is a linear combination (bit-by-bit exclusive-or) of increments $X_j^{(i)}$ $(j > 0)$ from distinct VL codewords. Each increment from a VL codeword is used in one linear combination I_i, with the exact linear combinations determined by a bipartite graph describing the inter-frame code. These I_i's form the CPR that are sent through the channel. In this paper, the point-to-point channel is a 2dB additive Gaussian channel.

The decoder of the inter-frame system can also be modeled as a bipartite graph as in Fig. 2, which corresponds to the inter-frame layer in Fig. 1 rotated counter-clockwise by 90 degrees. VL decoders (left nodes) attempt to decode immediately upon receiving the initial transmissions $X_0^{(i)}$ distorted by channel noise n. Some decoders will be able to succeed. The CPR (I_i+n's, right nodes) provides increments to the VL decoders that initially fail, as the increments become available from a peeling process. That is, each time a VL decoder succeeds in decoding, it generates its increments $X_j^{(i)}$'s and removes them from the linear combinations of the increments ($(I_i + n)$'s). When all but one increment has been removed from a linear

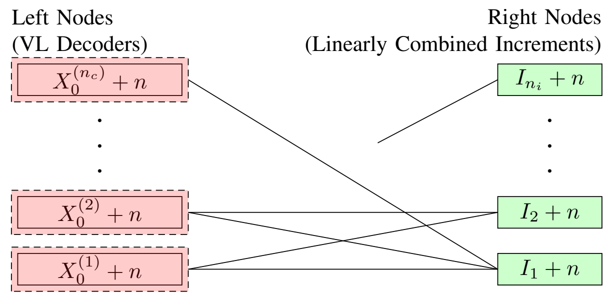


Fig. 2. Decoder structure for an inter-frame coding system

combination I_i, the remaining increment is available to its corresponding VL decoder.

The probability mass function

$$\boldsymbol{\delta} = \{\delta_0, \ldots, \delta_\omega, \ldots, \delta_m\} \tag{1}$$

represents the probability that the VL decoder requires ω increments beyond the initial transmission to decode successfully[1], i.e. $\omega+1$ total transmissions. For $0 \le \omega \le m-1$, $\delta(\omega)$ is the probability that a codeword is successfully decoded after ω increments, and $\delta(m) = \epsilon_{FB}$ is the probability that the VL decoder will fail even after all $m-1$ increments are received. The $\boldsymbol{\delta}$ distribution captures the decoding statistics of the VL code for a given point-to-point channel, and is used in the analysis of inter-frame code's decoding performance. As is in [4], we use the $k = 64$ 1024-state tail-biting convolution code as the VL code. The maximum number of transmission is $m = 5$ and $\boldsymbol{\delta} = \{0.33304,\ 0.44860,\ 0.18225,\ 0.03159,\ 0.00402,\ 0.00050\}$ [4].

III. Convergence, Failure Rate, and Throughput

This section presents a new convergence analysis for the peeling decoder that operates on the inter-frame layer. For the inter-frame layer, the analysis considers an ensemble of bipartite graphs of infinite length that have a specified degree distribution. The basic peeling algorithm [11] is used to correct binary erasures that are left nodes in a bipartite graph. Following the definitions in [11], define $\lambda(x) = \sum_{i=1}^{d_L} \lambda_i x^{i-1}$ as the left edge degree distribution polynomial with maximum left degree d_L and $\rho(x) = \sum_{i=1}^{d_R} \rho_i x^{i-1}$ as the right edge degree distribution polynomial with maximum right degree d_R. We will explore how the indeterminate x in the degree

[1]More precisely, $\delta(\omega)$ is the probability that the decoder requires ω increments to decode successfully *for the first time*.

distribution polynomials has an interpretation as a probability in the context of the peeling decoder.

In our analysis, the inter-frame layer recovers increments by using a ***generalized peeling decoder*** (GPD). The generalization is from a basic peeling decoder as in Luby et al. [11] where any *single* recovered edge always recovers the erased variable node to the generalized scenario where the number of edges required to recover a variable node is a random variable that can take on values larger than one. More than one edge is required when the VL decoder requires more than one increment to successfully decode. The PMF $\boldsymbol{\delta}$ described in (1) defines the random variable describing the number of required edges.

For the basic peeling decoder of [11], the left nodes are initial transmissions of a single bit over the binary erasure channel (BEC) with erasure probability α. The right nodes are linear combinations of single bits, The right nodes are received error free according to [11]. The basic peeling decoder is a special case of the GPD with PMF $\boldsymbol{\delta}$ described by $m = 2, \delta_0 = 1 - \alpha$, $\delta_1 = \alpha$ and $\delta_2 = 0$.

For the GPD, the left nodes correspond to multiple-bit initial transmissions and the right nodes correspond to linear combinations of multiple-bit increments. Also, unlike [11] the right nodes are not assumed to be received error-free. The actual communication channel over which both the initial transmissions and the increments are transmitted is typically not an erasure channel. In our examples we will use an additive white Gaussian (AWGN) channel.

The analysis in subsections A and B below produces essentially the same results as in [11], [13], [14], and [5], but the equations are derived from simple probability arguments rather than differential equations or And/Or trees. Furthermore, this analysis provides the foundation required for the development in subsections C and D.

A. Basic Peeling Decoder

In this subsection we provide a probabilistic analysis of the basic peeling decoder of [11]. We begin with a bipartite graph B with n_c left nodes and n_i right nodes. Graph B has E edges. Let G be the graph obtained after the deletion of the $(1 - \alpha)n_c$ left nodes that were ***not*** erased by the BEC.

At each decoding step, one right node with degree-one is selected. We refer to this node's single edge as a right-degree-one edge. Let Q_t be the t^{th} right-degree-one edge used to ***remove*** a left node and all of its incident edges from the graph. Define G_t as the graph obtained by removing $Q_1, \ldots, Q_t$ as well as their corresponding left nodes, and the incident edges of those left nodes.

Define $x(t)$ or simply x as the probability that a randomly selected edge from B does not belong to the set $\{Q_1, \ldots, Q_t\}$. This x will turn out to be exactly the indeterminate of the degree distribution polynomials in relevant probability calculations. Because $x(t)$ decreases monotonically and strictly with t, we will sometimes use $x(t)$ and t interchangeably. That is, understanding the evolution of a variable or probability as a function of time t is equivalent to understanding its evolution as a function of x.

We seek $r_1(x)$, which is the fraction as a function of x (or t) of the total number $|\boldsymbol{E}|$ of edges in graph B that are available at time t to be used as a selected right-degree-one edge Q_t to recover a left node. Recall that for the basic peeling decoder, each undecoded left node only requires one edge Q_t to be recovered and hence removed.

We begin the computation of $r_1(x)$ by determining the fraction of edges in B that connect to a degree-i left node in B whose $i-1$ other edges are not in the set $\{Q_1, \ldots, Q_t\}$. This is a requirement for the edge to be in G_t. This fraction is the probability $\delta_1 \lambda_i x^{i-1}$ (or $\alpha \lambda_i x^{i-1}$) of randomly selecting such an edge from B. Note that this edge itself may or may not be in the set $\{Q_1, \ldots, Q_t\}$ (which would exclude it from G_t) and its random selection was independent of whether is was or was not in $\{Q_1, \ldots, Q_t\}$. Summing over all initial left degrees, the total probability of an edge from B having a left node in B whose other edges are not in the set $\{Q_1, \ldots, Q_t\}$ is $\ell(x) = \delta_1 \lambda(x)$.

Now we look independently at the probability that a randomly selected edge from B is connected to a right node that has all its other edges removed at time t. Consider the probability that an edge connects to a right node with degree i in B whose $i - 1$ other edges have been removed at time t. This requires that all of the other $1 - i$ edges incident to the right node have been removed from the graph at time t by the decoding action of some *other* left node. This probability is $\rho_i \left(1 - \ell(x)\right)^{i-1}$. Summing over all initial right degrees, the total probability of an edge from B being connected to a right node that has all its other edges removed at time t is $\rho(1 - \ell(x))$. Note that this includes edges in the set $\{Q_1, \ldots, Q_t\}$.

Because the two previously discussed events are independent, the probability of an edge from B having both properties is the product of the two previous probabilities, and hence is $r_1^*(x) = \ell(x)\rho(1 - \ell(x))$

Now we must explicitly remove the edges in the set $\{Q_1, \ldots, Q_t\}$ from the edges we have identified above. The probability of an edge in B belonging to $\{Q_1, \ldots, Q_t\}$ is $1-x$ by the definition of x, and membership in $\{Q_1, \ldots, Q_t\}$ is independent of whether any other edges incident to its left node are in the set $\{Q_1, \ldots, Q_t\}$. Thus the probability of an edge needing to be removed from the edges already considered is $\ell(x)(1 - x)$. Thus the final probability of a randomly selected edge from B being in G_t and having right degree one in G_t and thus being available to provide an increment in the current iteration is

$$
\begin{aligned}
r_1(x) &= r_1^*(x) - \ell(x)(1 - x) && (2) \\
&= \ell(x)[\rho(1 - \ell(x)) - (1 - x)] && (3) \\
&= \delta_1 \lambda(x)[\rho(1 - \delta_1 \lambda(x)) - (1 - x)], && (4)
\end{aligned}
$$

where (4) is the expression obtained in [11], but the more general (3) facilitates the derivation in the next subsection.

B. Generalized Peeling Decoder

In this subsection we generalize the probabilistic analysis of the basic peeling decoder to the GPD of inter-frame coding. For the GPD, the edges in $\{Q_1, \ldots, Q_t\}$ do not always remove a left node. They provide an increment of redundancy that facilitates an additional decoding attempt that ***may*** allow the left node to be removed.

With this generalization in mind, we compute the probability p_ℓ that an edge $\mathcal{E}$ randomly selected from B has the property that at time t the set of ***other*** edges incident to its left node does not include ***enough*** members of $\{Q_1, \ldots, Q_t\}$ (which correspond to increments) to decode successfully. Recall that the probability that a randomly selected edge from B is not in $\{Q_1, \ldots, Q_t\}$ is x.

Consider the case where the the randomly selected edge $\mathcal{E}$ has a left node that requires ω increments to decode and has initial degree i in B. If $\omega > i-1$, $p_\ell(i,\omega) = 1$. If $\omega = i-1$, $p_\ell(i,\omega) = 1-(1-x)^\omega$ since all ω other edges must provide increments for successful decoding. If $\omega < i-1$,

$$p_\ell(i,\omega) = 1 - \sum_{j=\omega}^{i-1} \binom{i-1}{j} (1-x)^j x^{i-1-j} \tag{5}$$

$$= \sum_{j=0}^{\omega-1} \binom{i-1}{j} (1-x)^j x^{i-1-j}\,, \tag{6}$$

which is simply x^{i-1} for the special case of $\omega = 1$. To summarize

$$p_\ell(i,\omega) = \sum_{j=0}^{\min(\omega,i)-1} \binom{i-1}{j} (1-x)^j x^{i-1-j}\,. \tag{7}$$

Summing over all possible pairs (i,ω) for the selected edge $\mathcal{E}$, the probability that an edge $\mathcal{E}$ randomly selected from B has the property that at time t the ***other*** edges incident to its left node do not induce successful decoding is

$$\ell(x) = \sum_{\omega=1}^{m} \delta_\omega \sum_{i=1}^{d_L} \lambda_i \sum_{j=0}^{\min(\omega,i)-1} \binom{i-1}{j} (1-x)^j x^{i-1-j}\,. \tag{8}$$

Note that the edge $\mathcal{E}$ has left degree i with probability λ_i and requires ω increments to successfully decode with probability δ_ω. The term $\binom{i-1}{j}(1-x)^j x^{i-1-j}$ in (8) appears for all δ_ω for which $\omega > j$. Thus defining $\gamma_j = \sum_{j+1}^{m} \delta_\omega$, $\ell(x)$ is also equal to

$$\ell(x) = \sum_{i=1}^{d_L} \lambda_i \sum_{j=0}^{\min(m,i)-1} \gamma_j \binom{i-1}{j} (1-x)^j x^{i-1-j}\,. \tag{9}$$

In the case of interest where there is a single left degree d_L, the probability is

$$\ell(x) = \sum_{j=0}^{d_L-1} \gamma_j \binom{d_L-1}{j} (1-x)^j x^{d_L-1-j}\,. \tag{10}$$

Following similar arguments as for the BEC above, the final probability of a randomly selected edge from B also being a right-degree-one edge present in G_t and available to provide an increment is

$$r_1(x) = \ell(x)[\rho(1-\ell(x)) - (1-x)]\,, \tag{11}$$

which is identical to (3) except $\ell(x)$ is now as defined in (8).

C. GPD Probability of VL decoding failure ϵ_{GPD} a.k.a. ϵ_{FF}

In [11], $r_1(x)$ determines the decodability of the erasure code given the channel erasure probability $\alpha = \delta_1$. Recall that x represents the probability that a randomly selected edge from B is not in the set $\mathcal{Q} = \{Q_1, \ldots, Q_t\}$. At the start the peeling decoding process ($t = 0$), $x = 1$ because $\mathcal{Q} = \emptyset$. As more edges are removed from the graph, x decreases. When there is no right-degree-one edge in the graph, $r_1(x) = 0$, and decoding will stop. We use x_0 to represent x at this point.

For a basic peeling decoder, removal of any edge connected to a left node will result in the left node being recovered. Thus $r_1(x) > 0$ not only ensures the ability of the decoding to continue, but also the continued recovery of left nodes. The single goal of maintaining $r_1(x) > 0$ guides the design of degree distributions in [11] that guarantee the recovery of the overwhelming majority of left nodes. In other words, we seek degree distributions $\lambda(x)$ and $\rho(x)$ such that $x_0 \to 0$ as blocklength increases, where $r_1(x) > 0$ for $x \in (x_0, 1]$. If $x_0 > 0$, then it could be useful to compute a failure probability for the basic peeling decoder ϵ_{BPD}. However, $\epsilon_{BPD} = 0$ when $x_0 \to 0$, and degree distributions have been identified in [11] for which $x_0 \to 0$. Thus computing ϵ_{BPD} has not been a focus.

For the GPD, however, $\epsilon_{GPD} > 0$ even when $x_0 \to 0$. When a degree-i left node requires $\omega > i$ increments to decode, even if all the right nodes incident to the left node have degree 1 (i.e. all available increments have been provided), the left node will not be able to recover. Hence we must include the probability of VL decoding failure in the overall failure probability ϵ_{GPD}. Furthermore, such instances of VL decoding failure even with all available increments provided means that some right nodes are never able to provide increments, leading to $x_0 > 0$ even as blocklength grows without bound.

Using x_0, we can calculate the probability of failure ϵ_{GPD} for a given code and $\boldsymbol{\delta}$ distribution. The probability of failure is defined as the probability a left node cannot be decoded when $r_1(x) = 0$ (i.e. when no more degree-one right nodes exists in the graph). The analysis here is very similar to $\ell(x)$. The difference is that we now need to consider all the edges incident to a left node instead of all-but-one.

Consider the case where a degree-i left node requires $\omega > 0$ increments to decode. If $\omega > i$, the node will not be able to decode at all. If $\omega \leq i$, the node will not be able to decode if the total number of increments received so far is less than ω, and the resulting probability of failure is

$\sum_{j=0}^{\omega-1}\binom{i}{j}(1-x_0)^j {x_0}^{i-j}$. To summarize, the probability of failure of a degree i left node requiring ω increments is

$$\epsilon_{GPD}(i,\omega) = \sum_{j=0}^{\min(\omega-1,i)} \binom{i}{j}(1-x_0)^j {x_0}^{i-j} \,. \tag{12}$$

As a result, the probability of failure can be calculated as

$$\epsilon_{GPD} = \sum_{\omega=1}^{m}\delta_\omega \sum_{i=1}^{d_L}\Lambda_i \sum_{j=0}^{\min(\omega-1,i)} \binom{i}{j}(1-x_0)^j {x_0}^{i-j} \tag{13}$$

$$= \sum_{i=1}^{d_L}\Lambda_i \sum_{j=0}^{\min(m-1,i)} \gamma_j \binom{i}{j}(1-x_0)^j {x_0}^{i-j} \,, \tag{14}$$

where $\gamma_j = \sum_{j+1}^{m}\delta_\omega$ as in $\ell(x)$, and $\Lambda_i = \frac{\lambda_i/i}{\sum_{j=1}^{d_L}\lambda_i/i}$ is the left *node* degree distribution. In this paper, we also refer to ϵ_{GPD} as ϵ_{FF}, the failure probability of a VL decoder in the feedback-free system.

D. Throughput and average right degree

In order to support a target failure probability ϵ^*_{FF}, more linear combinations of increments must be transmitted than the number of increments that would have been requested by a system using ACK/NACK feedback. These additional transmissions result in a throughput loss for the inter-frame system as compared to the feedback system. This throughput loss is caused by three mechanisms in the decoding process that prevent a linear combination of increments (right node) from providing an increment to a VL decoder (left node):

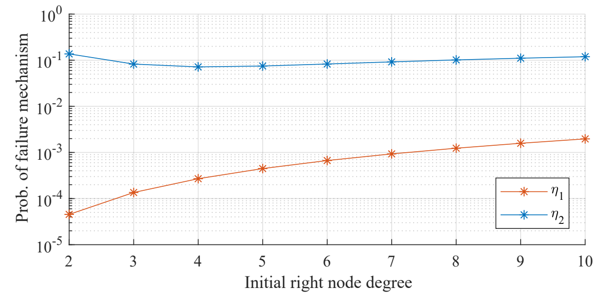
1) The degree of the right node of interest (RNOI) never decreases below two.
2) The degree of the RNOI decreases from two or more to zero in a single iteration of the peeling decoder so that it never provides an increment.
3) The degree of the RNOI achieves the value of one during an iteration so that it provides an increment to a left node, but other right nodes simultaneously provide the remaining required increments to that left node making the RNOI's increment superfluous.

Assign probabilities η_1, η_2 and η_3 to the mechanisms listed above respectively. The probability that a right node does not provide a useful increment to any left node is $\eta = \eta_1+\eta_2+\eta_3$.

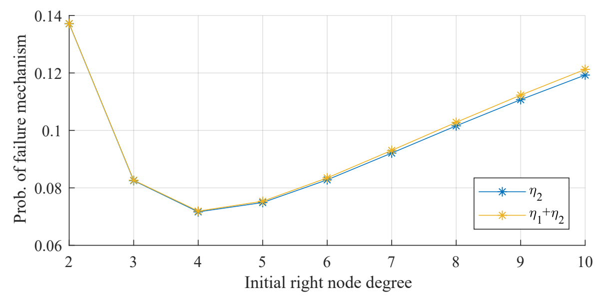
Even if the probability of η were zero, so that every right node provides a useful increment to a left node, in order for the inter-frame code to have the potential to provide the left nodes with the increments they need, the average number $\beta = n_i/n_c$ of right nodes (combined increments) per left node (VL decoder) must be *at least* the expected number of increments required by a VL decoder using ACK/NACK feedback with a maximum of $m-1$ increments, which is

$$\beta_{FB} = \sum_{i=1}^{m-1} i\delta_i + (m-1)\delta_m = E[\delta] - \delta_m \,, \tag{15}$$

where expectation is with respect to the PMF provided in (1).



(a) Probabilities of failure mechanisms as a function of right node degree.



(b) Probability of failure mechanism 1 and the sum of the probabilities of failure mechanisms 1 and 2 as a function of right node degree.

Fig. 3. Probability of failure mechanisms of right nodes for $\rho(x)$ from [4].

In this paper, the degree of left nodes will always be the maximum possible value of $m-1$ to guarantee that each VL decoder has the possibility to receive the maximum number of increments. Throughout this paper, we consider an example where $m=5$, $d_L = m-1 = 4$, and therefore $\lambda(x) = x^3$.

With constant left degree d_L, $\beta = d_L/a_R$, where a_R is the average right degree. The requirement that $\beta > \beta_{FB}$ implies that $d_L/a_R > \beta_{FB}$ which yields an upper bound on the average right degree: $a_R < d_L/\beta_{FB}$. For the VL code in our example from [4], $\beta_{FB} = 0.92598$, and so we require $a_R < 4.31975$. For the irregular $\rho(x)$ in [4], $a_R = 3.69242$.

For a constant value of d_L, the actual average right degree can be approximated using β_{FB} and the probability η that a right node does not provide a useful increment as follows: $a_R = \frac{d_L n_c}{n_i} \approx \frac{d_L(1-\eta)}{\beta_{FB}}$.

Fig. 3 shows the fraction of right nodes of each possible degree that suffer from the first two failure mechanisms for the irregular $\rho(x)$ in [4] according to density evolution analysis. Fig. 3a shows that η_1 is two or more orders of magnitude smaller than η_2 for this degree distribution. Fig. 3b show that the sum of η_1 and η_2 is very similar to the curve for η_2 alone. The probability of mechanism 3 is difficult to quantify, but should be extremely small for codes with good throughput.

As shown in Fig. 3, both high and low right degrees turn out to be undesirable because they are more likely to decrease directly from degree two-or-more to degree zero in a single iteration. This observation leads to the new design method in the next section, where the right degree distribution is quasi-regular.

TABLE I
PERFORMANCE CHARACTERISTICS OF SOME CANDIDATE RIGHT DEGREE DISTRIBUTIONS. $\lambda(x) = x^3$ IN ALL CASES.

$\rho(x)$	a_R	β	iter.	% $R_t^{(FB)}$	ϵ_{FF}
$\rho(x) = x^2$ from [4]	3	1.33333	15	93.52%	7.09×10^{-4}
Irregular $\rho(x)$ from [4]	3.69242	1.08330	32	96.48%	9.24×10^{-4}
Poisson $\rho(x)$ from [11]	3.57594	1.11859	38	96.04%	1.00×10^{-3}

IV. RIGHT DEGREE DISTRIBUTION DESIGN

This section uses the analytical tools discussed above to explore possible right degree distributions $\rho(x)$ in terms of the induced available right-degree-one edges $r_1(x)$ as defined in (11) with $\ell(x)$ as defined in (10) and the induced failure rate ϵ_{FF} according to (14). For this paper we set a target failure rate of $\epsilon^*_{FF} = 10^{-3}$ and seek right degree distributions that would achieve the lowest possible value of β (highest possible a_R) with the least possible number of iterations. The designs in this section are for a 2dB binary-input additive Gaussian channel, and the reference VL system is the same as in [4] with feed-back throughput $R_t^{FB} = 0.528756$. The corresponding $\boldsymbol{\delta}$ is shown in Sec. II.

The lowest possible value of β is sought because this will allow the system to achieve the highest possible percentage of the throughput $R_t^{(FB)}$ of the original feedback system. From the analysis in [4] we can express throughput of the baseline feedback system as

$$R_t^{(FB)} = \frac{k(1-\epsilon_{FB})}{\ell_0 + \beta_{FB}\ell_\Delta}\,, \tag{16}$$

where ℓ_0 is the length of the initial transmission, ℓ_Δ is the length of the increments, k is the number of message symbols, and ϵ_{FB} is the failure probability of the feedback system.

The feedback-free throughput $R_t^{(FF)}$ is computed as

$$R_t^{(FF)} = \frac{k(1-\epsilon_{FF})}{\ell_0 + \beta\ell_\Delta}\,. \tag{17}$$

The number of iterations is determined through an exact density evolution analysis that was found to agree with our actual simulation results as is shown in [4].

We begin by revisiting the two right-degree distributions considered in [4] and a Poisson right degree distribution as suggested in [11]. As shown in Table I the regular distribution had a relatively high β of 4/3, which achieved 93.53% of $R_t^{(FB)}$. The irregular distribution from [4] was found by differential evolution and included right degrees from 1 to 10, with details in [4]. This lowered β to 1.08330 to achieve 96.48% of $R_t^{(FB)}$ but required 32 iterations to complete.

Following [11], we considered a Poisson $\rho(x)$, which we truncated to degrees between 1 and 20. The Poisson distribution's parameter was chosen to minimize β while still achieving $\epsilon_{FF} \leq 10^{-3}$. The Poisson distribution performed well, achieving 96.04% of $R_t^{(FB)}$ with $\beta = 1.11859$ in 38 iterations, but did not outperform the irregular distribution from [4], which was concentrated primarily on degrees 3, 4, and 5.

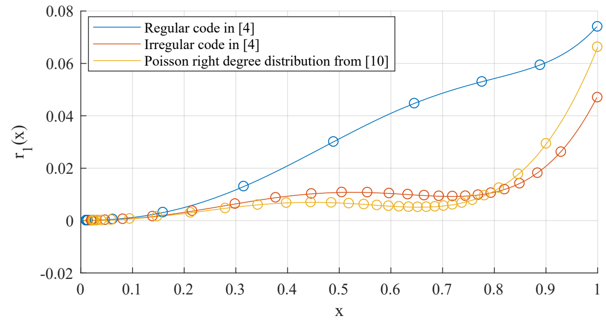


Fig. 4. $r_1(x)$ vs. x for the three choices of $\rho(x)$ in Table I. The curves are generated using (11). Circles indicate iteration points determined through density evolution. The circles at $x = 1$ represent the first iteration.

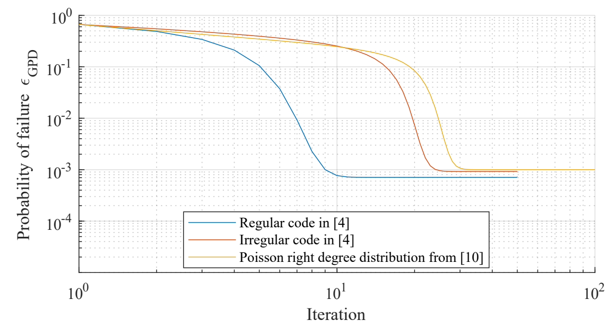


Fig. 5. Probability of failure vs. iterations for $\rho(x)$ distributions in Table I. The curves are generated using density evolution.

Fig. 4 shows the $r_1(x)$ vs. x performance for these three cases and also confirms that (11) matches the performance observed through density evolution. Fig. 5 shows how the probability of failure decreased with the number of iterations.

The derivations in [11] and [5] argue that their proposed heavy-tail Poisson distributions are one way to allow β to approach β_{FB}. However, there may be other distributions that provide good performance in practice. Motivated by [15] and Sec. III-D, we considered the simplest possible distributions: quasi-regular right degree distributions. We first used a quasi-regular right degree distribution of $\rho(x) = \alpha x^2 + (1-\alpha)x^3$ (degree 3 and 4) to explore β values in between these two points, where α is a design parameter. As shown in Table II and Figs. 6 and 7 these distributions did quite well. For example, the $\alpha = 0.244$ distribution outperformed both the irregular distribution from [4] and the Poisson distribution suggested by [11]. If more iterations are possible, over 97% of $R_t^{(FB)}$ can be achieved by $\alpha = 0.108$ with 100 iterations.

We explored even more iterations and three consecutive degrees, as shown in Table III, but even with thousands of iterations, negligible improvement over the 97.08% of $R_t^{(FB)}$ achieved by $\alpha = 0.108$ was seen.

The key difference to note between the $\rho(x)$ found by differential evolution in [4] and the results here is the design complexity. The complexity of finding degree distributions through differential evolution in [4] is very high. The complexity of designing degree distributions based on the quasi-regular

TABLE II
PERFORMANCE CHARACTERISTICS OF QUASI-REGULAR $\rho(x) = \alpha x^2 + (1-\alpha)x^3$. $\lambda(x) = x^3$ IN ALL CASES.

α	a_R	β	iter.	% $R_t^{(FB)}$	ϵ_{FF}
1	3	1.33333	15	93.52%	7.09×10^{-4}
0.531	3.39847	1.17700	20	95.36%	7.82×10^{-4}
0.244	3.69914	1.08133	30	96.52%	8.35×10^{-4}
0.168	3.78788	1.05600	40	96.83%	8.50×10^{-4}
0.139	3.82287	1.04633	50	96.94%	8.56×10^{-4}
0.108	3.86100	1.03600	100	97.08%	8.63×10^{-4}

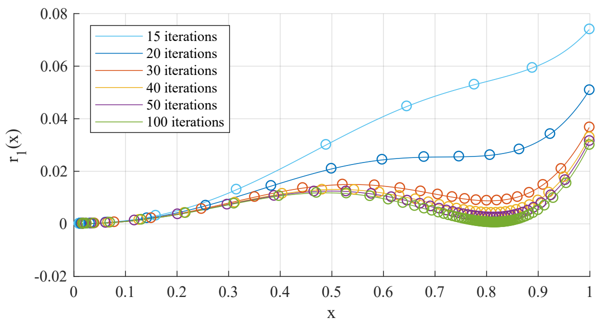


Fig. 6. $r_1(x)$ vs. x for Table II. The curves are generated using (11). Circles indicate iteration points determined through density evolution.

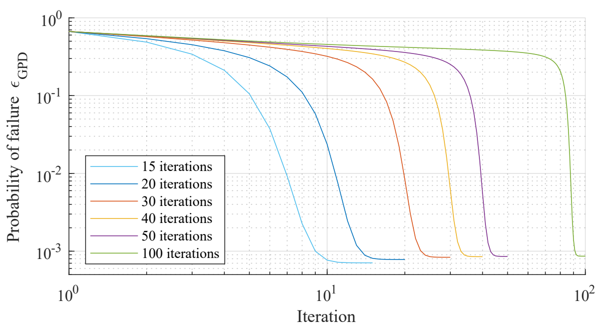


Fig. 7. Probability of failure vs. iterations for $\rho(x)$ distributions in Table II. The curves are generated using density evolution.

heuristic is very low as there is only a single design parameter α for the case with two degrees. This low complexity enables the brute force identification of the exact value of α that achieves a target ϵ_{FF} with the minimum number of iterations. In Table II and the second row of Table III, the resolution of degree distribution coefficients is 0.001. And in the first row of Table III, the resolution is 0.0001.

V. CONCLUSION

This paper considers channel codes that use an inter-frame code to transform a capacity-approaching code that uses incremental redundancy with feedback into a feedback-free system that achieves similar throughput and error performance. Results included an alternative derivation for the right-degree-one process $r_1(x)$, a new expression for the failure probability ϵ_{FF}, and analysis of the mechanisms that lower the throughput of a feedback-free system with respect to the original feedback system. For the practically interesting case where

TABLE III
PERFORMANCE CHARACTERISTICS OF LOWEST-β QUASI-REGULAR RIGHT DEGREE DISTRIBUTIONS. $\lambda(x) = x^3$ IN ALL CASES.

$\rho(x)$	a_R	β	iter.	% $R_t^{(FB)}$	ϵ_{FF}
$0.0998x^2+0.9002x^3$	3.87122	1.03327	1648	97.11%	8.65×10^{-4}
$0.409x^2+0.052x^3+0.539x^4$	3.88903	1.02853	3748	97.16%	9.16×10^{-4}

the left degree distribution is regular, we showed that quasi-regular right degree distributions work well in an absolute sense and better than any alternative considered. This simple construction achieves 97% of the feedback throughput rate while achieving the target failure probability of $\epsilon_{FF} < 10^{-3}$.

REFERENCES

[1] A. R. Williamson, T. Y. Chen, and R. D. Wesel, "Variable-length convolutional coding for short blocklengths with decision feedback," *IEEE Trans. Commun.*, vol. 63, no. 7, pp. 2389–2403, Jul. 2015.

[2] K. Vakilinia, S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "Optimizing transmission lengths for limited feedback with nonbinary LDPC examples," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2245–2257, Jun. 2016.

[3] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Feedback in the non-asymptotic regime," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 4903–4925, Aug. 2011.

[4] H. Wang, S. V. S. Ranganathan, and R. D. Wesel, "Approaching capacity using incremental redundancy without feedback," in *Proceedings of 2017 IEEE International Symposium on Information Theory (ISIT)*, June 2017, pp. 161–165.

[5] H. Zeineddine and M. M. Mansour, "Inter-frame coding for broadcast communication," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 437–452, Feb. 2016.

[6] E. Paolini, M. P. C. Fossorier, and M. Chiani, "Generalized and doubly generalized ldpc codes with random component codes for the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1651–1672, April 2010.

[7] E. Casini, R. D. Gaudenzi, and O. D. R. Herrero, "Contention resolution diversity slotted ALOHA (CRDSA): An enhanced random access scheme for satellite access packet networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 4, pp. 1408–1419, April 2007.

[8] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 477–487, February 2011.

[9] E. Paolini, G. Liva, and M. Chiani, "Coded slotted ALOHA: A graph-based method for uncoordinated multiple access," *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6815–6832, Dec 2015.

[10] E. Sandgren, A. G. i Amat, and F. Brännström, "On frame asynchronous coded slotted aloha: Asymptotic, finite length, and delay analysis," *IEEE Trans. Commun.*, vol. 65, no. 2, pp. 691–704, Feb 2017.

[11] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

[12] H. Wang, N. Wong, A. M. Baldauf, C. K. Bachelor, S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "An information density approach to analyzing and optimizing incremental redundancy with feedback," in *Proceedings of 2017 IEEE International Symposium on Information Theory (ISIT)*, June 2017, pp. 261–265.

[13] N. C. Wormald, "The differential equation method for random graph processes and greedy algorithms," *Lectures on approximation and randomized algorithms*, p. 72, 1999.

[14] M. G. Luby, M. Mitzenmacher, and M. A. Shokrollahi, "Analysis of random processes via and-or tree evaluation," in *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '98. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1998, pp. 364–373.

[15] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 585–598, Feb 2001.