Incremental Redundancy: A Comparison of a Sphere-Packing Analysis and Convolutional Codes

Tsung-Yi Chen
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, California 90034
Email: tychen@ee.ucla.edu

Nambi Seshadri
Broadcom
Irvine, California 92617-3073
Email: nambi@broadcom.com

Richard D. Wesel
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, California 90034
Email: wesel@ee.ucla.edu

Abstract—**Theoretical analysis has long indicated that feedback improves the error exponent but not the capacity of memoryless Gaussian channels. Chen et al. [1] demonstrated that a modified incremental redundancy scheme can use noiseless feedback to help short convolutional codes deliver the bit-error-rate performance of a long blocklength turbo code, but with much lower latency. This paper presents a code-independent analysis based on sphere-packing that approximates the throughput-vs.-latency achievable region possible with feedback and incremental redundancy for a specified AWGN SNR. Simulation results indicate that tail-biting convolutional codes employing feedback and incremental redundancy perform close to the sphere-packing approximation until the throughput reaches the limit of the system's ability to approach the channel capacity.**

I. Introduction

While feedback cannot increase the capacity of a memoryless channel [2], it can significantly reduce the complexity of encoding and decoding at rates below capacity, as shown in the works by Elias [3] and Chang [4] in 1956.

The error-exponent results of [5]–[8] suggest that feedback can be used to reduce latency. As a practical demonstration, [1] showed that using modified incremental redundancy with feedback (MIRF) allowed short convolutional codes to deliver bit-error-rate performance comparable to a long blocklength turbo code, but with lower latency. The demonstration of [1] qualitatively agrees with the error exponent analysis in [5]–[8]. However, the error-exponent theory does not provide a crisp prediction of the quantitative latency benefit possible with MIRF at a specific throughput.

This paper provides the needed quantitative, code-independent analysis of latency vs. throughput that describes the benefit of MIRF over a baseline system. The baseline system has a feedback link but uses it only for ACK/NACK; in other words, it's the simple ARQ scheme. The MIRF system is similar to the one studied in [1]. The analysis uses sphere-packing and bounded-distance decoding to model the behavior of a "good" code for the AWGN channel.

While the sphere-packing analysis is code-independent, it turns out to match well with simulations using "good" short-blocklength codes. Specifically, this paper compares the analysis with simulations of tail-biting convolutional codes. The excellent agreement between the analysis and simulation results indicates both that the analysis provides an accurate characterization and that the short-blocklength codes currently available perform similarly to sphere-packing with bounded-distance decoding.

The rest of the paper is organized as follows: Section II reviews the sphere-packing approximation of decoding error for a "good" code. Sections III and IV use the sphere-packing approximation of Section II to analyze simple ARQ and MIRF, respectively, and compare with simulations. Section V concludes the paper.

II. Sphere-Packing

This section reviews the sphere-packing analysis in [2] for a memoryless AWGN channel and shows that the probability of codeword error of a sphere-packing code with bounded-distance decoding is the complement of the cdf of a chi-square distribution.

Consider a $(2^{nR_c}, n)$ channel code that encodes nR_c bits of information into a length-n codeword with rate R_c. The input and output of the channel can be written as

$$Y = X(i) + Z, i \in 1, 2, \cdots, 2^{nR_c}$$

where Y is the received word, $X(i)$ is the codeword for the i-th message, and Z is a n-dimensional i.i.d. Gaussian vector.

Let the SNR be η and assume without loss of generality that the noise has unit variance. The average power of a received word Y is $P = n(1+\eta)$. Sphere-packing seeks a codebook that has 2^{nR_c} equally separated codewords within the n-dimensional sphere with radius $r_{\text{outer}} = \sqrt{n(1+\eta)}$.

One can visualize a large outer sphere that contains 2^{nR_c} decoding spheres, each with the same radius r_{inner}. An upper bound on the inner sphere radius perfectly packs 2^{nR_c} code spheres into the outer sphere. With this ideal sphere-packing in mind, a conservation of volume argument yields the following

This research was supported by a gift from the Broadcom Foundation. Dr. Wesel has also consulted for the Broadcom Corporation on matters unrelated to this research.

inequality:

$$\begin{aligned}\text{Vol(Inner sphere)} &= K_n \times r_{\text{inner}}^n \\ &\le \frac{\text{Vol(Outer sphere)}}{2^{nR_c}} \\ &= \frac{K_n \times \left(\sqrt{n(1+\eta)}\right)^n}{2^{nR_c}}\end{aligned}$$

where K_n is the spherical volume constant that depends only on n. Solving for the radius of the code spheres yields

$$r_{\text{inner}} \le \frac{\sqrt{n(1+\eta)}}{2^{R_c}} \,. \tag{1}$$

Now consider the bounded-distance decoding rule: if the received word is within r_{inner} of codeword $X(i)$, then declare the output of the decoder to be message i. If the received word is not within r_{inner} of any codeword or is within that distance of multiple codewords, then an error is declared. Nearest-neighbor decoding outperforms bounded-distance decoding, but is more difficult to analyze.

The total noise power is a chi-square with n degrees of freedom. Assuming the largest theoretically possible code sphere radius of (1), the probability of decoding error P_e is

$$\begin{aligned}P_e &= \Pr\left\{\sum_{i=1}^{n} z_i^2 > \left(\frac{\sqrt{n(1+\eta)}}{2^{R_c}}\right)^2\right\} \\ &= 1 - F_{\chi^2(n)}\left(\frac{n(1+\eta)}{2^{2R_c}}\right)\end{aligned}$$

where $F_{\chi^2(n)}(t)$ is the C.D.F. of the chi-square distribution with n degrees of freedom. For the rest of our analysis, we always assume the radius is equal to the upper bound of (1).

III. Analysis of Simple ARQ

Consider the simple ARQ protocol on an AWGN channel with noiseless feedback. The transmitter sends the codeword over the noisy channel and waits for the feedback from the receiver. The receiver will send an ACK/NACK over the noiseless feedback channel to the transmitter if the codeword is decoded successfully/unsuccessfully. If NACK is received at the transmitter, the transmitter will resend the same codeword until an ACK is received. Once an ACK is received, the transmitter encode and transmit a new codeword.

In the previous section we computed the probability of decoding error P_e based on the sphere-packing analysis. With P_e in hand, the expected number of transmissions $\overline{\tau}$ required to communicate a single message using the simple ARQ scheme is as follows:

$$\overline{\tau} = \frac{1}{1 - P_e} \,.$$

Define the throughput as the number of bits transmitted correctly per channel use. The expected throughput R_t of the simple ACK/NACK scheme is given by:

$$R_t = \frac{nR_c}{n\overline{\tau}} = R_c(1 - P_e) = R_c F_{\chi^2(n)}(r_{\text{inner}}^2). \tag{2}$$

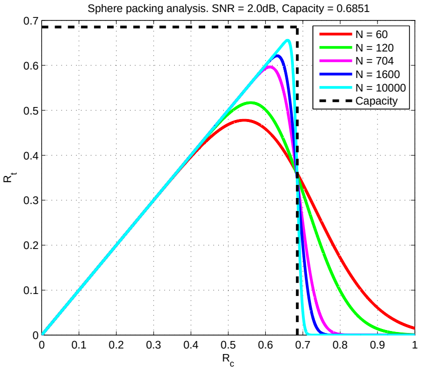

Fig. 1. Sphere-packing analysis of throughput vs. initial code rate for simple ARQ on an AWGN channel with SNR=2 dB and blocklengths ranging from 60 to 10,000.

Define the latency λ as the number of forward channel uses required to communicate a message. The expected latency $\overline{\lambda}$ is given by the product of expected number of transmissions $\overline{\tau}$ and the codeword block length n:

$$\overline{\lambda} = n\overline{\tau} = \frac{n}{1 - P_e} = \frac{n}{F_{\chi^2(n)}\left(\frac{n(1+\eta)}{2^{2R_c}}\right)} \tag{3}$$

Figure 1 shows throughput vs. initial code rate for a sphere-packing analysis of the simple ARQ with SNR=2 dB and blocklengths ranging from 60 to 10,000. Figure 2 compares the sphere-packing analysis of simple ARQ to 64-state tail-biting convolutional code simulations of the simple ARQ with block length 64. This short block length is where the 64-state convolutional code is most effective relative to the sphere-packing limit of performance.

Pseudo-random puncturing (circular buffer rate matching [9]) provides the high rate codes in these simulations. As shown in Figure 3, the encoder generates a rate-$1/3$ codeword. Then the output of each constituent encoder passes through a "sub-block" interleaver. The interleaved bits of each encoder are collected in a buffer and a proper number of coded bits is sent to the transmitter.

Our analysis and simulations fix the initial coded blocklength and vary the code rate. Hence the initial blocklength remains constant and the number of information bits per block increases as the initial rate grows. Therefore, the blocklength of the rate-1/3 mother code increases as the initial rate grows. The power of the convolutional code, however, remains the same as the blocklength of the mother code increases. (We'll discuss this further in section IV-C). This practical restriction differs from our assumption of sphere-packing in which the code becomes more powerful as the blocklength (number of symbols received) increases. This accounts for the disagreement in the high-rate regime of Figure 2.

Through a computational parametric analysis, latency can be examined as a function of throughput in the context of sphere-packing and bounded-distance decoding. Equations (2) and (3) respectively express expected throughput and latency as functions of the initial code rate R_c and block length n.

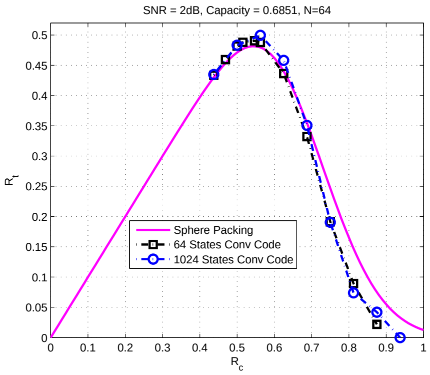

Fig. 2. Throughput R_t vs. initial code rate R_c for sphere-packing analysis and 64-state, 1024-state convolutional code simulations at blocklength 64, SNR= 2 dB for simple ARQ.

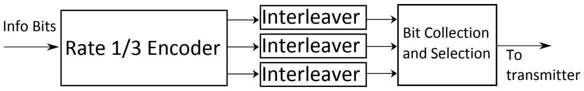

Fig. 3. Pseudo-random puncturing (or circular buffer rate matching) of convolutional code. At the bit selection block, a proper amount of coded bit is selected to match the desired code rate.

IV. Incremental Redundancy with Feedback

This section extends the sphere-packing analysis to examine the latency vs. throughput curve possible with the MIRF scheme of [1] for using incremental redundancy and feedback.

A. An Incremental Redundancy Scheme

Sphere-packing analysis of the MIRF scheme assumes a long and low-rate mother code. The rate of this code can be arbitrary low and its blocklength L can be arbitrarily large. We then pick 2^{nR_c} codewords out of the L-dimensional sphere described in Section II.

The MIRF scheme transmits an initial block length $n < L$ with initial code rate R_c. If the decoding is not successful, the transmitter will receive a NACK and will send s extra symbols. The decoder then attempts to decode again using all received symbols for the current codeword. The process continues until the decoding is successful or the maximum codeword length L is reached. The MIRF scheme can also be interpreted as a rateless coding scheme; the transmitter can send out additional redundancy bits continuously until it receives a ACK message from the receiver.

Let B_i be the vector of the symbols received at the i-th transmission, $B_1 \in \mathbb{R}^n$ and $B_i \in \mathbb{R}^{n_i}$ where $n_i = n+s(i-1)$. Let the power of the noise in the B_i be N_i. Define the event $\zeta_i = \{i\text{-th block cannot be decoded}\} = \{N_i : N_i > r_i^2\}$, where $r_i = \frac{\sqrt{n_i(1+\eta)}}{2^{nR_c/n_i}}$ is the corresponding inner sphere radius at the i-th transmission. The expected latency is computed as follows:

$$\begin{aligned}\overline{\lambda}_{\text{IR}} &= n + s\text{Pr}\left[\zeta_1\right] + s\text{Pr}\left[\zeta_1 \cap \zeta_2\right] + \cdots \\ &= n + s\sum_{i=1}^{m} \text{Pr}\left[\bigcap\nolimits_{j=1}^{i} \zeta_j\right]\end{aligned}$$

where m is the maximum number of transmissions allowed, which is constrained by L.

The joint probability can also be expressed as the product of conditional probabilities. Thus

$$\text{Pr}\left[\bigcap\nolimits_{j=1}^{i} \zeta_j\right] = \prod_{j=1}^{i} P_{e,j}\ , \text{ where } P_{e,i} = \text{Pr}\left[\zeta_i \middle| \bigcap\nolimits_{j=1}^{i-1} \zeta_j\right]$$

The sphere-packing analysis gives

$$P_{e,i} = \text{Pr}\left[\sum_{j=1}^{n_i} z_j^2 > r_i^2 \middle| \bigcap_{j=1}^{i-1} \zeta_j\right] .$$

$P_{e,i}$ is challenging to evaluate since the region of $\bigcap_{j=1}^{i} \zeta_j$ is difficult to characterize. We will approximate $P_{e,i}$ as described below.

B. Approximation of Noise in Successive Decodings

Suppose that the decoder is at the i-th transmission and trying to decode B_i. There are two important mechanisms at play. The first mechanism is that since B_{i-1} was decoded unsuccessfully, we know that B_{i-1} has a noise power larger than r_{i-1}^2. This increased noise power makes $P_{e,i}$ larger than if these $n + s(i-1)$ symbols were decoded as an initial transmission.

The second mechanism is that B_i has the advantage of the s extra symbols received at the i-th transmission, which will increase the radius r_i and thus increase the probability of successful decoding. In short, the code becomes more powerful as the number of symbols received increases according to the second mechanism but decoding becomes more challenging as the previously transmitted symbols are discovered to be noisier than originally hoped according to the first mechanism. The mixture of these two mechanisms must be captured in our analysis.

An optimistic approximation ignores the first mechanism and assumes that every attempt of decoding sees a new instance of noisy symbols with longer blocklength but at the original noise variance. Figures 4 and 5 show plots of this optimistic approximation to compare with the conditional analysis presented below.

The difficulty with properly accounting for the first mechanism is that conditioned on previous decoding failures, the noise is no longer i.i.d. Gaussian. However, we can make a worst-case analysis based on the following two observations.

As shown in [2] and the references therein, the Gaussian distribution is the worst memoryless noise possible given a specified noise power. Lapidoth [10] further showed that irrespective of the noise distribution and even regardless of whether the noise is i.i.d., the capacity assuming i.i.d. Gaussian noise is achievable with nearest-neighbor decoding and no rate above the i.i.d. Gaussian capacity is achievable with random Gaussian coding and nearest-neighbor decoding. Hence, given that our sphere-packing analysis is similar in structure to a Gaussian codebook and that our decoding is similar to nearest neighbor decoding, modeling the noise as i.i.d. Gaussian (with an appropriately computed variance) is a reasonable approximation.

Thus, to account for the first mechanism, we calculate the conditional expectation of the noise power in B_{i-1}. We then model the noise vector of B_{i-1} as i.i.d. Gaussian noise with this conditional expected noise power. To simplify the calculation, we further approximate by conditioning on ζ_{i-1} instead of $\bigcap_{j=1}^{i-1} \zeta_j$.

With that approximation, we are able to calculate the conditional decoding error and analyze the throughput and latency of the MIRF scheme. When the step size is large enough ($s \geq N/10$), the approximation matches well with the simulations on modified MIRF scheme using tail-biting convolutional codes and ML decoding.

Let $I_n(r)$ be the integral of the product of the noise power and the probability density over the complement of the n-dimensional sphere with radius r. The new expected noise power in B_{i-1}, denoted as $\overline{N}_{i-1}$, is

$$\begin{aligned} \mathrm{E}\left[\sum_{j=1}^{n_{i-1}} z_j^2 \middle| \zeta_{i-1} \right] &= \overline{N}_i,\ i = 2, 3 \ldots \\ &= \frac{I_{n_{i-1}}(r_{i-1})}{1 - F_{\chi^2(n_{i-1})}(r_{i-1}^2)} \,. \end{aligned}$$

The details of calculating $I_n(r)$ can be found in [11]. The error probability of the code conditioned on $\bigcap_{j=1}^{i} \zeta_j$ is approximated by the error probability of the same code but with a new noise vector $Z' = [z_1', z_2', \ldots, z_{n_i}'], z_i' \sim N(0, \sigma_c^2)$, where $\sigma_c^2 = \frac{\overline{N}_{i-1}+s}{n_i}$. The radius of the code is $r_i = \frac{\sqrt{n_i(1+\eta)}}{2^{nR_c/n_i}}$

This error probability is the same if we normalize the noise to unit variance and consider a code with radius $r_i' = \frac{\sqrt{n_i(1+\eta')}}{2^{nR_c/n_i}}$, where $\eta' = \frac{n_i \eta}{s+\overline{N}_{i-1}}$. This normalization allows us to calculate the probability with chi-square C.D.F.

We summarize the procedure of computing the conditional error probability for each transmission as follows

1) Calculate the expected noise power of the block B_{i-1} conditioned that B_{i-1} cannot be decoded and denote it as $\overline{N}_{i-1}$.
2) Update the new total noise power by $\overline{N}_{i-1} + s$.
3) Normalize the noise variance to unit variance and update new SNR $\eta' = \frac{n_i \eta}{s+\overline{N}_{i-1}}$.
4) Update the equivalent inner sphere r_i' according to the new SNR η'.
5) The conditional probability is approximated by $P_{e,i} = 1 - F_{\chi^2(n_i)}(r_i'^2)$ where $r_i' = \frac{\sqrt{n_i(1+\eta')}}{2^{nR_c/n_i}}$.

C. Analytic Depth of Convolutional Code

This section explains the performance degradation of the simulation at the high rate regime in terms of the decision depth (or traceback depth) of convolutional code.

The analytic decision depth [12] [13] is the pathlength at which the survivor path incident on the zero state has a path metric that is the unique minimum distance over all survivor path metrics (excluding the all zero path). The optimal decision depth of finite traceback Viterbi algorithm is usually determined by simulation. The analytic decision depth, however, gives a good lower bound on the decision depth. For example, the analytic decision depth of the standard rate $1/2$, 64-state feedforward convolutional encoder is 28. Simulation results show that a decision depth of 35 gives a noticeable performance improvement over 28, and decision depth larger than 35 give only negligible improvement.

Table I shows the profiles of some convolutional codes with different number of states. ν is the number of memory element, d_{free} is the free distance and D_{decision} is the analytic decision depth. Suppose the system is operating under SNR of 2dB and starts out with a 1024-state convolutional code, initial blocklength 128 and initial code rate 0.9 (information bits). To get the overall code rate below the capacity (0.6851), say 0.6, the decoding blocklength have to increase up to 174. This blocklength far exceeds the analytic decision depth of 31 for the 1024-state convolutional code. Once the blocklength is far greater than the analytic decision depth, performance does not improve, unlike the steady improvement provided by our sphere packing assumption.

Table II is the RCPC code profile when pseudo-random puncturing is used to obtain high rate codes. As the rate increases, the d_{free} decreases and the D_{decision} increases. The performance of the convolutional code will also degrade if the block length cannot support the minimum decision depth required. For example, a blocklength 64 convolutional code with initial code rate 0.9 cannot support the needed decision depth of 182.

The above two restrictions on the practical coding system are two causes of the disagreement at the high rate regime for both simple ARQ and MIRF.

TABLE I
PROFILE OF DIFFERENT RATE $1/3$ CONV. CODES

ν	Encoder (Octal)	d_{free}	D_{decision}
6	(133, 171, 165)	15	20
7	(365, 353, 227)	16	25
8	(561, 325, 747)	17	23
9	(1735, 1063, 1257)	20	28
10	(3645, 2133, 3347)	21	31

TABLE II
PROFILE OF RATE COMPATIBLE CONV. CODES

Mother Code:	$\nu = 10$	(3645, 2133, 3347)
Rate	d_{free}	D_{decision}
0.4	16	32
0.5	12	48
0.6	11	62
0.7	8	66
0.8	6	90
0.9	5	182

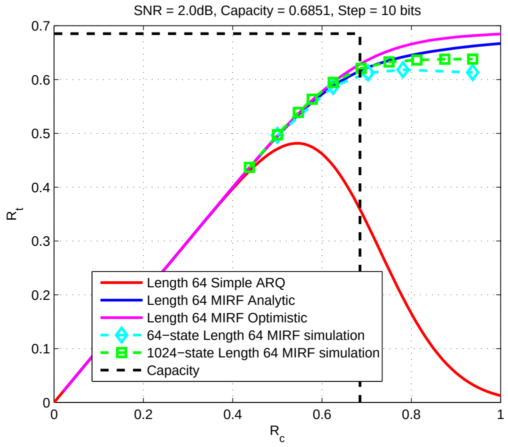

Fig. 4. Throughput versus code rate for a MIRF scheme for step size = 10 bits.

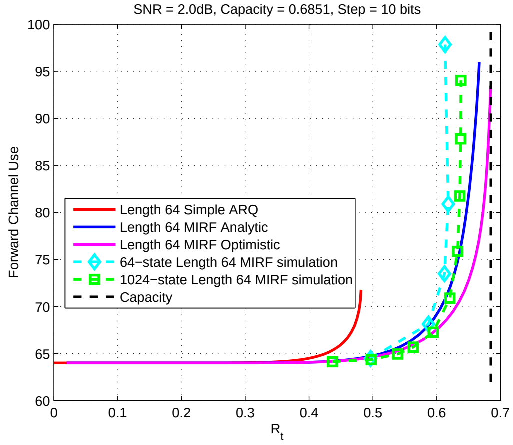

Fig. 5. Latency versus throughput for step size = 10 bits.

D. Comparison with Simulations

Figure 4 shows throughput vs. initial code rate, and Figure 5 shows latency vs. throughput. Both figures show results for sphere-packing analysis and simulations of a tail-biting convolutional code from the [9] and [14]. The lowest code rate for each simulation is $1/3$.

The agreement between analysis and simulation in the low-rate regime is striking. In the high-rate regime, the convolutional codes fall short of the analysis because the throughput has reached the limit of the system's ability to approach the channel capacity. Figure 4 shows that maximum throughput increases from below 0.5 with simple ARQ to above 0.5 with incremental redundancy. Figure 4 also shows that the initial code rate should be higher when using incremental redundancy than when using simple ARQ. Qualitatively, this is obvious. However, Figure 4 indicates how much higher the initial code rate should be. Figure 5 shows that incremental redundancy allows latency to remain low even in the throughput range between 0.4 and 0.5, where simple ARQ does not.

V. Conclusion

It is not surprising that incremental redundancy with feedback can reduce latency. The key result of this paper is a code-independent analysis that is able to accurately determine how much latency reduction and throughput improvement is possible with incremental redundancy of various step sizes. This is a useful tool in system design that was not previously available.

This paper presents a sphere-packing analysis of latency vs. throughput for a baseline simple ARQ scheme and a modified incremental redundancy scheme. This powerful analysis quantifies the latency benefit possible with incremental redundancy and closely predicts the performance of Chen's modified incremental redundancy scheme, validating it as an extremely efficient use of incremental redundancy.

For the simple ARQ scheme, the throughput curves based on the sphere-packing analysis match up well with the simulation results of the convolutional codes when an ML decoder is used. The simulated throughput curves for convolutional codes are match well with the analysis. The suboptimal decoder in our analysis accounts for the disagreement at the maximum throughput region, and inadequate strength of convolutional code when the blocklength increases explains the disagreement at high rate regime.

For the MIRF scheme, further approximations on the conditional probability were made to simplify the analyses. Simulations of the MIRF scheme using convolutional codes show that the approximations are not too far away from practice.

References

[1] T-Y. Chen, B-Z. Shen and N. Seshadri, "Is Feedback a Performance Equalizer of Classic and Modern Codes?" in *ITA Workshop*, San Diego, CA, USA, Feb. 2010.

[2] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 1991.

[3] P. Elias, "Channel Capacity Without Coding," in *MIT Res. Lab of Electronics*, Cambridge, MA, USA, Sep. 1956.

[4] S. Chang, "Theory of information feedback systems," *IEEE Trans. Inf. Theory*, vol. PGIT-2, pp. 29–40, Sep. 1956.

[5] J. Schalkwijk and T. Kailath, "A coding scheme for additive noise channel with feedback–I: No bandwidth constraint," *IEEE Trans. Inf. Theory*, vol. IT-12, no.2, pp. 172–182, Apr. 1966.

[6] J. Schalkwijk, "A coding scheme for additive noise channel with feedback–II: Band-limited signals," *IEEE Trans. Inf. Theory*, vol. IT-12, no.2, pp. 183–189, Apr. 1966.

[7] A. Kramer, "Improving communication reliability by use of an intermittent feedback channel," *IEEE Trans. Inf. Theory*, vol. IT-15, no.1, pp. 52–60, Jan. 1969.

[8] K. S. Zigangirov, "Upper bounds for the error probability for channels with feedback," *Probl. Pered. Inform.*, vol. 6, no.1, pp. 87–92.

[9] 3rd Generation Partnership Project (http://www.3gpp.org), "3GPP TS 36.212 Multiplexing and channel coding."

[10] A. Lapidoth, "Nearest Neighbor Decoding for Additive Non-Gaussian Noise Channels," *IEEE Trans. Inf. Theory*, vol. 42, No. 5, pp. 1520–1529, Sep. 1996.

[11] T-Y. Chen, N. Seshadri and R. D. Wesel , "Sphere-Packing Analysis of Incremental Redundancy with Feedback," in *Proc. Int. Conf. Commun. (to appear)*, Kyoto, Japan, Jun. 2011.

[12] R. D. Wesel, *Encyclopedia of Telecommunications - Convolutional Codes*. John Wiley & Sons Inc., 2003.

[13] J. B. Anderson and K. Balachandran, "Decision Depth of Convolutional Codes," *IEEE Trans. Inf. Theory*, vol. 35(2), pp. 455–459, Mar. 1989.

[14] S. Lin and D. J. Costello, *Error Control Coding*. Pearson Prentice Hall, 2004.