

A Sphere-Packing Analysis of Incremental Redundancy with Feedback

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. Recently, Chen et al. [1] demonstrated that an 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. Such a latency improvement is suggested by the error-exponent analysis, but there is no theoretical work that estimates how much latency improvement is possible with feedback for practical blocklengths and rates. This paper provides a code-independent analysis that quantifies the latency benefits possible by using modified incremental redundancy with feedback (MIRF). A sphere-packing analysis yields the throughput vs. latency performance of both a baseline ARQ scheme and MIRF. The sphere-packing analysis matches well with simulations using turbo and convolutional codes, showing that the analysis has a practical predictive value.

I. INTRODUCTION

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

In 1966, Schalkwijk and Kailath [4] presented a feedback coding scheme for additive white Gaussian noise (AWGN) channels. They described an algorithm that uses an ideal feedback link in a fixed-length block-coding scheme for the infinite bandwidth AWGN channel for which the probability of error decreases such that the error exponent is itself exponential in the code constraint length for rates lower than the capacity. This work was then extended by Schalkwijk himself, Kramer and Zigangirov [5]–[7].

The error-exponent results of [4]–[7] 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 [4]–[7]. However, the error-exponent theory does not provide a crisp prediction of the quantitative latency benefit possible with MIRF at a specific throughput.

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.

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 ARQ. 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 in AWGN channels.

While code-independent, the sphere-packing analysis matches well with simulations using “good” short-blocklength codes. Specifically, this paper compares the analysis with simulations of tail-biting convolutional codes and turbo codes taken from the 3GPP standard [8]. The excellent agreement between the analysis and simulation results suggests 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. The Appendix in Section VI presents mathematical derivations that support Section IV.

II. SPHERE-PACKING

This section reviews the sphere-packing analysis in [9] for a memoryless AWGN channel and shows that the probability of codeword error for a sphere-packing code with bounded-distance decoding is the complement of the cdf of a chi-square distribution.

Consider a $(2^{nR_c}, n)$ codebook that maps 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

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

where Y is the received word, $X(i)$ is the codeword of 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 each noise sample has a 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} “inner spheres”, each with the same radius r_{inner} . The largest inner sphere radius perfectly packs 2^{nR_c} inner spheres into the outer sphere. With this sphere-packing in mind, a conservation of volume argument yields the following inequality:

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

where K_n is the constant of evaluating the volume of the n -dimensional sphere. Solving for the inner sphere radius yields

$$r_{\text{inner}} \leq \frac{\sqrt{n(1+\eta)}}{2^{R_c}}. \quad (1)$$

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 inner 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 optimum radius 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 a 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 encodes and transmits 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 $\bar{\tau}$ required to communicate a single message using the simple ARQ scheme is as follows:

$$\bar{\tau} = \frac{1}{1 - P_e}.$$

Define the expected throughput R_t as the expected number of bits transmitted correctly per packet transmission divided

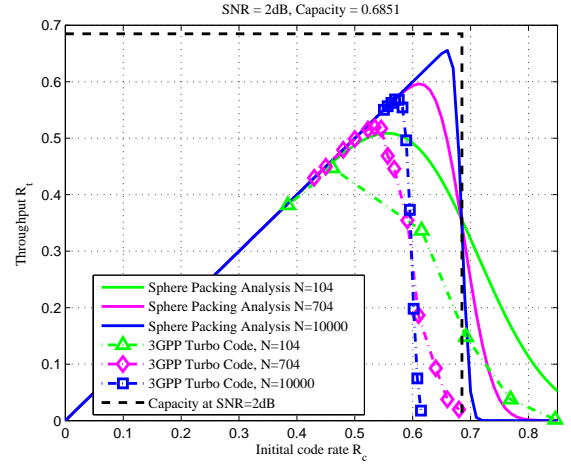


Fig. 1. Throughput R_t vs. initial code rate R_c for sphere-packing analysis and 3GPP turbo code simulations for blocklengths 104, 704 and 10,000 at SNR=2 dB.

by the number of channel uses per packet transmission. For the simple ARQ scheme, R_t is given by:

$$R_t = R_c(1 - P_e) = R_c F_{\chi^2(n)}(r_{\text{inner}}^2). \quad (2)$$

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

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

Figure 1 shows throughput vs. initial code rate for the sphere-packing analysis and 3GPP Turbo code simulations for blocklengths of 104, 704 and 10,000 at an SNR of 2 dB. Each point of a dashed curve represents a different Turbo code with the same blocklength but different code rate. Higher rates are obtained by pseudo random puncturing (circular buffer rate matching [8]). We considered only puncturing patterns that performed reasonably well.

The sphere-packing analysis uses an optimistic r_{inner} but sub-optimal bounded-distance decoding; it is a mixture a optimistic and pessimistic assumptions. Interestingly, after the initial no-codeword-error section where throughput equals code rate, the difference between the analytical curves and the simulated curves for turbo codes is consistently about 0.1 bits despite the large range of blocklengths considered.

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

IV. INCREMENTAL REDUNDANCY

This section extends the sphere-packing analysis to examine the latency vs. throughput curve possible with the incremental redundancy scheme.

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 as described in Section II.

The MIRF scheme starts out by transmitting a punctured version of the mother codeword with initial blocklength $n < L$ and 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.

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:

$$\begin{aligned} \bar{\lambda}_{\text{IR}} &= n + s\Pr[\zeta_1] + s\Pr[\zeta_1 \cap \zeta_2] + \dots \\ &= n + s \sum_{i=1}^m \Pr\left[\bigcap_{j=1}^i \zeta_j\right], \end{aligned}$$

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

$$\Pr\left[\bigcap_{j=1}^i \zeta_j\right] = \prod_{j=1}^i P_{e,j}, \text{ where } P_{e,i} = \Pr\left[\zeta_i \mid \bigcap_{j=1}^{i-1} \zeta_j\right].$$

The sphere-packing analysis gives

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

$P_{e,i}$ 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 noisier than originally hoped according to the first mechanism. The mixture of these two mechanisms must be captured in our analysis.

An optimistic approximation would be to ignore the first mechanism and assume that every attempt of decoding sees a new instance of noisy symbols with longer blocklength but at the original noise variance. Figures 2 and 3 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 [9] 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 Gaussian noise is achievable with nearest-neighbor decoding and no rate above the 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 further simplify the calculation, we further approximate by conditioning on ζ_{i-1} instead of $\bigcap_{j=1}^{i-1} \zeta_j$. We conjecture that this approximation is a lower bound of the conditional expectation.

With that approximation, we are able to calculate the conditional decoding error and analyze the throughput and latency of the MIRF scheme. The conditional expected noise power in B_{i-1} , denoted as \bar{N}_{i-1} , is

$$\bar{N}_{i-1} = \mathbb{E}\left[\sum_{j=1}^{n_{i-1}} z_j^2 \mid \zeta_{i-1}\right] = \frac{I_{n_{i-1}}(r_{i-1})}{1 - F_{\chi^2(n_{i-1})}(r_{i-1}^2)},$$

where $I_n(r)$ is the integral of the product of the noise power and the probability density over the complement of the n -dimensional sphere of radius r centered at the origin. The details of calculating $I_n(r)$ are in the Appendix (Section V).

Applying Lapidoth [10], the error probability of the code conditioned on $\bigcap_{j=1}^i \zeta_j$ is approximated by the error probability of the same code in the face of i.i.d. Gaussian noise with the noise vector $Z' = [z'_1, z'_2, \dots, z'_{n_i}]$, $z'_i \sim N(0, \sigma_c^2)$, where $\sigma_c^2 = \frac{\bar{N}_{i-1} + s}{n_i}$. The decoding radius of the code is $r_i = \frac{\sqrt{n_i(1+\eta)}}{2^{nR_c/n_i}}$.

Normalizing the noise to have unit variance and considering a code with decoding radius $r'_i = \frac{\sqrt{n_i(1+\eta')}}{2^{nR_c/n_i}}$, where $\eta' = \frac{n_i\eta}{s+\bar{N}_{i-1}}$ has the same error probability as the original scenario.

This normalization allows calculation of the probability of error using a 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 \bar{N}_{i-1} .
- 2) Update the new total noise power by $\bar{N}_{i-1} + s$.
- 3) Normalize the noise variance to unit variance and update new SNR $\eta' = \frac{n_i \eta}{s + \bar{N}_{i-1}}$.
- 4) Update the equivalent inner sphere r'_i according to the new SNR η' .
- 5) Approximate the conditional probability as

$$P_{e,i} = 1 - F_{\chi^2(n_i)}(r'_i{}^2), \text{ where } r'_i = \frac{\sqrt{n_i(1+\eta')}}{2^{nR_c/n_i}}.$$

C. Comparison with Simulations

Figures 2 and 3 show throughput vs. initial code rate and latency vs. throughput, respectively, for both the sphere-packing analysis and simulations of tail-biting convolutional codes from [11] and turbo codes from [8].

The agreement between the analysis and the tail-biting convolutional code simulations 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. (See details in [12]).

Comparing systems with an initial blocklength of 64, Figures 2 and 3 show that for latencies near 70 channel uses, simple ARQ achieves a throughput of less than 0.5 while incremental redundancy achieves throughput above 0.6.

These figures also show that the turbo code we simulated saturates at a lower throughput than both 64-state and 1024-state length-64 convolutional codes, even with longer blocklength. We expect that turbo codes with better performance at short blocklengths can be found. Still, the convolutional code performance is quite good in this short-blocklength regime.

As expected, the initial code rate should be higher when using incremental redundancy than when using simple ARQ. Figure 2 indicates how much higher the initial code rate should be and Figure 3 indicates the corresponding latency tradeoff.

Figure 3 shows that incremental redundancy allows latency to remain low for throughputs 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 ARQ scheme and a modified incremental redundancy scheme. This powerful analysis quantifies the latency benefit possible with incremental redundancy and

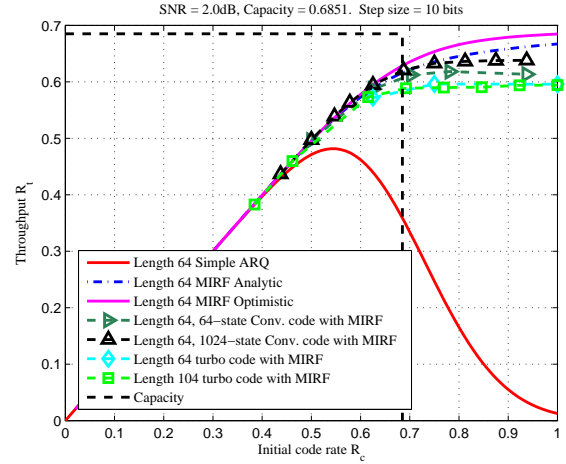


Fig. 2. MIRF and simple ARQ scheme throughput versus initial code rate with step size = 10 bits

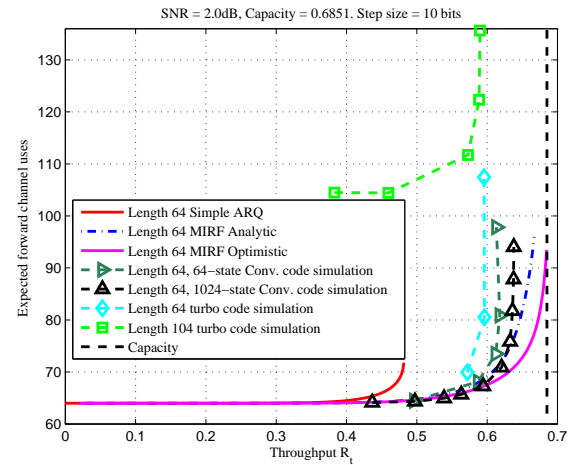


Fig. 3. MIRF and simple ARQ scheme latency versus throughput with step size = 10 bits

closely predicts the performance of the modified incremental redundancy scheme studied in [1], validating it as an extremely efficient use of incremental redundancy.

The simulated throughput vs. latency curves for turbo codes do not match the analysis as closely as the simulated convolutional codes [12], indicating that despite their proximity to the ultimate capacity with a long blocklength, they are not as close to the short-blocklength sphere-packing bound as the simulated convolutional codes. The gaps between the turbo codes simulations and the analytic curves using simple ARQ, however, are consistently close to 0.1 bit.

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

VI. APPENDIX: DERIVATION OF $I_n(r)$

In this section we present the derivation of $I_n(r)$.

A. Hyper-spherical coordinates

We first briefly introduce some preliminaries on hyper-spherical coordinates, which will be helpful in our analysis. A $n+1$ dimensional sphere, or n -sphere with radius r is defined as

$$S_n(r) = \{x \in \mathbb{R}^{n+1} : \|x\|_2 = r\}.$$

Let x_1, x_2, \dots, x_n be the Cartesian coordinates and $R, \phi_1, \phi_2, \dots, \phi_{n-1}$ be the spherical coordinates. The volume element of a sphere in n -dimensional Cartesian space can be converted to spherical coordinates as follows:

$$\begin{aligned} dV &= dx_1 dx_2 \cdots dx_n \\ &= r^{n-1} \sin^{n-2} \phi_1 \sin^{n-3} \phi_2 \cdots \sin \phi_{n-2} dR d\phi_1 \cdots d\phi_{n-1} \\ &= d_{S_{n-1}} V. \end{aligned}$$

Note that the relation $R^2 = x_1^2 + x_2^2 + \cdots + x_n^2$ also holds.

B. Integration Over S_{n-1}^c

Denote the region outside of S_{n-1} as S_{n-1}^c , then $I_n(r)$ is given by the following integration:

$$\iint_{S_{n-1}^c(r)} \cdots \int \left(\sum_{i=1}^n z_i^2 \right) f_Z(z_1, z_2, \dots, z_n) dz_1 dz_2 \cdots dz_n,$$

where $f_Z(z_1, \dots, z_n)$ is the P.D.F. of n -variate i.i.d Gaussian with unit variance. By a change of the coordinates we have

$$\begin{aligned} f_Z(z_1, z_2, \dots, z_n) &= \frac{e^{-\frac{z_1^2 + z_2^2 + \cdots + z_n^2}{2}}}{(\sqrt{2\pi})^n} \\ &= \frac{e^{-\frac{R^2}{2}}}{(\sqrt{2\pi})^n} = f_Z(R). \end{aligned}$$

Let S_{n-1}^c be the volume outside of the $(n-1)$ -sphere. Hence the integral over S_{n-1}^c becomes the following integral:

$$\iint_{S_{n-1}^c(r)} \cdots \int R^{n+1} f_Z(R) \sin^{n-2} \phi_1 \cdots \sin \phi_{n-2} dR d\phi_1 \cdots d\phi_{n-1}.$$

The P.D.F. in spherical coordinates does not depend on the variables $\phi_1, \phi_2, \dots, \phi_{n-1}$ and thus the integrations can be done separately. We separate the integral into two parts: the part with the variable R and the part without the variable R .

1) *Integration of variable R* : We focus on the P.D.F. which depends only on R in this part, and leave the other variable to the second part. We get the following equation using integration by parts:

$$\begin{aligned} &\int_r^\infty R^{n+1} \frac{e^{-\frac{R^2}{2}}}{(\sqrt{2\pi})^n} dR \\ &= \begin{cases} \frac{e^{-\frac{r^2}{2}}}{(2\pi)^{\frac{n}{2}}} [r^n + nr^{n-2} + \cdots + n(n-2)(n-4) \cdots 2] & \text{if } n \text{ is even} \\ \frac{e^{-\frac{r^2}{2}}}{(2\pi)^{\frac{n}{2}}} \left[(r^n + nr^{n-2} + \cdots + n(n-2) \cdots 3r) \right. \\ \quad \left. + n(n-2)(n-4) \cdots 3\sqrt{2\pi} Q(r) e^{\frac{r^2}{2}} \right] & \text{if } n \text{ is odd,} \end{cases} \end{aligned}$$

where $Q(x)$ is the Q -function.

2) *Integration of variables $\phi_1 \cdots \phi_{n-1}$* : Let $\Gamma(x)$ be the gamma function, the integral is given by

$$\begin{aligned} &\int_0^\pi \int_0^\pi \cdots \int_0^{2\pi} \sin^{n-2} \phi_1 \cdots \sin \phi_{n-2} d\phi_1 \cdots d\phi_{n-1} \\ &= \begin{cases} \frac{2\pi^{n/2}}{(\frac{n}{2}-1)!} & \text{if } n \text{ is even} \\ \frac{n\pi^{n/2}}{\Gamma(\frac{n}{2}+1)} & \text{if } n \text{ is odd.} \end{cases} \end{aligned}$$

Combining the two parts, we get the integral over S_{n-1}^c :

$$I_n(r) = \begin{cases} \frac{e^{-\frac{r^2}{2}}}{2^{\frac{n-2}{2}} (\frac{n}{2}-1)!} [r^n + nr^{n-2} + \cdots + n(n-2) \cdots 2] & \text{if } n \text{ is even} \\ \frac{ne^{-\frac{r^2}{2}}}{2^{\frac{n}{2}} \Gamma(\frac{n}{2}+1)} \left[(r^n + nr^{n-2} + \cdots + n(n-2) \cdots 3r) \right. \\ \quad \left. + n(n-2)(n-4) \cdots 3\sqrt{2\pi} Q(r) e^{\frac{r^2}{2}} \right] & \text{if } n \text{ is odd.} \end{cases}$$

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] P. Elias, "Channel Capacity Without Coding," in *MIT Res. Lab of Electronics*, Cambridge, MA, USA, Sep. 1956.
- [3] S. Chang, "Theory of information feedback systems," *IEEE Trans. Inf. Theory*, vol. PGIT-2, pp. 29–40, Sep. 1956.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] K. S. Zigangirov, "Upper bounds for the error probability for channels with feedback," *Probl. Pered. Inform.*, vol. 6, no.1, pp. 87–92.
- [8] 3rd Generation Partnership Project (<http://www.3gpp.org>), "3GPP TS 36.212 Multiplexing and channel coding."
- [9] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 1991.
- [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] S. Lin and D. J. Costello, *Error Control Coding*. Pearson Prentice Hall, 2004.
- [12] T-Y. Chen, N. Seshadri and R. D. Wesel, "Incremental Redundancy: A Comparison of a Sphere-Packing Analysis and Convolutional Codes," in *ITA Workshop*, San Diego, CA, USA, Feb. 2011.