TECHNICAL REPORT 3041
September 2016

Observations on Polar Coding with CRC-Aided List Decoding

David Wasserman

Approved for public release.

SSC Pacific
San Diego, CA 92152-5001

SSC Pacific
San Diego, California 92152-5001

K. J. Rothenhaus, CAPT, USN
Commanding Officer

C. A. Keeney
Executive Director

ADMINISTRATIVE INFORMATION

The work described in this report was performed by the Space Systems Branch (Code 56270) of the ISR Division (Code 56200), Space and Naval Warfare Systems Center Pacific (SSC Pacific), San Diego, CA. The Naval Innovative Science and Engineering (NISE) Program at SSC Pacific funded this Applied Research project.

.

Released by
A. M. Mroczek, Head
Space Systems Branch

Under authority of
C. A. Wilgenbusch, Head
ISR Division

SB

EXECUTIVE SUMMARY

This report may be of interest to researchers in polar codes. It describes some counterintuitive results we have encountered in our polar codes research. The following statements are based on simulation results; we have no proofs, and some of them we are unable to explain:

- The min-sum approximation causes undetected errors in a successive cancellation decoder
- In some cases, an undetected frame error tends to include fewer bit errors than a detected frame error
- Different cyclic redundancy check polynomials of the same length can produce different results.

CONTENTS

Figures

Tables

1. INTRODUCTION

Polar codes are a new type of forward error correction (FEC) codes, introduced by Arikan in [1], in which he proved that they can achieve the capacity of any binary memoryless symmetric (BMS) channel with efficient encoding and decoding.

In fiscal years (FY) 2014 through 2016, Space and Naval Warfare Systems Center Pacific's (SSC Pacific) Naval Innovative Science and Engineering (NISE) program funded the project "More Reliable Wireless Communications Through Polar Codes" to study polar codes, and determine if polar codes can outperform the forward error correction (FEC) currently used and planned for use in Navy wireless communication systems. The project's results from FY14 and FY15 are described in [2, 3]. In FY15 and FY16 we used cyclic redundancy check (CRC)-aided polar list decoding [4]. Section 2 describes the basics of polar coding, and gives details of the encoders and decoders we used.

In the course of our research, we performed simulations of polar codes in hundreds of cases, and we encountered some counterintuitive results. The following statements are based on simulation results; we have no proofs, and most of them we are unable to explain:

- Section 4: The min-sum approximation causes undetected errors in a *successive cancellation* (SC) decoder
- Section 5: In some cases, an undetected frame error tends to include fewer bit errors than a detected frame error
- Section 6: Different CRC polynomials of the same length can produce different results.

2. POLAR CODING

Several versions of polar coding have been published. This section is intended to indicate which versions are used in this work. For background and motivation of this material, see [1] or our previous technical report [2].

For any $n > 0$, we can specify a polar code of length $N = 2^n$ by choosing a subset $\mathcal{A} \subset \{1, \ldots, N\}$. If $\mathcal{A}$ has K elements, we get an (N, K) block code. $\mathcal{A}$ must be chosen well for good error-correction performance. We used the Tal/Vardy method of [5].

The polar encoder uses a row vector $\mathbf{u}$ of length N. Let $\mathbf{u}_{\mathcal{A}}$ be the subvector containing elements whose indices are in $\mathcal{A}$, and let $\mathbf{u}_{\mathcal{A}^C}$ be the subvector containing the remaining $N - K$ elements of $\mathbf{u}$. The encoder constructs $\mathbf{u}$ by filling $\mathbf{u}_{\mathcal{A}}$ with K information bits, and setting $\mathbf{u}_{\mathcal{A}^C}$ to predetermined values. These predetermined values are called *frozen* bits. We follow the usual practice of setting all frozen bits to 0. The encoder outputs $\mathbf{x} = \mathbf{u}\mathbf{F}^{\otimes n}$, where $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, and $\mathbf{F}^{\otimes n}$ is its nth tensor power.

Sometimes polar coding is done using $\mathbf{F}^{\otimes n}\mathbf{\Pi}_N$ instead of $\mathbf{F}^{\otimes n}$, where $\mathbf{\Pi}_N$ is a permutation matrix called the *bit-reversal operator*, defined in Section VII of [1]. We call this *bit-reversed polar coding*. This encoder can be implemented with complexity $O(N \log N)$.

Arikan showed in [1] that polar codes can achieve capacity using an SC decoder. This decoder computes estimates $\hat{u}_1, \ldots, \hat{u}_N$, one at a time, in order, with complexity $O(N \log N)$.

2.1 SYSTEMATIC POLAR CODING

Systematic polar coding was introduced in [6]. The systematic polar encoder also computes $\mathbf{x} = \mathbf{u}\mathbf{F}^{\otimes n}$, but the information bits are found in $\mathbf{x}$ rather than in $\mathbf{u}$. Specifically, [6] showed that for any vector of K information bits, there is a unique $\mathbf{u}$ such that $\mathbf{u}_{\mathcal{A}^C}$ is all 0's and $\mathbf{x}_{\mathcal{A}}$ is the specified information bits.

The systematic SC decoder computes the estimate $\hat{\mathbf{u}}$ in the same way as the non-systematic decoder, and also computes $\hat{\mathbf{x}} = \hat{\mathbf{u}}\mathbf{F}^{\otimes n}$. Then $\hat{\mathbf{x}}_{\mathcal{A}}$ is the desired estimate of the information bits.

Arikan proved in [6] that systematic polar coding has the same frame error rate (FER)[1] as non-systematic polar coding. Arikan also provided simulation results showing that systematic polar coding has a lower bit error rate (BER) than non-systematic polar coding. This result has been replicated, but to our knowledge it has never been proven.

2.2 CYCLIC REDUNDANCY CHECK (CRC)-AIDED POLAR LIST DECODING

CRC-aided polar list decoding was introduced by Tal and Vardy in [4]. This method uses a concatenated code: a CRC is added to the information bits before they are input to the encoder. The list decoder produces multiple candidate polar codewords, and then uses the CRC to help choose the correct codeword. The *list size* is the number of candidate codewords, and is denoted L. We provided a more detailed explanation in Section 3 of [3].

In this work we always use the symbol K_i for the number of information bits per block, and K for the number of non-frozen bits in a polar code. d is the number of CRC bits added, so $K = K_i + d$. The *code rate* is $r = K_i/N$. Table 1 lists all CRC polynomials used in this report.

[1] Also known as block error rate.

Table 1. CRC polynomials used in this report.

Name	Polynomial	Source
CRC-6-ITU	x^6+x+1	[7]
CRC-8	$x^8+x^7+x^6+x^4+x^2+1$	[7]
CRC-9K/3	$x^9+x^7+x^6+x^3+x^2+x+1$	[8]
CRC-10	$x^{10}+x^9+x^5+x^4+x+1$	[7]
CRC-12	$x^{12}+x^{11}+x^3+x^2+x+1$	[7]
CRC-14K/4.1	$x^{14}+x^9+x^8+x^7+x^6+x^4+1$	[8]
CRC-16-IBM	$x^{16}+x^{15}+x^2+1$	[7]
CRC-16F/3	$x^{16}+x^{12}+x^{11}+x^9+x^8+x^5+x^3+x+1$	[8]
CRC-17-CAN	$x^{17}+x^{16}+x^{14}+x^{13}+x^{11}+x^6+x^4+x^3+x+1$	[7]
CRC-18K/4	$x^{18}+x^{15}+x^{13}+x^{12}+x^{11}+x^{10}+x^9+x^7+x^5+x^2+x+1$	[8]
CRC-19K/4	$x^{19}+x^{18}+x^{17}+x^{15}+x^{14}+x^{13}+x^{12}+x^{10}+x^9+x^4+x^3+x^2+x+1$	[8]
CRC-21K/4	$x^{21}+x^{16}+x^{15}+x^{13}+x^{12}+x^{11}+x^{10}+x^9+x^8+x^4+x^3+x^2+x+1$	[8]
CRC-23K/4	$x^{23}+x^9+x^7+x^5+x^3+1$	[8]
CRC-25K/4	$x^{25}+x^{20}+x^{18}+x^{17}+x^{16}+x^{14}+x^{13}+x^{12}+x^{10}+x^6+x^4+x^3+x+1$	[8]

2.3 IMPLEMENTATION DETAILS

We created two different implementations of systematic polar encoding and CA-SCL decoding, which we will call “software A” and “software B.” Both are written in C++.

2.3.1 SOFTWARE A

Software A was written in FY16. It is a non-bit-reversed implementation, mostly based on “Increasing the speed of polar list decoders” ([9]). In particular, the decoder tries SC decoding first, and if the result passes CRC, it outputs this result; otherwise it uses list decoding. Also, at a rate 1 node it considers only four different continuations of each path. Unlike the list decoder in [9], ours computes with log-likelihood ratios (LLRs), a technique introduced in [10]. This technique requires path metrics, which can be computed using a transcendental formula (Equation 11b of [10]) or a piecewise-linear approximation (Equation 12 of [10]). Our decoder uses the approximation. LLRs are computed using the min-sum approximation. LLRs and path metrics are stored as type `float`.

2.3.2 SOFTWARE B

Software B is the same software we used in FY15. It is a bit-reversed implementation. The decoder is based on the pseudocode in [4]. It uses list decoding every time. Unlike most decoders described in the polar coding literature, this decoder does not use the logarithmic domain for probabilities, so it does not need the min-sum approximation. We used type `long double` for the probabilities.

2.4 CODE CONSTRUCTION

All polar codes in this report were constructed with the Tal/Vardy method ([5]) with $\mu = 512$.

3. CHANNEL MODEL

We assume that for each bit $x_i \in \{0, 1\}$ output by the encoder, the corresponding decoder input is

$$y_i = h_i * (-1)^{x_i} + n_i,$$

where h_i is a positive real number called the *gain*, and n_i is a real number called the *noise*. We usually assume h_i is a known constant; in this case we say the channel is *additive white Gaussian noise* (AWGN). We always assume n_i is an independent identically distributed (i.i.d.) sequence of random variables, normally distributed with mean 0 and standard deviation σ. n_i is also assumed to be independent of all h's and x's. We also assume σ is known.

The *bit energy* is $E_b = \mathrm{E}(h_i^2)/r$ where E means expected value. The *noise spectral density* is $N_0 = 2\mathrm{E}(n_i^2) = 2\sigma^2$.

4. THE MIN-SUM APPROXIMATION CAUSES UNDETECTED ERRORS IN AN SC DECODER

We say that a frame error is *detected* if the list decoder finds that none of the codewords in the list pass the CRC. All other frame errors are *undetected*. In particular, if a frame is incorrect when only the SC is used, that frame error is undetected.

The length of the CRC can have a large effect on the performance of CA-SCL. A longer CRC reduces the likelihood of undetected errors. However, this is offset by reducing the number of frozen bits, thus lowering the error-correction capability of the polar code. Section 5.5 of our FY15 report [3] presents extensive simulation results comparing different CRC lengths. We found that the ideal CRC length depends on the list size and the target BER. For a list size of 32 and a BER target 10^{-5}, we got the best results with a length 10 CRC.

However, the results in [3] used software B. As stated in Section 2.3.2, software B has no need for the min-sum approximation, and it uses list decoding every time instead of trying SC first.

When we used software A with CRC-10, we found that a large fraction of the frame errors were undetected errors in the SC decoder. We found that a longer CRC could reduce these errors and improve the BER. Table 2 shows an example. The results in this table were obtained in a simulated AWGN channel with E_b/N_0 0.68 dB, with N = 16384, K_i = 5291, and list size L = 32, using polar codes designed for E_b/N_0 0.409 dB.

Table 2. Comparing CRC lengths with N = 16384 and K_i = 5291 at E_b/N_0 0.68 dB.

CRC	BER	FER	Frame Errors		
			Total	Undetected	Undetected SC
CRC-10	$3.75 \cdot 10^{-5}$	$6.22 \cdot 10^{-4}$	500	273	263
CRC-12	$1.54 \cdot 10^{-5}$	$3.73 \cdot 10^{-4}$	500	120	113
CRC-14K/4.1	$1.00 \cdot 10^{-5}$	$3.32 \cdot 10^{-4}$	500	48	42
CRC-16-IBM	$8.52 \cdot 10^{-6}$	$3.44 \cdot 10^{-4}$	500	7	6
CRC-17-CAN	$7.26 \cdot 10^{-6}$	$2.82 \cdot 10^{-4}$	500	11	9
CRC-18K/4	$8.60 \cdot 10^{-6}$	$2.98 \cdot 10^{-4}$	500	1	1
CRC-19K/4	$1.04 \cdot 10^{-5}$	$3.39 \cdot 10^{-4}$	500	0	0
CRC-21K/4	$9.44 \cdot 10^{-6}$	$3.37 \cdot 10^{-4}$	500	0	0
CRC-23K/4	$1.06 \cdot 10^{-5}$	$3.70 \cdot 10^{-4}$	500	0	0
CRC-25K/4	$1.05 \cdot 10^{-5}$	$3.78 \cdot 10^{-4}$	500	0	0

We also ran a test in which each frame was SC decoded twice, once with the min-sum approximation, and once using the formula $2\,\mathrm{atanh}(\tanh(x/2)\tanh(y/2))$. In this test the CRC was checked after SC decoding but list decoding was never used. Table 3 shows the results obtained in a simulated AWGN channel with E_b/N_0 1.3 dB, with N = 16384, K_i = 5291, and CRC-12, using polar codes designed for E_b/N_0 1.124 dB. These SC decoders were written in MATLAB® and used 8-bit floats for LLRs.

Table 3. Comparing min-sum to the transcendental formula it approximates.

Formula	BER	FER	Total Frame Errors	Undetected Frame Errors
Min-sum	$1.56 \cdot 10^{-3}$	$5.23 \cdot 10^{-2}$	140	58
Transcendental	$6.58 \cdot 10^{-4}$	$3.74 \cdot 10^{-2}$	100	0

This does not mean we should use transcendental formula. It does mean we need to use a longer CRC to prevent these undetected errors. To compensate for the loss of frozen bits, we can increase the list size.

Since the min-sum formula is faster, we can probably get higher throughput, lower BER, and lower FER than we would get with the transcendental formula.

5. IN SOME CASES, AN UNDETECTED FRAME ERROR TENDS TO INCLUDE FEWER BIT ERRORS THAN A DETECTED FRAME ERROR

For each simulation run, we calculated the average number of bit errors per undetected frame error, and the average number of bit errors per detected frame error. We will call these numbers *uBEPFE* and *dBEPFE*, respectively. With software B, we consistently found that uBEPFE $<$dBEPFE when the BER was less than 10^{-5}. Table 4 lists some examples. This is strange because undetected errors are codewords of the polar-CRC concatenated code, which is a subcode of the polar code. In contrast, detected errors are polar codewords that are not codewords of the concatenated code. We expect nonzero codewords in the subcode to have larger Hamming weight. Note however that we count bit errors only among the information bits, so the number of bit errors is generally less than the Hamming weight of the error.

With software A, we found that in most cases uBEPFE $>$dBEPFE. However, in several simulation runs we did find uBEPFE $<$dBEPFE, with a difference too large to be explained by sampling error. Some of these results are also listed in Table 4.

Table 4. Sample runs in simulated AWGN showing uBEPFE $<$dBEPFE.

Software	N	Ki	CRC Polynomial	L	Code Design E_b/N_0 (dB)	E_b/N_0 (dB)	BER	FER	Undetected Frame Errors			Detected Frame Errors		
									Num.	uBEPFE	Std. Error	Num.	dBEPFE	Std. Error
A	1024	864	CRC-17-CAN	32	4.74	4.09	5.14E-06	2.21E-04	110	6.97	0.14	390	23.79	0.60
A	2048	1024	CRC-16-IBM	4	2.11	2.8	2.40E-08	1.61E-06	14	6.00	0.00	147	16.12	0.72
A	16384	8192	CRC-17-CAN	32	1.01	1.25	3.33E-06	2.81E-04	17	14.82	9.20	483	99.91	9.10
B	2048	1024	CRC-16-IBM	4	2.1	2.7	4.45E-08	2.97E-06	6	6.00	0.00	172	15.69	0.55
B	2048	1024	CRC-12	4	2.1	2.6	1.11E-07	6.13E-06	2	8.00	0.00	182	18.57	0.80
B	2048	1434	CRC-6-ITU	4	2.45	3.1	1.99E-06	2.03E-04	276	7.39	0.32	740	16.56	0.40
B	2048	1024	CRC-10	4	2.1	2.7	4.71E-08	3.48E-06	43	7.02	0.28	166	15.60	0.59
B	32768	16384	CRC-10	8	0.76	1.15	7.25E-06	1.82E-03	7	20.86	2.61	357	66.11	5.19
B	32768	16384	CRC-8	8	0.76	1.05	3.48E-05	5.41E-03	128	11.39	0.80	684	18.54	0.44
B	16384	6554	CRC-9K/3	16	0.23	0.9	7.47E-06	1.33E-03	45	21.11	2.34	354	38.83	2.75
B	2048	1024	CRC-10	32	1.65	1.7	1.07E-05	3.01E-04	45	19.84	2.21	556	37.84	1.23

6. DIFFERENT CRC POLYNOMIALS OF THE SAME LENGTH CAN PRODUCE DIFFERENT RESULTS

In Sec. 5.4 of [3] we reported an experiment that compared four CRCs of length 16, and found no difference in the results. We always knew that it was possible for a particular CRC polynomial to be good in general, but interact badly with a particular polar code. We considered this a remote possibility. However, we found a case where increasing CRC length from 14 to 16 did not bring the expected decrease in undetected errors. We tried another polynomial of length 16, and found no undetected errors, as shown in Table 5.

It might appear that this is caused by an intersection of two subspaces having a larger dimension than expected. This is not the case, because the CRC test is applied to the estimated input of the polar encoder, not the output. Regardless of the polar code and the CRC polynomial, there are 2^K possible codewords, and exactly one out of every 2^d will pass CRC.

However, the decoder does not produce codewords randomly. It usually produces codewords that are similar to the correct codeword. Equivalently, $\hat{\mathbf{x}} - \mathbf{x}$ is usually a codeword with lower Hamming weight than the average codeword. Each polar code has a number of low-weight codewords, and the number of these that pass CRC could depend on the choice of polynomial. If many low-weight codewords pass CRC, undetected errors are more likely. Can we test for this? We are not aware of an efficient way to find all low-weight codewords of a polar code. One possibility is to repeatedly input random bits to the encoder, compute the weight of the output, and if it is below some threshold, test it with each candidate CRC. If we use a systematic encoder, the output weight is at least as large as the input weight, so we should bias the random bits toward 0.

This test does not require using the decoder, so it is more efficient than directly testing the decoding performance. We have not had time to run this test.

Table 5 shows an example of a CRC polynomial that is more susceptible to undetected errors than another polynomial of the same length. The results in this table were obtained using software B in a simulated AWGN channel with E_b/N_0 2.85 dB, with N = 2048, K_i = 1024, and list size L = 4, using polar codes designed for E_b/N_0 2.1 dB.

Table 5. Comparing two CRCs of length 16 using software B at E_b/N_0 2.85 dB.

Polynomial	CRC-16-IBM	CRC-16F/3
Number of frames tested	187,860,000	332,980,000
Total frame errors	175	317
Undetected frame errors	15	0
BER	$1.20 * 10^{-8}$	$1.38 * 10^{-8}$
FER	$9.32 * 10^{-7}$	$9.52 * 10^{-7}$

We repeated this test with software A, using the same codes and polynomials, with E_b/N_0 ranging from 2.5 to 2.9 dB. The results are shown in Table 6 and Figure 1. Again we see that CRC-16-IBM is more susceptible to undetected errors.

Although switching from CRC-16-IBM to CRC-16F/3 yields a striking reduction in undetected errors, the resulting coding gain is on the order of 0.01 dB.

Table 6. Comparing two CRCs of length 16 using software A.

	E_b/N_0 (dB)	2.5	2.6	2.7	2.8	2.85	2.9
CRC-16-IBM	Num. frames	2.29E7	5.57E7	1E8	1E8	1E8	1E8
	Frame errors	500	500	384	163	109	78
	Undet. frame errors	16	22	25	14	14	9
CRC-16F/3	Num. frames	2.51E7	5.91E7	1E8	1E8	1E8	1E8
	Frame errors	500	500	346	153	106	74
	Undet. frame errors	3	2	3	1	0	0

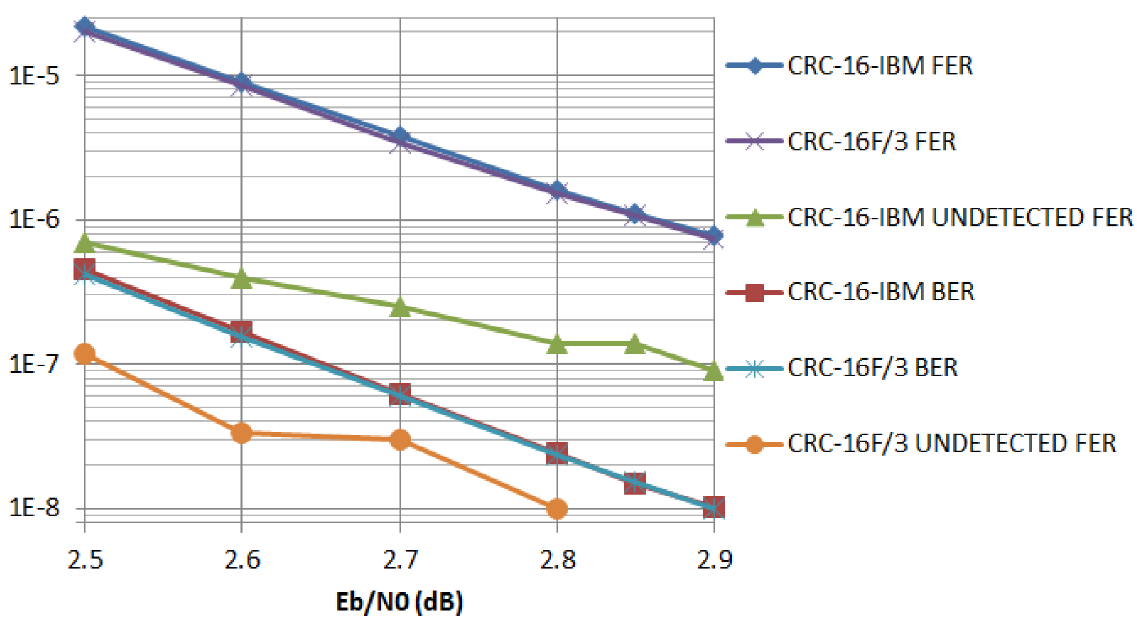

Figure 1. Comparing two CRCs of length 16 using software A.

REFERENCES

1. Arıkan, E. 2009. “Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels,” *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073.

2. Wasserman, D. 2014. “Polar Codes,” Technical Report 2054. Space and Naval Warfare Systems Center Pacific (SSC Pacific), San Diego, CA.

3. Wasserman, D. 2015. “Polar Coding With CRC-Aided List Decoding,” Technical Report 2087. Space and Naval Warfare Systems Center Pacific (SSC Pacific), San Diego, CA.

4. Tal, I., and A. Vardy. 2015. “List Decoding of Polar Codes,” *IEEE Transactions on Information Theory,* vol. 61, no. 5, pp. 2213–2226.

5. Tal, I., and A. Vardy. 2013. “How to Construct Polar Codes,” *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6562–6582.

6. Arıkan, E. 2011. “Systematic Polar Coding,” *IEEE Communications Letters*, vol. 15, no. 8, pp. 860–862.

7. Anonymous. 2016. “Cyclic Redundancy Check - Wikipedia, the free encyclopedia.” Available online at `https://en.wikipedia.org/wiki/Cyclic_redundancy_check`. Accessed September 30, 2016.

8. Koopman, P. 2016. “CRC Hamming Weight Data.” Available online at `http://users.ece.cmu.edu/~koopman/crc/hw_data.html`. Accessed September 30, 2016.

9. Sarkis, G., P. Giard, A. Vardy, C. Thibeault, and W. J. Gross. 2014. “Increasing the Speed of Polar List Decoders,” *2014 IEEE Workshop on Signal Processing Systems (SiPS)* (pp. 1–6). October 2–22, Belfast, Ireland. IEEE.

10. Balatsoukas-Stimming, A., M. Bastani Parizi, and A. Burg. 2015. “LLR-Based Successive Cancellation List Decoding of Polar Codes,” *IEEE Transactions on Signal Processing,*, vol. 63, no. 19, pp. 5165–5179.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-01-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
September 2016	Final	

4. TITLE AND SUBTITLE

Observations on Polar Coding with CRC-Aided List Decoding

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHORS

David Wasserman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

SSC Pacific
53560 Hull Street
San Diego, CA 92152–5001

8. PERFORMING ORGANIZATION REPORT NUMBER

TR 3041

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

SSC Pacific
Naval Innovative Science and Engineering (NISE) Program
53560 Hull Street
San Diego, CA 92152–5001

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report may be of interest to researchers in polar codes. It describes some counterintuitive results we have encountered in our polar codes research. The following statements are based on simulation results; we have no proofs, and some of them we are unable to explain: (1) the min-sum approximation causes undetected errors in a successive cancellation decoder, (2) in some cases, an undetected frame error tends to include fewer bit errors than a detected frame error, and (3) different cyclic redundancy check polynomials of the same length can produce different results.

15. SUBJECT TERMS

Mission Area: Communications
polar codes polar encoder polar code construction cyclic redundancy check polar coding algorithms
list decoding polar decoder Tal/Vardy Method forward error correction

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	U	13	David Wasserman
					19B. TELEPHONE NUMBER *(Include area code)* (619) 553-3003

INITIAL DISTRIBUTION

84300	Library	(1)
85300	Archive/Stock	(1)
56270	D. Wasserman	(1)

Defense Technical Information Center
Fort Belvoir, VA 22060–6218 (1)

Approved for public release.

SSC Pacific
San Diego, CA 92152-5001