A Heuristic Algorithm for Rate-Profiling of Polarization Adjusted Convolutional (PAC) codes

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC0

SUBMISSION DATE / POSTED DATE

05-10-2021 / 18-10-2021

CITATION

Mishra, Samir Kumar; Katyal, Digvijay; Ganapathi, Sarvesha Anegundi (2021): A Heuristic Algorithm for Rate-Profiling of Polarization Adjusted Convolutional (PAC) codes. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.16735351.v1>

DOI

[10.36227/techrxiv.16735351.v1](https://doi.org/10.36227/techrxiv.16735351.v1)

A Heuristic Algorithm for Rate-Profiling of Polarization Adjusted Convolutional (PAC) codes

Samir Kumar Mishra, Digvijay Katyal and Sarvesha Anegundi Ganapathi

Samsung Semiconductor India Research

Bangalore, Karnataka, India

{samir.mishra, d.katyal, sarvesha.ag}@samsung.com

Abstract—In this paper, we propose a heuristic algorithm for rate-profile construction of Arikan’s Polarization Assisted Convolutional (PAC) codes. This method can be used for any blocklength, rate and convolutional precoding polynomial. The proposed algorithm tries to create a rate-profile for which the corresponding PAC code is, in a sense, locally optimized for having maximum possible minimum distance.

Simulation results show that PAC codes constructed with the proposed algorithm perform better in terms of frame erasure rate (FER) compared to the PAC codes constructed with rate profiling designs in existing literature for various list lengths. Further, by using a (64, 32) PAC code as an example, it is shown that the choice of convolutional precoding polynomial can have a significant impact on FER performance. Finally, we demonstrate that for a target FER of 10^{-5} , (128, 72) PAC codes constructed with our proposed algorithm is just 0.35 dB away from the information theoretic limit at this blocklength.

I. INTRODUCTION

Polar codes introduced by Arikan [1] are the first provably capacity achieving codes for the class of binary input memoryless symmetric (BMS) channels with low encoding and decoding complexity of order $O(N \log_2 N)$ for a code of blocklength N . Polar code is based on a phenomenon of channel polarization where a communication channel is transformed into polarized sub-channels: either completely noisy or noiseless. Information bits are transmitted over a set of noiseless sub-channels, while fixed or *frozen* bits are sent over the noisy ones. Polar codes are already being used in 5G New radio (NR) [2] for encoding and decoding of control information.

Polar codes achieve channel capacity asymptotically as the blocklength N of the code approaches infinity. However, for short blocklengths, the performance of polar codes is not good enough. Figure 1 shows the performance of polar code and its variants for a blocklength $N = 128$ and rate $R = 0.5$ for a binary input additive white gaussian noise (BI-AWGN) channel. Figure 1 also shows the BI-AWGN dispersion bound which is the minimum probability of error $\epsilon^*(N, R)$ that can be achieved on a BI-AWGN channel by using a code of blocklength N and rate R under maximum likelihood (ML) decoding. It can be observed clearly that there is a big gap between polar code with successive cancellation decoding (SCD) and the dispersion bound. This poor performance can be partly attributed to poor distance properties of polar codes and also the sub-optimality of SCD as compared to ML decoding [3].

Fig. 1. FER Performance of (128, 64) polar code variants

Since Arikan’s ground breaking work, there have been a lot of efforts to enhance the performance of polar code for short blocklengths, a survey of which can be found in [4]. Specifically, CRC-Aided Polar codes under successive cancellation list (SCL) decoding [3] improve the performance quite a lot. Figure 1 shows the FER performance of a (128, 72) polar code combined with a (72, 64) cyclic code which acts as CRC under SCL decoding with a list size $L = 32$. This approach has been adopted to 5G NR standard and has remained the state of the art ever since.

In the Shannon Lecture at the International Symposium on Information Theory (ISIT) 2019, Arikan presented *polarization adjusted convolutional (PAC) codes*, which are a significant improvement over the state-of-the-art polar codes [5]. Under sequential decoding, at a target FER of 10^{-5} , the FER performance of PAC codes is just 0.25 dB away from the BI-AWGN dispersion bound approximation. Further, it was observed in [6] and [7] that nearly same FER performance can as well be obtained by list decoding as shown in Figure 1.

A. Brief overview of PAC code

The coding scheme of a PAC code is shown in Figure 2. In Figure 2, the solid blocks refer to actual blocks used in the communication system. The dotted blocks refer to the

Algorithm 1: Proposed algorithm for rate-profile construction of PAC code

Input : $N, K, w, p, \rho_d, Z_u, Z_l$
Output: $\mathcal{A}$
 $Z \leftarrow \text{ComputeZBitChannels}(N, K, \rho_d)$
 $\mathcal{A}_{init}, \mathcal{Q} \leftarrow \text{DesignInitialRateProfile}(N, K, Z, Z_u, Z_l)$
 $\mathcal{A} \leftarrow \mathcal{A}_{init}$
while $|\mathcal{A}| < K$ **do**
 $\mathbf{h} \leftarrow \mathbf{0}$
 $\mathbf{c} \leftarrow \mathbf{0}$
 foreach $i \in \mathcal{Q}$ **do**
 $\mathcal{A} \leftarrow \mathcal{A} \cup \{i\}$
 $h_i, c_i \leftarrow$
 $\text{FindNZMinDistCodewords}(N, |\mathcal{A}|, \mathcal{A}, w)$
 $\mathcal{A} \leftarrow \mathcal{A} \setminus \{i\}$
 $h_{min} \leftarrow \min(\mathbf{h})$
 $\mathcal{H} \leftarrow \{k \mid h_k = h_{min}\}$
 $c_{min} \leftarrow \min(\mathbf{c}_{\mathcal{H}})$
 $\mathcal{C} \leftarrow \{k \mid (k \in \mathcal{H}) \wedge (c_k = c_{min})\}$
 $i_{opt} \leftarrow \arg \min_{i \in \mathcal{C}} (Z_i)$
 $\mathcal{A} \leftarrow \mathcal{A} \cup \{i_{opt}\}$
 $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{i_{opt}\}$
return $\mathcal{A}$
subroutine $\text{DesignInitialRateProfile}(N, K, Z, Z_u, Z_l)$:
 $\mathbf{s} \leftarrow \mathbf{0}$
 for $i \leftarrow 0$ **to** $(N-1)$ **do**
 $s_i \leftarrow w(i-1)$ // Calculate RM score
 $\mathbf{r} = \text{sort}(\mathbf{s})$ // Sort in ascending order
 $s_b \leftarrow \mathbf{r}_{N-K}$ // Boundary RM score
 $\mathcal{A}_{init} \leftarrow \emptyset$ // Initial rate profile
 $\mathcal{Q} \leftarrow \emptyset$ // Set of indices with $s_i = s_b$
 for $i \leftarrow 0$ **to** $(N-1)$ **do**
 if $s_i > s_b$ **then**
 $\mathcal{A}_{init} \leftarrow \mathcal{A}_{init} \cup \{i\}$
 else if $s_i = s_b$ **then**
 if $Z_i < Z_l$ **then**
 $\mathcal{A}_{init} \leftarrow \mathcal{A}_{init} \cup \{i\}$
 else if $Z_i \leq Z_u$ **then**
 $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{i\}$
 return $\mathcal{A}_{init}, \mathcal{Q}$

rate profile and denote it by $\mathcal{A}_{init}$. Third, we have the set of indices whose RM score is equal to the boundary RM score. Among these indices, the ones whose Bhattacharyya parameter $W_N^{(i)}$ is less than Z_l are allocated to $\mathcal{A}_{init}$. The indices whose Bhattacharyya parameter $W_N^{(i)}$ is greater than Z_l but less than Z_u are allocated to a new set $\mathcal{Q}$. It is to be noted that $|\mathcal{A}_{init}| \leq K$. The pseudocode for the first phase has been provided as subroutine **DesignInitialRateProfile** in Algorithm 1.

An obvious simplification happens for some PAC codes

when the number of indices with RM score greater than the boundary RM score is equal to K . A typical example of this is the (128, 64) PAC code. In this case, $|\mathcal{A}_{init}| = K$ and our rate-profile construction algorithm falls back to the famous RM rate-profile as described by Arikan in [5].

In the second phase, the rate-profile construction algorithm selects the remaining $(K - |\mathcal{A}_{init}|)$ indices from available $|\mathcal{Q}|$ indices in order to create a (N, K) code. Here, the algorithm selects indices from $\mathcal{Q}$ such that the resulting $(N, K, \mathcal{A}, w)$ PAC code has maximum possible minimum distance. Searching for the best combination among possible $^{|\mathcal{Q}|}C_{(K-|\mathcal{A}_{init}|)}$ combinations is overwhelming in terms of computational resources. For example, let us consider a (64, 32) PAC code for design SNR 3 dB. Here, we have a boundary RM score of 3. $|\mathcal{A}_{init}| = 22$, $|\mathcal{Q}| = 20$ and $(K - |\mathcal{A}_{init}|) = (32 - 22) = 10$. Thus, the second phase algorithm has to select 10 indices from available 20 indices, which can be done in $^{20}C_{10} = 184756$ ways. Similarly, for a (128, 72) PAC code, the search space is $^{35}C_8 = 23535820$ which is obviously huge.

In order to reduce the search space, we make an important assumption as follows. Let $\mathcal{A}_{\tilde{K}}$ denote the set of information bit indices for a code of rate $\tilde{K}/N$. Then $\mathcal{A}_{\tilde{K}} \subset \mathcal{A}_{(\tilde{K}+1)}$. In other words, if we select a bit index to carry information bit for a code of rate $\tilde{K}/N$, then we can select that bit index as well for a code of rate $(\tilde{K} + 1)/N$. Thus, in the second phase, our proposed algorithm runs for $(K - |\mathcal{A}_{init}|)$ iterations. In each iteration, it selects one bit index from $\mathcal{Q}$ for which the corresponding $(N, |\mathcal{A}|, \mathcal{A}, w)$ PAC code achieves minimum hamming distance property.

The second phase of the rate-profile construction algorithm is mentioned as the main routine in **Algorithm 1**. First, the main routine receives $\mathcal{A}_{init}$ and $\mathcal{Q}$ from subroutine **DesignInitialRateProfile**. Then, the algorithm runs for $(K - |\mathcal{A}_{init}|)$ iterations till $|\mathcal{A}|$ reaches the value of K , i.e. a code of rate K/N is achieved. In each iteration, two vectors $\mathbf{h}$ and $\mathbf{c}$ are initialized to all zero vector. $\mathbf{h}$ is required to store the first non-zero (NZ) minimum distance while $\mathbf{c}$ is required to store the number of codewords at this minimum distance. Now, each bit index of $\mathcal{Q}$ is selected and added to the current information bit index set $\mathcal{A}$. For, the corresponding $(N, |\mathcal{A}|, \mathcal{A}, w)$ PAC code, we run the method of counting low weight codewords mentioned in [9]. Function **FindNZMinDistCodewords** uses this method to calculate minimum hamming distance h_i and number of codewords c_i at distance h_i . In order to select a bit index i_{opt} to add to the set $\mathcal{A}$, firstly, the set of bit indices $\mathcal{H}$ which have minimum hamming distance h_{min} is selected. Then, among the bit indices in $\mathcal{H}$, the set $\mathcal{C}$ of bit indices which have minimum number of codewords at hamming distance h_{min} is selected. Among the indices in $\mathcal{C}$, the index with the minimum value of Bhattacharyya parameter Z_i is selected as i_{opt} . i_{opt} is added to $\mathcal{A}$ and its cardinality $|\mathcal{A}|$ is increased by 1. Also, i_{opt} is removed from $\mathcal{Q}$ and its cardinality $|\mathcal{Q}|$ is decreased by 1. The algorithm terminates when $|\mathcal{A}| = K$.

Fig. 3. FER performance of (64, 32) polar code and PAC code variants with $\mathbf{w} = [1, 1, 1, 1, 0, 0, 1]$ under SCL decoding with $L = 8$

Fig. 5. FER performance of (64, 32) CRC-aided polar vs. PAC code variants with $\mathbf{w} = [1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1]$ under SCL decoding with $L = 32$

Fig. 4. FER performance of (64, 32) polar code and PAC code variants with $\mathbf{w} = [1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1]$ under SCL decoding with $L = 8$

Fig. 6. FER performance of (128, 72) polar code and PAC code variants with $\mathbf{w} = [1, 0, 1, 1, 0, 1, 1]$ under SCL decoding with $L = 8, 32, 128$

III. SIMULATION RESULTS AND DISCUSSION

For simulation, we first consider a (64, 32) PAC code transmitted over a BI-AWGN channel for which the rate-profile $\mathcal{A}$ was constructed using the proposed method. The convolutional precoding polynomial used is $\mathbf{w} = [1, 1, 1, 1, 0, 0, 1]$. The design SNR ρ_d is 3 dB. The minimum hamming weight of this code is 8 and there are 8 codewords with this weight. Figure 3 shows the FER performance of polar code and PAC code variants in existing literature compared with that of the PAC code constructed using our proposed Algorithm 1. For each code, SCL decoder with list size $L = 8$ was used. We first observe that PAC code with rate-profile based on 3GPP 5G NR specification [2] only marginally improves the FER performance. The algorithm proposed in [10] performs much better than the current state-of-the-art 5G NR rate

profile. Further, it can be easily observed that the PAC code constructed with our proposed Algorithm 1 outperforms all contemporary PAC code constructions at all values of SNR. Specifically, at a target FER of 10^{-5} , the performance of our code is around 0.3 dB better than that presented in [10].

Figure 4 shows (64, 32) PAC code variants transmitted over a BI-AWGN channel, but using a convolution precoding polynomial given by $\mathbf{w} = [1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1]$. The design SNR ρ_d is 3 dB. For each code, SCL decoder with list size $L = 8$ is used. In this case, PAC code constructed with our proposed Algorithm 1 marginally outperforms the PAC code constructed with algorithm in [10] for the range of SNR below 5 dB. PAC code constructed with both constructions performs nearly same at high SNRs beyond 5 dB.

Figure 5 shows (64, 32) PAC code variants with same

TABLE I
RATE PROFILES

(N, K)	Precoder $\mathbf{w}$ (binary)	Rate Profiles $\mathcal{A}$ (hexadecimal)
(64, 32)	1111001	0015037F035F17FF
	11010001001	0013117F135717FF
(128, 72)	1011011	0001011F01771F7F131737FF177F7FFF
(256, 128)	11010001001	00000001000101170001017F017F7FFF0007037F033F7FFF033F7FFF7FFFFF

Fig. 7. FER performance of (256, 128) polar code and PAC code variants with $\mathbf{w} = [1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1]$ under SCL decoding with $L = 8, 32$

parameters as those shown in Figure 4. However, we have used SCL decoder with list size $L = 32$. Further, we compare PAC code variants with the current state-of-the-art 8-bit CRC-Aided (64, 32) Polar code used in 3GPP 5G NR which contains 24 information bits and 8 CRC-bits. It can be observed that both the PAC code variants perform nearly same. Specifically, at a target FER of 10^{-5} , the performance of our both the codes is around 0.6 dB better than the CRC-Aided Polar code.

From the above mentioned results, it can be observed that PAC codes constructed with rate-profiling Algorithm 1 perform good for a wide range of list sizes. Further, it should be noted that, when using Algorithm 1, the choice of the convolutional precoding polynomial $\mathbf{w}$ can have a significant impact on the performance of the PAC code. Considering a target FER of 10^{-5} , it can be noted that the PAC code constructed with Algorithm 1 for $\mathbf{w} = [1, 1, 1, 1, 0, 0, 1]$ performs around 0.2 dB better than the PAC code constructed with Algorithm 1 for $\mathbf{w} = [1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1]$.

Figure 6 shows the FER performance of (128, 72) PAC code transmitted over a BI-AWGN channel for which the rate-profile $\mathcal{A}$ was constructed using Algorithm 1. The convolutional precoding polynomial used is $\mathbf{w} = [1, 0, 1, 1, 0, 1, 1]$.

The design SNR ρ_d is 3 dB. The minimum hamming weight of this code is 12 and there are 684 codewords with this weight. It can be easily observed that with increasing list length L of the SCL decoder, FER performance of the (128, 72) PAC code improves greatly. Specifically, at a target FER of 10^{-5} , when decoded with a SCL decoder with list length $L = 128$, the FER performance of this PAC code is just 0.35 dB away from the BI-AWGN dispersion bound approximation.

Figure 7 shows the comparison of FER performance of (256, 128) PAC code constructed using Algorithm 1 versus the PAC code constructed using the algorithm mentioned in [10]. The convolutional precoding polynomial used is $\mathbf{w} = [1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1]$. The design SNR ρ_d is 3 dB. It can be observed that our PAC code performs better than the one in [10] at all range of SNR. It is worth noting here that our PAC code construction decoded with a SCL decoder having list size $L = 8$ performs better than the one in [10] decoded with a SCL decoder having list size $L = 32$. Specifically, at a target FER of 10^{-5} , it is at-least 0.2 dB better.

Table I provides the rate-profiles that were obtained by Algorithm 1 and used to generate the simulation results just described.

IV. CONCLUSION AND FUTURE WORK

In this paper, we presented a heuristic algorithm for rate-profile construction of Arıkan's PAC codes. We demonstrated that the PAC codes constructed by using the proposed algorithm perform better in terms of FER compared to the PAC codes constructed with rate-profile designs in existing literature. It was shown how the choice of the convolutional precoding polynomial can affect the performance of PAC code. Finally, it was shown how the PAC codes constructed with our proposed algorithm can approach the information theoretic limits.

There are many open problems to ponder over. First, we note that the proposed heuristic algorithm is kind of a greedy algorithm and may terminate at a locally optimal solution. Sometimes, this might lead to poor construction of PAC codes. Is it possible to create a rate-profile construction algorithm using machine learning or other search algorithms that will provide a better performance? Second, for a given code rate and blocklength, does there exist a convolutional precoding polynomial for which the FER at a given SNR is minimum?

Third, the rate-profile construction algorithms in current literature including the proposed algorithm in current paper do not consider rate-matching [2] schemes which are needed for practical application of PAC codes. We consider these problems for future work.

REFERENCES

- [1] E. Arıkan, "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, 2009.
- [2] 3GPP, "NR; Multiplexing and channel coding," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.212, 01 2018, version 15.0.0. [Online]. Available: <http://www.3gpp.org/DynaReport/38212.htm>
- [3] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213–2226, 2015.
- [4] M. C. Coşkun, G. Durisi, T. Jerkovits, G. Liva, W. Ryan, B. Stein, and F. Steiner, "Efficient error-correcting codes in the short blocklength regime," *arXiv preprint arXiv:1812.08562*, 2018.
- [5] E. Arıkan, "From sequential decoding to channel polarization and back again," *arXiv preprint arXiv:1908.09594*, 2019.
- [6] H. Yao, A. Fazeli, and A. Vardy, "List decoding of arıkan's pac codes," *arXiv preprint arXiv:2005.13711*, 2020.
- [7] M. Rowshan, A. Burg, and E. Viterbo, "Polarization-adjusted convolutional (pac) codes: Fano decoding vs list decoding," *arXiv preprint arXiv:2002.06805*, 2020.
- [8] S. K. Mishra and K. Kim, "Selectively precoded polar codes," *arXiv preprint arXiv:2011.04930*, 2020.
- [9] B. Li, H. Shen, and D. Tse, "An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check," *IEEE Communications Letters*, vol. 16, no. 12, pp. 2044–2047, 2012.
- [10] M. Moradi and A. Mozammel, "A monte-carlo based construction of polarization-adjusted convolutional (pac) codes," *arXiv preprint arXiv:2106.08118*, 2021.
- [11] H. Li and J. Yuan, "A practical construction method for polar codes in awgn channels," in *IEEE 2013 Tencon - Spring*, 2013, pp. 223–226.