STU
SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

Journal of Electrical Engineering, Vol. 77, No. 5, 2026, pp. 529-535
https://reference-global.com/journal/JEE

Proof of concept for DNA storage system with noise-guessing decoder

Daniel Hudač[1], Peter Farkaš[2]

DNA storage has significant advantages if compared with existing electronic storage systems, usually concentrated in data centers, which form digital cloud infrastructure. These are: longevity, format used by nature, high capacity per volume and small CO_2 footprint. Information which is stored and later retrieved from DNA storage systems can be corrupted by insertions, deletions and substitutions (INDELS). Oxford nanopore sequencing is probably the closest to making DNA storage a mass product. On the other hand, different decoding systems based on noise-guessing paradigm also possess advantages when compared with classical error control coding, which has been demonstrated for wide variety of communications channels. Therefore, in this manuscript a proof of concept is given for a DNA storage system with a decoder inspired by the noise-guessing paradigm approach. It is given using a simple model for INDELS obtained from real measured data.

Keywords: DNA memory system, DNA nanopore sequencing, decoding, noise-guessing paradigm, insertion, deletion, substitution, independent and identically distributed impairments model

1 Introduction

By 2029, approximately 19.3 ZB of storage capacity is expected to be installed worldwide [1]. This will cause environmental problems. For example, in 2030 just in the USA it is expected that data-center emissions will reach 44 million metric tons of CO_2-equivalent annually [2].

A future storage technology that provides extremely high density and very low energy/embodied carbon could potentially have a substantial environmental advantage. DNA storage systems can partially help to remedy these problems at least for archival applications [3]. One of the promising technologies is the Oxford nanopore sequencing system. The stored information is encoded into the four nucleotides – Adenine (*A*), Cytosine (*C*), Guanine (*G*) and Thymine (*T*). The name nanopore sequencing is used because during information retrieval the DNA strand is pulled through a nanopore and the measured signal depends on the nucleotides which are inside the pore at the actual moment in time. Despite the fact that, different clever methods are used to deduce the order of nucleotides in the originally stored DNA string, inevitably some impairments arise. To correct them different error correcting methods and codes are used [4-10].

However, error correcting codes (ECC) have to be adapted to the payload size and particular ECC. For example, block codes such as, for example, Reed Solomon (RS) codes have different correcting capabilities for different codeword lengths n, different number of information symbols k in the codeword and depend also on the selection of the finite field over which the RS code is constructed. For each selection of these parameters the RS code requires a different decoder.

Recently decoding based on a noise-guessing paradigm was discovered [11]. The decoder built using it is universal. When the parameters k and n change it is only necessary to recalculate the probabilities of noise patterns, and based on this reorder them in the lists used for guessing. There are numerous recent publications which prove the usefulness of the mentioned paradigm in communications channels [12-30].

However, to the best of authors' knowledge there is not a publication proposing to use a noise-guessing strategy in DNA storage systems. Therefore, in this manuscript a proof of concept is given that this method can be used also in channels representing nanopore DNA storage system based on real measured data from [31].

The rest of this paper is organized as follows. In Section 2 a simple probabilistic model for representing the DNA storage system impairments based on measured data from [31] is presented. In Section 3 the encoding and in Section 4 the decoding of data for a fictional simplified DNA storage system is presented

[1] ESET, spol. s.r.o. Aupark Tower, 16th Floor, Einsteinova 24, 851 01 Bratislava Slovakia
[2] Senior Life Member IEEE
dano.hudac@gmail.com, p.farkas@ieee.org

https://doi.org/10.2478/jee-2026-0050, Print (till 2015) ISSN 1335-3632, On-line ISSN 1339-309X
© This is an open access article licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives License (http://creativecommons.org/licenses/by-nc-nd/4.0/).

with a decoder based on the noise-guessing paradigm. In Section 5 simulations of this system and its results are presented. In Conclusions, following a summary of this paper's material, some further directions are sketched which can improve the decoder based on noise-guessing paradigm for nanopore DNA storage systems.

2 Simple INDEL model

Based on the measured data for Oxford nanopore storage system [31] a simple independent and identically distributed (i.i.d.) impairments model was realized in MATLAB environment, which is illustrated in Fig. 1.

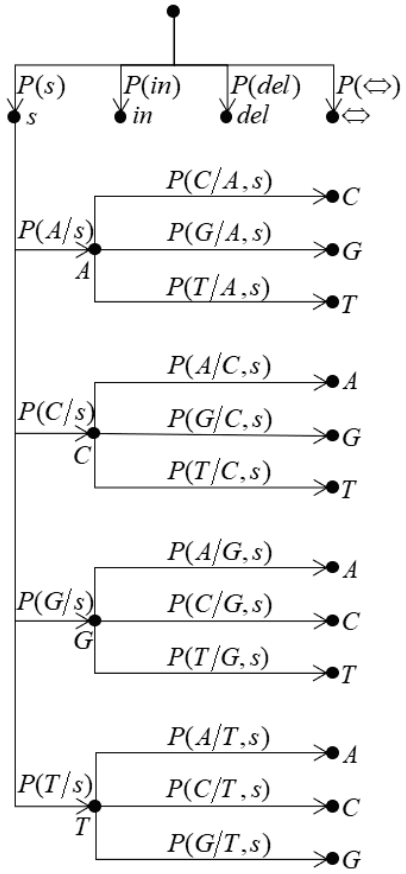


Fig. 1. Impairment model illustration

For the convenience of the reader, Tab. 1 gives probabilities for particular types of impairments which were approximated from frequencies obtained by real measurements in [31]. Table 2 contains the percentage of particular substitutions for each nucleotide which were also obtained by measurements of real data in [31].

Based on data from [31] shown in Tab. 2 the approximate conditional probabilities in Fig. 1 can be computed. The results are summarized in Tab. 3.

Table 1. Approximate probabilities obtained based on measured data in [31]

Event name	Approximated probability based on
Insertion	P_{in}=0.0216
Deletion	P_{del}=0.0295
Substitution	P_s=0.0175
Any impairment	P_{imp}=0.0687
Impairment absence	$P_{\leftrightarrow}$=0.9313

Table 2. Percentage of particular substitutions obtained from measured data in [31]

	A	*C*	*G*	*T*
A		4.3	19	3.3
C	5.6		2.5	13
G	12	2.3		4.7
T	3.5	26	3	

3 Encoding

In this Section the encoding is briefly described. The binary sequence with length k representing payload data is systematically encoded using a CRC denoted as CRC-x, where the x denotes the redundancy expressed in bits of the used CRC. For example, the CRC-10 used was:

$$g(x) = x^{10} + x^9 + x^5 + x^4 + x + 1 \quad (1)$$

If this polynomial is used, then the codeword length is:

$$n = k + 10 \quad (2)$$

The resulting binary sequence which represents the codeword is divided into consecutive fragments - each containing a pair of consecutive bits. Each different pair is mapped into a distinct nucleotide using a rule: 00 = *A*, 01 = *C*, 10 = *G*, 11 = *T*.

The sequence of these letters which represent nucleotides of DNA is a codeword stored in a hypothetical DNA storage system. Its block length is

$$N = n/2 \quad (3)$$

After corruption in accordance with the model illustrated in Fig. 1 this codeword is later used as a "received" sequence from which the decoder has to retrieve, if possible, the uncorrupted DNA codeword and consequently the uncorrupted original binary payload.

Table 3. Approximate conditional probabilities obtained based on measured data in [31]

Description	Conditional probability (approx. value)
A is substituted if substitution occurs	*P(A/s)* =0.266
C is substituted if substitution occurs	*P(C/s)* =0.211
G is substituted if substitution occurs	*P(G/s)* =0.19
T is substituted if substitution occurs	*P(T/s)* =0.325
A is substituted with *C* if substitution of *A* occurs	*P(C/A,s)* =0.160
A is substituted with *G* if substitution of *A* occurs	*P(G/A,s)* =0.709
A is substituted with *T* if substitution of *A* occurs	*P(T/A,s)* =0.123
C is substituted with *A* if substitution of *C* occurs	*P(A/C,s)* =0.263
C is substituted with *G* if substitution of *C* occurs	*P(G/C,s)* =0.117
C is substituted with *T* if substitution of *C* occurs	*P(T/C,s)* =0.610
G is substituted with *A* if substitution of *G* occurs	*P(A/G,s)* =0.625
G is substituted with *C* if substitution of *G* occurs	*P(C/G,s)* =0.120
G is substituted with *T* if substitution of *G* occurs	*P(T/G,s)* =0.245
T is substituted with *A* if substitution of *T* occurs	*P(A/T,s)* =0.107
T is substituted with *C* if substitution of *T* occurs	*P(C/T,s)* =0.8
T is substituted with *G* if substitution of *T* occurs	*P(G/T,s)* =0.092

4 Decoding

The decoder is the main block which distinguishes the proposed DNA storage system with noise-guessing from known systems with ordinary ECCs. The noise-guessing algorithm is quite simple, and it was described for different communication channels numerous times as was already mentioned in the Introduction.

Simply said the algorithm attempts to change the received codeword into the transmitted codeword by guessing the noise in the communication channel and reverting the corresponding errors caused by the particular sample of noise. The possible samples of noise are ordered in a list in accordance with their decreasing probability. The guessing attempts also follow this list, and the more probable samples are tried before the less probable samples during guessing. Usually a threshold (lower bound) λ is set in advance and the guessing is not continued for patterns with probabilities lower than λ.

The main difference between the communication channel model and the DNA storage model is that the later can, beyond errors (substitutions), also cause insertions and deletions.

For the i.i.d. model, in order to compute the probability of exactly τ_1 impairments of the first type, exactly τ_2 impairments of the second type and exactly τ_3 impairments of the third type, the following formula could be used

$$P(\tau_1,\tau_2,\tau_3) = \\ = \frac{N!\,P_1^{\tau_1}P_2^{\tau_2}P_3^{\tau_3}(1-P_1-P_2-P_3)^{N-\tau_1-\tau_2-\tau_3}}{\tau_1!\,\tau_2!\,\tau_3!\,(N-\tau_1-\tau_2-\tau_3)!} \quad (4)$$

where $\tau_1+\tau_2+\tau_3 \le N$.

For two types of impairments Eqn. (4) reduces to

$$P(\tau_1,\tau_2) = \frac{N!\,P_1^{\tau_1}P_2^{\tau_2}(1-P_1-P_2)^{N-\tau_1-\tau_2}}{\tau_1!\,\tau_2!\,(N-\tau_1-\tau_2)!} \quad (5)$$

where $\tau_1+\tau_2 \le N$.

In the case that only one type of impairment occurs, it becomes

$$P(\tau_1) = \binom{N}{\tau_1} P_1^{\tau_1}(1-P_1)^{N-\tau_1} \quad (6)$$

where $\tau_1 \le N$.

Observing Eqn. (6) and Tab. 1 one can see that the most probable pattern will be a single deletion in the block of length N.

For example, if we set λ=0.005 and choose codeword length N=25, then the guessing will not be attempted for patterns with five deletions because P_{del} (5) = 0.0007. However, if necessary, the patterns with quadruple deletions will be tried because P_{del} (4) = 0.0051.

Numerous further observations could be made which could decrease the computational complexity of the noise-guessing decoder. For example, if the decoder knows the codeword length and the received word is one symbol longer, there could not be single deletion which caused this situation.

5 Description of simulations and their results

In this Section the focus is on testing the functionality of the core system and evaluating the efficiency of the proposed algorithm. Assessing both functionality and efficiency requires the use of several metrics such as redundancy, decoding complexity, and residual error rate. Supplementary metrics include code rate $R_C=k/n$, threshold λ, simulation duration t and others.

In order to give proof of concept by simulation of the hypothetical DNA storage system with noise-guessing, a decoder software was created and used in MATLAB.

This software is available to interested readers in GitHub [32]. The software emulates all basic procedures of the hypothetical DNA storage system, namely encoding, channel impairment insertion into stored codewords, decoding and evaluation of performance of the appropriate simulations.

Before starting the simulation tests, the quality of the impairment's insertion model was tested. 100 000 bits of payload were encoded into 50 000 nucleotides and corrupted by the channel model. This test was repeated 10 times and averages of the impairment frequencies were calculated.

The results were as follows: $P_{imp} = 0.067$, $P_{del} = 0.02878$, $P_{in} = 0.020974$, $P_s = 0.017317$. The percentage of particular nucleotide substitutions and their distributions were also adequate. 26.9613% for A, 20.9945% for C, 18.674% for G and 33.3702% for T. The values achieved are satisfactorily close to values in Tabs. 1 and 2, which shows that the model is adequate.

All simulations were performed on a computer with an AMD Ryzen 5 3600 processor, an NVIDIA GeForce RTX 2070 graphics processing unit (GPU), and 16 GB of DDR4-3200 CL16 memory. The first set of simulation tests was done in order to estimate the overall bit error rate (BER) and block error rate (BLER) of the system dependent on the redundancy introduced, which is equal to the degree of the CRC polynomial (CRC-x). In these tests following basic parameters were chosen: total payload approximately 1 Mbit, k=16, CRC-6, CRC-8, CRC-10, CRC-16, $\lambda = 0.2$, $\lambda = 0.1$, $\lambda = 0.05$.

The results of these tests are presented in Tabs. 4, 5, and 6, and in Figs. 2, 3 and 4. E_r in these tables denotes the maximal number of impairments which was in the guessing list and was guessed if necessary. Observing the results in Tab. 4 and Fig. 2, it is obvious that for a short block the redundancy of CRC-6 gives better results in comparison with CRC-8 andCRC-10. CRC-16 gives best results in this setting, but it also causes the highest computational complexity measured by t. Observing the results in Tab. 5 and Fig. 3, it is obvious that for this scenario CRC-10 gives better results if compared with CRC-6 and CRC-8.

Table 4. Results of first set of tests for $\lambda = 0.2$, $k = 16$

CRC-x	t	R_C	E_r	BER	BLER
CRC-6	4′ 30″	5/7	1	0.032	0.194
CRC-8	5′ 53″	2/3	1	0.033	0.199
CRC-10	6′ 58″	5/8	1	0.034	0.218
CRC-16	149′	1/2	2	0.019	0.124

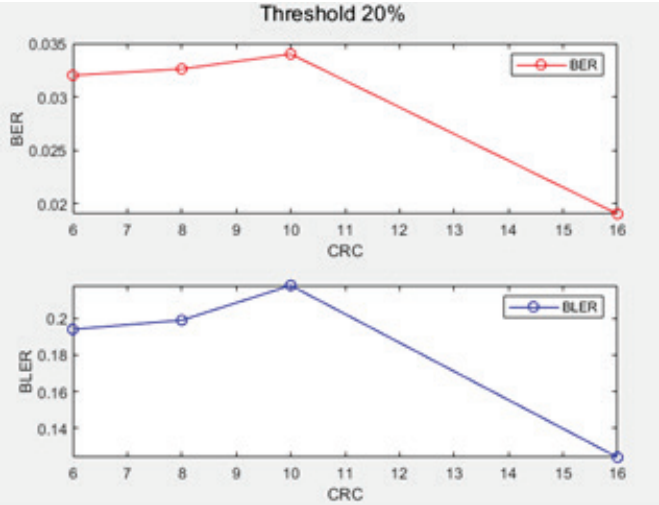


Fig. 2. BER and BLER as a function of redundancy for $\lambda = 0.2$

Table 5. Results of first set of tests for $\lambda = 0.1$, $k = 16$

CRC-x	t	R_C	E_r	BER	BLER
CRC-6	9′ 14″	5/7	2	0.029	0.170
CRC-8	26′ 29″	2/3	2	0.022	0.133
CRC-10	57′ 22″	5/8	2	0.017	0.100
CRC-16	153′ 11″	1/2	2	0.019	0.124

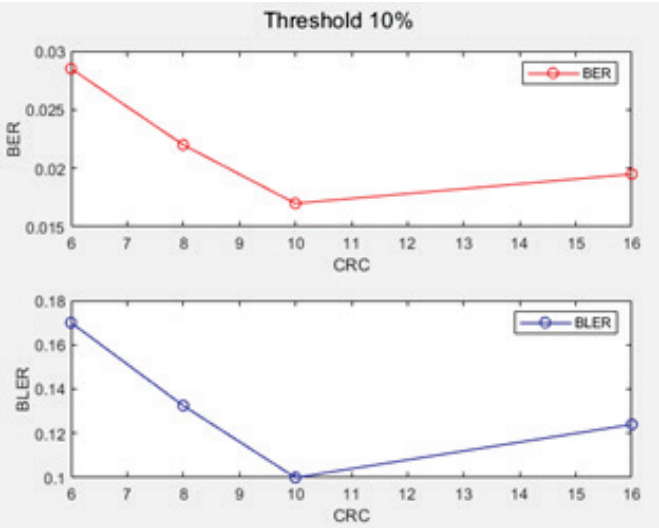


Fig. 3. BER and BLER as a function of redundancy for $\lambda = 0.1$

Table 6. Results of first set of tests for $\lambda = 0.05$, $k = 16$

CRC-x	t	R_C	E_r	BER	BLER
CRC-6	9′ 29″	5/7	2	0.028	0.170
CRC-8	28′	2/3	2	0.022	0.133
CRC-10	57′ 14″	5/8	2	0.017	0.104
CRC-16	1477′	1/2	3	0.014	0.094

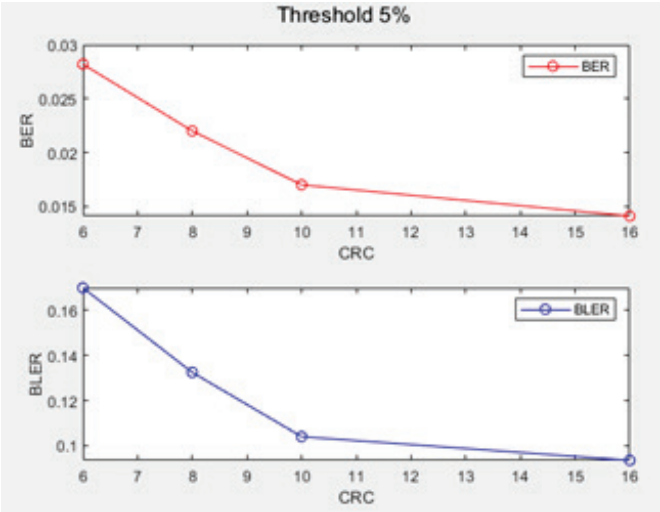


Fig. 4. BER and BLER as a function of redundancy for λ =0.05

Observing the results in Tab. 6 and Fig. 4, one can see that in this scenario CRC-16 gives better results if compared with CRC-6 and CRC-8 and CRC-10. but it also causes the highest computational complexity as measured by t.

The second set of simulation tests was done with the goal to estimate overall bit error BER and block BLER in dependency on the codeword length of the system by fixed CRC-10.

In these tests the following basic parameters were chosen: total payload approximately1 Mbit, $k = 10$, $k = 20$, $k = 30$, $k = 40$, $k = 50$, CRC-10, $\lambda = 0.05$, $\lambda = 0.1$, $\lambda = 0.2$. In order to achieve approximately 1Mbit for different codeword lengths different numbers of simulation repetition were necessary. In the following tables the number of these repetitions is given in the column named "count". The results of these tests are presented in Tabs. 7, 8 and 9, and in Figs. 5, 6 and 7.

Table 7. Results of first set of tests for $\lambda = 0.2$ and CRC-10

k	t	R_C	E_r	BER	BLER	count
10	7′ 15″	1/2	1	0.029	0.147	10^6
20	7′ 11″	2/3	1	0.038	0.270	5×10^5
30	100′ 36″	3/4	2	0.032	0.259	33334
40	122′ 6″	4/5	2	0.042	0.395	25000
50	197′ 12″	5/6	2	0.053	0.689	20000

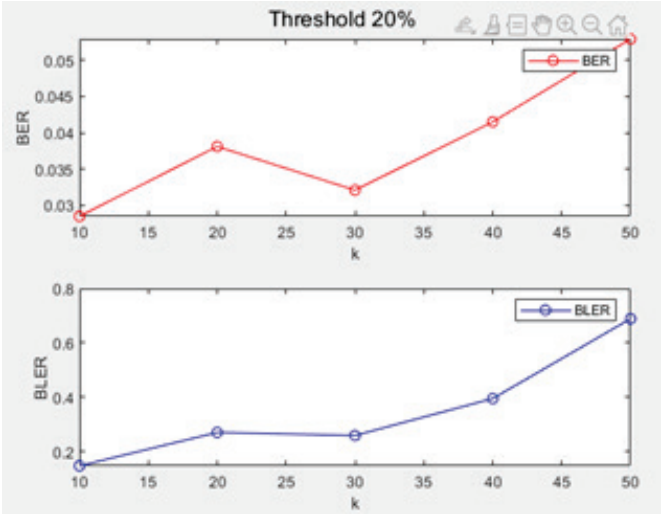


Fig. 5. BER and BLER as a function of k for λ =0.2

Table 8. Results of first set of tests for $\lambda = 0.1$ and CRC-10

k	t	R_C	E_r	BER	BLER	count
10	39′ 46″	1/2	2	0.01	0.055	10^6
20	77′ 21″	2/3	2	0.022	0.144	5×10^5
30	148′ 12″	3/4	3	0.0314	0.254	33334
40	172′ 44″	4/5	3	0.0413	0.392	25000
50	220′ 25″	5/6	3	0.0542	0.699	20000

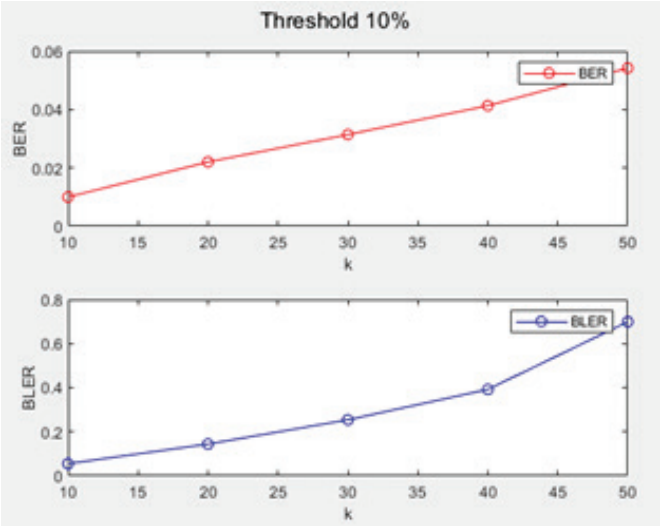


Fig. 6. BER and BLER as a function of k for λ =0.1

Table 9. Results of first set of tests for $\lambda = 0.05$ and CRC-10

k	t	R_C	E_r	BER	BLER	count
10	38′ 40″	1/2	2	0.011	0.059	10^6
20	129′ 35″	2/3	3	0.0212	0.142	5×10^5
30	152′ 25″	3/4	3	0.0303	0.247	33334
40	463′ 33″	4/5	4	0.0408	0.38	25000
50	250′ 58″	5/6	4	0.0537	0.7	20000

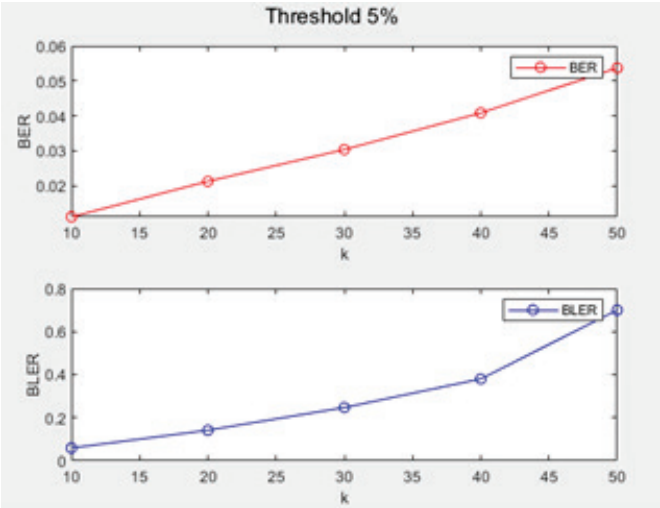


Fig. 7. BER and BLER as a function of k for λ =0.05

Observing the results of this second set of experiments the following notes can be made.

As expected, the BER and BLER are highest for the longest codeword. The decrease in BER at k=30 is caused by the fact that at this length up to double impairments are on the guessing list. For other tested lengths only, single impairments are guessed. Therefore, the time for k=30 is relatively high.

In the case that the threshold is λ =0.1, the method gives BLER = 0.055, or in other words, 94.5% of blocks is correct after decoding. At k = 30 the guessing list contains also triple impairments, but BER and BLER do not decrease. It is caused by the fact that CRC-10 is not adequate under these conditions. BER and BLER increase further at longer payloads. At k = 50, BLER is close to 0.7.

In case the threshold is λ = 0.05, the trends in results are similar as in the previous case. The difference is that at k = 40, k = 50 quadruple impairments are also on the guessing list. It does not help to decrease BER and BLER adequately. It proves that the CRC-10 is not effective at these values.

6 Conclusion

In Conclusions, following a summary of this paper's material, direction is mentioned which can improve the decoder based on noise-guessing paradigm for nanopore DNA storage systems. Simplified hypothetical system model was used and simple i.i.d. model of impairments, which was built in accordance with real measured data using Oxford nanopore sequencing [31]. It is obvious that in order to get better performance a more realistic impairment model which takes into account different dependencies [33] has to be used.

Acknowledgement

Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I05-03-V02-00051.

References

[1] IDC, 2025, May, Worldwide IDC Global DataSphere Forecast, 2025–2029 Available: https://my.idc.com/getdoc.jsp?containerId=US53363625

[2] M. S. Smith, "What will it Take to Build the World's Largest Data Center?: Meta's 5-Gigawatt AI Data Center is Forcing Engineers to Throw Out the Rule Book," in IEEE Spectrum, vol. 63, no. 4, pp. 22-27, 2026.

[3] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig and K. Strauss, "Toward a DNA-Based Archival Storage System," in IEEE Micro, vol. 37, no. 3, pp. 98-104, 2017.

[4] L. Mu, L. Liu, L. Ding and Z. Zhu, "A Survey on the Coding for DNA Storage: Challenges, Algorithms, and Future Directions," in IEEE Open Journal of the Communications Society, vol. 7, pp. 3528-3558, 2026.

[5] L. Xiang, Q. Liu, S. Chen, K. Yan, W. Wu and K. Yang, "A Tutorial on Coding Methods for DNA-Based Molecular Communications and Storage," in IEEE Internet of Things Journal, vol. 11, no. 7, pp. 11825- 11847, 1 April1, 2024.

[6] L. Welter, R. Sokolovskii, T. Heinis, A. Wachter-Zeh, E. Rosnes and A. G. i. Amat, "An End-to-End Coding Scheme for DNA-Based Data Storage with Nanopore-Sequenced Reads," in IEEE Journal on Selected Areas in Information Theory, vol. 7, pp. 17-32, 2026.

[7] J. Jeong et al., "Iterative Soft Decoding Algorithm for DNA Storage Using Quality Score and Redecoding," in IEEE Transactions on NanoBioscience, vol. 23, no. 1, pp. 81-90, 2024.

[8] J. Zhang and S. Zhao, "Design and Analysis of Two-Layer Coding Scheme for DNA-Based Data Storage," in IEEE Transactions on NanoBioscience, vol. 25, no. 2, pp. 259-269, 2026.

[9] M. Zhang, K. Cai, K. A. Schouhamer Immink and P. Chen, "Soft-Decision Decoding for DNA-Based Data Storage," 2018 International Symposium on Information Theory and Its Applications (ISITA), Singapore, 2018, pp. 16-20, 2018.

[10] R. Gurumayum, A. K. Jaiswal and R. Biswas, "Hybrid Error Correction Encoding for DNA Data Storage," 2026 International Conference on Intelligent Processing, Hardware, Electronics, and Radio Systems (CIPHER), Jalandhar, India, pp. 1-6, 2026.

[11] K. R. Duffy, J. Li and M. Médard, "Capacity-Achieving Guessing Random Additive Noise Decoding," in IEEE Transactions on Information Theory, vol. 65, no. 7, pp. 4023-4040, 2019.

[12] Q. Wang, J. Liang, P. Yuan, K. R. Duffy, M. Médard and X. Ma, "Guessing Decoding of Short Blocklength Codes," in IEEE BITS the Information Theory Magazine (Early Access) 2026.

[13] Feng, K. R. Duffy and M. Médard, "Enhanced GCD Through ORBGRAND-AI: Exploiting Partial and Total Correlation in Noise," in IEEE Communications Letters, vol. 30, pp. 2884-2888, 2026.

[14] P. Yuan, K. R. Duffy, E. P. Gabhart and M. Médard, "Discretized Soft GRAND for Front-End-Constrained Communication," in IEEE Transactions on Communications, vol. 74, pp. 12197-12210, 2026.

[15] Z. E. Kizilates et al., "Soft-Output Threshold-Guided CRC Decoding with SOGRAND in 40nm CMOS," 2026 IEEE Custom Integrated Circuits Conference (CICC), Seattle, WA, USA, pp. 1-4, 2026.

[16] A. Bali, Z. E. Kizilates, A. Riaz, M. Medard, K. R. Duffy and R. Tugce Yazicigil, "ORBGRAND: An SDR Study," 2025 59th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, pp. 1760-1765, 2025.

[17] A. Riaz, J. Feng, K. Galligan, M. Médard, R. T. Yazicigil and K. R. Duffy, "IGRAND Query Reduction using Even Codes," 2025 IEEE Workshop on Signal Processing Systems (SiPS), Hong Kong, pp. 1-5, 2025.

[18] L. Rapp, J. Feng, M. Médard and K. R. Duffy, "A Balanced Tree Transformation to Reduce GRAND Queries," 2025 IEEE International Symposium on Information Theory (ISIT), Ann Arbor, MI, USA, pp. 1-6, 2025.

[19] C. Wiame, K. R. Duffy and M. Médard, "Joint Error Correction and Fading Channel Estimation Enhancement Leveraging Grand," ICC 2025 - IEEE International Conference on Communications, Montreal, QC, Canada, pp. 2014-2019, 2025.

[20] L. Rapp, M. Médard and K. R. Duffy, "SOGRAND Assisted Guesswork Reduction," in IEEE Communications Letters, vol. 29, no. 11, pp. 2656-2660, 2025.

[21] J. C. Griffin, P. Yuan, R. Thesmar, P. Popovski, K. R. Duffy and M. Médard, "Code at the Receiver, Decode at the Sender: Feedback Communication With GRAND-CE," in IEEE Transactions on Communications, vol. 73, no. 12, pp. 13017-13032, 2025.

[22] A. Riaz et al., "Iterative Guessing Random Additive Noise Decoder for Universal Decoding of Product Codes," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 72, no. 12, pp. 8067-8080, 2025.

[23] K. Yang, M. Médard and K. R. Duffy, "GRAND-AM in uplink MIMO NOMA systems," 2025 59th Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, pp. 1-6, 2025.

[24] J. Griffin, P. Yuan, K. R. Duffy and M. Médard, "Using a Single Parity-Check to Reduce the Guesswork of Guessing Codeword Decoding," 2025 59th Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, pp. 1-6, 2025.

[25] P. Yuan, M. Médard, K. Galligan and K. R. Duffy, "Soft-Output (SO) GRAND and Iterative Decoding to Outperform LDPC Codes," in IEEE Transactions on Wireless Communications, vol. 24, no. 4, pp. 3386-3399, 2025.

[26] K. R. Duffy, P. Yuan, J. Griffin and M. Médard, "Soft-Output Guessing Codeword Decoding," in IEEE Communications Letters, vol. 29, no. 2, pp. 328-332, 2025.

[27] A. Riaz et al., "A Sub-0.8-pJ/bit Universal Soft-Detection Decoder Using ORBGRAND," in IEEE Journal of Solid-State Circuits, vol. 60, no. 7, pp. 2645-2659, 2025.

[28] J. Feng, K. R. Duffy and M. Médard, "Laplacian-ORBGRAND: Decoding for Impulsive Noise," MILCOM 2024 - 2024 IEEE Military Communications Conference (MILCOM), Washington, DC, USA, pp. 1-6, 2024.

[29] A. Riaz et al., "Demo: Universal Soft-Detection Decoder with Ultra-Low Energy Consumption Using ORBGRAND," 2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Boston, MA, USA, pp. 337-339, 2023.

[30] B. Ozaydin, M. Médard and K. R. Duffy, "GRAND-assisted Optimal Modulation," GLOBECOM 2022 - 2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 2022, pp. 813-818, 2022.

[31] R. Lopez et al. DNA assembly for nanopor data storage readout, Nature Communications 10, art. No. 2933. 2019.

[32] D. Hudač, P. Farkaš, "DNAStorageCorrection", GitHub repository, https://github.com/DanoHudac/DNAStorageCorrection, 2026

[33] T. Janvars, P. Farkaš, "Realistic Channel Model with Memory and Clustering for DNA Data Storage Using Nanopore Sequencing, 2026 6th International Conference on Electrical, Computer and Energy Technologies (ICECET), 06-09, Rome, Italy, 2026.

Received 15 August 2026