

Strongly Connected Ramanujan Graphs for Highly Symmetric LDPC Codes

Hussein Al-Hamdani (✉ haalhamdani@ualr.edu)

University of Arkansas at Little Rock <https://orcid.org/0000-0003-3449-2720>

Aqiel Almamori

University of Baghdad

Seshadri Mohan

University of Arkansas at Little Rock

Research Article

Keywords: Algebraic LDPC coding. Algorithms. GAP system. Symmetric LDPC codes. Ramanujan graphs

Posted Date: March 16th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-932857/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at Wireless Personal Communications on April 9th, 2022. See the published version at <https://doi.org/10.1007/s11277-022-09668-0>.

Strongly Connected Ramanujan Graphs for Highly Symmetric LDPC Codes

¹Hussein Al-Hamdani, ²Aqiel Almamori, and ¹Seshadri Mohan

¹University of Arkansas at Little Rock, Department of Systems Engineering, Little Rock, Arkansas, USA

²University of Baghdad, Department of Electronics and Communications Engineering, Iraq

Emails: haalhamdani@ualr.edu, a.eced@coeng.uobaghdad.edu.iq, sxmohan@ualr.edu

Abstract

A number of studies focus on Low-Density Parity-Check (LDPC) codes to ensure reliable data communications. This study proposes an algebraic algorithm to generate strongly connected Ramanujan graphs able to provide highly symmetric LDPC codes with minimized error floor. Several Ramanujan graphs are created using GAP system software to generate a rank-efficient parity-check matrix with fixed-rate LDPC codes. We find that Ramanujan LDPC codes achieve frame error rate and bit error rate on the order of 10^{-5} and 10^{-6} , respectively. Furthermore, the codes outperform QC LDPC codes and those Ramanujan LDPC codes in literature.

Keywords Algebraic LDPC coding. Algorithms. GAP system. Symmetric LDPC codes. Ramanujan graphs

1 Introduction

The channel coding theorem has captivated the attention of researchers and motivated them to design codes that have facilitated communication systems to transmit information with reliability while approaching capacity. In communication systems, the probability of decoding error of coded information is a significant parameter in measuring the reliability of communication systems, where the lower the probability of decoding error, the more reliable the communication system is. The weak converse of Fano's inequality [1] to the channel coding theorem shows two critical parameters that control the probability of decoding error, the code's length, and rate. The theorem shows that when the length of the codeword approaches infinity, the probability of decoding error goes exponentially to zero if the code rate is below the channel capacity. Conversely, this probability approaches one exponentially if the code rate is above the channel capacity.

Low-Density Parity-Check (LDPC) code forms a family of error correction codes that meets the above inequality. Because the parity-check matrix of LDPC code is precisely the adjacency matrix of a bipartite graph called the Tanner graph, iterative decoding techniques employ the Tanner graph as the underlying graph to decode the received words. When the underlying graphs exhibit a significant girth and expanding factor, the iterative decoders decode LDPC codes near Shannon's limit. The decoder requires a large girth to avoid any possible feedback in the decoding algorithm, which is a primary cause that degrades the decoding efficiency. Moreover, it requires a significant expanding factor, which is how fast the information is disseminated among incident vertex nodes of the graph; the more significant the expansion factor is, the faster the iterative decoder starts coverage.

One way to construct excellent LDPC codes is to choose suitable graphs that meet decoder requirements. Ramanujan graphs [2] are excellent candidates because they have a significant expanding factor and the highest lower known girth limit. For example, Margulis [3] proposed the first explicit construction of (3,6) regular LDPC codes based on Ramanujan graphs. As compared to the random construction of LDPC codes [4] and for a code of length n , Margulis's method undertook to realize a decrement in the code construction complexity in terms of the required memory and number of operations from $O(n^2)$ to $O(\log n)$ and from $O(n^j)$ to $O(\log n) \log \log n$ respectively. Rosenthal and Vontobel [5] developed an alternative approach based on Lubotzky, Phillips, and Sarnak's (LPS) Ramanujan graphs [2]. Their study revealed that the performance of generated LDPC codes from Ramanujan graphs presents a significant gain in the bit error rate (BER) in low SNR ranges compared to random regular LDPC codes of the same dimension.

In contrast, MacKay and Postol [6] reported weaknesses in the LDPC codes constructed from Ramanujan graphs. They pointed out that Margulis's code generates codewords that cause problems to iterative decoders. On the other hand, they recognized that Rosenthal's approach produces low-weight codewords, which is the leading cause of the error floor that occurs during code simulation. Meanwhile, two facts are revealed in [6]; Ramanujan- Margulis codes (Rosenthal's approach) outperform the same length random LDPC codes in low SNR ranges because the code rate of the Ramanujan-Margulis code is higher than 0.5. And because the [2184, 1092] code is not able to simulate down to frame error rate (FER) of 10^{-5} , it is classified as an invalid code for practical applications [6]. Hyungrok and Yamasaki's study [7] suggested a possible use of LPS [2] and Chiu's [8] Ramanujan graphs to construct a new family of LPS expanders based on the \mathbb{Q} rational numbers for the applications of secure Cayley hash functions. For practical

application support, Lio [9] partitioned the Ramanujan-Margulis parity-check matrix, which lowered the probability of decoding error.

Other studies dealt with trapping sets or decoders to lowering the error floor phenomenon in Random and Ramanujan LDPC codes demonstrated in [10], [11], [12], [13], [14]. Karimi [15] used a search algorithm to find the dominant trapping sets for Quasi-Cyclic (QC) LDPC codes. Zhang [16] proposed three different classes of type II QC LDPC codes with girth eight from the Sidon sequence; the simulations addressed some of the proposed codes could achieve BER value down to 10^{-4} at 1.5 dB. In [17], Zhang proposed an algebraic construction for type II QC LDPC codes from the Tanner method based on multiplicative subgroup analysis; the long codes in his work simulated down to 10^{-4} when the SNR is at least 2.8 dB. As a result, the error floor phenomenon was reduced in the previous works.

However, although the previous works [5], [6] have focused only on generating LDPC codes from Ramanujan graphs, little attention has been paid to minimize error floor problem by using edge-transitive Cayley graphs as Ramanujan graphs to construct highly symmetric LDPC codes [18] that have fixed-rate and closely fixed distance. To the best of our knowledge, the highly symmetric LDPC codes based on the edge-transitive approach have not been implemented and tested in real-life problems. Therefore, this paper provides an algebraic method for constructing highly symmetric LDPC codes to ensure the constructed LDPC parity-check matrix from Cayley graphs generates decodable codewords with reduced error floor compared to [6]. In this work, the Cayley graphs constructed are very much connected Ramanujan graphs and are used to construct (3,6) regular LDPC codes. Furthermore, the parity-check matrix of LDPC codes is a full-rank matrix.

What is more, designing symmetric Cayley graphs based on the proposed algorithm provides unique LDPC codes that outperform many QC LDPC codes of similar lengths. This paper is organized as follows; Section II presents the proposed methodology for the construction of Ramanujan Graph-based LDPC codes. Section III illustrates the results and discusses the proposed codes. Finally, Section IV draws some conclusions from this research study.

2 Methodology

The current study involves the construction of Cayley graphs as a suitable Ramanujan expander using Groups, Algorithms, and GAP Programming system software [19] to generate highly symmetric LDPC codes [20]. In an attempt to avoid feedback in the algorithm of an iterative decoder, which is the cause of error floor, and to ensure the decoder starts coverage close to Shannon's limit, Ramanujan graphs were chosen for generating the LDPC codes [21].

GAP software is chosen on account of its ability to provide thousands of functions and packages to implement algebraic algorithms for building graphs [19].

In all cases, LDPC codes were constructed to have a fixed-rate [18], [20] and full rank parity-check matrix, where the constraint space of the codewords is generated by the orbit of one constant weight constraint under a group action [19], [20]. Each of the graphs constructed in this work is very much connected graph because they have large ε values, where $0 < \varepsilon \leq 1$, and $\varepsilon = 1$ is the maximum value that one can get with these types of expanders [22]. According to *Definition 1.1* [22], a finite graph X on a set V of n vertices is called an ε -expander, $\varepsilon(X)$, if $|Y| \leq 0.5|V| = n/2$, and $|\partial Y| \geq \varepsilon|Y|$ holds for every subset Y of V , where the boundary $\partial Y = \{v \in V \setminus Y : (v, y) \in E \text{ for some } y \in Y\}$. The graphs constructed were split directly based on the procedure available in [5], [3], [6] to obtain the LDPC matrix and codes. Moreover, the generated LDPC matrices have full rank, and the code rate is constant and exactly equals 0.5. The value of the expanding factor, also known as the Cheeger constant [23], of the constructed graphs was measured to be six; however, splitting the graphs to generate the matrix (Tanner graph) makes its value equal to 3.

One way to construct Ramanujan graphs is by using Cayley graphs [18], [24], [5], [6]. Let us introduce Cayley graphs in simple words based on [25]. Assuming G is a finite or infinite group generated by a generator set $S \subseteq V$, the Cayley graph $Cay(V, S)$ is defined to be a graph with vertex set $V = G$, generator set S , and edge set $E = \{(x, y) : x, y \in G : \exists s \in S : y = xs\}$, where the edges are adjacent, and that one is obtained by right multiplication by some element of S . In general, Cayley graphs are often not symmetric, which means the group G acts transitively on the vertex of $Cay(V, S)$ but not on the edges. They are symmetric if and only if the generator set is symmetric, i.e., $S = S^{-1}$ [18]. Then they are called edge-transitive Cayley graphs such that Lemma 4 of [24] is satisfied. Since S is symmetric, the adjacency relation is also symmetric, and therefore, the graph is undirected. Generally speaking, the lemma, is based on [18], may be explained as follows. Assume there is an action group called act on the vertex set V of the Cayley graph. In other words, there exists homeomorphism $\varphi: act \rightarrow Aut(V)$. Let S be an orbit of this action, that is, there exists $\delta \in V$ such that $S = \{\varphi(\alpha)(\delta) \mid \alpha \in act\}$. Here, the semi-direct product group $H = V \times act$ acts on $Cay(V, S)$, and this action is transitive on the edges. The code is called highly symmetric code if *Corollary 1* [20] is satisfied, where it can be summarized as the group act acts on S and hence on \mathbb{F}_2^S , if a small code $B \subseteq \mathbb{F}_2^S$ is kept by act , then $H = V \times act$ keeps $C(G, S, B)$. Besides, if B is a single orbit symmetric with respect to act then $C(G, S, B)$ is also single orbit symmetric and therefore, it called highly symmetric when $|S|$ is bounded.

In this study, we implement Cayley symmetric graphs $Cay(V, S)$ as Ramanujan graphs when the vertex node set V is a Projective General Linear (PGL) group, and they are denoted by $\mathbb{X}^{p,q}(PGL_2(\mathbb{Z}_q), S)$, where $PGL_2(\mathbb{Z}_q)$ is the projective general linear group over the field \mathbb{Z}_q . The n -by- n adjacency matrix of $\mathbb{X}^{p,q}$ is symmetric, the eigenvalues are real [25]; furthermore, the graphs are k regular, connected, and are bi-colorable graphs. Therefore, the eigenvalues can be sorted as $\mu_0 = k \geq \mu_1 \geq \dots \geq \mu_{n-1} = -k$. By taking p to be a prime congruent to 1 modulo four, i.e., $p \equiv 1 \pmod{4}$, and $i^2 \equiv -1 \pmod{q}$, one can construct the generator set of Cayley graph such that the Cayley graph is a Ramanujan graph by using the Jacobi symbol; the prime p is constructed as

$$p = a_0^2 + a_1^2 + a_2^2 + a_3^2 \quad (1)$$

Consider $a_0 > 0$ and odd, and $a_j; j \in \{1,2,3\}$ are even. Then there are $p + 1$ available solutions to (1); for each solution, define \tilde{p} matrix in $PGL_2(\mathbb{Z}_q)$ as

$$\tilde{p} = \begin{pmatrix} a_0 + ia_1 & a_2 + ia_3 \\ -a_2 + ia_3 & a_0 - ia_1 \end{pmatrix} \quad (2)$$

These solutions, $s_1^\pm, s_2^\pm, s_3^\pm$, define the generator set $S = \{s_1, s_2, s_3, s_1^{-1}, s_2^{-1}, s_3^{-1}\}$ of $\mathbb{X}^{p,q}$; the determinant of each solution equals p . LPS performs a map-to-map the elements of $PGL_2(\mathbb{Z}_q)$ or, equivalently, $GL(\mathbb{Z}_q)$, to their images as

$$\begin{aligned} \varphi: PGL_2(\mathbb{Z}_q) &\rightarrow \{-1, 1\}, \\ A &\rightarrow \left(\frac{\det(A)}{q} \right): A \in GL_2(\mathbb{Z}_q) \end{aligned} \quad (3)$$

To use $\mathbb{X}^{p,q}$ in regular LDPC codes, $\mathbb{X}^{p,q}$ must have a bipartite structure; therefore, $q > p$ is chosen to be a prime number such that $q \equiv 1 \pmod{4}$. Based on the Legendre symbol, $\frac{p}{q}$, the graph $\mathbb{X}^{p,q}$ is bipartite if and only if $\frac{p}{q} = -1$, which means p is a quadratic non-residue modulo q ; otherwise, it is not a bipartite graph. In general, when x of the Legendre symbol $\frac{x}{q}$ is a quadratic residue modulo q , $g \in PGL_2(\mathbb{Z}_q)$ belongs to the right side of the bipartite graph with quadratic residue modulo q determinant. Otherwise, the mapping decides that the element belongs to the left side of the graph with a quadratic non-residue modulo q determinant [2], [5].

The kernel of the above mapping is the Special Linear group, $PSL_2(\mathbb{z}_q) \subseteq PGL_2(\mathbb{z}_q)$. However, the set of the elements on the right side of the graph is the set of the right-coset, R , of $PSL_2(\mathbb{z}_q)$ group; $PSL_2(\mathbb{z}_q) \cup R = PGL_2(\mathbb{z}_q)$ [6], and $|g| \in \mathbb{z}_q^* : g \in PGL_2(\mathbb{z}_q)$. Since $B \subset PSL_2(\mathbb{z}_q)$ [26] then the expansion constant of this subset is

$$h(B) = \frac{|\partial B|}{|B|} \quad (4)$$

However, the expansion constant of the graph $\mathbb{x}^{p,q}$ is

$$h(\mathbb{x}^{p,q}) = \min \left\{ \frac{|\partial B|}{|B|} : B \neq \phi, B \leq \frac{|R|}{2} \right\} \quad (5)$$

$$\frac{k - \mu_1}{2} \leq h(\mathbb{x}^{p,q}) \leq 2\sqrt{k(k - \mu_1)} \quad (6)$$

Furthermore, $|PSL_2(\mathbb{z}_q)| = |R|$, i.e., the number of elements on the left side of the LPS graph equals the number of elements on the right side of the graph [5], [6].

To construct the LDPC matrix [5], two disjoint copies of $PSL_2(\mathbb{z}_q)$ named L and \tilde{L} are used to index columns of the matrix [5]; however, the rows of the LDPC array are indexed by R as defined above. Therefore, the number of columns in the LDPC matrix is the same as the number of vertices in the Cayley graph. The nonzero elements, 1s, of column $l \in L$ in $H_{m \times n}$ are entries of ls_1, ls_2, ls_3 rows; however, those of column $\tilde{l} \in \tilde{L}$ are entries of ls_1^{-1}, ls_2^{-1} , and ls_3^{-1} rows in $H_{m \times n}$. This mapping of 1s to all columns of L and \tilde{L} , respectively, results in a full-rank (3,6) LDPC parity-check matrix; in this matrix, every column has a degree of 3, and every row has a degree of 6.

$H_{m \times n}$ the adjacency matrix of the split graph from $(\mathbb{x}^{p,q})$ is a Tanner graph of n vertices. Graph splitting, even for non-bipartite graphs, produces a bipartite graph; the effect of splitting on the girth and expanding factor of Tanner graph as compared to the $\mathbb{x}^{p,q}(PGL_2(\mathbb{z}_p), \mathcal{S})$ graph is that the graph after splitting has a girth equal to or greater than the girth of the $\mathbb{x}^{p,q}$ graph, while the expanding factor is at most the expanding factor of the $\mathbb{x}^{p,q}$ graph [26]. In this work, although the splitting procedure of the graphs is similar to the used method in [5], [6], our algorithm shows that the constructed LDPC codes generally have no error floor in the simulation. The algorithm proves the symmetric LDPC codes, which is the requirement of [18]. The steps of the suggested algorithm are followed to implement the graphs and LDPC codes.

2.1. Computational Discrete Algebraic Algorithm

In group theory, the elements of a group can have the matrix or permutation form. At first, the generator matrix set is computed based on (2), then converted to permutation form to fit GAP software requirements and have an isomorphic among the group elements [19]. The proposed algorithm shown below is used to create the Ramanujan graphs and the parity-check matrices of LDPC codes.

Construction Algorithm for Ramanujan's Graphs and LDPC codes Generations

1. initialize: $p, q \equiv 1 \pmod{4}$, $i \wedge 2 \equiv -1 \pmod{q}$
2. Generate:
 - General Linear Group (GL): $\longrightarrow G_l := GL(2, q)$;
 - Permutation domain: $\longrightarrow vec := [1, 0] * Z(q) \wedge 0$;
 - PGL group:
 - $orb := Orbit(G_l, vec, OnLines)$;
 - $act := ActionHomomorphism(G_l, orb, OnLines)$;
 - $pgl := Image(act)$;
 - Special Linear Group (SL) and PSL Group:
 - $SLq := SL(2, q)$;
 - $act2 := ActionHomomorphism(SLq, orb, OnLines)$;
 - $myPSL := Image(act2)$;
3. Cayley graph generator set in the polynomial form:
 - Take $mod2$ to each entry of the six 2×2 matrices of the generator set and transform the set into the polynomial form using
 - $si := Image(act, mati); \forall i \in \{1, 2, \dots, 6\}$
 - $F := [f_1, f_2, f_3, f_4, f_5, f_6]$;
4. Generate a highly symmetric Cayley graph:
 - $caygraph := Graph(pgl, Elements(pgl), OnRight,$
 $function(x, y) return ForAny(F, g \rightarrow y$
 $= g * x); end, true)$;
5. Determine:
 - $Diameter(caygraph)$; if required
 - $Girth(caygraph)$;
 - The adjacency matrix of the graph.
 - $L := RightCoset(myPSL, s1)$;
6. Generate the position of 1's in each row of the matrix:
 - a. the nonzero indices of the left-side hand of the parity-check matrix:
 - $LL := List(L)$; $P := List(myPSL)$;
 - $SS := Concatenation(P, P)$; $Size(SS)$; $Size(LL)$;
 - $k := []$;
 - for i in $[1..Size(LL)]$ do;
 - $pos1 := Position(LL, (SS[i] \times s_1))$;
 - $pos2 := Position(LL, (SS[i] \times s_2))$;
 - $pos3 := Position(LL, (SS[i] \times s_3))$;
 - $posl := [pos1, pos2, pos3]$;
 - $Append(k, posl)$;
 - od;
 - b. the nonzero indices of the right-side hand of the parity-check matrix:

```

        k2:= [];
    for j in [Size(LL) + 1..Size(SS)] do;
        pos4:= Position(LL, (SS[i] × s4));
        pos5:= Position(LL, (SS[i] × s5));
        pos6:= Position(LL, (SS[i] × s6));
        posr:= [pos4, pos5, pos6];
        Append(k2, posr);
    od;

```

7. Use MATLAB to get the H matrix from the vector generated in step 6 and test the matrix.

Although the steps of the GAP system in the algorithm ran efficiently in a short time, the part of simulation in MATLAB consumes a considerable amount of time as the SNR increased. Furthermore, the simulation time increased proportionally with the length of the LDPC code. Therefore, the simulations are carried out through an SNR of 1.25 dB for [50616, 25308] Ramanujan LDPC code. It is reasonable to accept this drawback in the algorithm because it is known that Ramanujan LDPC codes achieve near Shannon limits. With more powerful processors, simulations can be run more rapidly. This algorithm is suitable for low SNR regions.

3 Results and discussions

In earlier studies, attempts were made to construct LDPC codes from Ramanujan graphs [5], [6] to achieve closer to Shannon's limits than the performance of random LDPC codes. However, the error floor was reported during the simulations due to linearly dependent rows in their parity-check matrix [6]. Ramanujan graphs were constructed as well connected and highly symmetric graphs; therefore, the LDPC codes generated have a fixed rate and constant distance [18]. Moreover, the generated LDPC parity-check matrix of each code was full rank, and it appears that the error floor in the simulations is reduced as compared to [6]. All data generated or analysed during this study, including the results and conclusions, are included in this article. In figs. 1 and 2, we compare the frame error rate performance and the bit error rate of the proposed binary LDPC codes versus SNR.

As can be seen in fig.1, the proposed algorithm provides a substantial gain in terms of the frame error rate of the [2184, 1092] LDPC code as compared to the construction available in [6]. It shows approximately a two-fold magnitude gain compared to the results of exact code dimensions in [6]. However, the performance of [4896, 2448] LDPC code is identical to the codes in [5], [6] over the same SNR range. Since the iterative decoder complexity grows linearly with the code length and because the used computer in getting the results has limited capabilities, we used the Lenovo Yoga laptop, the performance of [50616, 25308] LDPC code has been tested only up to SNR= 1.2 dB for 64 iterations. The result shows an improvement of 0.1 dB over the performance of [4896, 2448] and [2184, 1092] codes,

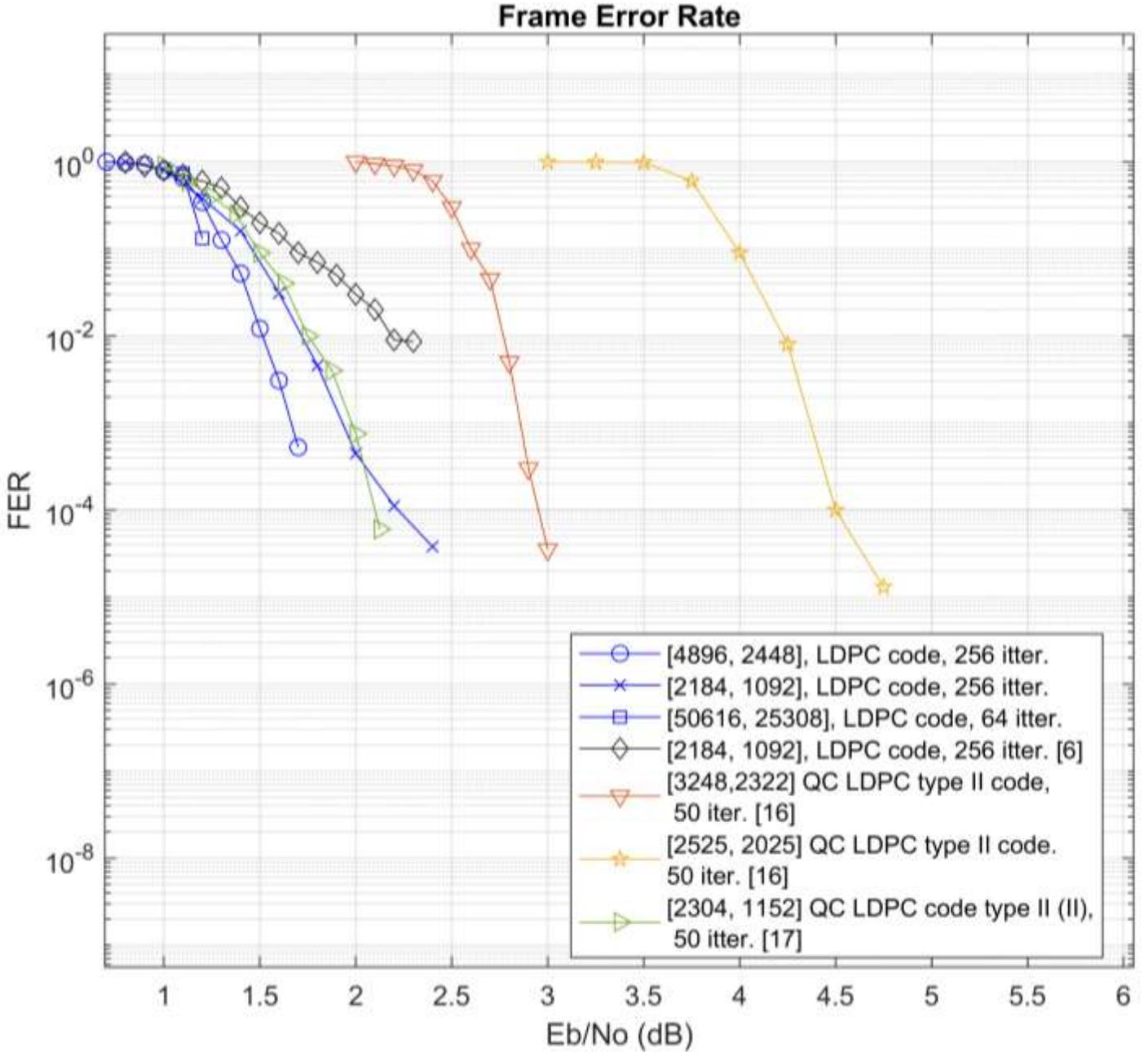


Fig. 1. FER of the proposed LDPC codes vs. Mackay and QC LDPC codes.

as shown in figs. 1 and 2. However, comparing the FER results of [2184, 1092] to the FER of QC LDPC code has similar dimensions [16] shows that the proposed codes in this work offer significant SNR gain as compared to [2525, 2025] QC LDPC type II codes. For example, the SNR gain is 2.5 dB when the frame error rate is 10^{-4} .

However, the same code provides a 0.7 dB compared to [3248, 2322] QC LDPC code at the same error performance. Furthermore, the second class of QC LDPC II (QC LDPC II (II)) [17], which is [2304, 1152] code simulates close to the proposed [2184, 1092] Ramanujan LDPC code. Meanwhile, [4896, 2448] and [50616, 25308] Ramanujan LDPC

codes outperform all the types mentioned of QC LDPC codes in fig. 1. Figure 2 shows the BER performance of the proposed Ramanujan LDPC codes over the performance of Ramanujan LDPC codes in [6] and the types mentioned above of QC LDPC codes. The results show that only the QC LDPC II (II) [16] code achieves performance close to

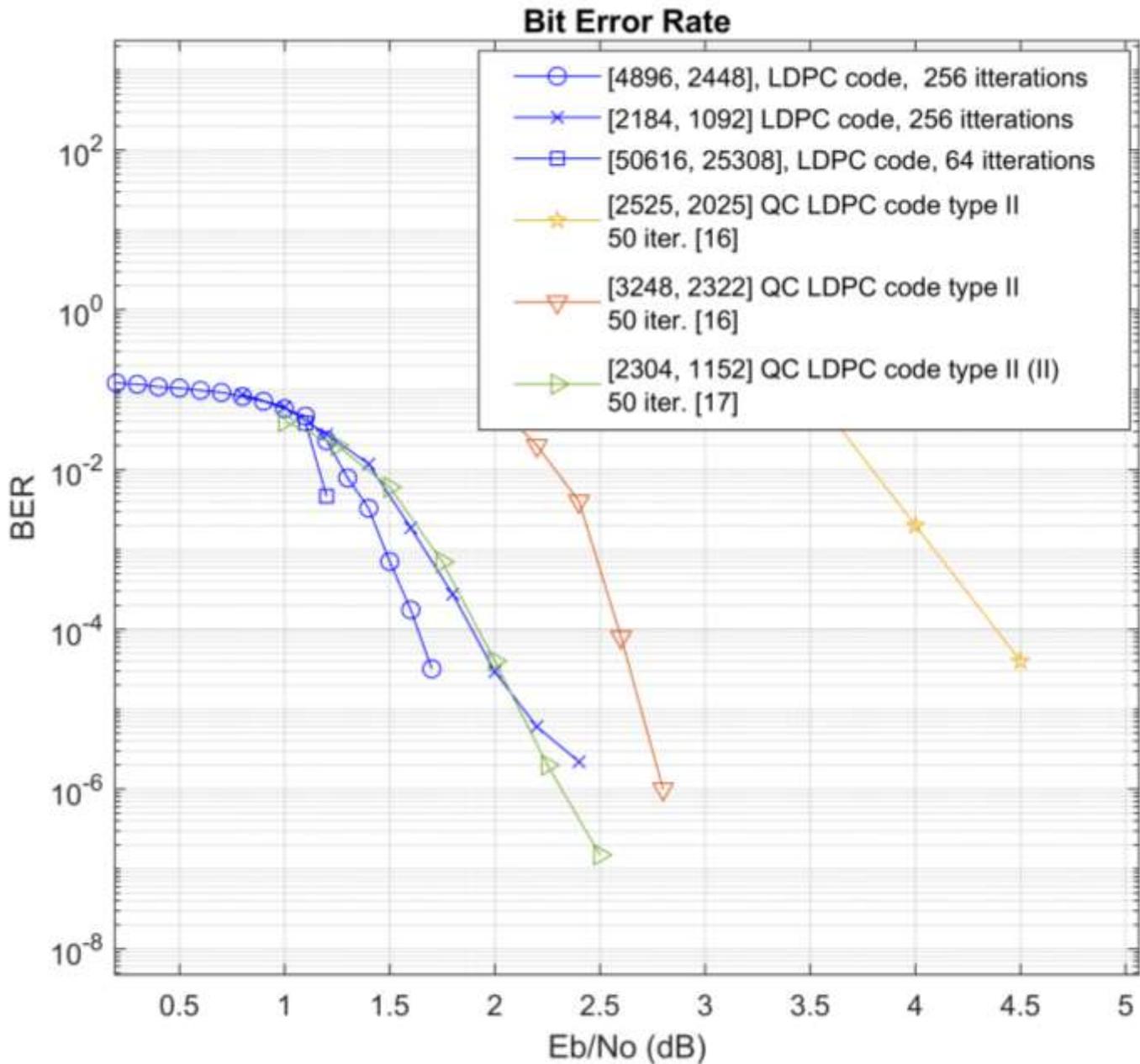


Fig. 2. BER of the proposed LDPC codes vs. Mackay and QC LDPC codes.

the proposed [2448, 1092] Ramanujan LDPC code. As can be seen from the code simulation, the suggested algorithm provides LDPC codes with a minimized error floor.

We use the condition number to verify the quality of the constructed parity-check matrix. The condition number, defined for rectangular matrix as the ratio of the maximum singular value to the lowest nonzero singular value of that matrix, which is a measure of the decoder's ability to recover the transmitted code, has been calculated for the matrices constructed in [5], [6], and for the matrices constructed in this article. In this work, we found that the parity-check matrix of the codes [4896, 2448], and [2184, 1092] respectively, are a rank-efficient matrix with condition number values of 6.95 and 5.60 respectively, as compared to its value $8.30e+15$ of [4896, 2448] of [5], [6].

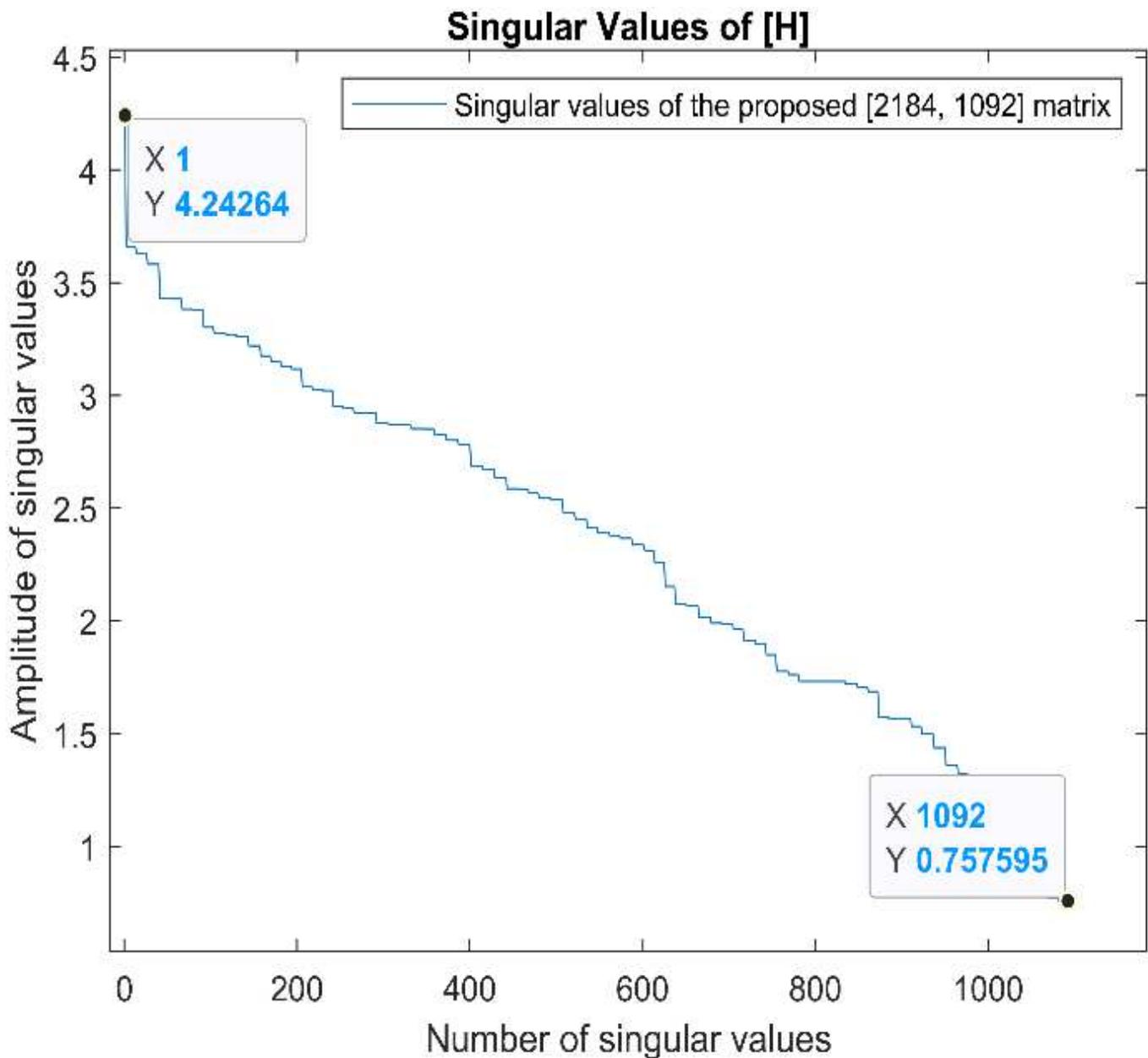


Fig. 3.a. Singular values of the proposed [2184, 1092] parity-check matrix.

The main reason why the condition number of the parity-check matrix of [5], [6] is high is the construction of the parity-check matrix in [5], [6] produces many linearly dependent rows. This property transforms the matrix into a

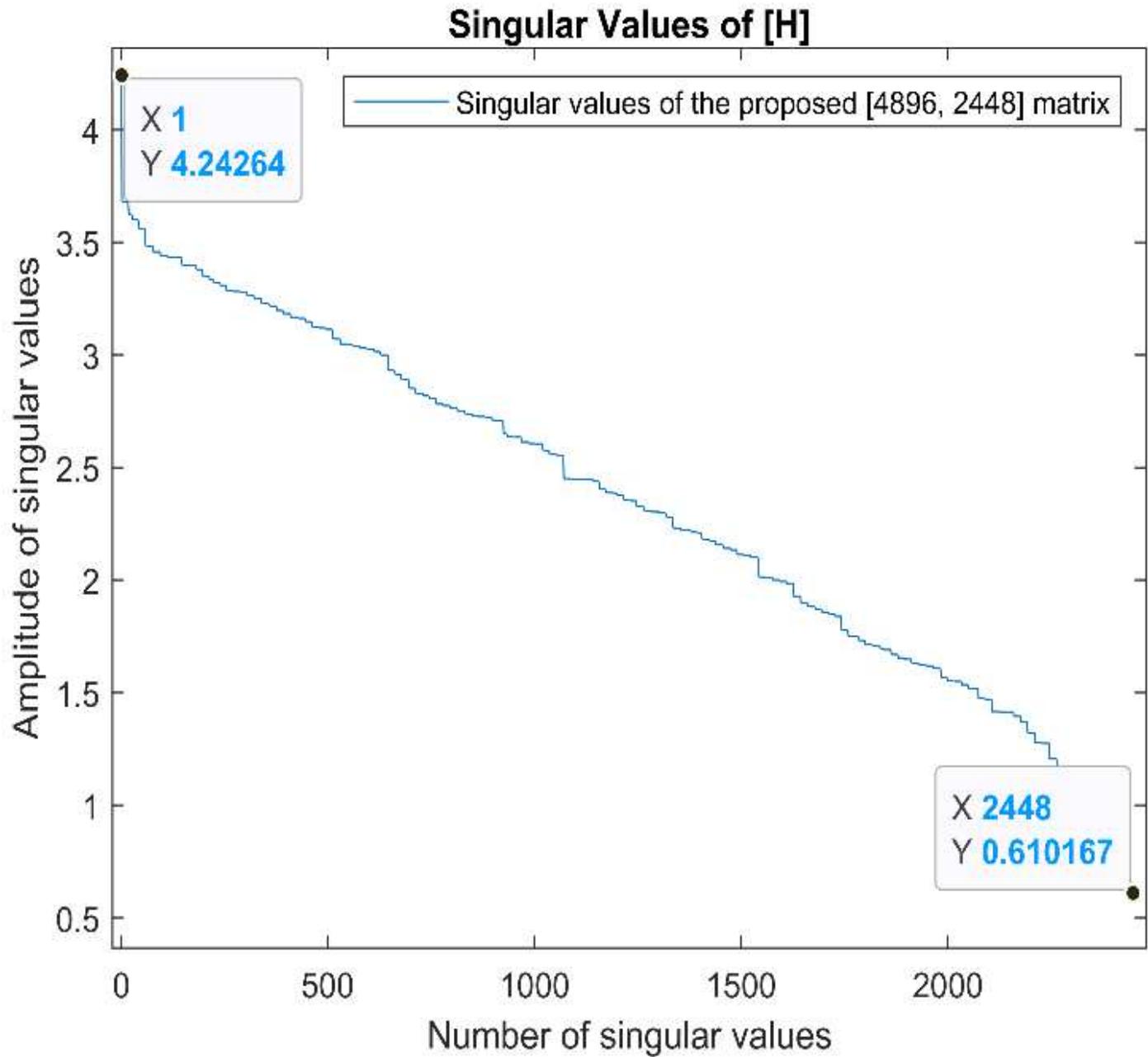


Fig. 3.b. Singular values of the proposed [4896, 2448] parity-check matrix.

singular matrix and produces a large number of errors even when the received word has only fewer errors. However, when a small error occurred in the received noisy word of the proposed parity-check matrices in this study, only a minor error may occur at the decoder since the underlying graph has a small condition value.

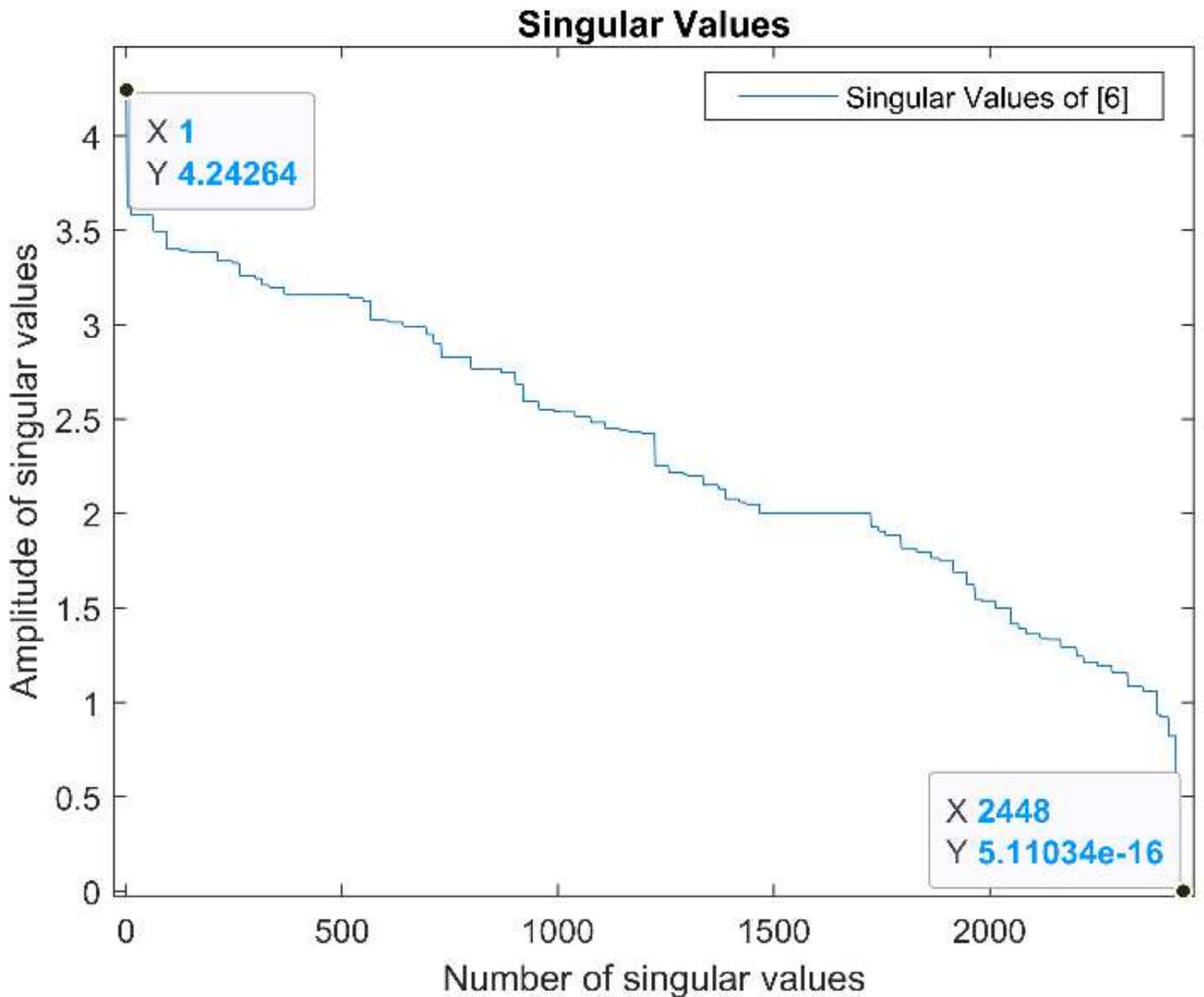


Fig. 3.c. Singular values of [4896, 2448], MacKay's parity-check matrix [6].

Figure 3 shows the singular values of the constructed parity-check matrix from the proposed [2184, 1092] and [4896, 2448] and of [6], [5] coeds, respectively. It appears in figs. 3.a and 3.b that the singular values are symmetric around $\frac{\mu_0-1}{2}$, this may imply that the reason behind this symmetry is that the proposed algorithm generates symmetric well-connected Ramanujan graphs. However, the singular values in fig. 3.c appears not symmetrical, implying that the parity-check matrix of the works in [5], [6] are not symmetric.

4 Conclusions

Prior studies have documented that the algebraic construction of LDPC codes based on Ramanujan graphs generates an error floor while decoding the received words from the channel. However, these studies have not focused on symmetric Ramanujan graphs. In this work, we propose an algorithm in the GAP system to construct highly symmetric Ramanujan-LDPC codes generated from edge-transitive Cayley graphs. The results show that the [2184, 1092] LDPC code reveals little error floor during code simulation compared to [6]. We calculated the condition number of our codes and the code in the studies [6], [5] and found that its value in the proposed codes in this paper is low; however, its value is large in the studies above, which makes the parity-check matrix singular. This is another reason why the decoder shows a high error floor in the literature because the graph is not optimum and lets the decoder generates errors while decoding the received words. However, the decoders based on the LDPCs matrices constructed in this work generate minor errors when a small error occurs in the transmitted codewords due to channel noise. The results show that the code [2184, 1092] can be used in practical applications because the FER and BER performance of this code are around 10^{-5} and 10^{-6} , respectively, as SNR increases. Finally, the proposed codes seem to perform better than the QC LDPC codes that have compact dimensions.

Declarations

Funding No funding was received.

Conflicts of interest The authors declare that they have no conflict of interest.

Availability of data and material All data generated or analysed during this study, including the results and conclusions, are included in this article.

Code availability The author used MATLAB based on the algorithm described in detail in the manuscript.

References

- [1] T. M. Cover and J. A. Thomas, Elements of Information Theory, Hoboken, New Jersey: John Wiley & Sons, Inc, 2006.
- [2] A. Lubotzky, R. Phillips and P. Sarnak, "Ramanujan graphs," *Combinatorica*, vol. 8, no. 3, p. 261–277, 1988.

- [3] G. A. Margulis, "Explicit constructions of graphs without short cycles and low density codes," *Combinatorica*, vol. 2, no. 1, pp. 71-78, 1982.
- [4] R. Gallager, "Low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 8, no. 1, pp. 21-28, 1962.
- [5] J. Rosenthal and P. O. Vontobel, "Constructions of LDPC Codes using Ramanujan Graphs and Ideas from Margulis," in *Allerton Conference on Communication, Control and Computing*, Allerton, 2000.
- [6] D. J. MacKay and M. S. Postol, "Weaknesses of Margulis and Ramanujan-Margulis Low-Density Parity-Check Codes," *Electronic Notes in Theoretical Computer Science*, vol. 74, pp. 97-104, 2003.
- [7] H. Jo and Y. Yamasaki, "LPS-type Ramanujan graphs," in *2018 International Symposium on Information Theory and Its Applications (ISITA)*, 2018.
- [8] P. Chiu, "Cubic Ramanujan graphs," *Combinatorica*, vol. 12, no. 3, pp. 275-285, 1992.
- [9] E. Liao, E. Yeo and B. Nikolic, "Low-density parity-check code constructions for hardware implementation," in *2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577)*, Paris, 2004.
- [10] R. Asvadi, A. H. Banihashemi and M. Ahmadian-Attari, "Lowering the Error Floor of LDPC Codes Using Cyclic Liftings," *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 2213 - 2224, 4 2011.
- [11] S.-R. Kim and D.-J. Shin, "Lowering Error Floors of Systematic LDPC Codes Using Data Shortening," *IEEE Communications Letters*, vol. 17, no. 12, pp. 2348 - 2351, 12 2013.
- [12] S. Tolouei and A. H. Banihashemi, "Lowering the Error Floor of LDPC Codes Using Multi-Step Quantization," *IEEE Communications Letters*, vol. 18, no. 1, pp. 86 - 89, 1 2014.
- [13] H.-C. Lee, P.-C. Chou and Y.-L. Ueng, "An Effective Low-Complexity Error-Floor Lowering Technique for High-Rate QC-LDPC Codes," *IEEE Communications Letters*, vol. 22, no. 10, pp. 1988 - 1991, 10 2018.
- [14] C. Soltanpur, M. Ghamari, B. M. Heravi and F. Zare , "Lowering Error Floors by Concatenation of Low-Density Parity-Check and Array Code," *International Journal of Mathematical and Computational Sciences*, vol. 11, no. 4, 2017.
- [15] B. Karimi and A. H. Banihashemi, "Construction of Irregular Protograph-based QC-LDPC Codes with Low Error Floor," *IEEE Transactions on Communications*, vol. 69, no. 1, 1 2021.

- [16] G. Zhang, Y. Hu, Y. Fang and J. Wang, "Constructions of Type-II QC-LDPC Codes With Girth Eight from Sidon Sequence," *IEEE Transactions on Communications*, vol. 67, no. 6, pp. 3865 - 3878, 6 2019.
- [17] G. Zhang, Y. Hu, D. Ren, Y. Liu and Y. Yang, "Type-II QC-LDPC Codes from Multiplicative Subgroup of Prime Field," *IEEE Access*, vol. 8, pp. 142459 - 142467, 3 8 2020.
- [18] T. Kaufman and A. Lubotzky, "Edge transitive Ramanujan graphs and symmetric LDPC good codes," in *STOC '12: Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, New Yor, 2012.
- [19] "GAP- Groups, Algorithms, Programming - a System for Computational Discrete Algebra," 2020. [Online]. Available: <https://www.gap-system.org/index.html>. [Accessed 2020].
- [20] T. Kaufman and A. Lubotzky, "Edge Transitive Ramanujan Graphs and Highly Symmetric LDPC Good Codes," 2011. [Online]. Available: <https://arxiv.org/abs/1108.2960>.
- [21] M. T. Koetz, "Algebraic Constructions of Low-Density Parity Check Codes," The University of Nebraska, Lincoln, Nebraska, 2005.
- [22] A. Lubotzky, "Expander Graphs in Pure and Applied Mathematics," *Bulletin of the American Mathematical Society*, vol. 49, no. 1, pp. 113 - 162, 1 2012.
- [23] B. Mohar, "Isoperimetric numbers of graphs," *Journal of Combinatorial Theory, Series B*, vol. 47, no. 3, pp. 274-291, 1989.
- [24] T. Kaufman and A. Wigderson, "Symmetric LDPC codes and local testing," *Combinatorica*, vol. 36, pp. 91-120, 2016.
- [25] G. Davidoff, P. Sarnak and A. Valette, *Elementary Number Theory, Group Theory, and Ramanujan Graphs*, Cambridge University Press, 2003.
- [26] M. Koetz, "Algebraic Constructions of Low-Density Parity Check Codes," The Graduate College at the University of Nebraska , Lincoln, Nebraska , 2005.