Quantum Topological Error Correction Codes: The Classical-to-Quantum Isomorphism Perspective

DARYUS CHANDRA, (Student Member, IEEE), **ZUNAIRA BABAR, HUNG VIET NGUYEN,**
DIMITRIOS ALANIS, (Student Member, IEEE), **PANAGIOTIS BOTSINIS,** (Member, IEEE),
SOON XIN NG, (Senior Member, IEEE), **AND LAJOS HANZO**, (Fellow, IEEE)

School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.

Corresponding author: Lajos Hanzo (lh@ecs.soton.ac.uk)

ABSTRACT We conceive and investigate the family of classical topological error correction codes (TECCs), which have the bits of a codeword arranged in a lattice structure. We then present the classical-to-quantum isomorphism to pave the way for constructing their quantum dual pairs, namely, the quantum TECCs (QTECCs). Finally, we characterize the performance of QTECCs in the face of the quantum depolarizing channel in terms of both the quantum-bit error rate (QBER) and fidelity. Specifically, from our simulation results, the threshold probability of the QBER curves for the color codes, rotated-surface codes, surface codes, and toric codes are given by 1.8×10^{-2}, 1.3×10^{-2}, 6.3×10^{-2}, and 6.8×10^{-2}, respectively. Furthermore, we also demonstrate that we can achieve the benefit of fidelity improvement at the minimum fidelity of 0.94, 0.97, and 0.99 by employing the 1/7-rate color code, the 1/9-rate rotated-surface code, and 1/13-rate surface code, respectively.

INDEX TERMS Quantum error correction codes, quantum stabilizer codes, quantum topological codes, lattice code, LDPC.

NOMENCLATURE

A. LIST OF ACRONYMS

BCH	Bose-Chaudhuri-Hocquenghem
CNOT	Controlled-NOT
CSS	Calderbank-Shor-Steane
GV	Gilbert-Varshamov
LDPC	Low Density Parity Check
ML	Maximum Likelihood
PCM	Parity Check Matrix
QBCH	Quantum Bose-Chaudhuri-Hocquenghem
QBER	QuBit Error Rate
QECC	Quantum Error Correction Code
QSC	Quantum Stabilizer Code
QTECC	Quantum Topological Error Correction Code
TECC	Topological Error Correction Code

B. LIST OF SYMBOLS

d	Minimum Distance
F	Fidelity
F_{th}	Threshold Fidelity
$H(x)$	Binary Entropy of x
$\mathbf{H}$	Parity Check Matrix, Hadamard Transformation
k	Information Bit Length, Number of Logical Qubits
n	Codeword Length, Number of Physical Qubits
p	Depolarizing Probability
p_{th}	Threshold Probability
r	Classical Coding Rate
r_Q	Quantum Coding Rate
S_i	Stabilizer Operator
$\mathcal{S}$	Stabilizer Group
t	Error Correction Capability
δ	Normalized Minimum Distance
$\otimes$	Kronecker Tensor Product
$|\psi\rangle$	Quantum State ψ
$\mathcal{C}(n, k, d)$	Classical Error Correction Codes Having Parameter n, k and d
$\mathcal{C}[n, k, d]$	Quantum Stabilizer Codes Having Parameter n, k and d

I. INTRODUCTION

One of the essential prerequisites to build quantum computers is the employment of quantum error correction codes (QECCs) to ensure that the computers operate reliably by mitigating the deleterious effects of quantum decoherence [1]–[3]. However, the law of quantum mechanics prevent us from transplanting classical error correction codes directly into the quantum domain. In order to circumvent the constraints imposed by the nature of quantum physics, the notion of quantum stabilizer codes (QSCs) emerged [4]–[6]. The invention of QECCs and specifically the QSC formalism did not immediately eradicate all of the obstacles of developing reliable quantum computers. Employing the QSCs requires redundancy in the form of auxiliary quantum bits (qubits) to encode the logical qubits onto physical qubits. The redundant qubits are then utilized to invoke the error correction. Hence, additional components such as the quantum encoder and decoder circuits built from quantum gates are required. Therefore, the employment of a QSC itself has to be fault-tolerant to guarantee that the QSC circuit does not introduce additional decoherence into the quantum computers.

The notion of QSC trigered numerous discoveries in the domain of QECCs, which are inspired by classical error corrrection codes. Essentially, QSCs represent the quantum version of the classical syndrome decoding-based error correction codes. Since the concept of utilizing the syndrome values for error correction is widely exploited in the classical domain, diverse classical error correction codes can be conveniently "quantumized". Consequently, we can find in the literature the quantum version of error correction codes based on algebraic formalisms such as those of the Bose-Chaudhuri-Hocquenghem (BCH) codes [7] and of Reed-Solomon (RS) codes [8], quantum codes based on a coventional trellis structure such as convolutional codes [9] and turbo codes [10], [11], as well as quantum codes based on bipartite graphs, such as low density parity check (LDPC) codes [12]–[16]. Another approach that can be exploited to develop both classical and quantum error correction codes hinges on code constructions based on lattice or topological structures. Unfortunately, this concept has not been widely explored in the classical domain. By contrast, in the quantum domain, having a code construction relying on the physical configuration of qubits is highly desirable for the low-complexity high-reliability quantum computers.

The development of QECCs was inspired by Shor [17], who proposed a 9-qubit code. The 9-qubit code, which is also referred to as Shor's code, can protect 9 physical qubits from any type of quantum errors, namely bit-flips (**X**), phase-flips (**Z**), as well as from simultaneous bit and phase-flips (**Y**). Not long after the discovery of the first QECCs, Steane invented the 7-qubit code, which was followed by Laflamme's perfect 5-qubit code [18], [19]. However, the construction of these codes does not naturally exhibit inherent fault-tolerance. The quantum circuit based implementation of these codes always involves a high number of qubit interactions within the codeword of physical qubits. As a consequence, an error caused by a faulty gate within either the encoder, or within the stabilizer measurement, and/or in the inverse encoder potentially propagates to other qubits and instead of being eliminated, the deleterious effects of quantum decoherence are actually further aggravated.
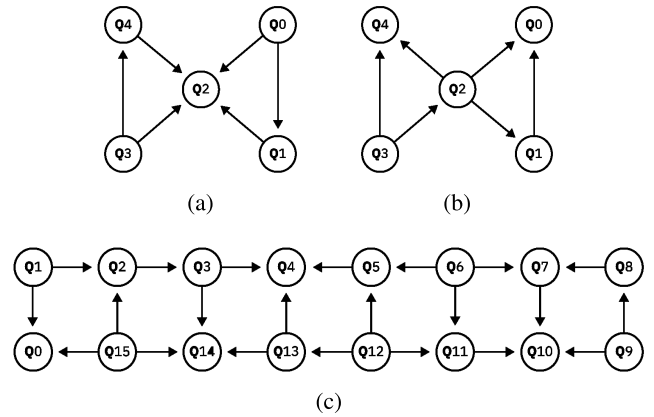

FIGURE 1. The qubit arrangement of IBM's superconducting quantum computers. The circles represent the qubits, while the arrows represent the possible qubit interactions within the computers [20]. (a) 5 qubits (ibmqx2). (b) 5 qubits (ibmqx4). (c) 16 qubits (ibmqx5).

The quantum version of the classical topological error correction codes (TECCs) [21], namely the quantum topological error correction codes (QTECCs), constitute beneficial fault-tolerant QSCs for improving quantum computer implementations. Firstly, they are capable of supporting the physical implementation of quantum memory. For instance, this strategy has been deployed for developing the IBM's superconducting quantum computers, as shown in Fig. 1. From this figure, we can see the qubit arrangement of the three prototypes of IBM's quantum computer - which can be viewed online - namely the ibmqx2, ibmqx4, and ibmqx5 configurations [20]. The first two of the quantum computers are the 5-qubit quantum computers, while the last one is a 16-qubit quantum computer. The circles in Fig. 1 represent the qubits, while the arrows represent all the possible two-qubit interactions. It can be clearly seen that the existing architectures impose a limitation, namely the two-qubit interactions can be only performed between the neighbouring qubits. Even though this particular limitation potentially imposes additional challenges, when it comes to QSCs deployment, the stabilizer effect can still be achieved by the corresponding qubit arrangement by invoking the QTECCs. Secondly, the locality of stabilizer measurements minimizes the requirements imposed on the corresponding quantum gates. The interdependence of the qubits within the codeword are inevitable. However, the interaction between the most distant qubits should be avoided, which imposes challenges on the realization. Another property that makes the QTECCs fault-tolerant is their growing minimum distance as a function of codeword length. More explicitly, the growing minimum
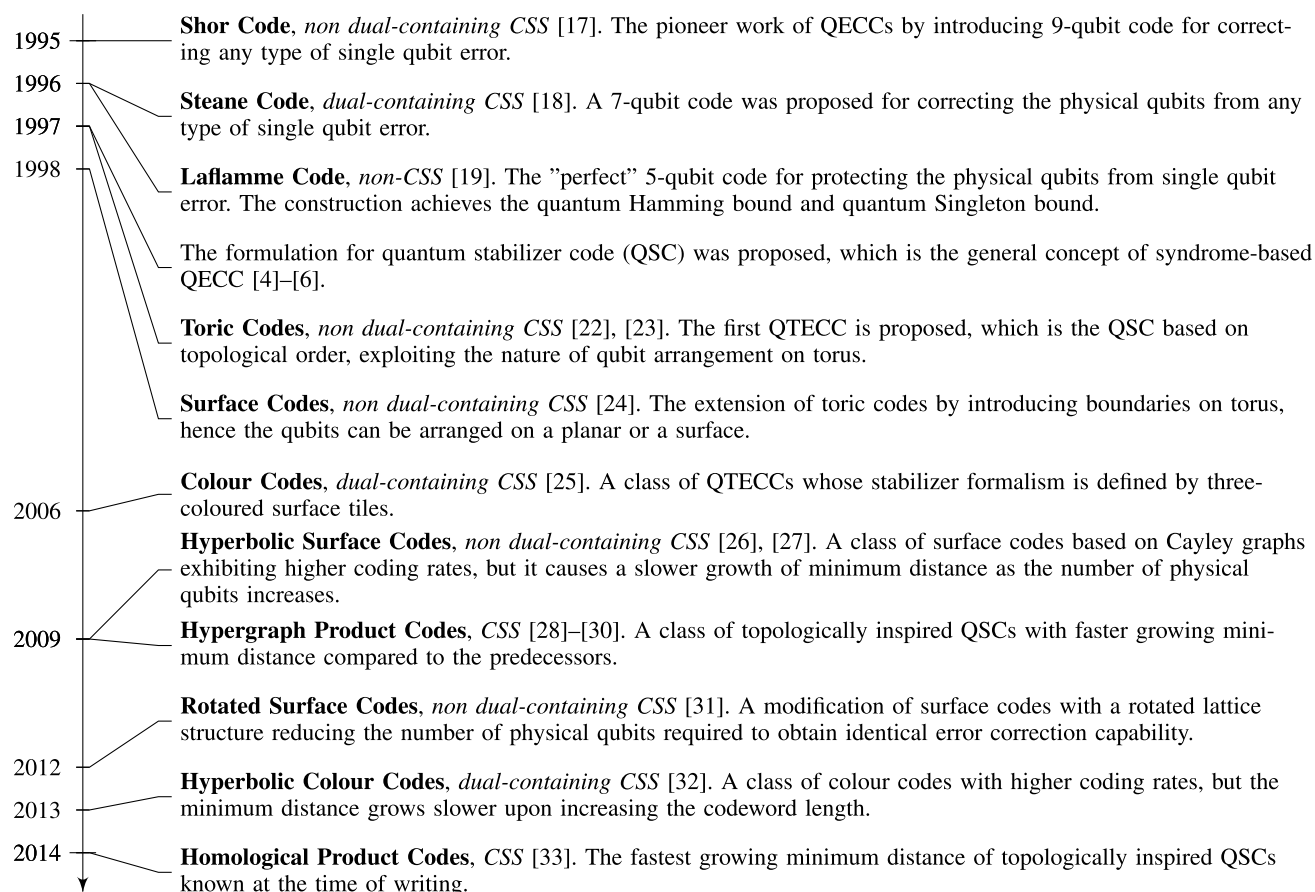
1995 — **Shor Code**, *non dual-containing CSS* [17]. The pioneer work of QECCs by introducing 9-qubit code for correcting any type of single qubit error.

1996 — **Steane Code**, *dual-containing CSS* [18]. A 7-qubit code was proposed for correcting the physical qubits from any type of single qubit error.

1997 —

1998 — **Laflamme Code**, *non-CSS* [19]. The "perfect" 5-qubit code for protecting the physical qubits from single qubit error. The construction achieves the quantum Hamming bound and quantum Singleton bound.

The formulation for quantum stabilizer code (QSC) was proposed, which is the general concept of syndrome-based QECC [4]–[6].

Toric Codes, *non dual-containing CSS* [22], [23]. The first QTECC is proposed, which is the QSC based on topological order, exploiting the nature of qubit arrangement on torus.

Surface Codes, *non dual-containing CSS* [24]. The extension of toric codes by introducing boundaries on torus, hence the qubits can be arranged on a planar or a surface.

2006 — **Colour Codes**, *dual-containing CSS* [25]. A class of QTECCs whose stabilizer formalism is defined by three-coloured surface tiles.

Hyperbolic Surface Codes, *non dual-containing CSS* [26], [27]. A class of surface codes based on Cayley graphs exhibiting higher coding rates, but it causes a slower growth of minimum distance as the number of physical qubits increases.

2009 — **Hypergraph Product Codes**, *CSS* [28]–[30]. A class of topologically inspired QSCs with faster growing minimum distance compared to the predecessors.

Rotated Surface Codes, *non dual-containing CSS* [31]. A modification of surface codes with a rotated lattice structure reducing the number of physical qubits required to obtain identical error correction capability.

2012 —

2013 — **Hyperbolic Colour Codes**, *dual-containing CSS* [32]. A class of colour codes with higher coding rates, but the minimum distance grows slower upon increasing the codeword length.

2014 — **Homological Product Codes**, *CSS* [33]. The fastest growing minimum distance of topologically inspired QSCs known at the time of writing.

FIGURE 2. Timeline of important milestones in the area of QTECCs. The code construction is highlighted with **bold** while the associated code type is marked in *italics*.

distance ensures having an increasing error correction capability per codeword for the QTECCs upon increasing the codeword length, albeit this does not necessarily increase the per-bit normalized error correction capability. To elaborate a little further, increasing the number of physical qubits[1] also increases the number of qubit interactions within the block. Thus, the per-codeword error correction capability of the code should grow fast enough to compensate for the potential error propagation, which may further aggravate the effect of quantum decoherence. The latter phenomenon is also related to the problem experienced in the classical coding theory field, associated with the trade-off between the coding rate and the error correction capability of the error correction code. The study of this particular trade-off in QSCs is a pivotal subject, because we can simply decrease the coding rate further and further to achieve a certain error correction capability without considering the sheer amount of redundant resources wasted, when aiming for achieving the target performance. Therefore, a comprehensive investigation related to this particular trade-off has to be conducted for characterizing the performances versus code parameters. A timeline portraying the

important milestones of the QTECCs' development is depicted in Fig. 2.[2]

Based on the aforementioned background, our novel contributions are:

1) *We conceive the construction of classical error correction codes based on topological or lattice structures. Additionally, we demonstrate for a long codeword that the resultant codes have a resemblance to the classical LDPC codes exhibiting reasonable code parameters.*
2) *We present a tutorial on both classical and quantum topological error correction codes as well as the classical-to-quantum isomorphism along with the comparative study of code parameters.*
3) *We derive the upper bound QBER performance of the QTECCs in the face of quantum depolarizing channel and the formula to determine the threshold fidelity.*

The structure of the paper is described in Fig. 3 and the rest of this treatise is organized as follows. In Section II, we commence with design examples of classical TECCs to pave the

---

[1]The terms 'number of physical qubits' is usually used to refer the 'codeword length' in quantum codes.

[2]Shor's, Steane's and Laflamme's codes do not belong to the QTECCs family. However, we believe that it is still important to include the three pioneering contributions on QECCs in the timeline for the sake of completeness.
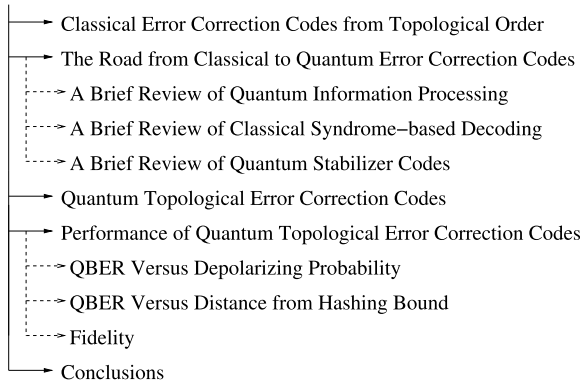
FIGURE 3. The structure of the paper.

FIGURE 4. Example of a classical bit arrangement on a square lattice structure. The black circles laying on the edges of the lattice denote the bits of the codeword, while the vertices of the lattice denoted by red squares define the parity check matrix and also the syndrome values.

way for delving into the quantum domain. In Section III, we provide a tutorial on the fundamentals of QSCs by exploiting its isomorphism with the classical syndrome-based decoding, while in Section IV we detail our QSC design examples for QTECCs. We continue by characterizing the performance of QTECCs over the popular quantum depolarizing channel in terms of QBER and fidelity in Section V. Finally, we conclude our discussion in Section VI.

II. CLASSICAL ERROR CORRECTION CODES FROM TOPOLOGICAL ORDER: DESIGN EXAMPLES

As we mentioned earlier in Section I, the classical error correction codes can be developed relying on diverse approaches [34]. We can find in the literature various family of codes based on algebraic formalisms (such as BCH codes and RS codes), codes based on conventional trellis structures (such as convolutional codes and turbo codes) and also codes based on bipartite graphs (such as LDPC codes). Another approach that can be adopted to formulate a classical error correction code is by exploiting the topological or lattice structure. By assuming that we can arrange the bits of a codeword on a lattice structure, it can inherently provide us with an error correction scheme [21]. For instance, let us assume that a codeword of classical bits is arranged on the square lattice given in Fig. 4. The black circles laying on the edges of the lattice define the encoded information bits or the codeword. The red squares laying on the vertices of the lattice define the parity check matrix (PCM) of the codes, which also directly defines the syndrome values of the received codeword. The number of black circles is associated with the codeword length of n bits and the number of red squares is associated with the length of the syndrome vector or the number of rows of the PCM, which is equal to $(n − k)$ bits. For the particular square lattice seen in Fig. 4, the codeword length n is equal to 13 bits and the length $(n − k)$ of the syndrome vector is equal to 6 bits. Hence, the number of information bits k is equal to 7 bits. Therefore, this code has $2^7 = 128$ legitimate codewords out of the $2^{13} = 8192$ possible received words. Based on the above-mentioned construction, for example in classical BCH codes, we would be able to distinguish $2^{(13−7)} = 2^6 = 64$
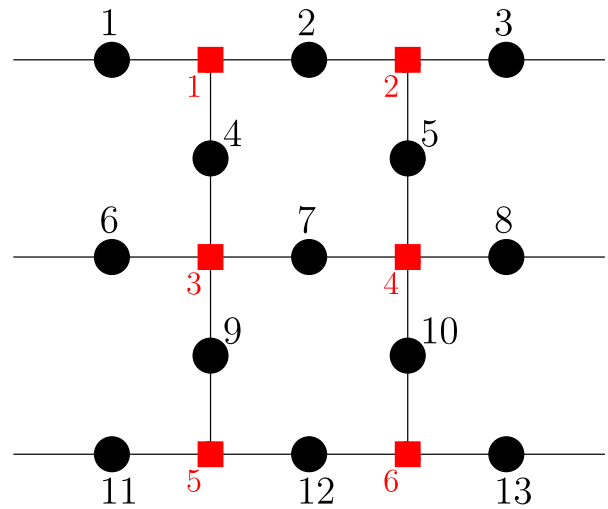
distinct error patterns (including the error free scenario) and correct a single bit error based on sphere packing bound.

The coding rate r is defined by the ratio between the number of information bits k to the codeword length n, yielding:

$$r = \frac{k}{n} \tag{1}$$

Hence, the coding rate of the square lattice code of Fig. 4 is $r = 7/13$.

Now, let us delve deeper into how the error correction works. Let us revisit the square lattice of Fig. 4. The k information bits are encoded to n-bit codewords, where $n > k$. Noise or decoherence imposed by the channel corrupts the legitimate codeword. The syndrome computation is invoked to generate the $(n−k)$-bit syndrome vector, which tells us both the predicted number and the position of the errors. In Fig. 4, the i-th red square indicates a syndrome bit of s_i. Hence, the syndrome vector $\mathbf{s}$ is a 6-bit vector, which is given by

$$\mathbf{s} = [s_1 \quad s_2 \quad s_3 \quad s_4 \quad s_5 \quad s_6]. \tag{2}$$

In the case of an error-free received codeword, the resultant syndrome vector is $\mathbf{s} = [0\ 0\ 0\ 0\ 0\ 0]$. By contrast, if an error is imposed on the codeword, it triggers a syndrome bit value of 1 at the adjacent syndrome bit positions. For example, if an error occurs at the bit index 4 of Fig. 4, it triggers the syndrome values of $s_1 = 1$ and $s_3 = 1$. The rest of the syndrome values remain equal to 0. Therefore, an error corrupting the bit index 4 generates a syndrome vector of $\mathbf{s} = [1\ 0\ 1\ 0\ 0\ 0]$. Hence, the decoder flips the value of bit index 4. Similarly, if an error occurs at bit number 3, it only triggers the syndrome value of $s_2 = 1$. Hence, it generates the syndrome vector of $\mathbf{s} = [0\ 1\ 0\ 0\ 0\ 0]$ and the error recovery procedure proceeds accordingly.

Now let us consider the ocurrence of two bit errors in the codeword. For instance, let us assume that errors occur at

bit indices of 6 and 7 of Fig. 4. Note that both these errors affect s_3, therefore they cancel each other effect on s_3 out, hence generating a syndrome bit value of $s_3 = 0$. However, we still do not receive an all-zero syndrome vector, because the bit index 7 results in a syndrome bit value of $s_4 = 1$ of Fig. 4. Therefore, the resultant syndrome vector due to a bit error in both bit 6 and 7 is $\mathbf{s} = [0\ 0\ 0\ 1\ 0\ 0]$. Since the syndrome vector of $\mathbf{s} = [0\ 0\ 0\ 1\ 0\ 0]$ is also associated with the error incident upon bit index 8, the error recovery procedure decides to flip bit 8 instead, because a single error occurance is more likely to happen than a double-error when the error probability less than $1/2$. This example is an illustration that the occurence of two bit errors in the codeword is beyond the error correction capability of the code given in Fig. 4. We conclude that the code based on the square lattice illustrated in Fig. 4 is capable of correcting only a single bit error. The error correction capability of t bits for a given code construction is defined by the minimum distance d of the code as formulated by

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor. \tag{3}$$

Hence, a code that is only capable of correcting a single error has a minimum distance of $d = 3$, as exemplified by the square lattice code given in Fig. 4. Moreover, the minimum distance of a square lattice code is defined by the dimension of the lattice. Therefore, to increase the error correction capability of the code, we can simply increase the dimension of the lattice, which directly translates into the increase of the minimum distance. The square lattice considered in our example can be generalized to a rectangular lattice structure having a dimension of $(l \times h)$, where l is the length of the lattice and h is the height of the lattice. In the case of a rectangular structure, the minimum distance is defined by

$$d = \min(l, h). \tag{4}$$

The codeword length is also uniquely defined by the dimension of the lattice. More explicitly, for a rectangular lattice of dimension $(l \times h)$, the codeword length is equal to the number of the lattice edges, which is given by

$$n\text{-edges} = n_{\text{square}} = 2lh - l - h + 1. \tag{5}$$

The number of rows in the PCM of a square lattice code is defined by the number of faces or plaquettes of the rectangular lattice, which is formulated as follows:

$$n\text{-vertices} = n_{\text{square}} - k_{\text{square}} = h(l-1). \tag{6}$$

Hence, from Eq. (5) and (6), the number of information bits k encoded by the rectangular lattice codes is

$$k_{\text{square}} = n_{\text{square}} - (n_{\text{square}} - k_{\text{square}})$$
$$= lh - l + 1. \tag{7}$$

The most efficient code can be constructed by a square lattice, where $d = l = h$. Therefore, the expression given in Eq. (5) and (7) can be simplified to

$$n_{\text{square}} = 2d^2 - 2d + 1 \tag{8}$$

$$k_{\text{square}} = d^2 - d + 1. \tag{9}$$

Hence, the coding rate of square lattice based codes can be formulated as follows:

$$r_{\text{square}} = \frac{k_{\text{square}}}{n_{\text{square}}} = \frac{d^2 - d + 1}{2d^2 - 2d + 1}. \tag{10}$$

TABLE 1. Constructing the PCM of the square lattice code of Fig. 4 with minimum distance of $d = 3$. Each row is associated with the syndrome operators denoted by red squares in Fig. 4.

	1	2	3	4	5	6	7	8	9	10	11	12	13
$\mathbf{h}_1$	1	1	0	1	0	0	0	0	0	0	0	0	0
$\mathbf{h}_2$	0	1	1	0	1	0	0	0	0	0	0	0	0
$\mathbf{h}_3$	0	0	0	1	0	1	1	0	1	0	0	0	0
$\mathbf{h}_4$	0	0	0	0	1	0	1	1	0	1	1	0	0
$\mathbf{h}_5$	0	0	0	0	0	0	0	0	1	0	1	1	0
$\mathbf{h}_6$	0	0	0	0	0	0	0	1	0	1	0	0	1

The PCM can be readily constructed in a similar fashion. Each red square of Fig. 4 represents the row of the PCM, where the adjacent black circles denote the index of the column containing a value of 1. For example, the first red square is adjacent to the black circles numbered 1, 2, and 4. Therefore, in the first row of the PCM, there are only three elements containing a value of 1 and those are marked by the index 1, 2, and 4. The remaining rows of the PCM are generated using the same principle. Explicitly, each row of the PCM of the square lattice code of Fig. 4 is portrayed in Table 1. Finally, the PCM $\mathbf{H}$ of the square lattice code of Fig. 4 is given by

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \\ \mathbf{h}_4 \\ \mathbf{h}_5 \\ \mathbf{h}_6 \end{bmatrix}. \tag{11}$$

The code construction based on the general lattice structure is not limited to a rectangular lattice. Let us consider, for example the triangular lattice of Fig. 5. The black circles laying on the vertex of the lattice define the codeword and the red squares on the faces of the lattice define the syndrome vector. The error correction principle of the triangular lattice code is similar to that of its square counterpart. Hence, the PCM of the triangular lattice code is readily derived using the following equation:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{bmatrix}, \tag{12}$$

where $\mathbf{h}_1$, $\mathbf{h}_2$, and $\mathbf{h}_3$ correspond to the syndrome bits given in Table 2. It is important to point out that the resultant triangular lattice code is one of the possible construction for the classical $\mathcal{C}(7, 4, 3)$ Hamming code. Specifically, both codes have a codeword length of $n = 7$ and number of information bits of $k = 4$. Hence, the length of syndrome vector is 3 bits.
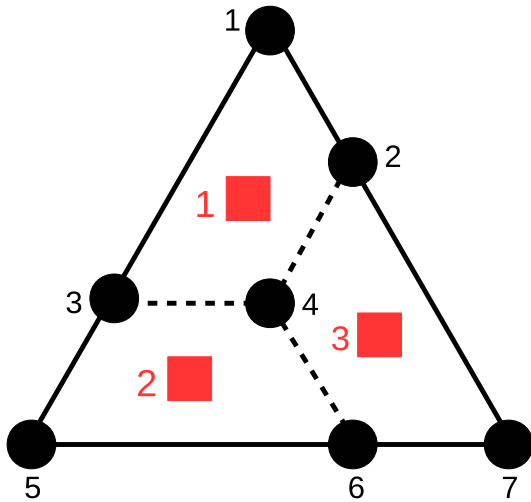
FIGURE 5. Example of a classical bit arrangement constructed over a triangular lattice structure. The black circles laying on the vertices of the lattice represent the codeword bits, while the faces or the plaquettes of the lattice denoted by red squares define the parity-check matrix and the syndrome bits of the error correction code. This configuration is an alternative representation for the $\mathcal{C}(7, 4, 3)$ classical Hamming code.

TABLE 2. Constructing the PCM of the triangular lattice code with minimum distance of $d = 3$. Each row is associated with the syndrome operators denoted by blue circles in Fig. 5.

	1	2	3	4	5	6	7
$\mathbf{h}_1$	1	1	1	1	0	0	0
$\mathbf{h}_2$	0	0	1	1	1	1	0
$\mathbf{h}_3$	0	1	0	1	0	1	1

Consequently, the codes have 2^4 legitimate codewords out of the possile 2^7 received words. Based on the sphere packing bound, the codes are capable of distinguishing $2^3 = 8$ distinct error patterns including the error-free scenario. Therefore, both constructions are capable of correcting exactly a single error with an identical coding rate of $r = 4/7$.

Similar to its rectangular counterpart, increasing the error correction capability of a triangular lattice code is achieved by expanding the underlying lattice configuration. However, increasing the number of vertices of the triangular lattice structure is not as straightforward as that of its rectangular counterpart because it can be carried out in several differennt ways. In this example, we use the construction proposed in [25] and Fig. 6 illustrates how to increase the number of encoded bits of the triangular lattice code of Fig. 5 by using hexagonal tiles.

Following the pattern of Fig. 6, the codeword length, which is also given by the number of vertices of the given lattices, is explicitly formulated as follows:

$$n\text{-vertices} = n_{\text{triangular}} = \frac{1}{4}(3d^2 + 1), \quad (13)$$

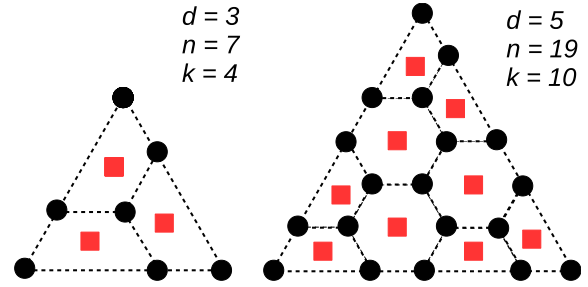where d is the minimum distance of the code. The number of faces in the triangular lattice, which corresponds to the

FIGURE 6. Extending the length of the triangular lattice code, which directly increases the numbers of error corrected.

number of rows of the PCM and also to the syndrome vector length, can be encapsulated as

$$n\text{-faces} = n_{\text{triangular}} - k_{\text{triangular}} = \frac{1}{8}(3d^2 - 3). \quad (14)$$

Hence, the number of information bits can be expressed as

$$k_{\text{triangular}} = n_{\text{triangular}} - (n - k)_{\text{triangular}}$$
$$= \frac{1}{8}(3d^2 + 5). \quad (15)$$

Finally, the coding rate of the triangular lattice codes of Fig. 6 is formulated as follows:

$$r_{\text{triangular}} = \frac{k_{\text{triangular}}}{n_{\text{triangular}}} = \frac{3d^2 + 5}{2(3d^2 + 1)}. \quad (16)$$

Then, the normalized minimum distance, which directly corresponds to the error correction capability per-bit of a code may be defined as:

$$\delta = \frac{d}{n} \quad (17)$$

For square lattice and triangular lattice codes, the normalized minimum distances are given by

$$\delta_{\text{square}} = \frac{d}{2d^2 - 2d + 1}$$
$$\delta_{\text{triangular}} = \frac{4d}{3d^2 + 1}. \quad (18)$$

In the rest of this treatise, we will consider the family of error correction codes based on lattice structures as a prominent representative of classical topological error correction codes (TECC). The lattice structures given in Fig. 4 and 5 can be transformed to Tanner graphs [35]. The dual representation of TECCs in the rectangular lattice domain and in the Tanner graph domain is given in Fig. 7 as exemplified by the square lattice code. We can observe that TECCs based on square lattices have a maximum row weight of $\rho_{\max} = 4$ and a maximum column weight of $\gamma_{\max} = 2$. By contrast, the codes based on triangular lattices have $\rho_{\max} = 6$ and $\gamma_{\max} = 3$. For a very long codeword, these properties lead to sparse PCMs. Hence, classical TECCs can be viewed as a specific family of LDPC codes. The asymptotical limit of the coding rate for LDPC codes based on TECCs can be directly derived from Eq. (10) and (16). As the codeword length tends to infinity
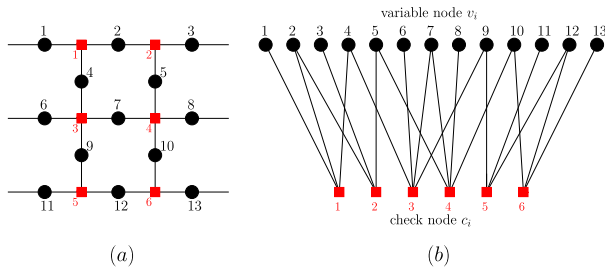
FIGURE 7. Example of how to represent the square lattice code. (a) The representation in lattice structure. (b) The representation in Tanner or bipartite graph.

$(n \to \infty)$, the minimum distance d is also expected to tend to infinity. Hence, at the asymptotical limit we have

$$r_{\text{square}}^{\infty} = \lim_{d \to \infty} \frac{d^2 - d + 1}{2d^2 - 2d + 1} = \frac{1}{2}, \qquad (19)$$

$$r_{\text{triangular}}^{\infty} = \lim_{d \to \infty} \frac{3d^2 + 5}{2(3d^2 + 1)} = \frac{1}{2}. \qquad (20)$$

TABLE 3. Code parameters of classical Hamming code having a single error correction capability, which is used in Fig. 8 and 9. The coding rate r and normalized minimum distance δ are calculated using Eq. (1) and (17), respectively.

n	k	d	n	k	d
3	1	3	127	120	3
7	4	3	255	247	3
15	11	3	511	502	3
31	26	3	1023	1013	3
63	57	3	...	...	...

Let us observe Fig. 8, where we plot the minimum distance (d) versus coding rate (r) of TECCs based on Eq. (10) and (16). We also include the classical codes based on the sphere packing concept, namely the Hamming codes and the BCH codes, whose parameters are portrayed in Table 3 and 4, respectively. We also include some labels for several codes in the figure, in order to show how to convert the code parameters into data points in the figure. More explicitly, let us consider the specific triangular codes T1 and T2, where T1 represents the triangular code having a minimum distance of 3, which we have already used in the example in Fig. 5. As it has been elaborated on earlier, the resultant code T1 is $\mathcal{C}(7, 4, 3)$. Hence, the coding rate is $r = 4/7 \approx 0.57$. Again, the triangular code T1 has identical code parameters to the Hamming code $\mathcal{C}(7, 4, 3)$, which is labeled H1. Hence, the same point in Fig. 8 represents both T1 and H1. Next, the code parameters of the triangular code T2 having a minimum distance of $d = 5$ are obtained using Eq. (13) and (15) for determining the codeword length n and the information length k, respectively. Explicitly, by substituting $d = 5$ into Eq. (13) and (15), we have $n = 19$ and $k = 10$. Finally, we arrive at the coding rate of $r = k/n = 10/19 \approx 0.53$ for the triangular code T2. The rest of the code parameters for square

TABLE 4. Code parameters of classical BCH codes having codeword length of $n = 255$, which is used in Fig. 8 and 9. The coding rate r and normalized minimum distance δ are calculated using Eq. (1) and (17), respectively.

n	k	d	n	k	d	n	k	d
255	1	255	255	87	53	255	171	23
255	9	127	255	91	51	255	179	21
255	13	119	255	99	47	255	187	19
255	21	111	255	107	45	255	191	17
255	29	95	255	115	43	255	199	15
255	37	91	255	123	39	255	207	13
255	45	87	255	131	37	255	215	11
255	47	85	255	139	31	255	223	9
255	55	63	255	147	29	255	231	7
255	63	61	255	155	27	255	239	5
255	71	59	255	163	25	255	247	3
255	79	55						

codes, triangular codes, Hamming codes and BCH codes are protrayed in the same way in Fig. 8.

In general, increasing the minimum distance of the codes while mantaining the codeword length can be achieved at the expense of reducing the coding rate. This penomenon is perfectly reflected by the behaviour of classical BCH codes in Fig. 8. Explicitly, in Fig. 8 we portray BCH codes having a constant codeword length of $n = 255$, which are described in Table 4. As seen, upon increasing the minimum distance of BCH codes, the coding rate is gradually reduced. Next, increasing the coding rate while maintaining the minimum distance of the code can indeed be achieved by increasing the codeword length. In this case, the Hamming codes, whose code parameters are described in Table 3, reflect perfectly this phenomenon. Observe in Fig. 8, that for the Hamming codes exhibiting a constant minimum distance of $d = 3$, we can see the gradual increase of coding rate upon increasing the codeword length. However, the behaviour of the BCH and Hamming codes is not reflected by the TECCs. Let us elaborate on the TECCs behaviour in Fig. 8. The increase of minimum distance of TECCs upon increasing the codeword length looks very impressive, since they do not seem to require much sacrifice in terms of coding rate reduction. In fact, the coding rate is saturated at approximately $r = 1/2$ for long codewords. This is indeed a rather different behaviour compared to that of the classical BCH codes. However, it is of pivotal importance to mention again that the increasing error correction capability per codeword does not necessarily imply the improvement of error correction capability per bit. Therefore, we have to normalize the performance to the codeword length in order to portray a fair comparison.

Let us now observe Fig. 9, where we plot the normalized minimum distance (δ) versus the coding rate (r) of TECCs based on Eq. (18). We include both the BCH codes as well as the Hamming codes for the sake of comparison. We also plot the classical Hamming bound [36] and Gilbert-Varshamov (GV) [37] bound in this figure to portray
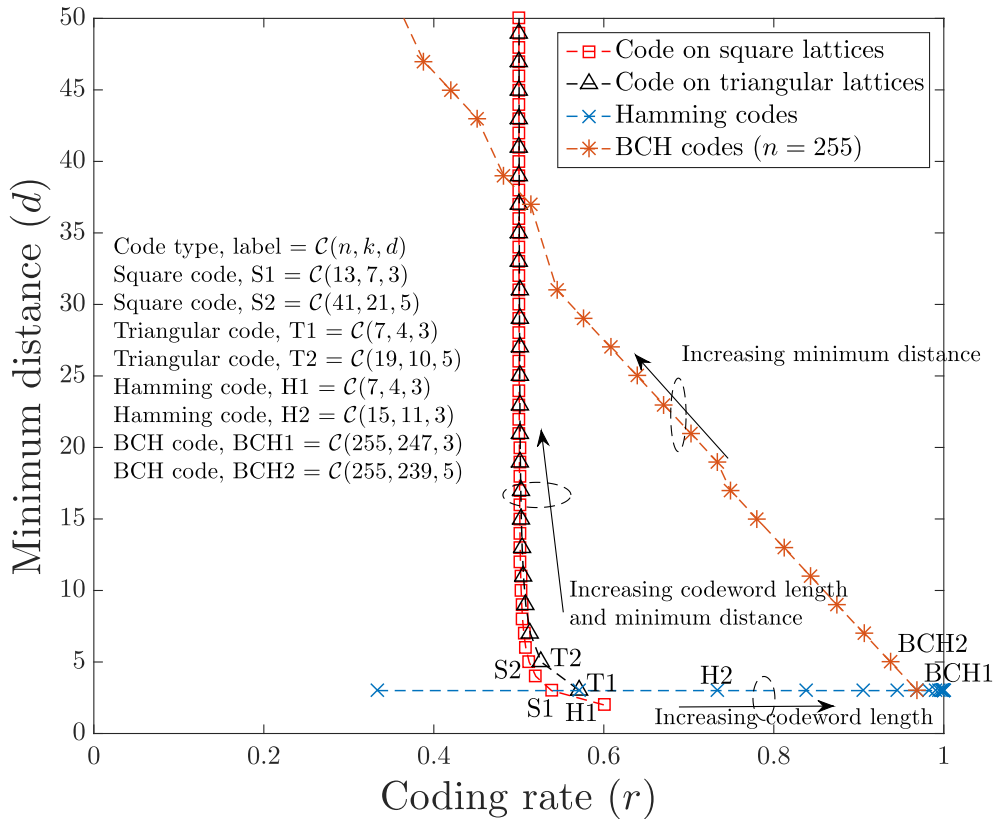
FIGURE 8. The coding rate versus minimum distance of TECCs. For asymptotical limit, the TECCs may be categorized into LPDC codes and the coding rates converge to $r = \frac{1}{2}$. We also include the BCH codes and Hamming codes for the sake of comparison. The coding rate for the square lattice based codes and the triangular lattice based codes are defined in Eq. (10) and (16), respectively. The code parameters for classical Hamming and BCH codes are described in Table 3 and 4, respectively. We put labels only for several codes as examples on how to convert the given code parameters into the figure.

the upper bound and lower bound of the normalized minimum distance, which correspond directly to the normalized error correction capability, given the coding rate. The classical Hamming bound is formulated as follows [36]:

$$\frac{k}{n} \leq 1 - H\left(\frac{d}{2n}\right), \tag{21}$$

where $H(x)$ is the binary entropy of x defined by $H(x) = -x\log_2 x - (1-x)\log_2(1-x)$, while the classical GV bound is expressed as [37]

$$\frac{k}{n} \geq 1 - H\left(\frac{d}{n}\right). \tag{22}$$

The classical Hamming bound and GV bound defined in Eq. (21) and (22) are valid for asymptotical limit where $n \to \infty$.

The classical Hamming codes constitute the so-called *perfect codes* for a finite-length, since they always achieve the Hamming bound for finite-length codes.[3] Therefore, the Hamming codes also mark the upper bound of normalized minimum distance, given the coding rate of finite-length

---

[3]The Hamming bound for finite length codes has a different formulation from that of asymptotical limit. Therefore, we refer to [38] for further explanations.

codewords. Secondly, the classical BCH codes having a codeword length of $n = 255$ lay perfectly - as expected - between the Hamming and GV bound in the asymptotical limit, as shown in Fig. 9. However, we observe an unusual behaviour for the family of TECCs, since the normalized minimum distance drops to zero upon increasing the codeword length, while the coding rate saturates at $r = 1/2$. We hypothesize that since these codes were not designed using the sphere packing concept - which the Hamming and BCH codes are based on - the Hamming distance radius of the associated decoding sphere in the TECCs codespace is most likely to be non-identical for the different codewords. In addition, the minimum distance of TECCs is only on the order of $\mathcal{O}(\sqrt{n})$, which implies that the codeword length of TECCs is proportional to the factor of $\mathcal{O}(d^2)$. By contrast, for clasical BCH and Hamming codes the growth of the minimum distance is approximately linear, i.e. of order $\mathcal{O}(n)$. It is clearly seen that even though the growth of minimum distance per codeword of the TECCs appears to be impressive in Fig. 8, it is not fast enough to compensate for the undesired effect of the increasing codeword length. Hence, the TECC error correction capability per bit tends to zero in the asymptotical limit. Nevertheless, we leave the definitive answer for this peculiar phenomenon open for future research, since our
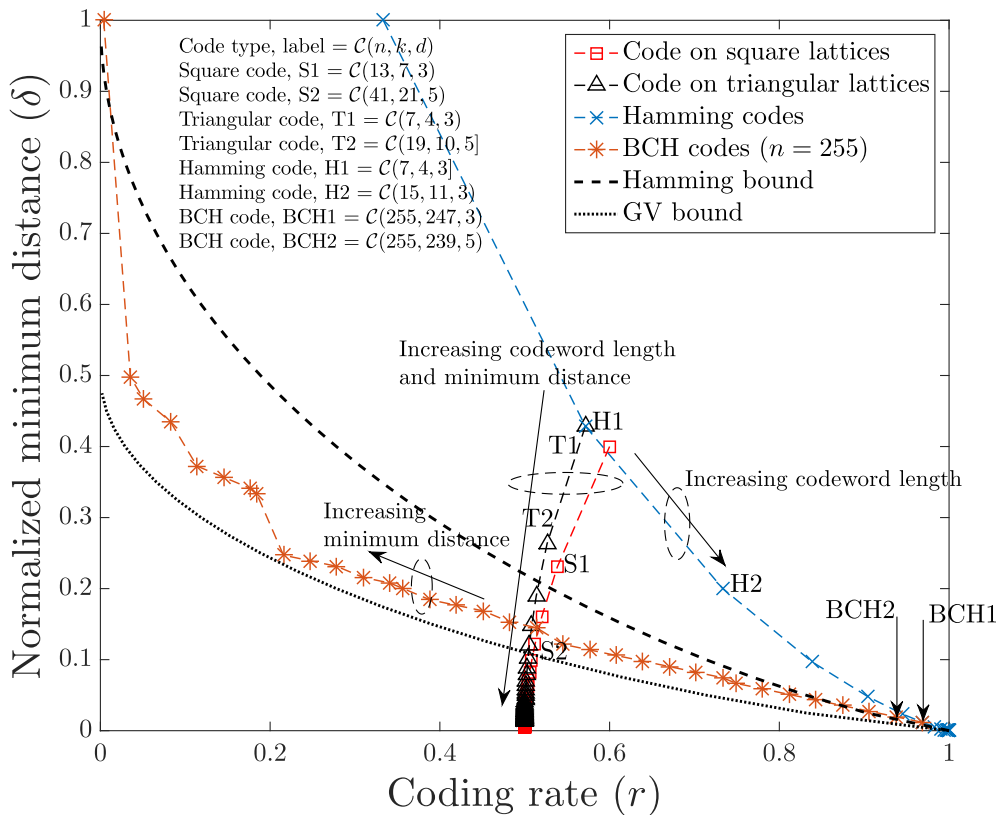
FIGURE 9. The coding rate versus normalized minimum distance of TECCs. For asymptotical limit, the TECCs may be categorized into LPDC codes and the coding rates converge to $r = \frac{1}{2}$, while the normalized minimum distances (δ) vanish to zero. In addition, we also include the classical Hamming and BCH codes, which constructed based on sphere packing bound, for the sake of comparison. The code parameters for classical Hamming and BCH codes are portrayed in Table 3 and 4, respectively. We put labels only for several codes as examples on how to convert the given code parameters into the figure.

focus in this treatise is on finding the classical-to-quantum isomorphism of TECCs.

Since the TECC associated with the asymptotical limit of $n \rightarrow \infty$ belongs to the family of LDPC codes, an efficient LDPC decoder such as the belief propagation (BP) technique [39] can be invoked for these code constructions. However, the normalized minimum distance of the LDPC codes based on topological order tends to zero, as the codeword length increases. Nevertheless, TECC-based LDPC codes exhibit several desirable code properties, such as an attractive coding rate ($r \approx 1/2$), structured construction and unbounded minimum distance. However, another aspect worth considering for TECC-based LDPC codes is the fact that we can find numerous cycles of length 4 in triangular constructions and cycles of length 6 in square constructions, which potentially degrades the performances of the codes. A brief summary of code parameters of TECC-based LDPC codes is given in Table 5.

III. THE ROAD FROM CLASSICAL TO QUANTUM ERROR CORRECTION CODES

In this section, we provide a brief review of quantum information processing. This will be followed by a rudimentary

TABLE 5. The code parameters of TECC-based LDPC codes.

Parameter	Square lattice	Triangular lattice
r	$\approx \frac{1}{2}$	$\approx \frac{1}{2}$
d	$\mathcal{O}(\sqrt{n})$	$\mathcal{O}(\sqrt{n})$
δ	$\frac{d}{2d^2 - 2d + 1}$	$\frac{4d}{3d^2 + 1}$
$\rho_{\max}$	4	6
$\gamma_{\max}$	2	3
Girth	6	4

introduction of classical syndrome-based decoding and how we can demonstrate the isomorphism towards quantum stabilizer codes.

A. A BRIEF REVIEW OF QUANTUM INFORMATION PROCESSING

In the classical domain the information is represented by a series of binary digits (bits), whilst in the quantum domain the information is conveyed by quantum bits (qubits). A classical bit can only hold a value of either '0' or '1' at a time, while the qubit can hold the value of '0', '1' and the superposition of both values. More specifically, the state of a single qubit

can be expressed mathematically as follows:

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle, \quad \alpha_0, \alpha_1 \in \mathbb{C}, \tag{23}$$

where $P_0 = |\alpha_0|^2$ and $P_1 = |\alpha_1|^2$ are the probability of obtaining the value of 0 and 1 upon measurement, respectively. Hence, the unitary constraint of having $|\alpha_0|^2 + |\alpha_1|^2 = 1$ is applied. Representing the pure states of '0' by the notation $|0\rangle$ and the pure state of '1' by the so-called *ket* notation $|1\rangle$,[4] as shown in Eq. (23), is referred to as the Dirac notation [40]. The pure state of $|0\rangle$ and $|1\rangle$ can also be represented as a 2-element vector in the Hilbert space $\mathcal{H}$ as follows:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{24}$$

Hence, substituting the vectors given in Eq. (24) into Eq. (23) yields:

$$|\psi\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}, \quad \alpha_0, \alpha_1 \in \mathbb{C}. \tag{25}$$

The state of a single qubit can be manipulated by using the quantum unitary transformations. A unitary transformation of U may be realized by a quantum gate, which is the elementary building block of quantum computers. All of the quantum domain unitary transformations are represented by unitary matrices to ensure that the final probability of quantum states remains 1, which can be explicitly formulated as

$$U^\dagger U = \mathbf{I}, \tag{26}$$

where $\mathbf{I}$ is an identity matrix. The Pauli gates or Pauli operators constitute a collection of unitary transformations representing the discrete set of errors that may be imposed on a single qubit. The Pauli operators are defined using the Pauli matrices, as follows:

$$\mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$
$$\mathbf{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{27}$$

The Pauli matrices can be physically interpreted as a bit-flip error, phase-flip error as well as both bit-flip and phase-flip error for the Pauli matrix $\mathbf{X}$, $\mathbf{Z}$ and $\mathbf{Y}$, respectively. The Pauli-$\mathbf{I}$ matrix is an identity matrix corresponding to the absence of errors.

The error imposed on multi-qubit systems can be described using the Kronecker tensor product. Explicitly, for the matrices $\mathbf{P}$ and $\mathbf{Q}$ having $(a \times b)$ elements and $(x \times y)$ elements, respectively, the resultant Kronecker product is a matrix having $(ax \times by)$ elements formulated by

$$\mathbf{P} \otimes \mathbf{Q} = \begin{pmatrix} p_{11}\mathbf{Q} & \cdots & p_{1(b-1)}\mathbf{Q} & p_{1b}\mathbf{Q} \\ p_{21}\mathbf{Q} & \cdots & p_{2(b-1)}\mathbf{Q} & p_{2b}\mathbf{Q} \\ \vdots & \ddots & \vdots & \vdots \\ p_{(a-1)1}\mathbf{Q} & \cdots & p_{(a-1)(b-1)}\mathbf{Q} & p_{(a-1)b}\mathbf{Q} \\ p_{a1}\mathbf{Q} & \cdots & p_{a(b-1)}\mathbf{Q} & p_{ab}\mathbf{Q} \end{pmatrix}. \tag{28}$$

[4] The terminology *ket* comes from the *bra-ket* notation. The *bra* notation refers to the $\langle\psi|$ notation, while *ket* notation is used for $|\psi\rangle$ notation.

For instance, a two-qubit system is represented by the Kronecker product between a pair of two-element vectors given in Eq. (24). More explicitly, let us consider the qubit having the state of $|\psi_1\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ and another one in the state of $|\psi_2\rangle = \beta_0|0\rangle + \beta_1|1\rangle$. The superimposed state can be described as follows:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \otimes \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{pmatrix}$$
$$\equiv \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle, \tag{29}$$

where $\alpha_0, \alpha_1, \beta_0, \beta_1 \in \mathbb{C}$. It can be observed that a two-qubit state is a superposition of all four possible states that can be generated by two bits i.e. 00, 01, 10 and 11. Moreover, the unitary condition of $|\alpha_0\beta_0|^2 + |\alpha_0\beta_1|^2 + |\alpha_1\beta_0|^2 + |\alpha_1\beta_1|^2 = 1$ still holds. The Kronecker product of a pair of two-element vectors yields a vector consisting of 2^2 elements. Hence, the N-qubit systems yield all of the 2^N possible states that can be generated by an N-bit sequence. If i is the decimal representation of an N-bit sequence, the N-qubit superposition state can be expressed by the Dirac notation as follows:

$$|\psi\rangle = \sum_{i=0}^{2^N-1} \alpha_i|i\rangle \quad \text{where } \alpha_i \in \mathbb{C} \text{ and } \sum_{i=0}^{2^N-1} |\alpha_i|^2 = 1. \tag{30}$$

Since the N-qubit state is represented by a 2^N-element column vector, the unitary transformation of the N-qubit system is defined by a $(2^N \times 2^N)$ elements unitary matrix. In quantum communication, the quantum decoherence may impose a bit-flip error, phase-flip error, as well as both bit-flip and phase-flip error. For the sake of modeling the behaviour of quantum information in the presence of quantum impairments, the Pauli channel model is widely used [41]. To elaborate a little further, the Pauli channel inflicts an error $\mathcal{P} \in \mathcal{G}_n$ on the state of an N-qubit system, where each qubit may independently experience either a bit-flip error ($\mathbf{X}$), a phase-flip error ($\mathbf{Z}$), or both bit-flip and phase-flip error ($i\mathbf{XZ} = \mathbf{Y}$). For an N-qubit system, the general Pauli group $\mathcal{G}_n$ is represented by an N-fold tensor product of $\mathcal{G}_1$, as described below:

$$\mathcal{G}_n = \{P_1 \otimes P_2 \cdots \otimes P_n | P_j \in \mathcal{G}_1\}, \tag{31}$$

where the Pauli group $\mathcal{G}_1$ is constituted by the unitary transformations applied to a single qubit state, which is closed under multiplication and is explicitly defined as follows:

$$\mathcal{G}_1 = \{eP : P \in \{\mathbf{I}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}, e \in \{\pm 1, \pm i\}\}. \tag{32}$$

The laws of quantum mechanics prevent us from directly transplanting the classical error correction codes into the quantum domain owing to the following obstacles:

1) **No Cloning Theorem**. In the classical domain, the basic technique of protecting the information bits in repetition coding is that of copying the same information several times. By contrast, in the quantum domain, this simple approach cannot be implemented, since no
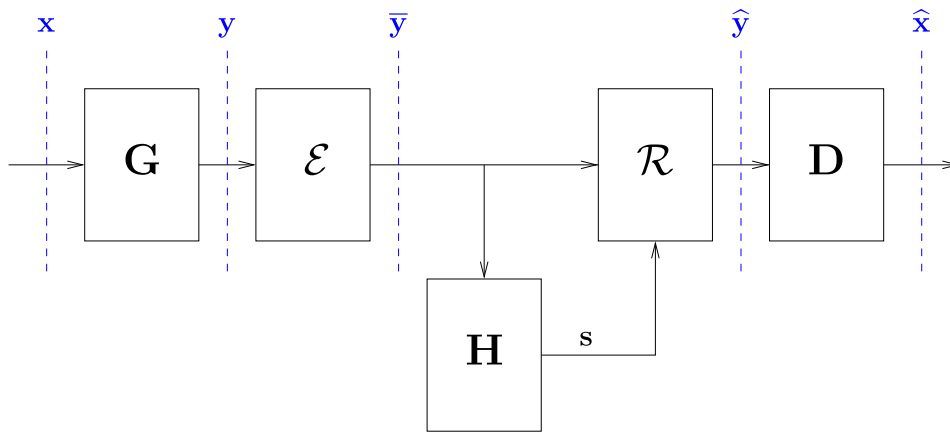
FIGURE 10. The basic model of classical error correction codes invoking syndrome-based decoding. The operation **G** denotes the generator matrix, which maps the *k* information bits **x** to the *n* coded bits **y**. The channel $\mathcal{E}$ inflicts an error vector $\mathbf{e} \in \{0, 1\}^n$ upon the codeword **y**, resulting in the corrupted received bits $\mathbf{\bar{y}}$. The receiver calculates the syndrome vector **s** based on the PCM **H** and the received bits $\mathbf{\bar{y}}$ to predict the number and the position of errors contained in the received bits $\mathbf{\bar{y}}$. The error recovery $\mathcal{R}$ generates the error recovery vector **r**, which is applied to the received bits $\mathbf{\bar{y}}$. This operation collapses the received bits $\mathbf{\bar{y}}$ to one of the legitimate codeword **y**, yielding the predicted codeword $\mathbf{\hat{y}}$. Finally, we can readily determine the predicted information bits $\mathbf{\hat{x}}$ from the predicted codeword $\mathbf{\hat{y}}$.

unitary quantum transformation is capable of performing this specific task.

2) **The quantum bit collapses into the corresponding classical bit upon measurement**. In the classical domain, the error correction schemes are typically fed by measuring the bits received at the output of the demodulator. In the quantum domain, measuring the qubits represented by the superposition of the classical states will collapse the superposition into a single classical post-measurement state and consequently we lose the original quantum information.

3) **QECCs have to handle not only bit-flip errors, but also phase-flip errors, as well as the simultaneous bit-flip and phase-flip errors**. By contrast, in the classical domain, we deal with a single type of error, which is the bit-flip error. In quantum domain, the nature of quantum decoherence is continuous and it can be modeled as a linear combination of bit-flip errors (**X**), phase-flip errors (**Z**), or both bit-flip and phase-flip errors ($i\mathbf{XZ} = \mathbf{Y}$). However, thanks to the beneficial effect of the stabilizer measurement, the continuous nature of quantum decoherence can be treated as a discrete set of independent errors imposed on the physical qubits.

Albeit all of the aforementioned obstacles hindering the development of QECC schemes, the invention of QSC formulation succeeded in circumventing these problems.

B. A BRIEF REVIEW OF CLASSICAL SYNDROME-BASED DECODING

As mentioned earlier, the problems revolving around the QECCs are effectively circumvented by QSCs, which essentially constitute the syndrome-based decoding version of

QECCs. Hence, for the sake of sheding some light onto the parallelism between the classical and quantum regime, we proceed with the classical syndrome-based decoding first.

In the classical domain a $\mathcal{C}(n, k)$ code maps k information bits into n coded bits, where $k < n$. The purpose of attaching $(n - k)$ redundant bits is to facilitate error detection or even error correction. Let us refer to Fig. 10 and consider the classical $\mathcal{C}(7, 4)$ Hamming code, which maps 4 information bits into 7 coded bits and hence becomes capable of correcting a single error. In general, the mapping of the k information bits is performed by multiplying the information row vector **x** consisting of k elements by the generator matrix **G** having $(k \times n)$ elements. Explicitly, the mapping can be formulated as

$$\mathbf{y} = \mathbf{x} * \mathbf{G}, \qquad (33)$$

where the resultant codeword **y** is a row vector having n elements, while the notation of $*$ represents the matrix multiplication over modulo-2. For instance, the generator matrix of the $\mathcal{C}(7, 4)$ Hamming code is defined by

$$\mathbf{G}_{\text{Hamming}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \qquad (34)$$

From Eq. (33) and (34) we can generate the code space mapping shown in Table 6, where $\mathbf{x}_i$ denotes all the possible combination of information bits and $\mathbf{y}_i$ represents the associated legitimate codeword bits.

The generator matrix **G** can be arranged into a systematic form as

$$\mathbf{G} = (\mathbf{I}_k | \mathbf{P}), \qquad (35)$$

where $\mathbf{I}_k$ is a $(k \times k)$ identity matrix and **P** is a matrix having $k \times (n - k)$ elements. The form given in Eq. (35) generates

a systematic codeword $\mathbf{y}$ consisting of the k-bit information word $\mathbf{x}$ followed by $(n-k)$ parity bits. A generator matrix $\mathbf{G}$ is associated with an $(n-k) \times n$-element PCM $\mathbf{H}$, which is defined as

$$\mathbf{H} = \left(\mathbf{P}^T | \mathbf{I}_{n-k} \right). \tag{36}$$

As an example, the generator matrix of the classical $\mathcal{C}(7, 4)$ Hamming code of Eq. (34) is associated with the following PCM:

$$\mathbf{H}_{\text{Hamming}} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}. \tag{37}$$

The PCM of $\mathbf{H}$ is constructed for ensuring that a valid codeword $\mathbf{y}$ satisfies the following requirement:

$$\mathbf{y} * \mathbf{H}^T = \mathbf{0}. \tag{38}$$

A received word $\bar{\mathbf{y}}$ may be contaminated by an error vector $\mathbf{e} \in \{0, 1\}^n$ due to channel impairments, which is denoted by $\mathcal{E}$ in Fig. 10. More explicitly, the resultant received words corrupted by the additive noise $\mathcal{E}$ can be formulated as

$$\bar{\mathbf{y}} = \mathbf{y} + \mathbf{e}. \tag{39}$$

The error syndrome $\mathbf{s}$ is a row vector having $(n-k)$ elements obtained by the following calculation:

$$\begin{aligned} \mathbf{s} = \bar{\mathbf{y}} * \mathbf{H}^T &= (\mathbf{y} + \mathbf{e}) * \mathbf{H}^T \\ &= \mathbf{y} * \mathbf{H}^T + \mathbf{e} * \mathbf{H}^T \\ &= \mathbf{0} + \mathbf{e} * \mathbf{H}^T \\ &= \mathbf{e} * \mathbf{H}^T. \end{aligned} \tag{40}$$

The syndrome vector $\mathbf{s}$ contains the information related to the error pattern imposed by the channel. To elaborate, we have 2^k legitimate codewords generated by the all possible combination of the k information bits, 2^n possible received bit patterns of $\hat{\mathbf{y}}$ and $2^{(n-k)}$ possible syndromes $\mathbf{s}$, each unambiguously identifying one of the $2^{(n-k)}$ error patterns, including the error-free scenario. Hence, for the classical $\mathcal{C}(7, 4)$ Hamming code, the syndrome vector $\mathbf{s}_i$ can detect and correct a single error pattern as specified in Table 7. The error recovery $\mathbf{r}_i$ is determined based on the most likely error pattern. After obtaining the syndrome vector, the recovery vector $\mathbf{r}_i$ is applied to the received words to obtain the predicted codeword $\hat{\mathbf{y}} = \bar{\mathbf{y}} + \mathbf{r}$, as depicted in Fig. 10. The application of the recovery operator $\mathbf{r}_i$ to the received word always collapses it into one of the legitimate codewords $\mathbf{y}$, hence the predicted codeword $\hat{\mathbf{y}}$ can be finally demapped in order to obtain the predicted information bits $\hat{\mathbf{x}}$ using Table 6, as illustrated in Fig. 10. For linear systematic codes, this process can be simply performed by chopping the last $(n-k)$ bits, namely the redundant bits.

For more a detailed example, let us consider k information bits of $\mathbf{x} = (1\,1\,0\,1)$. The information bits are encoded using the classical $\mathcal{C}(7, 4)$ Hamming code employing the generator matrix of Eq. (33), yielding the coded bits of $\mathbf{y} = (1\,1\,0\,1\,1\,0\,0)$. Let us assume that the channel corrupts

TABLE 6. The code space mapping of the $\mathcal{C}(7, 4)$ classical Hamming code.

i	$\mathbf{x}_i$				$\mathbf{y}_i$						
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	1	1	1	1
3	0	0	1	0	0	0	1	0	0	1	1
4	0	0	1	1	0	0	1	1	1	0	0
5	0	1	0	0	0	1	0	0	1	0	1
6	0	1	0	1	0	1	0	1	0	1	0
7	0	1	1	0	0	1	1	0	1	1	0
8	0	1	1	1	0	1	1	1	0	0	1
9	1	0	0	0	1	0	0	0	1	1	0
10	1	0	0	1	1	0	0	1	0	0	1
11	1	0	1	0	1	0	1	0	1	0	1
12	1	0	1	1	1	0	1	1	0	1	0
13	1	1	0	0	1	1	0	0	0	1	1
14	1	1	0	1	1	1	0	1	1	0	0
15	1	1	1	0	1	1	1	0	0	0	0
16	1	1	1	1	1	1	1	1	1	1	1

TABLE 7. The look-up table to determine the most likely error pattern $\mathbf{e}_i \in \mathcal{E}$ that corresponds to the syndrome value $\mathbf{s}_i$, which is created based on Eq. (37) and (40).

i	$\mathbf{s}_i$			$\mathbf{e}_i$						
1	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	1
3	0	1	0	0	0	0	0	0	1	0
4	0	1	1	0	0	1	0	0	0	0
5	1	0	0	0	0	0	0	1	0	0
6	1	0	1	0	1	0	0	0	0	0
7	1	1	0	1	0	0	0	0	0	0
8	1	1	1	0	0	0	1	0	0	0

the legitimate codeword $\mathbf{y}$ by imposing an error pattern of $\mathbf{e} = (1\,0\,0\,0\,0\,0\,0)$ yielding the received word of $\bar{\mathbf{y}} = (0\,1\,0\,1\,1\,0\,0)$. Next, the received word is fed to the syndrome calculation block, which contains the PCM of Eq. (37). Based on Eq. (40), the received word $\bar{\mathbf{y}} = (0\,1\,0\,1\,1\,0\,0)$ generates the syndrome vector of $\mathbf{s} = (1\,1\,0)$. Utilizing the look-up table of Table 7, the error recovery vector becomes $\mathbf{r} = (1\,0\,0\,0\,0\,0\,0)$. Upon applying the error recovery vector, the received word $\bar{\mathbf{y}}$ is collapsed to one of the legitimate codewords $\mathbf{y}$ in Table 6, which is $\hat{\mathbf{y}} = (1\,1\,0\,1\,1\,0\,0)$. Assuming that the predicted codeword $\hat{\mathbf{y}}$ is valid, the demapper decides to translate the predicted codeword $\hat{\mathbf{y}} = (1\,1\,0\,1\,1\,0\,0)$ to the predicted information bits as $\hat{\mathbf{x}} = (1\,1\,0\,1)$. Hence, the original information is successfully recovered. The whole process of syndrome calculation, error recovery and demapping jointly form the *decoding* process. It is important to note that in practice, the syndrome calculation, recovery operator and demapper are amalgamated into a single *decoder* block.

Let us now assume that the channel imposes an error pattern beyond the error correction capability of the classical
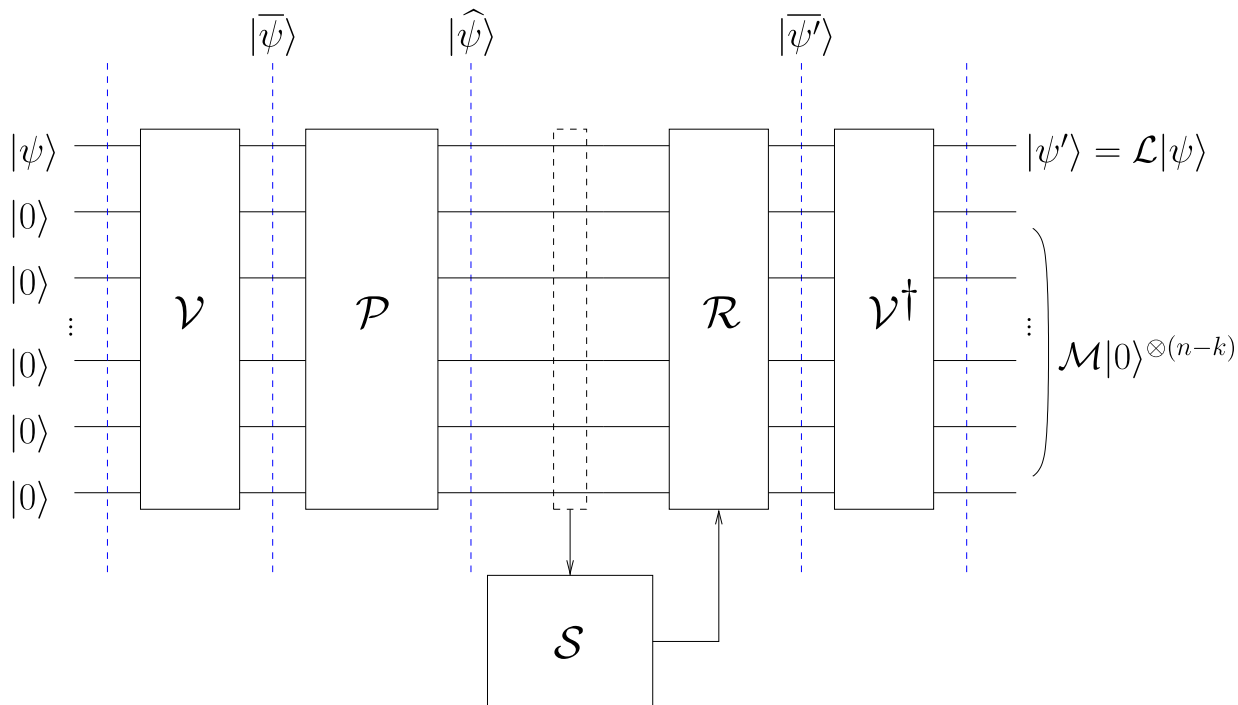
FIGURE 11. The basic model of QSCs implementation over the quantum depolarizing channel. The *k* logical qubits are mapped into *n* physical qubits with the aid of $(n - k)$ redundant/auxiliarry qubits (*ancillas*) to provide protection from the quantum decoherence. This schematic is similar to the classical error correction model where $(n - k)$ redundant bits are added to *k* information bits in order to provide error correction. The quantum encoder $\mathcal{V}$ serves the same purpose as **G** of the classical error correction codes in Fig. 10. The quantum encoder $\mathcal{V}$ transforms the state of *k* logical qubits $|\psi\rangle$ into the state of *n* physical qubits $|\overline{\psi}\rangle$ with the aid of $(n - k)$ ancillas. The quantum depolarizing channel imposes the error vector represented by the *n*-tupple Pauli operator $\mathcal{P} \in \mathcal{G}_n$. The syndrome operators $S_i \in \mathcal{S}$ generate the eigenvalues of ±1, which are analogous to the value 0 and 1 of the classical syndrome vector, which is provided by the PCM **H** in Fig. 10. The error recovery $\mathcal{R}$ applies the correction according to the syndrome values provided by the syndrome measurements. Finally, the quantum-domain inverse encoder $\mathcal{V}^\dagger$ transforms the predicted state of physical qubits $|\overline{\psi}'\rangle$ back to the predicted state of logical qubits $|\psi'\rangle$, which carries out the same function as the demapper **D** in the classical syndrome-based decoding of Fig. 10.

$\mathcal{C}(7, 4)$ Hamming code. For example, assume that we send *k* information bits of $\mathbf{x} = (1\ 1\ 0\ 1)$, similar to that of in the previous example, while the channel inflicts an error pattern of $\mathbf{e} = (1\ 1\ 0\ 0\ 0\ 0\ 0)$. As a result, we have the received codeword bits of $\overline{\mathbf{y}} = (0\ 0\ 0\ 1\ 1\ 0\ 0)$. Based on the received codeword, we have the syndrome vector of $\mathbf{s} = (0\ 1\ 1)$. Based on the syndrome vector, the error recovery of $\mathbf{r} = (0\ 0\ 1\ 0\ 0\ 0\ 0)$ is chosen. Consequently, the error recovery vector collapses the received word to the incorrect legitimate codeword, which is $\widehat{\mathbf{y}} = (0\ 0\ 1\ 1\ 1\ 0\ 0)$, instead of the correct codeword of $\mathbf{y} = (1\ 1\ 0\ 1\ 1\ 0\ 0)$. Since the demapper assumes that the error recovery completes the task perfectly, the demapper decides that the predicted information bits are $\mathbf{x} = (0\ 0\ 1\ 1)$. Compared to the original information bits, the predicted information bits are considered as an error. This example demonstrates that the classical $\mathcal{C}(7, 4)$ Hamming code is unable to operate flawlessly beyond its error correction capability.

C. A BRIEF REVIEW OF QUANTUM STABILIZER CODES
The formulation of QSCs is capable of detecting both the number and the position of errors without actually observing the state of physical qubits, which is vitally important since otherwise the quantum state will collapse to classical bits upon measurement. This was achieved by amalgamating the classical syndrome-based decoding with the QECCs. Similar to classical error correction codes, QSCs also rely on attaching redundant qubits to the information qubits for invoking error correction. The basic model of QSCs is depicted in Fig. 11, which will be contrasted to its classical pair in Fig. 10. In order to generate the codespace $\mathcal{C}$, the redundancy is constituted by $(n - k)$ auxiliary qubits. Next, a unitary transformation $\mathcal{V}$ transforms the *k* qubits in the state of $|\psi\rangle$ and the $(n - k)$ auxiliary qubits into an *n* qubits in the state of $|\overline{\psi}\rangle$. The unitary transformation of $\mathcal{V}$ represents the action of the *quantum encoder*. Explicitly, the mapping of the *logical qubits* constituting the state of $|\psi\rangle \in \mathbb{C}^{2^k}$ to the *physical qubits* forming the state of $|\overline{\psi}\rangle \in \mathbb{C}^{2^n}$ by the encoder $\mathcal{V}$ of Fig. 11 can be mathematically formulated as follows:

$$\mathcal{C} = \{|\overline{\psi}\rangle = \mathcal{V}(|\psi\rangle \otimes |0\rangle^{\otimes(n-k)})\}. \qquad (41)$$

The QSCs rely on the stabilizer operators $S_i \in \mathcal{S}$ for identifying the type, the number and also the position of the qubit errors. A stabilizer operator S_i is an *n*-tuple Pauli operator, which preserves the state of physical qubits as defined below:

$$S_i|\overline{\psi}\rangle = |\overline{\psi}\rangle. \qquad (42)$$

The quantum channel inflicts errors represented by n-tuple Pauli operators $\mathcal{P} \in \mathcal{G}_n$, as given in Eq. (31), which transforms the encoded physical qubits that were originally in the state of $|\overline{\psi}\rangle$ to the potentially corrupted physical qubits in the state of $|\widehat{\psi}\rangle$, as seen in Fig. 11. More explicitly, this process can be described as follows:

$$|\widehat{\psi}\rangle = \mathcal{P}|\overline{\psi}\rangle. \tag{43}$$

The stabilizer operators act similarly to the syndrome calculations routinely used in classical error correction codes. To elaborate a little further, a stabilizer operator will return an eigenvalue of $+1$, when an error operator $\mathcal{P}$ commutes with the stabilizer operator, while we arrive at the eigenvalue of -1, if it anti-commutes. The eigenvalues of $+1$ and -1 are analogous to the classic syndrome bit of 0 and 1, respectively, which can be defined as follows:

$$S_i|\widehat{\psi}\rangle = \begin{cases} |\widehat{\psi}\rangle, & S_i\mathcal{P} = \mathcal{P}S_i \\ -|\widehat{\psi}\rangle, & S_i\mathcal{P} = -\mathcal{P}S_i. \end{cases} \tag{44}$$

Therefore, the stabilizer operators naturally have to inherit the commutative property. Consequently, the product between the stabilizer operators S_i yields another legitimate stabilizer operator. Furthermore, the commutativity property implies that

$$S_i|\overline{\psi}\rangle = S_j|\overline{\psi}\rangle = S_iS_j|\overline{\psi}\rangle = |\overline{\psi}\rangle, \quad \forall S_{i,j} \in \mathcal{S}, \tag{45}$$

suggesting that the stabilizer group $\mathcal{S}$ is closed under multiplication.

Based on the syndrome measurement by the stabilizer operators S_i, a recovery operator constituted by the n-tupple Pauli operator of $\mathcal{R} \in \mathcal{G}_n$ seen in Fig. 11 is applied to the corrupted physical qubit state $|\widehat{\psi}\rangle$, yielding the predicted state of the original encoded logical qubit $|\overline{\psi'}\rangle$, which is formulated as

$$|\overline{\psi'}\rangle = \mathcal{R}|\widehat{\psi}\rangle. \tag{46}$$

Finally, the inverse encoder $\mathcal{V}^\dagger$ of Fig. 11 performs the following transformation[5]:

$$\begin{aligned} \mathcal{V}^\dagger|\overline{\psi'}\rangle &= \mathcal{V}^\dagger\mathcal{R}|\widehat{\psi}\rangle \\ &= \mathcal{V}^\dagger\mathcal{R}\mathcal{P}|\overline{\psi}\rangle \\ &= \mathcal{V}^\dagger\mathcal{R}\mathcal{P}\mathcal{V}(|\psi\rangle \otimes |0\rangle^{\otimes(n-k)}) \\ &= (\mathcal{L}|\psi\rangle) \otimes (\mathcal{M}|0\rangle^{\otimes(n-k)}), \end{aligned} \tag{47}$$

where we have $\mathcal{V}^\dagger\mathcal{R}\mathcal{P}\mathcal{V} \equiv \mathcal{L}\otimes\mathcal{M}$ and $\mathcal{L} \in \mathcal{G}_k$ represents the error inflicted on the logical qubits according to $|\psi'\rangle = \mathcal{L}|\psi\rangle$, while $\mathcal{M} \in \mathcal{G}_{n-k}$ represents the residual error remained in the $(n-k)$ auxiliary qubits after the error correction procedure. In the case of $\mathcal{R} = \mathcal{P}$, we arrive at $\mathcal{R}\mathcal{P} = \mathbf{I}^{\otimes n}$, where $\mathbf{I}^{\otimes n}$ denotes an n-fold tensor product Pauli-$\mathbf{I}$ matrix. Another possibility is to arrive at $\mathcal{R}\mathcal{P} = S_i$. In either of these cases, the state of the physical qubits is not altered, since we have

---

[5]The inverse encoder $\mathcal{V}^\dagger$ is the Hermitian transpose of encoder $\mathcal{V}$. It is referred to as the *inverse*, since it satisfies the unitary requirement of $\mathcal{V}^\dagger\mathcal{V} = \mathbf{I}$, as the inverse of the matrix does.

$\mathcal{R}\mathcal{P}|\overline{\psi}\rangle = |\overline{\psi}\rangle$. Therefore, the decoding procedure of Fig. 11 successfully recovers the original quantum state constituted by the logical qubits, yielding $|\psi'\rangle = |\psi\rangle$.

The stabilizer operators can be translated into the classical PCM $\mathbf{H}$ by mapping the Pauli matrices $\mathbf{I}$, $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ onto $(\mathbb{F}_2)^2$ as follows:

$$\begin{aligned} \mathbf{I} &\to \begin{pmatrix} 0 \mid 0 \end{pmatrix}, \\ \mathbf{X} &\to \begin{pmatrix} 0 \mid 1 \end{pmatrix}, \\ \mathbf{Y} &\to \begin{pmatrix} 1 \mid 1 \end{pmatrix}, \\ \mathbf{Z} &\to \begin{pmatrix} 1 \mid 0 \end{pmatrix}. \end{aligned} \tag{48}$$

This concept is also known as the *Pauli-to-binary isomorphism*. By exploiting the Pauli-to-binary isomorphism, the stabilizer operators of any QSC can be represented as a pair of PCMs $\mathbf{H}_z$ and $\mathbf{H}_z$, where $\mathbf{H}_z$ is invoked for handling the phase-flip ($\mathbf{Z}$) errors and $\mathbf{H}_x$ for handling the bit-flip ($\mathbf{X}$) errors. Explicitly, the classical PCM representation of the QSC stabilizer operators may be written as follows:

$$\mathbf{H} = (\mathbf{H}_z|\mathbf{H}_x). \tag{49}$$

The classical representation of the stabilizer operators gives the advantage of predicting and evaluating the performances of QSCs by treating them similarly to classical error correction codes. Additionally, it allows us to transform a pair of classical PCMs into the correponding quantum counterpart. However, to ensure that the commutative property is preserved in the quantum domain, a pair of classical PCMs have to satisfy the so-called *symplectic criterion* [6] given by

$$\mathbf{H}_z \cdot \mathbf{H}_x^T + \mathbf{H}_x \cdot \mathbf{H}_z^T = 0. \tag{50}$$

A special class of QSCs, namely the family of Calderbank-Shor-Steane (CSS) codes, treats the phase-flip ($\mathbf{Z}$) and bit-flip ($\mathbf{X}$) errors as two separate entities. More specifically, this can be interpreted as having the PCMs of $\mathbf{H}_z$ and $\mathbf{H}_x$ in Eq. (49) formulated as $\mathbf{H}_z = \begin{pmatrix} \mathbf{H}'_z \\ 0 \end{pmatrix}$ and $\mathbf{H}_x = \begin{pmatrix} 0 \\ \mathbf{H}'_x \end{pmatrix}$, respectively. Therefore, the binary PCM $\mathbf{H}$ can be expressed as follows:

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}'_z & 0 \\ 0 & \mathbf{H}'_x \end{pmatrix}. \tag{51}$$

Consequently, the symplectic criterion given in Eq. (50) can be reduced to the following criterion:

$$\mathbf{H}'_z \cdot \mathbf{H}'^T_x = 0. \tag{52}$$

Furthermore, we can formulate a CSS code by using a PCM of $\mathbf{H}'_z = \mathbf{H}'_x$ and the resultant quantum code may be referred to as a dual-containing quantum CSS code or self-orthogonal quantum CSS code. For dual-containing CSS codes, the symplectic criterion can be further simplified to $\mathbf{H}'_z\mathbf{H}'^T_z = 0$. For a more detailed example, please refer to [38].
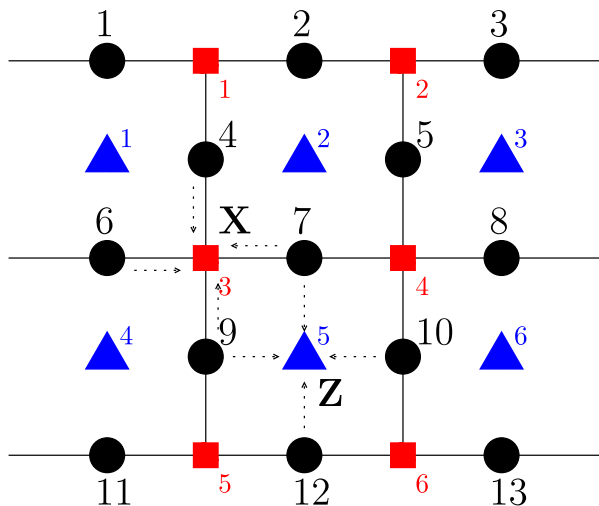
FIGURE 12. Example of qubit arrangement on a rectangular lattice structure. The black circle-based qubits on the edges of the lattice represent the physical qubits or the encoded state, the red square-based qubits lying on the vertices of the lattice act as the **X** stabilizer operators, while the blue triangle-based qubits lying on the plaquettes (faces) of the lattice constitute the **Z** stabilizer operators.

IV. QUANTUM TOPOLOGICAL ERROR CORRECTION CODES: DESIGN EXAMPLES

Let us now delve deeper into the TECC concept in the quantum domain. The quantum version of TECCs, namely the QTECCs, constitute a member of the QSC family, whose stabilizer operators are defined by the underlying lattice structure. This formalism offers several benefits for the implementation of quantum computers. Firstly, it explicitly accommodates the physical implementation of quantum memory by mapping the qubits to the lattice arrangement exemplified by Fig. 4 and 5. Secondly, the localized nature of the stabilizer measurements confines the interaction amongst qubits and also eliminates the interaction of qubits associated with a specific quantum gate that physically far from each other. Thirdly, the number of errors corrected can be increased simply by extending the size of the lattice. For now, let us assume having a square lattice structure similiar to Fig. 4 for defining the stabilizer operators of a surface code illustrated in Fig. 12 [24]. Explicitly, surface codes represent the quantum equivalent of classical TECCs on rectangular lattice structures. The physical qubits are portrayed by the black circles laying on the edge of the lattice, the **X** stabilizer operators are defined by the red squares on the lattice vertices, while the **Z** stabilizers are defined by the blue triangles on the lattice plaquettes (faces). The stabilizer operators of QTECCs are defined as follows:

$$A_v = \prod_{i \in \text{vertex}(v)} \mathbf{X}_i, \quad B_p = \prod_{i \in \text{plaquette}(p)} \mathbf{Z}_i, \quad (53)$$

where i indicates the index of stabilizer operators containing the Pauli matrix **X** as well as **Z** and the rest of the stabilizer operators are given by the Pauli identity matrix **I**. Hence, the encoded state of the physical qubits of QTECCs is

constrained within a code space $\mathcal{C}$ satisfying

$$\mathcal{C} = \{|\overline{\psi}\rangle \in \mathcal{H} | A_v |\overline{\psi}\rangle = |\overline{\psi}\rangle, B_p |\overline{\psi}\rangle = |\overline{\psi}\rangle; \forall v, p\}. \quad (54)$$

More specifically, let us revisit Fig. 12 for exemplifying the construction of the stabilizer operators of a QTECC, namely of the surface codes, which is one of the QTECC constructions whose stabilizer operators are defined by a rectangular lattice structure [24]. For instance, the red square on the vertex number 3 of Fig. 12 represents the **X** stabilizer operator of $A_3 = \mathbf{X}_4 \mathbf{X}_6 \mathbf{X}_7 \mathbf{X}_9$[6] as seen in the row S_3 of Table 8. Similarly, the blue triangle on the plaquette number 5 of Fig. 12 defines the **Z** stabilizer operator of $B_5 = \mathbf{Z}_7 \mathbf{Z}_9 \mathbf{Z}_{10} \mathbf{Z}_{12}$ as seen in the line B_5 of Table 8. By performing the same evaluation for all of the red squares and blue triangles, we arrive at the stabilizer operators for the quantum surface codes, as listed in Table 8.

TABLE 8. The stabilizer operators (S_i) of the quantum surface code having the lattice construction of Fig. 12. The code has a minimum distance of 3 ($d = 3$), which means that it is only capable of correcting a single qubit error.

S_i	A_v	S_i	B_p
S_1	$\mathbf{X}_1\mathbf{X}_2\mathbf{X}_4$	S_7	$\mathbf{Z}_1\mathbf{Z}_4\mathbf{Z}_6$
S_2	$\mathbf{X}_2\mathbf{X}_3\mathbf{X}_5$	S_8	$\mathbf{Z}_2\mathbf{Z}_4\mathbf{Z}_5\mathbf{Z}_7$
S_3	$\mathbf{X}_4\mathbf{X}_6\mathbf{X}_7\mathbf{X}_9$	S_9	$\mathbf{Z}_3\mathbf{Z}_5\mathbf{Z}_8$
S_4	$\mathbf{X}_5\mathbf{X}_7\mathbf{X}_8\mathbf{X}_{10}$	S_{10}	$\mathbf{Z}_6\mathbf{Z}_9\mathbf{Z}_{11}$
S_5	$\mathbf{X}_9\mathbf{X}_{11}\mathbf{X}_{12}$	S_{11}	$\mathbf{Z}_7\mathbf{Z}_9\mathbf{Z}_{10}\mathbf{Z}_{12}$
S_6	$\mathbf{X}_{10}\mathbf{X}_{12}\mathbf{X}_{13}$	S_{12}	$\mathbf{Z}_8\mathbf{Z}_{10}\mathbf{Z}_{13}$

Let us now consider an example of how the error correction procedure works using the QTECCs, which is similar to the classical TECCs, by revisiting Fig. 12. For instance, let assume that the quantum decoherence imposes a bit-flip (**X**) error on the physical qubit index 7. Since, the **X**-type error commutes with the **Z** stabilizer operators, which are represented by the blue triangles, the adjacent **Z** stabilizer operators return the eigenstate values of −1 upon measurement. Consequently, the **Z** stabilizer measurements yield a syndrome vector of $\mathbf{s}_z = [0\ 1\ 0\ 0\ 1\ 0]$, where only the vector elements of $i = 2, 5$ have the value of 1. For the short block code considered in Fig. 12, the error recovery operators $\mathcal{R}$ of Fig. 11 are determined based on hard-decision maximum-likelihood (ML) decoding, which is translated into a simple look-up table (LUT) decoder. Therefore, based on the syndrome vector of $\mathbf{s}_z$, the error recovery operator $\mathcal{R}$ of Fig. 11 is given by $\mathcal{R} = \mathbf{X}_7$. Likewise, let us now assume that the qubit on index 7 also suffers from a **Z**-type error imposed by the quantum channel. The associated syndrome vector gleaned from the **X** stabilizer operators is $\mathbf{s}_x = [0\ 0\ 1\ 1\ 0\ 0]$, where only the vector elements of $i = 3, 4$ have the value of 1. Thus, based on the syndrome vector of $\mathbf{s}_x$, the decoder applies the error recovery operator of $\mathcal{R} = \mathbf{Z}_7$.

---

[6]This representation is used for simplifying the original stabilizer operator of $A_3 = \mathbf{I}_1 \otimes \mathbf{I}_2 \otimes \mathbf{I}_3 \otimes \mathbf{X}_4 \otimes \mathbf{I}_5 \otimes \mathbf{X}_6 \otimes \mathbf{X}_7 \otimes \mathbf{I}_8 \otimes \mathbf{X}_9 \otimes \mathbf{I}_{10} \otimes \mathbf{I}_{11} \otimes \mathbf{I}_{12} \otimes \mathbf{I}_{13}$. For the rest of this paper, the simplified notation is used.
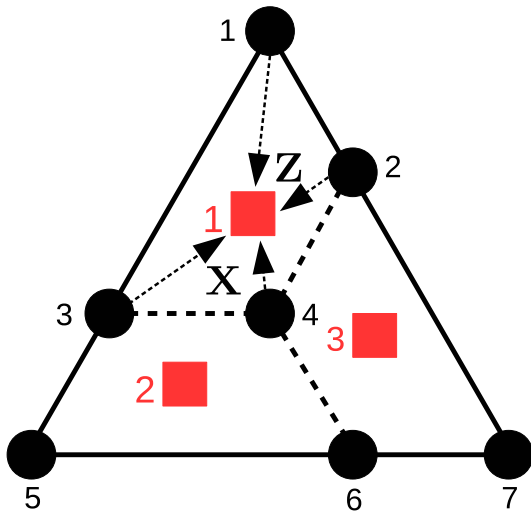
FIGURE 13. Example of a qubit arrangement for colour code, which is a type of QTECCs whose stabilizer operators are defined by a triangular lattice structure. The black circles-based qubits on the vertices of the lattice represent the physical qubits, while the faces or the plaquettes of the lattice denoted by red squares define stabilizer operators of the colour code. The resultant code has a minimum distance of $d = 3$ and hence becomes capable of correcting a single qubit error. This specific configuration bears a resemblance to the $C[7, 1, 3]$ Steane's 7 qubit code.

Again, similar to the classical TECCs, the construction of QTECCs is indeed not limited to the square lattice structure. Let us now elaborate on another construction inspired by the construction proposed in [25] using the triangular lattice based on the classic example of Fig. 5. In the proposal of [25], this specific code construction is often referred to as the (triangular) colour code, since the underlying triangular lattice is composed by the tri-coloured hexagonal tiles. However, constructing the stabilizer operators of colour codes slightly differs from that of the surface codes. The colour codes use the lattice plaquettes to define both the **Z** and **X** stabilizer operators. Consequently, the resultant colour codes belong to the family of dual-containing CSS codes, which is in contrast to the surface codes that belong to the class of non-dual-containing CSS codes. For colour codes, defining both the **Z** and **X** stabilizer operators using the same plaquette always guarantees satisfying the symplectic criterion of Eq. (50). However, for surface codes, we cannot always satisfy the symplectic criterion by using the same procedure. Therefore, the dual of the lattice is used for defining half of the stabilizer operators of the surface codes in order to satisfy the symplectic criterion.[7]

Let us consider Fig. 13 for constructing the stabilizer operators of distance-3 colour codes, which are only capable of correcting a single qubit error. The plaquette denoted by red square at index 3 is used to define both the **Z** and **X** stabilizer operators. Thus, the resultant **X** stabilizer operator is $A_3 = \mathbf{X}_2\mathbf{X}_4\mathbf{X}_6\mathbf{X}_7$ and the resultant of **Z** stabilizer operator is $B_3 = \mathbf{Z}_2\mathbf{Z}_4\mathbf{Z}_6\mathbf{Z}_7$. The stabilizer operators for the colour code having the minimum distance 3 in Fig. 13 are listed

in Table 9. We can observe that the colour code of Fig. 13 exhibits a strong resemblance to Steane's 7-qubit code.

TABLE 9. The stabilizer operators (S_i) of the colour code seen in Fig. 13. The code has a minimum distance of 3 ($d = 3$), which means that it is only capable of correcting a single qubit error.

S_i	A_p	S_i	B_p
S_1	$\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{X}_4$	S_4	$\mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_3\mathbf{Z}_4$
S_2	$\mathbf{X}_3\mathbf{X}_4\mathbf{X}_5\mathbf{X}_6$	S_5	$\mathbf{Z}_3\mathbf{Z}_4\mathbf{Z}_5\mathbf{Z}_6$
S_3	$\mathbf{X}_2\mathbf{X}_4\mathbf{X}_6\mathbf{X}_7$	S_6	$\mathbf{Z}_2\mathbf{Z}_4\mathbf{Z}_6\mathbf{Z}_7$

To draw on the parallelism between classical TECCs and QTECCs, let us consider the stabilizer operators of the colour code having a minimum distance of $d = 3$, as seen in Table 9. Since the distance-3 colour code belongs to the family of quantum CSS codes, the PCM **H** obtained by using Eq. (48) and (51) is encapsulted as follows:

A CSS stabilizer code $\mathcal{C}[n, k, d]$ having $(n - k)$ stabilizer operators can be portrayed as a classical code having a PCM **H** containing $(n-k) \times 2n$ elements. Therefore, the coding rate of the classical dual of a quantum CSS code can be expressed as follows [11]:

$$
\begin{aligned}
r_C &= \frac{2n - (n - k)}{2n}, \\
&= \frac{n + k}{2n}, \\
&= \frac{1}{2}\left(1 + \frac{k}{n}\right), \\
&= \frac{1}{2}\left(1 + r_Q\right), \quad (56)
\end{aligned}
$$

where r_C is the coding rate of the classical dual of the stabilizer code $C[n, k, d]$ exhibiting a quantum coding rate of r_Q. The relationship between the classical and quantum coding rate in Eq. (56) can be rewritten as

$$
r_Q = 2r_C - 1. \quad (57)
$$

For instance, let us consider the distance-3 colour codes $\mathcal{C}[n, k, d] = \mathcal{C}[7, 1, 3]$, as exemplified in Fig. 13, and its classsical dual $\mathcal{C}(n, k, d) = \mathcal{C}(7, 4, 3),$[8] as seen in Fig. 5. Explicitly, we have the classical coding rate of $r_C = 4/7$ for the $\mathcal{C}(7, 4, 3)$ code. By substituting $r_C = 4/7$ into Eq. (57), we obtain the quantum coding rate for its quantum counterpart as $r_Q = 1/7$, which is the quantum coding rate of distance-3 colour code $\mathcal{C}[7, 1, 3]$. The same goes for the classical square codes and their quantum counterpart, namely for the surface codes. Let us consider the distance-5 classical square code, which is labeled by S2 in Fig. 8 and its quantum pair, which is labeled by S2 in Fig. 14. We can readily determine the quantum coding rate of the surface code S2 $\mathcal{C}[41, 1, 5]$, which is $r_Q = 1/41$. Therefore, by substituting $r_Q = 1/41$ into Eq. (56), we arrive at the coding rate of its classical dual given by $r_C = 21/41$, which is indeed the coding rate of the classical square code S2 $\mathcal{C}(41, 21, 5)$.

---

[7]The dual of a lattice or a graph G is the graph that has a vertex for each plaquette of the graph.

[8]To avoid ambiguity, we use the notation $\mathcal{C}(n, k, d)$ for classical error correction codes and $\mathcal{C}[n, k, d]$ for quantum stabilizer codes.
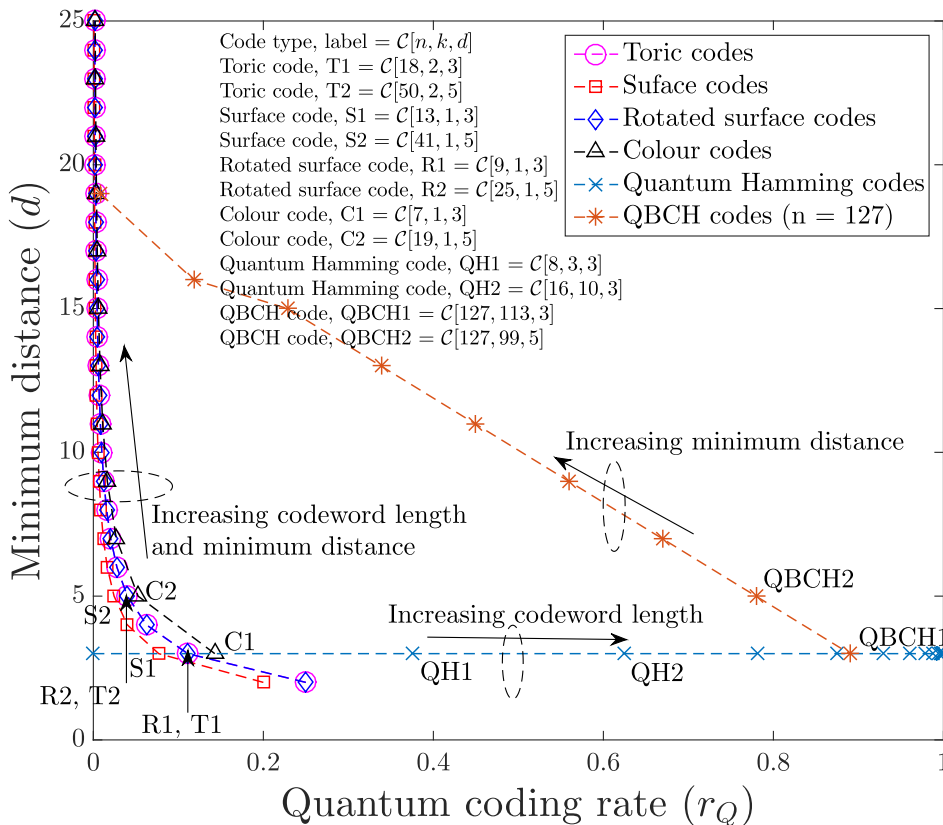
FIGURE 14. The minimum distance (*d*) versus quantum coding rate (r_Q) of QTECCs based on the code parameter given in Table 10. For QTECCs, the quantum coding rate tends to zero as we increase the minimum distance. We also include quantum Hamming codes and the QBCH codes having *n* = 127 physical qubits for the sake of comparing the QTECCs with the non-topological QSCs. The parameters of quantum Hamming codes and QBCH codes are listed in Table 11 and 12, respectively.

TABLE 10. The code parameters for various QTECCs based on the minimum distance *d* of the code.

Codes type	Dimension	Number of physical qubits	Number of stabilizers	Number of logical qubits
Colour	d^*	$\frac{1}{4}\left(3d^2 + 1\right)$	$\frac{1}{4}\left(3d^2 - 3\right)$	1
Rotated-surface	$d \times d$	d^2	$d^2 - 1$	1
Surface	$d \times d$	$2d^2 - 2d + 1$	$2d^2 - 2d$	1
Toric	$d \times d$	$2d^2$	$2d^2 - 2$	2

* for triangular colour codes the dimension is defined by the side length of the equilateral triangle

Similar to their classical counterparts, the code parameters of QTECCs, such as the number of logical qubits *k*, the number of physical qubits *n*, the minimum distance of the code *d*, as well as the quantum coding rate r_Q, depend on the size of the lattices. Following the same line of investigation as for the classical TECCs, we derive the complete formulation for the number of logical qubits *k* and the number of physical qubits *n* as a function of the minimum distance of the codes,

which is given in Table 10. We plot the minimum distance (*d*) versus quantum coding rate (r_Q) of QTECCs in Fig. 14 for colour codes [25], for rotated surface codes [31], for surface codes [24] and for toric codes [22]. We also include the non-topological QSCs, namely the QBCH codes [7] having *n* = 127 physical qubits and the quantum Hamming codes, which constitute the quantum analogue of Hamming bound-achieving code constructions [42]. Similarly to the

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}. \tag{55}$$

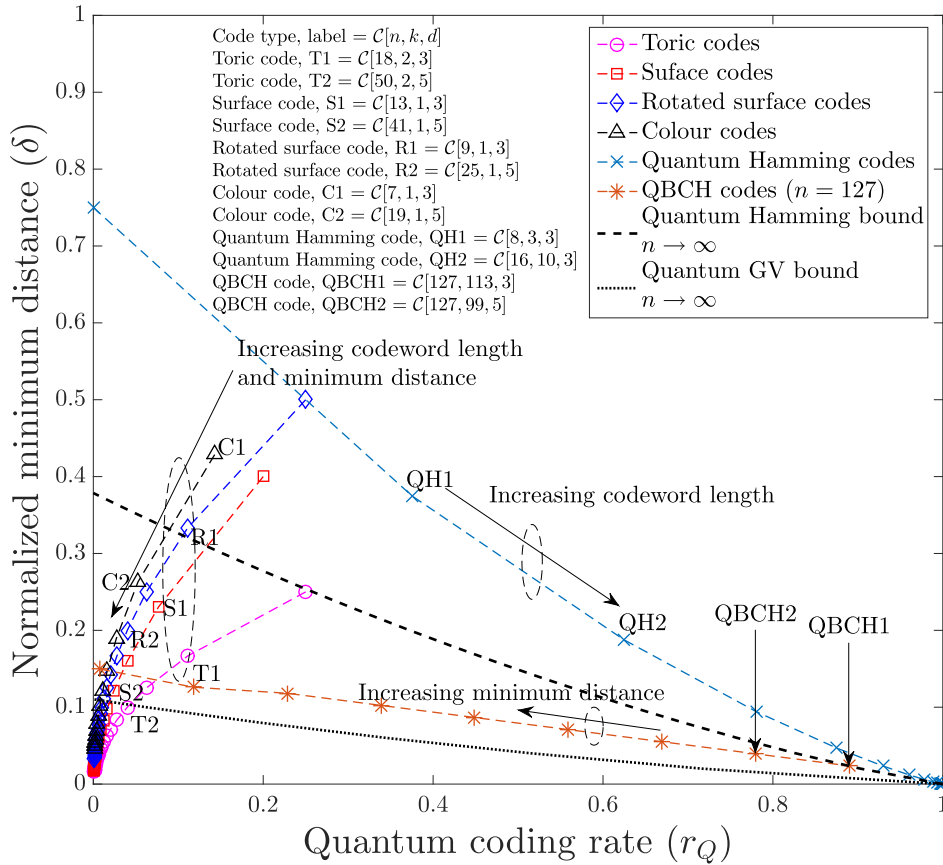Code type, label = $\mathcal{C}[n, k, d]$
Toric code, T1 = $\mathcal{C}[18, 2, 3]$
Toric code, T2 = $\mathcal{C}[50, 2, 5]$
Surface code, S1 = $\mathcal{C}[13, 1, 3]$
Surface code, S2 = $\mathcal{C}[41, 1, 5]$
Rotated surface code, R1 = $\mathcal{C}[9, 1, 3]$
Rotated surface code, R2 = $\mathcal{C}[25, 1, 5]$
Colour code, C1 = $\mathcal{C}[7, 1, 3]$
Colour code, C2 = $\mathcal{C}[19, 1, 5]$
Quantum Hamming code, QH1 = $\mathcal{C}[8, 3, 3]$
Quantum Hamming code, QH2 = $\mathcal{C}[16, 10, 3]$
QBCH code, QBCH1 = $\mathcal{C}[127, 113, 3]$
QBCH code, QBCH2 = $\mathcal{C}[127, 99, 5]$

FIGURE 15. The normalized minimum distance versus quantum coding rate of QTECCs based on parameter given in Table 10. For QTECCs, the normalized minimum distance and quantum coding rate tend to zero as we increase the minimum distance. We also include the QBCH codes having the physical qubits of $n = 127$, quantum Hamming codes, quantum Hamming bound and also quantum GV bound for CSS codes for the sake of comparing the QTECCs with the non-topological QSCs.

TABLE 11. Code parameters of quantum Hamming codes having a single error correction capability, which is used in Fig. 14 and 15. The quantum coding rate r_Q and normalized minimum distance δ are calculated using Eq. (1) and (17), respectively.

n	k	d	n	k	d
8	3	3	256	246	3
16	10	3	512	501	3
32	25	3	1024	1012	3
64	56	3	2048	2035	3
128	119	3	. . .	. . .	. . .

TABLE 12. Code parameters of QBCH codes having codeword length of $n = 127$, which is used in Fig. 14 and 15. The quantum coding rate r_Q and normalized minimum distance δ are calculated using Eq. (1) and (17), respectively.

n	k	d	n	k	d
127	1	19	127	71	9
127	15	16	127	85	7
127	29	15	127	99	5
127	43	13	127	113	3
127	57	11			

classical domain, the behaviour of both the QBCH codes and the quantum Hamming codes is as expected, exhibiting the behaviour inherited from their classical analogues. However, it is interesting to observe that the quantum coding rate of QTECCs tends to zero for long codewords. Nevertheless, this phenomenon is expected, if we consider the classical to quantum isomorphism in the context of the coding rate given in Eq. (56) and (57). For the classical TECCs, the coding rate r_C approaches the value of $r_C = 1/2$ for long codewords. Hence, by substituting $r_C = 1/2$ into Eq. (57), we arrive at $r_Q = 0$, which is the phenomenon we observe in Fig. 14.

Next, we plot the normalized minimum distance (δ) versus the quantum coding rate (r_Q) in Fig. 15. Once again, for the sake of comparison, we also include the quantum Hamming bound [43] and the quantum GV bound derived for CSS codes [44] in addition to the QBCH codes and the quantum Hamming codes. The quantum Hamming bound is defined by [43]

$$\frac{k}{n} \leq 1 - \left(\frac{d}{2n}\right) \log_2 3 - H\left(\frac{d}{2n}\right), \quad (58)$$

while the quantum GV bound for CSS codes is given by [44]

$$\frac{k}{n} \geq 1 - 2H\left(\frac{d}{n}\right). \quad (59)$$

Both the quantum Hamming bound and the quantum GV bound of Fig. 15 serve the same purpose as the classical Hamming bound and the GV bound seen in Fig. 9. Explicitly, they portray the upper bound and the lower bound of normalized minimum distance versus quantum coding rate trade-off. Once again, the puzzling behaviour of classical TECCs resurfaces for the QTECCs, as observed in Fig. 15. Since all the QBCH codes, quantum Hamming codes and QTECCs inherit the properties of their classical counterparts, their behaviour is reminiscent of that of their classical counterparts. As for the QTECCs, the definitive interpretation of this unusual behaviour is left for future exploration in our research. Nonetheless, for a relatively long codeword, the QTECCs are reminiscent of QLDPC codes. Observe from Fig. 15 that both the normalized minimum distance and the quantum coding rate of QTECCs tend to zero upon increasing the minimum distance by increasing the codeword length. Therefore, the QTECCs are deemed to be more favourable for short to medium codeword lengths.

V. PERFORMANCE OF QUANTUM TOPOLOGICAL ERROR CORRECTION CODES

In this treatise, we consider the performance of QTECCs under the popular quantum depolarizing channel. Explicitly, the quantum depolarizing channel is characterized by the quantum depolarizing probability p inflicting an error pattern constituted by the Pauli operators $\mathcal{P} \in \mathcal{G}_n$ upon the state of physical qubits, where each qubit may independently experience a bit-flip error ($\mathbf{X}$), a phase-flip error ($\mathbf{Z}$), or both bit-flip and phase-flip error ($\mathbf{Y}$) with an equal probability of $p/3$. In order to get a more precise insight into the performance trends of QTECCs, we have to distinguish how the different error patterns affect the state representing the physical qubits. Explicitly, the n-tupple Pauli error pattern may be classified as follows, which will be exemplified in Fig. 16 and 17 after their definitions:

1) **Harmful detected error pattern**. This specific type of error pattern has a similarity to the conventional bit error in the classical domain. The error pattern of $\mathcal{P}$ anti-commutes with the stabilizer operators $S_i \in \mathcal{S}$, hence triggers non-trivial syndrome values.
2) **Harmful undetected error pattern**. The error pattern commutes with all of the stabilizer operators, except that it does not belong to the stabilizer group $\mathcal{S}$. In the classical domain, this is similar to the error pattern that returns the all-zero syndrome. The error pattern is harmful, since it does not trigger a non-trivial syndrome value, yet it corrupts the legitimate state of the physical qubits.
3) **Harmless undetected error pattern**. This particular error pattern does not have any classical analogue. The error pattern is harmless, because it belongs to the stabilizer group $\mathcal{S}$. This is also referred to as a degenerate error pattern. Consequently, the error patttern does not alter the legitimate state of the physical qubits.

By considering the degeneracy, the actual performances of QTECCs are potentially improved.

In order to illustrate both the harmless and harmful undetected error patterns, we refer to Fig. 16 and 17. First, we commence with the harmless undetected error pattern, which is illustrated in Fig. 16. In this example, we consider a surface code having a minimum distance of 5, which implies that it is only capable of correcting two qubit errors. Following the stabilizer formulation of QTECCs discussed in Section IV, the physical qubits are arranged along the edges of the square lattice, while the $\mathbf{X}$ stabilizer operators are located in the vertices. Therefore, the $\mathbf{X}$ stabilizer operators on the vertices are used for indicating the $\mathbf{Z}$ errors, which will trigger eigenvalues of -1 if they anticommute with the $\mathbf{X}$ stabilizer operators. Let us assume that the quantum depolarizing channel inflicts three $\mathbf{Z}$ errors on the physical qubits, which are denoted by the filled black circles in Fig 16, while the hollow black circles represent the error free physical qubits. All of the error patterns given in Fig 16 (a), (b), and (c) trigger the eigenvalues of -1 for the stabilizer operators denoted by filled red squares, while the rest of the stabilizer operators are represented by hollow red squares, which return eigenvalues of $+1$. Since the decoder relies on hard-decision ML decoding, all of the error patterns given in Fig. 16 (a), (b), and (c) have the same probability of occurence. Let us assume that the decoder always decides to apply the error recovery pattern of Fig. 16 (a) for the specified values of stabilizer measurement. When the actual error pattern is the one given in Fig. 16 (a), the states of the physical qubits are fully recovered. By contrast, if the actual error pattern is the one seen in Fig. 16 (b), but it is corrected using the error recovery operator of Fig. 16 (a), we arrive at the accumulated error pattern shown in Fig. 16 (d). Lastly, when the actual error pattern is the one given by Fig. 16 (c), but we attempt to correct it using the error recovery of Fig. 16 (a), we obtain the error pattern seen Fig. 16 (e). However, if we observe closely the error pattern illustrated in Fig. 16 (d), it is reminiscent of a plaquette $\mathbf{Z}$ stabilizer operator denoted by the filled blue triangle. Therefore, based on the definition of stabilizer operators, the error pattern given in Fig. 16 (d) does not alter the legitimate state of physical qubits. Similarly, the error pattern of Fig. 16 (e) resembles the product of two adjacent plaquette stabilizer operators. Since the product between a pair of stabilizer operators return another valid stabilizer operator, the error pattern given in Fig. 16 (e) belongs to the stabilizer group $\mathcal{S}$. Once again, by definition, the error pattern given in Fig. 16 (e) does not corrupt the legitimate state of physical qubits. This is an example of *harmless undetectable error patterns*.

To elaborate a little further, a harmless undetected error can be directly generated by the quantum decoherence, where the Pauli operator $\mathcal{P} \in \mathcal{G}_n$ imposed by the quantum depolarizing channel is identical to the stabilizer operator S_i. Another possibility is that it is generated by the associated error recovery procedure, when trying to recover an ambiguous error pattern, where there are more than one possible error patterns associated with a specific syndrome value, as illustrated in
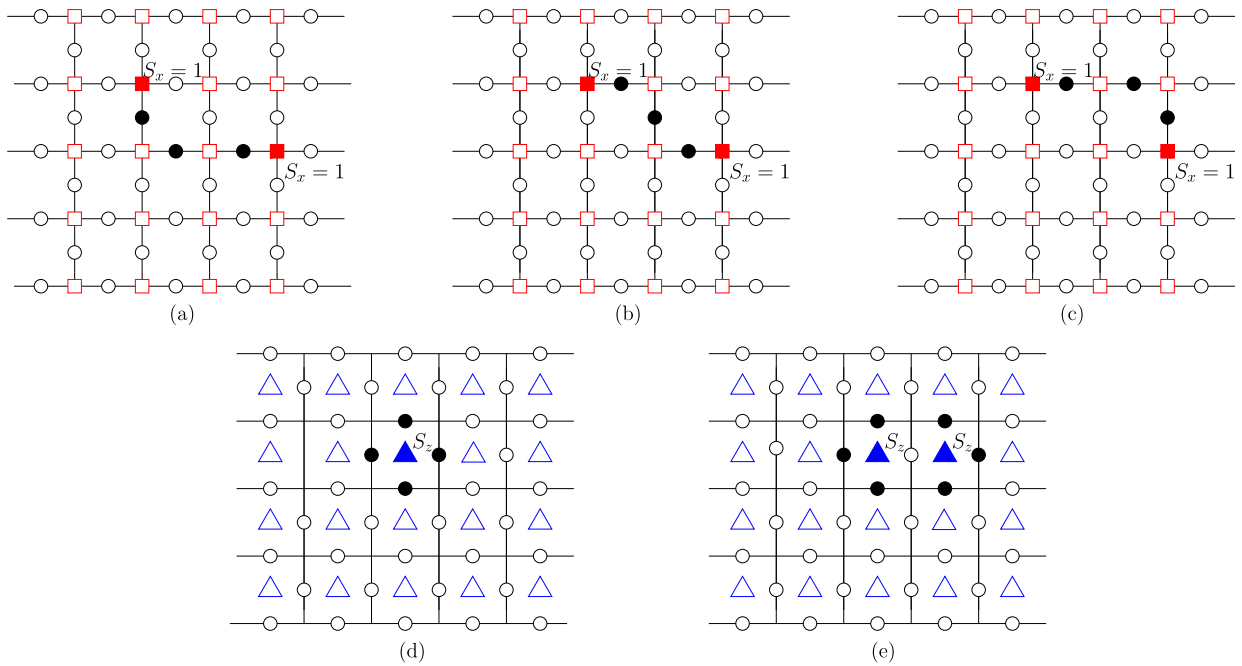
FIGURE 16. Illustration of how the error recovery operator $\mathcal{R}$ creates the degenerate error patterns and how the degeneracy nature of QECCs may improve the performance of QTECCs. All of error patterns given in (a), (b) and (c) represent error patterns generating an identical syndrome value. Without lose of generality, let us assume that based on the generated syndrome value, the decoder always decides to perform error recovery operator $\mathcal{R}$ of (a) on the corrupted state of physical qubits. If the actual error pattern is (a), the corrupted state of physical qubits will be fully recovered. By contrast, figure (d) shows the resultant error pattern if the actual error pattern is (b), but it is corrected using the error pattern given in (a). Moreover, figure (e) represents the resultant error pattern if the actual error pattern is (c) and it is corrected using the error recovery pattern of (a). As the result, the error pattern (d) represents a stabilizer operator of a plaquette, while the error pattern (e) resembles the product of two adjacent stabilizer operators. Both error patterns of (d) and (e) constitute the *harmless undetecteable error patterns*, since they belong to the stabilizer group $\mathcal{S}$. Therefore, the state of physical qubits is not altered after the recovery operator $\mathcal{R}$ of (a) is applied to all error patterns of (a), (b) and (c). In classical set up, both error patterns (d) and (e) are considered as error events. However, in quantum domain, both error patterns (d) and (e) are considered as error-free cases. This specific error-type has no similarity in quantum domain and hence potentially improves the performance of QTECCs.

Fig. 16. The degeneracy property, which is associated with the harmless undetectable error patterns, does not have a classical analogue, because in the classical setup, the resultant error patterns illustrated in Fig. 16 (d) and (e) will always be considered as an error. Ultimately, considering the degeneracy potentially improves the performance of QECCs.

Let us consider a range of different scenario for illustrating the presence of harmful undetected error patterns, which is portrayed in Fig. 17. Similar to the previous example of Fig. 16, three **Z** errors are imposed on the state of logical qubits by the quantum depolarizing channel. The error patterns given in Fig. 17 (a) and (b) trigger the eigenvalues of -1 for the stabilizer operators denoted by filled red squares in Fig. 17, while the rest of the stabilizer operators represented by hollow red squares return eigenvalues of $+1$. Given the associated syndrome value, the decoder always decides to apply the error recovery operator of Fig. 17 (a). In the specific scenario, where the actual error pattern is the one given by Fig. 17 (b), the resultant error pattern is given in Fig. 17 (c). We can observe that the resultant error pattern of Fig. 17 (c) commutes with all of the stabilizer operators in Fig. 17. However, this specific error pattern does not belong to the stabilizer operator $\mathcal{S}$, since we cannot represent a chain of errors

by the product of stabilizer operators. Consequently, this undetectable error pattern inevitably corrupts the legitimate state representing the physical qubits. This is an example of the *harmful undetectable error patterns*. This error pattern is similar to that of its counterpart in the classical domain, where the error pattern returns the all-zero syndrome.

Therefore, based on these conditions, by modifying the probabilty of correct decoding in the classical domain [45], we can readily formulate the worst-case upper-bound QBER performance of QTECCs as

$$
\text{QBER}_{\text{upper}}(n, d, p) = 1 - \sum_{i=0}^{t=\lfloor \frac{d-1}{2} \rfloor} \binom{n}{i} p^i (1-p)^{n-i}
$$
$$
- \sum_{i=1, \forall S_i \in \mathcal{S}}^{|\mathcal{S}|} p^{w(S_i)} (1-p)^{n-w(S_i)},
$$
$$(60)$$

where $w(S_i)$ is the weight of the stabilizer operator S_i, which is defined by the number of non-identity Pauli operators within the stabilizer operators. The second term of Eq. (60) represents all the correctable error patterns of QTECCs, while the last term of Eq. (60) represents the degenerate error patterns
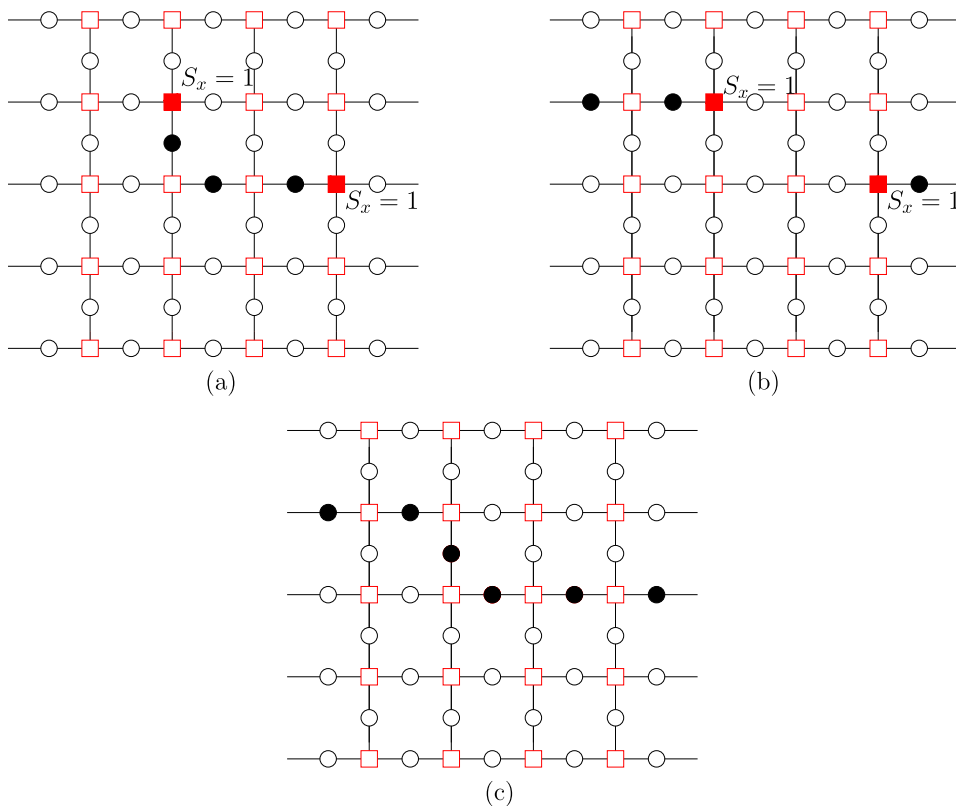
FIGURE 17. Illustration of the harmful undetectable error pattern in quantum domain. The actual error pattern inflicts the state of physical qubits is given in (b), while the decoder always decides to perform a recovery operator given in (a). Instead of recovering the legitimate state of the physical qubits, the specified error recovery procedure generates a chain of error that commutes with all of the stabilizer operators, as shown in (c). In quantum domain, it constitutes the *harmful undetectable error patterns*. In classical domain, it resembles the error pattern that generates all-zero syndrome values.

that belong to the stabilizer operators. For example, let us revisit the construction of the surface codes of Fig. 12. There are 12 stabilizer generators for a distance-3 surface code, as seen in Table 8. Hence, we can potentially generate in total 2^{12} unique stabilizer operators, since the product of the stabilizer operators returns another valid stabilizer operator. However, in order to further simplify the expression given in Eq. (60), we only consider the error patterns resembling the specified stabilizer operators given in Table 8, since they exhibit a lower weight of non-identity Pauli matrices and hence have a higher probability of occurance. Therefore, for surface codes, the last term of Eq. (60) can be approximated as $(2d^2 - 2d)p^4(1-p)^{n-4}$. The term $(2d^2 - 2d)$ represents the number of stabilizer operators, which is given in Table 10, and we assume that all the weight of the stabilizer operators $w(S_i)$ are equal to 4.

A. QBER VERSUS DEPOLARIZING PROBABILITY

In order to characterize the performance of QTECCs by simulations, we exploit the fact that the QTECCs belong to the family of quantum CSS codes, which handle the bit-flips ($\mathbf{X}$) and phase-flips ($\mathbf{Z}$) separately. Hence, we invoke

two independent binary symmetric channels (BSC), one for the $\mathbf{X}$ channel and one for the $\mathbf{Z}$ channel, where each channel is characterized by the flip probability of $2p/3$, where p is the associated depolarizing probability of the quantum depolarizing channel [13], [16]. The decoder utilizes hard-decision ML decoding relying on a simple LUT decoder, as exemplified in Section III. However, this classical-domain simulation only represents the performance of QTECCs without considering the degenerate error patterns. To elaborate a little further, we generate all-zero information bits at the input and send them through the two independent BSC channels. Therefore, we always consider all of non all-zero decoded bits at the decoder output as an error. However, in order to additionally consider several cases of degenerate error patterns, which is exemplified in Fig. 16, we performed an additional evaluation step. We evaluate the non all-zero corrected received words and check for the degenerate error patterns. If it satisfies the degenerate error pattern criterion that we have defined above, we conclude that this is an error free case. However, we are not capable of providing a complete list of all possible degenerate error patterns and in this treatise we only consider the error pattern resembling the stabilizer generators of S_i, which is exemplified in Table 8 and 9 for surface codes and
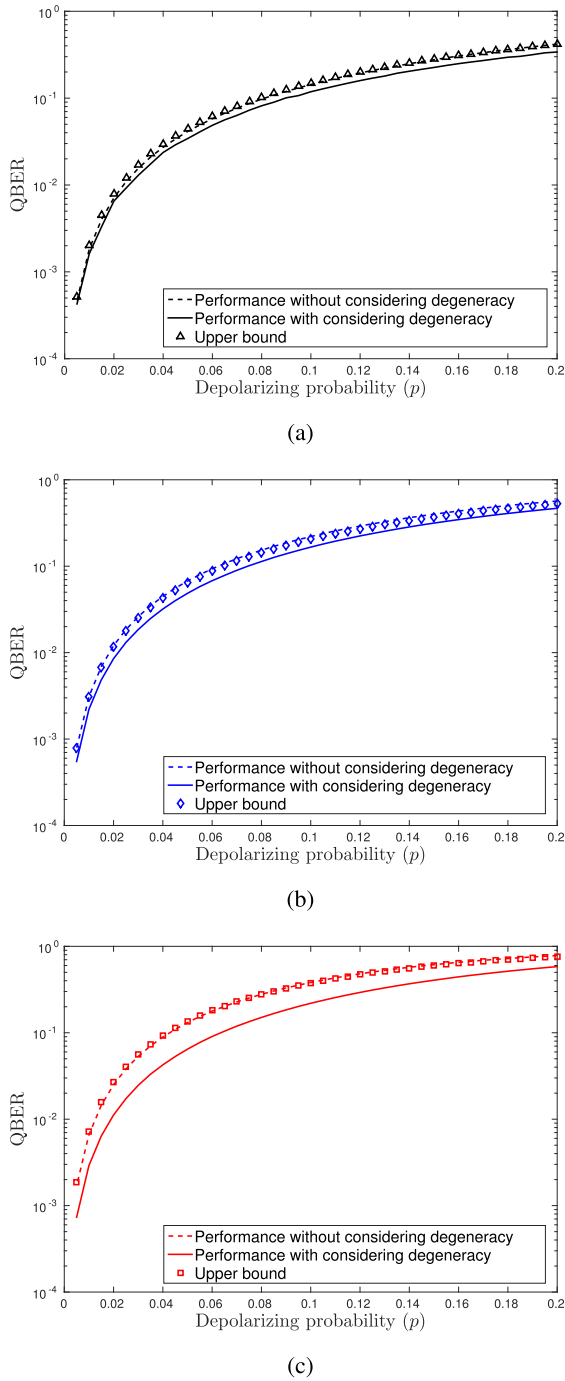
FIGURE 18. QBER performance of the distance-3 surface code, rotated-surface code and colour code over the quantum depolarizing channel, which is capable of correcting a single qubit error. The code parameters are given in Table 13. For this scenario, the decoder using hard-input ML decoding approach for predicting the error pattern. (a) Colour code. (b) Rotated surface code. (c) Surface code.

TABLE 13. Code parameters for distance-3 colour code, rotated surface code and surface code.

Code type	n	k	d	r_Q
Colour code	7	1	3	1/7
Rotated surface code	9	1	3	1/9
Surface code	13	1	3	1/13

be clearly observed that the upper bounds match with the QTECCs performance without considering the degenerate error patterns.

As we mentioned earlier, there are two sources of the degenerate error pattern at the output of the decoder. First, the degenerate error patterns that imposed ubiquitous directly by the quantum channel, where the error exhibits an identical pattern to the stabilizer operator S_i. Second, the degenerate error pattern generated by the recovery operator $\mathcal{R}$, when it tries to recover the legitimate physical qubits, as illustrated in Fig. 16. The second case is more dominant than the first one. The reason can be explained as follows. Let us assume the **Z** stabilizer operators of distance-3 surface code given in Table 8. There are six **Z** stabilizer operators correspond to the $2^6 = 64$ possible syndrome vector, including the error-free scenario. Remember that the distance-3 surface code can only flawlessly correct a single error qubit within the block of 13 physical qubits, where each of the single qubit error pattern is associated with only one syndrome vector. In other words, amongst all of 64 possible syndrome vectors, there are only 13 syndrome vectors used to uniquely distinguish the correctable error patterns, while the rest of the syndrome vectors are associated with the error pattern ambiguity, as exemplified in Fig. 16 and 17. Due to this reason, the QTECCs are considered as the highly degenerate QSCs. Hence, the upper bound of the QBER performance matches the simulation-based performance recorded without considering the degeneracy, since it considers only the first source of the degeneracy, where only a portion of all valid stabilizer operators $S_i \in \mathcal{S}$ in Eq. (60) is included in calculation. However, by accommodating both of the degeneracy cases, the QBER performance of QTECCs is indeed improved, as displayed in Fig. 18.

Increasing the minimum distance of a given QSC construction, which directly improves its per-codeword error correction capability (t), is achieved by increasing the number of physical qubits (n) or by decreasing the quantum coding rate. Specifically for QTECCs, increasing the minimum distance means simultaneously increasing the number of physical qubits (n) and decreasing the quantum coding rate (r_Q). Naturally, the goal of increasing the minimum distance of the QSCs is to achieve a better QBER performance. However, the improvement of QBER the performance can only be observed below a certain value of depolarizing probability (p), which may be referred to as the threshold probability (p_{th}). Using the upper bound QBER performance of Eq. (60), we plot the QBER curves for colour, rotated-surface, surface and

triangular codes, respectively. The QBER performance of distance-3 QTECCs versus the quantum depolarizing probability is portrayed in Fig. 18, where the code parameters are given in Table 13. We also include the upper bound of the QTECCs performance of Eq. (60) in Fig. 18. It can
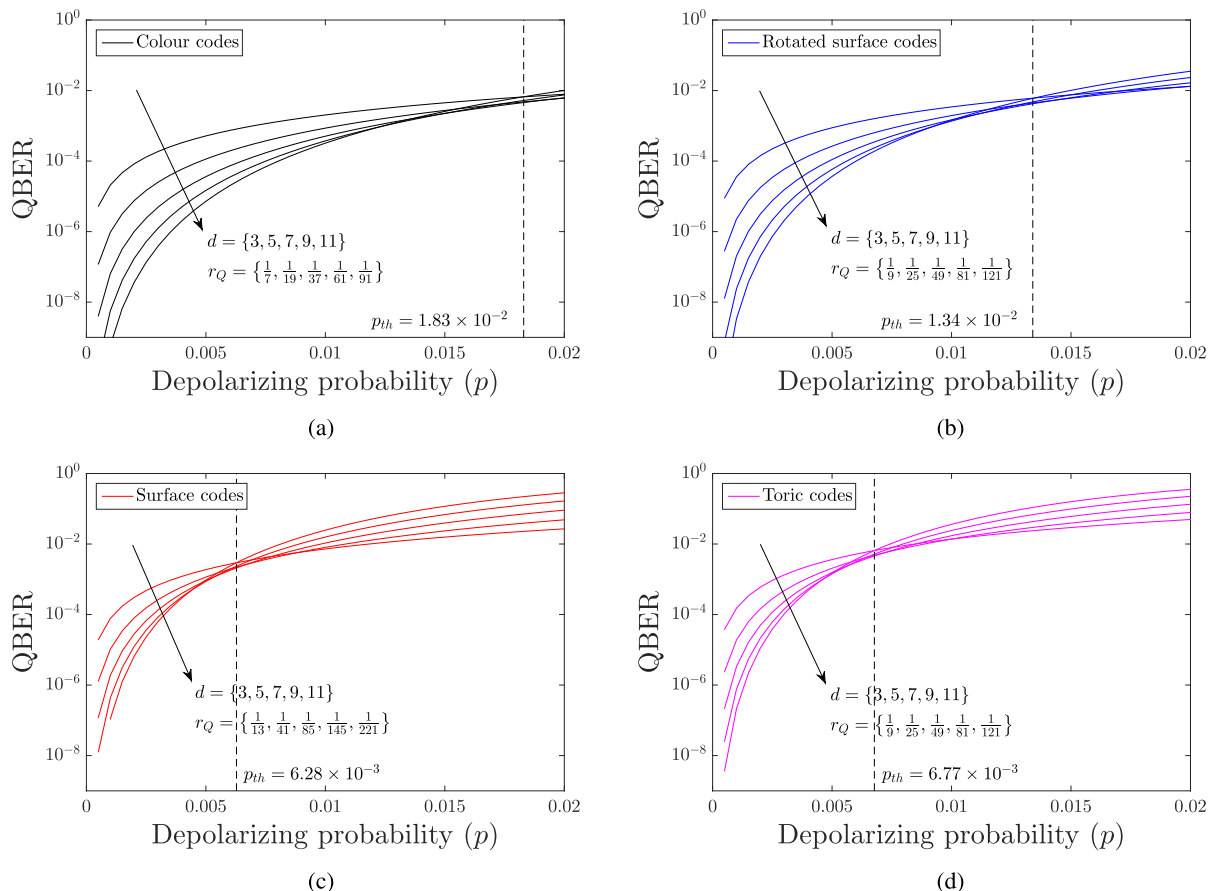
FIGURE 19. Upper bound QBER performance of QTECCs for the minimum distance of $d = \{3, 5, 7, 9, 11\}$ based on Eq. (60) and the code parameters given in Table 10. The crossover amongst the QBER curves represents the threshold probability (p_{th}), which are portrayed in dashed line. (a) Upper bound QBER performance of colour codes. (b) Upper bound QBER performance of rotated-surface codes. (c) Upper bound QBER performance of surface codes. (d) Upper bound QBER performance of toric codes.

toric codes in Fig. 19. For each of the QTECC constructions, we portray the upper bound QBER performance for the minimum distances of $d = \{3, 5, 7, 9, 11\}$. The threshold probability of each code is denoted by the crossover QBER curves, which we portray in dashed line. The threshold probability of colour, rotated-surface, surface and toric codes are 1.83×10^{-2}, 1.34×10^{-2}, 6.28×10^{-3} and 6.77×10^{-3}, respectively.

B. QBER VERSUS DISTANCE FROM HASHING BOUND

Presenting the performance of QTECCs over quantum depolarizing channel by portraying the QBER curves versus the depolarizing probability (p) does not take the quantum coding rate (r_Q) into consideration. As we mentioned earlier, we can simply decrease the quantum coding rate further and further in order to increase the error correction capability of the QTECCs. Nonetheless, for the sake of depicting a fair comparison upon reducing the quantum coding rate, we have to scrutinize how much performance improvement we obtain upon decreasing the quantum coding rate. Therefore, in order to demonstrate how much performance improvement we attain compared to the how much we decrease the quan-

tum coding rate, we normalize the QBER performance by incorporating the quantum hashing bound. More explicitly, the quantum hashing bound can be expressed as follows [46]:

$$C_Q(p) = 1 - H(p) - p \cdot \log_2(3), \qquad (61)$$

where $H(p)$ is the binary entropy of p. More specifically, the quantum hashing bound of Eq. (61) dictates that a random quantum code $\mathcal{C}$ having a sufficiently long codeword and a quantum coding rate $r_Q \leq C_Q(p)$ may yield an infinitesimally low QBER for a given depolarizing probability p. Alternatively, we can refer to $C_Q(p)$ as the hashing limit for the quantum coding rate r_Q associated with a given depolarizing probability p. In terms of its classical dual pair, the value of C_Q is similar to the capacity limit. Similarly, for a given coding rate r_Q, we can find a value of p^* satisfying $r_Q = C_Q(p^*)$, where p^* denotes the maximum value of depolarizing probability p so that a quantum code $\mathcal{C}$ having quantum coding rate of r_Q can operate at an infinitesimally low QBER. The value of p^* may be referred to as the hashing limit for depolarizing probability of p associated with a given quantum coding rate r_Q. In classical domain the value of p^* is similar to the noise limit. Therefore, in general, the aim is that of finding
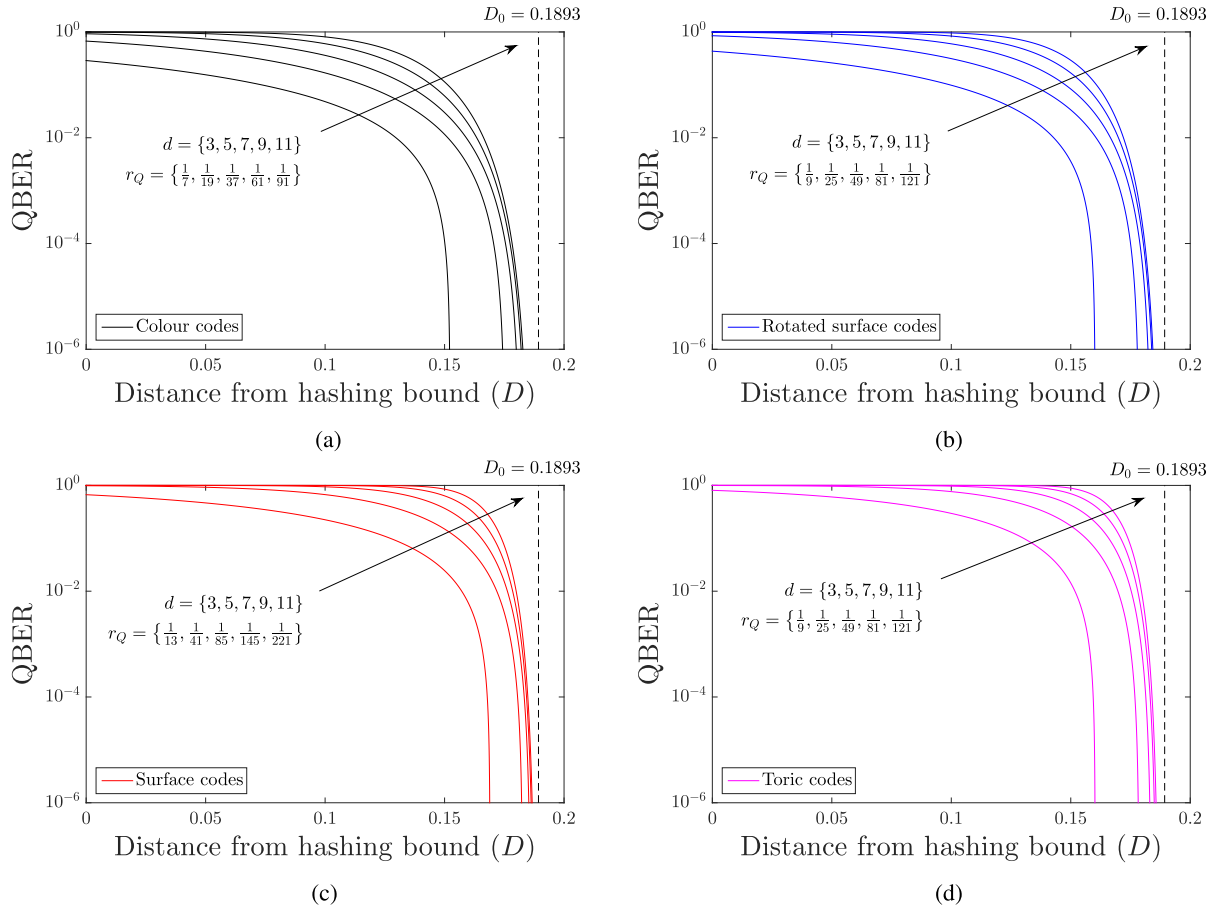
FIGURE 20. Upper bound performance of QTECCs in term of the QBER versus the distance *D* from the hashing bound. The code parameters are given in Table 10. The dashed lines portray the ultimate distance to the quantum hashing bound of $D_0 = 0.1893$. (a) Upper bound performance of colour codes. (b) Upper bound performance of rotated surface codes. (c) Upper bound performance of surface codes. (d) Upper bound performance of toric codes.

a QSC that is capable of performing as close as possible to the quantum hashing bound.

For example, let us consider the distance-3 and distance-5 rotated surface codes having quantum coding rate of $r_Q = 1/9$ and $r_Q = 1/25$, respectively. By substituting $C_Q = 1/9$ and $C_Q = 1/25$ into the Eq. (61), we obtain the noise limit of $p^* = 0.160$ and $p^* = 0.179$, respectively. It is clearly seen that the noise limit is higher for the quantum code exhibiting a lower quantum coding rate. To incorporate the quantum hashing bound into the QBER performances of QTECC, we define the distance from hashing bound as follows:

$$D \triangleq p(r_Q) - p, \qquad (62)$$

where $p(r_Q)$ is the hashing limit for depolarizing probability of p associated with a given quantum coding rate r_Q. In other words, by changing the horizontal axis from the depolarizing probability p to the distance D from hashing bound, we shift all the QBER curves according to their hashing bounds, so that all the hashing bounds are at the reference point of $D = 0$.

Several pertinent questions arise from the quantum hashing bound formulation. Firstly, is there a noise limit, where no QSC constructions are capable of achieving a satisfactorily

low QBER? Indeed, the answer is yes. By substituting the $C_Q = 0$ into Eq. (61), which is the lowest possible value of achievable quantum coding rate, we arrive at the ultimate hashing bound of $p(0) \approx 0.1893$. Secondly, what is the farthest possible distance from the quantum hashing bound for any QSC construction. To answer this question, we have to consider the worst-case scenario, where a QSC exhibiting a near zero quantum coding rate ($r_Q \approx 0$) achieves an infinitesimally low QBER at near zero quantum depolarizing probability ($p \approx 0$). By substituting the value of $r_Q = 0$ and $p = 0$ into Eq. (62), we define the ultimate distance of hashing bound D_0 as

$$
\begin{aligned}
D_0 &= p(0) - p \\
&= 0.1893 - 0 \\
&= 0.1893. \qquad (63)
\end{aligned}
$$

Therefore, the desirable performance of any QSCs quantified in terms of the QBER versus distance from the quantum hashing bound is represented by the curves exhibiting a reasonably low QBER as close as possible to the reference point of $D = 0$. Naturally, this implies having a low QBER as
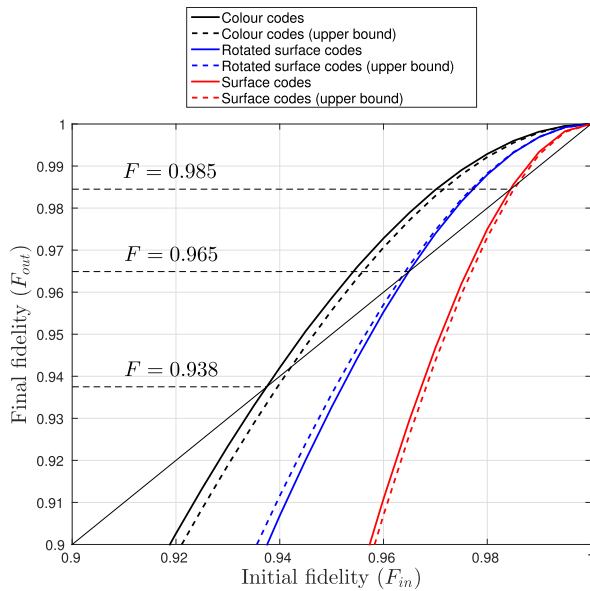
FIGURE 21. The performance of QTECCs having a minimum distance of 3 in terms of fidelity of Eq. (64). The colour code reaches the fidelity threshold earlier than the rotated-surface and surface code, since the colour code has the lowest number of physical qubits compared to the rotated surface code and the surface code. The code parameters are given in Table 13.

far as possible from the ultimate distance from the hashing bound of $D_0 = 0.1893$. In simpler terms, any QSCs can only operate at a reasonably low QBER within the hashing bound range of $0 \leq D \leq D_0$. Consequently, we should consider the reduction of the quantum coding rate r_Q as beneficial only if the associated QBER performance curve moves closer to the reference point of $D = 0$. Otherwise, it is more advisable to find a better code construction exhibiting an identical quantum coding rate, to increase the number of physical qubits, while maintaining the quantum coding rate, or to invoke more powerful decoding scheme, for example by utilizing a soft-decision-aided decoder. The QBER performance of QTECCs versus their distances from the quantum hashing bound are portrayed in Fig. 20. It can be observed that even though increasing the minimum distance of the QTECCs yields a performance improvement in terms of their QBER versus depolarizing probability p shown in Fig. 19, in terms of their distance from the hasing bound D, at low QBER, the curves are crowded in the vicinity of the ultimate hashing bound distance of D_0. Moreover, the results show an agreement with the quantum coding rate versus minimum distance evolution of QTECCs seen in Fig. 15. The improvement of the minimum distance, which is directly linked to the error correction capability, upon reducing the quantum coding rate is not fast enough to compensate the increasing number of physical qubits. Therefore, we believe that QTECCs are most suitable for short to moderate codeword lengths.

C. FIDELITY

From an implementational perspective, a quantum gate or quantum channel is often characterized by the so-called

fidelity, which represents the closeness of a pure quantum state of $|\overline{\psi}\rangle$ compared to the mixed states having the quantum density operator of ρ. More explicitly, since the quantum channel imposes the quantum decoherence on our legitimate quantum state representing the physical qubits $|\overline{\psi}\rangle$, there is a probability that decoder does not successfully recover the legitimate state. Therefore, the ensemble of all the possible predicted legitimate state of physical qubits $|\widehat{\psi}\rangle$ can be represented using the state of $|\psi_i\rangle$ having a probability of p_i. The fidelity can be formulated as follows [47]–[49]:

$$F = \langle \overline{\psi} | \rho | \overline{\psi} \rangle. \tag{64}$$

while ρ, which portrays the statistical characteristics of a the mixed states, is defined by

$$\rho = \sum_{i=1}^{N} p_i |\psi_i\rangle \langle \psi_i|, \tag{65}$$

where the $|\psi_i\rangle$ represents all of the possible state in the ensemble and p_i is the probability of having state $|\psi_i\rangle$ in the ensemble, which is subject to unity constraint of $\sum_{i=1}^{N} p_i = 1$.

In order to demonstrate the benefit of QTECCs in the context of quantum depolarizing channel, we compare the so-called initial fidelity F_{in} and final fidelity F_{out}. The initial fidelity is the fidelity of the pure quantum state of $|\psi\rangle$ over the quantum depolarizing channel $\mathcal{P}$ unprotected by any QSCs scheme. Therefore, the initial fidelity F_{in} can be expressed as follows:

$$F_{\text{in}} = 1 - p. \tag{66}$$

The final fidelity is that of the pure state of the desired output $|\psi'\rangle$ protected by the a QSC scheme after the recovery procedure $\mathcal{R}$ and inverse encoder $\mathcal{V}^\dagger$ of Fig. 11. Therefore, the final fidelity F_{out} of the quantum system can be readily formulated as

$$F_{\text{out}} = 1 - \text{QBER}. \tag{67}$$

The fidelity performance for the distance-3 QTECCs are depicted in Fig. 21. The black solid line represents the condition of $F_{\text{in}} = F_{\text{out}}$. The crossover point between the line of $F_{\text{in}} = F_{\text{out}}$ and fidelity performance curve of QTECCs is the break-even point, which we may referred to as the *threshold fidelity* F_{th}. The break-even point denotes the minimal initial fidelity required to ensure that we do acquire a fidelity improvement upon the applicaton of the QSC scheme, which is invoked for protecting the state of the physical qubits. The upper bound of threshold fidelity F_{th} for the different types of QTECCs having code parameters listed in Table 10 is depicted in Fig. 22. It can be observed that different code families having various minimum distances d result in different threshold fidelity F_{th}. For the QSCs utilizing hard-decision syndrome decoding, we derive the upper-bound approximation formula for determining the value of F_{th}. First, from
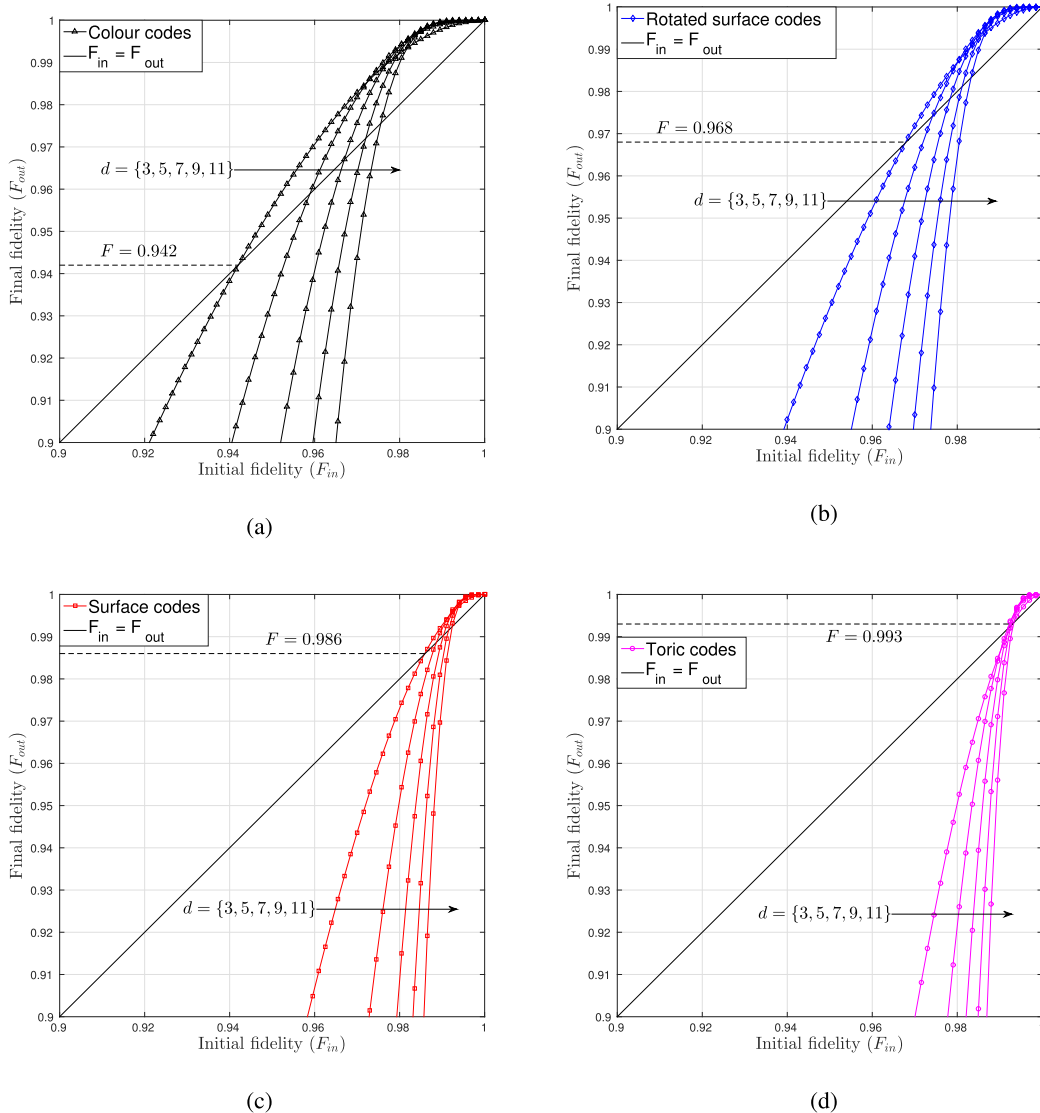
FIGURE 22. Upper bound fidelity performance of QTECCs. (a) Upper bound fidelity performance of colour codes.
(b) Upper bound fidelity performance of rotated surface codes. (c) Upper bound fidelity performance of surface codes.
(d) Upper bound fidelity performance of toric codes.

Eq. (60) and Eq. (67), we arrive at

$$
\begin{aligned}
F_{\text{out}} &= 1 - \text{QBER}_{\text{upper}} \\
&= 1 - \left(1 - \sum_{i=0}^{t=\lfloor \frac{d-1}{2} \rfloor} \binom{n}{i} p^i (1-p)^{n-i}\right) \\
&= 1 - \sum_{\lfloor \frac{d-1}{2} \rfloor + 1}^{n} \binom{n}{i} p^i (1-p)^{n-i}.
\end{aligned} \tag{68}
$$

For a low depolarizing probability p, the expression given in Eq. (68) can be approximated in order to determine the upper bound of the output fidelity as follows:

$$
F_{\text{out}} \approx 1 - \binom{n}{\lfloor \frac{d-1}{2} \rfloor + 1} p^{\lfloor \frac{d-1}{2} \rfloor + 1}. \tag{69}
$$

Since the threshold fidelity satisfies the relationship of $F_{\text{th}} = F_{\text{in}} = F_{\text{out}}$, we can substitute $F_{\text{out}} = F_{\text{th}}$ and $p = 1 - F_{\text{th}}$ into Eq. (69). Finally, the upper bound for the threshold probability can be encapsulated as

$$
F_{\text{th}}(n, d) = 1 - \binom{n}{\lfloor \frac{d-1}{2} \rfloor + 1}^{-1/\lfloor \frac{d-1}{2} \rfloor}. \tag{70}
$$

For example, the threshold for a distance-3 colour code having a quantum coding rate $r_Q = 1/7$ based on Fig. 22 is $F_{\text{th}} = 0.942$, while using the upper bound approximation of the fidelity threshold in Eq. (70) we have $F_{\text{th}} = 0.952$. For the distance-3 of rotated surface code, surface code and toric code, the threshold fidelity values based on Fig. 22 are $F_{\text{th}} = 0.968$, $F_{\text{th}} = 0.986$ and $F_{\text{th}} = 0.993$, respectively. By using the approximation of Eq. (70), the upper bound fidelity thresholds are given by $F_{\text{th}} = 0.972$, $F_{\text{th}} = 0.987$

and $F_{\text{th}} = 0.994$, respectively for the distance-3 rotated surface code, surface code and toric code. Here, we use the family of QTECCs as our representative examples, while the threshold fidelity of Eq. (70) is generically applicable for any QSCs using hard-decision syndrome decoding. Ultimately, the implementation of QTECCs are capable of reducing the effect of quantum decoherence, which is demonstrated by the QBER reduction and also improving the reliability of quantum channel, which is demonstrated by the fidelity improvement.

VI. CONCLUSIONS

We portrayed the evolution of the topological error correction codes designed in the classical domain to their quantum-domain dual pairs. We showed that by arranging the bits of the codeword on a lattice structure in classical domain provides a benificial inherent error correction capability. Furthermore, for a long codeword, the classical topological error correction codes (TECCs) correspond to the family of LDPC codes exhibiting attractive properties, such as unbounded minimum distance as a function of the codeword length, structured construction and a coding rate of $r = 1/2$. By contrast, the quantum topological error correction codes (QTECCs) are more suitable for applications requiring short to moderate codeword lengths, since the quantum coding rate of QTECCs tends to zero for a long codeword. We characterized the performance of QTECCs in the face of the quantum depolarizing channel in terms of the QBER attained. First, we showed that QTECCs are highly degenerate quantum codes, therefore the classical simulation is only capable of portraying the performance of QTECCs without considering the degeneracy property. Secondly, we demonstrated that increasing the minimum distance of the QTECCs improves the QBER performance. Additionally, we normalized the performance by taking the coding rate into consideration by introducing the *distance from the hashing bound*. Explicitly, we have shown that the growth of minimum distance of QTECCs upon increasing the codeword length is not fast enough to compensate for the increased codeword length. Consequently, the QBER performance of QTECCs gradually tends to the *ultimate distance from the hashing bound*. Finally, we determined the fidelity threshold for QSCs based on hard-decision syndrome decoding, which represents the minimum fidelity value required for a quantum system in order to glean benefits from QSCs. Ultimately, the employment of QSCs will improve the reliability of quantum computers.

ACKNOWLEDGMENT

REFERENCES

[1] P. W. Shor, "Fault-tolerant quantum computation," in *Proc. 37th Annu. Symp. Found. Comput. Sci.*, 1996, pp. 56–65, 1996.

[2] J. Preskill, "Reliable quantum computers," *Proc. Roy. Soc. London A, Math., Phys. Eng. Sci.*, vol. 454, no. 1969, pp. 385–410, 1998.

[3] D. P. DiVincenzo, "The physical implementation of quantum computation," *Fortschritte Phys.*, vol. 48, nos. 9–11, pp. 771–783, 2000.

[4] D. Gottesman, "Stabilizer codes and quantum error correction," Ph.D. dissertation, California Inst. Technol., Pasadena, CA, USA, 1997.

[5] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, "Quantum error correction and orthogonal geometry," *Phys. Rev. Lett.*, vol. 78, no. 3, p. 405, 1997.

[6] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, "Quantum error correction via codes over GF(4)," *IEEE Trans. Inf. Theory*, vol. 44, no. 4, pp. 1369–1387, Jul. 1998.

[7] M. Grassl and T. Beth, "Quantum BCH codes," in *Proc. Int. Symp. Theor. Elect. Eng.*, 1999, pp. 207–212.

[8] M. Grassl, W. Geiselmann, and T. Beth, "Quantum Reed–Solomon codes," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*. Berlin, Germany: Springer, 1999, pp. 231–244.

[9] H. Ollivier and J.-P. Tillich, "Description of a quantum convolutional code," *Phys. Rev. Lett.*, vol. 91, no. 17, p. 177902, 2003.

[10] D. Poulin, J. P. Tillich, and H. Ollivier, "Quantum serial turbo codes," *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2776–2798, Jun. 2009.

[11] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, "The road from classical to quantum codes: A hashing bound approaching design procedure," *IEEE Access*, vol. 3, pp. 146–176, Mar. 2015.

[12] M. S. Postol. (2001). "A proposed quantum low density parity check code." [Online]. Available: https://arxiv.org/abs/quant-ph/0108131

[13] D. MacKay, G. Mitchison, and P. McFadden, "Sparse-graph codes for quantum error correction," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2315–2330, Oct. 2004.

[14] T. Camara, H. Ollivier, and J.-P. Tillich. (2005). "Constructions and performance of classes of quantum LDPC codes." [Online]. Available: https://arxiv.org/abs/quant-ph/0502086

[15] T. Camara, H. Ollivier, and J.-P. Tillich, "A class of quantum LDPC codes: Construction and performances under iterative decoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2007, pp. 811–815, 2007.

[16] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, "Fifteen years of quantum LDPC coding and improved decoding strategies," *IEEE Access*, vol. 3, pp. 2492–2519, Nov. 2015.

[17] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A, Gen. Phys.*, vol. 52, no. 4, p. R2493(R), 1995.

[18] A. M. Steane, "Error correcting codes in quantum theory," *Phys. Rev. Lett.*, vol. 77, no. 5, p. 793, 1996.

[19] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, "Perfect quantum error correcting code," *Phys. Rev. Lett.*, vol. 77, no. 1, p. 198, 1996.

[20] *IBM Quantum Experience*. Accessed: Oct. 1, 2017. [Online]. Available: https://www.research.ibm.com/ibm-q/

[21] H. Bombin and M. A. Martin-Delgado, "Homological error correction: Classical and quantum codes," *J. Math. Phys.*, vol. 48, no. 5, p. 052105, 2007.

[22] A. Y. Kitaev, "Quantum computations: Algorithms and error correction," *Russian Math. Surv.*, vol. 52, no. 6, pp. 1191–1249, 1997.

[23] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," *Ann. Phys.*, vol. 303, no. 1, pp. 2–30, Jan. 2003.

[24] S. B. Bravyi and A. Y. Kitaev. (1998). "Quantum codes on a lattice with boundary." [Online]. Available: https://arxiv.org/abs/quant-ph/9811052

[25] H. Bombin and M. A. Martin-Delgado, "Topological quantum distillation," *Phys. Rev. Lett.*, vol. 97, no. 18, p. 180501, 2006.

[26] G. Zémor, "On Cayley graphs, surface codes, and the limits of homological coding for quantum error correction," in *Coding and Cryptology*. Berlin, Germany: Springer, 2009, pp. 259–273.

[27] A. Couvreur, N. Delfosse, and G. Zémor, "A construction of quantum LDPC codes from Cayley graphs," *IEEE Trans. Inf. Theory*, vol. 59, no. 9, pp. 6087–6098, Sep. 2013.

[28] J.-P. Tillich and G. Zémor, "Quantum LDPC codes with positive rate and minimum distance proportional to $n^{1/2}$," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun./Jul. 2009, pp. 799–803.

[29] A. A. Kovalev and L. P. Pryadko, "Improved quantum hypergraph-product LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2012, pp. 348–352.

[30] J.-P. Tillich and G. Zémor, "Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 1193–1202, Feb. 2014.

[31] C. Horsman, A. G. Fowler, S. Devitt, and R. Van Meter, "Surface code quantum computing by lattice surgery," *New J. Phys.*, vol. 14, no. 12, 2012.

[32] N. Delfosse, "Tradeoffs for reliable quantum information storage in surface codes and color codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2013, pp. 917–921.

[33] S. Bravyi and M. B. Hastings, "Homological product codes," in *Proc. 46th Annu. ACM Symp. Theory Comput.*, 2014, pp. 273–282.

[34] L. Hanzo, T. H. Liew, B. L. Yeap, R. Y. S. Tee, and S. X. Ng, *Turbo Coding, Turbo Equalisation and Space-Time Coding: EXIT-Chart-Aided Near-Capacity Designs for Wireless Channels*. Hoboken, NJ, USA: Wiley, 2011.

[35] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533–547, Sep. 1981.

[36] R. W. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. J.*, vol. 29, no. 2, pp. 147–160, Apr. 1950.

[37] E. N. Gilbert, "A comparison of signalling alphabets," *Bell Syst. Tech. J.*, vol. 31, no. 3, pp. 504–522, 1952.

[38] D. Chandra *et al.*, "Quantum coding bounds and a closed-form approximation of the minimum distance versus quantum coding rate," *IEEE Access*, vol. 5, pp. 11557–11581, 2017.

[39] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[40] P. A. M. Dirac, "A new notation for quantum mechanics," *Math. Proc. Cambridge Philos. Soc.*, vol. 35, no. 3, pp. 416–418, 1939.

[41] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[42] D. Gottesman, "Class of quantum error-correcting codes saturating the quantum Hamming bound," *Phys. Rev. A, Gen. Phys.*, vol. 54, no. 3, p. 1862, 1996.

[43] A. Ekert and C. Macchiavello, "Quantum error correction for communication," *Phys. Rev. Lett.*, vol. 77, no. 12, p. 2585, 1996.

[44] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Phys. Rev. A, Gen. Phys.*, vol. 54, no. 2, p. 1098, 1996.

[45] R. Steele and L. Hanzo, *Mobile Radio Communications: Second and Third Generation Cellular and WATM Systems*. Hoboken, NJ, USA: Wiley, 1999.

[46] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, "Mixed-state entanglement and quantum error correction," *Phys. Rev. A, Gen. Phys.*, vol. 54, no. 5, p. 3824, 1996.

[47] R. Jozsa, "Fidelity for mixed quantum states," *J. Mod. Opt.*, vol. 41, no. 12, pp. 2315–2323, 1994.

[48] B. Schumacher, "Quantum coding," *Phys. Rev. A, Gen. Phys.*, vol. 51, no. 4, p. 2738, 1995.

[49] E. Knill and R. Laflamme, "Theory of quantum error-correcting codes," *Phys. Rev. A, Gen. Phys.*, vol. 55, no. 2, p. 900, 1997.

ZUNAIRA BABAR received the B.Eng. degree in electrical engineering from the National University of Sciences and Technology, Islamabad, Pakistan, in 2008, and the M.Sc. degree (Hons.) and the Ph.D. degree in wireless communications from the University of Southampton, U.K., in 2011 and 2015, respectively.

Her research interests include quantum error correction codes, channel coding, coded modulation, iterative detection, and cooperative communications.

HUNG VIET NGUYEN received the B.Eng. degree in electronics and telecommunications from the Hanoi University of Science and Technology, Hanoi, Vietnam, in 1999, the M.Eng. degree in telecommunications from the Asian Institute of Technology, Bangkok, Thailand, in 2002, and the Ph.D. degree in wireless communications from the University of Southampton, Southampton, U.K., in 2013. Since 1999, he has been a Lecturer with the Post and Telecommunications Institute of Technology, Vietnam. He is involved in the OPTIMIX and CONCERTO European projects. He is currently a Post-Doctoral Researcher with the Southampton Wireless Group, University of Southampton.

His research interests include cooperative communications, channel coding, network coding, and quantum communications.

DIMITRIOS ALANIS (S'13) received the M.Eng. degree in electrical and computer engineering from the Aristotle University of Thessaloniki in 2011 and the M.Sc. and Ph.D. degrees in wireless communications from the University of Southampton, U.K., in 2012 and 2017, respectively. He is currently a Research Fellow with the Southampton Wireless Group, School of Electronics and Computer Science, University of Southampton.

His research interests include quantum computation and quantum information theories, quantum search algorithms, cooperative communications, resource allocation for self-organizing networks, bio-inspired optimization algorithms, and classical and quantum game theories.

DARYUS CHANDRA (S'15) received the M.Eng. degree in electrical engineering from Universitas Gadjah Mada, Indonesia, in 2014. He is currently pursuing the Ph.D. degree with the Southampton Wireless Group, School of Electronics and Computer Science, University of Southampton, U.K. He received the Scholarship Award from the Indonesia Endowment Fund for Education (Lembaga Pengelola Dana Pendidikan).

His research interests include classical and quantum error correction codes, quantum information, and quantum communications.

PANAGIOTIS BOTSINIS (S'12–M'16) received the M.Eng. degree from the School of Electrical and Computer Engineering, National Technical University of Athens, Greece, in 2010, and the M.Sc. degree (Hons.) and the Ph.D. degree in wireless communications from the University of Southampton, U.K., in 2011 and 2015, respectively. He is currently a Research Fellow with the Southampton Wireless Group, School of Electronics and Computer Science, University of Southampton. Since 2010, he has been a member of the Technical Chamber of Greece.

His research interests include quantum-assisted communications, quantum computation, iterative detection, orthogonal frequency-division multiplexing, multi-in multi-out, multiple access systems, coded modulation, channel coding, cooperative communications, as well as combinatorial optimization.

SOON XIN NG (S'99–M'03–SM'08) received the B.Eng. degree (Hons.) in electronic engineering and the Ph.D. degree in telecommunications from the University of Southampton, Southampton, U.K., in 1999 and 2002, respectively. From 2003 to 2006, he was a Post-Doctoral Research Fellow focusing on collaborative European research projects known as SCOUT, NEWCOM, and PHOENIX. Since 2006, he has been an Academic Staff Member with the School of Electronics and Computer Science, University of Southampton. He is involved in the OPTIMIX and CONCERTO European projects as well as the IU-ATC and UC4G projects. He is currently an Associate Professor of telecommunications with the University of Southampton.

His research interests include adaptive coded modulation, coded modulation, channel coding, space-time coding, joint source and channel coding, iterative detection, orthogonal frequency-division multiplexing, multi-in multi-out, cooperative communications, distributed coding, quantum error correction codes, and joint wireless-and-optical-fibre communications. He has authored over 200 papers and co-authored two books (John Wiley/IEEE Press) in this field. He is a Chartered Engineer and a fellow of the Higher Education Academy, U.K.

LAJOS HANZO (M'91–SM'92–F'04) received the master's degree in electronics in 1976 and the Ph.D. degree in 1983 from the Technical University of Budapest. During his 42-year career in telecommunications, he has held various research and academic posts in Hungary, Germany, and U.K. Since 1986, he has been with the School of Electronics and Computer Science, University of Southampton, U.K., where he is currently the Chair of telecommunications. He has successfully supervised 112 Ph.D. students. He has co-authored 18 books (John Wiley/IEEE Press) on mobile radio communications totaling in excess of 10 000 pages and published 1692 research contributions at IEEE Xplore. In 2009, he received the honorary doctorate "Doctor Honoris Causa" by the Technical University of Budapest. He was a TPC and General Chair of the IEEE conferences, presented keynote lectures and has been awarded a number of distinctions. He is directing a 100-strong academic research team, focusing on a range of research projects in the field of wireless multimedia communications sponsored by industry, the Engineering and Physical Sciences Research Council U.K., the European Research Council's Advanced Fellow Grant, and the Royal Society's Wolfson Research Merit Award. He is an enthusiastic supporter of industrial and academic liaison, and he offers a range of industrial courses.

Dr. Hanzo is a Chaired Professor with Tsinghua University, Beijing. He is a fellow of the Royal Academy of Engineering, Institution of Engineering and Technology, and the European Association for Signal Processing. He is also a Governor of the IEEE VTS. During 2008–2012, he was the Editor-in-Chief of the IEEE Press. He has 30 000+ citations. For further information on research in progress and associated publications please refer to http://www.wireless.ecs.soton.ac.uk.

● ● ●