Duality of Quantum and Classical Error Correction Codes: Design Principles and Examples

Zunaira Babar, Daryus Chandra, *Student Member, IEEE*, Hung Viet Nguyen, *Member, IEEE*, Panagiotis Botsinis, *Member, IEEE*, Dimitrios Alanis, *Student Member, IEEE*, Soon Xin Ng, *Senior Member, IEEE*, and Lajos Hanzo, *Fellow, IEEE*

***Abstract*—Quantum error correction codes (QECCs) can be constructed from the known classical coding paradigm by exploiting the inherent isomorphism between the classical and quantum regimes, while also addressing the challenges imposed by the strange laws of quantum physics. In this spirit, this paper provides deep insights into the duality of quantum and classical coding theory, hence aiming for bridging the gap between them. Explicitly, we survey the rich history of both classical as well as quantum codes. We then provide a comprehensive slow-paced tutorial for constructing stabilizer-based QECCs from arbitrary binary as well as quaternary codes, as exemplified by the dual-containing and non-dual-containing Calderbank–Shor–Steane (CSS) codes, non-CSS codes and entanglement-assisted codes. Finally, we apply our discussions to two popular code families, namely to the family of Bose–Chaudhuri–Hocquenghem as well as of convolutional codes and provide detailed design examples for both their classical as well as their quantum versions.**

***Index Terms*—Channel coding, quantum error correction, BCH codes, convolutional codes.**

ACRONYMS

ARQ	Automatic-Repeat-reQuest
AWGN	Additive White Gaussian Noise
BCH	Bose-Chaudhuri-Hocquenghem
BCJR	Bahl, Cocke, Jelinek and Raviv
BER	Bit Error Ratio
BICM	Bit-Interleaved Coded Modulation
BICM-ID	Bit-Interleaved Coded Modulation with Iterative Decoding
CCMC	Continuous-input Continuous-output Memoryless Channel
CNOT	Controlled-NOT
CRC	Cyclic Redundancy Check
CRSS	Calderbank-Rains-Shor-Sloane
CSS	Calderbank-Shor-Steane
EA	Entanglement-Assisted
EXIT	EXtrinsic Information Transfer
FPTD	Fully-Parallel Turbo Decoder
FPQTD	Fully-Parallel Quantum Turbo Decoder
GF	Galois Field
GV	Gilbert-Varshamov bound
IRCC	IRregular Convolutional Code
LDPC	Low Density Parity Check
LUT	Look-Up Table
MAP	Maximum A Posteriori
ML	Maximum Likelihood
MLSE	Maximum Likelihood Sequence Estimation
MRRW	McEliece-Rodemich-Rumsey-Welch
PCM	Parity Check Matrix
PGZ	Peterso-Gorenstein-Zierler
QBCH	Quantum Bose-Chaudhuri-Hocquenghem
QBER	Quantum Bit Error Ratio
QCC	Quantum Convolutional Code
QECC	Quantum Error Correction Code
QIRCC	Quantum IRregular Convolutional Code
QKD	Quantum Key Distribution
QLDPC	Quantum Low Density Parity Check
QRS	Quantum Reed-Solomon
QSC	Quantum Stabilizer Code
QSDC	Quantum Secure Direct Communication
QTC	Quantum Turbo Code
QURC	Quantum Unity Rate Code
RM	Reed-Muller
RRNS	Redundant Residue Number System
RS	Reed-Solomon
RSC	Recursive Systematic Convolutional
SISO	Soft-In Soft-Out
SNR	Signal-to-Noise Ratio
SOVA	Soft-Output Viterbi Algorithm
TCM	Trellis-Coded Modulation
TTCM	Turbo Trellis Coded Modulation
URC	Unity Rate Code
VA	Viterbi Algorithm
XOR	Exclusive OR.

Manuscript received February 6, 2018; revised June 12, 2018; accepted July 22, 2018. Date of publication July 31, 2018; date of current version February 22, 2019. This work was supported in part by the European Research Council through the Advanced Fellow Award QuantCom and in part by the Engineering and Physical Sciences Research Council under Grant EP/PO34284/1. The research data for this paper is available at [https://doi.org/10.5258/SOTON/D0616]. *(Corresponding author: Lajos Hanzo.)*

The authors are with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: zb2g10@ecs.soton.ac.uk; dc2n14@ecs.soton.ac.uk; sxn@ecs.soton.ac.uk; lh@ecs.soton.ac.uk).

Digital Object Identifier 10.1109/COMST.2018.2861361

LIST OF SYMBOLS

General Notation

- The notation $|.\rangle$ is used to indicate a quantum state. Therefore, $|\psi\rangle$ represents a qubit having the state ψ.

- The notation $|.|$ is used to indicate a magnitude operation. Therefore, $|\alpha|$ represents the magnitude of a complex number α.
- The notation $\star$ is used to indicate the symplectic product.
- The notation $\otimes$ is used to indicate the tensor product.
- The notation $\circledast$ is used to indicate the discrete convolution operation.
- The notation $\sum$ is used to indicate the sum operation.
- The notation $\langle , \rangle$ is used to represent the inner product.
- The GF(4) variables are represented with a $\hat{}$ on top, e.g., $\hat{x}$.
- The notation (*n, k*) is used for a classical code, while the notation [*n, k*] is used for a quantum code.
- The superscript ^{T} is used to indicate the matrix transpose operation. Therefore, $\mathbf{x}^T$ represents the transpose of the matrix $\mathbf{x}$.

Special Symbols

η	Spectral efficiency.
B	Classical channel bandwidth.
c	Number of pre-share entangled qubits (ebits).
C	Classical code space.
$\mathcal{C}$	Quantum code space.
$\mathbb{C}$	Set of complex numbers.
C	Classical channel channel.
$C_Q(.)$	Quantum channel capacity.
E	Entanglement consumption rate.
$\mathbb{F}_q$	Galois field GF(q).
G	Generator matrix.
$\mathcal{G}_n$	n-qubit Pauli group.
g_i	ith stabilizer generator.
H	Parity check matrix.
$\mathcal{H}$	Stabilizer group.
$H_2(.)$	Binary entropy function.
H	Hadamard gate.
I	Pauli-**I** operator.
k	Length of information word.
n	Length of codeword.
N	Classical noise power.
p	Channel error (or flip) rate, e.g., channel depolarizing probability.
$\mathcal{P}$	Pauli error inflicted on the transmitted codeword.
R_c	Equivalent classical coding rate of a quantum code.
R_Q	Quantum coding rate.
S	Classical signal power.
Tr[.]	Trace operator.
$\mathcal{V}$	Clifford encoder.
X	Pauli-**X** operator.
Y	Pauli-**Y** operator.
Z	Pauli-**Z** operator.

I. Introduction

IF *COMPUTERS that you build are quantum,*
Then spies everywhere will all want 'em.
Our codes will all fail,
And they'll read our e-mail,
Till we get crypto that's quantum, and daunt 'em.

Jennifer and Peter Shor

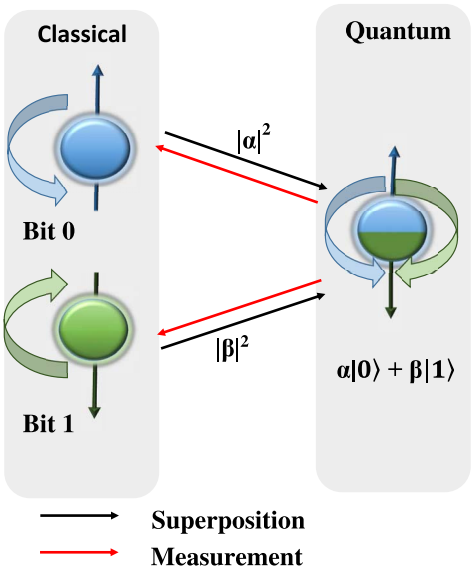


Fig. 1. Realization of a classical and quantum bit using the spin of an electron, where spin-up denotes the state $|0\rangle$ (or classical bit 0), while spin-down represents the state $|1\rangle$ (or classical bit 1). A qubit exists in superposition of the two states, but collapses to a single definite value (or state) upon measurement.

In the midst of the fast technological advances seen over the last several decades, 'Quantum Technology' has emerged as a promising candidate, which has the potential of radically revolutionizing the way we compute as well as communicate. Quantum technology derives its strengths from harnessing the peculiar laws of quantum physics, namely the superposition and entanglement. The fundamental postulates of quantum physics are rather different from the widely known and well-understood laws of classical physics, as exemplified by Newton's laws and Maxwell's equations.

A classical bit can assume the value of either 0 or 1 at any particular instant. By contrast, a quantum bit (qubit),[1] which is the integral constituent unit of a quantum system, exists in a 'superposition' of the states $|0\rangle$ and $|1\rangle$ until it is 'measured' or 'observed', as illustrated in Fig. 1. Explicitly, a qubit concurrently exists in the states $|0\rangle$ and $|1\rangle$. The resultant superimposed state of a qubit can be described using the state vector:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1}$$

where $|\ \rangle$ is called the Ket or Dirac notation [1] used for denoting a quantum state. Furthermore, the complex coefficients α and β may take any arbitrary value as long as $|\alpha|^2 + |\beta|^2 = 1$. Upon 'measurement' or 'observation' invoked for determining its value, the qubit $|\psi\rangle$ either collapses to the state $|0\rangle$ or to the state $|1\rangle$, which may happen with a probability of $|\alpha|^2$ and $|\beta|^2$, respectively, as exemplified in Fig. 1. Hence, a qubit is basically a 2-dimensional state vector, while an N-qubit composite system may be represented as a 2^N-dimensional state

[1] A classical bit or qubit can take different forms, for example two energy levels of an atom, different alignments of a nuclear/electronic/atomic spin, two different photon polarizations, or the charge/current/energy of a Josephson junction.

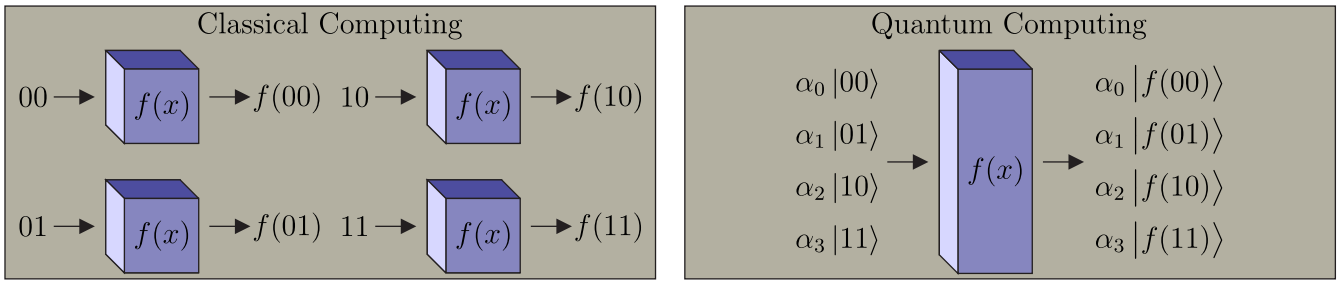


Fig. 2. Comparison of classical and quantum processing of a function $f(x)$ defined as $f(x) : \{0,1\}^2 \rightarrow \{0,1\}^2$ [3]. Classical system serially computes $f(x)$ for all possible $x \in \{00, 01, 10, 11\}$; hence, requiring four evaluations. By contrast, a quantum system concurrently processes all the possible x values, since a 2-qubit quantum register exists in superposition of all the four states, i.e., $|\psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle$; hence, requiring a single evaluation. The resulting outcome $(\alpha_0|f(00)\rangle + \alpha_1|f(01)\rangle + \alpha_2|f(10)\rangle + \alpha_3|f(11)\rangle)$ is also in superposition of all the four possibilities. Please note that it is not possible to read all the four values of $f(x)$, since the quantum register will collapse to one of the four values upon measurement. Nonetheless, the superimposed output may be processed further to get a desired property of the function $f(x)$, for example the minimum or maximum of $f(x)$ [4]–[7].

vector, which is formulated as:

$$\alpha_0|00\ldots0\rangle + \alpha_1|00\ldots1\rangle + \cdots + \alpha_{2^N-1}|11\ldots1\rangle, \tag{2}$$

where $\alpha_i \in \mathbb{C}$ and $\sum_{i=0}^{2^N-1} |\alpha_i|^2 = 1$. To elaborate further, an N-qubit system concurrently exists in superposition of all the 2^N possible values, which gives quantum systems the inherent property of quantum parallelism, as exemplified in Fig. 2.[2]

In contrast to superposition, 'entanglement', which Einstein termed as a 'spooky action at a distance' [8], is the mysterious, correlation-like property of two or more qubits, which implies that the entangled N-qubit state cannot be expressed as tensor product of the individual qubits. For example, consider a 2-qubit state $|\psi\rangle$ given by:

$$|\psi\rangle = \alpha|00\rangle + \beta|11\rangle, \tag{3}$$

and having non-zero coefficients α and β. It is impossible to express $|\psi\rangle$ as the tensor product of constituent qubits, because we have [2]:

$$\alpha|00\rangle + \beta|11\rangle \neq (\alpha_1|0\rangle + \beta_1|1\rangle) \otimes (\alpha_2|0\rangle + \beta_2|1\rangle), \tag{4}$$

for any choice of α_i and β_i subject to normalization, where $\otimes$ denotes the tensor product.[3] Consequently, a strange relationship exists between the two entangled qubits, which entails that measuring one of them also reveals the value of the other, even if they are geographically separated. Explicitly, if the first qubit of Eq. (3) collapses to the state $|0\rangle$ upon measurement, which may happen with a probability $|\alpha|^2$, then the second qubit is definitely $|0\rangle$. Similarly, if the first qubit collapses to the state $|1\rangle$, which may occur with a probability $|\beta|^2$, then the second qubit is also $|1\rangle$.

The phenomenon of 'superposition' as well as 'entanglement' have no counterparts in the classical domain, but they give rise to a new range of powerful computing and secure communication paradigms. For example, quantum computing algorithms have the potential to solve problems often deemed intractable at a substantially reduced complexity, as exemplified by Shor's pioneering factorization algorithm [6] and Grover's search algorithm [7]. This astounding processing power is derived from the inherent quantum parallelism resulting from quantum-domain superposition. More specifically, in contrast to an N-bit classical register, which can only store *one* of the 2^N possible values, an N-qubit quantum register can hold *all* the 2^N possible values (or states) concurrently, hence facilitating parallel processing, whose complexity is deemed equivalent to a single classical evaluation. This massive parallel processing potential may be beneficially exploited in large-scale communication systems' processes, for example in multi-user detection [9], [10] and in routing optimization [11], [12], as well as in diverse other applications, such as data mining [13] and Gait Recognition [14], [15], just to name a few.

It is anticipated that the enormous processing capability of quantum computing algorithms may threaten the integrity of the state-of-the-art trusted classical public key encryption, which relies on the computational complexity of the underlying mathematical functions. While classical cryptography is at risk of being deciphered due to quantum computing, quantum communications support secure data dissemination, since any 'measurement' or 'observation' by an eavesdropper perturbs the quantum superposition, hence intimating the parties concerned [2], [16]. Some of the main applications of secure quantum communications are Quantum Key Distribution (QKD) techniques [17], [18], Quantum Secure Direct Communication (QSDC) [19]–[21], and unconditional quantum location verification [22] for the future driverless 'Quantum Car' [23] and quantum geo encryption [24]. Deploying quantum communications is also imperative for making the future 'Quantum Internet' (Qinternet) [25] a reality. Explicitly, the Qinternet is envisaged as a global network of heterogeneous quantum systems, which may be interconnected through quantum channels in pursuit of building larger quantum systems, for example ultra-powerful distributed quantum computers [26], [27], long-haul secure QKD, QKD and quantum based location verification aided secure banking transactions, as well as ultra-precise quantum clocks for global synchronization, as illustrated in Fig. 3. It is pertinent to mention here that the quantum backhaul, which is likely to be a combination of free-space wireless channels and optical fibers, is particularly suitable for the

[2]Please refer to [2] for the fundamentals of quantum mechanics.

[3]The right hand side of Eq. (4) can be expanded as follows:

$$\alpha|00\rangle + \beta|11\rangle \neq \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle.$$

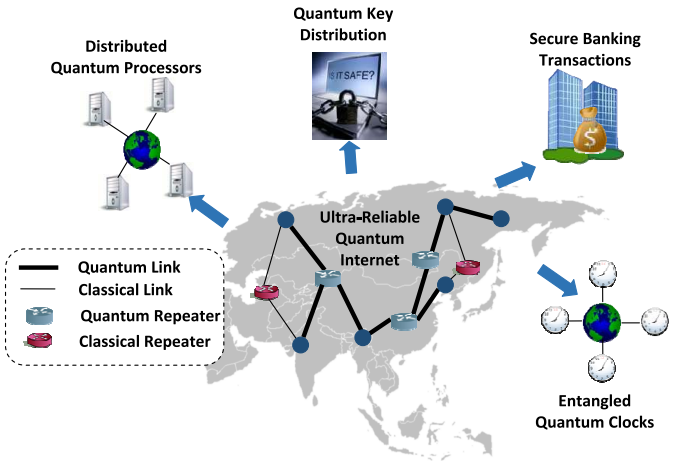


Fig. 3. Stylized illustration of the global 'Qinternet' interconnecting heterogeneous quantum processing and communication nodes over large distances, for example for distributed quantum computing, long-haul QKD, QKD and location verification aided secure banking transactions, as well as for quantum clock aided ultra-precise synchronization and navigation.

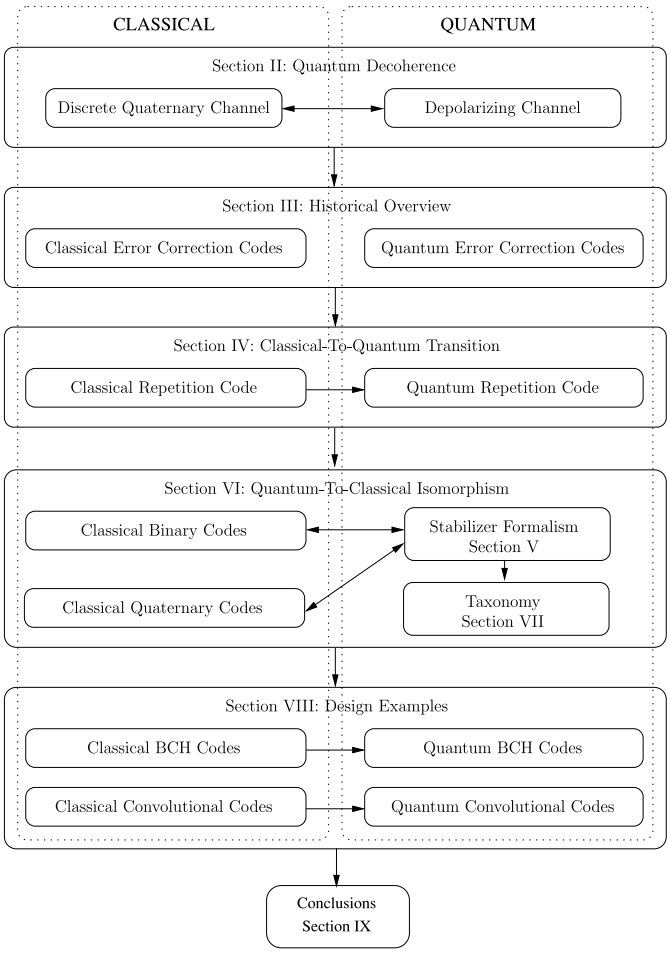


Fig. 4. Paper rationale.

Qinternet owing to the inherent quantum parallelism [25]. More specifically, an N-qubit quantum state would require only N uses of the quantum channel for transmitting the complete state information, while 2^N channel uses would be required if classical transmission is invoked. Similarly, if k N-qubit quantum nodes are entangled, then their overall capacity will be that of a (kN)-qubit system having a 2^{kN}-dimensional state space. By contrast, if the k N-qubit nodes are classically connected, they will have an effective state space of $k2^n$. Hence, quantum connectivity guarantees an exponentially larger state space compared to classical connectivity.

Unfortunately, the quantum channels as well as the quantum systems of Fig. 3 are not perfect, which is a major impediment to the practical realization of a global Qinternet. More specifically, qubits may experience both channel-induced as well as quantum processing impairments [28]. Explicitly, the deleterious quantum channel attenuation measured in dB per km severely limits the reliable transmission rate, or equivalently the transmission range. For example, the secret key transmission rate of a QKD system decays exponentially with the distance [29]. By contrast, the quantum processing impairments are inflicted by the imperfections in the quantum hardware, such as the quantum gates.

Quantum-based communication systems support the transmission of both classical as well as of quantum information. When the information to be transmitted is classical, we may invoke the family of classical error correction techniques for counteracting the impact of quantum impairments [30], [31]. More specifically, the classical information is first encoded using a classical error correction code. The encoded bits are then *mapped* onto the qubits, which are transmitted over a quantum channel. The mapping of classical bits to qubits may be carried out for example by the so-called superdense coding protocol [30], [32]. Likewise, QKD also relies on classical error correction codes [33], [34]. By contrast, for a more general communication system, which supports the transmission of both classical as well as quantum information, and for reliable quantum computation, we have to resort to Quantum Error Correction Codes (QECCs), which exploit redundancy in the quantum domain. More explicitly, similar to the classical error correction codes, QECCs redress the perturbations resulting from quantum impairments, hence enabling qubits to retain their coherent quantum states for longer durations with a high probability. This has been experimentally demonstrated in [35]–[37].

QECCs relying on the quantum-domain redundancy are indispensable for conceiving a quantum communication system supporting the transmission of quantum information and also for quantum computing. Therefore, in this paper, we survey the intricate journey from the realm of classical channel coding theory to that of the QECCs, while also providing a slow-paced tutorial on the duality of these two seemingly different coding regimes. In particular, we provide deeper insights into the subtle similarities and differences between them.

A. Outline

Fig. 4 captures the rationale of this paper, while Fig. 5 provides an overview at a glance. We commence our discourse in Section II, where we detail the various quantum channel models and highlight the duality between the widely used quantum depolarizing channel and the classical discrete

- I Introduction
- II Quantum Decoherence
  - II-A Amplitude Damping Channel
  - II-B Phase Damping Channel
  - II-C Pauli Channel
- III Historical Overview
  - III-A Classical Coding Theory
    - III-A1 Design Objectives
    - III-A2 Error Correction Codes
  - III-B Quantum Coding Theory
    - III-B1 Design Objectives
    - III-B2 Error Correction Codes
- IV Classical-to-Quantum Transition
- V Stabilizer Formalism
  - V-A Stabilizer-based Code Design
  - V-B Classification of Error Patterns
- VI Quantum-to-Classical Isomorphism
  - VI-A Pauli-to-Binary Isomorphism
  - VI-B Pauli-to-Quaternary Isomorphism
- VII Taxonomy of Stabilizer Codes
  - VII-A Calderbank-Shor-Steane Codes
  - VII-B Non-CSS Codes
  - VII-C Entanglement-Assisted Codes
- VIII Design Examples
  - VIII-A Bose-Chaudhuri-Hocquenghem (BCH) Codes
    - VIII-A1 Classical BCH
    - VIII-A2 Quantum BCH
  - VIII-B Convolutional Codes (CC)
    - VIII-B1 Classical CC
    - VIII-B2 Quantum CC
- IX Conclusions & Design Guidelines

Fig. 5. Paper structure.

quaternary channel. We then survey the rich history of classical and quantum codes in Section III. In Section IV, we detail the transition from the classical to the quantum code designs with the help of simple design examples. Specifically, we design the quantum counterpart of the simple classical rate-1/3 repetition code. We then generalize our discussions in Section V, where we present the quantum version of classical linear block codes by relying on the so-called stabilizer formalism, which is a theoretical framework conceived for constructing quantum codes from the existing families of classical error correction codes. Continuing further our discussions, we next detail the quantum to classical isomorphism in Section VI, which is a useful analysis technique for mapping quantum codes onto the equivalent classical codes and vice versa. The quantum-to-classical mapping allows us to use the state-of-the-art classical syndrome decoding techniques in the quantum realm, while the inverse mapping, i.e., the classical-to-quantum mapping, helps in importing arbitrary classical codes into the quantum domain. Furthermore, based on this isomorphism, we present the taxonomy of stabilizer codes in Section VII. We also detail the associated design principles with examples. In Section VIII, we delve deeper into a pair of popular code families, explicitly the Bose-Chaudhuri-Hocquenghem (BCH) codes and the convolutional codes, by providing tutorial insights into their classical as well as quantum counterparts. Finally, we conclude our discourse in Section IX.

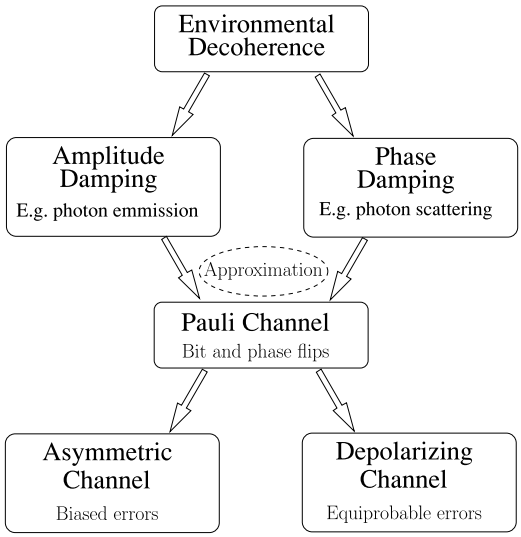


Fig. 6. Quantum channel models.

II. Quantum Decoherence

Environmental decoherence generally constitutes a major source of quantum impairments, which may occur for example during quantum transmission or quantum processing as well as in quantum memories. In this section, we review the quantum channels of Fig. 6, which are widely used for modeling environmental decoherence. Explicitly, our intention is to help the readers understand the duality between quantum and classical channels.

A. Amplitude Damping Channel

In the simple terms, environmental decoherence may be described as the undesired interaction, or more specifically entanglement, of the qubit with the environment, which perturbs its coherent superposition of basis states. In one such instance, the qubit (or quantum system) loses energy due to its interaction with the environment, for example the excited state of the qubit decays due to the spontaneous emission of a photon or the photon is lost (or absorbed) during its transmission through optical fibers [38], [39]. This decoherence process can be conveniently modeled using an amplitude damping channel. Let us consider a qubit realized using a two-level atom having the ground state $|0\rangle$ and the excited state $|1\rangle$. Furthermore, let $|0\rangle_E$ and $|1\rangle_E$ be the basis states of the environment initialized to the vacuum state $|0\rangle_E$. Then, the amplitude damping channel characterizes the evolution of the resultant system as follows [39]:

$$\begin{aligned} &|0\rangle|0\rangle_E \rightarrow |0\rangle|0\rangle_E, \\ &|1\rangle|0\rangle_E \rightarrow \sqrt{1-\gamma}|1\rangle|0\rangle_E + \sqrt{\gamma}|0\rangle|1\rangle_E, \end{aligned} \tag{5}$$

where γ is the damping probability, or more specifically the probability of losing a photon. In physically tangible terms, Eq. (5) implies that the state of the qubit remains the same if it is in the ground state $|0\rangle$, while it looses a photon with a probability of γ, when in the excited state $|1\rangle$. Explicitly, in the event of a photon loss, the state of the qubit changes from $|1\rangle$ to $|0\rangle$, while that of the environment changes from $|0\rangle_E$

to $|1\rangle_E$; hence resulting in the state $|0\rangle|1\rangle_E$ of Eq. (5), which may occur with a probability of γ. Based on Eq. (5), a qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, which is in coherent superposition of the basis states, entangles with the environment as:

$$|\psi\rangle|0\rangle_E \rightarrow \Big(\alpha|0\rangle + \beta\sqrt{1-\gamma}|1\rangle\Big)|0\rangle_E + \sqrt{\gamma}\beta|0\rangle|1\rangle_E. \tag{6}$$

It is pertinent to mention here that $|\psi\rangle$ is generally not an isolated qubit. It may be entangled with other qubits as part of an *N*-qubit composite quantum system. Hence, slightly 'abusing' the usual notation, the coefficients α and β represent the (*N*-1)-qubit states entangled to the states $|0\rangle$ and $|1\rangle$, respectively, of the qubit undergoing decoherence. We furthermore assume that each qubit interacts independently with the environment, hence the associated decoherence process is temporally and spatially uncorrelated. We can readily infer from Eq. (6) that if the environment is found to be in state $|0\rangle_E$, then $|\psi\rangle$ decoheres to $(\alpha|0\rangle + \beta\sqrt{1-\gamma}|1\rangle)$, which reduces to $\left(\frac{\alpha}{\sqrt{1-\gamma\beta^2}}|0\rangle + \frac{\beta\sqrt{1-\gamma}}{\sqrt{1-\gamma\beta^2}}|1\rangle\right)$ upon normalization, otherwise $|\psi\rangle$ collapses to $|0\rangle$.

If a quantum system is a statistical ensemble of pure states, then it may be described using the density operator (also called density matrix) ρ, as follows:

$$\rho \equiv \sum_i p_i|\psi_i\rangle\langle\psi_i|, \tag{7}$$

where p_i denotes the probability of occurrence of the *i*th pure state $|\psi_i\rangle$. Explicitly, a pure quantum state is one whose state vector $|\psi\rangle$ is exactly known. Hence, it is described by a single state vector $|\psi\rangle$ and the density operator of Eq. (7) reduces to $\rho = |\psi\rangle\langle\psi|$. By contrast, the mixed state of Eq. (7) is a probabilistic mixture (not superposition) of different pure states $|\psi_i\rangle$. Hence, we do not exactly know the state of the system and it may be found in the *i*th pure state $|\psi_i\rangle$ with a probability of p_i. This may happen for example due to coupling with the environment or due to inaccuracies of the equipment. The loss of energy in a generalized quantum system described by Eq. (7) may be modeled using an amplitude damping channel $\mathcal{N}_{\rm AD}$, which maps an input state, having the density operator ρ, as follows:

$$\mathcal{N}_{\rm AD}(\rho) = \mathbf{E}_0\rho\mathbf{E}_0^\dagger + \mathbf{E}_1\rho\mathbf{E}_1^\dagger, \tag{8}$$

where the error operators (also called Kraus operators)[4] $\mathbf{E}_0$ and $\mathbf{E}_1$ are given by [2]:

$$\mathbf{E}_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \quad \mathbf{E}_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}. \tag{9}$$

The decohered state of a qubit may be readily described by using the error operators of Eq. (9). Resuming our previous example of $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the error operator $\mathbf{E}_0$ corrupts $|\psi\rangle$ as follows:

$$\begin{aligned} \mathbf{E}_0|\psi\rangle &= \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ &= \begin{pmatrix} \alpha \\ \sqrt{1-\gamma}\beta \end{pmatrix} \\ &\equiv \alpha|0\rangle + \sqrt{1-\gamma}\beta|1\rangle, \end{aligned} \tag{10}$$

which occurs with a probability of $|\mathbf{E}_0|\psi\rangle|^2 = (1-\gamma\beta^2)$. Upon normalization, the corrupted state of Eq. (10) is reduced to:

$$\mathbf{E}_0|\psi\rangle = \frac{\alpha}{\sqrt{1-\gamma\beta^2}}|0\rangle + \frac{\beta\sqrt{1-\gamma}}{\sqrt{1-\gamma\beta^2}}|1\rangle. \tag{11}$$

Similarly, the error operator $\mathbf{E}_1$ acts on $|\psi\rangle$ as follows:

$$\begin{aligned} \mathbf{E}_1|\psi\rangle &= \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ &= \begin{pmatrix} \sqrt{\gamma}\beta \\ 0 \end{pmatrix} \\ &\equiv \sqrt{\gamma}\beta|0\rangle, \end{aligned} \tag{12}$$

which happens with a probability of $|\mathbf{E}_1|\psi\rangle|^2 = \gamma\beta^2$ and is equivalent to the classical bit $|0\rangle$. In realistic systems, γ at time instant *t* is characterized by the qubit relaxation time T_1 as follows [40]:

$$\gamma = 1 - e^{-t/T_1}. \tag{13}$$

B. Phase Damping Channel

Another instantiation of environmental decoherence, known as dephasing or phase damping, characterizes the loss of quantum information without the loss of energy, which may occur for example due to the scattering of photons, or the perturbation of electronic states caused by stray electrical charges. The error operators of the resultant phase damping channel $\mathcal{N}_{\rm PD}$ are defined as follows [2]:

$$\mathbf{E}_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{pmatrix}, \quad \mathbf{E}_1 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{pmatrix}, \tag{14}$$

where λ is the scattering probability of a photon (without loss of energy). We may observe that $\mathbf{E}_0$ of Eq. (14) is similar to the $\mathbf{E}_0$ of the amplitude damping channel, while the error operator $\mathbf{E}_1$ acts on $|\psi\rangle$ as follows:

$$\begin{aligned} \mathbf{E}_1|\psi\rangle &= \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ \sqrt{\lambda}\beta \end{pmatrix} \\ &\equiv \sqrt{\lambda}\beta|1\rangle, \end{aligned} \tag{15}$$

which occurs with a probability of $|\mathbf{E}_1|\psi\rangle|^2 = \lambda\beta^2$ and it is equivalent to the classical state $|1\rangle$. The probability λ relies on the relaxation time T_1 as well as on the dephasing time T_2, i.e., we have [40]:

$$\lambda = 1 - e^{\frac{t}{T_1} - \frac{2t}{T_2}}. \tag{16}$$

Intuitively, Eq. (13) and Eq. (16) imply that the qubit is likely to decohere if the operation time (transmission or processing

[4]A quantum channel $\mathcal{N}$ is a completely positive, trace-preserving linear mapping, which maps an input state having the density ρ as [2]:

$$\mathcal{N}(\rho) = \sum_k \mathbf{E}_k\rho\mathbf{E}_k^\dagger,$$

where the matrices $\mathbf{E}_k$ are known as the Kraus operators or error operators of the channel. Furthermore, we have $\sum_k \mathbf{E}_k^\dagger\mathbf{E}_k = \mathbf{I}$, where $\mathbf{I}$ is an identity matrix.

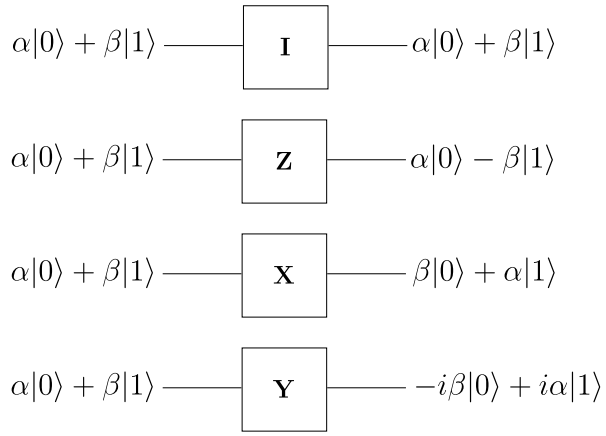


Fig. 7. Schematic of Pauli-**I**, Pauli-**Z**, Pauli-**X** and Pauli-**Y** gates.

or storage) t is comparable to the relaxation time T_1 and the dephasing time T_2. Equivalently, T_1 and T_2 characterize the life-time of a reliable qubit.

C. Pauli Channel

The environmental decoherence can be modeled using a combined amplitude and phase damping channel. However, it is not feasible to classically simulate such channels for an N-qubit composite system, since the resultant system has a 2^N-dimensional Hilbert space. For the sake of facilitating efficient classical simulations,[5] the combined amplitude and phase damping channel can be approximated using a so-called Pauli channel $\mathcal{N}_{\rm P}$, which maps an input state, having the density operator ρ, as follows [41]:

$$\mathcal{N}_{\rm P}(\rho) = (1 - p_z - p_x - p_y)\mathbf{I}\rho\mathbf{I} + p_z\mathbf{Z}\rho\mathbf{Z} + p_x\mathbf{X}\rho\mathbf{X} + p_y\mathbf{Y}\rho\mathbf{Y}, \tag{17}$$

where **I**, **X**, **Y** and **Z** are single-qubit Pauli operators (or gates) of Fig. 7 defined as:

$$\mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \ \mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \\ \mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \ \mathbf{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \tag{18}$$

while p_z, p_x and p_y are the probabilities of encountering **Z**, **X** and **Y** Pauli errors, respectively, which rely on the qubit relaxation and dephasing time as given below:

$$p_x = p_y = \frac{1}{4}\Big(1 - e^{-t/T_1}\Big) \\ p_z = \frac{1}{4}\Big(1 + e^{-t/T_1} - 2e^{-t/T_2}\Big). \tag{19}$$

Explicitly, **I** is an identity operator, or merely a repeat gate, which leaves the state $|\psi\rangle$ intact, as shown below:

$$\mathbf{I}|\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \equiv \alpha|0\rangle + \beta|1\rangle. \tag{20}$$

[5]Classical modeling of quantum systems is discussed in Section VI.

The operator **Z** is a phase-flip operator, which acts as:

$$\mathbf{Z}|\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix} \equiv \alpha|0\rangle - \beta|1\rangle, \tag{21}$$

while **X** is a bit-flip operator analogous to the classical NOT gate, which yields:

$$\mathbf{X}|\psi\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} \equiv \beta|0\rangle + \alpha|1\rangle. \tag{22}$$

By contrast, **Y** is a combined bit-and-phase-flip operator (**Y** = i**XZ**), which acts on $|\psi\rangle$ as:

$$\mathbf{Y}|\psi\rangle = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ = \begin{pmatrix} -i\beta \\ i\alpha \end{pmatrix} \equiv -i(\beta|0\rangle - \alpha|1\rangle). \tag{23}$$

Hence, the Pauli channel of Eq. (17) maps the input state $|\psi\rangle$ onto a linear combination of the original state (Pauli-**I** operation), phase-flipped state (Pauli-**Z** operation), bit-flipped state (Pauli-**X** operation), as well as bit-and-phase-flipped state (Pauli-**Y** operation) during the process of decoherence. In essence, the resultant quantum error is continuous in nature. We may observe in Eq. (19) furthermore that the time T_1 affects bit-flips, phase-flips as well as bit-and-phase-flips. By contrast, the time T_2 is only related to the phase-flip errors. This is because the bit-flip as well as bit-and-phase-flip errors are associated with amplitude damping, while the phase-flip errors result from phase damping. In most practical systems, the value of T_1 is several orders of magnitude higher than that of T_2 [42], [43]. Consequently, most practical quantum systems behave as so-called asymmetric channels and they experience more phase-flips than bit-flips as well as bit-and-phase-flips. Furthermore, a special class of Pauli channels, known as the 'depolarizing channel', models the worst-case scenario by assuming that all three errors are equally likely, i.e., $(p_z = p_x = p_y)$. Explicitly, a depolarizing channel having the probability p inflicts a phase-flip (Pauli-**Z**) or a bit-flip (Pauli-**X**) or bit-and-phase-flip (Pauli-**Y**) error with a probability of $p/3$ each, which may be mathematically encapsulated as:

$$\mathcal{N}_{\rm DP}(\rho) = (1 - p)\rho + \frac{p}{3}(\mathbf{Z}\rho\mathbf{Z} + \mathbf{X}\rho\mathbf{X} + \mathbf{Y}\rho\mathbf{Y}). \tag{24}$$

In this treatise, we will only consider the widely used depolarizing channel model.

The aforementioned quantum channel models are summarized in Fig. 8. We may observe in Fig. 8 that the Pauli channel may be deemed to be the quantum analogue of the classical discrete quaternary channel. However, while the classical quaternary channel may inflict only one of the four possible errors, the error inflicted by the Pauli channel may be in superposition of the four possible errors, i.e., **I**, **Z**, **X** and **Y**. The Pauli channel may further be simplified by using two independent bit-flip and phase-flip channels, which

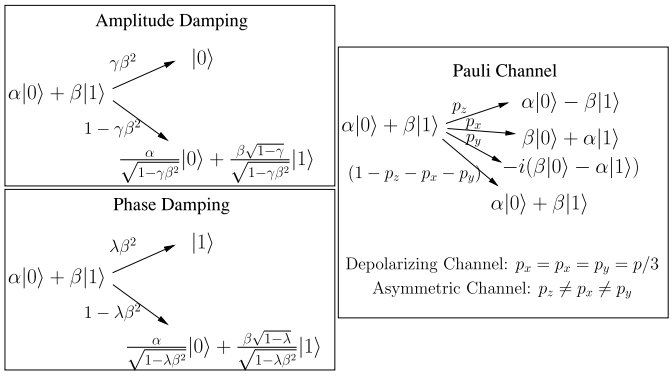


Fig. 8. Mathematical interpretation of quantum channel models.

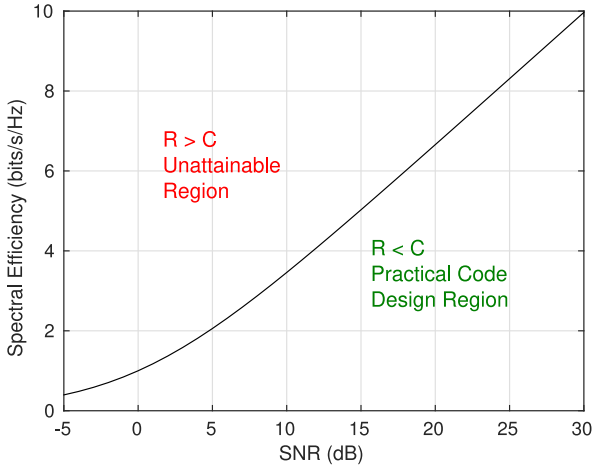


Fig. 9. Shannon capacity limit for AWGN channel characterized by Eq. (26).

are analogous to classical binary symmetric channels having cross-over probabilities of $(p_x + p_y)$ and $(p_z + p_y)$, respectively.

III. Historical Overview of Classical & Quantum Error Correction Codes

In this section, we survey the major milestones both in the realm of classical as well as in quantum coding theory, which are chronologically arranged in Table I.

A. Classical Coding Theory

1) Design Objectives: Shannon's pioneering work [44] on classical channel capacity marks the beginning of classical coding theory. Explicitly, Shannon predicted that sophisticated channel coding techniques, having coding rate R lower than the Shannon limit (or channel capacity) C, may be invoked for the sake of achieving reliable transmission over a noisy bandwidth-limited channel. Intuitively, this implies that it is possible to transmit information virtually free from errors, as long as the coding rate does not exceed the Shannon limit,which is characterized by the channel bandwidth B (Hz), the signal power S (Watts) and the uncorrelated Additive White Gaussian Noise (AWGN) power N (Watts) as follows:

$$\mathrm{C} = B \log_2\left(1 + \frac{S}{N}\right), \tag{25}$$

or equivalently in terms of the spectral efficiency (bits/s/Hz) as:

$$\eta = \frac{\mathrm{C}}{B} = \log_2\left(1 + \frac{S}{N}\right). \tag{26}$$

Hence, the Shannon limit of Eq. (25) (and equivalently Eq. (26)) quantifies the highest possible coding rates still capable of ensuring error-free transmission, as illustrated in Fig. 9. Furthermore, we may infer from Eq. (25) that the resultant information transfer rate of a system is limited by the channel bandwidth B as well as the system's Signal-to-Noise Ratio (SNR) S/N. As demonstrated in Fig. 9, the capacity limit increases upon increasing the SNR. Ultimately, when the SNR approaches infinity in the noiseless scenario, it is possible to achieve an infinite transmission rate even for a very low bandwidth. Similarly, the capacity limit also increases upon increasing the bandwidth. Hence, we may strike a trade off between the bandwidth and the SNR, as detailed and exemplified in [141, Sec. 2.13.1]. However, an infinite bandwidth does not guarantee an infinite transmission rate, because the noise power also increases upon increasing bandwidth, as shown mathematically in [141].

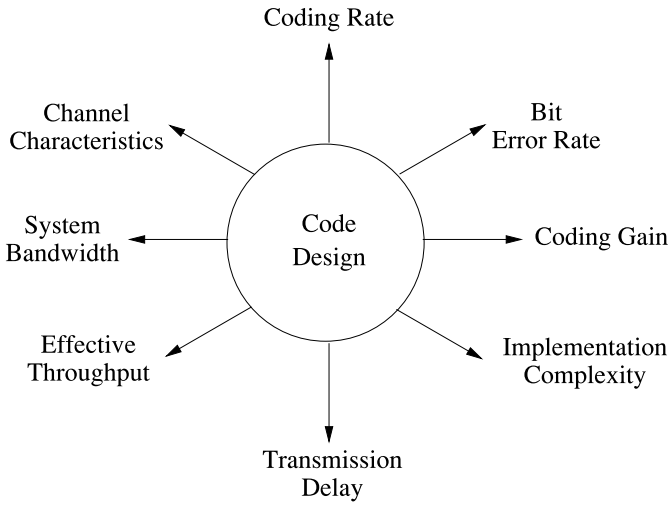


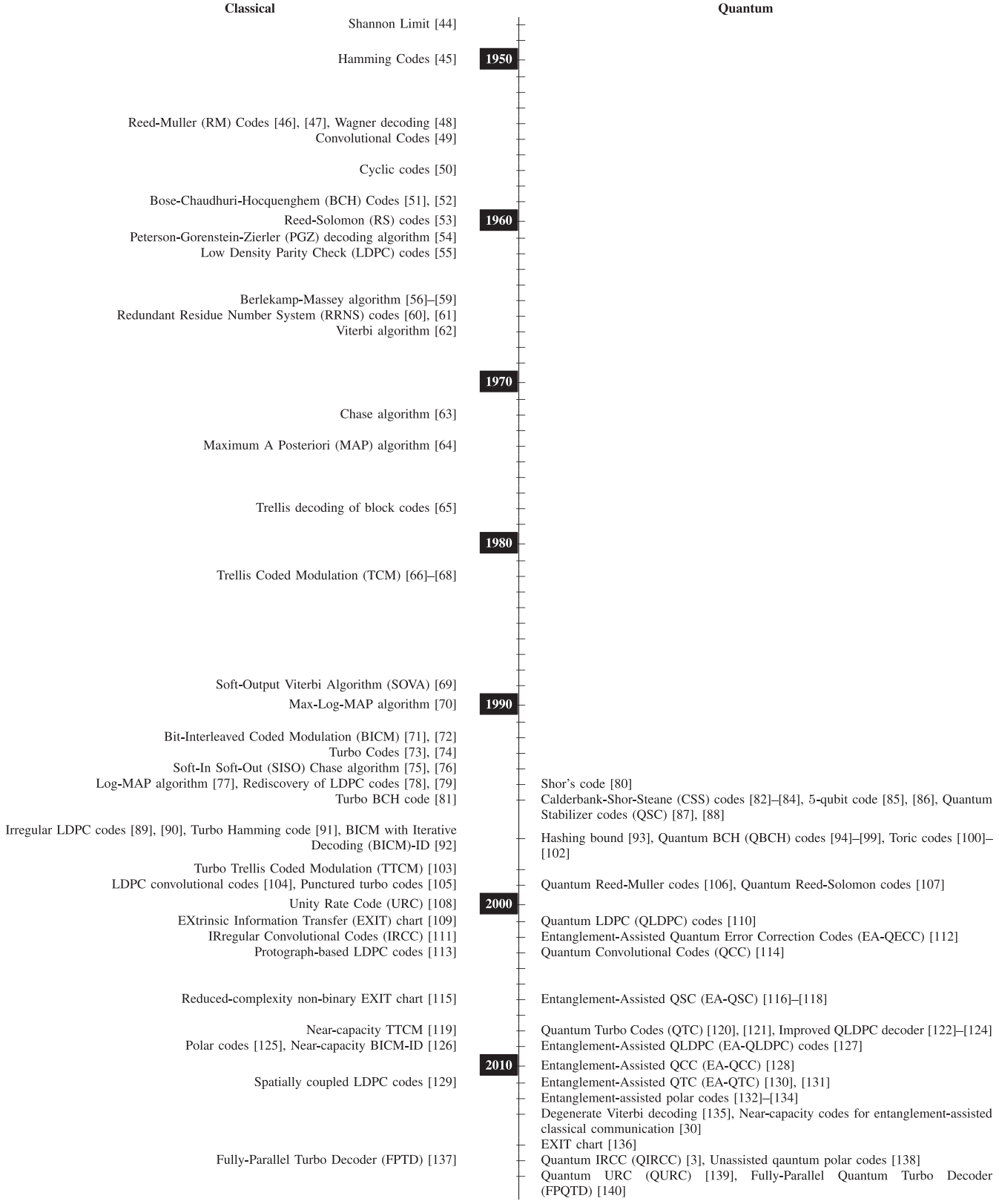
Fig. 10. Stylized representation of conflicting design parameters affecting the design of classical codes.

Shannon did not provide any explicit code constructions in his seminal work [44]. However, his work inspired the research community to design practical codes in line with the achievable code design region of Fig. 9. This in turn highlighted various other conflicting design trade-offs, which are captured in Fig. 10. For example, given particular channel conditions, a code may be optimized to achieve a lower Bit Error Ratio (BER) or a higher coding gain.[6] However, this typically imposes an increased decoding complexity and transmission delay, or reduced effective throughput, as detailed in [141] and [142].

The Shannon limit of Eq. (25) quantifies the capacity of a Continuous-input Continuous-output Memoryless Channel

[6]Coding gain quantifies the reduction in bit-energy achieved at a certain BER, when error correction is invoked.

TABLE I
MAJOR ACHIEVEMENTS IN THE CLASSICAL AND QUANTUM CODING PARADIGMS



(CCMC), which may only be achieved by infinitely long random-like codes. Since the state-of-the-art communication systems transmit binary information, several bounds have been conceived for characterizing the rate-versus-minimum-distance trade-off, rather than the rate-versus-SNR trade-off of Fig. 9. Explicitly, these bounds provide either an upper or a lower

TABLE II
RATE-VERSUS-MINIMUM-DISTANCE BOUNDS FOR CLASSICAL CODES [147], [148]. $H_2(p)$ DENOTES THE BINARY ENTROPY FUNCTION, WHICH IS EQUIVALENT TO $H_2(p) = p\log_2(p) + (1-p)\log_2(1-p)$

Classical Coding Bound	Finite	Asymptotic	Notes
Singleton [143]	$\frac{k}{n} \leq 1 - \left(\frac{d_{\min}-1}{n}\right)$	$\frac{k}{n} \leq 1 - \left(\frac{d_{\min}}{n}\right)$	a loose upper bound
Hamming [45]	$\frac{k}{n} \leq 1 - \frac{1}{n}\log_2\left(\sum_{j=0}^{t=\lfloor\frac{d_{\min}-1}{2}\rfloor}\binom{n}{j}\right)$	$\frac{k}{n} \leq 1 - H_2\left(\frac{d_{\min}}{2n}\right)$	tight upper bound for very high code rate
McEliece-Rodemich-Rumsey-Welch (MRRW) [144]		$\frac{k}{n} \leq H_2\left(\frac{1}{2} - \sqrt{\frac{d_{\min}}{n}\left(1-\frac{d_{\min}}{n}\right)}\right)$	tightest known asymptotic upper bound for medium and low rate codes
Plotkin [145]	$\frac{k}{n} \leq \frac{1}{n}\left(1 - \log_2\left(2 - \frac{n}{d_{\min}}\right)\right)$		tight upper bound for finite-length at $\frac{d_{\min}}{n} > \frac{1}{2}$
Gilbert-Varshamov (GV) [146]	$\frac{k}{n} \geq 1 - \frac{1}{n}\log_2\left(\sum_{j=0}^{d_{\min}-1}\binom{n}{j}\right)$	$\frac{k}{n} \geq 1 - H_2\left(\frac{d_{\min}}{n}\right)$	tightest known lower bound

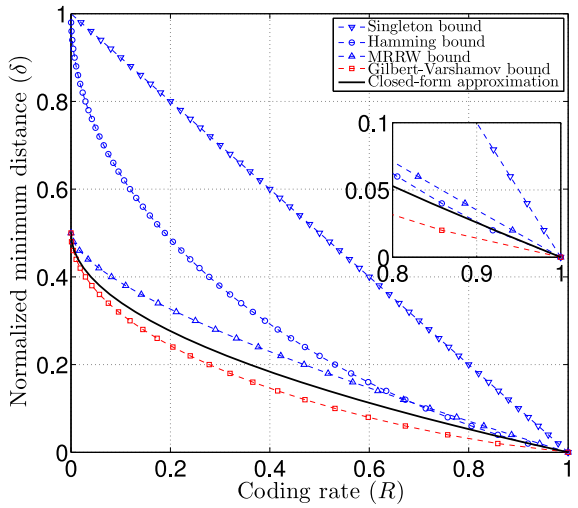


Fig. 11. Rate (R = k/n) versus normalized minimum distance $\delta = \frac{d_{\min}}{n}$ asymptotic bounds [148]. The closed-form approximation of [147] is also plotted, which relies on a simple quadratic function $R(\delta) = (2\delta - 1)^2$ and satisfies all the bounds. Upper bounds are plotted in blue, while the lower bound is plotted in red.

limit on the maximum coding rate $R = k/n$ given the minimum Hamming distance $d_{\min}$, or vice versa. Here, k and n denote the number of information and coded bits, respectively, while the minimum Hamming distance is defined as the minimum distance between any two legitimate binary codewords. Hence the resultant code is capable of correcting $t = (d_{\min} - 1)/2$ errors. Table II enlists the popular finite block-length as well as asymptotic ($n \to \infty$) coding bounds, while Fig. 11 plots the asymptotic bounds. Specifically, the Singleton bound [143] is a loose upper bound, while the Gilbert-Varshamov (GV) bound [146] is the tightest lower bound. Furthermore, the Hamming bound [45] provides a tight upper bound at high coding rates, while the McEliece-Rodemich-Rumsey-Welch (MRRW) bound [144] is the tightest upper bound for low and medium coding rates. The bounds of Table II give a range of achievable minimum distances, or more specifically the normalized minimum distances $\delta = \frac{d_{\min}}{n}$, for the desired coding rate, and hence do not provide a precise solution to the rate-versus-minimum-distance trade-off. Consequently, for the sake of approximating the optimum trade-off between the coding rate and the minimum distance, a simple invertible closed-form analytical expression $R(\delta) = (2\delta - 1)^2$ was proposed in [147], which satisfies all the asymptotic bounds of Table II, as demonstrated in Fig. 11. Akhtman *et al.* [147] also formulated the corresponding closed-form expressions for finite block-lengths, satisfying all the finite bounds of Table II.

2) Error Correction Codes: In 1950, Hamming conceived the first practical family of classical error correction codes [45]. More specifically, Hamming proposed an infinite family of binary linear block codes capable of encoding $k = (2^r - 1 - r)$ information bits into $n = (2^r - 1)$ coded bits for $r \geq 2$. The resultant codewords had a minimum Hamming distance of $d_{\min} = 3$, hence correcting $t = (d_{\min} - 1)/2 = 1$ errors. The Hamming codes may be classified as being 'perfect' codes, since the associated coding rate $R = k/n = 1 - r/(2^r - 1)$ is the maximum coding rate achievable for $d_{\min} = 3$ and for a block length of $n = (2^r - 1)$. Following these developments, in 1954, Reed [46] and Muller [47] independently conceived a class of multiple error correcting block codes, known as Reed-Muller (RM) codes. Reed also introduced a simple majority-logic based hard-decision decoder for RM codes in [46]. The same year, a soft-decision based decoding algorithm, known as Wagner decoding [48], was developed for a special class of RM codes.

The afore-mentioned linear block codes primarily relied on maximizing the minimum distance for a given pair of (n, k) codewords encoding k bits into n, or equivalently maximizing the coding rate given the $d_{\min}$ and n. The resultant families of Hamming and RM codes only support a limited range of code parameters given by $(n, k, d_{\min})$. For the sake of designing more codes offering a wider range of code parameters at an affordable implementation complexity, Elias discovered convolutional codes in 1955 [49], which marks the commencement of the so-called probabilistic coding era. Convolutional codes are capable of supporting encoding and decoding procedures operating in a sliding window, hence resulting in lower latencies than the above block codes. In this spirit, Viterbi

invented a *Maximum Likelihood Sequence Estimation* (MLSE) (or equivalently minimum Euclidean distance) algorithm for convolutional codes [62]. Explicitly, the Viterbi Algorithm (VA) aims for finding the most likely error sequence at an affordable decoding complexity. Although the VA is an MLSE algorithm, the resultant BER of the system is close to the minimum possible BER, but the latter was only achievable by a complex Maximum Likelihood (ML) decoder, which evaluates all valid coded sequences. To circumvent the high complexity of the latter ML decoder, Bahl *et al.* proposed the minimum BER decoding algorithm in 1974 [64], which was named the Maximum A Posteriori (MAP) algorithm. It is also known as BCJR after its inventors Bahl, Cocke, Jelinek and Raviv.

Pursuing further the realm of block codes, Prange investigated cyclic codes in 1957 [50]. Since the cyclic shift of codewords of cyclic codes are also legitimate codewords, the associated encoding and decoding procedures can be efficiently implemented using shift registers. Inspired by these developments, Hocquenghem [51] as well as Bose and Ray-Chaudhuri [52], [149] independently discovered the family of Bose-Chaudhuri-Hocquenghem (BCH) codes in 1959 and 1960, respectively. Specifically, BCH codes constitute the family of multiple-error correcting cyclic block codes, which encode $k \geq (n - rt)$ information bits into $n = (2^r - 1)$ coded bits, so that the resultant codewords exhibit the maximum possible minimum Hamming distance. In 1960, Reed and Solomon conceived a non-binary version of BCH codes referred to as Reed-Solomon (RS) codes [53], while the following year Gorenstein and Zierler developed the Peterson-Gorenstein-Zierler (PGZ) decoding scheme for non-binary RS/BCH codes. Later, Berlekamp and Massey developed the Berlekamp-Massey decoding algorithm for cyclic RS/BCH codes in [56]–[59], while a soft-decision aided Chase decoder was proposed in [63]. Both these decoding algorithms are widely adopted for decoding BCH as well as RS codes. Unfortunately BCH codes did not find much practical applications, except as Cyclic Redundancy Check (CRC) codes in Automatic-Repeat-reQuest (ARQ) systems. By contrast, RS codes have found several practical applications owing to their inherent capability of correcting both random as well as burst of errors. Explicitly, RS codes are widely employed in magnetic tape and disk storage, which are susceptible to burst errors. Furthermore, they are also used as outer codes in concatenated coding schemes, which have been integrated in various standardized systems, such as the deep-space coding standard [150]. Another major milestone in algebraic coding was achieved with the development of non-binary Redundant Residue Number System (RRNS) codes [60], [61], which are also maximum minimum-distance codes and hence exhibit similar distance properties to RS codes.

By 1980, error correction codes were successfully deployed in various deep-space, satellite and mobile communications systems in conjunction with modulation schemes. However, the error correction and modulation modules were treated independently and the redundancy of the codes extended the bandwidth requirement, when the signal constellation size was fixed. For the sake of circumventing this disadvantage of coding, Ungerboeck invented a bandwidth-efficient trellis-based joint coding and modulation scheme called Trellis-Coded Modulation (TCM) [66]–[68]. Explicitly, TCM is a joint channel coding and modulation scheme, which absorbs the redundant coding bits by expanding the constellation size to accommodate more bits/symbols and hence maintains a fixed bandwidth. TCM provides attractive performance gains over convolutional codes, while incurring a similar decoding complexity. In 1992, another coded modulation scheme termed as Bit-Interleaved Coded Modulation (BICM) [71], [72] was conceived for transmission over fading channels, which invoked bit-based interleavers in conjunction with Gray-coded bit-to-symbol mapping. More specifically, parallel bit interleavers are used at the output of a convolutional code in this joint coding and modulation scheme for the sake of increasing the resultant diversity gain by exploiting the fading of the bits in a multi-bit symbol; hence enhancing the system's performance over fading channels. However, BICM does not outperform TCM over AWGN channels, since it exhibits a reduced minimum Euclidean distance.

Despite being into the fifth decade of coding theory, the notion of operating near the Shannon limit was far from realization until Berrou *et al.* [73], [74] conceived turbo codes in 1993. More specifically, turbo code rely on a parallel concatenation of Recursive Systematic Convolutional (RSC) codes with an interleaver between them. At the decoder, soft iterative decoding is invoked, which relies on the Soft-In Soft-Out (SISO) MAP algorithm of [64]. It is pertinent to mention here that the MAP algorithm only slightly outperforms the VA in terms of the achievable BER for non-iteratively decoded convolutional codes, while imposing a substantially higher complexity. Consequently, MAP decoding was rarely used for decoding convolutional codes, until turbo codes were invented. But given that turbo decoders require bit-by-bit soft-metrics, they required complex MAP decoding. Fortunately, the complexity of turbo decoders may be reduced by invoking less complex SISO decoders, for example the Soft-Output Viterbi Algorithm (SOVA) [69], the Max-Log-MAP algorithm [70] and the Log-MAP algorithm [77].

Berrou's turbo revolution triggered intensive research efforts directed towards designing iterative 'turbo-like' codes. In particular, it led to the renaissance of Low Density Parity Check (LDPC) codes in 1995 [78], [79]. LDPC codes were proposed by Gallager as early as 1962 [55]. However, the associated complexity was deemed enormous in that era. Consequently, LDPC codes were abandoned for decades to come. However, the invention of turbo codes revived the research interest in LDPC codes. Various variants of LDPC codes have been proposed over the years, which are known to operate arbitrarily close to the Shannon limit at sufficiently long codeword lengths, for example irregular LDPC codes [89], [90], LDPC convolutional codes [104], protograph-based LDPC codes [113] and spatially coupled LDPC codes [129].

Turbo revolution also led to other iterative coding schemes, which include for example Turbo BCH codes [81], Turbo Hamming codes [91], BICM with Iterative Decoding (BICM-ID [92], Turbo Trellis Coded Modulation (TTCM) [103], punctured turbo codes [105] and Unity Rate Code (URC) assisted concatenated coding schemes [108]. The invention of

EXtrinsic Information Transfer (EXIT) charts [109], [115] by Ten Brink in 2001 marks another important milestone in the realm of the afore-mentioned concatenated schemes relying on iterative decoding. More specifically, EXIT charts constitute a semi-analytical tool, which aids the design of near-capacity iterative schemes [142], [151]. Quantitatively, the resultant systems may operate within 1 dB of the Shannon limit, see for example the IRregular Convolutional Code (IRCC) assisted concatenated schemes of [111], the TTCM of [119] and the BICM-ID of [126].

With the help of intensive research efforts, turbo coding was successfully commercialized within just a few years and was incorporated into various standardized systems, such as mobile communication systems and video broadcast systems [142]. In particular, turbo coding was incorporated in the 3G UMTS [152] and 4G LTE [153] mobile standards. However, the high latency associated with turbo codes is anticipated to be a major impediment in next-generation systems supporting 'tactile services'. Consequently, a Fully-Parallel Turbo Decoder (FPTD) was recently conceived by Maunder in [137], which significantly reduces the associated latency; hence making turbo codes a promising candidate for next-generation systems. Over the years, the LDPC coding scheme has proved to be a fierce competitor of turbo codes, which has also been adopted by various standards, for example WiMax, IEEE 802.11n, IEEE 802.3an, and DVB-S2.

Arikan's polar code [125] conceived in 2009 sparked another wave of excitement within the coding community, since it is the first class of channel codes, which provably achieves the capacity of symmetric memoryless channels, while imposing only a modest encoding and decoding complexity. Polar codes invoke a short and simple kernel code, so that the physical channels are polarized into virtual channels, which are either perfectly noiseless or completely random, provided that the block length is sufficiently long. At practical block lengths, the channels are polarized into a set of high-reliability and low-reliability virtual channels. Finally, the information bits are sent across the high-reliability channels, while dummy bits, called 'frozen bits', are transmitted via the low-reliability channels. If the block lengths are sufficiently long, then the fraction of high reliability virtual channels is equivalent to the achievable channel capacity. At the receiver, the polar decoder invokes a low-complexity successive cancellation decoding algorithm, which processes the received bits serially. Despite having a low encoding and decoding complexity, Polar codes, relying on cyclic redundancy check-aided successive cancellation list decoding, are capable of outperforming the standardized LTE turbo and WiMax LDPC codes at moderate block lengths, as demonstrated in [154]. Furthermore, the coding rate of polar codes can be varied almost continuously by changing the number of frozen bits, hence making them ideal for rate-compatible scenarios. However, a major limitation of polar codes is the high latency associated with the polar decoder, since it sequentially processes the received information. Nonetheless, polar codes have already found their way into the 5G system for enhanced mobile broadband communications, where polar codes and LDPC codes have been chosen for the control and data channels, respectively.

To conclude, classical turbo, LDPC and polar codes have made it possible to operate arbitrarily close to the Shannon limit of Fig. 9. For example, the 1/2-rate turbo code of [73] operates within 0.7 dB of the Shannon limit at a block length of 65,536 bits, while the 1/2-rate irregular LDPC code of [90] surpasses the performance of comparable turbo codes and operates only 0.13 dB away from Shannon capacity at a block length of 10^6 bits. Furthermore, polar codes [125] provably achieve the capacity, albeit at infinitely long block lengths. Hence, our ambition to reach the Shannon limit in turn resulted in long decoding delays, which motivated the research on parallel decoding architectures, for example on the fully-parallel LDPC and turbo decoders of [137] and [155], respectively. The decoding latencies associated with polar codes are even higher due to the serial nature of the polar decoder. Hence, it seems that the research community first designed practically infeasible codes in the spirit of reaching the Shannon limit and then changed the ultimate goal to that of reducing the decoding delays. Therefore, it remains an open challenge to design codes, which strike exactly the desired design trade-offs amongst all the parameters of Fig. 10. Explicitly, we need a code, which maximizes the coding rate for the given channel conditions, while minimizing the achievable BER, system bandwidth, delay and implementation complexity. It is also desirable that the code should be rate-compatible, hence capable of operating in diverse use-cases under diverse channel conditions. This is particularly important in the context of the on-going debates concerning the 5G systems promising seamless connectivity for diverse use-cases.

B. Quantum Coding Theory

1) Design Objectives: With around seven decades of rich history, classical coding theory is already quite mature. By contrast, quantum coding theory is still in its infancy, since the implementation of quantum technology has not been commercialized. Researchers have been actively working on discovering the quantum versions of the existing classical codes. In duality to the classical coding theory, QECCs are designed to achieve the quantum channel capacity [93], [156], [157], or more precisely the hashing bound. Explicitly, the hashing bound is only a lower bound, because the actual capacity of a quantum channel may be higher due to the 'degenerate' nature of quantum codes [158], [159]. To elaborate further, the notion of degeneracy implies that different error patterns may yield the same corrupted quantum state. For instance, let us consider the state $|\psi\rangle = |00\rangle + |11\rangle$, which may experience the channel-induced error **IZ** or **ZI**. We may observe that both these error patterns result in the same channel output, i.e., $(|00\rangle - |11\rangle)$. Consequently, the error patterns **IZ** and **ZI** are classified as degenerate errors, as further discussed in Section V. Similarly, the error pattern **ZZ** leaves the state $|\psi\rangle$ intact analogous to the error-free scenario; hence **ZZ** and **II** are also degenerate errors.

In duality to the Shannon limit of Eq. (25), the hashing bound is completely specified by the channel's depolarizing

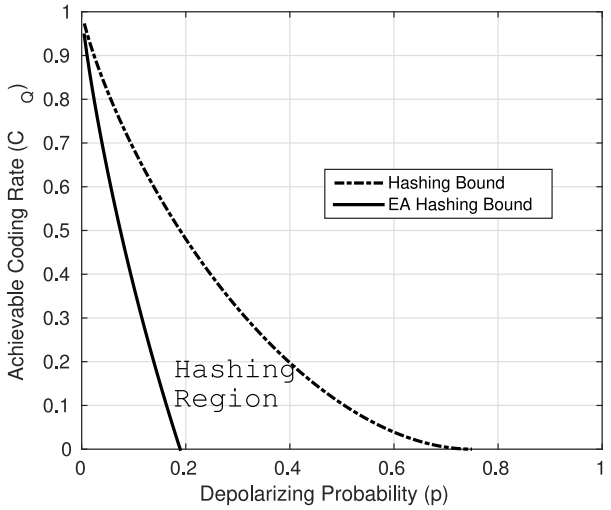


Fig. 12. Hashing bounds for the unassisted ($c = 0$) and maximally-entangled ($c = n - k$) quantum codes, characterized by Eq. (27) and Eq. (29), respectively. The enclosed region, labeled the 'hashing region', quantifies the capacity for $0 < c < (n - k)$.

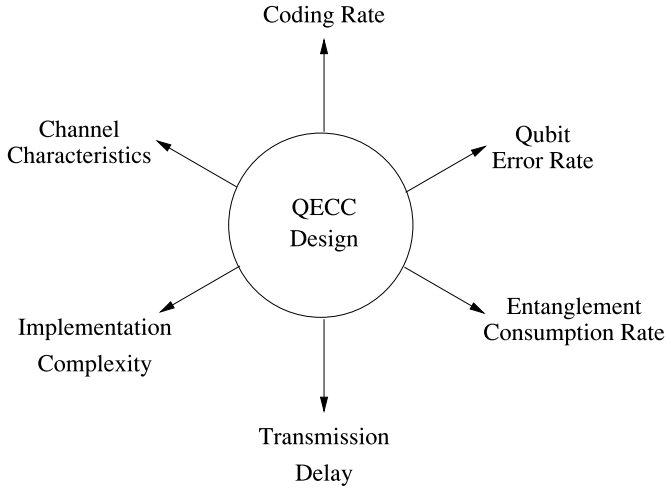


Fig. 13. Stylized representation of conflicting design parameters affecting the design of quantum codes.

probability p as follows [85], [131]:

$$C_Q(p) = 1 - H_2(p) - p\log_2(3), \tag{27}$$

where $H_2(p)$ denotes the binary entropy function. Explicitly, a random quantum code $\mathcal{C}$ may exhibit an arbitrarily low Quantum Bit Error Ratio (QBER) at a depolarizing probability of p, if its coding rate does not exceed the hashing limit $C_Q(p)$ of Eq. (27) and the codeword has a sufficiently long length.

The Hashing bound of Eq. (27) is only valid for unassisted quantum codes. Explicitly, there exists a family of Entanglement-Assisted (EA) quantum codes [112], [116]–[118], which does not exist in the classical domain. In contrast to the unassisted quantum codes, the EA quantum codes rely on pre-shared noiseless entangled qubits, which naturally increases the achievable capacity. Given that c entangled qubits are pre-shared with the receiver over a noiseless channel, the associated EA hashing bound is given by [131], [160]:

$$C_Q(p) = 1 - H_2(p) - p\log_2(3) + \mathrm{E}, \tag{28}$$

where E denotes the 'entanglement consumption' rate, which is equivalent to $\mathrm{E} = \frac{c}{n}$ for a code having k information qubits, n coded qubits and $0 \leq c \leq (n - k)$ pre-shared qubits. Explicitly, when $c = 0$, Eq. (28) reduces to the unassisted hashing bound of Eq. (27). By contrast, when c has the maximum value of $(n - k)$, we get the maximally-entangled quantum codes and the associated maximally-entangled hashing bound is [131], [160]:

$$C_Q(p) = 1 - \frac{H_2(p) - p\log_2(3)}{2}. \tag{29}$$

Hence, as shown in Fig. 12, an EA quantum code can operate anywhere in the hashing region, which is bounded by Eq. (27) and Eq. (29). Furthermore, in duality to Fig. 10, the parameters involved in the design of QECCs are illustrated in Fig. 13.

In duality to the classical coding bounds of Table II, Table III enlists the quantum coding bounds, which characterize the rate-versus-minimum-distance trade-off for quantum codes. Analogous to the classical coding bounds, the quantum Singleton bound serves as a loose upper bound, the quantum Hamming bound as a tighter upper bound, and the quantum GV bound as the tightest lower bound. Furthermore, Ashikhmin and Litsyn extended the classical linear programming approach to quantum codes using the MacWilliams identities [163] for the sake of tightening the quantum Hamming bound. However, despite all efforts, a wide gap existed between the upper and lower coding bounds until Chandra *et al.* conceived a closed-form expression [148] for characterizing the rate-versus-minimum-distance trade-off for quantum codes. As demonstrated in Fig. 14, the closed-form formulation of [148] satisfies all the known coding bounds. Fig. 15 portrays the growth of achievable minimum distance upon increasing the codeword length based on the finite-length closed-form formulation of [148]. We may observe in Fig. 15 that the minimum distance increases almost linearly with the codeword length, hence it is termed as the 'unbounded minimum distance'. Consequently, it is desirable to conceive code structures having an unbounded minimum distance.

2) Error Correction Codes: The rate-1/3 repetition code is the simplest single-error correcting code in the classical coding paradigm, which relies on the cloning of information bits. Unfortunately, qubits cannot be cloned owing to the existence of the no-cloning theorem. Hence, it was generally believed that QECCs are infeasible, until Shor pioneered the first quantum code in 1995 [80]. Shor's code of [80] is a rate-1/9 code capable of correcting a single bit-flip, phase-flip as well as bit-and-phase-flip error. Motivated by this breakthrough, Calderbank and Shor [83] as well as Steane [82], [84] independently conceived a generalized framework for constructing quantum codes from classical binary linear codes, which constitutes the popular family of Calderbank-Shor-Steane (CSS) codes. Explicitly, the CSS construction relies

TABLE III
RATE-VERSUS-MINIMUM-DISTANCE BOUNDS FOR QUANTUM CODES [148]

Quantum Coding Bound	Finite-Length	Asymptotic	Notes
Singleton [161]	$\frac{k}{n} \leq 1 - 2\left(\frac{d_{\min}-1}{n}\right)$	$\frac{k}{n} \leq 1 - 2\left(\frac{d_{\min}}{n}\right)$	very loose upper bound
Hamming [162]	$\frac{k}{n} \leq 1 - \frac{1}{n}\log_2\left(\sum_{j=0}^{t=\lfloor\frac{d_{\min}-1}{2}\rfloor}\binom{n}{j}3^j\right)$	$\frac{k}{n} \leq 1 - \left(\frac{d_{\min}}{2n}\right)\log_2 3 - H_2\left(\frac{d_{\min}}{2n}\right)$	tight upper bound for moderate and high coding rate
Linear Programming [163]		$\frac{k}{n} \leq H_2(\tau) + \tau\log_2 3 - 1$ $\tau = \frac{3}{4} - \frac{1}{2}\frac{d_{\min}}{n} - \frac{1}{2}\sqrt{3\frac{d_{\min}}{n}\left(1-\frac{d_{\min}}{n}\right)}$	strengthen the upper bound
Gilbert-Varshamov (GV) [162]	$\frac{k}{n} \geq 1 - \frac{1}{n}\log_2\left(\sum_{j=0}^{d_{\min}-1}\binom{n}{j}3^j\right)$	$\frac{k}{n} \geq 1 - \left(\frac{d_{\min}}{n}\right)\log_2 3 - H_2\left(\frac{d_{\min}}{n}\right)$	tightest lower bound

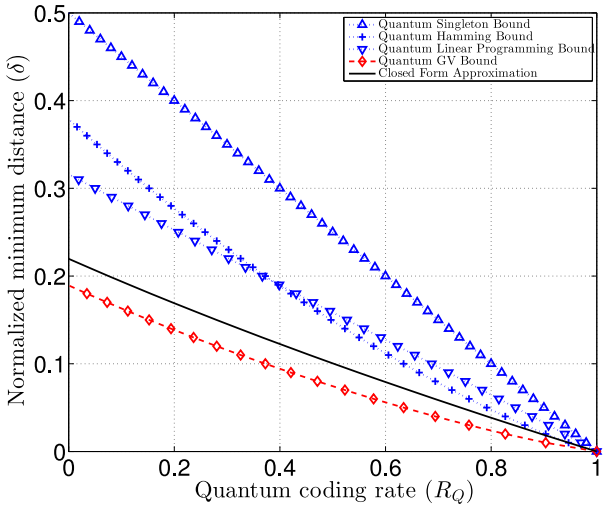


Fig. 14. Rate ($R_Q = k/n$) versus normalized minimum distance $\delta = \frac{d_{\min}}{n}$ asymptotic bounds [148]. The closed-form approximation of [148] is also plotted, which relies on a simple quadratic function $R_Q(\delta) = \frac{32}{9}\delta^2 - \frac{16}{3}\delta + 1$ and satisfies all the bounds. Upper bounds are plotted in blue, while the lower bound is plotted in red.

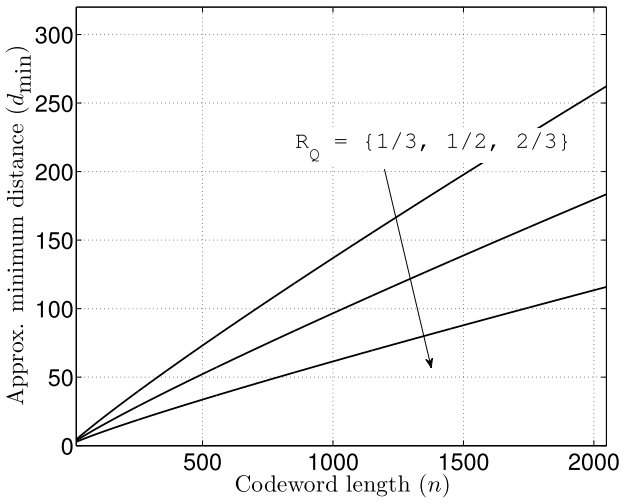


Fig. 15. The growth of achievable minimum distance with increasing codeword length based on the finite-length closed-form formulation of [148].

on a pair of classical binary linear block codes C_1 and C_2, which satisfy the criterion $C_2 \subset C_1$. Furthermore, a special class of CSS codes, called dual-containing CSS codes, was also introduced, which was derived from dual-containing binary codes. Explicitly, dual-containing CSS codes constitute a special type of CSS codes having $C_2 = C_1^{\perp}$, where $C_1^{\perp}$ is the dual code[7] of C_1. Following these principles, Steane [84] constructed a rate-1/7 single-error correcting code from the classical [7, 4, 3] Hamming code. In the spirit of further improving the coding rate, Bennett *et al.* [85] and Laflamme *et al.* [86] independently designed the optimal rate-1/5 single-error correcting quantum code, having the smallest possible codeword length.

The CSS construction of [82]–[84] does not exploit the redundant qubits efficiently, since the bit-flip and the phase-flip errors are corrected independently by concatenating a pair of classical binary codes. For the sake of designing an optimal code having the smallest codeword length, similar to the rate-1/5 code of [85] and [86], it is important to jointly correct bit-flip and phase-flip errors. In pursuit of designing such optimized codes, Gottesman established the theory of Quantum Stabilizer Codes (QSCs) [87] during his Ph.D. [88]. Explicitly, Gottesman presented a more general formalism, called stabilizer formalism, capable of facilitating the design of quantum codes from the classical binary and quaternary codes. As compared to the CSS codes, the stabilizer formalism imposes a more relaxed constraint, generally called the 'symplectic product' criterion, on the underlying classical codes; hence, the resultant QECCs can have either a CSS or a non-CSS (also called unrestricted) structure. In simple terms, the symplectic product criterion is the constraint imposed on the Parity Check Matrix (PCM) of the constituent classical code (or codes), which ensures that the resultant quantum code is a valid stabilizer code.[8] Furthermore, while the CSS-type codes independently correct bit-flip and phase-flip errors, the non-CSS codes jointly correct bit-flip and phase-flip errors. The advent of stabilizer formalism sparked a major revolution in the history of quantum coding, leading to the development of various code families, which includes Quantum Bose-Chaudhuri-Hocquenghem (QBCH) codes [94]–[99], toric codes [100]–[102], Quantum Reed-Muller codes [106], Quantum Reed-Solomon codes (QRS) [107], Quantum Low Density Parity Check (QLDPC) codes [110], [164]–[166],

[7]Let C be a classical linear block code having the generator matrix $\mathbf{G}$ and the PCM $\mathbf{H}$, then the dual code $C^{\perp}$ is the code having the generator matrix $\mathbf{H}^T$ and the PCM $\mathbf{G}^T$.

[8]Further details are given in Section VI.

Quantum Convolutional Codes (QCC) [114], [167]–[169], Quantum Turbo Codes (QTC) [120], [121], Quantum IRregular Convolutional Codes (QIRCC) [3] as well Quantum Unity Rate Codes (QURC) [139].

The Quantum research fraternity has invested the last three decades in designing the quantum counterparts of the existing families of classical codes. Except for the parallel concatenated codes as well as for the joint coding and modulation schemes of the classical regime, virtually all major families of classical codes have a quantum counterpart. Amongst these, short block codes are particularly important from an implementation perspective, since the quantum technology is still in its infancy and hence decoherence would prevent the implementation of long codes. However, the desire to approach the hashing bound of Fig. 13 motivated researchers to design QLDPC [110], [164]–[166] codes and QTCs [120], [121], which exploit iterative decoding. In particular, the sparse nature of LDPC matrix is particularly important in the quantum domain for achieving fault-tolerant decoding, since the qubits interact with only a limited number of other qubits during the syndrome computation process. Furthermore, since the LDPC matrix is sparse, the resultant QLDPC codes exhibit high degeneracy. However, the strict symplectic product criterion associated with the design of stabilizer codes severely limits the performance of QLDPC codes. More specifically, owing to the symplectic criterion, the QLDPC matrix consists of numerous short cycles, which have a length of 4. This in turn degrades the performance of the LDPC decoder relying on the message passing algorithm, as detailed in [124]. Unfortunately, the LDPC decoder is not capable of capturing the impact of degenerate errors. In fact, the LDPC decoder suffers from the so-called 'symmetric degeneracy error' [124], which results from the degenerate errors. For the sake of improving the performance of the LDPC decoder in the wake of length-4 cycles and the symmetric degeneracy error, Poulin *et al.* conceived heuristic methods in [122]. These methods primarily relied on introducing random perturbations for triggering decoding convergence. Then the QLDPC decoding methods were further improved in [123] and [124]. Despite these developments, the performance of QLDPC codes is still not comparable to that of classical LDPC codes.

In 2008, Poulin *et al.* constructed the quantum counterparts of turbo codes in [120] and [121]. While classical turbo codes generally rely on the parallel concatenation of convolutional codes, the QTCs of [120] and [121] rely on the serial concatenation of QCCs. As compared to QLDPC codes, QTCs offer more flexible code parameters, for example the frame length, coding rate, constraint length as well as the interleaver type. Furthermore, the iterative decoding of QTCs takes into account the impact of degenerate errors. However, the stabilizer-based QCCs cannot be concurrently recursive as well as noncatastrophic[9] [120], [121], [170]. Both these properties are essential for constructing good turbo codes. Explicitly, a recursive inner code is required for achieving an unbounded minimum distance, while both component codes of a serially concatenated code must be noncatastrophic for ensuring decoding convergence to an infinitesimally low error rate. Hence, the QTCs of [120] and [121] exhibit a bounded minimum distance, since they rely on non-recursive non-catastrophic QCCs. For the sake of designing near-capacity QTCs, Babar *et al.* [136] developed EXIT charts for the quantum domain, while a Quantum IrRegular Convolutional Code (QIRCC) structure and Quantum Unity Rate Code (QURC) were proposed in [3] and [139], respectively. Recently, a Fully-Parallel Quantum Turbo Decoder (FPQTD) was conceived in [140], which substantially reduces the decoding latency.

Recall that stabilizer codes must satisfy the stringent symplectic product criterion. Consequently, not every classical code can be 'imported' into the quantum realm. Furthermore, the symplectic product criterion results in undesired code characteristics, for example the unavoidable length-4 cycles of QLDPC codes and the non-recursive nature of non-catastrophic QCCs. For the sake of overcoming the issues associated with the symplectic product criterion, the theory of EA quantum codes was developed in [112] and [116]–[118], which relies on the pre-sharing of entanglement between the transmitter and the receiver. The notion of EA codes was adopted for nearly all coding families, including EA-QLDPC codes [127], EA-QCCs [128] and EA-QTCs [130], [131], hence alleviating the issues arising from the symplectic product criterion. Explicitly, EA-QLDPC codes may be designed with no length-4 cycles in the binary formalism. Consequently, the resultant performance is comparable to that of the classical LDPC codes. Similarly, EA-QCCs can be concurrently recursive as well as non-catastrophic [130], [131]. Consequently, EA-QTCs are capable of having an unbounded minimum distance. Hence, the family of EA quantum codes finally brought the performance of quantum codes in line with that of their classical counterparts.

Polar codes have also attracted considerable attention within the quantum research fraternity. Inspired by the provably capacity achieving nature of Arikan's polar codes as well as their efficient encoding and decoding structures, Wilde and Guha demonstrated the existence of the quantum channel polarization phenomenon for classical and quantum information in [132] and [133], respectively. The quantum polar codes of [132] and [133] invoked a quantum-domain successive cancellation decoder, which is based on the notion of quantum hypothesis testing. The resultant decoder failed to match the decoding complexity of Arikan's successive cancellation decoder. This issue was addressed by Renes *et al.* [134], where CSS-type quantum polar codes were constructed from the classical polar codes, resulting in quantum codes having efficient encoders as well as decoders. However, the quantum polar codes of [132]–[134] rely on the sharing of noiseless entanglement between the transmitter and the receiver. In this context, the first unassisted quantum polar codes were recently conceived in [138], which marks another major milestone in the development of quantum codes.

[9] An encoder is catastrophic if it outputs a finite-weight coded sequence for an infinite-weight input sequence. Consequently, a catastrophic code may result in catastrophic error propagation, since a finite number of errors on the coded sequence may yield infinite number of errors on the decoded sequence. This in turn implies that the constituent codes of a concatenated code must be non-catastrophic for the sake of achieving decoding convergence.

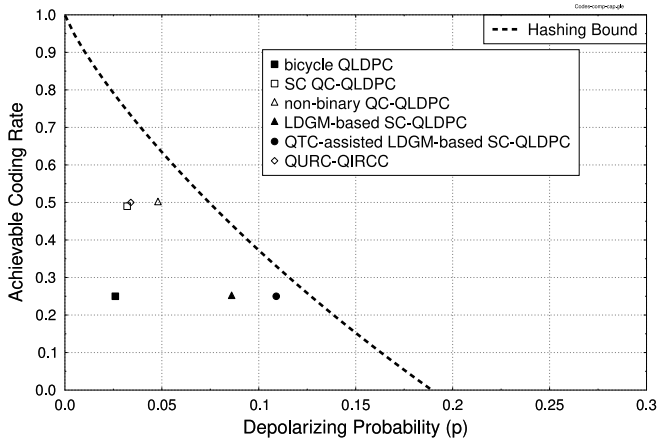


Fig. 16. Achievable performance at a word error rate (or frame error rate) of 10^{-3} benchmarked against the Hashing bound for the 'bicycle' code ($R = 0.25$, $n = 19{,}014$) of [164], 'SC QC-QLDPC' code ($R = 0.49$, $n = 1{,}81{,}000$) of [171], 'non-binary QC-QLDPC' code ($R = 0.5$, $n = 20{,}560$, GF(2^{10})) of [172] and [173], 'LDGM-based SC-QLDPC' code ($R = 0.25$, $n = 76{,}800$) of [174], 'QTC-assisted LDGM-based SC-QLDPC' code ($R = 0.25$, $n = 8{,}21{,}760$) of [175] and QURC-QIRCC code ($R = 0.5$, $n = 2{,}000$) of [139].

In a nutshell, similarly to classical coding, quantum coding research has also been steered towards approaching the capacity limit. In this pursuit, codes relying on long codeword lengths were designed, as exemplified by the bicycle QLDPC code ($R = 0.25$, $n = 19{,}014$) of [164], the Spatially-Coupled Quasi-Cyclic (SC QC) QLDPC code ($R = 0.49$, $n = 1{,}81{,}000$) of [171], the non-binary QC-QLDPC' code ($R = 0.5$, $n = 20{,}560$, GF(2^{10})) of [172] and [173], the Low Density Generator Matrix (LDGM)-based QLDPC code ($R = 0.25$, $n = 76{,}800$) of [174], the QTC-assisted LDGM-based SC-QLDPC code ($R = 0.25$, $n = 8{,}21{,}760$) of [175] and the concatenated QURC-QIRCC code ($R = 0.5$, $n = 2{,}000$) of [139], whose performance is benchmarked against the hashing bound in Fig. 16. Such long codeword lengths are particularly detrimental in the quantum domain, because of the short relaxation and dephasing times of qubits. Explicitly, if the codewords are very long, then the qubits may decohere faster than they can be corrected. Hence, quantum codes relying on short block lengths are highly desirable, at least until the relaxation and dephasing times of qubits become sufficiently increased, as quantum-hardware matures. Furthermore, in the quest for designing the quantum counterparts of the known classical codes, various EA schemes have been proposed, which impose the additional overhead of 'noiseless' pre-shared qubits. This overhead must be minimized for practical implementations. Overall, *it remains an open challenge to holistically optimize the design trade-offs depicted in Fig. 13*. It would be an extremely beneficial research objective to catalogue both the classical and quantum codes on the hypothetical pareto front of optimal solutions. Explicitly, the optimal pareto front is the collection of optimal solutions in the spirit of Fig. 13, where none of the metrics can be improved without degrading at least one of the other metrics. This research could commence with a low-complexity triple-parameter optimization, including the QBER (or BER for classical), coding rate and delay. Then it could be extended to the complexity and other relevant metrics in future research.

IV. Classical-to-Quantum Transition

The peculiar laws of quantum mechanics make quantum coding intrinsically different from their classical counterparts. Nevertheless, efficient quantum codes can be designed from the existing families of classical codes by cautiously addressing the following challenges, which do not exist in the classical realm.

1) *No-Cloning Theorem:* Most classical error correction codes rely on cloning. Explicitly, multiple copies of the information bits are transmitted for the sake of providing redundancy. Unfortunately, it is not possible to clone an arbitrary unknown qubit due to the no-cloning theorem [176].
2) *Measurement Operation:* Classical codes rely on measuring (or observing) the values of the received bits for hard-decision as well as soft-decision aided decoding. Unfortunately, it is not possible to measure (or observe) a qubit without perturbing it, which would result in the superimposed quantum states collapsing to the classical domain upon measurement.
3) *Nature of Quantum Errors:* Classical channels only impose bit-flip errors. By contrast, quantum channels inflict both bit-flips as well as phase-flip errors. Furthermore, quantum impairments are continuous in nature, since the received qubit may assume any value on the Bloch sphere.

In this Section, we elaborate on these challenges by designing the quantum counterparts of the simple rate-1/3 classical repetition code, which can only correct a single classical error. The overall evolution is summarized in Fig. 17 at a glance.

1) *No-Cloning Theorem:* Quantum codes exploit quantum-domain redundancy without cloning the information qubits.

The encoder of a 3-bit classical repetition code copies each information bit thrice. Explicitly, the information bits 0 and 1 are encoded as follows:

$$0 \rightarrow (000) \qquad 1 \rightarrow (111). \tag{30}$$

The encoding process of Eq. (30) does not have a quantum equivalent, because quantum information processing does not permit cloning. Let $\mathcal{U}$ be a hypothetical cloning (or copying) operation described as:

$$\mathcal{U}|\psi\rangle = |\psi\rangle \otimes |\psi\rangle. \tag{31}$$

Eq. (31) can be expanded as:

$$\begin{aligned}\mathcal{U}|\psi\rangle &= (\alpha|0\rangle + \beta|1\rangle) \otimes (\alpha|0\rangle + \beta|1\rangle) \\ &= \alpha^2|00\rangle + \alpha\beta|01\rangle + \alpha\beta|10\rangle + \beta^2|11\rangle. \end{aligned}\tag{32}$$

Alternatively, Eq. (31) can also be evaluated by considering the linearity of the cloning operator. Consequently, we have:

$$\begin{aligned}\mathcal{U}|\psi\rangle &= \mathcal{U}(\alpha|0\rangle + \beta|1\rangle) \\ &= \alpha\,\mathcal{U}|0\rangle + \beta\,\mathcal{U}|0\rangle \\ &= \alpha|00\rangle + \beta|11\rangle. \end{aligned}\tag{33}$$

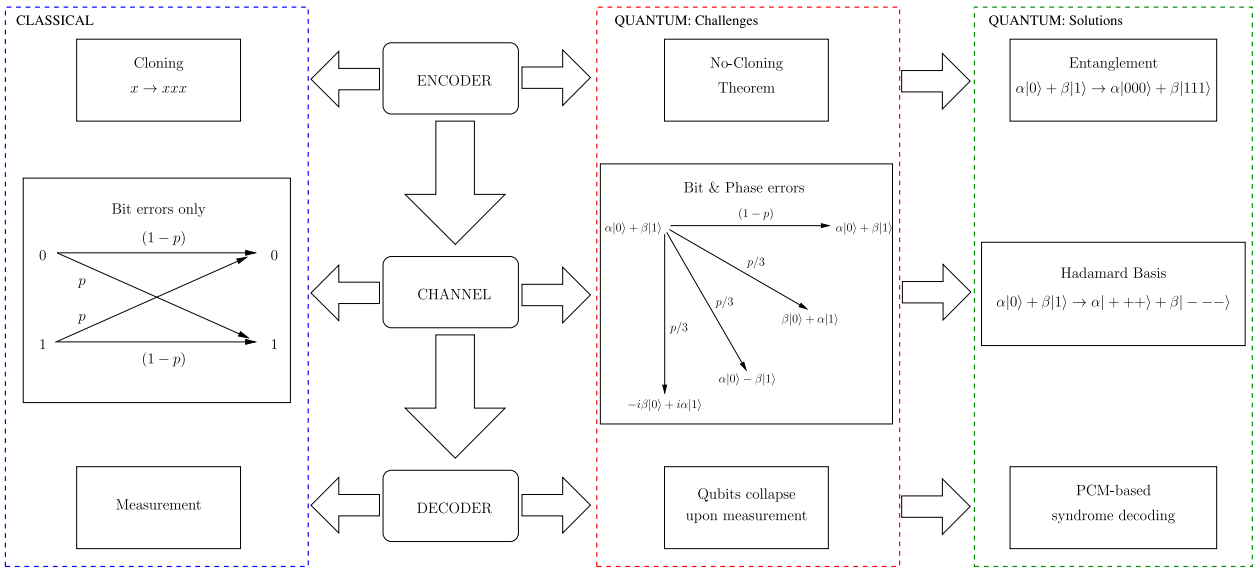


Fig. 17. Transition of error correction codes from the classical to the quantum domain [3]. **Encoder**: Classical encoders copy the information bits. Unfortunately, no quantum cloning operator exists. Consequently, quantum codes entangle the information qubits with the auxiliary qubits, so that the information is cloned in the basis states. **Channel**: Classical information may experience only bit-flip errors, while qubits may experience bit-flip as well as phase-flip errors. The additional phase-flip errors of the quantum domain may be corrected by using the Hadamard basis $\{|+\rangle, |-\rangle\}$. **Decoder**: Classical decoders measure the received bits for estimating the transmitted information. Unfortunately, qubits cannot be measured without perturbing their superimposed quantum state. As an alternate, quantum codes rely on the PCM-based syndrome decoding, hence estimating the channel-induced error patterns without measuring the received qubits.

It can be readily seen in Eq. (32) and Eq. (33) that:

$$\mathcal{U}(\alpha|0\rangle + \beta|1\rangle) \neq \alpha\,\mathcal{U}|0\rangle + \beta\,\mathcal{U}|0\rangle, \tag{34}$$

which violates the linearity of cloning operation. Hence, no cloning operator $\mathcal{U}$ exists in the quantum domain. Consequently, $|\psi\rangle$ cannot be encoded to $(|\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle)$. The 3-qubit bit-flip repetition code overcomes the cloning constraint by cloning the basis states rather than the state $|\psi\rangle$, i.e., the computational basis states $|0\rangle$ and $|1\rangle$ are encoded as follows:

$$\begin{aligned} |0\rangle &\rightarrow |\overline{0}\rangle \equiv |000\rangle, \\ |1\rangle &\rightarrow |\overline{1}\rangle \equiv |111\rangle. \end{aligned} \tag{35}$$

Explicitly, two auxiliary qubits in state $|0\rangle$ are *entangled* with the information qubit $|\psi\rangle$ with the aid of Controlled-NOT (CNOT) gates, as shown in the circuit of Fig. 18. CNOT represents a two-qubit gate, which takes as its input a control qubit and a target qubit. When the control qubit is in state $|1\rangle$, the target qubit is flipped; otherwise, the target qubit is left unchanged. More precisely, the output may be viewed as the reversible counterpart operation of a classical Exclusive OR (XOR) gate; hence, the CNOT gate may be deemed to represent a quantum counterpart of the classical XOR gate.[10] This can be mathematically expressed as:

$$\text{CNOT}(|\psi_0\rangle, |\psi_1\rangle) = |\psi_0\rangle \otimes |\psi_0 \oplus \psi_1\rangle, \tag{36}$$

[10]Please note that while the classical XOR gate's operation is irreversible, since two inputs are combined to yield a single XOR-ed output, a CNOT gate's operation is reversible, because we can reconstruct the two inputs (control and target) from the two outputs (control and target). In other words, CNOT is basically a reversible XOR gate in the classical domain.

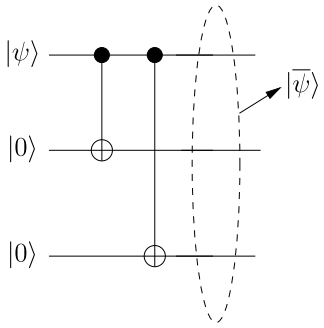


Fig. 18. Encoding circuit of 3-qubit bit-flip repetition code, where the information qubit $|\psi\rangle$ is encoded into $|\overline{\psi}\rangle$ with the help of two auxiliary qubits.

where $|\psi_0\rangle$ is the control qubit, while $|\psi_1\rangle$ is the target qubit. Consequently, the encoder of Fig. 18 replicates the computational basis states $|0\rangle$ and $|1\rangle$ three times in the encoded 3-qubit output $|\overline{\psi}\rangle$, which is given by:

$$\begin{aligned} |\psi\rangle \otimes |0\rangle^{\otimes 2} \rightarrow |\overline{\psi}\rangle &= \alpha|\overline{0}\rangle + \beta|\overline{1}\rangle \\ &\equiv \alpha|000\rangle + \beta|111\rangle. \end{aligned} \tag{37}$$

2) Measurement Operation: Quantum codes have to estimate the channel errors imposed without measuring (or observing) the received qubits.

At the receiver, the decoder of a 3-bit classical repetition code reads the received bits and decodes on the basis of majority voting. For example, the received codeword (011) is decoded to 1, while (100) is decoded to 0. This requires measuring (or observing) the received sequence, which is unfortunately not possible in the quantum domain. Explicitly,

TABLE IV
LOOK-UP TABLE FOR THE RATE-1/3 CLASSICAL REPETITION CODE

Syndrome s	Error e
(00)	(000)
(11)	(100)
(10)	(010)
(01)	(001)

if the received qubit $(\alpha|0\rangle + \beta|1\rangle)$ is measured in the computational basis, it will collapse to the states $|0\rangle$ and $|1\rangle$ with a probability of $|\alpha|^2$ and $|\beta|^2$, respectively.

Alternatively, an (n, k) classical linear block code can be decoded using an $(n-k) \times n$-element PCM $\mathbf{H}$, so that all error-free legitimate codewords $\overline{\mathbf{x}}$ yield:

$$\overline{\mathbf{x}}\mathbf{H}^T = 0. \tag{38}$$

Given a received codeword $\mathbf{y} = \overline{\mathbf{x}} + \mathbf{e}$, where $\mathbf{e}$ is the channel-induced error vector, the associated $(n-k)$-bit syndrome vector, which uniquely and unambiguously identifies the error vector (if the number of channel-induced errors is within the error correction capability of the code), is computed as:

$$\mathbf{s} = \mathbf{y}\mathbf{H}^T = (\overline{\mathbf{x}} + \mathbf{e})\mathbf{H}^T = \overline{\mathbf{x}}\mathbf{H}^T + \mathbf{e}\mathbf{H}^T = \mathbf{e}\mathbf{H}^T. \tag{39}$$

Hence, the syndrome can be used for estimating the error vector $\mathbf{e}$ using a pre-computed Look-Up Table (LUT). More explicitly, since an (n, k) linear block code has $(n-k)$ parity bits, we have $2^{(n-k)}$ unique syndromes. Consequently, we can estimate $2^{(n-k)}$ unique n-bit error patterns, which are pre-computed and stored in an LUT. Similarly, a 3-bit classical repetition code can also be decoded using the PCM-based syndrome decoding.[11] The associated PCM is given by:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \tag{40}$$

which yields a zero-valued syndrome vector for both valid codewords (111) and (000), while at least one of the two syndrome elements is 1 when a single bit-flip error is experienced. The resultant LUT is given in Table IV, which records all the single bit-flip errors that may be estimated with the help of a 3-bit classical repetition code. Intuitively, the first row of $\mathbf{H}$ compares the first two received bits of $\mathbf{y}$. If both bits are equal, the associated syndrome bit is 0, while if they are different, then the syndrome bit is 1. Similarly, the second row of $\mathbf{H}$ compares the first and third bit of $\mathbf{y}$.

Working along similar lines, a 3-qubit bit-flip repetition code can be decoded using a syndrome decoder, which simply compares the qubits without actually knowing their specific values. This is achieved by using two additional auxiliary qubits and the CNOT gates of Eq. (36), as shown in the 'Syndrome Processing' block of Fig. 19. Explicitly, it may be observed in Fig. 19 that the first auxiliary qubit is flipped, if the first two qubits are different, while the second auxiliary qubit is flipped, when the first and third qubits are different.

[11] In contrast to the conventional codeword decoding, which finds the most likely codeword, having the minimum Hamming distance, syndrome decoding finds the most likely error, having the minimum Hamming weight.

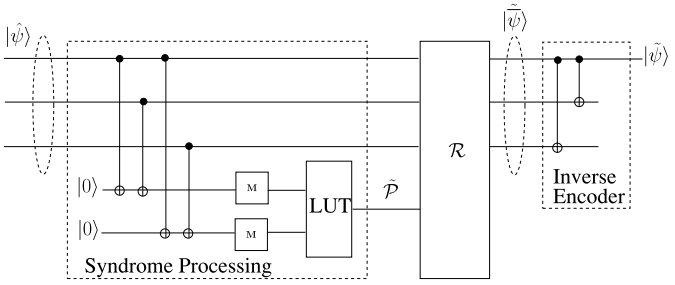


Fig. 19. Decoding circuit of 3-qubit bit-flip repetition code.

Explicitly, if $|\psi\rangle$ is transmitted, then we may receive one of the following four codewords $|\hat{\psi}\rangle$, assuming that only a single bit-flip is incurred during transmission:

$$\begin{aligned}
\alpha|000\rangle + \beta|111\rangle &\xrightarrow{\textbf{III}} \alpha|000\rangle + \beta|111\rangle, \\
\alpha|000\rangle + \beta|111\rangle &\xrightarrow{\textbf{XII}} \alpha|100\rangle + \beta|011\rangle, \\
\alpha|000\rangle + \beta|111\rangle &\xrightarrow{\textbf{IXI}} \alpha|010\rangle + \beta|101\rangle, \\
\alpha|000\rangle + \beta|111\rangle &\xrightarrow{\textbf{IIX}} \alpha|001\rangle + \beta|110\rangle.
\end{aligned} \tag{41}$$

The syndrome computation process operates on each of the possible received codeword $|\hat{\psi}\rangle$ as follows. Firstly, if both the first and second qubits as well as the first and third qubits remain identical, i.e., all three qubits remain identical, as in the case of error vector **III**, the auxiliary qubits remain unaltered:

$$\begin{aligned}
\alpha|000\rangle + \beta|111\rangle \otimes |0\rangle^{\otimes 2} &\rightarrow \alpha|00000\rangle + \beta|11111\rangle \\
&= (\alpha|000\rangle + \beta|111\rangle)|00\rangle.
\end{aligned} \tag{42}$$

Secondly, when both the first and second qubits as well as the first and third qubits are different, as in the case of error vector **XII**, both auxiliary qubits are flipped:

$$\begin{aligned}
\alpha|100\rangle + \beta|011\rangle \otimes |0\rangle^{\otimes 2} &\rightarrow \alpha|10011\rangle + \beta|01111\rangle \\
&\equiv (\alpha|100\rangle + \beta|011\rangle)|11\rangle.
\end{aligned} \tag{43}$$

Thirdly, when the first and second qubits are different, but the first and third qubits are identical, as in the case of error vector **IXI**, only the first auxiliary qubit is flipped.

$$\begin{aligned}
\alpha|010\rangle + \beta|101\rangle \otimes |0\rangle^{\otimes 2} &\rightarrow \alpha|01010\rangle + \beta|10110\rangle \\
&= (\alpha|010\rangle + \beta|101\rangle)|10\rangle.
\end{aligned} \tag{44}$$

Finally, when the first and second qubits are identical, but the first and third qubits are different, as in the case of error vector **IIX**, only the second auxiliary qubit is flipped.

$$\begin{aligned}
\alpha|001\rangle + \beta|110\rangle \otimes |0\rangle^{\otimes 2} &\rightarrow \alpha|00101\rangle + \beta|11001\rangle \\
&= (\alpha|001\rangle + \beta|110\rangle)|01\rangle.
\end{aligned} \tag{45}$$

Then the auxiliary qubits of Eq. (42) Eq. (45) are measured in the block M of Fig. 19 to yield the classical syndrome $\mathbf{s}$, which can take one of the four possible values, i.e., 00, 11, 10 and 01. The syndrome $\mathbf{s}$ can then be used for estimating the error $\tilde{\mathcal{P}}$ using the LUT of Fig. 19 seen in Table IV. Thereafter, the transmitted codeword is recovered by applying the recovery operation $\mathcal{R}$ of Fig. 19, which aims for correcting the channel-induced flips based on the estimated error $\tilde{\mathcal{P}}$. Explicitly, in the context of the 3-qubit bit-flip repetition code, Pauli-**X** gates are applied during the recovery

process for counteracting the impact of the estimated channel error patterns of Table IV. Finally, the estimated information word $|\tilde{\psi}\rangle$ is retrieved by feeding the recovered codeword $|\tilde{\overline{\psi}}\rangle$ to the inverse encoder circuit, which is the same as that in Fig. 18, but operates from right to left, hence mapping the recovered encoded qubits onto the information qubits. It is pertinent to mention here that a classical repetition code is systematic in nature. Consequently, the information bit can be extracted from the received codeword without invoking an inverse encoding operation. By contrast, the information qubit of a quantum repetition code is entangled with auxiliary qubits and hence cannot be separated without an inverse encoder. For example, if $|\tilde{\overline{\psi}}\rangle = \alpha|000\rangle + \beta|111\rangle$, then applying the two CNOT gates of the inverse encoder of Fig. 18 yields:

$$\begin{aligned}\alpha|000\rangle + \beta|100\rangle &= (\alpha|0\rangle + \beta|1\rangle)|00\rangle \\ &\equiv |\tilde{\psi}\rangle|00\rangle, \end{aligned} \tag{46}$$

hence separating the information qubit $|\tilde{\psi}\rangle$ from the auxiliary qubits $|00\rangle$.

3) Nature of Quantum Errors: Quantum codes correct quantum bit-flip, phase-flip as well as bit-and-phase-flip errors.

When the classical coded bits (000) or (111) are transmitted, a 0 may be flipped to a 1 and a 1 may be flipped to a 0. Consequently, only discrete bit-flip errors are imposed on the transmitted codewords. By contrast, when a qubit is transmitted over the depolarizing channel of Section II-C, it may experience bit-flip, phase-flip as well as bit-and-phase flip errors, as discussed in Section II. A 3-qubit phase-flip repetition code may be designed analogous to the bit-flip repetition code, since phase-flips and bit-flips only differ in their basis of operation. More specifically, bit-flips flip the computational basis $\{|0\rangle, |1\rangle\}$, while phase-flips flip the Hadamard basis $\{|+\rangle, |-\rangle\}$ defined as:

$$\begin{aligned}|+\rangle \equiv \mathrm{H}|0\rangle &= \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \\ |-\rangle \equiv \mathrm{H}|1\rangle &= \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \end{aligned} \tag{47}$$

where H represents a Hadamard gate acting on a single qubit and specified by the matrix [2]:

$$\mathrm{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{48}$$

Therefore, a phase-flip (Pauli-**Z**) switches the Hadamard basis states as follows:

$$\begin{aligned}\mathbf{Z}|+\rangle &= |-\rangle, \\ \mathbf{Z}|-\rangle &= |+\rangle, \end{aligned} \tag{49}$$

while a bit-flip (Pauli-**X**) switches the computational basis, i.e., we have:

$$\begin{aligned}\mathbf{X}|0\rangle &= |1\rangle, \\ \mathbf{X}|1\rangle &= |0\rangle. \end{aligned} \tag{50}$$

Hence, a 3-qubit phase-flip repetition code protects against single phase-flip errors by replicating the Hadamard basis states rather than the information qubit as follows:

$$\begin{aligned}|0\rangle \rightarrow |\overline{0}\rangle &\equiv |++\rangle, \\ |1\rangle \rightarrow |\overline{1}\rangle &\equiv |--\rangle. \end{aligned} \tag{51}$$

This can be achieved by using the encoding circuit of Fig. 20, which entangles two auxiliary qubits with the information qubit $|\psi\rangle$ using CNOT and Hadamard gates. The circuit of Fig. 20 is similar to that of the bit-flip repetition code. However, it invokes additional Hadamard gates, which transform the computational basis to the Hadamard basis. Consequently, $|\psi\rangle$ is encoded as:

$$\begin{aligned}|\psi\rangle \otimes |0\rangle^{\otimes 2} \rightarrow |\overline{\psi}\rangle &= \alpha|\overline{0}\rangle + \beta|\overline{1}\rangle \\ &\equiv \alpha|++\rangle + \beta|--\rangle. \end{aligned} \tag{52}$$

Analogous to the 3-qubit bit-flip repetition decoder, the decoder of a 3-qubit phase-flip repetition code also uses two auxiliary qubits for computing the associated 2-bit syndromes. The first syndrome compares the phase of the first and second qubits, while the second syndrome compares the phase of the first and third qubits. This may be achieved using the decoding circuit of Fig. 21, which is the same as that of the 3-qubit bit-flip repetition code with the additional Hadamard gates invoked for transforming the Hadamard basis back to the computational basis. In other words, we may say that Hadamard gates are used at the input and output of the channel to transform the phase-flips to bit-flips. Hence, both bit-flip and phase-flip errors can be corrected by concatenating the 3-qubit phase-flip and bit-flip repetition codes, which actually constitutes the rate-1/9 Shor code [80] capable of correcting a single bit-flip, or phase-flip or alternatively a bit-and-phase-flip error. More specifically, the information qubit is first encoded

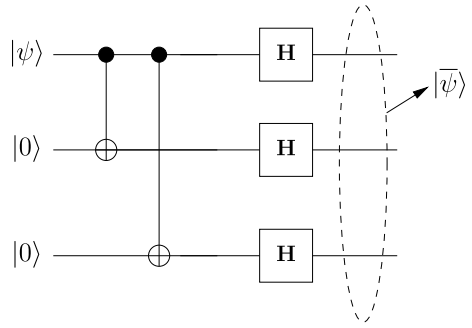


Fig. 20. Encoding circuit of 3-qubit phase-flip repetition code, where the information qubit $|\psi\rangle$ is encoded into $|\overline{\psi}\rangle$ with the help of two auxiliary qubits.

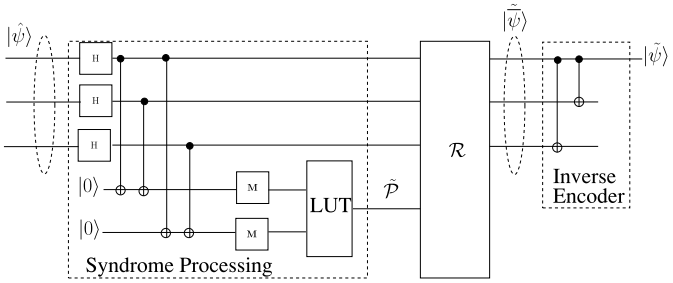


Fig. 21. Decoding circuit of 3-qubit phase-flip repetition code.

in Hadamard basis using the mapping of Eq. (52). The resultant three coded qubits are then independently encoded using the bit-flip repetition code of Eq. (37).[12] Hence, the basis states are mapped onto three 3-qubit blocks as follows:

$$\begin{aligned}|\overline{0}\rangle &\equiv \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \otimes \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \\ &\quad \otimes \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle), \\ |\overline{1}\rangle &\equiv \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) \otimes \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) \\ &\quad \otimes \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle), \end{aligned} \tag{53}$$

where the three qubits within a block are the codewords of a bit-flip repetition code, while the three blocks are the result of phase-flip repetition encoding. Taking the tensor product in Eq. (53) yields:

$$\begin{aligned}|\overline{0}\rangle &\equiv \frac{1}{\sqrt{8}}(|000000000\rangle + |000000111\rangle + |000111000\rangle \\ &\quad + |000111111\rangle + |111000000\rangle + |111000111\rangle \\ &\quad + |111111000\rangle + |111111111\rangle), \\ |\overline{1}\rangle &\equiv \frac{1}{\sqrt{8}}(|000000000\rangle - |000000111\rangle - |000111000\rangle \\ &\quad + |000111111\rangle - |111000000\rangle + |111000111\rangle \\ &\quad + |111111000\rangle - |111111111\rangle). \end{aligned} \tag{54}$$

Consequently, the encoded state $|\overline{\psi}\rangle$ is equivalent to:

$$\begin{aligned}\alpha|\overline{0}\rangle + \beta|\overline{1}\rangle \equiv \frac{1}{\sqrt{8}}&(\alpha+\beta)(|000000000\rangle + |000111111\rangle \\ &+ |111000111\rangle + |111111000\rangle) + \frac{1}{\sqrt{8}}(\alpha-\beta) \\ &\times (|000000111\rangle + |000111000\rangle + |111000000\rangle \\ &+ |111111111\rangle), \end{aligned} \tag{55}$$

which may be decoded by concatenating the decoding circuits of Fig. 19 and Fig. 21. Explicitly, the three 3-qubit blocks of Eq. (53) are first independently decoded using the 3-qubit bit-flip repetition decoder of Fig. 19, resulting in three information qubits, which constitute the received codeword for the 3-qubit phase-flip repetition decoder. Consequently, the resultant three qubits are decoded using the 3-qubit phase-flip repetition decoder of Fig. 21.

Furthermore, as encapsulated in Eq. (24), the received qubit may be in the superposition of all the possible errors. In essence, an (n, k) classical code, designed to protect a k-bit message by encoding it into an n-bit codeword, aims for restoring one of the 2^k valid codewords. By contrast, since a k-qubit information word is completely described by 2^k continuous-valued complex coefficients, quantum codes have to restore all the 2^k complex coefficients [164]. Fortunately, this continuous search space is reduced to a discrete one upon the measurement of the auxiliary qubits used for computing the syndrome. More specifically, although the 2^k coefficients are continuous-valued, some what serendipitously, the entire continuum of errors can be rectified, if the code is capable of correcting discrete bit-flip, phase-flip as well as bit-and-phase-flip errors acting on the constituent qubits. For example, let us assume that only a single bit-flip error may be inflicted during transmission. Then the received codeword of a 3-bit repetition code can be expressed as:

$$|\hat{\psi}\rangle = p_0\mathbf{III}|\overline{\psi}\rangle + p_1\mathbf{XII}|\overline{\psi}\rangle + p_2\mathbf{IXI}|\overline{\psi}\rangle + p_3\mathbf{IIX}|\overline{\psi}\rangle, \tag{56}$$

where p_0 is the probability of error-free transmission, while p_i is the probability of encountering a bit-flip error on the ith qubit. The syndrome computation process of Fig. 19 entangles two auxiliary qubits with $|\hat{\psi}\rangle$ of Eq. (56) as:

$$\begin{aligned}|\hat{\psi}\rangle \otimes |0\rangle^{\otimes 2} \rightarrow\ & p_0\big(\mathbf{III}|\overline{\psi}\rangle\big)|00\rangle + p_1\big(\mathbf{XII}|\overline{\psi}\rangle\big)|11\rangle \\ &+ p_2\big(\mathbf{IXI}|\overline{\psi}\rangle\big)|10\rangle + p_3\big(\mathbf{IIX}|\overline{\psi}\rangle\big)|01\rangle, \end{aligned} \tag{57}$$

which collapses to one of the four superimposed states when the auxiliary qubits are measured. The resultant state can then be corrected based on the specific syndrome observed.

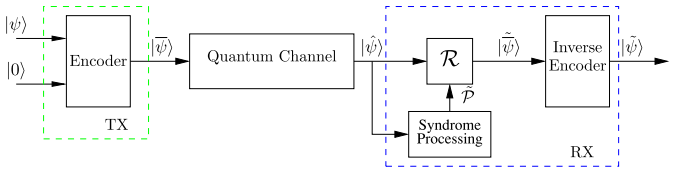


Fig. 22. Schematic of a quantum communication system invoking a quantum stabilizer code for error correction [124].

V. Stabilizer Formalism

The family of Quantum Stabilizer Codes (QSCs) rely on the same design principles as that of the repetition codes of Section IV. In particular, QSCs rely on the PCM-based syndrome decoding of classical linear block codes, hence, finding the channel-induced error by measuring the auxiliary syndrome qubits, rather than by observing the received qubits. Intuitively, the stabilizer formalism [87], [88] may be interpreted as the quantum-domain dual of the classical linear block coding paradigm. Furthermore, most classical codes exploit the same basic infrastructure as that of the classical linear block codes. Consequently, the stabilizer formalism provides a general theoretical framework for designing the quantum versions of the known classical codes. In Section V-A, we provide deeper insights into the duality of QSCs and classical linear block codes, while in Section V-B, we discuss the classification of error patterns for both the QSCs as well as the classical linear block codes.

A. Stabilizer-Based Code Design

Fig. 22 shows the system model of a quantum communication system relying on a QSC. A classical linear block code C (n, k) encodes k-bit information word x into an n-bit codeword

[12]The order of concatenation is very important. If the order of concatenation is reversed, i.e., if we invoke a bit-flip repetition code followed by a phase-flip repetition code, then the resultant quantum code encodes the basis states into:

$$\begin{aligned}|\overline{0}\rangle &\rightarrow |+++\rangle \otimes |+++\rangle \otimes |+++\rangle, \\ |\overline{1}\rangle &\rightarrow |---\rangle \otimes |---\rangle \otimes |---\rangle, \end{aligned}$$

which constitutes a strong rate-1/9 phase-flip repetition code, but it is not capable of correcting bit-flip errors.

$\overline{x}$ with the aid of $(n-k)$ parity bits $\mathbf{0}^{n-k}$ (initialized to zeros) as follows:

$$C = \{\overline{x} = \left(x{:}\mathbf{0}^{n-k}\right)\mathbf{V}\}, \tag{58}$$

where $\mathbf{V}$ is an invertible encoding matrix of size $(n \times n)$. Similarly, a QSC $\mathcal{C}[n,k]$[13] encodes a k-qubit information word (logical qubits) $|\psi\rangle$ into an n-qubit codeword (physical qubits) $|\overline{\psi}\rangle$ with the help of $(n-k)$ auxiliary qubits (also known as ancilla), as follows:

$$\mathcal{C} = \left\{|\overline{\psi}\rangle = \mathcal{V}(|\psi\rangle \otimes |\mathbf{0}_{n-k}\rangle)\right\}, \tag{59}$$

where $\mathcal{V}$ is an n-qubit unitary encoder. Explicitly, the auxiliary qubits of a QSC are analogous to the classical parity bits. The encoded qubits $|\overline{\psi}\rangle$ are transmitted over the quantum depolarizing channel of Section II-C, which imposes an n-qubit channel error vector $\mathcal{P}$. The erroneous channel output $|\hat{\psi}\rangle$ may then be expressed as:

$$|\hat{\psi}\rangle = \mathcal{P}|\overline{\psi}\rangle. \tag{60}$$

Similar to the decoders of the 3-qubit bit-flip and phase-flip repetition codes of Fig. 19 and Fig. 21, the decoder of a QSC invokes a 3-step process for correcting the transmission errors, which includes syndrome processing, error recovery ($\mathcal{R}$) and the inverse encoder.

Let us now revisit the 'syndrome processing' block of 3-qubit bit-flip repetition code from the perspective of the stabilizer formalism. Recall from Fig. 19 that we compute the first syndrome bit by comparing the first and second qubits in computational basis, while the second syndrome is obtained by comparing the first and third qubits. This is equivalent to measuring the eigenvalues[14] corresponding to the 3-qubit Pauli operators $g_1 = \mathbf{ZZI}$ and $g_2 = \mathbf{ZIZ}$, which are known as the stabilizer generators. Explicitly, Pauli-**Z** based stabilizer generators are used for comparing qubits in computational basis, because they are capable of detecting errors in the computational basis, i.e., bit-flip errors. If the qubits, which are being compared, are identical in computational basis, then the Pauli-**Z** based stabilizer generators yield an eigenvalue of $+1$, while if they are different, then the eigenvalue is -1. For example, if the received codeword is a valid one, implying that both the first and second qubits as well as the first and third qubits are identical as in Eq. (42), then we have:

$$\begin{aligned} g_1\left[|\overline{\psi}\rangle\right] &= \mathbf{ZZI}(\alpha|000\rangle + \beta|111\rangle) = |\overline{\psi}\rangle, \\ g_2\left[|\overline{\psi}\rangle\right] &= \mathbf{ZIZ}(\alpha|000\rangle + \beta|111\rangle) = |\overline{\psi}\rangle. \end{aligned} \tag{61}$$

Hence, the resultant eigenvalue is $+1$ for both g_1 as well as g_2, when a legitimate codeword is received. By contrast, if the corrupted codeword of $|\hat{\psi}\rangle = |100\rangle + \beta|011\rangle$ is received, implying that both the first and second qubits as well as the first and third qubits are different as in Eq. (43), then we have:

$$\begin{aligned} g_1\left[|\hat{\psi}\rangle\right] &= \mathbf{ZZI}(\alpha|100\rangle + \beta|011\rangle) \\ &= -\alpha|100\rangle - \beta|011\rangle = -|\hat{\psi}\rangle, \\ g_2\left[|\hat{\psi}\rangle\right] &= \mathbf{ZIZ}(\alpha|100\rangle + \beta|011\rangle) \\ &= -\alpha|100\rangle - \beta|011\rangle = -|\hat{\psi}\rangle, \end{aligned} \tag{62}$$

where both g_1 as well as g_2 yield an eigenvalue of -1. Recall from Eq. (38) that the PCM of a classical linear block code is designed so that it yields an all-zero syndrome vector for legitimate codewords, while yielding a non-zero syndrome vector for erroneous codewords, provided the number of channel-induced errors is within the error correction capability of the code. Similarly, the stabilizer generators of a QSC have to be designed, so that they yield an eigenvalue of $+1$ for legitimate codewords, while resulting in an eigenvalue of -1 for corrupted codewords. Hence, in duality to the PCM **H**, which completely specifies the codes space of a classical code C, the stabilizer generators define the code space a QSC. Furthermore, the complete stabilizer group $\mathcal{H}$ of a QSC consists of all the stabilizer generators and their products. For example, the stabilizer group $\mathcal{H}$ of the 3-qubit bit-flip repetition code consists of the independent generators g_1 and g_2 as well as the product of g_1 and g_2, i.e., **IZZ**.

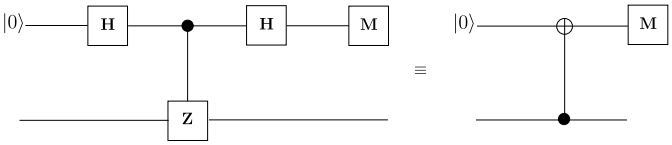


Fig. 23. Quantum circuit of measuring the **Z** operator acting on the bottom qubit [2] for bit-flip correction. The top qubit is the auxiliary qubit used for computing the syndrome. The circuit on the left is more popular, while the one on the right is more suitable for implementation.

TABLE V
SINGLE-QUBIT BIT-FLIP ERRORS TOGETHER WITH THE ASSOCIATED EIGENVALUES FOR THE 3-QUBIT BIT-FLIP REPETITION CODE HAVING $g_1 = \mathbf{ZZI}$ AND $g_2 = \mathbf{ZIZ}$

$\|\hat{\psi}\rangle = \mathcal{P}\|\overline{\psi}\rangle$	$g_1\|\hat{\psi}\rangle$	$g_2\|\hat{\psi}\rangle$	**Syndrome (s)**	$\hat{\mathcal{P}}$
$\alpha\|000\rangle + \beta\|111\rangle$	$+1$	$+1$	(00)	**III**
$\alpha\|100\rangle + \beta\|011\rangle$	-1	-1	(11)	**XII**
$\alpha\|010\rangle + \beta\|101\rangle$	-1	$+1$	(10)	**IXI**
$\alpha\|001\rangle + \beta\|110\rangle$	$+1$	-1	(01)	**IIX**

The $+1$ and -1 eigenvalues of Eq. (62) are mapped onto the classical syndromes 0 and 1, respectively, when the constituent **Z** operators are realized using the quantum circuit of Fig. 23, where the circuit on the left may be deemed more popular, while the one on the right is the equivalent circuit more suitable for implementation [2]. In both circuits of Fig. 23, the top qubit is the auxiliary qubit used for computing the syndrome, while the bottom qubit is the coded qubit subjected to the **Z** operator. The resultant syndromes are listed in Table V together with the associated single-qubit bit-flip errors, eigenvalues and the estimated error pattern $\hat{\mathcal{P}}$, which may be estimated using the syndrome decoding approach.

Analogous to the 3-qubit bit-flip repetition code, the codeword of a 3-qubit phase-flip repetition code is stabilized by the

[13]We consistently use round brackets (.) for classical codes, while the square brackets [.] are used for quantum codes.

[14]The eigenvector of a linear transformation **T** is a non-zero vector **v**, which only changes by a scaling factor when **T** is applied, i.e., $\mathbf{T}(\mathbf{v}) = \lambda\mathbf{v}$. The associated scaling factor λ is known as the eigenvalue.

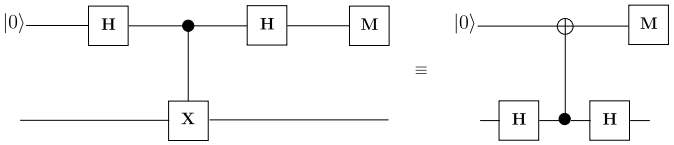


Fig. 24. Quantum circuit of measuring the **X** operator acting on the bottom qubit [2] for phase-flip correction. The top qubit is the auxiliary qubit used for computing the syndrome. The circuit on the left is the more usual conceptual construction, while the one on the right is more suitable for implementation.

generators $g_1 = \mathbf{XXI}$ and $g_2 = \mathbf{XIX}$. We may notice here that while Pauli-**Z** based stabilizer generators are invoked for bit-flip detection, Pauli-**X** based stabilizer generators are invoked for comparing qubits in the Hadamard basis, because they are capable of detecting errors in the Hadamard basis, i.e., phase-flip errors. The associated **X** operators can be implemented using the circuit of Fig. 24.

Recall from Section IV that Shor's codewords consist of three 3-qubit blocks, so that the three qubits within each block constitute the codeword of a 3-qubit bit-flip repetition code. Consequently, bit-flips may be detected by independently applying the stabilizer generators of the 3-qubit bit-flip repetition code to the three 3-qubit blocks, which is equivalent to comparing the three qubits within each block. This results in the following six stabilizer generators:

$$\begin{aligned} g_1 &= \mathbf{ZZIIIIIII}, \\ g_2 &= \mathbf{ZIZIIIIII}, \\ g_3 &= \mathbf{IIIZZIIII}, \\ g_4 &= \mathbf{IIIZIZIII}, \\ g_5 &= \mathbf{IIIIIZZI}, \\ g_6 &= \mathbf{IIIIIZIZ}, \end{aligned} \tag{63}$$

which helps in detecting single bit-flip errors occurring in each 3-qubit block. By contrast, phase-flip errors may be detected by comparing the blocks using Pauli-**X** operators. Explicitly, the phase information of a 3-qubit block is extracted by applying the **XXX** operator to the three qubits. For the 9-qubit Shor's code, which consists of three 3-qubit blocks, this may be implemented using the following two stabilizer generators:

$$\begin{aligned} g_7 &= \mathbf{XXXXXXIII}, \\ g_8 &= \mathbf{XXXIIIXXX}, \end{aligned} \tag{64}$$

where g_7 compares the phase of the first two blocks, while g_8 compares the phase of the first and third blocks.

Based on the above discussions, the 3-step decoding process of Fig. 22 may be generalized as follows:

1) *Syndrome Processing:* While the code space C of a classical linear block code is defined by a PCM **H** having $(n-k)$ independent rows, the associated code space $\mathcal{C}$ of a QSC is described by $(n-k)$ independent n-qubit Pauli operators g_i, for $1 \leq i \leq (n-k)$, which are generally termed as the stabilizer generators (or stabilizers in short). Explicitly, stabilizers are unique operators, which do not perturb the state of legitimate codewords, hence yielding an eigenvalue of $+1$. Furthermore, stabilizers yield an eigenvalue of -1 for corrupted codewords, provided the number of channel-induced errors is within the error correction capability of the stabilizer code. This is equivalent to the classical syndrome values of 0 and 1, respectively, which are the elements of the syndrome vector of Eq. (39). Alternatively, we may say that resulting eigenvalue is $+1$, when the channel-induced error $\mathcal{P}$ commutes with the stabilizer g_i, while it is -1, when the error anti-commutes with g_i. This can be mathematically encapsulated as:

$$g_i|\hat{\psi}\rangle = \begin{cases} |\overline{\psi}\rangle, & g_i\mathcal{P} = \mathcal{P}g_i \\ -|\overline{\psi}\rangle, & g_i\mathcal{P} = -\mathcal{P}g_i, \end{cases} \tag{65}$$

where $|\hat{\psi}\rangle = \mathcal{P}|\overline{\psi}\rangle$. The resultant eigenvalues can be mapped onto the classical error syndrome s by invoking the quantum circuits of Fig. 23 and Fig. 24. Hence, the set of stabilizers constitute the quantum counterpart of the classical PCM. However, the stabilizers must exhibit the additional commutativity property, which states that the stabilizers must be each other's commutative pairs. Explicitly, for a pair of stabilizers g_1 and g_2, we have:

$$g_1 g_2|\overline{\psi}\rangle = g_1|\overline{\psi}\rangle = |\overline{\psi}\rangle, \tag{66}$$

and similarly:

$$g_2 g_1|\overline{\psi}\rangle = g_2|\overline{\psi}\rangle = |\overline{\psi}\rangle. \tag{67}$$

Hence, the commutativity criterion naturally arises, which does not exist in the classical realm. Furthermore, the associated stabilizer group $\mathcal{H}$, which contains the $(n-k)$ stabilizers g_i as well as all the products of g_i, forms an Abelian subgroup of $\mathcal{G}_n$. The decoder of Fig. 22 processes the syndrome of the received sequence $|\hat{\psi}\rangle$ with the aid of the associated stabilizers, which are implemented using auxiliary qubits. Analogous to the decoders of the 3-qubit bit-flip and phase-flip repetition codes seen in Fig. 19 and Fig. 21, respectively, the auxiliary qubits collapse to classical syndromes upon measurement, hence mapping the eigenvalues of $+1$ and -1 onto the classical bits 0 and 1, respectively. The resultant classical syndrome bits are then fed to an LUT or to a classical PCM-based syndrome decoder for estimating the channel error vector $\tilde{\mathcal{P}}$ (discussed further in Section VI).

2) *Error Recovery ($\mathcal{R}$):* The error recovery block $\mathcal{R}$ of Fig. 22 recovers the potentially error-free codeword $|\tilde{\overline{\psi}}\rangle$ using the estimated error pattern $\tilde{\mathcal{P}}$. Naturally, if the number of errors exceeds the codes' error-correction capability, the recovery process becomes flawed. Hence, its flawed corrective action actually precipitates more errors than we originally had.
3) *Inverse Encoder:* Finally, the inverse encoder of Fig. 22 maps the recovered codeword $|\tilde{\overline{\psi}}\rangle$ onto the estimated transmitted information word $|\tilde{\psi}\rangle$. More specifically, while an encoder maps the information words onto the codewords, an inverse encoder works in the reverse

direction, hence mapping the codewords onto the information words.

Recall from Eq. (66) and Eq. (67) that the $(n-k)$ stabilizer generators g_i of a QSC always commute with each other. This implies that the constituent **X**, **Y** and **Z** operations must be selected so that all the resultant stabilizers commute. Explicitly, the non-Identity **X**, **Y** and **Z** operators intrinsically anti-commute with each other. For example, we have:

$$\mathbf{XY} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix} = i\mathbf{Z}, \tag{68}$$

while:

$$\mathbf{YX} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix} = -i\mathbf{Z}. \tag{69}$$

This implies that the operators **XY** and **YX** anti-commute, i.e., we have:

$$\mathbf{XY} = -\mathbf{YX}. \tag{70}$$

Similarly, we can readily show that:

$$\begin{aligned} &\mathbf{YZ} = i\mathbf{X},\ \mathbf{ZY} = -i\mathbf{X} \rightarrow \mathbf{YZ} = -\mathbf{ZY} \\ &\mathbf{ZX} = i\mathbf{Y},\ \mathbf{XZ} = -i\mathbf{Y} \rightarrow \mathbf{ZX} = -\mathbf{XZ}. \end{aligned} \tag{71}$$

Owing to this anti-commutative nature of non-Identity Pauli operators, the stabilizers have to be designed so that there are only an even number of indices having different non-Identity operators. For example, the 3-qubit Pauli operators **ZZI** and **XYZ** commute, because they consist of two indices having different non-Identity operators. By contrast, the operators **ZZI** and **YZI** anti-commute, since there is a single index, which has different non-identity operators.

B. Classification of Error Patterns

Based on the aforementioned discussions, we may conclude that the stabilizer generators play the same role in quantum error correction as the classical PCM **H** in classical error correction. Explicitly, analogous to the classical PCM, stabilizers yield syndrome bits, which in turn help in estimating the quantum channel errors. More specifically, the error set of a classical linear block code *C* having a PCM **H** can be classified as:

1) *Detected Error Patterns:* These error patterns yield a non-trivial syndrome, i.e., $e\mathbf{H}^T \neq 0$, which may be corrected by the code.
2) *Undetected Error Patterns:* This class of error patterns results in a trivial syndrome, i.e., $e\mathbf{H}^T = 0$, which cannot be detected by the code. More specifically, an undetected error maps the transmitted codeword onto another valid codeword. Since the resultant codeword still lies in the code space *C*, it does not trigger a non-zero syndrome. These undetected error patterns result from the limited minimum distance of the code.

Analogous to the classical detected error patterns, quantum-domain detected error patterns anti-commute with at least one of the stabilizer generators, which results in a non-trivial syndrome. Similarly, the quantum undetected error patterns commute with all the stabilizer generators, yielding an all-zero syndrome. This commuting set of error patterns is also known as the centralizer (or normalizer) of the stabilizer code having the stabilizer group $\mathcal{H}$, which is denoted as $C(\mathcal{H})$ (or $N(\mathcal{H})$). In particular, the centralizer of an $[n, k]$ QSC is a dual subspace consisting of *n*-tuple Pauli errors $\mathcal{P} \in \mathcal{G}_n$, which are orthogonal to all the stabilizers of the stabilizer group $\mathcal{H}$. Furthermore, since the $\mathcal{H}$ is itself an Abelian group consisting of mutually orthogonal generators, it is contained in the centralizer, i.e., we have $\mathcal{H} \subset N(\mathcal{H})$. Recall that the stabilizer generators do not modify the state of valid codewords. This in turn implies that errors which belong to the stabilizer group, i.e., we have $\mathcal{P} \in \mathcal{H}$, do not corrupt the transmitted codewords and therefore may be classified as harmless undetected error patterns. This class of errors does not have any classical counterpart. By contrast, those error patterns, which lie in the subspace $N(\mathcal{H}) \setminus \mathcal{H}$, are the harmful undetected errors, which map one valid codeword onto another. Hence, as depicted in Fig. 25, quantum error patterns can be classified as follows:

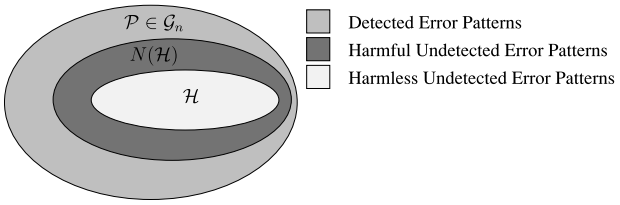


Fig. 25. Error pattern classification for stabilizer codes.

1) *Detected Errors Patterns:* These error patterns fall outside the normalizer subspace, i.e., they satisfy $\mathcal{P} \in \mathcal{G}_n \setminus N(\mathcal{H})$.
2) *Harmful Undetected Error Patterns:* This class of error patterns is defined as $N(\mathcal{S}) \setminus \mathcal{H}$.
3) *Harmless Undetected Errors Patterns:* These error patterns fall in the stabilizer group $\mathcal{H}$.

The class of harmless undetected error patterns makes quantum codes 'degenerate' [135]. More specifically, error patterns $\mathcal{P}$ and $\mathcal{P}' = g_i\mathcal{P}$ are said to be degenerate, because they differ only by the elements of the stabilizer group, which are harmless. Consequently, both $\mathcal{P}$ as well as $\mathcal{P}'$ yield the same output, as shown below:

$$\mathcal{P}'[|\overline{\psi}\rangle] = g_i\mathcal{P}[|\overline{\psi}\rangle] = \mathcal{P}g_i\left[|\overline{\psi}\rangle\right]. \tag{72}$$

Since $g_i[|\overline{\psi}\rangle] = |\overline{\psi}\rangle$, we get:

$$\mathcal{P}'[|\overline{\psi}\rangle] = \mathcal{P}[|\overline{\psi}\rangle]. \tag{73}$$

This in turn implies that degenerate error patterns can be rectified by the same recovery operation.

Let us consider the error patterns $\mathcal{P} = \mathbf{IIX}$ and $\mathcal{P}' = g_1\mathcal{P} = \mathbf{ZZX}$, where g_1 is the stabilizer of the 3-qubit bit-flip repetition code defined in Eq. (61). When these error patterns are applied to the legitimate codeword of Eq. (37), we get:

$$\begin{aligned} \mathbf{IIX}[\alpha|000\rangle + \beta|111\rangle] &= \alpha|001\rangle + \beta|110\rangle, \\ \mathbf{ZZX}[\alpha|000\rangle + \beta|111\rangle] &= \alpha|001\rangle + \beta|110\rangle. \end{aligned} \tag{74}$$

Hence, $\mathcal{P}$ and $\mathcal{P}'$ are degenerate errors, since both error patterns yield the same corrupted codeword. Furthermore, degeneracy enhances the achievable capacity, because the codewords are not corrupted by the harmless undetected error patterns; hence, the impact of quantum impairments is

TABLE VI
QUANTUM-TO-CLASSICAL ISOMORPHISM

Pauli	$(\mathbb{F}_2)^2$	**GF**(4)
I	00	0
X	01	1
Y	11	$\overline{\omega}$
Z	10	ω
Multiplication	Bit-wise Addition	Addition
Commutativity	Symplectic Product	Trace Inner Product

reduced. Equivalently, we may say that degeneracy enables a quantum code to pack more information as compared to the underlying classical design, because it can operate at a higher coding rate.

VI. QUANTUM-TO-CLASSICAL ISOMORPHISM

Based on the duality of QSCs and classical linear block codes established in Section V, in this section we present the isomorphism between these two regimes, which in turn helps in constructing the quantum-domain versions of the known classical codes. Explicitly, QSCs may be designed from binary and quaternary classical codes using the quantum-to-classical mappings of Table VI, as detailed in Sections VI-A and VI-B, respectively. Furthermore, this quantum-to-classical isomorphism also allows us to use the classical PCM-based syndrome decoding procedures for decoding QSCs.

A. Pauli-to-Binary Isomorphism

Recall from Section V that stabilizers constitute the counterparts of the classical PCM. Based on this duality, QSCs can be described using an equivalent binary PCM, which in turn aids in designing quantum codes from the existing families of classical codes. More specifically, QSCs can be completely characterized in the binary formalism by an equivalent binary PCM $\mathbf{H}$ derived from the associated stabilizer generators. The rows of $\mathbf{H}$ correspond to the stabilizers, while the constituent **I**, **X**, **Y** and **Z** Pauli operators of the stabilizers are mapped onto a pair of binary digits as follows:

$$\mathbf{I} \rightarrow (00), \quad \mathbf{X} \rightarrow (01), \quad \mathbf{Z} \rightarrow (10), \quad \mathbf{Y} \rightarrow (11), \tag{75}$$

where a binary 1 at the first index represents a **Z** operator, while a binary 1 at the second index represents an **X** operator. The PCM $\mathbf{H}$ resulting from the Pauli-to-binary mapping of Eq. (75) can also be expressed as:

$$\mathbf{H} = (\mathbf{H}_z|\mathbf{H}_x), \tag{76}$$

where $\mathbf{H}_z$ and $\mathbf{H}_x$ are $(n-k) \times n$ binary matrices corresponding to the **Z** and **X** operators, respectively. Let us recall that the 3-qubit bit-flip repetition code relied on the stabilizers $g_1 = \mathbf{ZZI}$ and $g_2 = \mathbf{ZIZ}$. Consequently, the associated PCM $\mathbf{H}$ is given by:

$$\mathbf{H} = \left(\begin{array}{ccc|ccc} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{array}\right), \tag{77}$$

where $\mathbf{H}_x$ is an all-zero matrix, since g_1 and g_2 do not contain any Pauli-**X** operators. Furthermore, the $\mathbf{H}_z$ of Eq. (77) is identical to the PCM $\mathbf{H}$ of the classical repetition code given in Eq. (40), hence both yield identical syndrome patterns in Table IV and Table V. Similarly, the PCM of the 3-qubit phase-flip repetition code is:

$$\mathbf{H} = \left(\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{array}\right), \tag{78}$$

where we have $g_1 = \mathbf{XXI}$ and $g_2 = \mathbf{XIX}$, while that of Shor's code is given in Eq. (79).

$$\mathbf{H} = \left(\begin{array}{ccccccccc|ccccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right). \tag{79}$$

Hence, an $[n, k]$ QSC, having $(n-k)$ stabilizers, can be characterized by a binary PCM of size $(n-k) \times 2n$. Furthermore, the equivalent classical coding rate R_c can be determined as follows:

$$\begin{aligned} R_c &= \frac{2n-(n-k)}{2n} \\ &= \frac{n+k}{2n} \\ &= \frac{1}{2}\left(1+\frac{k}{n}\right) \\ &= \frac{1}{2}\big(1+R_Q\big), \end{aligned} \tag{80}$$

where R_Q is its quantum coding rate. Based on Eq. (80), the equivalent classical coding rate of the rate-1/3 quantum repetition code is 2/3, while that of Shor's rate-1/9 code is 5/9.

TABLE VII
$(\mathbb{F}_2)^2$ ADDITION

$+$	00	01	10	11
00	00	01	10	11
01	01	00	11	10
10	10	11	00	01
11	11	10	01	00

The binary formalism of Eq. (75) transforms the multiplication of Pauli operators into the bit-wise addition of the corresponding binary representation. For example, multiplying the set of Pauli operators $\{\mathbf{I}, \mathbf{X}, \mathbf{Z}, \mathbf{Y}\}$ with Pauli-**X** is equivalent to the second column of Table VII, if the Pauli operators are mapped onto $(\mathbb{F}_2)^2$ according to Eq. (75). Similarly, the commutative property of stabilizers in the Pauli formalism implies that the rows of the PCM $\mathbf{H}$ must be orthogonal to each other with respect to symplectic product (also referred to as a twisted product) in the binary formalism. Explicitly, if the ith row of $\mathbf{H}$ is denoted as $\mathbf{H}_i = (\mathbf{H}_{z_i}|\mathbf{H}_{x_i})$ following the notation of Eq. (76), then the commutativity of the stabilizers g_i and $g_{i'}$ is transformed into the symplectic product of rows $\mathbf{H}_i$ and $\mathbf{H}_{i'}$, which is computed as follows:

$$\mathbf{H}_i \star \mathbf{H}_{i'} = \big(\mathbf{H}_{z_i} \cdot \mathbf{H}_{x_{i'}} + \mathbf{H}_{z_{i'}} \cdot \mathbf{H}_{x_i}\big) \bmod 2. \tag{81}$$

The resultant symplectic product yields a value of zero, if the number of different non-Identity operators (**X**, **Y** or **Z**) in

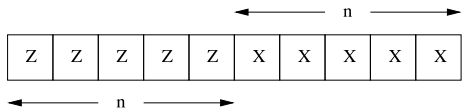


Fig. 26. Effective error P corresponding to the n-qubit Pauli error $\mathcal{P}$.

the stabilizers g_i and $g_{i'}$ is even; hence, satisfying the commutativity criterion. Furthermore, since all stabilizers must be commutative, the symplectic product must be zero for all rows of $\mathbf{H}$, i.e., the PCM $\mathbf{H}$ should satisfy:

$$\mathbf{H}_z\mathbf{H}_x^T + \mathbf{H}_x\mathbf{H}_z^T = 0 \bmod 2. \tag{82}$$

This in turn implies that any pair of classical binary codes having the PCMs $\mathbf{H}_z$ and $\mathbf{H}_x$ and satisfying the symplectic product of Eq. (82) may be used for constructing a valid QSC.

The symplectic product of Eq. (82) may also be exploited for computing the syndrome of a QSC in the binary domain, for example during the PCM-based syndrome decoding. More specifically, the Pauli-to-binary isomorphism of Eq. (75) transforms an n-qubit Pauli error $\mathcal{P} \in \mathcal{G}_n$ into an effective error vector P of length $2n$. Explicitly, analogous to the $\mathbf{H}$ of Eq. (76), the effective error vector P may be expressed as $P = (P_z|P_x)$, where P_z and P_x denote the Pauli-$\mathbf{Z}$ and Pauli-$\mathbf{X}$ errors, respectively. More precisely, a 1 at the tth index of P_z denotes a Pauli-$\mathbf{Z}$ (phase-flip) error on the tth qubit, while a 1 at the tth index of P_x represents the occurrence of the Pauli-$\mathbf{X}$ (bit-flip) error on the tth qubit. Similarly, the Pauli-$\mathbf{Y}$ (bit-and-phase-flip) error on the tth qubit yields a 1 at the tth index of P_z as well as P_x. Finally, the syndrome of a QSC can be computed in the binary formalism using the symplectic product and the effective error vector P as follows:

$$s = \mathbf{H} \star P^T = \left(\mathbf{H}_z P_x^T + \mathbf{H}_x P_z^T\right) \bmod 2, \tag{83}$$

where the $\mathbf{H}_z$ and $\mathbf{H}_x$ are used for correcting bit-flip and phase-flip errors, respectively, as previously discussed in the context of 3-qubit bit-flip and phase-flip repetition codes. The resultant syndrome has either a value of 0 or 1. Thus, the quantum-domain syndrome processing may be carried out in the binary domain using the PCM $\mathbf{H}$ and the effective error P. This in turn implies that the quantum decoding process is equivalent to the syndrome decoding of the equivalent classical code relying on the PCM $\mathbf{H}$ [164]. However, since quantum codes are degenerate, as discussed in Section V, quantum decoding aims for estimating the most probable error coset, while the classical syndrome decoding estimates the most probable error.

B. Pauli-to-Quaternary Isomorphism

Analogous to the Pauli-to-binary isomorphism, the Pauli-to-quaternary isomorphism facilitates the design of quantum codes from the existing classical quaternary codes. Explicitly, the $\mathbf{I}$, $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ Pauli operators may be transformed into the elements of Galois Field GF(4) using the mapping given below:

$$\mathbf{I} \rightarrow 0, \quad \mathbf{X} \rightarrow 1, \quad \mathbf{Z} \rightarrow \omega, \quad \mathbf{Y} \rightarrow \overline{\omega}, \tag{84}$$

TABLE VIII
GF(4) ADDITION

$+$	0	1	ω	$\overline{\omega}$
0	0	1	ω	$\overline{\omega}$
1	1	0	$\overline{\omega}$	ω
ω	ω	$\overline{\omega}$	0	1
$\overline{\omega}$	$\overline{\omega}$	ω	1	0

TABLE IX
GF(4) MULTIPLICATION

$\times$	0	1	ω	$\overline{\omega}$
0	0	0	0	0
1	0	1	ω	$\overline{\omega}$
ω	0	ω	$\overline{\omega}$	1
$\overline{\omega}$	0	$\overline{\omega}$	1	ω

TABLE X
GF(4) HERMITIAN INNER PRODUCT

$\langle , \rangle$	0	1	ω	$\overline{\omega}$
0	0	0	0	0
1	0	1	ω	$\overline{\omega}$
ω	0	$\overline{\omega}$	1	ω
$\overline{\omega}$	0	ω	$\overline{\omega}$	1

where 0, 1, ω and $\overline{\omega}$ are the elements of GF(4). Furthermore, the multiplication operation in the Pauli domain is equivalent to the addition operation in GF(4), while the commutativity (symplectic product) criterion in the Pauli domain is equivalent to the trace[15] inner product [88] in GF(4). The associated additive and multiplicative rules of GF(4) are listed in Table VIII and Table IX,[16] respectively. To elaborate further, multiplying the Pauli operators $\{\mathbf{I}, \mathbf{X}, \mathbf{Z}, \mathbf{Y}\}$ with Pauli-$\mathbf{X}$ is equivalent to adding the GF(4) element 1 (corresponding to Pauli-$\mathbf{X}$) to each element of GF(4), as done in the second column of Table VIII. On the other hand, the commutative relationship between two GF(4) elements $\hat{A}$ and $\hat{B}$ may be established with the help of the trace inner product as follows:[17]

$$\mathrm{Tr}\langle \hat{A}, \hat{B} \rangle = \mathrm{Tr}\left(\hat{A} \times \overline{\hat{B}}\right) = 0, \tag{85}$$

where $\langle , \rangle$ denotes the Hermitian inner product, while $\overline{\hat{B}}$ is the conjugate[18] of $\hat{B}$. Moreover, $\mathrm{Tr}(0) = \mathrm{Tr}(1) = 0$, while $\mathrm{Tr}(\omega) = \mathrm{Tr}(\overline{\omega}) = 1$. Explicitly, both the Hermitian inner product and the trace inner product between the elements of GF(4) are tabulated in Table X and Table XI, respectively.

If a QSC is characterized by the classical PCM $\hat{\mathbf{H}}$ in the quaternary domain, then the commutativity constraint of the stabilizers g_i and g_i' is transformed into the trace inner product of the ith and i'th row of $\hat{\mathbf{H}}$. Explicitly, this may be

[15]The trace operator of GF(4) maps x onto $(x + \overline{x})$, where $\overline{x}$ denotes the conjugate of x [96].

[16]The addition and multiplication rules for GF(p), having a prime p, are the same as the modulo p addition and multiplication, while the rules for GF(p^m), having $m > 1$, do not follow the conventional rules for modulo p^m addition and multiplication. For example, the addition of the elements of GF(4) is equivalent to the bitwise modulo 2 addition of the equivalent 2-bit patterns. Hence, Table VIII may be obtained by mapping the 2-bit patterns of Table VII onto the corresponding GF(4) elements.

[17]GF(4) variables are denoted with a ˆ on top, e.g., $\hat{x}$.

[18]The conjugate operation of GF(4) is defined as $\overline{x} = x^2$ [96]. Consequently, conjugation has no impact on the GF(4) elements 0 and 1, while the elements ω and $\overline{\omega}$ are swapped upon taking the conjugate.

TABLE XI
GF(4) TRACE INNER PRODUCT

tr⟨, ⟩	0	1	ω	$\overline{\omega}$
0	0	0	0	0
1	0	0	1	1
ω	0	1	0	1
$\overline{\omega}$	0	1	1	0

formulated as:

$$\hat{\mathbf{H}}_i \star \hat{\mathbf{H}}_{i'} = \mathrm{Tr}\langle \hat{\mathbf{H}}_i, \hat{\mathbf{H}}_{i'} \rangle = \mathrm{Tr}\left(\sum_{t=1}^{n} \hat{\mathbf{H}}_{it} \times \overline{\hat{\mathbf{H}}}_{i't} \right) = 0, \quad (86)$$

where $\hat{\mathbf{H}}_{it}$ is the element in the *i*th row and *t*th column of $\hat{\mathbf{H}}$.

Let us now prove the equivalence of Eq. (81) and Eq. (86), since both these equations correspond to the commutativity requirement. Given $\mathbf{H}_i = (\mathbf{H}_{z_i}, \mathbf{H}_{x_i})$ and the mapping of Eq. (84), $\hat{\mathbf{H}}_i$ may be expressed as:

$$\hat{\mathbf{H}}_i = \omega \mathbf{H}_{z_i} + \mathbf{H}_{x_i}. \quad (87)$$

Substituting Eq. (87) into Eq. (86) yields:

$$\begin{aligned} \hat{\mathbf{H}}_i \star \hat{\mathbf{H}}_{i'} &= \mathrm{Tr}\langle (\omega \mathbf{H}_{z_i} + \mathbf{H}_{x_i}), \left(\omega \mathbf{H}_{z_{i'}} + \mathbf{H}_{x_{i'}}\right) \rangle \\ &= \mathrm{Tr}\left((\omega \mathbf{H}_{z_i} + \mathbf{H}_{x_i}) \left(\overline{\omega} \mathbf{H}_{z_{i'}} + \mathbf{H}_{x_{i'}}\right) \right) \\ &= \mathrm{Tr}\left(\mathbf{H}_{z_i} \mathbf{H}_{z_{i'}} + \omega \mathbf{H}_{z_i} \mathbf{H}_{x_{i'}} + \overline{\omega} \mathbf{H}_{x_i} \mathbf{H}_{z_{i'}} + \mathbf{H}_{x_i} \mathbf{H}_{x_{i'}} \right). \end{aligned} \quad (88)$$

Recall that $\mathrm{Tr}(1) = 0$ and $\mathrm{Tr}(\omega) = \mathrm{Tr}(\overline{\omega}) = 1$. Therefore, Eq. (88) reduces to:

$$\hat{\mathbf{H}}_i \star \hat{\mathbf{H}}_{i'} = \mathbf{H}_{z_i} \mathbf{H}_{x_{i'}} + \mathbf{H}_{x_i} \mathbf{H}_{z_{i'}}, \quad (89)$$

which is the same as Eq. (81). Consequently, analogous to Eq. (83), the syndrome in the quaternary domain is computed as:

$$s_i = \mathrm{Tr}(\hat{s}_i) = \mathrm{Tr}\left(\sum_{t=1}^{n} \hat{\mathbf{H}}_{it} \times \overline{\hat{P}}_t \right), \quad (90)$$

where s_i is the syndrome corresponding to the *i*th row of $\hat{\mathbf{H}}$ and $\hat{P}_t$ is the *t*th element of $\hat{P}$, which represents the error inflicted on the *t*th qubit.

Any arbitrary classical quaternary linear code, which is self-orthogonal with respect to the trace inner product of Eq. (86), can be used for constructing a QSC. Since a quaternary linear code is closed under multiplication by the elements of GF(4), this condition reduces to satisfying the Hermitian inner product, rather than the trace inner product [96]. This can be proved as follows.

Let *C* be a classical linear code in GF(4) having codewords *u* and *v*. Furthermore, let us assume that:

$$\langle u, v \rangle = \alpha + \beta\omega. \quad (91)$$

For the sake of satisfying the symplectic product, we must have:

$$\mathrm{Tr}\langle u, v \rangle = 0. \quad (92)$$

Since $\mathrm{Tr}(\omega) = 1$, Eq. (92) is only valid, when β is zero in Eq. (91). Furthermore, since the code *C* is GF(4)-linear, Eq. (92) leads to:

$$\mathrm{Tr}\langle u, \overline{\omega} v \rangle = 0, \quad (93)$$

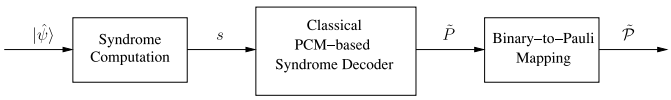


Fig. 27. Syndrome processing block of Fig. 22.

which in turn implies that α should also be zero in Eq. (91). Hence, for a classical GF(4)-linear code, the Hermitian inner product of Eq. (91) must be zero, when the trace inner product of Eq. (92) is zero.

To conclude, the stabilizers may be mapped onto the equivalent binary or quaternary representations, as summarized in Table VI. These mappings in turn help in designing quantum codes from the existing classical codes, as discussed further in the next section. Furthermore, since a QSC can be mapped onto an equivalent classical binary or quaternary PCM, classical PCM-based syndrome decoding may be invoked during the quantum decoding process. More explicitly, the 'syndrome processing' block of Fig. 22 may be expanded, as shown in Fig. 27. The process begins with the computation of the syndrome of the received sequence $|\hat{\psi}\rangle$ using the stabilizer generators, which collapse to a binary 0 or 1 upon measurement. The binary syndrome sequence *s* is then fed to a classical PCM-based syndrome decoder, which operates over the equivalent classical PCM associated with the QSC for estimating the equivalent channel error $\tilde{P}$ (or $\tilde{\hat{P}}$ in quaternary domain). The classical PCM-based syndrome decoder of Fig. 27 is exactly the same decoder, which we would use for any conventional classical code, with the exception of the following two differences:

1) In contrast to the syndrome of a classical code, which is the product of the PCM and the transpose of the channel error ($\mathbf{H}P^T$), the syndrome of a quantum code is computed using the symplectic product of Eq. (83) (or the trace inner product of Eq. (90)).
2) The conventional classical decoding aims for estimating the most probable error, given the observed syndrome, while quantum decoding aims for estimating the most probable error coset, which takes into account the degeneracy of quantum codes, as discussed in Section V.

Finally, the binary-to-Pauli mapping of Eq. (75) (or quaternary-to-Pauli mapping of Eq. (84)) is invoked for mapping the estimated binary (or quaternary) error onto the equivalent Pauli error $\tilde{\mathcal{P}}$.

VII. TAXONOMY OF STABILIZER CODES

The quantum-to-classical isomorphism of Section VI provides a solid theoretical framework for building quantum codes from the known classical codes, which have already found their way into commercial applications. Particularly, quantum codes can be designed from a pair of arbitrary classical binary codes, if they meet the symplectic criterion, or from arbitrary classical quaternary codes, if they satisfy the Hermitian inner product. Continuing further our discussions, in this section we present the taxonomy of stabilizer codes with the aid of Fig. 28, which is based on the structure of the underlying equivalent classical PCM **H**.

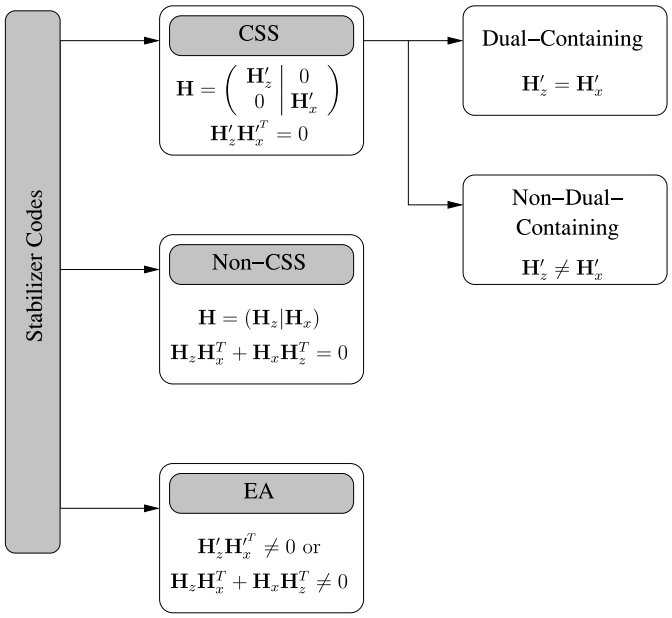


Fig. 28. Taxonomy of Stabilizer Codes *(CSS: Calderbank-Shor-Steane, EA: Entanglement-Assisted).*

A. Calderbank-Shor-Steane Codes

Calderbank-Shor-Steane (CSS) codes [82]–[84] is a class of stabilizer codes constructed from a pair of binary classical codes. Specifically, the family of CSS codes may be defined as:

An $[n, k_1 - k_2]$ CSS code can be designed from the binary linear block codes $C_1(n, k_1)$ and $C_2(n, k_2)$, if the code space of C_1 subsumes that of C_2 ($C_2 \subset C_1$). Furthermore, if both C_1 as well as the dual of C_2, i.e., $C_2^{\perp}$, exhibit a minimum Hamming distance of $d_{\min}$, then the resultant CSS code also exhibits a minimum distance of $d_{\min}$; hence, it is capable of concurrently correcting $(d_{\min} - 1)/2$ bit-flips as well as $(d_{\min} - 1)/2$ phase-flips.

Explicitly, analogous to Shor's code, a CSS code independently corrects bit-flip and phase-flip errors. More specifically, the binary code C_1 is invoked for correcting bit-flips, while the code $C_2^{\perp}$ is used for phase-flip correction. Hence, if $\mathbf{H}'_z$ and $\mathbf{H}'_x$ are the PCMs of C_1 and $C_2^{\perp}$, respectively, then the resultant CSS code has the following PCM:

$$\mathbf{H} = [\mathbf{H}_z|\mathbf{H}_x] = \left(\begin{array}{c|c} \mathbf{H}'_z & \mathbf{0} \\ \mathbf{0} & \mathbf{H}'_x \end{array}\right), \tag{94}$$

where we have $\mathbf{H}_z = \begin{pmatrix} \mathbf{H}'_z \\ \mathbf{0} \end{pmatrix}$, $\mathbf{H}_x = \begin{pmatrix} \mathbf{0} \\ \mathbf{H}'_x \end{pmatrix}$, while $\mathbf{H}'_z$ and $\mathbf{H}'_x$ are $(n - k_1) \times n$ and $k_2 \times n$ binary matrices, respectively. Furthermore, since $C_2 \subset C_1$, the symplectic condition of Eq. (82) is reduced to:

$$\mathbf{H}'_z\mathbf{H}'^T_x = 0. \tag{95}$$

Hence, the process of designing a QSC is reduced to finding a pair of binary codes whose PCMs conform to the symplectic criterion of Eq. (95). Since the resultant PCM of Eq. (94) has $(n - k_1 + k_2)$ rows, the quantum code encodes $(k_1 - k_2)$ information qubits into *n* qubits. Moreover, if we have $\mathbf{H}'_z = \mathbf{H}'_x$, then the resultant code is called a dual-containing (or self-orthogonal) code having $\mathbf{H}_z\mathbf{H}'^T_z = 0$, which is equivalent to $C_1^{\perp} \subset C_1$. Explicitly, in case of dual-containing CSS codes, $C_2(n, k_2)$ is the dual code of $C_1(n, k_1)$. Therefore, we have $k_2 = (n - k_1)$ and the resultant dual-containing CSS codes encodes $(k_1 - k_2) = (2k_1 - n)$ qubits into *n* coded qubits. We classify the remaining CSS constructions, having $\mathbf{H}'_z \neq \mathbf{H}'_x$, as non-dual-containing CSS codes.

TABLE XII
UNIQUE COSETS OF $C_1^{\perp}$ IN C_1

Coset 1	Coset 2
0000000	1111111
0111001	1000110
1011010	0100101
1100011	0011100
1101100	0010011
1010101	0101010
0110110	1001001
0001111	1110000

An $[n, k_1 - k_2]$ CSS code, relying on the binary codes C_1 and $C_2^{\perp}$, is implemented by finding the unique cosets[19] of C_2 in C_1, so that each of the $2^{k_1-k_2}$ superimposed state can be mapped onto a unique coset of C_2 in C_1. These unique cosets are in turn derived by adding (bit-wise modulo-2) each codeword of C_1 to the code space of C_2. More specifically, if $x_1 \in C_1$ and $x_2 \in C_2$, then the normalized addition operation can be formulated as:

$$|x_1 + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{x_2 \in C_2} |x_1 + x_2\rangle. \tag{96}$$

Since the cardinality of C_1 is $|C_1| = 2^{k_1}$, while that of C_2 is $|C_2| = 2^{k_2}$, we get $|C_1|/|C_2| = 2^{k_1-k_2}$ unique cosets of C_2 in C_1. Consequently, each of the $2^{k_1-k_2}$ $(k_1 - k_2)$-qubit orthogonal quantum state can be mapped onto a superposition of the codewords of the unique coset.

Let us now consider the construction of Steane's [7, 1] code, which is derived from the dual-containing classical (7, 4) Hamming code having the PCM:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}. \tag{97}$$

The PCM $\mathbf{H}$ of Eq. (97) yields $\mathbf{H}\mathbf{H}^T = 0$, hence lending itself to constructing a dual-containing CSS code. More specifically, C_1 is the (7, 4) Hamming code, while C_2 is its dual code, i.e., $C_2 = C_1^{\perp}$, having the parameters (7, 3). Since $\mathbf{H}\mathbf{H}^T = 0$, the code space of C_2 is contained in that of C_1, i.e., we have $C_2 \subset C_1$. Furthermore, both C_1 and $C_2^{\perp} = C_1$ can correct a single error. Consequently, a single-error correcting CSS code can be constructed by finding the unique cosets of $C_1^{\perp}$ in C_1 using Eq. (96). This results in two unique cosets, which are listed in Table XII. These two cosets together yield the code

[19]Assume $C_1 = (0, 1, 2, 3)$ with $k_1 = 2$ and $C_2 = (0, 2)$ with $k_2 = 1$, modulo 4 addition yields following cosets:

$$\begin{aligned} 0 + C_2 &\equiv (0, 2) = C_2, \\ 1 + C_2 &\equiv (1, 3) = 1 + C_2, \\ 2 + C_2 &\equiv (2, 0) = C_2, \\ 3 + C_2 &\equiv (3, 1) = 1 + C_2. \end{aligned}$$

Hence, resulting in two different cosets of C_2 in C_1, i.e., (0, 2) and (1, 3). Equivalently, we may say that the two unique cosets (0, 2) and (1, 3) of C_2 together constitute the code space of C_1.

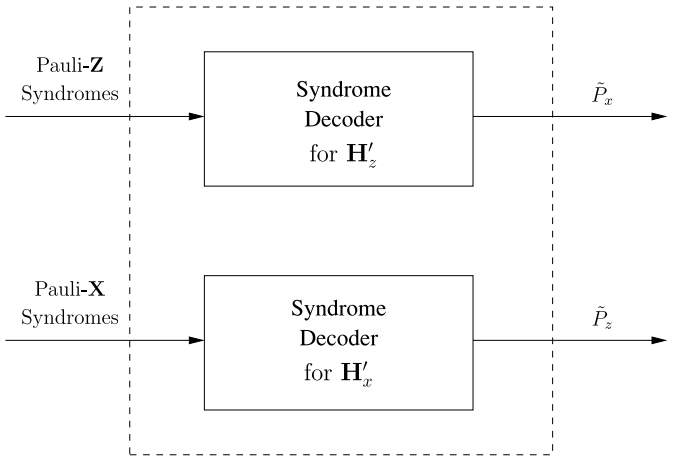


Fig. 29. Syndrome decoder for CSS-type Quantum Codes.

space of the (7, 4) Hamming code. The two orthogonal states $|0\rangle$ and $|1\rangle$ of the single qubit information word are hence encoded as follows:

$$\begin{aligned}|\overline{0}\rangle \equiv \frac{1}{\sqrt{8}}(&|0000000\rangle + |0111001\rangle + |1011010\rangle + |1100011\rangle \\ &+ |1101100\rangle + |1010101\rangle + |0110110\rangle + |0001111\rangle), \\ |\overline{1}\rangle \equiv \frac{1}{\sqrt{8}}(&|1111111\rangle + |1000110\rangle + |0100101\rangle + |0011100\rangle \\ &+ |0010011\rangle + |0101010\rangle + |1001001\rangle + |1110000\rangle).\end{aligned} \tag{98}$$

In other words, $|\overline{0}\rangle$ and $|\overline{1}\rangle$ are the equally weighted superpositions of all the codewords of the two cosets of Table XII. Furthermore, $\mathbf{H}'_z$ and $\mathbf{H}'_x$ of the resultant quantum code space are equivalent to the binary PCM of the Hamming code (Eq. (97)). Hence, the associated bit-flip and phase-flip detecting stabilizers of the [7, 1] Steane's code are as follows:

$$\begin{aligned} g_1 &= \mathbf{ZZIZZII} \\ g_2 &= \mathbf{ZIZZIZI} \\ g_3 &= \mathbf{IZZZIIZ} \\ g_4 &= \mathbf{XXIXXII} \\ g_5 &= \mathbf{XIXXIXI} \\ g_6 &= \mathbf{IXXXIIX}. \end{aligned} \tag{99}$$

We may observe in Eq. (99) as well as in Eq. (94) that the bit-flip and phase-flip detecting stabilizers (or equivalently syndromes) of a CSS-type quantum code are independent. Therefore, bit-flip and phase-flip estimation may be carried out independently by two separate classical syndrome decoders using $\mathbf{H}'_z$ and $\mathbf{H}'_x$, respectively, as illustrated in Fig. 29. Furthermore, when the simplified decoder of Fig. 29 is invoked, the performance of CSS codes observed in the face of the depolarizing channel of Eq. (24) is isomorphic to their performance over two independent phase-flip and bit-flip channels, where each has a marginalized depolarizing probability of 2p/3. Hence, the QBER performance of CSS codes may be approximated by adding together the BERs of the constituent binary codes. More explicitly, given that p_e^x and p_e^z are the classical BERs for $\mathbf{H}'_z$ and $\mathbf{H}'_x$, respectively, the resultant CSS code exhibits a QBER of:

$$\text{QBER} = p_e^x + p_e^z - p_e^x p_e^z \approx p_e^x + p_e^z, \tag{100}$$

which is equivalent to $2p_e^z$ for a dual-containing CSS code having $\mathbf{H}'_x = \mathbf{H}'_z$.

B. Non-CSS Codes

We observed in the previous section that CSS codes independently correct bit-flip and phase-flip errors. This in turn results in a low coding rate. By contrast, non-CSS stabilizer codes are capable of exploiting the redundancy more efficiently, since they jointly correct bit-flip and phase-flip errors. The PCM of a non-CSS code assumes the general structure of Eq. (76). Consequently, a pair of binary PCMs conforming to the symplectic product criterion of Eq. (82) or a classical quaternary PCM satisfying the trace inner product of Eq. (86) may be used for designing a non-CSS stabilizer code.

Calderbank, Rains, Shor and Sloane conceived a special class of non-CSS codes, called Calderbank-Rains-Shor-Sloane (CRSS) codes, which are constructed from the known classical quaternary codes as follows [96]:

An [n, k] QSC can be designed in the quaternary domain from a classical self-orthogonal (under the Hermitian inner product) GF(4)-linear block code $C(n,(n-k)/2)$. Furthermore, if the dual (also called orthogonal) code $C^{\perp}(n,(n+k)/2)$ exhibits a minimum Hamming distance of $d_{\min}$, then the resultant non-CSS code also exhibits a minimum distance of $d_{\min}$; hence, it is capable of concurrently correcting $(d_{\min}-1)/2$ bit-flips as well as $(d_{\min}-1)/2$ phase-flips.

Based on this formalism, the PCM of the resultant CRSS code is characterized as:

$$\hat{\mathbf{H}} = \begin{pmatrix} \hat{\mathbf{H}}_c \\ \omega\hat{\mathbf{H}}_c \end{pmatrix}, \tag{101}$$

where $\hat{\mathbf{H}}_c$ is the PCM of the dual code $C^{\perp}(n,(n+k)/2)$. For example, there exists a classical self-orthogonal GF(4)-linear code $C(5, 2)$, whose dual code $C^{\perp}(5,3)$ is a Hamming code having the PCM $\hat{\mathbf{H}}_c$ given by [169]:

$$\hat{\mathbf{H}}_c = \begin{pmatrix} 0 & \overline{\omega} & \omega & \omega & \overline{\omega} \\ \overline{\omega} & 0 & \overline{\omega} & \omega & \omega \end{pmatrix}. \tag{102}$$

Consequently, the (5, 1) quantum Hamming code can be constructed as:

$$\hat{\mathbf{H}} = \begin{pmatrix} 0 & \overline{\omega} & \omega & \omega & \overline{\omega} \\ \overline{\omega} & 0 & \overline{\omega} & \omega & \omega \\ 0 & 1 & \overline{\omega} & \overline{\omega} & 1 \\ 1 & 0 & 1 & \overline{\omega} & \overline{\omega} \end{pmatrix}. \tag{103}$$

Using the Pauli-to-GF(4) mapping of Eq. (84), the PCM $\hat{\mathbf{H}}$ of Eq. (103) is mapped onto the stabilizer generators listed below:

$$\begin{aligned} g_1 &= \mathbf{IYZZY} \\ g_2 &= \mathbf{YIYZZ} \\ g_3 &= \mathbf{IXYYX} \\ g_4 &= \mathbf{XIXYY}. \end{aligned} \tag{104}$$

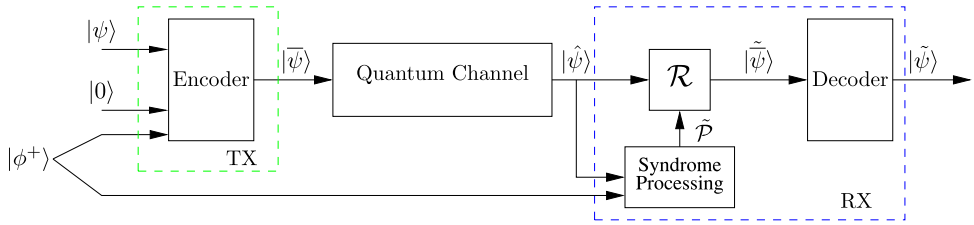


Fig. 30. System Model: Quantum communication system relying on an entanglement-assisted quantum stabilizer code.

Hence, while a single-error correcting CSS-type code has a coding rate of 1/7, a single-error correcting non-CSS code exhibits an improved coding rate of 1/5. The resultant codes may be decoded by invoking a classical non-binary syndrome decoder or a binary syndrome decoder operating over the binary PCM of Eq. (76), which exploit the correlation between the bit-flip and phase-flip errors, hence facilitating the joint decoding of bit-flip and phase-flip errors. This in turn provides enhanced decoding performance, albeit at the cost of an increased decoding complexity.

C. Entanglement-Assisted Codes

Let us recall that QSCs may be constructed from the classical binary and quaternary codes only if the constituent classical codes conform to the symplectic criterion of Eq. (82). Consequently, while every QSC may have a classical counterpart, we cannot claim that every classical code has a stabilizer-based quantum version. Furthermore, the stringent symplectic criterion may result in various design issues, such as the unavoidable short cycles in QLDPC codes and the non-recursive nature of non-catastrophic QCCs. For the sake of overcoming these limitations, the entanglement-assisted stabilizer formalism of [112] and [116] was conceived, which relies on entangled qubits pre-shared with the receiver over a noiseless channel. Explicitly, the EA formalism helps in transforming a set of non-commuting Pauli generators into a set of commuting generators, which in turn constitute valid stabilizer codes. Consequently, when the underlying classical codes do not satisfy the symplectic criterion, the EA formalism is invoked for making the resultant stabilizers commutative.

Fig. 30 shows the system model of a quantum communication system relying on an Entanglement-Assisted Quantum Stabilizer Code (EA-QSC). Explicitly, an [n, k, c] EA-QSC encodes a k-qubit information word $|\psi\rangle$ into an n-qubit codeword $|\overline{\psi}\rangle$ with the help of $(n-k-c)$ auxiliary qubits in state $|0\rangle$ and c pre-shared entangled qubits (ebits). Explicitly, ebits may be created in the Bell state $|\phi^+\rangle$, expressed as:

$$|\phi^+\rangle = \frac{|00\rangle^{T_X R_X} + |11\rangle^{T_X R_X}}{\sqrt{2}}, \tag{105}$$

so that the first qubit is retained at the transmitter, while the associated entangled qubit is sent to the receiver before actual transmission commences, for example during off-peak hours, when the channels are under-utilized. The notations T_X and R_X in Eq. (105) are used to identify the transmitter's and receiver's half of the ebit, respectively. It is generally assumed that the pre-sharing of ebits takes place over a noiseless channel. Furthermore, as illustrated in Fig. 30, the transmitter only utilizes the transmitter's half of the ebits for encoding the information word $|\psi\rangle$ into the codeword $|\overline{\psi}\rangle$. Finally, the encoded information is sent over a noisy quantum channel. At the receiver, the received noisy codeword $|\hat{\psi}\rangle$ is combined with the receiver's half of the c ebits during the decoding process. Specifically, the stabilizers of an EA-QSC jointly act on $|\hat{\psi}\rangle$ and the receiver's ebits for computing the syndrome vector, which is then fed to a classical syndrome decoder for estimating the error pattern $\tilde{\mathcal{P}}$, as previously shown in Fig. 27. The rest of the processing at the receiver is identical to that of the unassisted QSC of Fig. 22.

The Bell state of Eq. (105) has unique properties, which facilitate the mapping of a set of non-commuting generators into a set of commuting generators. More explicitly, the 2-qubit commuting generators $\mathbf{X}^{T_X}\mathbf{X}^{R_X}$ and $\mathbf{Z}^{T_X}\mathbf{Z}^{R_X}$ stabilize the state $|\phi^+\rangle$, i.e., we have:

$$\left[\mathbf{X}^{T_X}\mathbf{X}^{R_X}, \mathbf{Z}^{T_X}\mathbf{Z}^{R_X}\right] = 0. \tag{106}$$

However, the Pauli operators acting on the individual qubits anti-commute with each other, i.e., we have:

$$\begin{aligned}\left[\mathbf{X}^{T_X}, \mathbf{Z}^{T_X}\right] &\neq 0,\\ \left[\mathbf{X}^{R_X}, \mathbf{Z}^{R_X}\right] &\neq 0.\end{aligned} \tag{107}$$

Therefore, if we have a pair of non-commutative generators $\mathbf{X}^{T_X}$ and $\mathbf{Z}^{T_X}$, which only act on the transmitter's ebit, then these generators can be transformed into a pair of commuting generators by appropriately augmenting them with an additional operator acting on the receiver's ebit. Explicitly, the operator acting on the receiver's ebits is specifically chosen for ensuring that the resultant 2-qubit generators have an even number of indices, which have different non-identity operators; hence, resolving the anti-commutativity of the initial single qubit operators.

Let us now construct an EA-QSC from two binary codes having the PCMs[20]:

$$\mathbf{H}_z = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \tag{108}$$

[20]This is an arbitrary, random example only conceived for illustrating the construction of EA codes from the known classical codes. The associated classical/quantum code may not have good error correction capabilities.

and:

$$\mathbf{H}_x = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}. \tag{109}$$

The PCMs $\mathbf{H}_z$ and $\mathbf{H}_x$ may be concatenated for constructing a non-CSS code having:

$$\mathbf{H} = \left(\begin{array}{cccc|cccc} 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{array}\right). \tag{110}$$

Unfortunately, the PCM of Eq. (110) does not meet the symplectic product criterion of Eq. (82). Furthermore, the PCM $\mathbf{H}$ may be transformed into the following non-commutative Pauli generators using the Pauli-to-binary mapping of Eq. (75):

$$\mathbf{H}_Q = \begin{pmatrix} \mathbf{X} & \mathbf{Z} & \mathbf{X} & \mathbf{I} \\ \mathbf{X} & \mathbf{X} & \mathbf{I} & \mathbf{X} \\ \mathbf{Y} & \mathbf{Z} & \mathbf{Z} & \mathbf{X} \\ \mathbf{X} & \mathbf{Y} & \mathbf{Y} & \mathbf{Z} \end{pmatrix}. \tag{111}$$

Explicitly, the first two generators (or rows) of $\mathbf{H}_Q$ anti-commute, while all other generators (or rows) commute with each other. This is because the first two generators have a single index having different non-Identity operators. In other words, only the operators acting on the second qubit anti-commute, while the operators individually acting on all other qubits commute. For the sake of making the generators of Eq. (111) commutative, the first two rows of $\mathbf{H}_Q$ may be augmented with a pair of anti-commuting operators, as shown below:

$$\mathbf{H}_Q = \left(\begin{array}{cccc|c} \mathbf{X} & \mathbf{Z} & \mathbf{X} & \mathbf{I} & \mathbf{Z} \\ \mathbf{X} & \mathbf{X} & \mathbf{I} & \mathbf{X} & \mathbf{X} \\ \mathbf{Y} & \mathbf{Z} & \mathbf{Z} & \mathbf{X} & \mathbf{I} \\ \mathbf{X} & \mathbf{Y} & \mathbf{Y} & \mathbf{Z} & \mathbf{I} \end{array}\right), \tag{112}$$

where the operators to the left of the vertical bar ($|$) act on the n-qubit transmitted codewords, while those on the right of the vertical bar act on the receiver's half of the ebits. Hence, only a single ebit is required in this design example.

VIII. Design Examples

We may conclude from the above discussions that the stabilizer formalism is a useful framework for exploiting the known classical coding families. In this section, we extend our discussions to the two widely used channel coding families, i.e., the BCH codes (Section VIII-A) and the convolutional codes (Section VIII-B), emphasizing the duality between their classical and quantum versions.

A. Bose-Chaudhuri-Hocquenghem Codes

1) Classical Bose-Chaudhuri-Hocquenghem Codes [142]: Bose-Chaudhuri-Hocquenghem (BCH) codes are classified as maximum minimum-distance multiple-error correcting cyclic block codes. A classical BCH code denoted as BCH$(n, k, d_{\min})$ encodes $k \geq (n - mt)$ information bits into n-bit codewords, where $n = 2^m - 1$, so that the resultant code space has an odd minimum Hamming distance of $d_{\min}$, hence it is capable of correcting $t = (d_{\min} - 1)/2$ errors. Furthermore, BCH codes can be both systematic as well as non-systematic. However, systematic BCH codes are known to outperform their non-systematic counterparts [142]. This is because they can exploit their error-detection capability for disabling the decoding operations, when this would result in correcting the wrong symbols owing to having more than t errors. In such instances, the systematic BCH decoder simply retains the systematic part of the codeword. Unfortunately, the non-systematic decoder does not have separate information and parity segments, hence it would correct the wrong symbols, when it is overloaded by more than t errors. This causes even more errors after decoding than we had at the channel's output.

A systematic binary BCH code encodes k information bits into n coded bits by appending $(n - k)$ parity bits to the block of k information bits. The parity bits are computed from the information bits based on the generator polynomial $g(x)$, which is given by:

$$g(x) = g_0 + g_1 x + g_2 x^2 + \cdots + g_{n-k} x^{n-k}. \tag{113}$$

As detailed in [142] and [177], the systematic encoder operates by first shifting the information polynomial $d(x)$ to the highest order position of the codeword $c(x)$ by multiplying $d(x)$ with $x^{(n-k)}$ and then attaching the parity segment to it. Explicitly, the parity symbols denoted by the polynomial $p(x)$ are defined according to the generator polynomial $g(x)$, so that the resulting codeword $c(x)$ is a valid codeword. The overall systematic encoding process may be summarized as:

$$c(x) = x^{(n-k)}.d(x) + p(x), \tag{114}$$

where $p(x)$ is defined as:

$$p(x) = -\mathrm{Rem}\left[\frac{x^{(n-k)}.d(x)}{g(x)}\right], \tag{115}$$

for the sake of ensuring that $c(x)$ constitutes a valid codeword, hence yielding a zero-valued remainder upon division by the generator polynomial $g(x)$, i.e., we have:

$$\begin{aligned} \mathrm{Rem}\left[\frac{c(x)}{g(x)}\right] &= \mathrm{Rem}\left[\frac{x^{(n-k)}.d(x) + p(x)}{g(x)}\right] \\ &= \mathrm{Rem}\left[\frac{x^{(n-k)}.d(x)}{g(x)}\right] + \mathrm{Rem}\left[\frac{p(x)}{g(x)}\right] = 0, \end{aligned} \tag{116}$$

since,

$$\mathrm{Rem}\left[\frac{p(x)}{g(x)}\right] = p(x), \tag{117}$$

according to Eq. (115). The corresponding polynomial multiplications and divisions of Eq. (114) and Eq. (115), respectively, may be carried out by low-complexity shift register based operations, as exemplified below.

The encoder of a systematic BCH code may be implemented using shift registers, as depicted in Fig. 31, where $\otimes$ denotes the multiplication operation, while $\oplus$ is the modulo-2 addition.

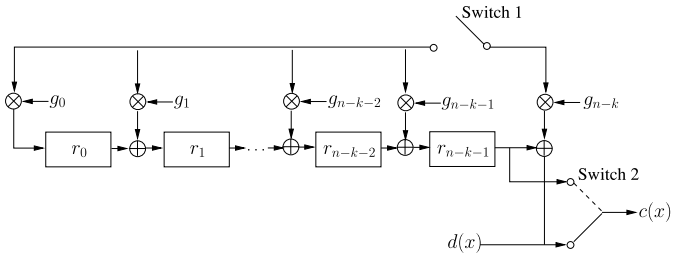


Fig. 31. Schematic of the systematic BCH$(n, k, d_{\min})$ encoder.

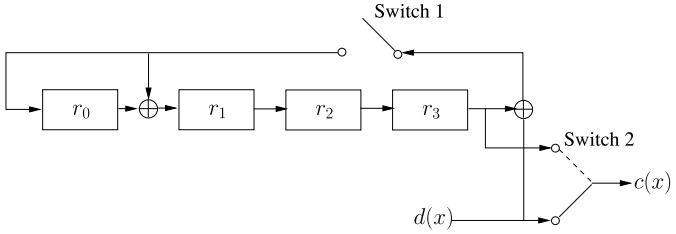


Fig. 32. Encoder of systematic BCH(15, 11, 3).

Specifically, the information bits $d(x)$ are encoded into the coded bits $c(x)$ as follows:

1) Switch 1 is closed during the first k time instants (or clock cycles), hence allowing the information bits $d(x)$ to flow into the $(n-k)$ shift registers according to the rules defined by the generator polynomial $g(x)$. Explicitly, the contents of the shift registers after the kth time instant constitute the parity bits.
2) Concurrently, Switch 2 is in the down position, so that the k information bits $d(x)$ constitute the first k bits of $c(x)$.
3) After k time instants, Switch 1 is opened, while Switch 2 is moved to the upper position. This clears the shift registers by moving their contents to the output $c(x)$.

Let us consider the BCH(15, 11, 3) code having the generator polynomial[21]:

$$\begin{aligned} g &= 23_{\text{octal}} \\ &= 10011_{\text{bin}}, \\ g(x) &= x^4 + x + 1. \end{aligned} \tag{118}$$

The associated encoding circuit of Fig. 32 can be easily derived from Fig. 31 based on the generator polynomial of Eq. (118). We may observe in Eq. (118) that the coefficients can only have a value of 1 or 0. Consequently, the multiplier is replaced by a direct hard-wire connection, if the corresponding coefficient is 1, while no connection is made, when the coefficient is 0. Let us assume an 11-bit input sequence d = 11001110001, which may also be represented as $d(x) = 1 + x + x^4 + x^5 + x^6 + x^{10}$. The encoding process proceeds as follows:

1) The shift registers are initialized to the all-zero state. During the first $k = 11$ time instances, when the Switch 1 is closed, the input bits flow into the shift registers of Fig. 32. The resultant states are tabulated in Table XIII at each time instant.
2) Furthermore, since Switch 2 is downward position for the first $k = 11$ time instances, the coded bits of $c(x)$ are the same as the information bits $d(x)$.
3) Thereafter, since Switch 1 is opened and Switch 2 is moved to the upper position, the values within the shift registers represent the coded bits, as demonstrated in Table XIII. Eventually, the shift registers are returned to the initial all-zero state.

[21]The generator polynomial $g(x)$ is often represented by an octal number, so that when it is converted to the binary notation, the right-most bit constitutes the coefficient of x^0, i.e., the zero-degree coefficient.

TABLE XIII
BCH(15, 11, 3) ENCODING PROCESS FOR $d = 11001110001$ ($d(x) = 1 + x + x^4 + x^5 + x^6 + x^{10}$), WHICH YIELDS THE CODEWORD $c = 101011001110001$ ($c(x) = x^2 + x^4 + x^5 + x^8 + x^9 + x^{10} + x^{14}$)

Index	Input Bit	State ($r_0 r_1 r_2 r_3$) Binary	State ($r_0 r_1 r_2 r_3$) Decimal	Output Bit
0	-	0000	0	-
1	1	1100	12	1
2	0	0110	6	0
3	0	0011	3	0
4	0	0001	1	0
5	1	1100	12	1
6	1	1010	10	1
7	1	1001	9	1
8	0	0100	4	0
9	0	0010	2	0
10	1	1101	13	1
11	1	1010	10	1
12	-	0101	5	0
13	-	0010	2	1
14	-	0001	1	0
15	-	0000	0	1

Equivalently, the encoding process of Table XIII may also be represented by using the state transition diagram of Fig. 33, which shows all possible transitions for the BCH encoder of Fig. 32. In its *conceptually simplest form*, the decoding relies on a simple decoding table, which has a total of $2^{15} = 32768$ entries and $2^{11} = 2048$ legitimate codewords. Since this code has $d_{\min} = 3$, the received corrupted codeword is readily corrected in case of a single error, but the wrong legitimate codeword is selected in case of two errors. The state transition diagram of Fig. 33 also facilitates trellis decoding [65] of BCH codes. However, the number of trellis states increases exponentially with $(n - k)$, since the trellis has a total of $2^{(n-k)}$ states. As an alternative strategy, the Berlekamp-Massey algorithm [56]–[59] and Chase algorithm [63] are widely used for efficiently decoding BCH codes. Fig. 34 portrays the coding gain versus coding rate trend at a BER of 10^{-6} for different-rate BCH codes relying on the same codeword length, i.e., for n = (15, 31, 63, 127). We may observe in Fig. 34 that the coding gain increases upon increasing the coding rate (or equivalently increasing k) until it reaches the maximum value. More specifically, the maximum coding gain is typically achieved when the coding rate is between 0.5 and 0.6.

2) Quantum Bose-Chaudhuri-Hocquenghem Codes: Quantum BCH codes [94]–[99] can be derived from the classical dual-containing binary BCH codes as well as self-orthogonal quaternary BCH codes. In this section, we will detail the construction of a dual-containing BCH code, based on our discussions of Section VII-A.

Let us recall from Section VII-A that if C is the classical code specified by the PCM $\mathbf{H}$ and having the dual code

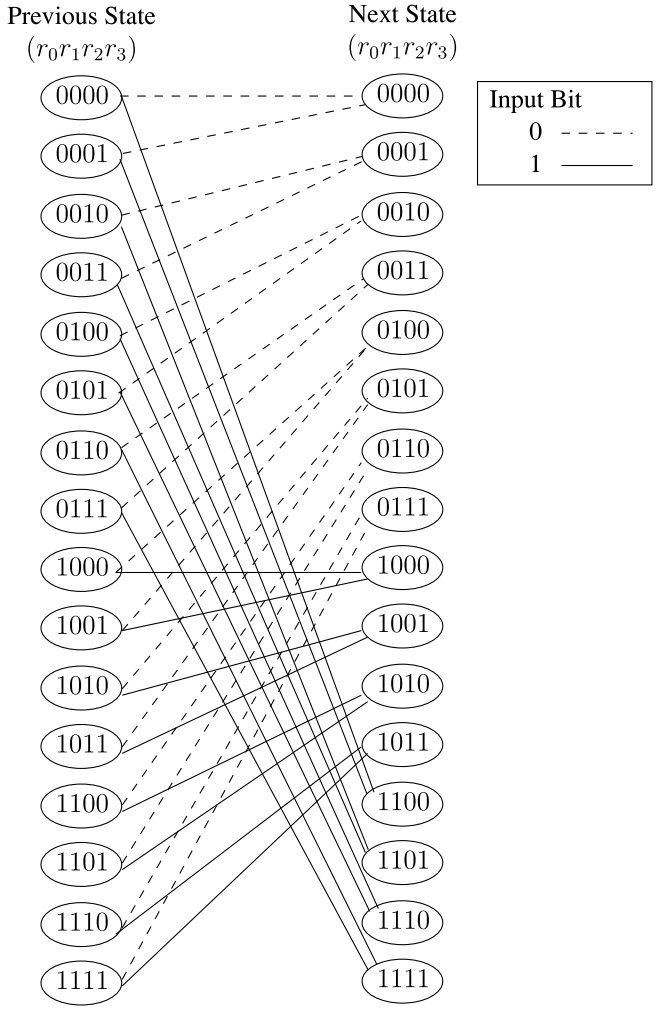


Fig. 33. State transition diagram for BCH(15, 11, 3).

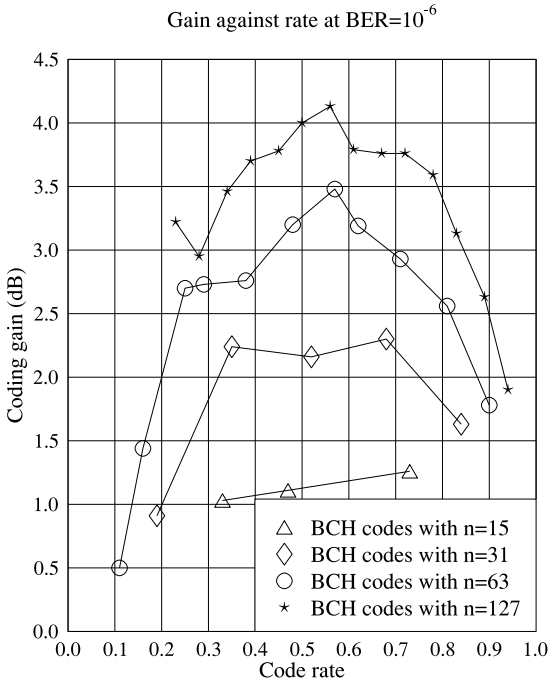


Fig. 34. Coding gain versus coding rate for various families of BCH codes at a BER of 10^{-6} over AWGN channel [142]. *Berlekamp-Massey algorithm was invoked for decoding.*

$C^{\perp}$, whose code space is subsumed by that of C ($C^{\perp} \subset C$), then the resultant $[n, k']$ dual-containing CSS code, having $k' = (2k - n)$, maps each of the $2^{k'}$ superimposed states of a k'-qubit information word onto a unique coset of the dual code $C^{\perp}$ in the code space of C. The cosets of $C^{\perp}$ in C may be obtained by adding a legitimate codeword of C to all the codewords of $C^{\perp}$, as previously shown in Eq. (96). However, only those codewords of C generate a unique coset of $C^{\perp}$, which do not differ by an element of $C^{\perp}$. Explicitly, the codewords x_1 and x_1' of C are said to differ by an element of $C^{\perp}$, if their bit-wise modulo-2 addition yields a codeword of $C^{\perp}$, i.e., $x_1 + x_1' = x_2$, where $x_2 \in C^{\perp}$. Consequently, such codewords of C yield the same coset of $C^{\perp}$.

Let us elaborate on this by constructing the single-error correcting QBCH[15,7] code from the dual-containing classical BCH(15, 11) code of Fig. 32, whose PCM is:

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}. \quad (119)$$

The encoder of QBCH[15, 7] may be derived using the method conceived by MacKay *et al.* [164], which proceeds as follows:

1) The classical dual-containing PCM $\mathbf{H}$ is first transformed into the matrix $\tilde{\mathbf{H}} = [\mathbf{I}_{(n-k)}|\mathbf{P}]$ using elementary row operations as well as column permutations. Explicitly, the elementary row operations include row permutations and addition of one row to the other. Since $\mathbf{H}$ is an $(n - k) \times n$ matrix, the resultant matrix $\mathbf{I}_{(n-k)}$ has dimensions $(n - k) \times (n - k)$, while $\mathbf{P}$ is an $(n - k) \times k$ binary matrix. For the PCM $\mathbf{H}$ of Eq. (119), we have $\tilde{\mathbf{H}} = \mathbf{H}$.
2) As a next step, apply row operations to $\mathbf{P}$ so that it is reduced to $\tilde{\mathbf{P}} = [\mathbf{I}_{(n-k)}, \mathbf{Q}]$, where $\mathbf{Q}$ is an $(n-k) \times k'$ binary matrix. Therefore, we get

$$\tilde{\mathbf{P}} = \left(\begin{array}{cccc|ccccccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{array}\right). \quad (120)$$

3) The associated encoder may be implemented in two steps, as shown in Fig. 35. In the first step, the matrix $\mathbf{Q}$ acts on the second block of $(n - k) = 4$ auxiliary (or parity) qubits controlled by the last $k' = (2k - n) = 7$ information qubits, which constitute the information word. More explicitly, a Controlled NOT (CNOT) gate acts on the *i*th qubit of the second block of $(n - k)$ qubits, which is controlled by the *j*th information qubit, if $Q_{ij} = 1$. This may be formulated as follows

$$|0\rangle^{\otimes(n-k)}|0\rangle^{\otimes(n-k)}|q\rangle \rightarrow |0\rangle^{\otimes(n-k)}|\mathbf{Q}q\rangle|q\rangle. \quad (121)$$

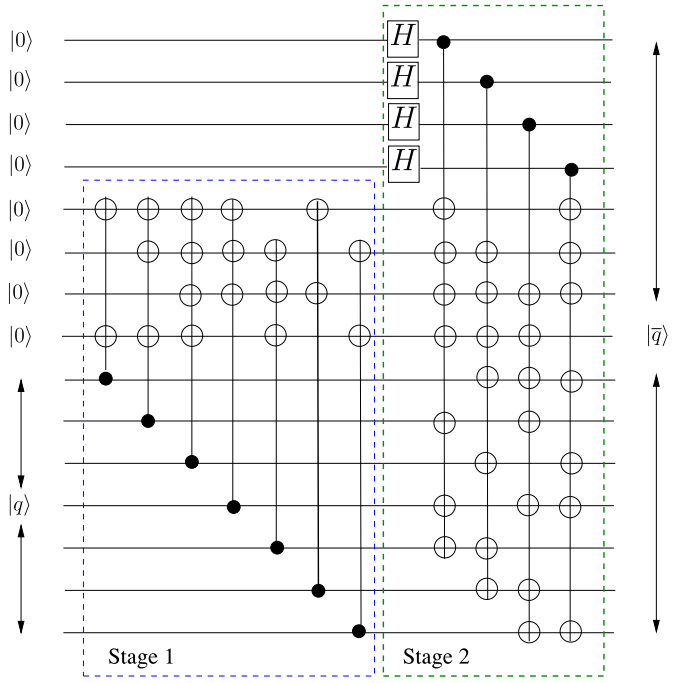


Fig. 35. Encoder of QBCH[15, 7] [178].

TABLE XIV
STABILIZERS OF THE QBCH[15, 7]

	Stabilizer
g_1	ZIIIZZZZIZIZZII
g_2	IZIIIZZZZIZIZZI
g_3	IIZIIIZZZZIZIZZ
g_4	IIIZZZZIZIZZIIZ
g_5	XIIIXXXXIXIXXII
g_6	IXIIIXXXXIXIXXI
g_7	IIXIIIXXXXIXIXX
g_8	IIIXXXXIXIXXIIX

The resultant states constitute the set of codewords in $\mathcal{C}$, which do not differ by any element of $\mathcal{C}^{\perp}$ and therefore are capable of generating unique cosets of $\mathcal{C}^{\perp}$.

4) The second stage adds the codewords of $\mathcal{C}^{\perp}$ to the codewords of $\mathcal{C}$ generated in the previous step. More specifically, the second stage on its own generates the code space of $\mathcal{C}^{\perp}$ according to the PCM $\mathbf{H}$. For a classical code $\mathcal{C}^{\perp}$, the first $(n-k)$ bits are the systematic information bits, which can have either the value of 0 or 1. Consequently, the first $(n-k)=4$ auxiliary qubits undergo a Hadamard transformation for the sake of generating the complete code space of the classical code $\mathcal{C}^{\perp}$. Finally, the matrix $\mathbf{P}$ acts on the last k qubits controlled by the first $(n-k)$ qubits, hence generating the code space of $\mathcal{C}^{\perp}$. More explicitly, a CNOT gate acts on the jth qubit, which is controlled by the ith qubit, if $P_{ij}=1$.

The stabilizers of the QBCH[15, 7] code are constructed using the PCM of Eq. (119) by replacing the 1's with **Z** (or **X**), while the 0's are replaced with **I**. The resultant stabilizer generators are listed in Table XIV. Furthermore, due to the cyclic nature of BCH codes, both the encoder of Fig. 35 as well as the stabilizer generators of Table XIV can be implemented using quantum shift registers,[22] which in turn makes the QBCH codes suitable for systems having cyclic symmetries, for example circular ion traps [179]. The binary syndrome values obtained by applying the stabilizers of Table XIV are then fed to a classical Berlekamp-Massey decoder, which estimates the most likely error.

B. Convolutional Codes

1) Classical Convolutional Codes: Recall that an (n, k) block code encodes each block of k information bits independently into n coded bits. By contrast, an (n, k, m) convolutional code exemplified in Fig. 36 encodes the entire information sequence into a single coded sequence. More specifically, each k-bit input is encoded into n bits, so that the encoded output at each time instant also depends on the k information bits received in the m previous time instances. The resultant convolutional code has a memory of m, or equivalently a constraint length of $(m+1)$, which is implemented using linear shift registers. Furthermore, the code is specified by n generator polynomials, which define the topology of modulo-2 gates for generating the required coded sequence. Explicitly, generator polynomials define the connectivity between the current and m previous input sequences, which in turn ensures that the encoded sequence is a legitimate coded sequence.

Let us consider the systematic (2, 1, 2) convolutional code of Fig. 36, which is specified by the following generator polynomials:

$$\begin{aligned} g_0(x) &= 1 \\ g_1(x) &= 1 + x + x^2. \end{aligned} \tag{122}$$

The generator polynomials may also be expressed as a binary vector, where each bit signifies the presence or absence of a link. Consequently, the generator polynomials of Eq. (122) may also be expressed as:

$$\begin{aligned} g_0 &= (100) \\ g_1 &= (111), \end{aligned} \tag{123}$$

which are seen in Fig. 36. We may observe in Eq. (123) that g_0 has a single non-zero entry. This is because of the systematic nature of the code. By contrast, a non-systematic convolutional code would have more than one non-zero term. Consequently, the polynomial g_0 of a non-systematic code would impose more constraints on the encoded sequence, hence resulting in a more powerful code.

Let us consider a 10-bit input sequence $d = 0011011000$, which may also be represented as $d(x) = x^2 + x^3 + x^5 + x^6$. This input sequence is encoded into a 20-bit coded sequence using the encoder of Fig. 36. The associated encoding process is illustrated in Table XV. More explicitly, the shift register is initialized to the all-zero state. With each clock cycle, the state of register r_0 is updated with the incoming information bit, while its previous value is shifted to the next register r_1. Furthermore, the incoming information bit d_i constitutes the systematic part of the coded bit c, while the output of the modulo-2 adder of Fig. 36 constitutes the parity part.

[22]Please note that implementation of quantum circuits is beyond the scope of this paper.

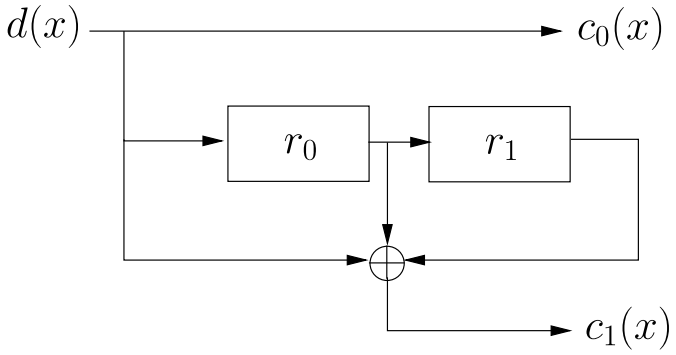


Fig. 36. Schematic of the systematic (2, 1, 2) convolutional encoder.

TABLE XV
SYSTEMATIC (2, 1, 2) CONVOLUTIONAL CODE ENCODING PROCESS FOR $d = 0011011000$ ($d(x) = x^2 + x^3 + x^5 + x^6$), WHICH YIELDS THE CODEWORD $c = 01001010001011000000$ ($c(x) = x + x^4 + x^6 + x^{10} + x^{12} + x^{13}$)

Index	Input Bit	State ($r_0 r_1$) Binary	Decimal	Output Bits
0	-	00	0	-
1	0	00	0	00
2	0	00	0	00
3	0	00	0	00
4	1	10	0	11
5	1	11	2	10
6	0	01	3	00
7	1	10	1	10
8	1	11	2	10
9	0	01	3	00
10	0	00	1	01

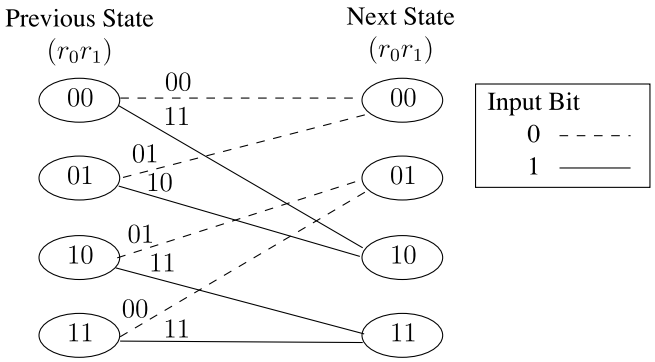


Fig. 37. State transition diagram for systematic (2, 1, 2) convolutional code. *Broken lines indicate legitimate transitions due to a* 0*-valued input, while continuous lines represent a* 1*-valued input. Furthermore, transitions are labeled with the coded bits* $(c_0 c_1)$.

Analogous to BCH codes, the encoding operation of a convolution code may also be characterized using a state transition diagram, as demonstrated in Fig. 37 for the (2, 1, 2) convolutional code of Fig. 36. Consequently, convolutional codes invoke trellis decoding techniques, for example the Viterbi [62] or MAP [64] algorithm, whose decoding complexity is proportional to the number of trellis states 2^m.

2) Quantum Convolutional Codes: Quantum Convolutional Codes (QCCs) may be designed from the classical convolutional codes by exploiting their semi-infinite block nature. Explicitly, convolutional codes may be represented as linear block codes having a semi-infinite length [180]. This equivalence in turn helps in constructing the stabilizer based counterparts of the known classical codes.

Let us first elaborate on the semi-infinite block structure of convolutional codes using a (2, 1, *m*) classical convolutional code having the generators:

$$g_0 = \left(g_0^{(0)} g_0^{(1)} \dots g_0^{(m)}\right)$$
$$g_1 = \left(g_1^{(0)} g_1^{(1)} \dots g_1^{(m)}\right). \tag{124}$$

In essence, the generator polynomials g_0 and g_1 describe the encoder's impulse response functions, which are convolved with the input sequence $[d = (d_0 d_1 d_2 \dots)]$ to yield the encoded bit sequences $[c_0 = (c_0^{(0)} c_0^{(1)} c_0^{(2)} \dots)]$ and $[c_1 = (c_1^{(0)} c_1^{(1)} c_1^{(2)} \dots)]$, respectively. This encoding process can be mathematically encapsulated as:

$$c_0 = d \circledast g_0$$
$$c_1 = d \circledast g_1, \tag{125}$$

where $\circledast$ represents discrete convolution (modulo 2). The convolution process of Eq. (125) may also be expressed as:

$$c_j^{(l)} = \sum_{i=0}^{m} d_{l-i} g_j^{(i)} = d_l g_j^{(0)} + d_{l-1} g_j^{(1)} + \dots + d_{l-m} g_j^{(m)}, \tag{126}$$

where $j = 0, 1$, $l \geq 0$ and $u_{l-i} \triangleq 0$ for all $l < i$. Finally, the pair of encoded sequences c_0 and c_1 are multiplexed, yielding a single encoded sequence c as follows:

$$c = \left(c_0^{(0)} c_1^{(0)} c_0^{(1)} c_1^{(1)} c_0^{(2)} c_1^{(2)} \dots\right). \tag{127}$$

The encoding process of Eq. (126) can also be represented in matrix notation as follows:

$$c = d\mathbf{G}, \tag{128}$$

where the generator matrix $\mathbf{G}$ is constructed g_1 as follows[23]:

$$\mathbf{G} = \begin{pmatrix} g_{01}^{(0)} & g_{01}^{(1)} & \dots & g_{01}^{(m)} & & \\ & g_{01}^{(0)} & g_{01}^{(1)} & \dots & g_{01}^{(m)} & \\ & & g_{01}^{(0)} & g_{01}^{(1)} & \dots & g_{01}^{(m)} \\ & & \ddots & & \dots & \ddots \end{pmatrix}, \tag{129}$$

and $g_{01}^{(i)} \triangleq (g_0^{(i)} g_1^{(i)})$. The resultant matrix $\mathbf{G}$ of Eq. (129) has a semi-infinite length, since the input sequence *d* may have an arbitrary length. Furthermore, we may observe that the *i*th row of $\mathbf{G}$ is obtained by shifting the $(i-1)$th row to the right by $(n = 2)$ places. When *d* is truncated to have a finite length of *N*, then the matrix $\mathbf{G}$ of Eq. (129) is of size $(N \times 2(m + N))$. For a more general convolutional code, having the parameters (*n, k, m*), the generator matrix $\mathbf{G}$ can be expressed as:

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}^{(0)} & \mathbf{G}^{(1)} & \dots & \mathbf{G}^{(m)} & & \\ & \mathbf{G}^{(0)} & \mathbf{G}^{(1)} & \dots & \mathbf{G}^{(m)} & \\ & & \mathbf{G}^{(0)} & \mathbf{G}^{(1)} & \dots & \mathbf{G}^{(m)} \\ & & \ddots & & \dots & \ddots \end{pmatrix}, \tag{130}$$

[23]Zeros indicate blank spaces in the matrix.

TABLE XVI
DESIGN GUIDELINES FOR CONSTRUCTING STABILIZER CODES

Code Type	Parity Check Matrix	Design Criteria	Design Examples: Classical	Design Examples: Quantum	Decoder
Dual-containing CSS	$\left(\begin{array}{c\|c} \mathbf{H}'_z & \mathbf{0} \\ \mathbf{0} & \mathbf{H}'_z \end{array}\right)$	$\mathbf{H}'_z \mathbf{H}'^{T}_z = 0$	$(7,4)$ Hamming code $(15,11)$ BCH $(3,1,2)$ CC	$[7,1]$ Steane's code (Section VIII-B) $[15,7]$ QBCH code (Section VIII-A) $[3,1,2]$ QCC (Section VIII-B)	Binary
Non-dual-containing CSS	$\left(\begin{array}{c\|c} \mathbf{H}'_z & \mathbf{0} \\ \mathbf{0} & \mathbf{H}'_x \end{array}\right)$	$\mathbf{H}'_z \neq \mathbf{H}'_x$ and $\mathbf{H}'_z \mathbf{H}'^{T}_x = 0$	$(3,1)$ Repetition code	$[9,1]$Shor's code (Section VI-A)	Binary
Non-CSS	$(\mathbf{H}_z \vert \mathbf{H}_x)$	$\mathbf{H}_z \mathbf{H}_x^T + \mathbf{H}_x \mathbf{H}_z^T = 0$	$(5,3)$ Non-binary Hamming code $(3,1,2)$ Non-binary CC	$[5,1]$ Hamming code (Section VII-B) $[3,1,2]$ QCC (Section VIII-B)	Non-Binary
EA	$\left(\begin{array}{c\|c} \mathbf{H}'_z & \mathbf{0} \\ \mathbf{0} & \mathbf{H}'_z \end{array}\right)$ and $(\mathbf{H}_z \vert \mathbf{H}_x)$	Minimize the number of pre-shared qubits	-	Random EA-QSC (Section VII-C)	Binary & Non-Binary

where $\mathbf{G}^{(l)}$ is defined as:

$$\mathbf{G}^{(l)} = \begin{pmatrix} g_{1,1}^{(l)} & g_{1,2}^{(l)} & \cdots & g_{1,n-1}^{(l)} \\ g_{2,1}^{(l)} & g_{2,2}^{(l)} & \cdots & g_{2,n-1}^{(l)} \\ \vdots & \vdots & & \vdots \\ g_{k,1}^{(l)} & g_{k,2}^{(l)} & \cdots & g_{k,n-1}^{(l)} \end{pmatrix}. \tag{131}$$

The PCM $\mathbf{H}$ of a convolutional code can also be expressed as a semi-infinite matrix similar to the generator matrix $\mathbf{G}$ of Eq. (130), as shown below:

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}^{(0)} & & & & & \\ \mathbf{H}^{(1)} & \mathbf{H}^{(0)} & & & & \\ \mathbf{H}^{(2)} & \mathbf{H}^{(1)} & \mathbf{H}^{(0)} & & & \\ \vdots & \vdots & \vdots & & & \\ \mathbf{H}^{(m)} & \mathbf{H}^{(m-1)} & \mathbf{H}^{(m-2)} & \cdots & \mathbf{H}^{(0)} & \\ & \mathbf{H}^{(m)} & \mathbf{H}^{(m-1)} & \mathbf{H}^{(m-2)} & \cdots & \mathbf{H}^{(0)} \\ & & \vdots & \vdots & & \vdots \end{pmatrix}, \tag{132}$$

where $\mathbf{H}^{(l)}$ is a submatrix of size an $((n-k) \times n)$. The PCM $\mathbf{H}$ of Eq. (132) exhibits a block-band structure, which is also illustrated in Fig. 38. More specifically, if each row of submatrices $(\mathbf{H}^{(m)}\mathbf{H}^{(m-1)}\mathbf{H}^{(m-2)} \ldots \mathbf{H}^{(0)})$ is viewed as a single block, then $\mathbf{H}$ has a block-band structure, so that each block is a time-shifted version of the previous block and the successive blocks have *m* overlapping submatrices. This block-band structure, which appears after the first *m* blocks, may be expressed as:

$$h_{j,i} = \Big[\mathbf{0}^{j \times n}, h_{0,i}\Big],\ 1 \leq i \leq (n-k),\ 0 \leq j, \tag{133}$$

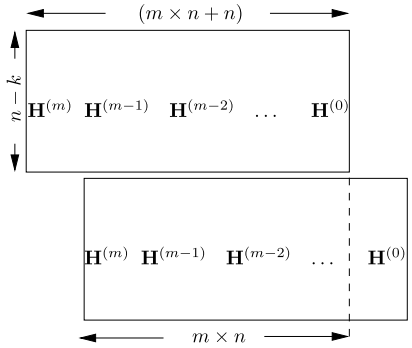


Fig. 38. Semi-infinite classical PCM $\mathbf{H}$ having a block-band structure.

where *i* denotes the row index within a block, while *j* is for the block index. Furthermore, $\mathbf{0}^{j \times n}$ is an all-zero row-vector of size $(j \times n)$. In duality to Eq. (133), the stabilizer group $\mathcal{H}$ of an [*n, k, m*] QCC may be formulated as [167]:

$$\mathcal{H} = sp\Big\{g_{j,i} = I^{\otimes jn} \otimes g_{0,i}\Big\},\ 1 \leq i \leq (n-k),\ 0 \leq j, \tag{134}$$

where *sp* denotes a symplectic group.

Let us now design a CSS-type rate-1/3 QCC [168], [169] from a classical self-dual rate-2/3 binary convolution code having the PCM:

$$\mathbf{H} = \left(\begin{array}{ccc|ccc|ccc|ccc|c} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & \cdots \\ & & & & \cdots & & & & & & & & \end{array}\right), \tag{135}$$

and a minimum distance of 3. The corresponding **X** and **Z** stabilizers of a CSS-type QCC may be obtained by replacing the 1's of Eq. (135) with Pauli **X** and **Z** operators, respectively. Hence, the stabilizers of the resultant [3, 1] QCC are:

$$g_{0,1} = [\mathbf{XXX}, \mathbf{XII}, \mathbf{XXI}], \tag{136}$$

$$g_{0,2} = [\mathbf{ZZZ}, \mathbf{ZII}, \mathbf{ZZI}], \tag{137}$$

which can correct a single error. The associated stabilizer group $\mathcal{H}$ may be constructed using Eq. (134).

Next, we design a non-CSS, or more precisely CRSS, QCC given by Forney in [168] and [169]. It is constructed from the classical rate-2/3 quaternary convolutional code having the PCM:

$$\mathbf{H} = \left(\begin{array}{ccc|ccc|ccc|c} 1 & 1 & 1 & 1 & w & \bar{w} & 0 & 0 & 0 & \ldots \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & w & \bar{w} & \ldots \\ & & & & \ldots & & & & & \end{array}\right), \tag{138}$$

which is self-orthogonal. The stabilizers of the corresponding [3, 1] QCC may be constructed using Eq. (101). Explicitly, the stabilizers $g_{0,i}$, for $1 \leq i \leq 2$, are obtained by multiplying the **H** of Eq. (138) with the GF(4) elements w and $\bar{w}$, and mapping the resultant GF(4) elements onto the Pauli operators. Hence, the resultant stabilizers are:

$$g_{0,1} = (\mathbf{XXX}, \mathbf{XZY}), \tag{139}$$

$$g_{0,2} = (\mathbf{ZZZ}, \mathbf{ZYX}). \tag{140}$$

Analogous to other stabilizer codes, the binary syndrome values obtained using the stabilizers of a QCC are fed to a classical syndrome decoder. However, classical convolutional codes generally employ either the Viterbi [62] or the MAP [64] decoding algorithm operating over a code trellis for the sake of estimating the most likely codeword. By contrast, QCCs invoke the syndrome-based error trellis [181]–[185] for estimating the most likely error pattern rather than the most likely codeword. Explicitly, unlike the classic trellis of a convolutional code seen in Fig. 37, which is constructed using the encoding circuit, syndrome-based trellis is constructed using the PCM **H** of Eq. (132). Furthermore, the conventional trellis, for example the one obtained using the state transition diagram of Fig. 37, is known as a code trellis, because each path of it is a valid codeword. By contrast, each path of the error trellis is a legitimate error sequence for a given observed syndrome. Therefore, a code trellis is used for codeword decoding, while an error trellis is used for syndrome decoding. However, both trellis representations are equivalent, since every path in the error trellis corresponds to a path in the code trellis. Furthermore, a degenerate Viterbi decoding algorithm was also conceived for QCCs in [135], which takes into account degenerate quantum errors, hence improving the decoding process.

IX. Conclusion & Design Guidelines

QECCs are essential for rectifying the undesirable perturbations resulting from quantum decoherence. Unfortunately, the well-developed classical coding theory, which has evolved over seven decades, cannot be directly applied to the quantum regime. Explicitly, unlike a classical bit, a qubit cannot be copied and it collapses to a classical bit upon measurement. Furthermore, while bit flips are the only type of errors experienced during transmission over a classical channel, a quantum channel may inflict both bit-flips as well as phase-flips. Therefore, it is not feasible to directly map classical codes onto their quantum counterparts. Nevertheless, quantum codes may be designed from the existing classical codes by exploiting the subtle similarities between these two coding regimes. In particular, as detailed in Section II, quantum decoherence may be modeled using the quantum depolarizing channel, which is deemed equivalent to a pair of binary symmetric channels, or more specifically to a classical 4-ary channel. This similarity has helped researchers to develop the quantum versions of the known classical codes, as evident from our survey of Section III. For the sake of providing deeper insights into the transition from classical to quantum coding theory, we started our discussions in Section IV with a simple repetition code, which brought forth three fundamental design principles:

- The copying operation of classical codes is equivalent to quantum entanglement;
- Measurement of a qubit may be circumvented by invoking the classical syndrome decoding techniques;
- Phase-flips may be corrected by using the Hadamard basis.

Based on these design principles, we detailed the stabilizer formalism in Section V, which is in essence the quantum-domain counterpart of classical linear block codes. Since most of the classical codes rely on the basic construction of linear block codes, the stabilizer formalism has helped researchers to build on most of the known families of classical codes. In Section VI, we detailed the equivalence between the quantum and classical parity check matrices, focusing specifically on the Pauli-to-binary isomorphism as well as on the Pauli-to-quaternary isomorphism. The Pauli-to-binary isomorphism helps in designing quantum codes from arbitrary classical binary codes, if they meet the symplectic product criterion, while the Pauli-to-quaternary isomorphism allows us to harness arbitrary classical quaternary codes, if they satisfy the Hermitian inner product. Furthermore, based on this isomorphism, we presented the taxonomy of stabilizer codes in Section VII, namely the dual-containing and non-dual-containing Calderbank-Shor-Steane (CSS) codes non-CSS codes and entanglement-assisted codes, which are summarized in Table XVI. Finally, in Section VIII, we applied our discussions to a pair of popular code families of the classical world, namely the BCH codes and the convolutional codes, for designing their quantum counterparts.

References

[1] P. A. Dirac, *The Principles of Quantum Mechanics*. London, U.K.: Oxford Univ., 1982.

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

[3] 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, 2015.

[4] D. Deutsch, "Quantum theory, the church-turing principle and the universal quantum computer," *Proc. Roy. Soc. London A Math. Phys. Eng. Sci.*, vol. 400, no. 1818, pp. 97–117, 1985.

[5] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," *Proc. Roy. Soc. London A Math. Phys. Eng. Sci.*, vol. 439, no. 1907, pp. 553–558, 1992.

[6] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, Santa Fe, NM, USA, 1994, pp. 124–134. [Online]. Available: http://portal.acm.org/citation.cfm?id=1398518.1399018

[7] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput. (STOC)*, Philadelphia, PA, USA, 1996, pp. 212–219. [Online]. Available: http://doi.acm.org/10.1145/237814.237866

[8] M. Born, *The Born-Einstein Letters*. New York, NY, USA: Walker, 1971.

[9] P. Botsinis, D. Alanis, Z. Babar, S. X. Ng, and L. Hanzo, "Noncoherent quantum multiple symbol differential detection for wireless systems," *IEEE Access*, vol. 3, pp. 569–598, 2015.

[10] P. Botsinis *et al.*, "Quantum-aided multi-user transmission in non-orthogonal multiple access systems," *IEEE Access*, vol. 4, pp. 7402–7424, 2016.

[11] D. Alanis, P. Botsinis, Z. Babar, S. X. Ng, and L. Hanzo, "Non-dominated quantum iterative routing optimization for wireless multihop networks," *IEEE Access*, vol. 3, pp. 1704–1728, 2015.

[12] D. Alanis *et al.*, "Quantum-assisted joint multi-objective routing and load balancing for socially-aware networks," *IEEE Access*, vol. 4, pp. 9993–10028, 2016.

[13] T. J. Hastie, R. J. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2009. [Online]. Available: http://opac.inria.fr/record=b1127878

[14] J. Lu, G. Wang, and P. Moulin, "Human identity and gender recognition from gait sequences with arbitrary walking directions," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 1, pp. 51–61, Jan. 2014.

[15] D. S. Matovski, M. S. Nixon, S. Mahmoodi, and J. N. Carter, "The effect of time on gait recognition performance," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 543–552, Apr. 2012.

[16] S. Imre and F. Balazs, *Quantum Computing and Communications: An Engineering Approach*. Hoboken, NJ, USA: Wiley, 2005.

[17] S. Wiesner, "Conjugate coding," *SIGACT News*, vol. 15, no. 1, pp. 78–88, Jan. 1983. [Online]. Available: http://doi.acm.org/10.1145/1008908.1008920

[18] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *Proc. IEEE Int. Conf. Comput. Syst. Signal Process.*, 1984, pp. 175–179.

[19] A. Beige, B.-G. Englert, C. Kurtsiefer, and H. Weinfurter, "Secure communication with single-photon two-qubit states," *J. Phys. A Math. Gen.*, vol. 35, no. 28, Jul. 2002, Art. no. L407. [Online]. Available: http://stacks.iop.org/0305-4470/35/i=28/a=103

[20] K. Boström and T. Felbinger, "Deterministic secure direct communication using entanglement," *Phys. Rev. Lett.*, vol. 89, Oct. 2002, Art. no. 187902. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.89.187902

[21] C. Wang, F.-G. Deng, Y.-S. Li, X.-S. Liu, and G. L. Long, "Quantum secure direct communication with high-dimension quantum superdense coding," *Phys. Rev. A*, vol. 71, Apr. 2005, Art. no. 044305. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.71.044305

[22] R. A. Malaney, "Location-dependent communications using quantum entanglement," *Phys. Rev. A*, vol. 81, Apr. 2010, Art. no. 042319. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.81.042319

[23] R. Malaney, "The quantum car," *IEEE Wireless Commun. Lett.*, vol. 5, no. 6, pp. 624–627, Dec. 2016.

[24] R. Malaney, "Quantum geo-encryption," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6, doi: 10.1109/GLOCOM.2016.7842191.

[25] H. J. Kimble, "The quantum Internet," *Nature*, vol. 453, no. 7198, pp. 1023–1030, Jun. 2008. [Online]. Available: http://dx.doi.org/10.1038/nature07127

[26] L. Jiang, J. M. Taylor, A. S. Sørensen, and M. D. Lukin, "Distributed quantum computation based on small quantum registers," *Phys. Rev. A*, vol. 76, Dec. 2007, Art. no. 062323. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.76.062323

[27] C. Monroe *et al.*, "Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects," *Phys. Rev. A*, vol. 89, Feb. 2014, Art. no. 022317. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.89.022317

[28] S. Muralidharan *et al.*, "Optimal architectures for long distance quantum communication," *Sci. Rep.*, vol. 6, Feb. 2016, Art. no. 20463. [Online]. Available: http://dx.doi.org/10.1038/srep20463

[29] M. Takeoka, S. Guha, and M. M. Wilde, "Fundamental rate-loss tradeoff for optical quantum key distribution," *Nat. Commun.*, vol. 5, p. 5235, Oct. 2014. [Online]. Available: http://dx.doi.org/10.1038/ncomms6235

[30] Z. Babar, S. X. Ng, and L. Hanzo, "Near-capacity code design for entanglement-assisted classical communication over quantum depolarizing channels," *IEEE Trans. Commun.*, vol. 61, no. 12, pp. 4801–4807, Dec. 2013.

[31] Z. Babar, S. X. Ng, and L. Hanzo, "EXIT-chart aided code design for symbol-based entanglement-assisted classical communication over quantum channels," in *Proc. IEEE Veh. Technol. Conf. (VTC Fall)*, Vancouver, BC, Canada, Sep. 2014, pp. 1–5.

[32] C. H. Bennett and S. J. Wiesner, "Communication via one- and two-particle operators on Einstein–Podolsky–Rosen states," *Phys. Rev. Lett.*, vol. 69, no. 20, pp. 2881–2884, 1992. [Online]. Available: http://dx.doi.org/%7B10.1103/PhysRevLett.69.2881%7D

[33] G. Brassard and L. Salvail, *Secret-Key Reconciliation by Public Discussion*. Berlin, Heidelberg: Springer, 1994, pp. 410–423.

[34] W. T. Buttler *et al.*, "Fast, efficient error reconciliation for quantum cryptography," *Phys. Rev. A*, vol. 67, May 2003, Art. no. 052303. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.67.052303

[35] D. G. Cory *et al.*, "Experimental quantum error correction," *Phys. Rev. Lett.*, vol. 81, pp. 2152–2155, Sep. 1998. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.81.2152

[36] M. D. Reed *et al.*, "Realization of three-qubit quantum error correction with superconducting circuits," *Nature*, vol. 482, no. 7385, pp. 382–385, Feb. 2012. [Online]. Available: http://dx.doi.org/10.1038/nature10786

[37] G. Arrad, Y. Vinkler, D. Aharonov, and A. Retzker, "Increasing sensing resolution with error correction," *Phys. Rev. Lett.*, vol. 112, no. 15, Apr. 2014, Art. no. 150801. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.112.150801

[38] I. L. Chuang, D. W. Leung, and Y. Yamamoto, "Bosonic quantum codes for amplitude damping," *Phys. Rev. A*, vol. 56, pp. 1114–1125, Aug. 1997. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.56.1114

[39] J. Preskill, *Lecture Notes for Physics 229: Quantum Information and Computation*, vol. 16, California Inst. Technol., 1998.

[40] J. Ghosh, A. G. Fowler, and M. R. Geller, "Surface code with decoherence: An analysis of three superconducting architectures," *Phys. Rev. A*, vol. 86, Dec. 2012, Art. no. 062318. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.86.062318

[41] P. K. Sarvepalli, A. Klappenecker, and M. Rötteler, "Asymmetric quantum codes: Constructions, bounds and performance," *Proc. Roy. Soc. London A Math. Phys. Eng. Sci.*, vol. 465, no. 2105, pp. 1645–1672, 2009. [Online]. Available: http://rspa.royalsocietypublishing.org/content/465/2105/1645

[42] L. M. K. Vandersypen *et al.*, "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," *Nature*, vol. 414, no. 6866, pp. 883–887, 2001.

[43] H. V. Nguyen *et al.*, "EXIT-chart aided quantum code design improves the normalised throughput of realistic quantum devices," *IEEE Access*, vol. 4, pp. 10194–10209, 2016.

[44] C. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

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

[46] I. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *Trans. IRE Prof. Group Inf. Theory*, vol. 4, no. 4, pp. 38–49, Sep. 1954.

[47] D. E. Muller, "Application of Boolean algebra to switching circuit design and to error detection," *Trans. IRE Prof. Group Electron. Comput.*, vol. EC-3, no. 3, pp. 6–12, Sep. 1954.

[48] R. Silverman and M. Balser, "Coding for constant-data-rate systems," *Trans. IRE Prof. Group Inf. Theory*, vol. 4, no. 4, pp. 50–63, Sep. 1954.

[49] P. Elias, "Coding for noisy channels," in *IRE International Convention Record*. New York, NY, USA: Institute, 1955, pp. 37–46.

[50] E. Prange, *Cyclic Error-Correcting Codes in Two Symbols*. Cambridge, MA, USA: Air Force Cambridge Res. Center, 1957.

[51] A. Hocquenghem, "Codes Correcteurs d'Erreurs," *Chiffres (Paris)*, vol. 2, pp. 147–156, Sep. 1959.

[52] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Inf. Control*, vol. 3, no. 1, pp. 68–79, 1960.

[53] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 2, pp. 300–304, 1960.

[54] D. Gorenstein and N. Zierler, "A class of error-correcting codes in p^m symbols," *J. Soc. Ind. Appl. Math.*, vol. 9, no. 2, pp. 207–214, 1961.

[55] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[56] E. Berlekamp, "On decoding binary Bose–Chadhuri–Hocquenghem codes," *IEEE Trans. Inf. Theory*, vol. 11, no. 4, pp. 577–579, Oct. 1965.

[57] E. R. Berlekamp, *Algebraic Coding Theory*. New York, NY, USA: McGraw-Hill, 1968.

[58] J. Massey, "Step-by-step decoding of the Bose–Chaudhuri–Hocquenghem codes," *IEEE Trans. Inf. Theory*, vol. 11, no. 4, pp. 580–585, Oct. 1965.

[59] J. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inf. Theory*, vol. IT-15, no. 1, pp. 122–127, Jan. 1969.

[60] R. W. Watson and C. W. Hastings, "Self-checked computation using residue arithmetic," *Proc. IEEE*, vol. 54, no. 12, pp. 1920–1931, Dec. 1966.

[61] N. S. Szabo and R. I. Tanaka, *Residue Arithmetic and Its Applications to Computer Technology*. New York, NY, USA: McGraw-Hill, 1967.

[62] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 2, pp. 260–269, Apr. 1967.

[63] D. Chase, "Class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 1, pp. 170–182, Jan. 1972.

[64] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 284–287, Mar. 1974.

[65] J. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 1, pp. 76–80, Jan. 1978.

[66] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 1, pp. 55–67, Jan. 1982.

[67] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets part I: Introduction," *IEEE Commun. Mag.*, vol. CM-25, no. 2, pp. 5–11, Feb. 1987.

[68] G. Ungerboeck, "Trellis-coded modulation with redundant signal sets part II: State of the art," *IEEE Commun. Mag.*, vol. CM-25, no. 2, pp. 12–21, Feb. 1987.

[69] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. IEEE Glob. Telecommun. Conf. (GLOBECOM)*, 1989, pp. 1680–1686.

[70] W. Koch and A. Baier, "Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference (applicable to digital mobile radio receivers)," in *Proc. IEEE Glob. Telecommun. Conf. (GLOBECOM)*, San Diego, CA, USA, 1990, pp. 1679–1684.

[71] E. Zevahi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Trans. Commun.*, vol. 40, no. 5, pp. 873–884, May 1992.

[72] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 927–946, May 1998.

[73] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proc. Tech. Program IEEE Int. Conf. Commun. (ICC)*, vol. 2. Geneva, Switzerland, May 1993, pp. 1064–1070.

[74] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[75] R. Pyndiah, A. Glavieux, A. Picart, and S. Jacq, "Near optimum decoding of product codes," in *Proc. IEEE Glob. Telecommun. Conf. (GLOBECOM)*, San Francisco, CA, USA, 1994, pp. 339–343.

[76] R. M. Pyndiah, "Near-optimum decoding of product codes: Block turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003–1010, Aug. 1998.

[77] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 2. Seattle, WA, USA, 1995, pp. 1009–1013.

[78] D. J. C. MacKay and R. M. Neal, "Good codes based on very sparse matrices," *Cryptography and Coding*. Berlin, Germany: Springer, 1995, pp. 100–111.

[79] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 32, no. 18, p. 1645, Aug. 1996.

[80] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A*, vol. 52, no. 4, pp. R2493–R2496, Oct. 1995. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.52.R2493

[81] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

[82] A. Steane, "Multiple-particle interference and quantum error correction," *Roy. Soc. London Proc. A*, vol. 452, no. 1954, pp. 2551–2577, Nov. 1995.

[83] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Phys. Rev. A*, vol. 54, no. 2, pp. 1098–1105, Aug. 1996.

[84] A. M. Steane, "Error correcting codes in quantum theory," *Phys. Rev. Lett.*, vol. 77, no. 5, pp. 793–797, Jul. 1996.

[85] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, "Mixed-state entanglement and quantum error correction," *Phys. Rev. A*, vol. 54, no. 5, pp. 3824–3851, Nov. 1996. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.54.3824

[86] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, "Perfect quantum error correcting code," *Phys. Rev. Lett.*, vol. 77, no. 1, pp. 198–201, Jul. 1996. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.77.198

[87] D. Gottesman, "Class of quantum error-correcting codes saturating the quantum Hamming bound," *Phys. Rev. A*, vol. 54, no. 3, pp. 1862–1868, Sep. 1996.

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

[89] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. 29th Annu. ACM Symp. Theory Comput.*, El Paso, TX, USA, 1997, pp. 150–159.

[90] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[91] H. Nickl, J. Hagenauer, and F. Burkert, "Approaching Shannon's capacity limit by 0.2 dB using simple Hamming codes," *IEEE Commun. Lett.*, vol. 1, no. 5, pp. 130–132, Sep. 1997.

[92] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Lett.*, vol. 1, no. 6, pp. 169–171, Nov. 1997.

[93] S. Lloyd, "Capacity of the noisy quantum channel," *Phys. Rev. A*, vol. 55, pp. 1613–1622, Mar 1997. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.55.1613

[94] A. M. Steane, "Simple quantum error-correcting codes," *Phys. Rev. A*, vol. 54, no. 6, pp. 4741–4751, 1996.

[95] M. Grassl, T. Beth, and T. Pellizzari, "Codes for the quantum erasure channel," *Phys. Rev. A*, vol. 56, pp. 33–38, Jul. 1997. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.56.33

[96] A. R. Calderbank, E. M. Rains, P. M. 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.

[97] M. Grassl and T. Beth, "Quantum BCH codes," in *Proc. Int. Symp. Theor. Elect. Eng. Magdeburg*, Oct. 1999, pp. 207–212. [Online]. Available: http://arxiv.org/abs/quant-ph/9910060

[98] A. M. Steane, "Enlargement of Calderbank–Shor–Steane quantum codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 7, pp. 2492–2495, Nov. 1999.

[99] L. Xiaoyan, "Quantum cyclic and constacyclic codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 3, pp. 547–549, Mar. 2004.

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

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

[102] K. Fujii, *Quantum Computation With Topological Codes: From Qubit to Topological Fault-Tolerance*, vol. 8. Singapore: Springer, 2015.

[103] P. Robertson and T. Worz, "Bandwidth-efficient turbo trellis-coded modulation using punctured component codes," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 206–218, Feb. 1998.

[104] A. J. Felstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181–2191, Sep. 1999.

[105] O. F. Acikel and W. E. Ryan, "Punctured turbo-codes for BPSK/QPSK channels," *IEEE Trans. Commun.*, vol. 47, no. 9, pp. 1315–1323, Sep. 1999.

[106] A. M. Steane, "Quantum Reed–Muller codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1701–1703, Jul. 1999.

[107] M. Grassl, W. Geiselmann, and T. Beth, "Quantum Reed–Solomon codes," in *Proc. Appl. Algebra Algebraic Algorithms Error Correcting Codes (AAECC)*, 1999, pp. 231–244.

[108] D. Divsalar, S. Dolinar, and F. Pollara, "Serial concatenated trellis coded modulation with rate-1 inner code," in *Proc. IEEE Glob. Telecommun. Conf.*, San Francisco, CA, USA, Nov. 2000, pp. 777–782.

[109] S. T. Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[110] M. S. Postol, "A proposed quantum low density parity check code," *arXiv:quant-ph/0108131v1*, 2001.

[111] M. Tüchler and J. Hagenauer, "EXIT charts of irregular codes," in *Proc. Conf. Inf. Sci. Syst.*, Princeton, NJ, USA, Mar. 2002, pp. 748–753.

[112] G. Bowen, "Entanglement required in achieving entanglement-assisted channel capacities," *Phys. Rev. A*, vol. 66, Nov. 2002, Art. no. 052313. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.66.052313

[113] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *IPN Progr. Rep.*, vol. 42, no. 154, pp. 42–154, 2003.

[114] H. Ollivier and J.-P. Tillich, "Description of a quantum convolutional code," *Phys. Rev. Lett.*, vol. 91, Oct. 2003, Art. no. 177902. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.91.177902

[115] J. Kliewer, S. X. Ng, and L. Hanzo, "Efficient computation of EXIT functions for non-binary iterative decoding," *IEEE Trans. Commun.*, vol. 54, no. 12, pp. 2133–2136, Dec. 2006.

[116] T. A. Brun, I. Devetak, and M.-H. Hsieh, "Correcting quantum errors with entanglement," *Science*, vol. 314, no. 5798, Oct. 2006.

[117] T. A. Brun, I. Devetak, and M.-H. Hsieh, "General entanglement-assisted quantum error-correcting codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Nice, France, Jun. 2007, pp. 2101–2105.

[118] M.-H. Hsieh, I. Devetak, and T. Brun, "General entanglement-assisted quantum error-correcting codes," *Phys. Rev. A*, vol. 76, Dec. 2007, Art. no. 062313. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.76.062313

[119] S. X. Ng, O. R. Alamri, Y. Li, J. Kliewer, and L. Hanzo, "Near-capacity turbo trellis coded modulation design based on EXIT charts and union bounds," *IEEE Trans. Commun.*, vol. 56, no. 12, pp. 2030–2039, Dec. 2008.

[120] D. Poulin, J.-P. Tillich, and H. Ollivier, "Quantum serial turbo-codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Toronto, ON, Canada, Jul. 2008, pp. 310–314.

[121] 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.

[122] D. Poulin and Y. Chung, "On the iterative decoding of sparse quantum codes," *Quantum Inf. Comput.*, vol. 8, no. 10, pp. 987–1000, Nov. 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=2016985.2016993

[123] Y.-J. Wang, B. C. Sanders, B.-M. Bai, and X.-M. Wang, "Enhanced feedback iterative decoding of sparse quantum codes," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 1231–1241, Feb. 2012.

[124] 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, 2015.

[125] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[126] R. Y. S. Tee, R. G. Maunder, and L. Hanzo, "EXIT-chart aided near-capacity irregular bit-interleaved coded modulation design," *IEEE Trans. Wireless Commun.*, vol. 8, no. 1, pp. 32–37, Jan. 2009.

[127] M.-H. Hsieh, T. A. Brun, and I. Devetak, "Entanglement-assisted quantum quasicyclic low-density parity-check codes," *Phys. Rev. A*, vol. 79, Mar. 2009, Art. no. 032340. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.79.032340

[128] M. M. Wilde and T. A. Brun, "Entanglement-assisted quantum convolutional coding," *Phys. Rev. A*, vol. 81, Apr. 2010, Art. no. 042333. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.81.042333

[129] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.

[130] M. M. Wilde and M.-H. Hsieh, "Entanglement boosts quantum turbo codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Aug. 2011, pp. 445–449.

[131] M. M. Wilde, M.-H. Hsieh, and Z. Babar, "Entanglement-assisted quantum turbo codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 1203–1222, Feb. 2014.

[132] M. M. Wilde and S. Guha, "Polar codes for classical-quantum channels," *IEEE Trans. Inf. Theory*, vol. 59, no. 2, pp. 1175–1187, Feb. 2013.

[133] M. M. Wilde and S. Guha, "Polar codes for degradable quantum channels," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4718–4729, Jul. 2013.

[134] J. M. Renes, F. Dupuis, and R. Renner, "Efficient polar coding of quantum information," *Phys. Rev. Lett.*, vol. 109, Aug. 2012, Art. no. 050504. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.109.050504

[135] E. Pelchat and D. Poulin, "Degenerate Viterbi decoding," *IEEE Trans. Inf. Theory*, vol. 59, no. 6, pp. 3915–3921, Jun. 2013.

[136] Z. Babar, S. X. Ng, and L. Hanzo, "EXIT-chart-aided near-capacity quantum turbo code design," *IEEE Trans. Veh. Technol.*, vol. 64, no. 3, pp. 866–875, Mar. 2015.

[137] R. G. Maunder, "A fully-parallel turbo decoding algorithm," *IEEE Trans. Commun.*, vol. 63, no. 8, pp. 2762–2775, Aug. 2015.

[138] J. M. Renes, D. Sutter, F. Dupuis, and R. Renner, "Efficient quantum polar codes requiring no preshared entanglement," *IEEE Trans. Inf. Theory*, vol. 61, no. 11, pp. 6395–6414, Nov. 2015.

[139] Z. Babar *et al.*, "Serially concatenated unity-rate codes improve quantum codes without coding-rate reduction," *IEEE Commun. Lett.*, vol. 20, no. 10, pp. 1916–1919, Oct. 2016.

[140] Z. Babar *et al.*, "Fully-parallel quantum turbo decoder," *IEEE Access*, vol. 4, pp. 6073–6085, 2016.

[141] L. Hanzo, *Near-Capacity Variable-Length Coding: Regular and EXIT-Chart-Aided Irregular Designs*, vol. 20. Chichester, U.K.: Wiley, 2010.

[142] 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*, 2nd ed. New York, NY, USA: Wiley, 2011.

[143] R. C. Singleton, "Maximum distance Q-nary codes," *IEEE Trans. Inf. Theory*, vol. IT-10, no. 2, pp. 116–118, Apr. 1964.

[144] R. J. McEliece, E. R. Rodemich, H. Rumsey, Jr., and L. R. Welch, "New upper bounds on the rate of a code via the Delsarte–MacWilliams inequalities," *IEEE Trans. Inf. Theory*, vol. IT-23, no. 2, pp. 157–166, Mar. 1977.

[145] M. Plotkin, "Binary codes with specified minimum distance," *IRE Trans. Inf. Theory*, vol. 6, no. 4, pp. 445–450, Sep. 1960.

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

[147] J. Akhtman, R. Maunder, N. Bonello, and L. Hanzo, "Closed-form approximation of maximum free distance for binary block codes," in *Proc. IEEE 70th Veh. Technol. Conf. Fall (VTC-Fall)*, 2009, pp. 1–3.

[148] 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.

[149] R. C. Bose and D. K. Ray-Chaudhuri, "Further results on error correcting binary group codes," *Inf. Control*, vol. 3, no. 3, pp. 279–290, 1960.

[150] *Blue Book: Recommendations for Space Data System Standards: Telemetry Channel Coding*, Consultative Committee for Space Data Syst., May 1984.

[151] M. El-Hajjar and L. Hanzo, "EXIT charts for system design and analysis," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 127–153, 1st Quart., 2014.

[152] *Universal Mobile Telecommunications System (UMTS); Multiplexing and Channel Coding (FDD), V9.3.0*, ETSI Standard TS 125 222, 2012.

[153] *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and Channel Coding, V13.1.0*, ETSI Standard TS 136 212, 2016.

[154] K. Niu, K. Chen, J. Lin, and Q. T. Zhang, "Polar codes: Primary concepts and practical decoding algorithms," *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 192–203, Jul. 2014.

[155] N. Onizawa, T. Hanyu, and V. C. Gaudet, "Design of high-throughput fully parallel LDPC decoders based on wire partitioning," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 3, pp. 482–489, Mar. 2010.

[156] P. W. Shor, "The quantum channel capacity and coherent information," in *Proc. Lecture Notes MSRI Workshop Quantum Comput.*, 2002.

[157] I. Devetak, "The private classical capacity and quantum capacity of a quantum channel," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 44–55, Jan. 2005.

[158] D. P. DiVincenzo, P. W. Shor, and J. A. Smolin, "Quantum-channel capacity of very noisy channels," *Phys. Rev. A*, vol. 57, pp. 830–839, Feb. 1998. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.57.830

[159] G. Smith and J. A. Smolin, "Degenerate quantum codes for Pauli channels," *Phys. Rev. Lett.*, vol. 98, Jan. 2007, Art. no. 030501. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.98.030501

[160] C.-Y. Lai, T. Brun, and M. Wilde, "Dualities and identities for entanglement-assisted quantum codes," *Quantum Inf. Process.*, vol. 13, no. 4, pp. 957–990, 2014, doi: 10.1007/s11128-013-0704-8.

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

[162] A. Ekert and C. Macchiavello, "Quantum error correction for communication," *Phys. Rev. Lett.*, vol. 77, no. 12, pp. 2585–2588, 1996.

[163] A. Ashikhmin and S. Litsyu, "Upper bounds on the size of quantum codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 4, pp. 1206–1215, May 1999.

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

[165] T. Camara, H. Ollivier, and J.-P. Tillich, "Constructions and performance of classes of quantum LDPC codes," *arXiv:quant-ph/0502086v2*, 2005.

[166] 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*, Jun. 2007, pp. 811–815.

[167] H. Ollivier and J. P. Tillich, "Quantum convolutional codes: Fundamentals," *arXiv:quant-ph/0401134*, 2004.

[168] G. D. Forney and S. Guha, "Simple rate-1/3 convolutional and tail-biting quantum error-correcting codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Sep. 2005, pp. 1028–1032.

[169] G. D. Forney, M. Grassl, and S. Guha, "Convolutional and tail-biting quantum error-correcting codes," *IEEE Trans. Inf. Theory*, vol. 53, no. 3, pp. 865–880, Mar. 2007.

[170] M. Houshmand and M. M. Wilde, "Recursive quantum convolutional encoders are catastrophic: A simple proof," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6724–6731, Oct. 2013.

[171] M. Hagiwara, K. Kasai, H. Imai, and K. Sakaniwa, "Spatially coupled quasi-cyclic quantum LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Aug. 2011, pp. 638–642.

[172] K. Kasai, M. Hagiwara, H. Imai, and K. Sakaniwa, "Non-binary quasi-cyclic quantum LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Aug. 2011, pp. 653–657.

[173] K. Kasai, M. Hagiwara, H. Imai, and K. Sakaniwa, "Quantum error correction beyond the bounded distance decoding limit," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 1223–1230, Feb. 2012.

[174] I. Andriyanova, D. Maurice, and J.-P. Tillich, "Spatially coupled quantum LDPC codes," in *Proc. IEEE Inf. Theory Workshop*, Sep. 2012, pp. 327–331.

[175] D. Maurice, J.-P. Tillich, and I. Andriyanova, "A family of quantum codes with performances close to the hashing bound under iterative decoding," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 907–911.

[176] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, no. 5886, pp. 802–803, Oct. 1982, doi: 10.1038/299802a0.

[177] L. Hanzo, T. H. Liew, and B. L. Yeap, *Turbo Coding, Turbo Equalisation and Space-Time Coding*. Hoboken, NJ, USA; Wiley, 2002.

[178] P. Botsinis *et al.*, "Quantum error correction protects quantum search algorithms against decoherence," *Sci. Rep.*, vol. 6, Dec. 2016, Art. no. 38095.

[179] M. Grassl and T. Beth, "Cyclic quantum error–correcting codes and quantum shift registers," in *Proc. Roy. Soc. London A Math. Phys. Eng. Sci.*, vol. 456, no. 2003, pp. 2689–2706, 2000.

[180] S. Lin and D. J. Costello, *Error Control Coding*. Upper Saddle River, NJ, USA: Pearson, 2004.

[181] J. Schalkwijk and A. Vinck, "Syndrome decoding of convolutional codes," *IEEE Trans. Commun.*, vol. COM-23, no. 7, pp. 789–792, Jul. 1975.

[182] J. Schalkwijk and A. Vinck, "Syndrome decoding of binary rate-1/2 convolutional codes," *IEEE Trans. Commun.*, vol. COM-24, no. 9, pp. 977–985, Sep. 1976.

[183] J. Schalkwijk, A. Vinck, and K. Post, "Syndrome decoding of binary-rate k/n convolutional codes," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 5, pp. 553–562, Sep. 1978.

[184] M. Ariel and J. Snyders, "Soft syndrome decoding of binary convolutional codes," *IEEE Trans. Commun.*, vol. 43, nos. 2–4, pp. 288–297, Feb./Apr. 1995.

[185] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *IEEE Trans. Inf. Theory*, vol. 40, no. 5, pp. 1663–1666, Sep. 1994.

Zunaira Babar received the B.Eng. degree in electrical engineering from the National University of Science and Technology, Islamabad, Pakistan, in 2008 and the M.Sc. degree (Distinction) and the Ph.D. degree in wireless communications from the University of Southampton, U.K., in 2011 and 2015, respectively, where she is currently a Research Fellow with the Southampton Wireless Group.

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

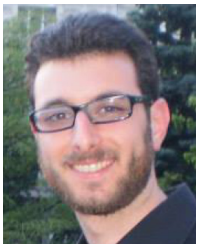
Daryus Chandra (S'13) received the M.Eng. degree in electrical engineering from Universitas Gadjah Mada, Indonesia, in 2014. He is currently pursuing the Ph.D. degree with the Southampton Wireless Group, School of Electronics and Computer Science, University of Southampton. He was a recipient of the Scholarship Award from Indonesia Endowment Fund for Education (LPDP).

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

Hung Viet Nguyen (S'09–M'14) received the B.Eng. degree in electronics and telecommunications from the Hanoi University of Science and Technology, Hanoi, Vietnam, in 1999 and the M.Eng. degree in telecommunications from the Asian Institute of Technology, Bangkok, Thailand, in 2002. Since 1999, he has been a Lecturer with the Post and Telecommunications Institute of Technology, Vietnam. He worked for the OPTIMIX and CONCERTO European as well as EPSRC funded projects. He is currently a Research Fellow with 5G Innovation Centre, University of Surrey, U.K. His research interests include cooperative communications, channel coding, network coding, and quantum communications.

Panagiotis Botsinis (S'12–M'15) 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 (with Distinction) and the Ph.D. degree in wireless communications from the University of Southampton, U.K., in 2011 and 2015, respectively, where he is currently a Research Fellow with the Southampton Wireless Group, School of Electronics and Computer Science. Since 2010, he has been a member of the Technical Chamber of Greece.

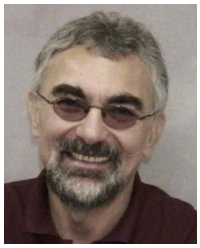
His research interests include quantum-assisted communications, quantum computation, iterative detection, OFDM, MIMO, multiple access systems, coded modulation, channel coding, cooperative communications, and combinatorial optimization.



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. degree in wireless communications from the University of Southampton in 2012, where he is currently pursuing the Ph.D. degree with the Southampton Wireless Group, School of Electronics and Computer Science.

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

Soon Xin Ng (S'99–M'03–SM'08) received the B.Eng. degree (First Class) 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 working 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, where he is currently an Associate Professor in telecommunications. His research interests include adaptive coded modulation, coded modulation, channel coding, space-time coding, joint source and channel coding, iterative detection, OFDM, MIMO, cooperative communications, distributed coding, quantum communications, quantum error correction codes, and joint wireless-and-optical-fibre communications. He has published over 240 papers and co-authored two Wiley/IEEE Press books in the above areas. He was involved in the OPTIMIX and CONCERTO European projects as well as the IU-ATC and UC4G projects. He was the Principal Investigator of an EPSRC project on Cooperative Classical and Quantum Communications Systems. He is a fellow of the Higher Education Academy in the U.K., a Chartered Engineer and a fellow of IET.



Lajos Hanzo (M'91–SM'92–F'04) received the degree in electronics in 1976, the Doctorate in 1983, and the Honorary Doctorate degrees (*Doctor Honoris Causa*) from the Technical University of Budapest in 2009 and the University of Edinburgh in 2015. During his 40-year career in telecommunications he has held various research and academic posts in Hungary, Germany, and the U.K. Since 1986, he has been with the School of Electronics and Computer Science, University of Southampton, U.K., where he holds the Chair in telecommunications. He has successfully supervised 112 Ph.D. students, co-authored 18 Wiley/IEEE Press books on mobile radio communications totalling in excess of 10 000 pages, published 1761 research contributions at IEEE Xplore, acted both as a TPC and the General Chair of IEEE conferences, presented keynote lectures and has been awarded a number of distinctions. He is currently directing a 40-strong academic research team, working 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.

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