Stabilizer codes: Encoding Schemes and applications

Shiyu Shi

Ph.D

The Hong Kong Polytechnic University

2017

The Hong Kong Polytechnic University
Department of Applied Mathematics

Stabilizer codes: Encoding Schemes and applications

Shiyu Shi

A thesis submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

April 2016

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

SHI Shiyu (Name of student)

Dedicate to my parents.

Abstract

Quantum information science is a rapidly growing research area. It concerns information theory that makes use of quantum nature of the microscopic world. In reality, quantum systems are vulnerable to disturbance from an external environment, which can lead to decoherence in the system. Thus, the system must be protected from the environmental noise to keep information stored in the quantum registers. In order to realize a working quantum computer and dependable quantum information processing, researchers and engineers have to overcome this difficulty. One of the most promising candidates for overcoming decoherence is Quantum Error Correction. The idea of quantum error correction is to protect quantum information from errors due to decoherence and other quantum noise during the transmission of information in quantum channels. One fundamental question of quantum error correction is the existence of quantum error correcting code for a noisy quantum system. Moreover, constructing practical and operational quantum error correcting schemes in actual quantum computing is of great interest to quantum information scientists.

In this thesis, stabilizer codes and a scheme for constructing recovery channels without error syndrome detection are studied. The motivation for construction of recovery channel without error syndrome detection is also given. We first review some basic concepts on stabilizer groups and stabilizer codes. In particular, we consider theories and principles involved in the construction of encoding circuits from the generators of stabilizer group, and propose a new procedure to derive recovery

channel for a well known quantum code, the $[n, k, d]$ code.

First, an algorithm to obtain the generators for a stabilizer code and the corresponding computational basis codewords defined in terms of Pauli operators are reviewed and illustrated in detail. Examples are given to demonstrate the relation between the $X-$ and $Z-$ matrices of generators of stabilizer group and the corresponding encoding circuit. Then based on the general framework of operator quantum error correction, we provide a general scheme on the construction of encoding and decoding circuits for the $[n, k, d]$ codes. Finally, a detailed procedure to construct the recovery channel using encoding circuits and encoded computational basis codewords are demonstrated for $[5, 1, 3]$ code and $[8, 3, 3]$ code step by step as examples, with heuristic explanations based on necessary and sufficient conditions for quantum error correction. Possible future study and open problems will also be mentioned.

Acknowledgements

Research needs relentless efforts and interactions with people around you. I owe my thanks to the individuals who have guided me, enlightened me, and supported me during my pursuit of PhD and I would like to bear their help and kindness in mind for the rest of my life.

First and foremost, I would like to express my gratitude for my supervisor Dr. Raymond Sze, for his patience and kindness and selfless support during my study in PolyU. What I have learned from him will benefit me in so many ways and aspects.

Also, I wish to express my deep thanks to Professors Chi-Kwong Li and Yiu-Tung Poon for their helpful talks and discussions during meetings with them.

I also would like to thank some people who helped me when I encountered difficulties and had no clue what to do, Michelle Feng, Huili Zhang, Shunjun Wang and Zhiyuan Dong.

Finally, I would like to express my special thanks to my parents and my relatives for their support, help and love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to quantum error correction

Quantum information science concerns information theory that makes use of quantum nature of the microscopic world. In quantum information theory, the elementary unit of information is represented by a quantum bit, which has two basic states $|0\rangle$ and $|1\rangle$. The two states forms a computational basis. And a quantum bit can be in any superposition state of the two basic states, which is $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, α and β are complex numbers satisfying $|\alpha|^2 + |\beta|^2 = 1$. Therefore, the qubit can be in a continuous state, and contains classical bit 0 and 1 with probability α and β respectively.

In reality, quantum systems are vulnerable to noise from the environment, which may leads to errors and decoherence in the system. Thus, the system must be protected against from noise to keep information uncorrupted in the quantum registers. Further, we want to achieve fault-tolerant quantum computation that can deal not only with noise on stored quantum information, but also with faulty quantum gates, faulty quantum preparation, and faulty measurements. In order to realize a working quantum computer and dependable quantum information processing, researchers and engineers have to overcome this difficulty. One of the most promising candidates is Quantum Error Correction(QEC) (see [1, 17, 15, 16, 14, 25, 29, 30]). The idea of

QEC is to protect data against from the noise by encoding the data together with some ancillary(redundant) states, so that even if the combined data is corrupted by the noise, there is still enough redundancy for the data to be recovered/decoded.

In quantum information theory, a quantum code C is a subspace of the state space of a quantum system. For a given code, the set of detectable error operators are closed under linear combinations. So one only has to check the elements of a linear basis for the space of error operators. In quantum error correction, the main question is that for given a set of error operators, when there is a code such that all the errors can be detected and corrected.

Stabilizer code is one of the schemes to correct error in quantum system, in particular to tensor products of Pauli operators (see [49]). Given a stabilizer code, it is easy to determine which Pauli product errors are detectable, and can be interpreted as the classical linear code. A stabilizer code of length n is a subspace of state space of n qubits that is characterized by the set of products of Pauli operators leaving each state in the code invariant. A quantum code with stabilizer S will detect all the errors that are either in S or anticommute with some element of S. To perform the error correction operation for a stabilizer code, one can measure the eigenvalues of each generator of the stabilizer. There are several ways to describe the stabilizer. One is to use binary vector spaces, which is often written as $(n-k) \times n$ binary matrices. Another approach is connected with the classical theory of codes over the field $GF(4)$ (see [4]).

Another key issue in quantum error correction is fault tolerance operation. A fault tolerant operation is an operation for which a single operational error can only product one error within a single encoded block of the code. Operations for which each qubit in a block only interacts with the corresponding qubit, either in another block or in a specialized ancilla, are called transversal operations. Any transversal

operation is fault tolerant. For example, if one measures the operator $\sigma_{z_1}\sigma_{z_2}$ in the Shor's nine-qubit code, the eigenvalue is $+1$ if the first two qubits are the same, and -1 if they are not. If the first two qubits interact with the same ancilla qubit $|0\rangle$ by CNOT operations, then a single phase error on the ancilla qubit could produce errors in both data qubits, which produce two errors in the block. So, this procedure is not a transversal operation. In order to have transversal operation, one has to pick the superposition state $|00\rangle + |11\rangle$ as the ancilla state and perform CNOT operations from data qubits to ancilla qubits. Then measuring the ancilla qubits will tell us the parity of the data qubits, but one won't deduce the state of the data, that is, measuring the ancilla will not destroy a superposition of these two states of the data.

Bounds for the quantum error correcting codes, which is related to the efficiency of an error correcting code of a given block size, is also of interest in the quantum information community. One of the upper bounds is quantum Hamming bound, which can be used to determine the efficiency of nondegenerate codes. For the degenerate codes, one has Knill-Laflamme bound. In classical coding theory, researchers used the weights of codewords, which contain a lot of information. The distribution of weights is often encoded in coefficients of polynomials, and algebraic relations between the polynomials can be used to set bounds for the classical codes. Part of this idea was adapted to give bounds on the quantum error correcting codes too.

1.2 Literature review

Quantum error correction, which is necessary for preserving coherent states against noise and other unwanted interactions with the environment, has been studied by many researchers (see [3, 4, 29, 53]). The first example of quantum error-correcting code was constructed by Shor[52]. Later, Calderbank and Shor[5] and Steane[55] proposed a general approach to construct quantum codes.

In [5], Calderbank and Shor brought up the idea of good quantum error correcting codes. They proposed that a quantum error correcting code should be a unitary mapping of k qubits into a subspace of the quantum state space of n qubits such that when any t of the qubits go through arbitrary decoherence, the resulting n qubits can be used to reconstruct the original quantum state of the encoded qubits.

In [3], Entanglement purification protocols(EPP) and Quantum error correcting code(QECC) were studied to protect quantum states from being corrupted by the environment. The difference and connection were given by the authors to show that in certain condition, EPP can be transformed into QECC and vice versa. They also showed that certain noisy channel can be used to realize trustworthy transmission of quantum states with two-way communication, but not practical only with one-way communication.

In [53], Steane gave out a new type of uncertain relation concerning the information-bearing properties of a discrete quantum system, which places a limit on the largest minimum distance simultaneously achievable in two different basis. He also showed that a pair of states which are microscopically different can form a superposition in which the interference phase is measurable.

In [29], Knill and Laflamme developed a general theory of quantum error correction based on encoding states into larger Hilbert spaces. They obtained necessary and sufficient conditions for the recovery of an encoded state after corruption by an interaction. The authors also brought up a recovery-operator-independent definition of error correcting codes and related this definition to four other: the existence of a left inverse of the interaction, an explicit representation of the error syndrome using tensor products, perfect recovery of the completely entangled state, and an information theoretic identity.

In [4], Calderbank, Rains, Shor and Sloane transformed the problem of finding quantum error correcting codes into the problem of finding additive codes over

the field $GF(4)$, which is self orthogonal with respect to some trace inner product. Specifically, the authors transformed the problem into finding a particular type of binary space first, and then showed that these spaces are equivalent to a certain class of additive codes over $GF(4)$. The authors also gave out upper and lower bounds on cyclic, self dual codes and other codes.

In [51], Schumacher and Nielsen studied the properties of noisy quantum channel, and gave a necessary and sufficient condition for perfect quantum error correction to exist.

A universal quantum computation on decoherence free subsystem(DFS) is examined in [24], and also a necessary and sufficient condition for the existence of decoherence free (noiseless) subsystem in Markovian regime was derived for the first time. A stabilizer formalism for DFSs was also given which allows us to understand these in their dual role as quantum error correcting codes explicitly.

Recently, Li et al. studied quantum error correction for general noise and fully correlated noise in [34, 35, 36]. They proved that although it is hard to physically realize quantum error correction without error syndrome measurement, they can implement the method called Operator Quantum Error Correction(OQEC) by applying unitary gates followed by a partial trace operation.

1.2.1 Contributions of the thesis

Based on the theory given by Li et al and other scholars on OQEC(see [32]), we implement this scheme on the well known $[n, k, d]$ codes. In particular, we provide a general scheme on the construction of encoding and decoding circuits for $[n, k, d]$ codes and give detail examples for $[5, 1, 3]$ code and $[8, 3, 3]$ code. Contrary to the traditional approach to error correction, the scheme saves $(n - k)$ ancillary qubits that are used in the error syndrome detection.

1.2.2 Research Methodology

This research is based on the quantum theory, coding theory and operator theory. We first compared the results in classical error correction and quantum error correction. We also studied and summarized the existing quantum error correcting codes, and proposed some general rules for effective construction of quantum error correcting codes. We also studied the possibility to construct some new codes that can improve the flaws of existing quantum error correcting codes.

We investigated mechanisms that can be used to control the quantum error and find numerical/computational algorithms of detecting and correcting quantum error, and constructing error detection and correction subsystem. We also studied practical methods for recovering error(noise) generated from various quantum systems. For example, quantum systems with collective noise as well as quantum systems influenced by errors from certain Pauli group, and other realistic quantum systems proposed by experimentalists are studied in detail. Each target quantum system is examined carefully and we mainly focus on the following directions.

1. study the existence of quantum error correcting code for target quantum systems.

2. find simple/recursive methods for constructing correcting codes.

3. decide simple quantum encoding and decoding circuits for the target systems.

4. implement these models/circuits in experiments with experimentalists.

1.2.3 Structure of the thesis

The remains of the thesis go as follows:

- Chapter 2 introduces the background and basic concepts of stabilizer codes and we use examples to interpret the structure of stabilizer codes and how to

construct stabilizer codes. We will introduce the encoding scheme first given out in [7]. and try to explain this algorithm in a more straightforward way, which will help us in constructing the recovery channel for several specific stabilizer codes.

- Chapter 3 introduces another approach to quantum error correction.
- Chapter 4 will give out some recovery channel for $[5, 1, 3]$ code.
- Chapter 5 will give out a recovery channel for $[8, 3, 3]$ code and we will illustrate and prove that there indeed exists a realizable algorithm to construct a recovery channel for a stabilizer code as long as this stabilizer code exists.
- Chapter 6 will conclude our present work and discuss the possible directions and approaches to apply the idea to a more general stabilizer code.
- In Appendix A, the Matlab codes used for searching recovery operations of $[5, 1, 3]$ and $[8, 3, 3]$ codes in Chapters 4 and 5 will be presented.

Apart from the topic of stabilizer code and quantum error correction, the author also works with his supervisor and Dr. Zejun Huang on the topic of linear preserver raised from quantum information science. In particular, they gave a complete characterization for linear maps $\phi : M_{n_1 \cdots n_k} \to M_{n_1 \cdots n_k}$ satisfying

$$\operatorname{rank} \phi(A_1 \otimes \cdots \otimes A_k) = \operatorname{rank} (A_1 \otimes \cdots \otimes A_k) \quad \text{for all } A_i \in M_{n_i},\ i = 1, \ldots, k$$

for only rank one matrices $A_1 \otimes \cdots \otimes A_k$ with $A_i \in M_{n_i}$. The detail will be presented in Appendix B, see also [22].

Chapter 2

Stabilizer Codes

The purpose of this chapter is to review the basic theories on stabilizer codes. We will give an detailed description of how to construct the generators of stabilizer codes, how to transform the associated matrices of generators into standard forms and how to use the standard forms to construct the encoding circuit.

2.1 Basic concepts

Let's recall some basic concepts first.

Definition 2.1. *We can describe a classical system by a finite set of states denoted by* Γ*. A quantum system can be described by a Hilbert space denoted by* $\overline{\Gamma}$ *with a standard orthonormal basis* $\{|\phi\rangle : \phi \in \Gamma\}$*.*

If a quantum system is in the state $|\phi_i\rangle$ *with probability* p_i*, we say such a system is in a mixed state, while a system whose vector is uniquely specified is said to be in a pure state, and a pure quantum state is a unit vector* $|\phi\rangle \in \overline{\Gamma}$ *defined to be within a phase factor* c *such that* $|c| = 1$*. In other words, a quantum state is a one-dimensional subspace in* $\overline{\Gamma}$*. And the evolution of a quantum state in a given time interval is given by* $|\phi\rangle \mapsto U|\phi\rangle$*, where* U *is a unitary operator.*

A density matrix is a convex combination of pure states in general, and it can be denoted by

$$\rho = \sum_{i=1}^{n} p_i \, |\phi_i\rangle\langle\phi_i|.$$

A quantum system composed of two separate components is called bipartite. And a state $|\phi\rangle = \sum_i p_i \, |\phi_{1,i}\rangle \otimes |\phi_{2,i}\rangle \in \mathcal{H}$, *where* $0 \leqslant p_i \leqslant 1, \sum_i p_i = 1$, *decomposed as a sum of tensor products is called a separable state. Non-separable states are called entangled states.*

Definition 2.2. *Let* $\mathcal{H}_n$ *be a* 2^n-*dimensional Hilbert space(n qubits), and let* C *be a* $k-$*dimensional subspace of* $\mathcal{H}_n$. *Then* C *is an* (n, k) *(binary) quantum error correcting code*$(QECC)$ *correcting the set of errors* $\mathcal{E} = E_a$ *if and only if there exists* $\mathcal{R}$ *such that* $\mathcal{R}$ *is a quantum operation and* $\mathcal{R} \circ E_a(|\psi\rangle) = |\psi\rangle$ *for all* $E_a \in \mathcal{E}$ *and all* $|\psi\rangle \in C$. $\mathcal{R}$ *is called the recovery and serves to actually perform the correction of the state.*

Definition 2.3. *Given a finite dimensional complex Hilbert space* $\mathcal{H}$, *a quantum channel can be viewed as a trace preserving completely positive linear map*

$$\Phi : B(\mathcal{H}) \longrightarrow B(\mathcal{H}),$$

with the operator sum representation

$$\Phi(\rho) = \sum_a E_a \rho E_a^\dagger \text{ with } \sum_a E_a^\dagger E_a = I.$$

And we are interested in a general evolution of a quantum system, which is described by quantum operation. One kind of quantum operation is a unitary time evolution of a closed system.

Let ρ_s be a density matrix of a closed system at $t = 0$ and let U_t be the time evolution operator. Then the quantum map ϵ is defined to be

$$\epsilon(\rho_s) = U_t \rho_s U_t^\dagger.$$

Definition 2.4. *A map which describes a general change of the state from ρ_s to $\epsilon(\rho_s)$ is called a quantum operation.*

A quantum operation maps a density matrix to another density matrix, such an operator is called a superoperator.

A map λ which maps a positive operator acting on $\mathcal{H}_s$ to another positive operator acting on $\mathcal{H}_s$ is said to be positive. Moreover if its extension $\lambda_T = \lambda \otimes I_n$ remains a positive operator for an arbitrary $n \in N$, then it is called a completely positive map. And a quantum channel is a completely positive trace preserving map.

2.2 Stabilizer Codes

2.2.1 Introduction

In general, a quantum error correcting code is a subspace of a Hilbert space designed so that any of a set of errors can be corrected by an appropriate quantum operation.(see [30, 53]) A quantum code C can detect an error operator E if for every quantum state $|x\rangle$ in C, $PE|x\rangle = c|x\rangle$, where P is the operator projecting the quantum system onto C and c is a constant depending on E.

Researchers are interested in codes that correct any error affecting t or fewer physical qubits. First we introduce Pauli matrices.

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

And let us consider tensor products of the Pauli matrices. Define the Pauli group $\mathcal{P}_n$ as the group consisting of tensor products of I, X, Y and Z on n qubits, with an overall phase of ± 1 or $\pm i$. The weight $wt(P)$ of a Pauli operator $P \in \mathcal{P}_n$ is the number of qubits on which it acts as X, Y or Z. Then the Pauli operators of weight t or less form a basis for the set of all errors acting on up to t or fewer qubits, so a

QECC which corrects these Pauli operators will correct all errors acting on up to t qubits.

Now we will introduce the concept of $[n, k, d]$ code.

Let's consider the following local operators on n-qubit system

$$E = \sigma_1 \otimes \sigma_2 \otimes \cdots \otimes \sigma_n \text{ with } \sigma_j \in \{I, X, Y, Z\}.$$

The weight of the operator E is defined to be the number of states σ_j where it is different from I, i.e. $w(E) = \#\{j : \sigma_j \neq I\}$.

The distance between two operators E_a and E_b is defined to be

$$d(E_a, E_b) = w(E_a^\dagger E_b).$$

Let S be a set of commuting Pauli matrices in the n-qubit system and $\{M_1, M_2, ..., M_p\}$ be the generators of the set. Let

$$V = \{|\psi\rangle : M |\psi\rangle = |\psi\rangle, \forall M \in S\}.$$

The generators $\{M_1, M_2, ..., M_p\}$ can distinguish E_a and E_b if for any $|\psi\rangle \in V, \exists M \in S$, s.t. $\langle\psi| E_a^\dagger M E_a |\psi\rangle \neq \langle\psi| E_b^\dagger M E_b |\psi\rangle$.

The subspace V of $\mathbb{C}^{2^n}$ with stabilizer S is an $[n, k, d]$ code if

1. $dim(V) = 2^k$,
2. $\{M_1, M_2, ..., M_p\}$ can distinguish E_a and E_b for any E_a, E_b with $d(E_a, E_b) < d$.

2.2.2 stabilizer group S

A QECC that encodes k qubits into n qubits is through an encoding map from the $k-$qubit Hilbert space onto a 2^k-dimensional subspace of the $n-$qubit Hilbert space

H_2^n, and a QECC is identified with the image space C_q. In quantum stabilizer codes, C_q is identified with the unique subspace of F_2^n which is fixed by the elements of an Abelian group S.

Theorem 2.1. *([29]) A quantum code C_q can be extended to an error correcting code if and only if for all encoded computational basis $\overline{|i\rangle}$, $\overline{|j\rangle}(i \neq j)$ and error operators E_a, $E_b \in E$:*

$$\overline{\langle i|}E_a^\dagger E_b\overline{|i\rangle} = \overline{\langle j|}E_a^\dagger E_b\overline{|j\rangle}, \tag{2.1}$$

and

$$\overline{\langle i|}E_a^\dagger E_b\overline{|j\rangle} = 0. \tag{2.2}$$

Proof. Here we give a simple proof for the necessary condition, since C_q can be extended to an error correcting code, which implies there exists a trace preserving recovery operation R such that for each $E_a \in E$ and $R_r \in R$, $R_rE_a = \gamma_{ra}I$, and since R is trace preserving, R_r should satisfy

$$\textstyle\sum_r R_r^\dagger R_r = I.$$

Thus,

$$\begin{aligned}
\overline{\langle i|}E_a^\dagger E_b\overline{|j\rangle} &= \overline{\langle i|}E_a^\dagger I E_b\overline{|j\rangle} \\
&= \sum_r \overline{\langle i|}E_a^\dagger R_r^\dagger R_r E_b\overline{|j\rangle} \\
&= \sum_r \overline{\langle i|}\gamma_{ra}^\star\gamma_{rb}\overline{|j\rangle} \\
&= (\sum_r \gamma_{ra}^\star\gamma_{rb})\delta_{\overline{ij}} \\
&= \lambda_{ab}\delta_{\overline{ij}}.
\end{aligned} \tag{2.3}$$

λ_{ab} is independent of computational basis. And the Kronecker delta guarantees that (2.1) and (2.2) are satisfied. □

Remark 2.1. *Conditions* (2.1) *and* (2.2) *can be summarized into one:*

$$\overline{\langle i|}E_a^\dagger E_b\overline{|j\rangle} = C_{ab}\delta_{ij}. \tag{2.4}$$

Define matrix $C = (C_{ab})$, then it is not hard to see that C is a Hermitian matrix. A quantum error correcting code is said to be degenerate if C is singular. And a quantum error correcting code has distance d if all errors $E_i \in E$ of weight less than d satisfy $\overline{\langle i|}E\overline{|j\rangle} = C_E\delta_{ij}$, while there exists at least one error that does not satisfy this condition.

Remark 2.2. *C_q can detect an error E if*

$$\overline{\langle i|}E\overline{|j\rangle} = C_E\delta_{ij}. \tag{2.5}$$

The most widely used mathematical structure gives a class of codes known as stabilizer codes(see[4, 15]). They are less general than arbitrary quantum codes, but have useful properties that make them easier to work with than the general QECC.

Definition 2.5. *Let $S \subset \mathcal{P}_n$ be an Abelian subgroup of the Pauli group that does not contain $-I$ or $\pm iI$, and let $C(S) = \{|\psi\rangle : P|\psi\rangle = |\psi\rangle, \forall P \in S\}$. Then $C(S)$ is a stabilizer code and S is its stabilizer.*

Because of the simple structure of the Pauli group, the order of any Abelian subgroup is 2^{n-k} for some k and can easily be specified by giving a set of $n-k$ commuting generators.

The codewords of the QECC are by definition in the +1-eigenspace of all elements of the stabilizer. An error E acting on a codeword will move the state into the -1-eigenspace of any stabilizer element M which anticommutes with E:

$$M(E\,|\psi\rangle) = -EM\,|\psi\rangle = -E\,|\psi\rangle.$$

Thus, measuring the eigenvalues of the generators of S will tell us information about the error that has occurred. The set of such eigenvalues can be represented as an $(n-k)-$dimensional binary vector known as error syndrome. Note that error syndrome does not tell us anything about the encoded state, but only about the error that has occurred.

We use $[n, k, d]$ to denote a quantum error correcting code, and use $[[n, k, d]]$ to refer to a stabilizer code. The middle term k refers to the number of encoded qubits, and not the dimension 2^k of the encoded space.

Notice that $S^{\perp}$ is the set of Pauli operators that commute with all elements of the stabilizer. They would appear to be errors that cannot be detected by the code. However, the theorem specifies the distance of the code by considering $S^{\perp} \backslash S$. A Pauli operator $P \in S$ cannot be detected by the code, but there is no need to detect it, since all codewords remain fixed under the action of P, making it equivalent to the identity operation. A distance d stabilizer code which has nontrivial $P \in S$ with $\mathrm{wt}(P) < d$ is called degenerate, whereas one which does not is called non-degenerate.

The stabilizer group S can be constructed from a set of $n-k$ operators $g_1, g_2, \cdots, g_{n-k}$ known as the generators of S. Each element can be expressed as a unique product of the generators.

$$s = g_1^{p_1} g_2^{p_2} \cdots, g_{n-k}^{p_{n-k}}, p_i \in \mathbb{Z}.$$

We notice that each generator has order 2, i.e. $g_i^2 = 1$, which means that S is isomorphic to F_2^{n-k}, which is the vector space of $n-k$ components, where F is the field containing $0, 1, -1$, and so the order of S is 2^{n-k}. And notice that S does not contain the elements $-I$ or $\pm iI$.

Definition 2.6. *Let C_q be a stabilizer code with generators $g_1, g_2, \cdots, g_{n-k}$ and let e be an error in the Pauli group, that is, $e \in G_n$. The error syndrome $S(e)$ is the bit string $l = l_1, l_2 \cdots l_{n-k}$, where $l_i, i = 1, \cdots, n-k$ are determined by*

$$l_i = \begin{cases} 0, & \text{if } [e, g_i] = 0 \\ 1, & \text{if } \{e, g_i\} = 0. \end{cases} \tag{2.6}$$

Remark 2.3.

1. *Any error which has a nontrivial error syndrome must anti-commute with a subset of the generators of S, and for such error, it satisfies $\langle i| e |j\rangle = 0$ for all computational basis codewords $|i\rangle$ and $|j\rangle, i, j = 0, \cdots, n-1$.*

2. *Let $E = \{E_a\}$ be errors in G_n for which $S(E_a^\dagger E_b) \neq 0$ for all E_a and $E_b \in E$, they satisfy $\langle i| E_a^\dagger E_b |j\rangle = 0$, for all basis codewords $|i\rangle$ and $|j\rangle$.*

3. *Errors which have a trivial syndrome $S(e) = 0$ commute with all the generators. The set of errors $e \in G_n$ which commute with all the generators is defined to be the centralizer of S, denoted by $C(S)$.*

Theorem 2.2. *([14]) Let E be an error and S be the stabilizer group for a stabilizer code. If S contains an element s that anticommutes with E, then for all $|c\rangle$, $|c'\rangle \in C_q$, $E|c\rangle$ is orthogonal to $|c'\rangle$:*

$$\langle c'| E |c\rangle = 0.$$

Proof. Since s anticommutes with E, we have

$$\begin{aligned} E|c\rangle &= Es|c\rangle \\ &= -sE|c\rangle, \end{aligned} \tag{2.7}$$

and

$$\begin{aligned} \langle c'| E |c\rangle &= -\langle c'| sE |c\rangle \\ &= -\langle c'| E |c\rangle . \end{aligned} \tag{2.8}$$

Therefore, $\langle c'| E |c\rangle = 0$, for all $|c'\rangle \in C_q$. □

Theorem 2.3. *([14]) Two errors $e_1, e_2 \in G_n$ have the same error syndrome if and only if they are in the same coset of $C(S)$.*

Proof. By the definition of $C(S)$, if e_1 and e_2 are in the same coset, then there exists an element $c \in C(S)$, such that $e_1 = e_2 c$, and c commutes with all the elements in S,

$$\begin{aligned} e_1 g_i &= e_2 c g_i \\ &= e_2 g_i c \\ &= (-1)^{l_i^{e_2}} g_i e_2 c \\ &= (-1)^{l_i^{e_2}} g_i e_1, \end{aligned} \tag{2.9}$$

which means if $l_i^{e_2} = 0$, i.e. $e_2 g_i = g_i e_2$, $e_1 g_i = g_i e_1$, that is, $l_i^{e_1} = 0$, and if $l_i^{e_2} = 1$, i.e.$e_2 g_i = -g_i e_2$, $e_1 g_i = -g_i e_1$, that is $l_i^{e_1} = 1$.

If e_1 and e_2 have the identical error syndrome detection, we have $l_i^{e_1} = l_i^{e_2}, i = 1, \cdots, n-k$. Since

$$e_1 g_i = (-1)^{l_i^{e_1}} g_i e_1,$$

and

$$e_2 g_i = (-1)^{l_i^{e_2}} g_i e_2,$$

we have

$$\begin{aligned} e_1 e_2 g_i &= (-1)^{l_i^{e_2}} e_1 g_i e_2 \\ &= (-1)^{l_i^{e_1} + l_i^{e_2}} g_i e_1 e_2 \end{aligned} \tag{2.10}$$

Since $l_i^{e_1} = l_i^{e_2}$, we get $e_1 e_2 g_i = g_i e_1 e_2$, thus, $e_1 e_2 \in C(S)$, that means there exists an element denoted by g such that $e_1 e_2 = g$, and $e_2 = e_1^\dagger g$, e_1 has order 2, $e_2 = e_1 g$, which means e_1 and e_2 are in the same coset of $C(S)$. □

We define the distance of a QECC to be d if all errors $e_i \in G_n$ of weight less than d are detectable, and there exists at least one error of the weight d is non detectable. Non detectable errors are in $C(S)-S$, a QECC have distance d if and only if $C(S)-S$ has an element of weight d and does not contain errors of weight less than d.

Theorem 2.4. *([14]) A quantum stabilizer code C_q with distance d is a degenerate code if and only if its stabilizer S has an element of weight less than d, excluding the identity element.*

Proof. By the definition of degenerate code, the coefficient matrix C is singular, and there exists a linear combination F of errors E_i such that

$$F\left|i\right\rangle = 0, \text{ for all basis codewords } \left|i\right\rangle,$$

here $F = \sum_a U_a E_a$,$U = (U_a)$ diagonalizes C. suppose $F = E_1 - E_2$, then $(E_1 - E_2)\left|i\right\rangle = 0$. Thus $E_1\left|i\right\rangle = E_2\left|i\right\rangle$, that is $E_1^\dagger E_2\left|i\right\rangle = \left|i\right\rangle$, so $E_1^\dagger E_2 \in S$. Since E_1 and E_2 are correctable, $\left\langle i\right| E_1^\dagger E_2\left|j\right\rangle = C_{12}\delta_{ij}$, and thus $E_1^\dagger E_2$ is detectable, since C_q has distance d, thus the weight of $E_1^\dagger E_2$ is less than d.

Conversely, if $s \in S$ has weight less than d. Take $s_a \neq s$ and let $s_a s = s_b$, then $s_a^\dagger s_b \in S$, so $s_a^\dagger s_b\left|i\right\rangle = \left|i\right\rangle$, that is, $s_a\left|i\right\rangle = s_b\left|i\right\rangle$, so $(s_a - s_b)\left|i\right\rangle = 0$, for all basis codewords $\left|i\right\rangle$. And $s_a - s_b$ is correctable, and so $\left\langle i\right|(s_a - s_b)^\dagger(s_a - s_b)\left|j\right\rangle = 0$, which means coefficient matrix C has an eigenvalue 0, so C_q is degenerate. □

Remark 2.4. *Suppose C_q is a non degenerate QECC. And let E_a and E_b be two linearly independent errors with error syndrome $S(E_a)$ and $S(E_b)$ respectively. We can see that $E_a^\dagger E_b$ has weight less than d, and it is not in S since C_q is non degenerate,*

and it is not contained in $C(S) - S$, since the distance is d, so $E_a^\dagger E_b \in G_n - C(S)$, so it anticommutes with at least one generator. Thus, $S(E_a) \neq S(E_b)$, that is, linearly independent correctable errors have different error syndromes.

The phenomenon of degeneracy has no analogue for the classical error correcting codes, and makes the study of quantum codes substantially more difficult than the study of classical error correcting codes.

An example of a stabilizer code is the five-qubit code, a $[[5, 1, 3]]$ code whose stabilizer can be generated by

$$X \otimes Z \otimes Z \otimes X \otimes I,$$

$$I \otimes X \otimes Z \otimes Z \otimes X,$$

$$X \otimes I \otimes X \otimes Z \otimes Z,$$

$$Z \otimes X \otimes I \otimes X \otimes Z.$$

The five-qubit code is a non-degenerate code, and is the smallest possible QECC which corrects one error.

It is useful to consider other representations of stabilizer codes. For instance, $P \in \mathcal{P}_n$ can be represented by a pair of $n-$bit binary vectors$(p_X|p_Z)$ where p_X is 1 for any location where P has an X or Y tensor factor and is 0 elsewhere, and p_Z is 1 for any location where P has an Z or Y tensor factor and is 0 elsewhere. So two Pauli operators P and Q are commutative if and only if $p_X \bullet q_Z + p_Z \bullet q_X = 0$. The the stabilizer for a code becomes a pair of $(n-k) \times n$ matrices. Another useful representation is to map the single-qubit Pauli operators I, X, Y and Z to the finite field $GF(4)$.

According to the principles of error syndrome detection, the stabilizer codes should satisfy three conditions.(see [7])

1. All the columns of the $X-$, $Y-$ and $Z-$matrices should be pairwise different.

2. The sum of $X-$, $Y-$ and $Z-$matrices should be all one matrix modulo 2.

3. The $X-$, $Y-$ and $Z-$matrices should satisfy $(Y+Z)X'+(X+Z)Y'+(X+Y)Z'$ should be zero matrix modulo 2.

Definition 2.7. *Define the $X-$vector of $X-$matrix of of the generator*

$$g_i = g_{i1}g_{i2}\cdots g_{in}$$

as the $n-$bit vector, denoted by X_{g_i}, where

$$(X_{g_i})_j = \begin{cases} 1, & \text{if } g_{ij} = X \text{ or } Y \\ 0, & \text{if } g_{ij} = I \text{ or } Z. \end{cases} \tag{2.11}$$

The $Z-$vector of g_i, denoted by Z_{g_i}, is defined as

$$(Z_{g_i})_j = \begin{cases} 1, & \text{if } g_{ij} = Z \text{ or } Y \\ 0, & \text{if } g_{ij} = I \text{ or } X. \end{cases} \tag{2.12}$$

And the $X-$matrix of the generators $g_1, g_2, \cdots, g_{n-k}$ is defined as the $n \times (n-k)$ matrix, denoted by X_g, where

$$(X_g)_{ji} = \begin{cases} 1, & \text{if } g_{ij} = X \text{ or } Y \\ 0, & \text{if } g_{ij} = I \text{ or } Z. \end{cases} \tag{2.13}$$

that is , the columns of the $X-$matrix are $X_{g_1}, X_{g_2}, \cdots, X_{g_{n-k}}$. The $Z-$matrix of $g_1, g_2, \cdots, g_{n-k}$, denoted by Z_g, is defined similarly.

Example 2.1. *Generators for an eight-qubit code protecting three-qubit states with at most one error are as follows:*

$$
\begin{aligned}
g_1 &= X \otimes X \otimes X \otimes X \otimes X \otimes X \otimes X \otimes X, \\
g_2 &= Z \otimes Z \otimes Z \otimes Z \otimes Z \otimes Z \otimes Z \otimes Z, \\
g_3 &= X \otimes I \otimes X \otimes I \otimes Z \otimes Y \otimes Z \otimes Y, \\
g_4 &= X \otimes I \otimes Y \otimes Z \otimes X \otimes I \otimes Y \otimes Z, \\
g_5 &= X \otimes Z \otimes I \otimes Y \otimes I \otimes Y \otimes X \otimes Z.
\end{aligned} \tag{2.14}
$$

The $X-$matrix and $Z-$matrix for the generators (5.1) *of the 8-qubit code are as follows:*

$$
X_g = \begin{bmatrix}
1 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0
\end{bmatrix}, \; Z_g = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 1
\end{bmatrix}.
$$

In general, generator whose $X-$vectors are linearly independent are called primary generators, and generators whose $X-$vectors are null are called secondary generators. Generators can always be transformed so that they contain primary and secondary ones.

To choose the code words, we add a set of seed generators(see [7]) to the $n-k$ generators in a way that seed generators and the $n-k$ generators are linearly independent and each seed generator commutes with each secondary generator.

Let $M_1, \cdots, M_b, L_1, \cdots, L_{n-k-b}$ and $N_1, \cdots, N_k$ be the primary, secondary and seed generators, then each $k-$qubit basis state $|c_1 c_2 \cdots c_k\rangle$ can be associated with a quantum codeword,

$$
\tfrac{1}{\sqrt{2^b}} \textstyle\sum_{\{a_1, a_2, \cdots, a_b\} \in \{0,1\}^b} \; M_1^{a_1} M_2^{a_2} \cdots M_b^{a_b} N_1^{c_1} N_2^{c_2} \cdots N_k^{c_k} |0\rangle^{\otimes n}.
$$

Then we can see that these 2^k codewords are mutually different and orthogonal to each other.

we can rewrite the expression as

$$\frac{1}{\sqrt{2^b}}(I + M_1)(I + M_2) \cdots (I + M_b) N_1^{c_1} N_2^{c_2} \cdots N_k^{c_k} |0\rangle^{\otimes n}.$$

And we can see that the expression is stabilized by each primary M_i and secondary generators L_j.

Example 2.2. *For the 8-qubit code, the seed generators can be chosen as follows:*

$$\begin{aligned} N_1 &= X \otimes X \otimes I \otimes I \otimes I \otimes I \otimes I \otimes I, \\ N_2 &= X \otimes I \otimes X \otimes I \otimes I \otimes I \otimes I \otimes I, \\ N_3 &= X \otimes I \otimes I \otimes I \otimes X \otimes I \otimes I \otimes I. \end{aligned} \tag{2.15}$$

Remark 2.5. *Operators of the form $\frac{1}{\sqrt{2}}(I + M)$ are not unitary, so we need explore the properties of stabilizer codes to construct the efficient gate arrays, that is, the encoding circuit.*

The encoding mainly consists of two parts:

1. X-matrix and Z-matrix of the generators are converted into standard forms.
2. The set of generators of the stabilizer group are converted into a gate array according to the standard forms of X-matrix and Z-matrix.

The procedure of encoding is presented in detail:

1. First, we can convert the X-matrix into one of the following form using Gaussian elimination:

$$\begin{bmatrix} O & A \\ O & I \end{bmatrix}$$

and Z-matrix has no special form. Suppose it has the form:

$$\begin{bmatrix} B & C \\ D & E \end{bmatrix}$$

2. Then we can apply the same procedure to the submatrix B of Z-matrix, and a block matrix of the following form will be derived:

$$\begin{bmatrix} O & B_1 & C_1 \\ O & I & C_2 \\ D_2 & D_1 & E \end{bmatrix}$$

then we analyze the rank of the first subblock O of this matrix, since the counterpart of X-matrix for the columns that O lies in is

$$\begin{bmatrix} O \\ O \\ O \end{bmatrix}$$

then we get that the submatrix D_2 should not be O, otherwise the corresponding generator is the tensor product of all I's. Thus we will get that the corresponding generators don't commute with those of last b columns. Therefore the Z-matrix should be of the form:

$$\begin{bmatrix} B_1 & C_1 \\ I & C_2 \\ D_1 & E \end{bmatrix}$$

3. Then we set up the standard forms of the seed generators, and by the rule of the relationship between the seed generators and the primary and secondary ones we may have the following forms for the $X-$matrix and $Z-$matrix of the seed generators:

$$\begin{bmatrix} I \\ B_1^T \\ O \end{bmatrix}, \begin{bmatrix} O \\ O \\ O \end{bmatrix}$$

we can verify that the generator of the stabilizer group are commutative. That is why we set the $X-$matrix and $Z-$matrix of seed generators to be of the form above.

$$\begin{bmatrix} I \\ B_1^T \\ O \end{bmatrix}^T \begin{bmatrix} B_1 \\ I \\ D_1 \end{bmatrix} = \begin{bmatrix} I & B_1 & O \end{bmatrix} \begin{bmatrix} B_1 \\ I \\ D_1 \end{bmatrix} = B_1 + B_1 = O.$$

that means the seed generators and the secondary generators are commutative.

4. Finally, we put the standard forms of the primary generators, secondary generators and seed generators together to form an augmented matrix. The augmented matrices, denoted by $X^\star$ and $Z^\star$ have the following block matrix form:

$$X^\star = \begin{bmatrix} I & O & A_1 \\ B_1^T & O & A_2 \\ O & O & I \end{bmatrix}, \quad Z^\star = \begin{bmatrix} O & B_1 & C_1 \\ O & I & C_2 \\ O & D_1 & E \end{bmatrix}.$$

Remark 2.6. *The reason why we want the submatrix I to be at the lower right of the standard form of the X-matrix is that when we augment the standard form with those of the seed generators, we want two different I's to be at different positions of the augmented matrix, so that when we apply the actions of the N's and M's to the n-qubit, we want the two types of actions triggered by two sets of parameters, and the other reason is that the X-matrix of the primary and seed generators should be linearly independent.*

In order for the two sets of parameters to act without interfering with each other, we have to put two different I's at two different positions.

Example 2.3. *The augmented $X-$matrix and $Z-$matrix for the generators* (5.1) *of the 8-qubit code are as follows:*

$$X^\star = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, Z^\star = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

The first three columns correspond to seed generators, the fourth column correspond to secondary generators, and the last four columns correspond to primary generators.

2.2.3 Standard forms of encoded Pauli X^i and Z^i operator

The principle that plays an important part is that encoded Pauli X^i and Z^i operator should be commutative with generators of the stabilizer group S.

First we introduce an inner product which maps two vectors onto F_2, if $v_1 = (a_1|b_1), v_2 = (a_2|b_2) \in F_2^{2n}$,

$< v_1, v_2 >= a_1 \bullet b_2 + a_2 \bullet b_1$, here $\bullet$ denotes an inner product of two vectors. .

Let

$$J = \begin{bmatrix} O & I_n \\ I_n & O \end{bmatrix},$$

then

$$< v_1, v_2 >= v_1^T J v_2.$$

And let $v((X^i)^T) = (u_1(i)^T, u_2(i)^T, u_3(i)^T | v_1(i)^T, v_2(i)^T, v_3(i)^T)$, here $u_1(i)^T$ and $v_1(i)^T$ have r components, $u_2(i)^T$ and $v_2(i)^T$ have $n-k-r$ components, and $u_3(i)^T$

and $v_3(i)^T$ have k components, so that the partition matches the standard forms of $X-$ matrix and $Z-$matrix of generators.

By the rules of choices of encoded Pauli X^i and Z^i operators, they satisfy the following conditions:

1. $< v((X^i)), v((g^j)) >= 0$, $j = 1, \cdots, n-k$
2. $< v((X^i)), v((Z^j)) >= 0$, $j \neq i$,
3. $< v((X^i)), v((X^j)) >= 0$,
4. $< v((X^i)), v((Z^i)) >= 1$,

The vector $v((X^i)^T)$ has $2n$ components, and there are $n-k+k-1+k+1$ equations, so the components of $v((X^i)^T)$ has $n-k$ degrees of freedom.

We can put $u_1(i)^T = O$ and $v_2(i)^T = O$, then

$$v((X^i)^T) = (O, u_2(i)^T, u_3(i)^T | v_1(i)^T, O, v_3(i)^T).$$

Since

$$X = \begin{bmatrix} O & A_1 \\ O & A_2 \\ O & I \end{bmatrix}, Z = \begin{bmatrix} B_1 & C_1 \\ I & C_2 \\ D_1 & E \end{bmatrix},$$

we have

$$\begin{bmatrix} I & A_2 & A_1 & E & C_2 & C_1 \\ O & O & O & D_1 & I & B_1 \end{bmatrix} \begin{bmatrix} v_1(i) \\ O \\ v_3(i) \\ O \\ u_2(i) \\ u_3(i) \end{bmatrix} = O.$$

Thus we can obtain the following equations:

$$\begin{aligned} v_1(i) + A_1 v_3(i) + C_2 u_2(i) + C_1 u_3(i) = O, \\ u_2(i) + B_1 u_3(i) = O. \end{aligned} \tag{2.16}$$

Let

$$\chi = \begin{bmatrix} v^T(X^1) \\ v^T(X^2) \\ \cdots \\ v^T(X^k) \end{bmatrix} = \begin{bmatrix} O & u_2^T & u_3^T | & v_1^T & O & v_3^T \end{bmatrix}.$$

By the conditions mentioned before, we have

$$\chi J \chi = O.$$

That is

$$O = \begin{bmatrix} O & u_2^T & u_3^T | & v_1^T & O & v_3^T \end{bmatrix} \begin{bmatrix} v_1 \\ O \\ v_3 \\ O \\ u_2 \\ u_3 \end{bmatrix} = u_3^T v_3 + v_3^T u_3.$$

We can put $u_3 = I$ and $v_3 = O$. And put them back in (2.16), we can get

$$\chi = \begin{bmatrix} O & B_1^T & I | & (B_1^T C_2^T + C_1^T) & O & O \end{bmatrix}$$

Next one can use $X-$ and $Z-$ matrices to produce the gate array of the encoding circuit.

Generation of the encoding circuit

By the encoding rule of the stabilizer code, we need to construct a gate array to realize the following operation:

$|c_1 c_2 \cdots c_k\rangle \otimes |0\rangle^{\otimes d} \longmapsto \sum_{\{a_1,a_2,\cdots,a_b\}\in\{0,1\}^b} M_1^{a_1} M_2^{a_2} \cdots M_b^{a_b} N_1^{c_1} N_2^{c_2} \cdots N_k^{c_k} |0\rangle^{\otimes n}$.

The operation above can be decomposed as a composition of two actions:

1. $|c_1 c_2 \cdots c_k\rangle \otimes |0\rangle^{\otimes d} \longmapsto \frac{1}{\sqrt{2^b}} \sum_{\{a_1,a_2,\cdots,a_b\}\in\{0,1\}^b} |c_1 c_2 \cdots c_k\rangle \otimes |0\rangle^{\otimes r} \otimes |a_1 a_2 \cdots a_b\rangle$

2. $|c_1 c_2 \cdots c_k\rangle \otimes |0\rangle^{\otimes r} \otimes |a_1 a_2 \cdots a_b\rangle \longmapsto M_1^{a_1} M_2^{a_2} \cdots M_b^{a_b} N_1^{c_1} N_2^{c_2} \cdots N_k^{c_k} |0\rangle^{\otimes n}$

The process is basically made up of two steps:

1. First, $|0\rangle^{\otimes b}$ is transformed into $\frac{1}{\sqrt{2^b}}\sum_{\{a_1,a_2,\cdots,a_b\}\in\{0,1\}^b}|a_1a_2\cdots a_b\rangle$, and the reason why we have this operation is that in the second step, the set of qubits $\{a_i\}$ play the role of control qubits, so that when the initial state is put into the encoding circuit, the set of qubits $\{a_i\}$ can control the action of encoding on the target qubits to realize the encoding.

2. Secondly, the $|a_1a_2\cdots a_b\rangle$ and $|c_1c_2\cdots c_k\rangle$ are used to trigger the action of the operator $M_1^{a_1}M_2^{a_2}\cdots M_b^{a_b}$ and $N_1^{c_1}N_2^{c_2}\cdots N_k^{c_k}$.

2.2.4 Explanation of generation of encoding circuit

The computational basis codewords have the following expression:

$$\overline{|\delta_1\delta_2\cdots\delta_k\rangle} = \xi\,|\delta_1\delta_2\cdots\delta_k\rangle,$$

here ξ is the encoding map and

$$|\delta_1\delta_2\cdots\delta_k\rangle = X_1^{\delta_1}X_2^{\delta_2}\cdots X_k^{\delta_k}|0\rangle^{\otimes k}.$$

Thus we have another further expression for $\overline{|\delta_1\delta_2\cdots\delta_k\rangle}$ as follows:

$$\overline{|\delta_1\delta_2\cdots\delta_k\rangle} = \xi X_1^{\delta_1}X_2^{\delta_2}\cdots X_k^{\delta_k}|0\rangle^{\otimes k},$$

which is

$$(\xi X_1^{\delta_1}\xi^\dagger)(\xi X_2^{\delta_2}\xi^\dagger)\cdots(\xi X_k^{\delta_k}\xi^\dagger)\xi|0\rangle^{\otimes k},$$

and we can rewrite it as

$$(X^1)^{\delta_1}(X^2)^{\delta_2}\cdots(X^k)^{\delta_k}\overline{|0\rangle^{\otimes n}},$$

We can show that the basis codeword $\overline{|0\rangle^{\otimes n}}$ is convenient to be defined as

$$\overline{|0\rangle^{\otimes n}} = \sum_{s\in S} s|0\rangle^{\otimes n}.$$

Since S is the stabilizer group and it can be generated by its $n-k$ generators, therefore we can write the codeword equivalently as follows:

$$\overline{|0\rangle^{\otimes n}} = \prod_{i=1}^{n-k}(I+g_i)|0\rangle^{\otimes n}.$$

Therefore,

$$\overline{|\delta_1\delta_2\cdots\delta_k\rangle} = (X^1)^{\delta_1}(X^2)^{\delta_2}\cdots(X^k)^{\delta_k}\overline{|0\rangle^{\otimes n}},$$

that is,

$$\overline{|\delta_1\delta_2\cdots\delta_k\rangle} = (X^1)^{\delta_1}(X^2)^{\delta_2}\cdots(X^k)^{\delta_k}\prod_{i=1}^{n-k}(I+g_i)|0\rangle^{\otimes n},$$

The encoding circuit we want to implement is to realize the action above with the application of an appropriate sequence of single-qubit and controlled multiple qubit operations to the initial n-qubit input state $|0\cdots0\delta_1\delta_2\cdots\delta_k\rangle$.

1. First, we need to transform the expression so that the right hand side have an item of $|0\cdots0\delta_1\delta_2\cdots\delta_k\rangle$, which depends on the action of $(X^1)^{\delta_1}(X^2)^{\delta_2}\cdots(X^k)^{\delta_k}$, and we can find out that the result is realized by the standard form of $X^j, j = 1, 2, \cdots, k$.

   The standard form of X^j is

   $$\begin{bmatrix} 0 & u_2^T(j) & u_3^T(j) \\ v_2^T(j) & 0 & 0 \end{bmatrix},$$

   where $u_3^T(j) = (0\cdots1_j\cdots0)$ and the subscript j shows the column number. Since the first row corresponds to the $X-$vector and the second row corresponds to the $Z-$vector, we find that $X^j, j = 1, \cdots, k$, have the more specific expression as follows:

   $$X^j = (X_{r+1})^{u_{2,1}(j)}(X_{r+2})^{u_{2,2}(j)}\cdots(X_{n-k})^{u_{2,n-k-r}(j)} \times (Z_1)^{v_{1,1}(j)}(Z_2)^{v_{1,2}(j)}\cdots(Z_r)^{v_{1,r}(j)}(X_{n-k+j}), \quad (2.17)$$

thus when $(X^1)^{\delta_1}(X^2)^{\delta_2}\cdots(X^k)^{\delta_k}$ acts on $|0\rangle^{\otimes n}$, we can find that the last component (X_{n-k+j}) can transform $|0_{n-k+j}\rangle$ into $|\delta_{n-k+j}\rangle$.

Denote $(X_{r+1})^{u_{2,1}(j)}(X_{r+2})^{u_{2,2}(j)}\cdots(X_{n-k})^{u_{2,n-k-r}(j)}(Z_1)^{v_{1,1}(j)}(Z_2)^{v_{1,2}(j)}\ldots(Z_r)^{v_{1,r}(j)}$ by $\tilde{U}_j$, then we have

$$(X^1)^{\delta_1}(X^2)^{\delta_2}\cdots(X^k)^{\delta_k}|0\rangle^{\otimes n} = (\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})\,|0\cdots0\delta_1\delta_2\cdots\delta_k\rangle.$$

We can see that $\tilde{U_j}^{\delta_j}$ is a controlled$-\tilde{U}_j$ operation. And when $\delta_j = 0$, the state $|0\cdots0\delta_1\delta_2\cdots\delta_k\rangle$ won't change and when $\delta_j = 1, \tilde{U}_j$ will act on the state.

2. To acquire the basis codewords $\overline{|\delta_1\delta_2\cdots\delta_k\rangle}$, we still need to apply $\prod_{i=1}^{n-k}(I+g_i)$ to $(\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})\,|0\cdots0\delta_1\delta_2\cdots\delta_k\rangle$, and we split the product into two parts:

$$\prod_{i=1}^{r}(I+g_i) \text{ and } \prod_{j=r+1}^{n-k}(I+g_j).$$

Notice that $g_i, i = 1,\ldots,r$, are primary generators and $g_i, i = r+1,\ldots,n-k$, are secondary generators. Thus we can write the basis codewords as follows:

$$\overline{|\delta_1\delta_2\cdots\delta_k\rangle} = (X^1)^{\delta_1}(X^2)^{\delta_2}\cdots(X^k)^{\delta_k}\prod_{i=1}^{r}(I+g_i)\prod_{j=r+1}^{n-k}(I+g_j)|0\rangle^{\otimes n},$$

since $X^j, j = 1,\cdots,k$ are in $C(S)$, i.e. the center of the stabilizer group. We can interchange the order of the actions of $(X^1)^{\delta_1}(X^2)^{\delta_2}\cdots(X^k)^{\delta_k}$ and $\prod_{i=1}^{r}(I+g_i)$, since they are commutative.

$$\overline{|\delta_1\delta_2\cdots\delta_k\rangle} = \prod_{i=1}^{r}(I+g_i)(X^1)^{\delta_1}(X^2)^{\delta_2}\cdots(X^k)^{\delta_k}\prod_{j=r+1}^{n-k}(I+g_j)|0\rangle^{\otimes n},$$

Since $g_j, j = r+1,\cdots,n-k$ are secondary generators, which means it fixes $|0\rangle^{\otimes n}$, we can write the codeword as follows:

$$\overline{|\delta_1\delta_2\cdots\delta_k\rangle} = \prod_{i=1}^{r}(I+g_i)(X^1)^{\delta_1}(X^2)^{\delta_2}\cdots(X^k)^{\delta_k}|0\rangle^{\otimes n},$$

Plugging

$$(X^1)^{\delta_1}(X^2)^{\delta_2}\cdots(X^k)^{\delta_k}|0\rangle^{\otimes n} = (\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})\,|0\cdots 0\delta_1\delta_2\cdots\delta_k\rangle.$$

into the codeword, we can obtain

$$\overline{|\delta_1\delta_2\cdots\delta_k\rangle} = \prod_{i=1}^{r}(I+g_i)(\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})\,|0\cdots 0\delta_1\delta_2\cdots\delta_k\rangle.$$

The standard form of g_j is

$$\begin{bmatrix} 0\cdots 1_j\cdots 0 & A_1(j) & A_2(j) \\ B(j) & C_1(j) & C_2(j) \end{bmatrix},$$

Similarly, the first row corresponds to the $X-$vector and the second row corresponds to the $Z-$vector. and

$$g_j = T_j X_j Z_j^{B_j(j)},$$

here T_j is the operators that remain when we factor out X operator and Z operator associated with qubit j, and subscript j also points to the column number.

$$\begin{aligned}(I+g_j)(\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})\,|0\cdots 0\delta_1\delta_2\cdots\delta_k\rangle = \quad & (\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})\,|0\cdots 0\delta_1\delta_2\cdots\delta_k\rangle \\ +T_jX_jZ_j^{B_j(j)}(\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})\,|0\cdots 0\delta_1\delta_2\cdots\delta_k\rangle&.\end{aligned} \tag{2.18}$$

We notice that $(\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})$ acts on qubit $r+1$ to $n-k$, and $j<r$, thus $X_jZ_j^{B_j(j)}$ and $(\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})$ can be exchanged, and

$$X_jZ_j^{B_j(j)}\,|0\cdots 0\delta_1\delta_2\cdots\delta_k\rangle = |0\cdots 1_j\cdots 0\delta_1\delta_2\cdots\delta_k\rangle.$$

Therefore we have

$$\begin{aligned}(I+g_j)(\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})\,|0\cdots 0\delta_1\delta_2\cdots\delta_k\rangle = (\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})\,|0\cdots 0\delta_1\delta_2\cdots\delta_k\rangle \\ +T_j(\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})\,|0\cdots 1_j\cdots 0\delta_1\delta_2\cdots\delta_k\rangle\,.\end{aligned} \tag{2.19}$$

$$\overline{|\delta_1\delta_2\cdots\delta_k\rangle} = \prod_{i=1}^{r}(I+g_i)(\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})\,|0\cdots0\delta_1\delta_2\cdots\delta_k\rangle\,. \tag{2.20}$$

And we want to show that we can construct the basis codewords using Hadamard gates and controlled gates. Let H_j be the single qubit Hadamard gate acting on qubit j,

$$H_j\,|\delta_j\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + (-1)^{\delta_j}\,|1\rangle), \delta_j = 0, 1.$$

Therefore we have

$$\begin{aligned} H_j(\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})\,|0\cdots0\delta_1\delta_2\cdots\delta_k\rangle &= (\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})H_j\,|0\cdots0\delta_1\delta_2\cdots\delta_k\rangle \\ = (\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})(&|0\cdots0_j\cdots0\delta_1\delta_2\cdots\delta_k\rangle + |0\cdots1_j\cdots0\delta_1\delta_2\cdots\delta_k\rangle). \end{aligned} \tag{2.21}$$

Here we use the fact that H_j and $\prod_{j=1}^{k}\tilde{U_j}^{\delta_j}$ are commutative since they act on different qubits, and we have omitted the factor of $\frac{1}{\sqrt{2}}$.

Finally, in order to obtain the basis codewords, we have to apply the controlled quantum gates, that is, $W_j = T_j^{\alpha_j}, j = 1, \cdots, r, \alpha_j = 0, 1$, then we find that

$$\begin{aligned} W_jH_j(\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})\,|0\cdots0\delta_1\delta_2\cdots\delta_k\rangle &= W_j(\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})H_j\,|0\cdots0\delta_1\delta_2\cdots\delta_k\rangle \\ &= W_j(\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})(|0\cdots0_j\cdots0\delta_1\delta_2\cdots\delta_k\rangle + |0\cdots1_j\cdots0\delta_1\delta_2\cdots\delta_k\rangle) \\ = (\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})\,|0\cdots0_j\cdots0\delta_1\delta_2\cdots\delta_k\rangle &+ T_j(\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})\,|0\cdots1_j\cdots0\delta_1\delta_2\cdots\delta_k\rangle\,. \end{aligned} \tag{2.22}$$

Since $(\prod_{j=1}^{k}\tilde{U_j}^{\delta_j})$ acts on qubits $r+1$ to $n-k$ so it won't change the value of α_j, which remains to be o in the first term and 1 in the second one. That means $W_j = I$ in the first term and $W_j = T_j$ in the second term. We can see that

$$W_i H_i(\prod_{j=1}^{k} \tilde{U_j}^{\delta_j}) \left|0\cdots 0\delta_1\delta_2\cdots\delta_k\right\rangle = (I+g_i)(\prod_{j=1}^{k} \tilde{U_j}^{\delta_j}) \left|0\cdots 0\delta_1\delta_2\cdots\delta_k\right\rangle.$$

So for the basis codewords, we have

$$\begin{aligned}\overline{\left|\delta_1\delta_2\cdots\delta_k\right\rangle} &= \prod_{i=1}^{r}(I+g_i)(\prod_{j=1}^{k} \tilde{U_j}^{\delta_j}) \left|0\cdots 0\delta_1\delta_2\cdots\delta_k\right\rangle \\ &= (\prod_{i=1}^{r} W_i H_i)(\prod_{j=1}^{k} \tilde{U_j}^{\delta_j}) \left|0\cdots 0\delta_1\delta_2\cdots\delta_k\right\rangle .\end{aligned} \tag{2.23}$$

Remark 2.7. *The aim is to construct the basis codewords using controlled quantum gates, since in* (2.19)*, the right hand side has two parts, which implies that we can use controlled gates to realize the action, so that we can apply Hadamard gate first then apply controlled gate.*

The first step can be realized by use of Hadamard gate, and for the second step, take [8,3,3] code for example, it can be realized by the following action:

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \longmapsto \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}$$

The second step is also crucial since it determines the transformation from a matrix to a specific encoding circuit.

The matrix is exactly the augmented matrix of the generators of the stabilizer group. So as long as we can acquire a set of generators of the stabilizer code, then we can apply the procedure above to obtain the encoding circuit, which is the most important part of quantum error correction.

Remark 2.8. *The k-qubits input may be at different positions, but the correspondence between the action of the operators $M_1, M_2, \cdots, M_d$ and the n-qubits should be unchanged, that is the each operator is a tensor product of Pauli matrix, and each of the component should act on the correspondent qubit.*

2.2.5 Clifford Codes v.s. Stabilizer Codes

Definition 2.8. *A finite group E is said to be an abstract error group if it has a faithful irreducible unitary representation ρ of degree $= |E : Z(E)|^{1/2}$.*

In the special case of binary stabilizer code the error group is given by an extra special 2-group and the representing matrices $\rho(g)$ by tensor products of Pauli matrices. The irreducibility of the representation ensures that any error acting on the code space C^d can be expressed as a linear combination of the matrices $\rho(g)$, with $g \in E$. The faithfulness of the representation and the largest possible degree ensures that the set of matrices $\{\rho(g)|g \in T\}$, where T is a set of representatives of $E/Z(E)$.

A clifford code is constructed with the help of a normal subgroup N of the error group E and an irreducible character χ of N.

Definition 2.9. *Let ϕ denote the irreducible character corresponding to the representation ρ of E, that is $\phi(g) = Tr\rho g$ for $g \in E$. Suppose that N is a normal subgroup of E and χ is an irreducible character of N such that $(\chi, \phi_N) > 0$. Then the* **Clifford code** *C corresponding to (E, ρ, N, χ) is defined to be the image of the orthogonal projector*

$$P = \frac{\chi(1)}{|N|} \sum_{n \in N} \chi(n^{-1})\rho(n),$$

that is, a subspace of dimension trP of C^d. And if the normal subgroup N is abelian, then the Clifford code is called a **stabilizer code** *(see [27, 28]). And $(\chi, \phi_N) > 0$ implies that dim $C > 0$.*

The criteria to decide whether a clifford code is a stabilizer code or not is based on the quasikernel of the group E.

Let Q be a Clifford code with data (E, ρ, N, χ). The inertia subgroup is

$$T = \{g \in E | \chi(n) = \chi(gng^{-1}) \ \forall n \in N\},$$

which consists of all the elements g of E such that $\rho(g)Q = Q$. Let $\omega(Irr(E))$ denote the character such that $[\omega, \phi_T] \neq 0$, and $[\omega_N, \chi] \neq 0$. This is the character afforded by the irreducible $\mathbb{C}T-$module Q. The quasikernel

$$Z(\omega) = \{g \in E| \ |\omega(g)| = \omega(1)\}$$

consists of the elements g of E that act on the code Q by scalar multiplication. These two groups characterize the errors in E that are detectable by the code.

An error $\rho(g)$ is detectable by the code Q if and only if $g \notin T - Z(\omega)$. The group $Z(\omega)$ can tell us whether the Clifford code Q is a stabilizer code or not.

Denote by $\mathcal{A}$ the set of all normal subgroups A of E that are contained in $Z(\omega)$. And the following results show that in the case that Q is a stabilizer code, its stabilizer can be found in terms of a maximal group of $\mathcal{A}$.

Lemma 2.1. *If $A \in \mathcal{A}$, then there exists a linear character θ of A such that the image of the orthogonal projector*

$$P_A = \frac{1}{|A|} \sum_{a \in A} \theta(a^{-1})\rho(a)$$

contains Q, meaning that $P_A v = v$ holds for all $v \in Q$.

Lemma 2.2. *Let A be an abelian normal subgroup of E with linear character θ. If the image of the projector contains the Clifford code Q, then $A \leqslant Z(\omega)$.*

Theorem 2.5. *Let Q be a Clifford code with data (E, ρ, N, χ), and denote by ϕ the irreducible character of E afforded by the representation ρ. Keeping the notations above, we can conclude that Q is a stabilizer code if and only if*

$$dimQ = |A \cap Z(E)|\frac{\phi(1)}{A}$$

holds for some $A \in \mathcal{A}$.

Example 2.4. *Let G be the finite group generated by three elements a, b, c subject to the relations:*

$$a^2 = b^2 = [a, b] = 1 \text{ and } a^c = b, b^c = a, c^4 = 1.$$

This is the index group that we introduce.

An abstract error group E is obtained by a central extension of the index group G by a cyclic group of order 2. More explicitly, E is presented by four generators a, b, cd that are subject to the relations

$$a^2 = b^2 = [a, b] = 1 \text{ and } d^2 = [a, d] = [b, d] = [c, d] = 1.$$

The group E is nilpotent of class 3 and of order 32. A faithful irreducible representation of E is given by

$$\rho(a) = \begin{bmatrix} \bullet & \bullet & -1 & \bullet \\ \bullet & \bullet & \bullet & -1 \\ -1 & \bullet & \bullet & \bullet \\ \bullet & -1 & \bullet & \bullet \end{bmatrix}, \rho(b) = \begin{bmatrix} \bullet & \bullet & \bullet & -i \\ \bullet & \bullet & i & \bullet \\ \bullet & -i & \bullet & \bullet \\ i & \bullet & \bullet & \bullet \end{bmatrix}, \rho(c) = \begin{bmatrix} \bullet & 1 & \bullet & \bullet \\ 1 & \bullet & \bullet & \bullet \\ \bullet & \bullet & -i & \bullet \\ \bullet & \bullet & \bullet & i \end{bmatrix}.$$

and the generator d of the enter of E is represented by $\rho(d) = -1$, and it has a nonabelian index group and yet all its Clifford codes are stabilizer codes. This follows from the fact that all nontrivial normal subgroups of G are abelian.

Chapter 3

Operator Quantum Error Correction

3.1 Standard model of quantum error correction

The Standard Model for error correction of quantum operations(see [3, 29, 52, 54]) consists of triples $(\mathbb{R}, \mathbb{E}, \mathbb{C})$ where $\mathbb{C}$ is a quantum code, a subspace of some Hilbert space $\mathbb{H}$ associated with a given quantum system. The error $\mathbb{E}$ and recovery $\mathbb{R}$ are quantum operations on $\mathbb{B}(\mathbb{H})$ such that $\mathbb{R}$ annihilates the effects of $\mathbb{E}$ on $\mathbb{C}$ in the following sense:

$$(\mathbb{R} \circ \mathbb{E})(\sigma) = \sigma, \forall \sigma = P_{\mathbb{C}} \sigma P_{\mathbb{C}},$$

where $P_{\mathbb{C}}$ is the projection of $\mathbb{H}$ onto the subspace $\mathbb{C}$. When there exists such an $\mathbb{R}$ for a given $(\mathbb{E}, \mathbb{C})$, the subspace $\mathbb{C}$ is said to be correctable for $\mathbb{E}$. The existence of a recovery operation $\mathbb{R}$ of $\mathbb{E} = \{E_a\}$ on $\mathbb{C}$ may be phrased in terms of $\{E_a\}$ as follows (see [29, 54]):

$$P_{\mathbb{C}} E_a^{\dagger} E_b P_{\mathbb{C}} = \lambda_{ab} P_{\mathbb{C}} \forall a, b$$

for some Hermitian matrix $\Lambda = (\lambda_{ab})$. It is easy to see that this condition is independent of the operator-sum representation for $\mathbb{E}$.

3.2 Noiseless subsystems and decoherence free subspaces

Definition 3.1. *An open system undergoes decoherence if its evolution is a non unitary evolution. And an open system undergoing purely unitary evolution is called a decoherence free subsystem.*

In [31, 32], Kribs et al. developed a new scheme called operator quantum error correction formalism that combined three know techniques, the standard error correction model, the method of decoherence-free subspaces and the noiseless subsystem method. Also a generalized framework has been introduced for noiseless subsystems that can be applied to arbitrary quantum operations.

Definition 3.2. *Let $E = \{E_a\}$ be a quantum operation on $\mathcal{H}$. Let $\mathcal{A}$ be the $C^\star-$algebra generated by the E_a, so $\mathcal{A} = Alg\{E_a, E_a^\dagger\}$ is the set of polynomials in the E_a and $E_a^\dagger$. Then $\mathcal{A}$ has a unique decomposition up to unitary equivalence of the form*

$$\mathcal{A} \cong \bigoplus(\mathcal{M}_{m_j} \otimes 1_{n_j}).$$

This means that there is an orthonormal basis such that the matrix representations of operators in $\mathcal{A}$ with respect to this basis have the form of direct sum of tensor products. And $\mathcal{A}$ is called the **interaction algebra** associated with the operation E.

The standard noiseless subsystem method makes use of the operator algebra structure of the noise commutant associated with E.

$$\mathcal{A}' = \{\sigma \in \mathcal{B}(\mathcal{H}) : E\sigma = \sigma E, \forall E \in \{E_a, E_a^\dagger\}\}.$$

And when E is unital, all the states encoded in $\mathcal{A}'$ are immune to the errors of E. The structure of $\mathcal{A}$ implies that the noise commutant is unitarily equivalent to

$$\mathcal{A} \cong \bigoplus (1_{m_J} \otimes \mathcal{M}_{n_J}).$$

The elements of $\mathcal{A}'$ are immune to the errors of $\mathcal{A}$ when E is unital. And in [26], the converse of the statement was proved. Specially, when E is unital, the noise commutant coincides with the fixed point set for E.

$$\mathcal{A}' = Fix(E) = \{\sigma \in \mathcal{B}(\mathcal{H}) : E(\sigma) = \textstyle\sum_a E\sigma E^\dagger = \sigma\}.$$

This is why $\mathcal{A}'$ can be used to produce noiseless subsystems for unital E. And the noiseless subsystems may be regarded as containing the method of decoherence-free subspaces as a special case, in the sense that this method uses $1_{m_J} \otimes \mathcal{M}_{n_J}$ where $m_J = 1$ inside the noise commutant $\mathcal{A}'$ for encoding information.

Also, a generalized framework for noiseless subsystems that can be applied to arbitrary quantum operations is brought up. A subsystem that is noiseless for a certain map will also be noiseless for any other map whose Kraus operators are linear combinations of the Kraus operators of the original map. Hence, for the purpose of noiseless encoding, any map whose Kraus operators span is closed under conjugation is equivalent to a unital map.

The structure of $\mathcal{A}$ induces a natural decomposition of the Hilbert space

$$\mathcal{H} = \bigoplus (\mathcal{H}_J^A \otimes \mathcal{H}_J^B),$$

where the noisy subsystems $\mathcal{H}_J^A$ have dimensionm_J and the noiseless subsystems $\mathcal{H}_J^B$ have dimension n_J.

First, the case where information is encoded in a single noiseless sector of $\mathcal{B}(\mathcal{H})$ is considered, and hence

$$\mathcal{H} = (\mathcal{H}^A \otimes \mathcal{H}^B) \oplus \mathcal{K}.$$

with $\dim(\mathcal{H}^A) = m$ and $\dim(\mathcal{H}^B) = n$. We write σ^A for operators in $\mathcal{B}(\mathcal{H}^A)$ and σ^B for operators in $\mathcal{B}(\mathcal{H}^B)$. Thus the restriction of the noise commutant $\mathcal{A}'$ to $\mathcal{H}^A \otimes \mathcal{H}^B$ consists of operators of the form $\sigma = 1^A \otimes \sigma^B$, where 1^A is the identity element of $\mathcal{B}(\mathcal{H}^A)$.

Kribs used the orthonormal bases and matrix representation of the subalgebra $\mathcal{A}'$ to prove the following property:

Lemma 3.1. *The map $\Gamma : \mathcal{B}(\mathcal{H}) \longrightarrow \mathcal{B}(\mathcal{H})$ given by $\Gamma = \{P_{kl}\}$ satisfies the following equality: $\Gamma(\sigma) = \sum_{k,l} P_{kl}\sigma(P_{kl})^\dagger = 1^A \otimes (tr_A \circ \mathcal{P})(\sigma) \in 1^A \otimes \mathcal{B}(\mathcal{H}^B)$, for all operators $\sigma \in \mathcal{B}(\mathcal{H})$, so in particular, $\Gamma(\sigma^A \otimes \sigma^B) \propto 1^A \otimes \sigma^B$ for all σ^A and σ^B, where $\mathcal{P} = \sum_{i=1}^m P_{ii}$, and $P_{kl} = |\alpha_k\rangle\langle\alpha_l| \otimes 1^B$, $\forall 1 \leqslant k, l \leqslant m$ with respect to the orthonormal basis $\{|\alpha_i\rangle\}_{i=1}^m$, so that $\mathcal{P}\mathcal{H} = \mathcal{H}^A \otimes \mathcal{H}^B$.*

Also, Kribs brought up a genearlized noiseless subsystem method (see [31, 32]). In this framework, the quantum information is also assumed to be encoded in σ^B, i.e. the state of noiseless subsystem. But the case that the noisy subsystem remains in the maximally mixed state 1^A under $\mathcal{E}$ is not assumed, as is the case for the noiseless subsystems of unital channels, so it could get mapped to any other state.

Lemma 3.2. *Given a fixed decomposition $\mathcal{H} = (\mathcal{H}^A \otimes \mathcal{H}^B) \oplus \mathcal{K}$, and a quantum operation $\mathcal{E}$ the following properties of the noiseless subsystem B are equivalent:*

1. *$\forall \sigma^A$ and σ^B , $\exists \tau^A$ s.t. $\mathcal{E}(\sigma^A \otimes \sigma^B) = \tau^A \otimes \sigma^B$;*

2. *σ^B , $\exists \tau^A$ s.t. $\mathcal{E}(1^A \otimes \sigma^B) = \tau^A \otimes \sigma^B$;*

3. *$\forall \sigma = \sigma^A \otimes \sigma^B$ for some σ^A and σ^B, $tr_A \circ \mathcal{P} \circ \mathcal{E}(\sigma) = tr_A(\sigma)$.*

Definition 3.3. *The subsystem B is said to be noiseless for $\mathcal{E}$ when it satisfies one of the conditions in 3.2.*

Also, he proved a necessary and sufficient condition for a subsystem to be noiseless for a map $\mathcal{E} = \{E_a\}$.

Theorem 3.1. *Let $\mathcal{E} = \{E_a\}$ be a quantum operation on $\mathcal{B}(\mathcal{H})$ and let $\mathcal{U} = \{\sigma \in \mathcal{B}(\mathcal{H}) : \sigma = \sigma^A \otimes \sigma^B$ for some σ^A and $\sigma^B\}$. Then the following three conditions are equivalent:*

1. *The $B-$sector of $\mathcal{U}$ encodes a noiseless subsystem for $\mathcal{E}$ (decoherence-free subspace in the case $m = 1$).*

2. *The subspace $\mathcal{PH} = \mathcal{H}^A \otimes \mathcal{H}^B$ is invariant for the operators E_a and the restrictions $E_a \mid_{\mathcal{PH}}$ belong to the algebra $\mathcal{B}(\mathcal{H}^A) \otimes 1^B$.*

3. *The following two conditions hold for any choice of matrix units $\{P_{kl} : 1 \leqslant k, l \leqslant m\}$ for $\mathcal{B}(\mathcal{H}^A) \otimes 1^B$:*

$$P_{kk} E_a P_{ll} = \lambda_{akl} P_{kl}, \ \forall a, k, l$$

for some set of scalars (λ_{akl}) and

$$E_a \mathcal{P} = \mathcal{P} E_a \mathcal{P} \ \forall a.$$

Example 3.1. *[32] As a simple illustration of a noiseless subsystem in a non-unital case, consider the quantum channel $\mathcal{E} : \mathcal{M}_4 \longrightarrow \mathcal{M}_4$ with errors $\mathcal{E} = \{E_1, E_2\}$ obtained as follows. Fix $\gamma, 0 \leqslant \gamma \leqslant 1$, and with respect to the basis $\{|0\rangle, |1\rangle\}$, let*

$$F_0 = \begin{bmatrix} \sqrt{\gamma} & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix}, \ F_1 = \begin{bmatrix} 0 & \sqrt{\gamma} \\ \sqrt{1-\gamma} & 0 \end{bmatrix}.$$

and we define $E_i = F_i \otimes I_2$, for $i = 0, 1$. Then $\sum_i E_i^\dagger E_i = I_4$ follows from $\sum_i F_i^\dagger F_i = I_2$. Decompose $\mathbb{C}^4 = \mathcal{H}^A \otimes \mathcal{H}^B$, with respect to the standard basis, so that $\mathcal{H}^A = \mathcal{H}^B = \mathbb{C}^2$. Then for all $\sigma = \sigma^A \otimes \sigma^B$, we have

$$\mathcal{E} = \sum_{i=0}^{1} E_i \sigma^A \otimes \sigma^B E_i^\dagger = (\sum_{i=0}^{1} F_i \sigma^A F_i^\dagger) \otimes \sigma^B.$$

The operator τ is given by $\tau = \sum_{i=0}^{1} F_i \sigma^A F_i^\dagger$ in this case. It follows that B encodes a noiseless subsystem for $\mathcal{E}$. Also as opposed to the completely error-free evolution that characterizes the unital case, we have $\mathcal{E}(I^A \otimes \sigma^B) \neq I^A \otimes \sigma^B$ in this case.

3.3 Theory of Recovery without Error Syndrome Detection

Peres showed that the ability to distinguish non-orthogonal quantum states could be used to construct a cyclic process that would violate the second law of thermodynamics (see [47]). And it is impossible to unambiguously distinguish non-orthogonal quantum states.

When constructing a quantum error correcting code that can detect and correct a set of errors $\{E_a\}$, we must be able to distinguish the error E_a acting on a codeword $|\psi_i\rangle$ from the error E_b acting on a different codeword $|\psi_j\rangle$. Based on this theorem, the erroneous image $E_a |\psi_i\rangle$ and $E_b |\psi_j\rangle$ must be orthogonal if the code is to distinguish these errors correctly.

In [29], Knill and Laflamme gave a necessary and sufficient condition for the existence of a quantum error correcting code.

Theorem 3.2. *Let $\Phi : B(\mathcal{H}) \longrightarrow B(\mathcal{H})$ be a quantum channel. Suppose $\mathcal{V}$ is a subspace of $\mathcal{H}$ and $P_\mathcal{V}$ is the orthogonal projection of $\mathcal{H}$ with $\mathcal{V}$ as the range space. Then the following statements are equivalent:*

1. *$\mathcal{V}$ is a QECC for Φ.*

2. *$P_\mathcal{V} E_a^\dagger E_b P_\mathcal{V} = \lambda_{ab} P_\mathcal{V}$ for some complex number λ_{ab} for all possible E_a, E_b.*

In [36], Li, Nakahara, Poon, Sze and Tomita have slightly modified the above result as follows:

Theorem 3.3. *Let $\Phi : M_n \longrightarrow M_n$ be a quantum channel, and suppose the necessary and sufficient condition for QECC holds and $P = WW^\dagger$ with $W^\dagger W = I_k$ so that a density matrix $\rho \in M_n$ satisfying $P\rho P = \rho$ has the form $W\tilde{\rho}W^\dagger$ with $\tilde{\rho} \in M_k$. Then there is a $R \in U(n)$ and a positive definite matrix $\xi \in M_q$ with $q \leqslant min\{r, \frac{n}{k}\}$ such that for any density matrix $\tilde{\rho} \in M_k$ and $\rho = W\tilde{\rho}W^\dagger \in M_n$, we have*

$$R^\dagger \Phi(\rho) R = (\xi \otimes \tilde{\rho}) \oplus 0_{n-qk}.$$

In particular, if k divides n so that M_n can be regarded as $M_{\frac{n}{k}} \otimes M_k$, then

$$R^\dagger \Phi(\rho) R = \tilde{\xi} \otimes \tilde{\rho}, \text{ with } \tilde{\xi} = \xi \oplus 0_{\frac{n}{k}-q}.$$

A recovery channel can be constructed as the map $\Psi : M_n \longrightarrow M_n$ defined by

$$\Psi(\rho') = W tr_1(R^\dagger(\rho')R)W^\dagger.$$

As a result, a decoding scheme can be realized by a unitary operation followed by a partial trace operation.

3.4 Application to collective noise

This new approach was applied to the study of collective noise. The collective noise was studied by many scholars [39, 40, 56, 61]. In particular, Li et al studied error of the form $\{X^{\otimes n}, Y^{\otimes n}, Z^{\otimes n}\}$ in [34], that is, all qubits constituting the codeword are affected by the same Pauli operator. They showed that (i) an$n$-qubit quantum system can encode $(n-1)$ data qubits when n is odd while (ii) an n-qubit quantum system can encode $(n-2)$ data qubits when n is even. Quantum circuits implementing this scheme were also proposed in their paper.

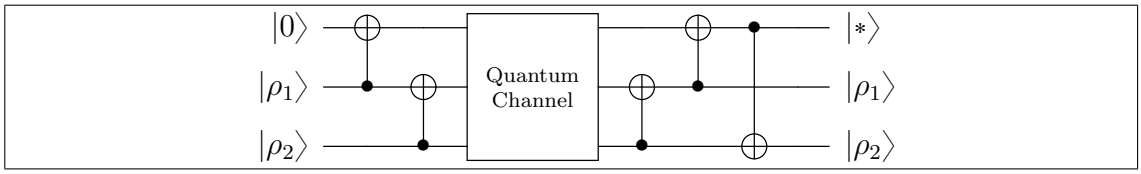


Figure 3.1: An encoding and decoding circuit for 3-qubit quantum channel with error operators $\{X^{\otimes 3}, Y^{\otimes 3}, Z^{\otimes 3}\}$.

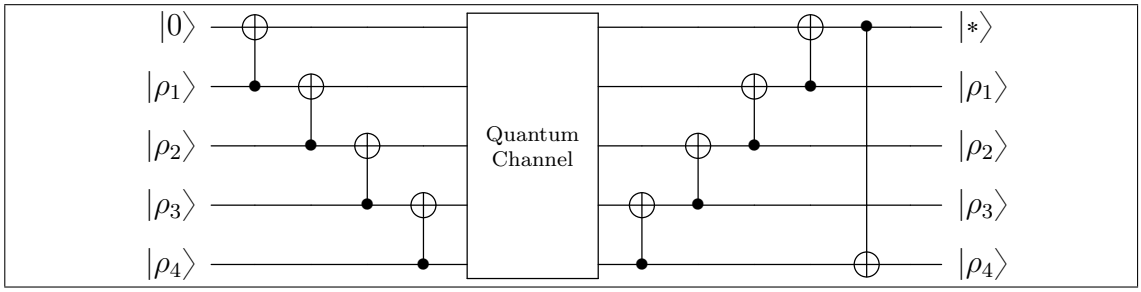


Figure 3.2: An encoding and decoding circuit for 5-qubit quantum channel with error operators $\{X^{\otimes 5}, Y^{\otimes 5}, Z^{\otimes 5}\}$.

In another paper [35] of Li et al., they also studied the general collective noise, namely, error of the form $\{U^{\otimes n} : U \in SU(2)\}$. By consider the decomposition into irreducible representations up to unitary similarity, every error operator has the form $\bigoplus_j I_{r_j} \otimes B_j$ with $B_j \in M_{n_j}$ with $\sum_j r_j n_j = 2^n$. Take $M_2^n \simeq \left(I_{r_j} \otimes M_{n_j}\right) \oplus M_q$ with $q = 2^n - r_j n_j$,. According to this decomposition, this will give raise to a noiseless subsystem. The author also suggested the implementation in terms of quantum circuits for $n = 3$ and all odd n using a recursive construction so that $(n-1)/2$ qubit state can be encoded in the circuits, see Figures 3.5, 3.6 and 3.7. Here, $G_1 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & \sqrt{2} \\ -\sqrt{2} & 1 \end{bmatrix}$ and $G_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$. This scheme is also extended to the study of quantum error correction for qudit in [18, 33].

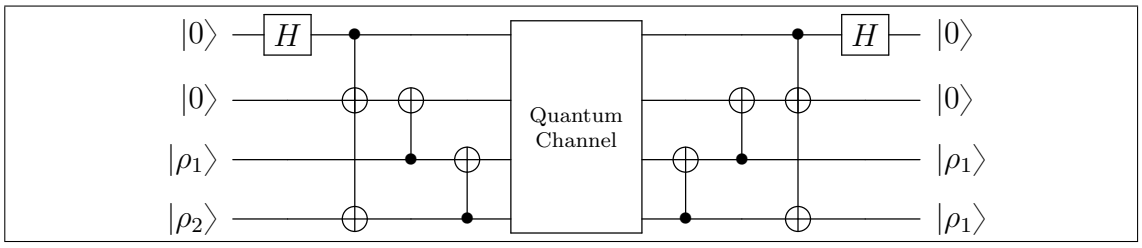


Figure 3.3: An encoding and decoding circuit for 4-qubit quantum channel with error operators $\{X^{\otimes 4}, Y^{\otimes 4}, Z^{\otimes 4}\}$.

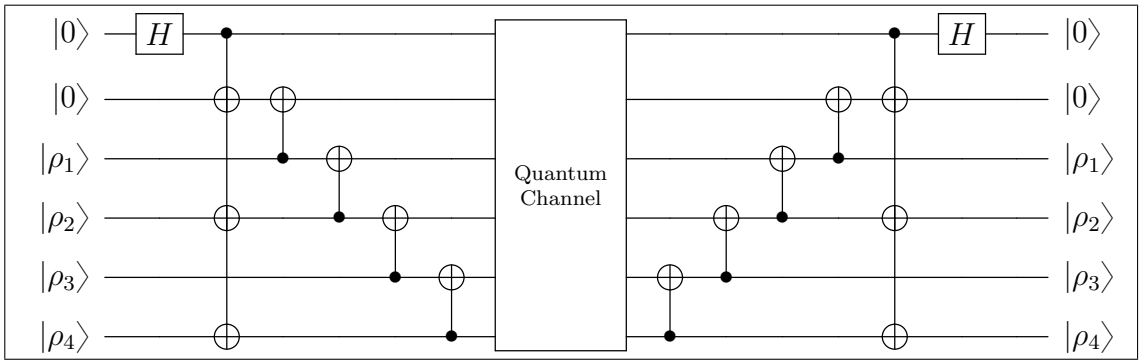


Figure 3.4: An encoding and decoding circuit for 6-qubit quantum channel with error operators $\{X^{\otimes 6}, Y^{\otimes 6}, Z^{\otimes 6}\}$.

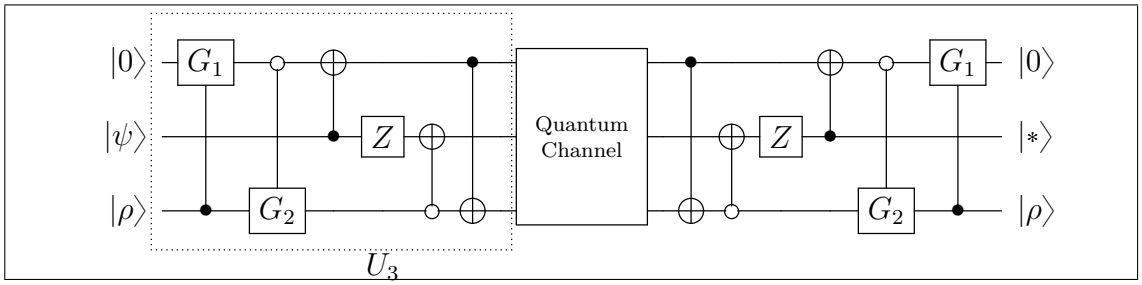


Figure 3.5: An encoding and decoding circuit for 3-qubit quantum channel with error operators $\{U^{\otimes 3} : U \in SU(2)\}$.

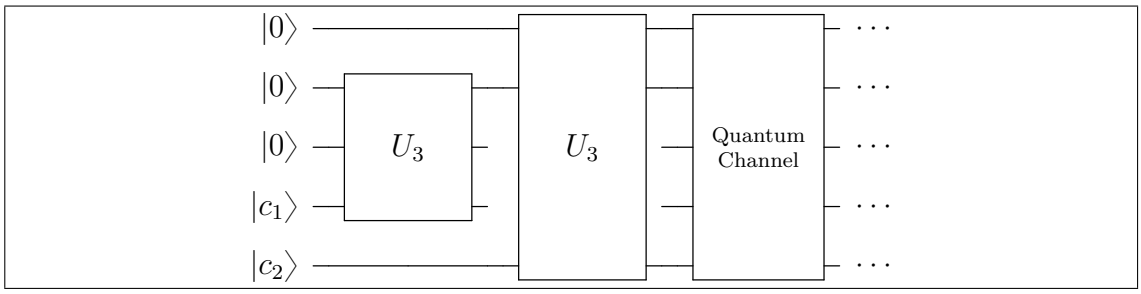


Figure 3.6: An encoding circuit for 5-qubit quantum channel with error operators $\{U^{\otimes 5} : U \in SU(2)\}$.

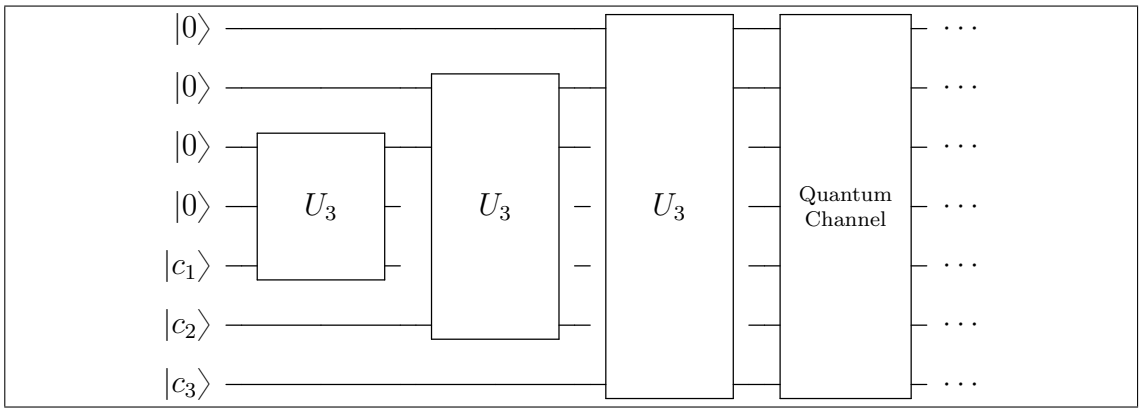


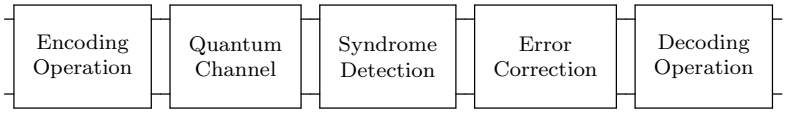
Figure 3.7: An encoding circuit for 7-qubit quantum channel with error operators $\{U^{\otimes 7} : U \in SU(2)\}$.

Chapter 4

Recovery Channel for [5,1,3] Code

4.1 Motivation

For an $[n, k, d]$ code, the classical approach for error correction is syndrome detection and correction method. The information is encoded in the quantum states, then passed the noisy quantum channel. Then one has to measure the error syndrome and correct the error based one the syndrome detected, as stated in the following diagram.



For $[5, 1, 3]$ code, the traditional approach is that we use four ancillary qubits to detect error syndrome, as in Figure 4.1.

So the traditional encoding and error correcting circuit for $[5, 1, 3]$ code is Figure 4.2:

In this approach, other than the $n-k$ ancillary qubits used in encoding, another $n-k$ qubits are needed in order to measure the error syndrome. In this chapter, a new error correction approach for $[n, k, d]$ code will be introduced, namely, an error correction approach without error syndrome detection and correction for $[n, k, d]$

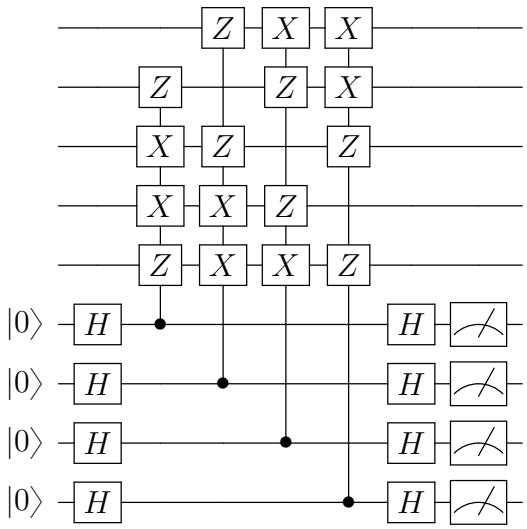
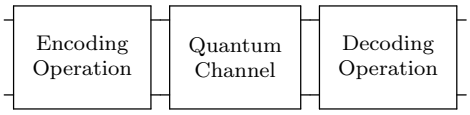


Figure 4.1: An syndrome detection circuit of [5,1,3] code.

code, as showed in the following diagram.



And in 2011, Li, Nakahara, Poon, Sze and Tomita showed that an n-qubit quantum system can encode $(n-1)$ data qubits when n is odd and encode $(n-2)$ data qubits when n is even. AND they can avoid fully correlated noise of the form $\{X^{\otimes n}, Y^{\otimes n}, Z^{\otimes n}\}$ without using ancillary qubits.

In 2013, Kondo, Bagnasco and Nakahara showed that they can avoid fully correlated noise by making use of a three qubit NMR quantum computer experimentally, requiring no equipment of ancillary qubits, see Figure 4.3.

We will first demonstrate this approach for $[5, 1, 3]$ code.

4.2 Demonstration for $[5, 1, 3]$ code

The $[5, 1, 3]$ code whose stabilizer can be generated by

$$g_1 = X \otimes Z \otimes Z \otimes X \otimes I,$$

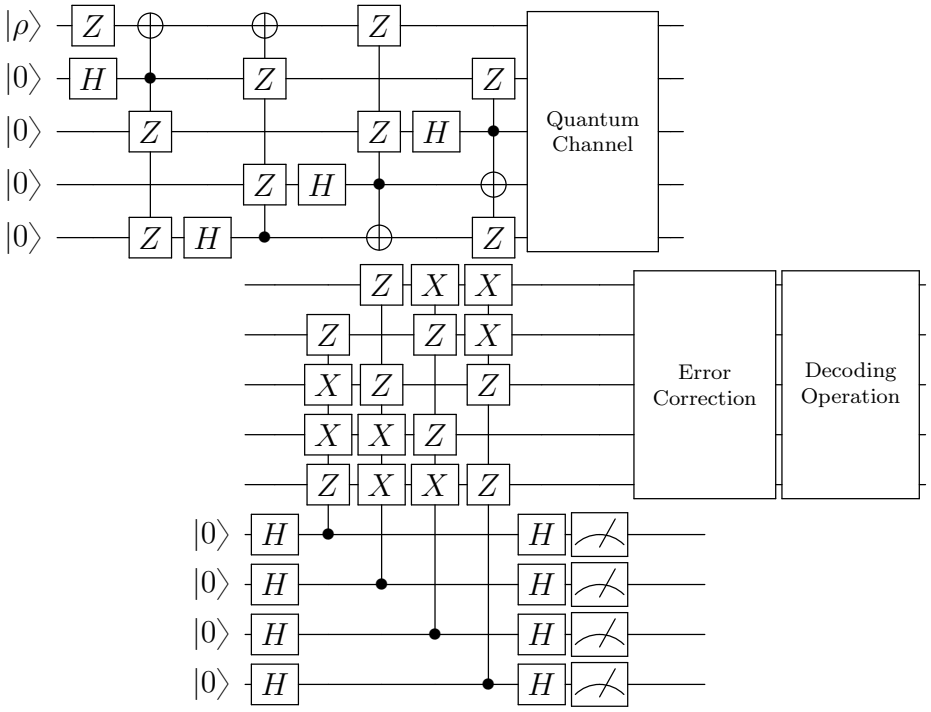


Figure 4.2: An encoding and error correcting circuit of [5,1,3] code.

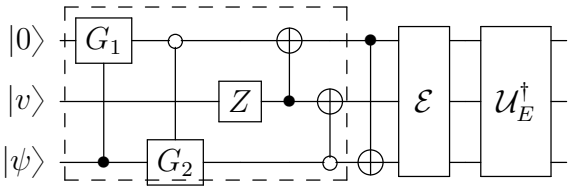


Figure 4.3: NS against fully correlated noise.

$$g_2 = I \otimes X \otimes Z \otimes Z \otimes X,$$

$$g_3 = X \otimes I \otimes X \otimes Z \otimes Z,$$

$$g_4 = Z \otimes X \otimes I \otimes X \otimes Z.$$

The five-qubit code is the smallest possible QECC which corrects one error, see for example [14]. In the following, we show that we can fully recover the original information without detecting error syndromes.

We have the following encoding:

- $|0\rangle_L = \frac{1}{4}(I+g_1)(I+g_2)(I+g_3)(I+g_4)|00000\rangle$.
- $|1\rangle_L = \frac{1}{4}\overline{X}(I+g_1)(I+g_2)(I+g_3)(I+g_4)|00000\rangle$.

Here $\overline{X} = X \otimes X \otimes X \otimes X \otimes X$.

As mentioned in Chapter 2, it can be easily verified that $|0\rangle_L$ and $|1\rangle_L$ are invariant under the action of the four generators. However the factors $(I+g_k)$ above mentioned are not unitary and therefore it is different to be implemented. Practically, it can be verified that the two codewords can be formulated by

$$|0\rangle_L = V_3V_4V_5V_2Z_1|00000\rangle \quad \text{and} \quad |1\rangle_L = V_3V_4V_5V_2Z_1|10000\rangle,$$

where

$$\begin{aligned} V_3 &= (I \otimes I \otimes |0\rangle\langle 0|H \otimes I \otimes I) + (I \otimes Z \otimes |1\rangle\langle 1|H \otimes X \otimes Z)\,, \\ V_4 &= (I \otimes I \otimes I \otimes |0\rangle\langle 0|H \otimes I) + (Z \otimes I \otimes Z \otimes |1\rangle\langle 1|H \otimes X)\,, \\ V_5 &= (I \otimes I \otimes I \otimes I \otimes |0\rangle\langle 0|H) + (X \otimes Z \otimes I \otimes Z \otimes |1\rangle\langle 1|H)\,, \\ V_2 &= (I \otimes |0\rangle\langle 0|H \otimes I \otimes I \otimes I) + (X \otimes |1\rangle\langle 1|H \otimes Z \otimes I \otimes Z)\,. \end{aligned} \tag{4.1}$$

are all unitary operators. Then one can construct the encoding circuit with the above unitary operations, as in Figure 4.4, see also [45].

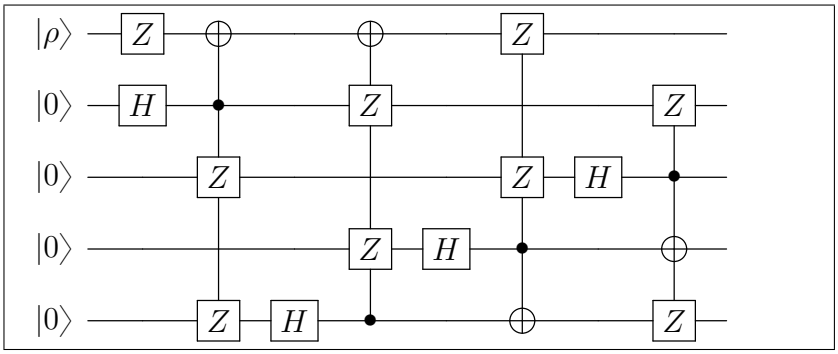


Figure 4.4: An encoding circuit of [5,1,3] code.

4.2.1 Steps to obtain the recovery operations for $[5, 1, 3]$ code.

After we encode the single qubit quantum information, we want to make sure that the receiver receives the correct information. Inevitably, there are single qubit error operators acting on the quantum state during the transmission.

We have the following observation that helps us to build the recovery operations for $[5, 1, 3]$ code, since we have assumed that at most one qubit error can occur, and each error operator can act on each qubit, and there are three types of Pauli operators, thus we have (3×5) possible error operators plus one identity operator. We will consider pairs of codewords $\{E|0\rangle_L, E|1\rangle_L\}$, where E is one of the error operators including the identity operator. There are totally 16 pairs. On the other hand, the Hilbert space under consideration is $2^5 = 32$ dimensional and the quantum subspace spanned by $\{|0\rangle_L, |1\rangle_L\}$ is $2-$dimensional, and we are only interested in this single qubit subspace, while the ancillary subspace is not really important, so we will separate the Hilbert space to 16 different 2-dimensional subspaces spanned by $\{|0j_1j_2j_3j_4\rangle_L, |1j_1j_2j_3j_4\rangle_L\}$, where

$$|0j_1j_2j_3j_4\rangle_L = V_3V_4V_5V_2Z_1|0j_1j_2j_3j_4\rangle \quad \text{and} \quad |1j_1j_2j_3j_4\rangle_L = V_3V_4V_5V_2Z_1|1j_1j_2j_3j_4\rangle,$$

for $j_1, j_2, j_3, j_4 \in \{0, 1\}$. Let

$$\mathcal{S} = \{|0\rangle_L, |1\rangle_L\} \quad \text{and} \quad \mathcal{S}(j_1j_2j_3j_4) = \text{Span}\,\{|0j_1j_2j_3j_4\rangle_L, |1j_1j_2j_3j_4\rangle_L\}.$$

Also denote $E\mathcal{S} = \text{Span}\,\{E|0\rangle_L, E|1\rangle_L\}$ for any error operators E. After comparing these two sets of 2-dimensional subspaces, we obtained the following table indicating the relation between them.

That is, the subspace $X_1\mathcal{S}$ is equal to the subspace $\mathcal{S}(0011)$. Furthermore, the operators displayed in the third column in the table has the following meaning, say

$$X_1|0\rangle_L = XV_3V_4V_5V_2Z_1|00011\rangle \quad \text{and} \quad X_1|1\rangle_L = XV_3V_4V_5V_2Z_1|10011\rangle.$$

$X_1\mathcal{S}$	$\mathcal{S}(0011)$	X
$X_2\mathcal{S}$	$\mathcal{S}(1110)$	X
$X_3\mathcal{S}$	$\mathcal{S}(1011)$	$-Y$
$X_4\mathcal{S}$	$\mathcal{S}(1001)$	$-Y$
$X_5\mathcal{S}$	$\mathcal{S}(1111)$	X
$Y_1\mathcal{S}$	$\mathcal{S}(0010)$	$-Y$
$Y_2\mathcal{S}$	$\mathcal{S}(0110)$	X
$Y_3\mathcal{S}$	$\mathcal{S}(1100)$	$-Y$
$Y_4\mathcal{S}$	$\mathcal{S}(1101)$	$-Y$
$Y_5\mathcal{S}$	$\mathcal{S}(0101)$	X
$Z_1\mathcal{S}$	$\mathcal{S}(0001)$	Z
$Z_2\mathcal{S}$	$\mathcal{S}(1000)$	I
$Z_3\mathcal{S}$	$\mathcal{S}(0111)$	I
$Z_4\mathcal{S}$	$\mathcal{S}(0100)$	I
$Z_5\mathcal{S}$	$\mathcal{S}(1010)$	I

Table 4.1: Relation between the subspaces $E\mathcal{S}$ and $\mathcal{S}(j_1j_2j_3j_4)$.

Remark 4.1. The reason why we can compare seemingly unrelated two sets of codewords is that we find that each set consists of an orthogonal codewords which means that they can form a basis of the whole space. And since each basis can be represented by the other and vice versa, we want to find out what is the difference between the two basis. And from the table above we can see that for the encoded computational basis codewords $|i\rangle_L$ and $|j\rangle_L, i \neq j$, we have

$$\langle i|_L E_a^\dagger E_b |j\rangle_L = 0, \quad \langle i|_L I E_b |j\rangle_L = 0,$$

which verifies the necessary and sufficient conditions for the existence of QECC. Detail will be discussed in the later section.

Next, we want to do is to construct the recovery operation for $[5, 1, 3]$ code. And

we have the following observation: there are ten bit flip errors and six phase flip errors we need to correct during the construction of the recovery operation with respect to the single qubit quantum information. And in order to construct such a recovery operation we have to take all these errors into consideration once and for all. Here we can choose to correct bit flip errors first then deal with phase flip error to extract the original information.

The algorithm is to use the binary system and controlled gates and double controlled gates to transform the ancillary qubits of some corrupted codewords into a fixed state and ancillary qubits of other corrupted codewords will be transformed into some other state, and it won't be the same as the fixed state. Then we use the fixed state ancillary qubits as controlled qubits to act on the single qubit target information to correct the bit flip error, but since there are ten bit flip errors, we won't be able to fix them using just one controlled gate, at least we need two controlled gates to correct the bit flip error.

And when we use two controlled gates, we need to transform three out of four ancillary qubits into a fixed state, which implies a triple controlled gate. But triple controlled gate is hard to find in the process of searching, since it needs to transform three ancillary qubits into the same quantum state, which may take quite a few quantum gates to realize that, so here we use two double controlled gates to fix eight bit flip error. And then use single controlled gate to fix the remaining two.

As for which three ancillary qubits to choose as the controlling qubits, we also have to search based on the number of quantum gates to transform them and the states of the rest of error corrupted encoded codewords after being transformed by these quantum gates, because we need them to correct the rest errors. So in all, we have to construct the quantum gates step by step. But the central idea is clear, we transform ancillary qubits into a fixed state so that we can use them as controlling qubits to fix the bit and phase flip errors for the recovery channel. We now present

the detail procedure.

4.2.2 Detail procedure for the construction

1. First we use two, three or more controlled gates to transform one ancillary qubit into fixed state, then use this state to fix eight bit flip errors out of ten. For example, we can choose ancillary qubits $|0100\rangle$, $|0101\rangle$, $|0110\rangle$, $|0111\rangle$, $|1100\rangle$, $|1101\rangle$, $|1110\rangle$, $|1111\rangle$ to be the transformed quantum state and use the second ancillary qubit to be the control qubit. And to realize the single qubit controlled gate we can use the following circuit to correct eight qubit error first.

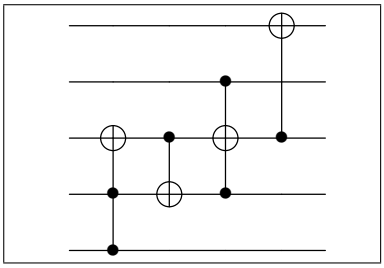


Figure 4.5: Step one of Recovery channel for [5,1,3] code.

2. After we have corrected eight bit flip errors, we want to fix the remaining two, and we need also to transform the remaining two computational basis codewords into a fixed state so that we can use a triple controlled quantum gate to fix the remaining two errors.

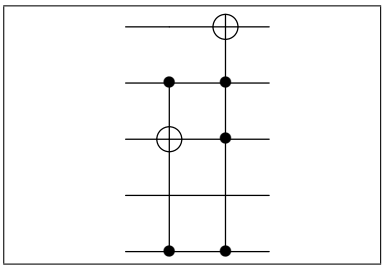


Figure 4.6: Step two of recovery operations for [5,1,3] code.

3. Based on the quantum gates used, we can obtain what the codewords corrupted by phase flip operators have been transformed into. And since we need to correct six phase flip error. Similar to the process of finding recovery channel for bit flip errors, we need to fix four phase flip errors first and then the rest two.

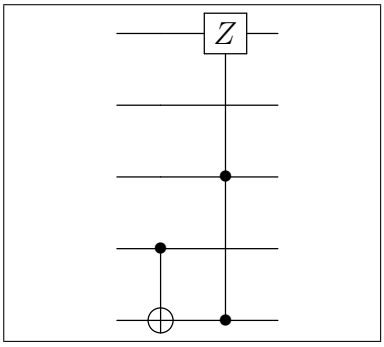


Figure 4.7: Step three of recovery operations for [5,1,3] code.

4. The last step is a bit more complex since we need to find a triple qubit controlled gate to fix two phase flip errors, which means that we need to transform three ancillary qubits into a fixed quantum state so that we can use these three qubits to control the action on the target qubit. And we have

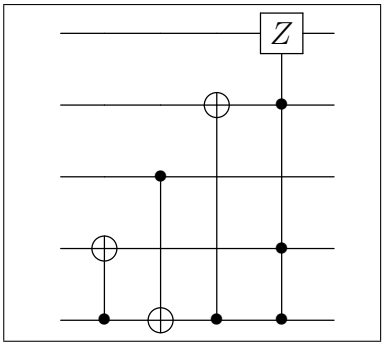


Figure 4.8: Step four of recovery operations for [5,1,3] code.

Remark 4.2. Each step is dependent on the one before and the quantum gates we have searched will in a way have impact on the gates in the following steps, so we

have to be careful about the gates we choose in each step.

And the basic routine here is to correct bit flip and phase flip errors with double controlled gates and triple controlled gates, to derive that, we need to determine which ancillary qubits need to be transformed and what states we want to transform them into.

Since usually we need to fix even number of bit flip errors and phase flip errors, thus we can always decompose the correction into several steps based on the binary expression of the number of errors.

For [5,1,3] code, the decomposition is $2^3 + 2$ and $2^2 + 2$ for bit flip and phase flip errors respectively. So correspondingly, we use a single qubit controlled gate and triple controlled gate to fix bit flip errors and a double controlled gate and triple controlled gate to fix phase flip errors.

Using the 1-1 correspondence between the set of error corrupted encoded codewords and the set of encoded computational basis codewords, we can give a recovery channel without error syndrome detection. All we need to do is to decode and apply the circuit we have found, then the original information is guaranteed to be received correctly.

4.2.3 Circuit diagram for $[5, 1, 3]$ code

After the above discussion in the previous two subsections, we have the following circuit diagram for $[5, 1, 3]$ code.

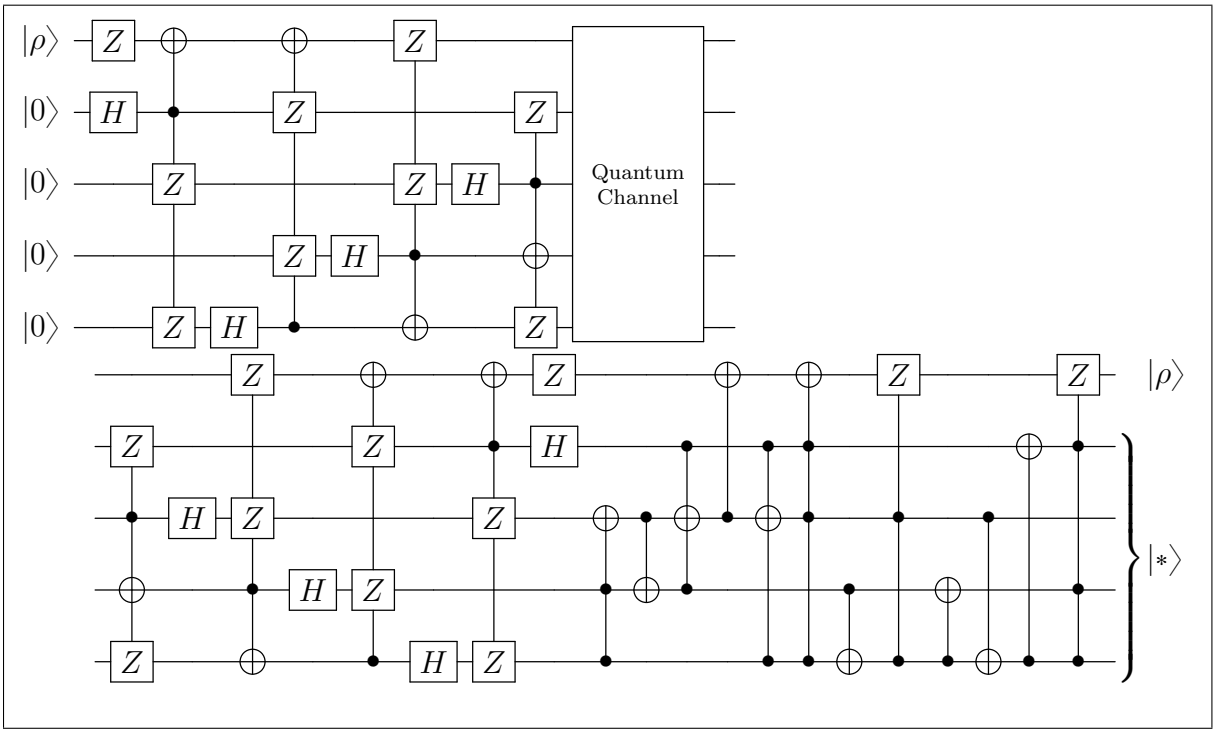


Figure 4.9: An encoding and decoding quantum circuit of [5,1,3] code.

4.3 Theoretical explanation for the circuit diagram construction

Recall that

$$\begin{aligned} g_1 &= I \otimes Z \otimes X \otimes X \otimes Z \\ g_2 &= Z \otimes I \otimes Z \otimes X \otimes X \\ g_3 &= X \otimes Z \otimes I \otimes Z \otimes X \\ g_4 &= X \otimes X \otimes Z \otimes I \otimes Z \end{aligned}$$

In theory, the two codewords $|0\rangle_L$ and $|1\rangle_L$ are defined in principal by

$$\begin{aligned} |0\rangle_L &= \frac{1}{4}(I+g_1)(I+g_2)(1+g_3)(1+g_4)|00000\rangle, \\ |1\rangle_L &= \frac{1}{4}\overline{X}(I+g_1)(I+g_2)(1+g_3)(1+g_4)|00000\rangle. \end{aligned}$$

Notice that for any two distinct error operators E_a and E_b with $w(E_a)$ and $w(E_b)$ are at most one, there exists g_k such that $E_a^\dagger E_b$ is anti-commute with g_k. Then $E_a^\dagger E_b(I+g_k) = (I-g_k)E_i^\dagger E_j$. Then $(I+g_k)^\dagger(I-g_k) = 0$ implies

$$\langle i|_L E_a^\dagger E_b|j\rangle_L = 0 \quad i,j \in \{0,1\}. \tag{4.2}$$

However, as previously mentioned, the factors $(I+g_k)$ are not unitary and therefore it is different to be implemented. Practically, it can be verified that the two codewords can be formulated by

$$|0\rangle_L = V_3V_4V_5V_2Z_1|00000\rangle \quad \text{and} \quad |1\rangle_L = V_3V_4V_5V_2Z_1|10000\rangle$$

where V_3, V_4 ,V_5, and V_2 are unitary operators defined in (4.1). Let $U = V_3V_4V_5V_2Z_1$ and for consistence of notation, set $G_3 = g_1$, $G_4 = g_2$, $G_5 = g_3$, and $G_2 = g_4$. Also let S_i and T_j are the two tensor product components in the definition of V_j, that is, $V_j = S_j + T_j$. It is now claim that for any tensor product of Pauli matrices

$$Q = Q_1 \otimes Q_2 \otimes Q_3 \otimes Q_4 \otimes Q_5 \quad \text{with} \quad Q_j \in \{I, X, Y, Z\},$$

$U^\dagger QU$ is also always a tensor product of PAuli matrices, i.e., we have

$$U^\dagger QU = P_1 \otimes P_2 \otimes P_3 \otimes P_4 \otimes P_5 \quad \text{with} \quad P_j \in \{X, Y, Z\}.$$

We divide the proof into four cases. Fix an index k. If $Q_k = I$, then

$$QV_k = \begin{cases} (S_k + T_k)Q & \text{if } QT_k = T_kQ \\ (S_k - T_k)Q & \text{if } QT_k = -T_kQ \end{cases}$$

and hence

$$V_k^\dagger QV_k = \begin{cases} (S_k + T_k)^\dagger(S_k + T_k)Q = Q & \text{if } QT_k = T_kQ \\ (S_k + T_k)^\dagger(S_k - T_k)Q = X_jQ & \text{if } QT_k = -T_kQ \end{cases}$$

If $Q_k = X$, then

$$V_k^\dagger Q V_k = \begin{cases} Z_k G_k Q & \text{if } (QX_k)T_k = T_k(QX_k) \\ Y_k G_k Q & \text{if } (QX_k)T_k = -T_k(QX_k) \end{cases}$$

If $Q_k = Y$, then

$$V_k^\dagger Q V_k = \begin{cases} -Y_k G_k Q & \text{if } (QY_k)T_j = T_k(QY_k) \\ -Z_k G_k Q & \text{if } (QY_k)T_j = -T_k(QY_k) \end{cases}$$

Finally, if $Q_k = Z$, then

$$V_k^\dagger Q V_k = \begin{cases} X_k Z_k Q & \text{if } (QZ_k)T_j = T_j(QZ_k) \\ Z_k Q & \text{if } (QZ_k)T_j = -T_j(QZ_k) \end{cases}$$

In all cases, $V_k^\dagger Q V_k$ are tensor products of Pauli matrices and and so as $U^\dagger Q U$. Thus, the claim holds. It follows that for any error operator E, for any $j \in \{0,1\}$,

$$\begin{aligned} U^\dagger E U|j0000\rangle &= U(P_1 \otimes P_2 \otimes P_3 \otimes P_4 \otimes P_5)|j0000\rangle \\ &= (P_1|j0000\rangle \otimes |P_2|0\rangle \otimes P_3|0\rangle \otimes P_4|0\rangle \otimes P_5|0\rangle), \end{aligned}$$

therefore

$$U^\dagger E U|j0000\rangle = \pm P_1|j\rangle \otimes |j_2 j_3 j_4 j_5\rangle \quad \text{for some} \quad j_2, j_3, j_4, j_5 \in \{0,1\}. \tag{4.3}$$

Furthermore, for any error operators E_a and E_b, by the fact that $(U^\dagger E_a U)^\dagger (U^\dagger E_b U) = U^\dagger E_a^\dagger E_b U$ and the equation (4.2), one can see that if

$$U^\dagger E_a U|j0000\rangle = \pm P_1^a|j\rangle \otimes |j_2^a j_3^a j_4^a j_5^a\rangle \quad \text{and} \quad U^\dagger E_b U|j0000\rangle = \pm P_1^b|j\rangle \otimes |j_2^b j_3^b j_4^b j_5^b\rangle,$$

then $|j_2^a j_3^a j_4^a j_5^a\rangle$ and $|j_2^b j_3^b j_4^b j_5^b\rangle$ are linearly independent. Thus, if one absorbs the sign of the vector, then

$$\left\{U^\dagger E U|j0000\rangle : \text{all error } E \text{ and } j \in \{0,1\}\right\} = \{|j_1 j_2 j_3 j_4 j_5\rangle : j_1, j_2, j_3, j_4, j_5 \in \{0,1\}\}.$$

E	$U^\dagger EU$	$U^\dagger EU\|00000\rangle$	$U^\dagger EU\|10000\rangle$
X_1	$X\otimes I\otimes I\otimes X\otimes X$	$\|10011\rangle$	$\|00011\rangle$
X_2	$X\otimes Y\otimes X\otimes Y\otimes Z$	$\|11110\rangle$	$\|01110\rangle$
X_3	$Y\otimes X\otimes I\otimes X\otimes Y$	$\|11011\rangle$	$-\|01011\rangle$
X_4	$Y\otimes X\otimes Z\otimes I\otimes Y$	$\|11001\rangle$	$-\|01001\rangle$
X_5	$X\otimes X\otimes X\otimes Y\otimes Y$	$\|11111\rangle$	$\|01111\rangle$
Y_1	$Y\otimes I\otimes I\otimes X\otimes I$	$\|10010\rangle$	$-\|00010\rangle$
Y_2	$X\otimes Z\otimes X\otimes Y\otimes Z$	$\|10110\rangle$	$\|00110\rangle$
Y_3	$Y\otimes X\otimes X\otimes I\otimes Z$	$\|11100\rangle$	$-\|01100\rangle$
Y_4	$Y\otimes X\otimes Y\otimes I\otimes Y$	$\|11101\rangle$	$-\|01101\rangle$
Y_5	$X\otimes I\otimes X\otimes Z\otimes Y$	$\|10101\rangle$	$\|00101\rangle$
Z_1	$Z\otimes I\otimes I\otimes I\otimes X$	$\|00001\rangle$	$-\|10001\rangle$
Z_2	$I\otimes X\otimes I\otimes I\otimes I$	$\|01000\rangle$	$\|11000\rangle$
Z_3	$I\otimes I\otimes X\otimes X\otimes X$	$\|00111\rangle$	$\|10111\rangle$
Z_4	$I\otimes I\otimes X\otimes I\otimes I$	$\|00100\rangle$	$\|10100\rangle$
Z_5	$I\otimes X\otimes I\otimes X\otimes I$	$\|01010\rangle$	$\|11010\rangle$

Table 4.2: Relation between the the error operator E and $U^\dagger EU$.

In fact, direct computations show that following.

In the final stage, we perform further recovery operation R so that all the operation P_1 in equation (4.3) become identity, that is,

$$RU^\dagger EU|j0000\rangle = \pm|j\rangle\otimes|j_2j_3j_4j_5\rangle \quad \text{for some} \quad j_2,j_3,j_4,j_5 \in \{0,1\}.$$

As explained before, the current recovery operations obtained in Figure 4.9 is by computer search.

4.4 A new proposed approach to obtain the circuit diagram for $[n,k,d]$ code

Based on the construction and theoretical explanation for the case of $[5,1,3]$ code, we proposed the following general approach to obtain the circuit diagram for $[n,k,d]$ code.

1. Given a set of generators g_i of the $[n, k, d]$ code is obtained, following the algorithm presented in Section 2.2.3, construct the computational codewords for the code.

2. Define the control-unitary operations based on the generators $(I + g_i)$, similar to Section 4.2. Then the encoding circuit can be obtained accordingly.

3. Compare two sets of codewords $E\mathcal{S}$ and $\mathcal{S}(j_1 \cdots j_{n-k})$ which are mutually orthogonal to each other, and find the mapping relation between them.

4. Build the correspondence between the related two codewords and find the types of corresponding codewords.

5. Based on the types obtained and the number of bit flip and phase flip errors, search for possible quantum gates to apply to the encoded qubits, which may takes double or triple controlled gates to realize that.

In Chapter 5, we will demonstrate that the above proposed algorithm will also work another code, namely, $[8, 3, 3]$ code.

Chapter 5

Recovery Channel for [8,3,3] Code

The purpose of this chapter is to demonstrate the proposed approach for $[8, 3, 3]$ code and give out specific recovery operations for this code.

5.1 Construction for $[8, 3, 3]$ code

As we have mentioned in Chap. 2, generators for an eight-qubit code protecting three-qubit states with at most one error are as follows:

$$\begin{aligned} g_1 &= X \otimes X \otimes X \otimes X \otimes X \otimes X \otimes X \otimes X, \\ g_2 &= Z \otimes Z \otimes Z \otimes Z \otimes Z \otimes Z \otimes Z \otimes Z, \\ g_3 &= X \otimes I \otimes X \otimes I \otimes Z \otimes Y \otimes Z \otimes Y, \\ g_4 &= X \otimes I \otimes Y \otimes Z \otimes X \otimes I \otimes Y \otimes Z, \\ g_5 &= X \otimes Z \otimes I \otimes Y \otimes I \otimes Y \otimes X \otimes Z. \end{aligned} \tag{5.1}$$

and the seed generators given have the following forms:

$$\begin{aligned} N_1 &= X \otimes X \otimes I \otimes I \otimes I \otimes I \otimes I \otimes I, \\ N_2 &= X \otimes I \otimes X \otimes I \otimes I \otimes I \otimes I \otimes I, \\ N_3 &= X \otimes I \otimes I \otimes I \otimes X \otimes I \otimes I \otimes I. \end{aligned} \tag{5.2}$$

Define

$$\begin{aligned}
V_1 &= (I\otimes I\otimes I\otimes I\otimes I\otimes I\otimes |0\rangle\langle 0|\, H)\\
&\quad + (I\otimes Y\otimes X\otimes X\otimes Z\otimes Z\otimes I\otimes |1\rangle\langle 1|\, H)\\
V_2 &= (I\otimes I\otimes I\otimes I\otimes I\otimes I\otimes |0\rangle\langle 0|\, H\otimes I)\\
&\quad + (X\otimes I\otimes Y\otimes X\otimes Z\otimes I\otimes |1\rangle\langle 1|\, H\otimes Z)\,,\\
V_3 &= (I\otimes I\otimes I\otimes I\otimes I\otimes |0\rangle\langle 0|\, H\otimes I\otimes I)\\
&\quad + (X\otimes Y\otimes I\otimes Y\otimes Z\otimes |1\rangle\langle 1|\, H\otimes Z\otimes I)\,,\\
V_4 &= (I\otimes I\otimes I\otimes I\otimes |0\rangle\langle 0|\, H\otimes I\otimes I\otimes I)\\
&\quad + (X\otimes X\otimes Y\otimes Z\otimes |1\rangle\langle 1|\, H\otimes Z\otimes I\otimes I)\,,\\
C_1 &= (|0\rangle\langle 0|\otimes I\otimes I\otimes I\otimes I\otimes I\otimes I\otimes I)\\
&\quad + (|1\rangle\langle 1|\otimes I\otimes I\otimes X\otimes I\otimes I\otimes I\otimes I)\,,\\
C_2 &= (I\otimes |0\rangle\langle 0|\otimes I\otimes I\otimes I\otimes I\otimes I\otimes I)\\
&\quad + (I\otimes |1\rangle\langle 1|\otimes I\otimes X\otimes I\otimes I\otimes I\otimes I)\,,\\
C_3 &= (I\otimes I\otimes |0\rangle\langle 0|\otimes I\otimes I\otimes I\otimes I\otimes I)\\
&\quad + (I\otimes I\otimes |1\rangle\langle 1|\otimes X\otimes I\otimes I\otimes I\otimes I)\,.
\end{aligned}$$

Set $U = V_1V_2V_3V_4C_1C_2C_3$ and define

$$|c_1c_2c_3\rangle_L = U\,|c_1c_2c_300000\rangle \quad c_1, c_2, c_3 \in \{0, 1\}.$$

Then these eight quantum states form an 8-dimensional QECC for $[8, 3, 3]$ code. Let $\mathcal{S} = \{|000\rangle_L, \ldots |111\rangle_L\}$. One can now construct the encoding circuit with the above unitary operations as in Figure 5.1.

After encoding operation, a total of 256 basis codewords

$$U\,|00000000\rangle, U\,|00000001\rangle \cdots, U\,|11111111\rangle$$

are obtained. On the other hand, there are 200 codewords in the set

$$\{E\mathcal{S} : \text{error operators } E\}.$$

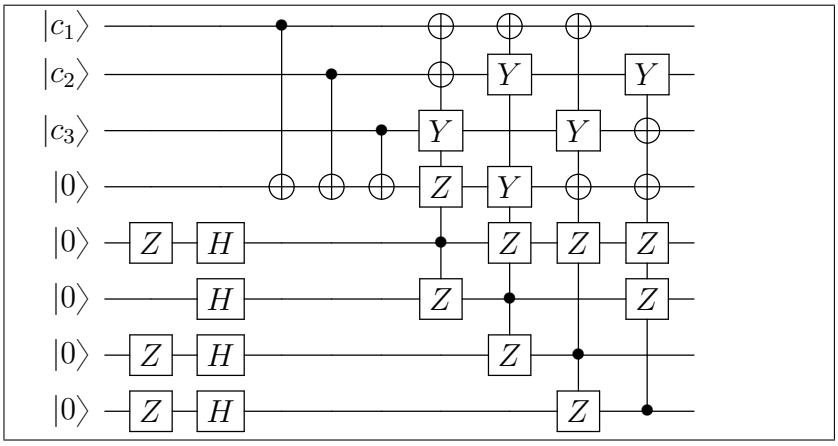


Figure 5.1: An encoding circuit of [8,3,3] code.

Next, we can find out that here exists an injective map between the two sets. So that means we can build an one to one correspondence between two sets of 200 codewords which is the basis for our recovery channel for $[8, 3, 3]$ code. Similar to the case of $[5, 1, 3]$ code, after some computation, we have the following relations between the two sets of 8-dimensional subspaces in Tables 6.1, 6.2, and 6.3.

We can fix the error according to the correspondence and the deviation between each two groups of codewords. The procedure is a little bit longer, basically we have

$X_1\mathcal{S}$	$\mathcal{S}(00001)$	$-X \otimes Y \otimes I$
$X_2\mathcal{S}$	$\mathcal{S}(10101)$	$-Y \otimes I \otimes X$
$X_3\mathcal{S}$	$\mathcal{S}(01011)$	$-Z \otimes X \otimes Y$
$X_4\mathcal{S}$	$\mathcal{S}(00111)$	$X \otimes Y \otimes Y$
$X_5\mathcal{S}$	$\mathcal{S}(11111)$	$I \otimes I \otimes I$
$X_6\mathcal{S}$	$\mathcal{S}(10011)$	$X \otimes I \otimes I$
$X_7\mathcal{S}$	$\mathcal{S}(01101)$	$I \otimes X \otimes I$
$X_8\mathcal{S}$	$\mathcal{S}(11001)$	$I \otimes I \otimes X$

Table 5.1: Relation between the subspaces $X_j\mathcal{S}$ and $\mathcal{S}(j_1j_2j_3j_4j_5)$.

$Y_1\mathcal{S}$	$\mathcal{S}(10001)$	$X \otimes Y \otimes I$
$Y_2\mathcal{S}$	$\mathcal{S}(11101)$	$Y \otimes I \otimes X$
$Y_3\mathcal{S}$	$\mathcal{S}(01111)$	$Z \otimes X \otimes Y$
$Y_4\mathcal{S}$	$\mathcal{S}(00101)$	$-X \otimes Y \otimes Y$
$Y_5\mathcal{S}$	$\mathcal{S}(00011)$	$Z \otimes Z \otimes Z$
$Y_6\mathcal{S}$	$\mathcal{S}(01001)$	$-Y \otimes I \otimes I$
$Y_7\mathcal{S}$	$\mathcal{S}(11011)$	$-I \otimes Y \otimes I$
$Y_8\mathcal{S}$	$\mathcal{S}(10111)$	$-I \otimes I \otimes Y$

Table 5.2: Relation between the subspaces $Y_j\mathcal{S}$ and $\mathcal{S}(j_1j_2j_3j_4j_5)$.

$Z_1\mathcal{S}$	$\mathcal{S}(10000)$	$I \otimes I \otimes I$
$Z_2\mathcal{S}$	$\mathcal{S}(01000)$	$I \otimes I \otimes I$
$Z_3\mathcal{S}$	$\mathcal{S}(00100)$	$I \otimes I \otimes I$
$Z_4\mathcal{S}$	$\mathcal{S}(00010)$	$I \otimes I \otimes I$
$Z_5\mathcal{S}$	$\mathcal{S}(11100)$	$Z \otimes Z \otimes Z$
$Z_6\mathcal{S}$	$\mathcal{S}(11010)$	$Z \otimes I \otimes I$
$Z_7\mathcal{S}$	$\mathcal{S}(10110)$	$I \otimes Z \otimes I$
$Z_8\mathcal{S}$	$\mathcal{S}(01110)$	$I \otimes I \otimes Z$

Table 5.3: Relation between the subspaces $Z_j\mathcal{S}$ and $\mathcal{S}(j_1j_2j_3j_4j_5)$.

two approaches, fix the bit flip error then the phase flip error or phase flip error first then bit flip error.

Remark 5.1. Similar to $[5, 1, 3]$ code, The reason why we compare seemingly unrelated two sets of codewords is that each set consists of codewords which are mutually orthogonal to each other. And the difference between $[8, 3, 3]$ code and $[5, 1, 3]$ code is that for $[5, 1, 3]$ code, the two sets have the same number of codewords, while

for $[8, 3, 3]$ code, there are $3 \times 2^3 \times 2^3$ error corrupted encoded codewords, and the encoded computational basis codewords are 2^8, which means that some encoded computational basis codewords won't be used or be mapped to. And we find out that the corrupted set is a subset of the set consisting of encoded basis codewords. Although it is not one to one correspondence, but we can still apply the similar procedure to find the recovery channel for $[8, 3, 3]$ code.

Then we want to construct the recovery channel for $[8, 3, 3]$ code. And we have the following observation: different from $[5, 1, 3]$ code, there are three qubit information we need to encode and for that we need to apply three rounds of recovery for each qubit, the idea seems straightforward, and it is hard to correct all the bit flip errors for the three qubit once and for all then correct the phase flip errors for them. Because the same bit flip error operator on different qubits have different types of results, which lead to the difficulty of correct one type of error at the same time and the other type of error after that.

Based on the table and types we observe that there are eight bit flip errors and eight phase flip errors we need to correct for each qubit during the construction of the recovery channel with respect to three qubit quantum information. And in order to construct such a recovery channel we need to correct bit flip and phase flip errors for the three qubit one by one. And in each round of correction we can choose to correct bit flip errors first then deal with phase flip error to extract the original information.

Similary, the algorithm is to controlled gates and double controlled gates and triple controlled gates to transform the ancillary qubits of some corrupted codewords into a fixed state, and we use the fixed state ancillary qubits as controlled qubits to act on each single qubit target information to correct the bit flip error and phase flip error, but since there are eight bit flip errors and eight phase flip errors, it is possible that we correct the eight errors using just one double controlled quantum gate, but in

reality, we could not find such gates for all three qubits, since the action of quantum gates on each corrupted codeword will impact the state of ancillary qubits thus have influence on the choice of quantum gates for the remaining two or one qubit ,which will result in the difficulty of finding quantum gates for the next qubit, and so on and so forth. So we have to turn around to also use triple controlled gates in the process of correction.

And when we use controlled gates, we need to transform two out of five ancillary qubits into a fixed state for a double controlled gate or three out of five ancillary qubits into a fixed state for a triple controlled gate. But triple controlled gate is hard to find in the process of searching, especially for the later corrections. And since it need to transform two or three ancillary qubits into the same quantum state, which may take several quantum gates to realize that, and here we use one triple controlled gate and two double controlled gates to correct bit flip and phase flip errors respectively for the first two qubits, and two double controlled gates to correct bit flip errors and two double controlled gates to correct phase flip errors for the third qubit.

Also, as for which ancillary qubits need to be chosen as the controlling qubits, we have to determine based on the number of quantum gates to transform them, and the number of gates is the smaller the better. And we also need to observe the states of the rest of error corrupted encoded codewords after being transformed by these quantum gates, because we need them to correct the rest errors. So all in all, we have to construct the quantum gates for the three qubits one by one and step by step. But the main idea is similar, we transform ancillary qubits into a fixed state so that we can use them as controlling qubits to fix the bit and phase flip errors for the recovery channel.

5.1.1 procedure to construct recovery channel for [8,3,3] code

1. First we use two, three or more controlled gates to transform two ancillary qubits into a fixed state, then use these two ancillary qubits to fix eight bit flip errors. For example, we can choose ancillary qubits $|00001\rangle$, $|00011\rangle$, $|00101\rangle$, $|00111\rangle$, $|10001\rangle$, $|10011\rangle$, $|10101\rangle$, $|10111\rangle$ to be the transformed quantum state and use the second and fifth ancillary qubits to be the control qubits. And to obtain the double qubit controlled gate, we can use the following circuit to correct eight bit flip errors first (Figure 5.2).

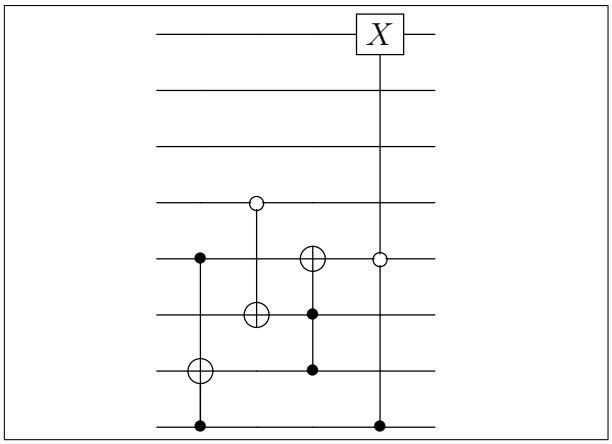


Figure 5.2: Step one of recovery channel for [8,3,3] code.

2. After we have corrected eight bit flip error operators for the first qubit, what we need to do is to correct eight phase flip errors. But we find that after the action of quantum gates in the step one, we can not find a double controlled gate to correct them. So we have to use two triple controlled gates to realize it (Figure 5.3).

3. Based on the quantum gates used for the correction of errors for the first qubit, we can obtain for the second qubit what the codewords related to the second qubit corrupted by error operators have been transformed into. And similarly we need to correct eight bit flip error first (Figure 5.4).

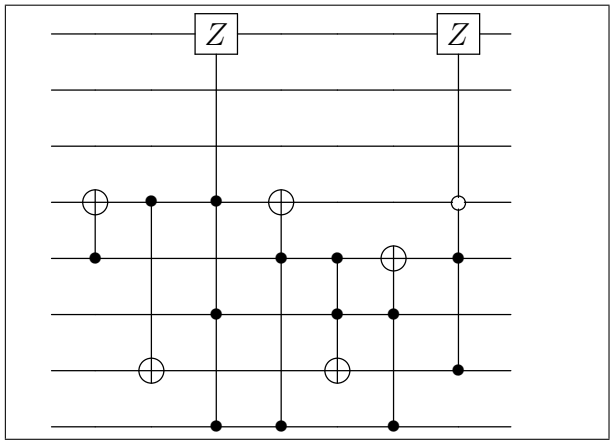


Figure 5.3: Step two of recovery channel for [8,3,3] code.

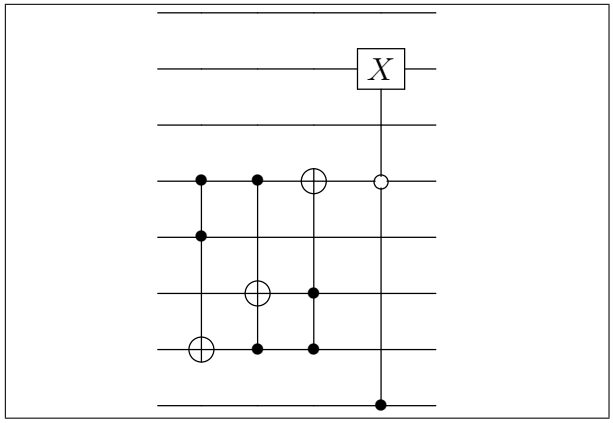


Figure 5.4: Step three of recovery channel for [8,3,3] code.

4. What we need to do now is to correct eight phase flip errors for the second qubit. Similarly we find that after the action of quantum gates applied to the second qubit for the bit error correction, we can not find a double controlled gate to correct them. So we have to use two triple controlled gates to realize it (Figure 5.5).

5. For the third qubit, it is a little bit more complex than the first two qubits, since we can not find a double controlled gate to correct eight bit flip errors at the same time. We need two triple controlled gates to correct them (Figure 5.5).

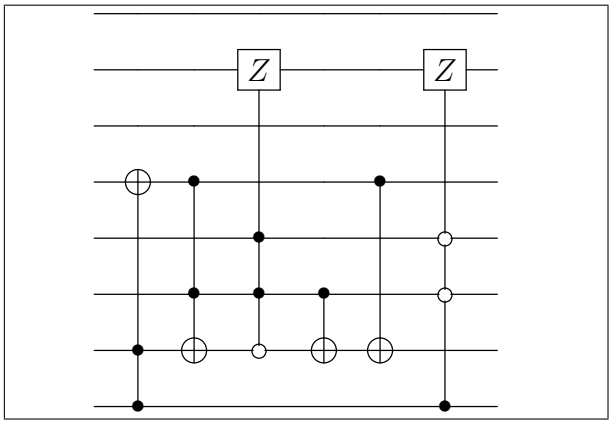


Figure 5.5: Step four of recovery channel for [8,3,3] code.

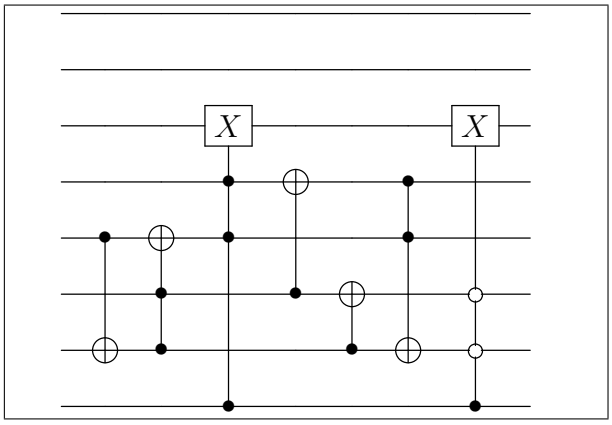


Figure 5.6: Step five of recovery channel for [8,3,3] code.

6. And finally for the third qubit, similarly we need to use two triple controlled gates to correct eight phase flip errors (Figure 5.6).

Finally, the complete realization of encoding and decoding circuit diagram without error syndrome detection for $[8, 3, 3]$ code is presented in Figure 5.8.

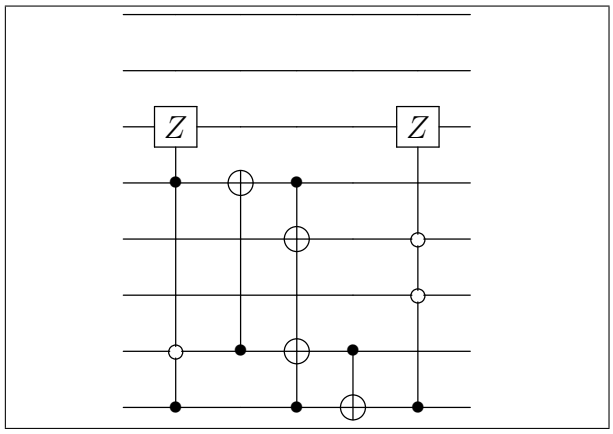


Figure 5.7: Step six of recovery channel for [8,3,3] code.

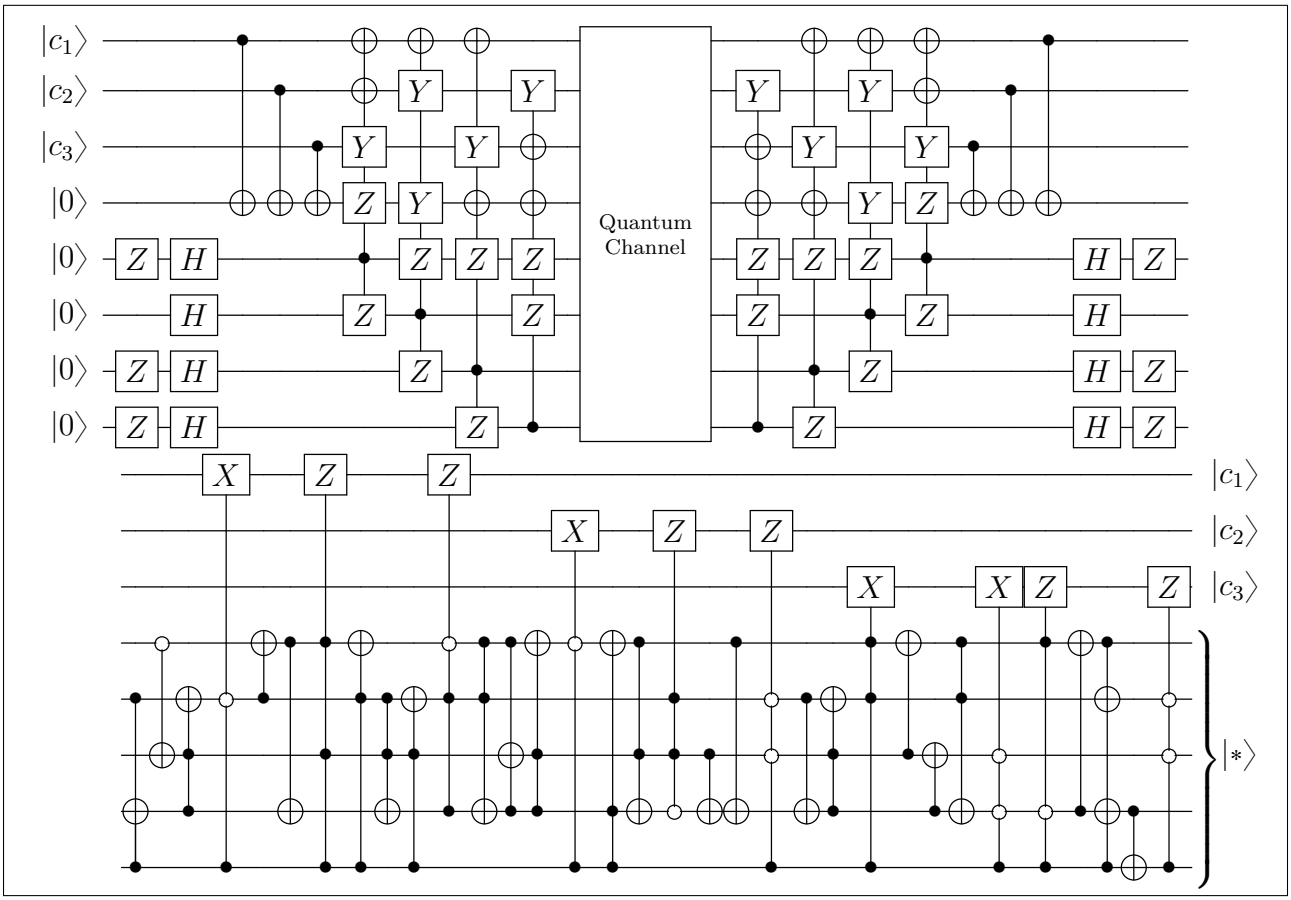


Figure 5.8: An encoding and decoding quantum circuit of $[8, 3, 3]$ code.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, stabilizer codes are reviewed and a scheme for constructing recovery channels without error syndrome detection is proposed. We first review some basic concepts on stabilizer groups and stabilizer codes. In particular, we consider theories and principles involved in the construction of encoding circuits from the generators of stabilizer group, and propose a new procedure to derive recovery channel for a well known quantum code, the $[n, k, d]$ code.

Next, an algorithm to obtain the generators for a stabilizer code and the corresponding computational basis codewords defined in terms of Pauli operators are reviewed and illustrated in detail. Then based on the general framework of operator quantum error correction, we provide a general scheme on the construction of encoding and decoding circuits for the $[n, k, d]$ codes.

Finally, a detailed procedure to construct the recovery channel using encoding circuits and encoded computational basis codewords are demonstrated for $[5, 1, 3]$ code and $[8, 3, 3]$ code step by step as examples, with heuristic explanations based on necessary and sufficient conditions for quantum error correction.

Contrary to the traditional approach to error correction, the scheme saves $(n-k)$ ancillary qubits that are used in the error syndrome detection. Although there might

be some time tradeoff, the computability of number of quantum gates in recovery channel for simple quantum error correcting codes is obvious. So we can use limited number of quantum gates and n qubits, without using ancillary qubits, to recover the original information.

6.2 Future Work

After studied $[5, 1, 3]$ and $[8, 3, 3]$ code, the next possible code to be studied is $[10, 4, 3]$. Following the algorithm proposed in Section 4.4, we can first obtain an encoding circuit diagram in Figure 6.1.

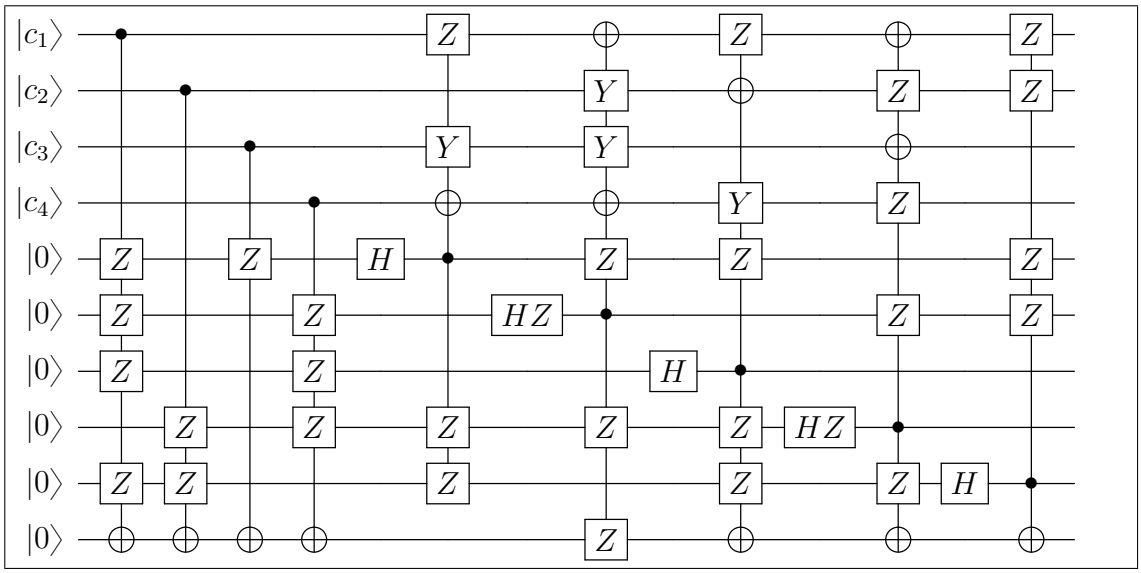


Figure 6.1: An encoding circuit of [10,4,3] code.

Then we can compare the two sets of codewords $E\mathcal{S}$ and $\mathcal{S}(j_1j_2j_3j_4j_5j_6)$ and obtained the relation listed in Tables 6.1, 6.2, and 6.3. The remaining and the most difficult part is to construct and search for possible quantum operations that can fix these bit flip and phase flip errors listed in the tables. The main difficulty is that we have to consider 2^6 dimensional subspace, which the computer program searching time is huge. So alternative method should be explored to find the right recovery

operations.

Actually, apart from the $[10, 4, 3]$ code, the $[11, 1, 5]$ code is also another target code that is under consideration. Notice that there are a total of 627 different error operators for this code. Therefore, although only one qubit state have to be fixed, the computational complexity is also huge.

$X_1\mathcal{S}$	$\mathcal{S}(101110)$	$I \otimes I \otimes X \otimes X$
$X_2\mathcal{S}$	$\mathcal{S}(101010)$	$X \otimes Y \otimes Y \otimes X$
$X_3\mathcal{S}$	$\mathcal{S}(001001)$	$I \otimes X \otimes I \otimes X$
$X_4\mathcal{S}$	$\mathcal{S}(111001)$	$Y \otimes Z \otimes Y \otimes Z$
$X_5\mathcal{S}$	$\mathcal{S}(010101)$	$I \otimes I \otimes I \otimes I$
$X_6\mathcal{S}$	$\mathcal{S}(010001)$	$Z \otimes I \otimes I \otimes Z$
$X_7\mathcal{S}$	$\mathcal{S}(111111)$	$X \otimes I \otimes Z \otimes Z$
$X_8\mathcal{S}$	$\mathcal{S}(111011)$	$I \otimes Y \otimes Z \otimes I$
$X_9\mathcal{S}$	$\mathcal{S}(110001)$	$I \otimes I \otimes Y \otimes Z$
$X_{10}\mathcal{S}$	$\mathcal{S}(110101)$	$I \otimes Z \otimes Z \otimes X$

Table 6.1: Relation between the subspaces $X_j\mathcal{S}$ and $\mathcal{S}(j_1j_2j_3j_4j_5j_6)$.

$Y_1\mathcal{S}$	$\mathcal{S}(001110)$	$Z \otimes I \otimes Y \otimes X$
$Y_2\mathcal{S}$	$\mathcal{S}(111010)$	$Y \otimes Y \otimes Y \otimes Y$
$Y_3\mathcal{S}$	$\mathcal{S}(000001)$	$Z \otimes X \otimes I \otimes Y$
$Y_4\mathcal{S}$	$\mathcal{S}(000101)$	$X \otimes I \otimes X \otimes I$
$Y_5\mathcal{S}$	$\mathcal{S}(000011)$	$I \otimes I \otimes I \otimes I$
$Y_6\mathcal{S}$	$\mathcal{S}(110011)$	$I \otimes I \otimes I \otimes I$
$Y_7\mathcal{S}$	$\mathcal{S}(010011)$	$Y \otimes Z \otimes I \otimes Z$
$Y_8\mathcal{S}$	$\mathcal{S}(100011)$	$I \otimes X \otimes Z \otimes I$
$Y_9\mathcal{S}$	$\mathcal{S}(111101)$	$Z \otimes Z \otimes X \otimes Z$
$Y_{10}\mathcal{S}$	$\mathcal{S}(001101)$	$Z \otimes Z \otimes I \otimes Y$

Table 6.2: Relation between the subspaces $Y_j\mathcal{S}$ and $\mathcal{S}(j_1j_2j_3j_4j_5j_6)$.

$Z_1\mathcal{S}$	$\mathcal{S}(100000)$	$Z \otimes I \otimes Z \otimes I$
$Z_2\mathcal{S}$	$\mathcal{S}(010000)$	$Z \otimes I \otimes I \otimes Z$
$Z_3\mathcal{S}$	$\mathcal{S}(001000)$	$Z \otimes I \otimes I \otimes Z$
$Z_4\mathcal{S}$	$\mathcal{S}(111100)$	$Z \otimes Z \otimes Z \otimes Z$
$Z_5\mathcal{S}$	$\mathcal{S}(010110)$	$I \otimes I \otimes I \otimes I$
$Z_6\mathcal{S}$	$\mathcal{S}(100010)$	$Z \otimes I \otimes I \otimes Z$
$Z_7\mathcal{S}$	$\mathcal{S}(101100)$	$Z \otimes Z \otimes Z \otimes I$
$Z_8\mathcal{S}$	$\mathcal{S}(011000)$	$I \otimes Z \otimes I \otimes I$
$Z_9\mathcal{S}$	$\mathcal{S}(001100)$	$Z \otimes Z \otimes Z \otimes I$
$Z_{10}\mathcal{S}$	$\mathcal{S}(111000)$	$Z \otimes I \otimes Z \otimes Z$

Table 6.3: Relation between the subspaces $Z_j\mathcal{S}$ and $\mathcal{S}(j_1j_2j_3j_4j_5j_6)$.

Appendix A

Matlab code for searching recovery operations of $[5, 1, 3]$ and $[8, 3, 3]$ codes

In this appendix, we present the Matlab code used for searching recovery operations of $[5, 1, 3]$ and $[8, 3, 3]$ codes in Chapters 4 and 5.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Setup of operators, single qubit controlled gates, double qubit and triple controlled gates
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
I = [1 0; 0 1];
X = [0 1; 1 0];
Z = [1 0; 0 -1];
Y = [0 -1; 1 0];
e0 = [1 0]';
e1 = [0 1]';
V1 = kron(kron(kron(e0,e0),e0),e0);
V2 = kron(kron(kron(e0,e0),e0),e1);
V3 = kron(kron(kron(e0,e0),e1),e0);
V4 = kron(kron(kron(e0,e0),e1),e1);
V5 = kron(kron(kron(e0,e1),e0),e0);
V6 = kron(kron(kron(e0,e1),e0),e1);
```

```
V7 = kron(kron(kron(e0,e1),e1),e0);
V8 = kron(kron(kron(e0,e1),e1),e1);
V9 = kron(kron(kron(e1,e0),e0),e0);
V10 =kron(kron(kron(e1,e0),e0),e1);
V11 =kron(kron(kron(e1,e0),e1),e0);
V12 =kron(kron(kron(e1,e0),e1),e1);
V13 =kron(kron(kron(e1,e1),e0),e0);
V14 =kron(kron(kron(e1,e1),e0),e1);
V15 =kron(kron(kron(e1,e1),e1),e0);
V16 =kron(kron(kron(e1,e1),e1),e1);
V=[V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16];
V11 =kron(kron(kron(e1,e0),e1),e0);
V7 = kron(kron(kron(e0,e1),e1),e0);
V5 = kron(kron(kron(e0,e1),e0),e0);
V6 = kron(kron(kron(e0,e1),e0),e1);
V14 =kron(kron(kron(e1,e1),e0),e1);
V12 =kron(kron(kron(e1,e0),e1),e1);
V15 =kron(kron(kron(e1,e1),e1),e0);
V4 = kron(kron(kron(e0,e0),e1),e1);
V10 =kron(kron(kron(e1,e0),e0),e1);
V16 =kron(kron(kron(e1,e1),e1),e1);
E0 = [1 0; 0 0];
E1 = [0 0; 0 1];
CONT1 = kron(kron(kron(E0,I),I),I)+kron(kron(kron(E1,I),I),X);
CONT2 = kron(kron(kron(E0,I),I),I)+kron(kron(kron(E1,I),X),I);
CONT3 = kron(kron(kron(E0,I),I),I)+kron(kron(kron(E1,X),I),I);
CONT4 = kron(kron(kron(I,E0),I),I)+kron(kron(kron(I,E1),I),X);
CONT5 = kron(kron(kron(I,E0),I),I)+kron(kron(kron(I,E1),X),I);
CONT6 = kron(kron(kron(I,E0),I),I)+kron(kron(kron(X,E1),I),I);
CONT7 = kron(kron(kron(I,I),E0),I)+kron(kron(kron(I,I),E1),X);
CONT8 = kron(kron(kron(I,I),E0),I)+kron(kron(kron(I,X),E1),I);
CONT9 = kron(kron(kron(I,I),E0),I)+kron(kron(kron(X,I),E1),I);
CONT10 = kron(kron(kron(I,I),I),E0)+kron(kron(kron(I,I),X),E1);
CONT11 = kron(kron(kron(I,I),I),E0)+kron(kron(kron(I,X),I),E1);
```

```
CONT12 = kron(kron(kron(I,I),I),E0)+kron(kron(kron(X,I),I),E1);
CONT = [CONT1; CONT2; CONT3; CONT4; CONT5; CONT6;
CONT7; CONT8; CONT9; CONT10; CONT11; CONT12;]
DCONT1 = kron(kron(kron(E1,E1),I),X)+kron(kron(kron(E0,E0),I),I)
+kron(kron(kron(E0,E1),I),I)+kron(kron(kron(E1,E0),I),I);
DCONT2 = kron(kron(kron(E1,E1),X),I)+kron(kron(kron(E0,E0),I),I)
+kron(kron(kron(E0,E1),I),I)+kron(kron(kron(E1,E0),I),I);
DCONT3 = kron(kron(kron(E1,I),E1),X)+kron(kron(kron(E0,I),E0),I)
+kron(kron(kron(E0,I),E1),I)+kron(kron(kron(E1,I),E0),I);
DCONT4 = kron(kron(kron(E1,X),E1),I)+kron(kron(kron(E0,I),E0),I)
+kron(kron(kron(E0,I),E1),I)+kron(kron(kron(E1,I),E0),I);
DCONT5 = kron(kron(kron(E1,I),X),E1)+kron(kron(kron(E0,I),I),E0)
+kron(kron(kron(E0,I),I),E1)+kron(kron(kron(E1,I),I),E0);
DCONT6 = kron(kron(kron(E1,X),I),E1)+kron(kron(kron(E0,I),I),E0)
+kron(kron(kron(E0,I),I),E1)+kron(kron(kron(E1,I),I),E0);
DCONT7 = kron(kron(kron(I,E1),E1),X)+kron(kron(kron(I,E0),E0),I)
+kron(kron(kron(I,E0),E1),I)+kron(kron(kron(I,E1),E0),I);
DCONT8 = kron(kron(kron(X,E1),E1),I)+kron(kron(kron(I,E0),E0),I)
+kron(kron(kron(I,E0),E1),I)+kron(kron(kron(I,E1),E0),I);
DCONT9 = kron(kron(kron(I,E1),X),E1)+kron(kron(kron(I,E0),I),E0)
+kron(kron(kron(I,E0),I),E1)+kron(kron(kron(I,E1),I),E0);
DCONT10 = kron(kron(kron(X,E1),I),E1)+kron(kron(kron(I,E0),I),E0)
+kron(kron(kron(I,E0),I),E1)+kron(kron(kron(I,E1),I),E0);
DCONT11 = kron(kron(kron(I,X),E1),E1)+kron(kron(kron(I,I),E0),E0)
+kron(kron(kron(I,I),E0),E1)+kron(kron(kron(I,I),E1),E0);
DCONT12 = kron(kron(kron(X,I),E1),E1)+kron(kron(kron(I,I),E0),E0)
+kron(kron(kron(I,I),E0),E1)+kron(kron(kron(I,I),E1),E0);
DCONT = [DCONT1; DCONT2; DCONT3; DCONT4; DCONT5; DCONT6;
DCONT7; DCONT8; DCONT9; DCONT10; DCONT11; DCONT12;]
TCONT1 = kron(kron(kron(E1,E1),E1),X)+kron(kron(kron(E0,E0),E0),I)
+kron(kron(kron(E0,E0),E1),I)+kron(kron(kron(E0,E1),E0),I)
+kron(kron(kron(E0,E1),E1),I)+kron(kron(kron(E1,E0),E0),I)
+kron(kron(kron(E1,E0),E1),I)+kron(kron(kron(E1,E1),E0),I);
TCONT2 = kron(kron(kron(E1,E1),X),E1)+kron(kron(kron(E0,E0),I),E0)
```

```
+kron(kron(kron(E0,E0),I),E1)+kron(kron(kron(E0,E1),I),E0)
+kron(kron(kron(E0,E1),I),E1)+kron(kron(kron(E1,E0),I),E0)
+kron(kron(kron(E1,E0),I),E1)+kron(kron(kron(E1,E1),I),E0);
TCONT3 = kron(kron(kron(E1,X),E1),E1)+kron(kron(kron(E0,I),E0),E0)
+kron(kron(kron(E0,I),E0),E1)+kron(kron(kron(E0,I),E1),E0)
+kron(kron(kron(E0,I),E1),E1)+kron(kron(kron(E1,I),E0),E0)
+kron(kron(kron(E1,I),E0),E1)+kron(kron(kron(E1,I),E1),E0);
TCONT4 = kron(kron(kron(X,E1),E1),E1)+kron(kron(kron(I,E0),E0),E0)
+kron(kron(kron(I,E0),E0),E1)+kron(kron(kron(I,E0),E1),E0)
+kron(kron(kron(I,E0),E1),E1)+kron(kron(kron(I,E1),E0),E0)
+kron(kron(kron(I,E1),E0),E1)+kron(kron(kron(I,E1),E1),E0);
CONT = [CONT1; CONT2; CONT3; CONT4; CONT5; CONT6;
CONT7; CONT8; CONT9; CONT10; CONT11; CONT12;]
DCONT = [DCONT1; DCONT2; DCONT3; DCONT4; DCONT5; DCONT6;
DCONT7; DCONT8; DCONT9; DCONT10; DCONT11; DCONT12;]
TCONT = [TCONT1; TCONT2; TCONT3; TCONT4;]
CONTROL=[CONT; DCONT; TCONT;]
```

%%%%%%%%%%%%%%%%%%%%

Step one to correct eight bit flip errors.

%%%%%%%%%%%%%%%%%%%%

```
p=1;
index = 1;
for t = 0:23;
for s = 0:23;
for r = 0:23;
```

%%

Iteration depends on which qubits we want to transform into fixed state, here we find that two controlled gates won't work, so we try to search three layers of quantum gates.

%%

```
q=1;
```

```
CONTROLV1=CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])*V4;
CONTROLV2=CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])*V5;
CONTROLV3=CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])*V6;
CONTROLV4=CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])*V7;
CONTROLV5=CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])*V10;
CONTROLV6=CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])*V11;
CONTROLV7=CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])*V12;
CONTROLV8=CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])*V14;
CONTROLV9=CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])*V15;

CONTROLV10=CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])*V16;


CONTROLV = [CONTROLV1 CONTROLV2 CONTROLV3 CONTROLV4 CONTROLV5 CONTROLV6
CONTROLV7 CONTROLV8 CONTROLV9 CONTROLV10];
for i = 1:10;
for j = 1:16;
if CONTROLV(:,i) == V(:,j);
A(p,q) = j;
q=q+1;
end;
end;
end;
for m = 9:-1:1;
for n = 9:-1:10-m;
if A(p,n+1) < A(p,n);
a= A(p,n+1);
A(p,n+1) = A(p,n);
A(p,n) = a;
end;
end;
end;
if A(p,:) == [5 6 7 8 10 12 13 14 15 16];
B(index,:) = [t s r];
index = index+1;
end;
p=p+1;
end;
```

```
end;

end;
```

%%%%%%%%%%%%%%%%%%%%%%%%%%

Result of step one to correct eightbit flip errors.

%%%%%%%%%%%%%%%%%%%%%%%%%%

15 4 22 DCONT4*CONT5*DCONT11

15 13 22 DCONT4 DCONT2 DCONT11

15 22 13 DCONT4 DCONT11 DCONT2

22 15 13 DCONT11 DCONT4 DCONT2

%%

So there are four kinds of operations that can transform eight of ten bit flip errors to normal for the choice of [5 6 7 8 13 14 15 16].

%%

%%%%%%%%%%%%%%%%%%%%%%%%%%

Step two to correct remaining two bit flip errors.

%%%%%%%%%%%%%%%%%%%%%%%%%%

```
B = [5 6 7 8 10 12 13 14 15 16];
p=1;
index=1;
for t = 0:23;
q=1;
RCONTROLV1= DCONT4*CONT5*DCONT11*V4;
RCONTROLV2= DCONT4*CONT5*DCONT11*V5;
RCONTROLV3= DCONT4*CONT5*DCONT11*V6;
RCONTROLV4= DCONT4*CONT5*DCONT11*V7;
RCONTROLV5= DCONT4*CONT5*DCONT11*V10;
RCONTROLV6= DCONT4*CONT5*DCONT11*V11;
RCONTROLV7= DCONT4*CONT5*DCONT11*V12;
RCONTROLV8= DCONT4*CONT5*DCONT11*V14;
RCONTROLV9= DCONT4*CONT5*DCONT11*V15;
```

```
RCONTROLV10=DCONT4*CONT5*DCONT11*V16;
CONTROLV1= CONTROL([(t*16+1):(t+1)*16],[1:16])*RCONTROLV1;
CONTROLV2= CONTROL([(t*16+1):(t+1)*16],[1:16])*RCONTROLV2;
CONTROLV3= CONTROL([(t*16+1):(t+1)*16],[1:16])*RCONTROLV3;
CONTROLV4= CONTROL([(t*16+1):(t+1)*16],[1:16])*RCONTROLV4;
CONTROLV5= CONTROL([(t*16+1):(t+1)*16],[1:16])*RCONTROLV5;
CONTROLV6= CONTROL([(t*16+1):(t+1)*16],[1:16])*RCONTROLV6;
CONTROLV7= CONTROL([(t*16+1):(t+1)*16],[1:16])*RCONTROLV7;
CONTROLV8= CONTROL([(t*16+1):(t+1)*16],[1:16])*RCONTROLV8;
CONTROLV9= CONTROL([(t*16+1):(t+1)*16],[1:16])*RCONTROLV9;
CONTROLV10=CONTROL([(t*16+1):(t+1)*16],[1:16])*RCONTROLV10;
CONTROLV = [CONTROLV1 CONTROLV2 CONTROLV3 CONTROLV4 CONTROLV5 CONTROLV6
CONTROLV7 CONTROLV8 CONTROLV9 CONTROLV10];
for i = 1:10;
for j = 1:16;
if CONTROLV(:,i) == V(:,j);
A(p,q) = j;
q=q+1;
end;
end;
end;
if A(p,8) < A(p,5);
a= A(p,8);
A(p,8) = A(p,5);
A(p,5) = a;
end;
if [A(p,5) A(p,8)] == [8 16];
B(index,:) = [t];
index = index+1;
else if [A(p,5) A(p,8)] == [12 16];
B(index,:) = [t];
index = index+1;
else if [A(p,5) A(p,8)] == [14 16];
B(index,:) = [t];
index = index+1;
```

```
else if [A(p,5) A(p,8)] == [15 16];
B(index,:) = [t];
index = index+1;
end;
end;
end;
end;
p=p+1;

end;
```

%%%%%%%%%%%%%%%%%%%%%%

Step three to correct four phase flip errors.

%%%%%%%%%%%%%%%%%%%%%%

```
RCONTROLV0= DCONT4*CONT5*DCONT11*V2;
RCONTROLV1= DCONT4*CONT5*DCONT11*V4;
RCONTROLV2= DCONT4*CONT5*DCONT11*V5;
RCONTROLV3= DCONT4*CONT5*DCONT11*V6;
RCONTROLV4= DCONT4*CONT5*DCONT11*V7;
RCONTROLV5= DCONT4*CONT5*DCONT11*V10;
RCONTROLV6= DCONT4*CONT5*DCONT11*V11;
RCONTROLV7= DCONT4*CONT5*DCONT11*V12;
RCONTROLV8= DCONT4*CONT5*DCONT11*V14;
RCONTROLV9= DCONT4*CONT5*DCONT11*V15;
RCONTROLV10=DCONT4*CONT5*DCONT11*V16;
DCONTROLV0= DCONT6*RCONTROLV0;
DCONTROLV1= DCONT6*RCONTROLV1;
DCONTROLV2= DCONT6*RCONTROLV2;
DCONTROLV3= DCONT6*RCONTROLV3;
DCONTROLV4= DCONT6*RCONTROLV4;
DCONTROLV5= DCONT6*RCONTROLV5;
DCONTROLV6= DCONT6*RCONTROLV6;
DCONTROLV7= DCONT6*RCONTROLV7;
DCONTROLV8= DCONT6*RCONTROLV8;
DCONTROLV9= DCONT6*RCONTROLV9;
DCONTROLV10=DCONT6*RCONTROLV10;
```

```
p=1;
index=1;
ind=1;
for t = 0:27;
q=1;
CONTROLV0= CONTROL([(t*16+1):(t+1)*16],[1:16])*DCONTROLV0;
CONTROLV1= CONTROL([(t*16+1):(t+1)*16],[1:16])*DCONTROLV1;
CONTROLV2= CONTROL([(t*16+1):(t+1)*16],[1:16])*DCONTROLV2;
CONTROLV3= CONTROL([(t*16+1):(t+1)*16],[1:16])*DCONTROLV3;
CONTROLV5= CONTROL([(t*16+1):(t+1)*16],[1:16])*DCONTROLV5;
CONTROLV6= CONTROL([(t*16+1):(t+1)*16],[1:16])*DCONTROLV6;
CONTROLV = [CONTROLV0 CONTROLV1 CONTROLV2 CONTROLV3 CONTROLV5 CONTROLV6];
for i = 1:6;
for j = 1:16;
if CONTROLV(:,i) == V(:,j);
A(p,q) = j;
q=q+1;
end;
end;
end;
for m = 5:-1:1;
for n = 5:-1:6-m;
if A(p,n+1) < A(p,n);
a= A(p,n+1);
A(p,n+1) = A(p,n);
A(p,n) = a;
end;
end;
end;
if A(p,:) == [2 6 7 8 14 16];
B(index,:) = [t];
index = index+1;
end;
if A(p,:) == [2 6 7 8 15 16];
C(ind,:) = [t];
```

```
ind = ind+1;
end;
p=p+1;

end;
```

%%%%%%%%%%%%%%%%%%%%%%%%%%

Step four to correct remaining two phase flip errors.

%%%%%%%%%%%%%%%%%%%%%%%%%%

```
RCONTROLV0= DCONT4*CONT5*DCONT11*V2;
RCONTROLV1= DCONT4*CONT5*DCONT11*V4;
RCONTROLV2= DCONT4*CONT5*DCONT11*V5;
RCONTROLV3= DCONT4*CONT5*DCONT11*V6;
RCONTROLV4= DCONT4*CONT5*DCONT11*V7;
RCONTROLV5= DCONT4*CONT5*DCONT11*V10;
RCONTROLV6= DCONT4*CONT5*DCONT11*V11;
RCONTROLV7= DCONT4*CONT5*DCONT11*V12;
RCONTROLV8= DCONT4*CONT5*DCONT11*V14;
RCONTROLV9= DCONT4*CONT5*DCONT11*V15;
RCONTROLV10=DCONT4*CONT5*DCONT11*V16;
DCONTROLV0= DCONT6*RCONTROLV0;
DCONTROLV1= DCONT6*RCONTROLV1;
DCONTROLV2= DCONT6*RCONTROLV2;
DCONTROLV3= DCONT6*RCONTROLV3;
DCONTROLV4= DCONT6*RCONTROLV4;
DCONTROLV5= DCONT6*RCONTROLV5;
DCONTROLV6= DCONT6*RCONTROLV6;
DCONTROLV7= DCONT6*RCONTROLV7;
DCONTROLV8= DCONT6*RCONTROLV8;
DCONTROLV9= DCONT6*RCONTROLV9;
DCONTROLV10=DCONT6*RCONTROLV10;
PCONTROLV0= CONT7*DCONTROLV0;
PCONTROLV1= CONT7*DCONTROLV1;
PCONTROLV2= CONT7*DCONTROLV2;
PCONTROLV3= CONT7*DCONTROLV3;
PCONTROLV5= CONT7*DCONTROLV5;
```

```
PCONTROLV6= CONT7*DCONTROLV6;
p=1;
index=1;
ind=1;
for t = 0:27;
for s = 0:27;
for r = 0:27;

q=1;
CONTROLV0= CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])
*PCONTROLV0;
CONTROLV1= CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])
*PCONTROLV1;
CONTROLV2= CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])
*PCONTROLV2;
CONTROLV3= CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])
*PCONTROLV3;
CONTROLV5= CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])
*PCONTROLV5;
CONTROLV6= CONTROL([(t*16+1):(t+1)*16],[1:16])*CONTROL([(s*16+1):(s+1)*16],[1:16])*CONTROL([(r*16+1):(r+1)*16],[1:16])

*PCONTROLV6;
CONTROLV = [CONTROLV0 CONTROLV1 CONTROLV2 CONTROLV3 CONTROLV5 CONTROLV6];
for i = 1:6;
for j = 1:16;
if CONTROLV(:,i) == V(:,j);
A(p,q) = j;
q=q+1;
end;
end;
end;
if A(p,4) < A(p,1);
a= A(p,4);
A(p,4) = A(p,1);
A(p,1) = a;
end;
if [A(p,1) A(p,4)] == [12 16];
B(index,:) = [t s r];
```

```
index = index+1;
else if [A(p,1) A(p,4)] == [14 16];
B(index,:) = [t s r];
index = index+1;
else if [A(p,1) A(p,4)] == [15 16];
B(index,:) = [t s r];
index = index+1;
end;
end;
end;
p=p+1;
end;
end;
end;
```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Result of step four to correct remaining two phase flip errors.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

B =

3 8 9

5 6 10

5 9 10

5 10 9

5 18 10

5 20 10

5 22 9

6 5 10

8 3 9

8 9 10

8 10 9

8 18 9

8 20 10

8 22 9

9 5 10

10 8 9

11 3 9

11 6 10

11 18 9

11 18 10

12 8 9

14 5 10

16 5 10

17 8 9

18 5 10

18 8 9

19 9 10

19 10 9

19 20 10

19 22 9

20 5 10

21 6 10

21 18 10

22 8 9

23 3 9

23 18 9

24 5 10

24 8 9

25 5 10

26 8 9

And the corresponding controlled gates for some of these are:

3 8 9 CONT4*CONT9*CONT10

5 6 10 CONT6*CONT7*CONT11

5 9 10 CONT6*CONT10*CONT11

6 5 10 CONT7*CONT6*CONT11

8 3 9 CONT9*CONT4*CONT10

8 9 10 CONT9*CONT10*CONT11

9 5 10 CONT10*CONT6*CONT11

10 8 9 CONT11*CONT9*CONT10

11 3 9 CONT12*CONT4*CONT10

11 6 10 CONT12*CONT7*CONT11

Appendix B

Linear rank preservers of tensor products of rank one matrices

B.1 Introduction and statement of main results

Let $n \geqslant 2$ be positive integers. Denote by M_n the set of $n \times n$ complex matrices and $\mathbb{C}^n$ the set of complex column vectors with n components. Linear preserver problems concern the study of linear maps on matrices or operators with some special properties, which has a long history. In 1897, Frobenius [13] showed that a linear operator $\phi : M_n \to M_n$ satisfies $\det(\phi(A)) = \det(A)$ for all $A \in M_n$ if and only if there are $M, N \in M_n$ with $\det(MN) = 1$ such that ϕ has the form

$$A \mapsto MAN \quad \text{or} \quad A \mapsto MA^tN.$$

Since then, lots of linear preservers have been characterized, see [11, 37] and their references. In particular, Marcus and Moyls [44] determined linear maps that send rank one matrices to rank one matrices, which have the form $A \mapsto MAN$ or $A \mapsto MA^TN$ for some nonsingular matrices M and N.

Recently, linear maps that preserve certain properties of tensor products are studied. The *tensor product* (*Kronecker product*) of two matrices $A \in M_m$ and $B \in M_n$ is defined to be $A \otimes B = [a_{ij}B]$, which is in M_{mn}. In [11], the authors determined linear maps on Hermitian matrices that leave the spectral radius of all

tensor products invariant. In [8, 10, 9, 38] the authors determine linear maps on M_{mn} that preserve Ky Fan norms, Shattern norms, numerical radius, k-numerical range, product numerical range of all matrices of the form $A \otimes B$ with $A \in M_m$ and $B \in M_n$. Notice that the set of matrices of tensor product form shares only a very small portion in M_{mn} and the sum of two tensor products is in general no longer a tensor product form. Therefore, such linear preserver problems are more challenging than the traditional problems. In some of the above mentioned papers, the authors have also extended their results to multipartite system, i.e., matrices of the form $A_1 \otimes \cdots \otimes A_k$ with $k \geqslant 2$.

In the literature, rank preserver problem is known to be one of the fundamental problems in this subject as many other preserver problems can be deduced to rank preserver problems. For example, the result Marcus and Moyls [44] on linear rank one preservers have been applied in many other preserver results. More discussion can be found in [20]. Let $n_1, \ldots, n_k$ be positive integers of at least two. In [62], Zheng, Xu and Fošner showed that a linear map $\phi : M_{n_1 \cdots n_k} \to M_{n_1 \cdots n_k}$ satisfies

$$\operatorname{rank} \phi(A_1 \otimes \cdots \otimes A_k) = \operatorname{rank} (A_1 \otimes \cdots \otimes A_k) \quad \text{for all } A_i \in M_{n_i},\ i = 1, \ldots, k \tag{B.1}$$

if and only if ϕ has the form

$$\phi(A_1 \otimes \cdots \otimes A_k) = M(\psi_1(A_1) \otimes \cdots \otimes \psi_k(A_k))N \tag{B.2}$$

where $M, N \in M_{n_1 \cdots n_k}$ are nonsingular and ψ_i, $i = 1, \ldots, k$, is either the identity map or the transpose map. Their proof was done by induction on k with some smart argument on the rank of sum of certain matrices. The same authors also considered in [60] the injective maps on the space of Hermitian matrices satisfying (B.1) for rank one matrices only. By using a structure theorem of Westwick [57], Lim [43] improved the result of Zheng et al. and showed that a linear map $\phi : M_{n_1 \cdots n_k} \to M_{n_1 \cdots n_k}$ satisfies (B.1) for rank one matrices and nonsingular matrices has the form (B.2) too.

In this paper, we characterize linear maps $\phi : M_{n_1\cdots n_k} \to M_{n_1\cdots n_k}$ satisfying (B.1) for only rank one matrices $A_1 \otimes \cdots \otimes A_k$ with $A_i \in M_{n_i}$. In this case, the structure of maps is more complicated and the maps of the form (B.2) is only one of the special cases. To state our main result, we need the following notations. Denote by

$$\mathbb{C}^m \otimes \mathbb{C}^n = \{x \otimes y : x \in \mathbb{C}^m,\ y \in \mathbb{C}^n\} \quad \text{and} \quad M_m \otimes M_n = \{A \otimes B : A \in M_m,\ B \in M_n\}.$$

Also $\mathbb{C}^{n_1} \otimes \mathbb{C}^{n_2} \otimes \cdots \otimes \mathbb{C}^{n_k}$ can be defined accordingly. For a matrix $A = [a_{ij}] \in M_n$, denote by

$$\mathrm{vec}(A) = [a_{11}\ \ a_{12}\ \cdots\ a_{1n}\ \ a_{21}\ \ a_{22}\ \cdots\ a_{2n}\ \cdots\ a_{n1}\ \ a_{n2}\ \cdots\ a_{nn}]^T \in \mathbb{C}^{n^2}.$$

In particular, if $A = xy^T$ is rank one matrix with $x, y \in \mathbb{C}^n$, then $\mathrm{vec}(xy^T) = x \otimes y$. Given a set S, a partition $\{P_1, \ldots, P_r\}$ of S is a collection of subsets of S such that $P_i \cap P_j = \varnothing$ for $i \neq j$ and $P_1 \cup \cdots \cup P_r = S$.

We are now ready to present the main result of this paper.

Theorem B.1. *Let $n_1, \ldots, n_k$ be positive integers larger than or equal to 2 and $m = \prod_{i=1}^k n_i$. Suppose $\phi : M_m \to M_m$ is a linear map. Then*

$$\mathrm{rank}\,(\phi(A_1 \otimes \cdots \otimes A_k)) = 1 \quad \textit{whenever} \quad \mathrm{rank}\,(A_1 \otimes \cdots \otimes A_k) = 1 \quad \textit{for all} \quad A_i \in M_{n_i}, \tag{B.3}$$

if and only if there is a partition $\{P_1, P_2, P_3, P_4\}$ of the set $K = \{1, \ldots, k\}$, a $m \times p_1 p_2 p_3^2$ matrix M and a $m \times p_1 p_2 p_4^2$ matrix N with $p_\ell = \prod_{i \in P_\ell} n_i$ and $p_\ell = 1$ if $P_\ell = \varnothing$, for $\ell = 1, 2, 3, 4$, satisfying

$$\mathrm{Ker}(M) \cap \left(\bigotimes_{i \in P_1 \cup P_2} \mathbb{C}^{n_i} \otimes \bigotimes_{j \in P_3} (\mathbb{C}^{n_j} \otimes \mathbb{C}^{n_j}) \right) = \{0\}$$

and

$$\mathrm{Ker}(N) \cap \left(\bigotimes_{i \in P_1 \cup P_2} \mathbb{C}^{n_i} \otimes \bigotimes_{j \in P_4} (\mathbb{C}^{n_j} \otimes \mathbb{C}^{n_j}) \right) = \{0\}$$

such that

$$\phi(A_1 \otimes \cdots \otimes A_k) = M \left(\bigotimes_{i \in P_1} A_i \otimes \bigotimes_{i \in P_2} A_i^T \otimes \bigotimes_{i \in P_3} \mathrm{vec}(A_i) \otimes \bigotimes_{i \in P_4} \mathrm{vec}^T(A_i) \right) N^T. \quad \text{(B.4)}$$

Furthermore, for any given partition $\{P_1, P_2, P_3, P_4\}$ *of* K*, there always exists some* M *and* N *that satisfy the above kernel condition, except the case* $k = 2$*,* $K = \{1, 2\}$*,* $2 \in \{n_1, n_2\}$*, and* $(P_1, P_2, P_3, P_4) = (\varnothing, \varnothing, K, \varnothing)$ *or* $(\varnothing, \varnothing, \varnothing, K)$*.*

Shortly after the authors obtained the above result, they learned via a private communication that, by using another structure result of Westwick [58, 59], Lim [41] has also obtained a characterization of linear maps between rectangular matrices over an arbitrary field that is rank one non-increasing on tensor products of matrices. In the same project, Lim also considered linear maps sending tensor products of (non)-symmetric rank one matrices to (non)-symmetric rank one matrices.

The rest of the paper is organized as follows. In Section 2, the bipartite case ($k = 2$) of the main result will be discussed and examples will be given to demonstrate the importance of the kernel condition for the matrices M and N stated in Theorem B.1. The proof of the main result and related corollaries will be presented in Section 3.

B.2 Bipartite case

In this section, we will focus on the bipartite case (when $k = 2$). Let $\{E_{11}, \ldots, E_{mm}\}$ be the standard basis of M_m. A matrix $X \in M_{mn}$ can be expressed as

$$X = \begin{bmatrix} X_{11} & \cdots & X_{1m} \\ \vdots & \ddots & \vdots \\ X_{m1} & \cdots & X_{mm} \end{bmatrix} = \sum_{1 \leqslant i,j \leqslant m} E_{ij} \otimes X_{ij} \quad \text{with} \quad X_{ij} \in M_n.$$

The partial transposes of X on the first and the second system are defined by

$$X^{PT_1} = \sum_{1 \leqslant i,j \leqslant m} E_{ji} \otimes X_{ij} \quad \text{and} \quad X^{PT_2} = \sum_{1 \leqslant i,j \leqslant m} E_{ij} \otimes X_{ij}^T.$$

Also denote by

$$X^{R_1} = \sum_{1 \leqslant i,j \leqslant m} \mathrm{vec}(E_{ij}) \otimes X_{ij} \quad \text{and} \quad X^{R_2} = \sum_{1 \leqslant i,j \leqslant m} E_{ij} \otimes \mathrm{vec}(X_{ij}).$$

Furthermore, define the $m^2 \times n^2$ realigned matrix of X by

$$X^R = \sum_{1 \leqslant i,j \leqslant m} \mathrm{vec}(E_{ij}) \otimes \mathrm{vec}^T(X_{ij}).$$

In particular, $X^{PT_1} = X_1^T \otimes X_2$, $X^{PT_2} = X_1 \otimes X_2^T$, $X^{R_1} = \mathrm{vec}(X_1) \otimes X_2$, $X^{R_2} = X_1 \otimes \mathrm{vec}(X_2)$, and $X^R = \mathrm{vec}(X_1) \otimes \mathrm{vec}^T(X_2)$ if $X = X_1 \otimes X_2$.

Finally, for any two linear maps ψ_1 and ψ_2 on matrix spaces, we say that these two maps are permutationally similar if there are permutation matrices P and Q such that $\psi_2(A) = P\psi_1(A)Q$ for all A. For example, it is clear that $A \mapsto \mathrm{vec}(A)$ and $A \mapsto \mathrm{vec}(A^T)$ are permutationally similar.

Proposition B.1. *Let n_1, n_2 be positive integers and $m = n_1 n_2$. Given $\psi_P : M_m \to M_m$ defined by $\psi_P(A) = A^{PT_j}$ with $j = 1, 2$. The composite map $\psi_R \circ \psi_P$ is permutationally similar to the map ψ_R , when ψ_R is one of the following maps.*

$$\textit{(i)}\ A \mapsto A^{R_j}, \quad \textit{(ii)}\ A \mapsto A^R, \quad \textit{or} \quad \textit{(iii)}\ A \mapsto \mathrm{vec}(A).$$

Proof. For $j = 1, 2$, it is obvious that there is a permutation matrix $P_j \in M_{n_j}$ such that $\mathrm{vec}(X_j^T) = P_j \,\mathrm{vec}(X_i)$ for all $X_j \in M_{n_j}$. Also there is a permutation matrix $P_{12} \in M_m$ such that $\mathrm{vec}(X_1 \otimes X_2) = P_{12}\,(\mathrm{vec}(X_1) \otimes \mathrm{vec}(X_2))$ for all $X_i \in M_{n_i}$, $i = 1, 2$. We now consider the case when $j = 1$. The case $j = 2$ can be proved in a similar way.

First suppose $\psi_R : A \mapsto A^{R_1}$. For any $X_i \in M_{n_i}$, $i = 1, 2$,

$$\psi_R \circ \psi_P(X_1 \otimes X_2) = \left((X_1 \otimes X_2)^{PT_1}\right)^{R_1} = (X_1^T \otimes X_2)^{R_1} = \mathrm{vec}(X_1^T) \otimes X_2$$
$$= (P_1 \otimes I_{n_2})(\mathrm{vec}(X) \otimes X_2) = (P_1 \otimes I_{n_2})(X_1 \otimes X_2)^{R_1} = (P_1 \otimes I_{n_2})\psi_R(X_1 \otimes X_2).$$

By linearity of the two maps, we conclude that $\psi_R \circ \psi_P(A) = (P_1 \otimes I_{n_2})\psi_R(A)$ for all $A \in M_m$.

Suppose now $\psi_R : A \mapsto A^R$. For any $X_i \in M_{n_i}$, $i = 1, 2$,

$$\psi_R \circ \psi_P(X_1 \otimes X_2) = \left((X_1 \otimes X_2)^{PT_1}\right)^{R} = (X_1^T \otimes X_2)^{R} = \mathrm{vec}(X_1^T) \otimes \mathrm{vec}^T(X_2)$$
$$= P_1(\mathrm{vec}(X) \otimes \mathrm{vec}^T(X_2)) = P_1(X_1 \otimes X_2)^R = P_1\psi_R(X_1 \otimes X_2).$$

Thus, the same conclusion holds. Finally assume $\psi_R : A \mapsto \mathrm{vec}(A)$. For any $X_i \in M_{n_i}$, $i = 1, 2$,

$$\begin{aligned}
\psi_R \circ \psi_P(X_1 \otimes X_2) = \mathrm{vec}(X_1^T \otimes X_2) &= P_{12}\left(\mathrm{vec}(X_1^T) \otimes \mathrm{vec}(X_2)\right) \\
&= P_{12}(P_1 \otimes I_{n_2})\left(\mathrm{vec}(X_1) \otimes \mathrm{vec}(X_2)\right) \\
&= P_{12}(P_1 \otimes I_{n_2})P_{12}^T \mathrm{vec}(X_1 \otimes X_2) \\
&= P_{12}(P_1 \otimes I_{n_2})P_{12}^T \psi_R(X_1 \otimes X_2).
\end{aligned}$$

Again by linearity of the maps, we conclude that $\psi_R \circ \psi_P(A) = P_{12}(P_1 \otimes I_{n_2})P_{12}^T \psi_R(A)$ for all $A \in M_m$. □

It turns out that for the bipartite case ($k = 2$), Theorem B.1 can be expressed in terms of partial transpose and realigned matrix as follows.

Theorem B.2. *Let n_1, n_2 be positive integers larger or equal to two and $m = n_1 n_2$. Suppose $\phi : M_m \to M_m$ is a linear map. Then*

$$\mathrm{rank}\,(\phi(A_1 \otimes A_2)) = 1 \quad \textit{whenever} \quad \mathrm{rank}\,(A_1 \otimes A_2) = 1 \quad \textit{for all } A_i \in M_{n_i},\ i = 1, 2, \tag{B.5}$$

if and only if $\phi = \psi_T \circ \psi_M \circ \psi_R \circ \psi_P$, where

(i) $\psi_P : A \mapsto A$, $A \mapsto A^{PT_1}$ *or* $A \mapsto A^{PT_2}$;

(ii) $\psi_R : A \mapsto A$, $A \mapsto A^{R_1}$, $A \mapsto A^{R_2}$, $A \mapsto A^R$ *or* $A \mapsto \mathrm{vec}(A)$;

(iii) $\psi_M : A \mapsto MAN^T$;

(iv) $\psi_T : A \mapsto A$ *or* $A \mapsto A^T$,

which has totally 16 *different forms, and* M *and* N *are matrices of appropriate size satisfying*

(1) $\mathrm{Ker}(M) \cap (\mathbb{C}^{n_1} \otimes \mathbb{C}^{n_2}) = \{0\}$ *and* $\mathrm{Ker}(N) \cap (\mathbb{C}^{n_1} \otimes \mathbb{C}^{n_2}) = \{0\}$ *if* ψ_R *is the map* $A \mapsto A$;

(2) $\mathrm{Ker}(M) \cap (\mathbb{C}^{n_1} \otimes \mathbb{C}^{n_1}) = \{0\}$ *and* $\mathrm{Ker}(N) \cap (\mathbb{C}^{n_2} \otimes \mathbb{C}^{n_2}) = \{0\}$ *if* ψ_R *is the map* $A \mapsto A^R$;

(3) $\mathrm{Ker}(M) \cap (\mathbb{C}^{n_1} \otimes \mathbb{C}^{n_1} \otimes \mathbb{C}^{n_2}) = \{0\}$ *and* N *has full column rank equal to* n_2 *if* ψ_R *is the map* $A \mapsto A^{R_1}$;

(4) $\mathrm{Ker}(M) \cap (\mathbb{C}^{n_1} \otimes \mathbb{C}^{n_2} \otimes \mathbb{C}^{n_2}) = \{0\}$ *and* N *has full column rank equal to* n_1 *if* ψ_R *is the map* $A \mapsto A^{R_2}$;

(5) $\mathrm{Ker}(M) \cap (\mathbb{C}^{n_1} \otimes \mathbb{C}^{n_1} \otimes \mathbb{C}^{n_2} \otimes \mathbb{C}^{n_2}) = \{0\}$ *and* N *is a* $m \times 1$ *nonzero matrix if* $2 \notin \{n_1, n_2\}$ *and* ψ_R *is the map* $A \mapsto \mathrm{vec}(A)$.

Proof. It is easy to verify that the two maps

$$X_1 \otimes X_2 \mapsto X_1 \otimes X_2 \quad \text{and} \quad X_1 \otimes X_2 \mapsto X_2 \otimes X_1$$

are premuationally similar. Applying Theorem B.1 with $k = 2$ and taking the above observation into account, the equation (B.4) can be reduced to the following 16 cases.

1) $\{P_1, P_2, P_3, P_4\} = \{\{1\}, \{2\}, \varnothing, \varnothing\}$ and
$\phi(A_1 \otimes A_2) = M(A_1 \otimes A_2^T)N^T = M(A_1 \otimes A_2)^{PT_2}N^T$.

2) $\{P_1, P_2, P_3, P_4\} = \{\{2\}, \{1\}, \varnothing, \varnothing\}$ and
$\phi(A_1 \otimes A_2) = M(A_1^T \otimes A_2)N^T = M(A_1 \otimes A_2)^{PT_1}N^T$.

3) $\{P_1, P_2, P_3, P_4\} = \{\{1\}, \varnothing, \{2\}, \varnothing\}$ and
$\phi(A_1 \otimes A_2) = M(A_1 \otimes \mathrm{vec}(A_2))N^T = M(A_1 \otimes A_2)^{R_2}N^T$.

4) $\{P_1, P_2, P_3, P_4\} = \{\{2\}, \varnothing, \{1\}, \varnothing\}$ and
$\phi(A_1 \otimes A_2) = M(\mathrm{vec}(A_1) \otimes A_2)N^T = M(A_1 \otimes A_2)^{R_1}N^T$.

5) $\{P_1, P_2, P_3, P_4\} = \{\{1\}, \varnothing, \varnothing, \{2\}\}$ and
$\phi(A_1 \otimes A_2) = M(A_1 \otimes \mathrm{vec}^T(A_2))N^T = \left(N((A_1 \otimes A_2)^{PT_1})^{R_2}M^T\right)^T$.

6) $\{P_1, P_2, P_3, P_4\} = \{\{2\}, \varnothing, \varnothing, \{1\}\}$ and
$\phi(A_1 \otimes A_2) = M(\mathrm{vec}^T(A_1) \otimes A_2)N^T = \left(N((A_1 \otimes A_2)^{PT_2})^{R_1}M^T\right)^T$.

7) $\{P_1, P_2, P_3, P_4\} = \{\varnothing, \{1\}, \{2\}, \varnothing\}$ and
$\phi(A_1 \otimes A_2) = M(A_1^T \otimes \mathrm{vec}(A_2))N^T = M((A_1 \otimes A_2)^{PT_1})^{R_2}N^T$.

8) $\{P_1, P_2, P_3, P_4\} = \{\varnothing, \{2\}, \{1\}, \varnothing\}$ and
$\phi(A_1 \otimes A_2) = M(\mathrm{vec}(A_1) \otimes A_2^T))N^T = M((A_1 \otimes A_2)^{PT_2})^{R_1}N^T$.

9) $\{P_1, P_2, P_3, P_4\} = \{\varnothing, \{1\}, \varnothing, \{2\}\}$ and
$\phi(A_1 \otimes A_2) = M(A_1^T \otimes \mathrm{vec}^T(A_2))N^T = \left(N(A_1 \otimes A_2)^{R_2}M^T\right)^T$.

10) $\{P_1, P_2, P_3, P_4\} = \{\varnothing, \{2\}, \varnothing, \{1\}\}$ and
$\phi(A_1 \otimes A_2) = M(\mathrm{vec}^T(A_1) \otimes A_2^T))N^T = \left(N(A_1 \otimes A_2)^{R_1}M^T\right)^T$.

11) $\{P_1, P_2, P_3, P_4\} = \{\varnothing, \varnothing, \{1\}, \{2\}\}$ and
$\phi(A_1 \otimes A_2) = M(\mathrm{vec}(A_1) \otimes \mathrm{vec}^T(A_2))N^T = M(A_1 \otimes A_2)^R N^T$.

12) $\{P_1, P_2, P_3, P_4\} = \{\varnothing, \varnothing, \{2\}, \{1\}\}$ and
$\phi(A_1 \otimes A_2) = M(\mathrm{vec}^T(A_1) \otimes \mathrm{vec}(A_2))N^T = \left(N(A_1 \otimes A_2)^R M^T\right)^T$.

13) $\{P_1, P_2, P_3, P_4\} = \{\{1,2\}, \varnothing, \varnothing, \varnothing\}$ and
$\phi(A_1 \otimes A_2) = M(A_1 \otimes A_2)N^T$.

14) $\{P_1, P_2, P_3, P_4\} = \{\varnothing, \{1,2\}, \varnothing, \varnothing\}$ and
$\phi(A_1 \otimes A_2) = M(A_1^T \otimes A_2^T)N^T = \left(N(A_1 \otimes A_2)M^T\right)^T$.

15) $\{P_1, P_2, P_3, P_4\} = \{\varnothing, \varnothing, \{1,2\}, \varnothing\}$ and
$\phi(A_1 \otimes A_2) = M\left(\mathrm{vec}(A_1) \otimes \mathrm{vec}(A_2)\right)N^T = MP_{12}^T\left(\mathrm{vec}(A_1 \otimes A_2)\right)N^T$.

16) $\{P_1, P_2, P_3, P_4\} = \{\varnothing, \varnothing, \varnothing, \{1,2\}\}$ and
$\phi(A_1 \otimes A_2) = M\left(\mathrm{vec}^T(A_1) \otimes \mathrm{vec}^T(A_2)\right)N^T = \left(NP_{12}^T(\mathrm{vec}\,(A_1 \otimes A_2))\,M^T\right)^T$.

Here, M and N are matrices with appropriate size, and satisfy the kernel condition in Theorem B.1 (In some cases, the roles of M and N may interchange). Also the cases 15) and 16) hold only when $2 \notin \{n_1, n_2\}$. In all these cases, the map ϕ can be represented by $A \mapsto \psi_T \circ \psi_M \circ \psi_R \circ \psi_P(A)$ where ψ_P, ψ_M, ψ_R, ψ_T are of the forms in (i), (ii), (iii) and (iv) respectively. Furthermore, by Proposition B.1, if ψ_P is a partial transport map with respect to the jth subsystem, $\psi_R \circ \psi_P$ is permutationally similar to ψ_R, when ψ_R has the form $A \mapsto A^{R_j}$, $A \mapsto A^R$ or $A \mapsto \mathrm{vec}(A)$. Therefore, instead of 15 different types, there are actually only 9 different types of compositions of $\psi_R \circ \psi_P$. Finally, since $(A^{PT_1})^T = A^{PT_2}$ and $(A^{PT_2})^T = A^{PT_1}$, the maps $A \mapsto (MA^{PT_1}N^T)^T$ and $A \mapsto (MA^{PT_2}N^T)^T$ are the same as $A \mapsto N^T A^{PT_2} M$ and $A \mapsto N^T A^{PT_1} M$, respectively. Therefore, the map $\psi_T \circ \psi_M \circ \psi_R \circ \psi_P$ has totally 16 different forms only. □

In the following, we give some low dimensional examples of M and N that satisfy the conditions (2), (3) and (5) of Theorem B.2.

Example B.1. Assume $(n_1, n_2) = (2, 3)$ and define the 6×4 matrix M and the 6×9

matrix N by

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad N = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Clearly, $\text{rank}\,(M) = 4$ and $\text{rank}\,(N) = 5$. Also

$$\text{Ker}(M) = \{0\} \quad \text{and} \quad \text{Ker}(N) = \left\{ \begin{bmatrix} a & b & c & d & a & b & c & d & 0 \end{bmatrix}^T : a, b, c, d \in \mathbb{C} \right\}.$$

Therefore, $\text{Ker}(N)$ does not contain any nonzero element in $\mathbb{C}^3 \otimes \mathbb{C}^3$. Then the map $A \mapsto MA^R N^T$ satisfies the condition (B.5) and its range space contains matrices of rank at most 4 only.

Example B.2. Assume $(n_1, n_2) = (2, 3)$ and define the 6×12 matrix M and the 6×3 matrix N by

$$M = \begin{bmatrix} I_6 & \hat{M} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad N = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Clearly, $\text{Ker}(N) = \{0\}$. Suppose $M(x \otimes y \otimes z) = 0$ for some nonzero $x, y \in \mathbb{C}^2$ and $z \in \mathbb{C}^3$. Then

$$0 = M(x \otimes y \otimes z) = M(x \otimes I_6)(y \otimes z) = (x_1 I_6 + x_2 \hat{M})(y \otimes z),$$

where $x = [x_1 \; x_2]^T$. So $(x_1 I_6 + x_2 \hat{M})$ is singular and hence $x_1 = 0$ as $\det(x_1 I_6 + x_2 \hat{M}) = x_1^6$. Thus, the vector $y \otimes z$ is in the kernel of $\hat{M}$. However, $\text{Ker}(\hat{M}) = \left\{ [a \; 0 \; 0 \; 0 \; a \; 0]^T : a \in \mathbb{C} \right\}$, which does not contain any nonzero element of $\mathbb{C}^2 \otimes \mathbb{C}^3$.

Therefore, even $\mathrm{Ker}(M)$ is a 6 dimensional subspace of $\mathbb{C}^{12}$, $\mathrm{Ker}(M)$ does not contain any nonzero element of $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^3$.

Example B.3. Assume $(n_1, n_2) = (3, 3)$ and define the 9×81 matrix M by

$$M = \begin{bmatrix} I_9 & R & R^2 & R^3 & -I_9 & -R & -R^2 & -R^3 & R^4 \end{bmatrix}$$

with

$$R = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Notice that

$$\begin{aligned} \mathrm{Ker}(R) \subseteq \mathrm{Ker}(R^2) &\subseteq \mathrm{Ker}(R^3) \\ &\subseteq \mathrm{Ker}(R^4) = \left\{ \begin{bmatrix} a & b & c & d & a & b & c & d & 0 \end{bmatrix}^T : a, b, c, d \in \mathbb{C} \right\}. \end{aligned}$$

Suppose $M(x \otimes y \otimes z \otimes w) = 0$ for some nonzero $x, y, z, w \in \mathbb{C}^3$. Set $x \otimes y = \begin{bmatrix} u_1 & \cdots & u_9 \end{bmatrix}^T \in \mathbb{C}^9$ and define

$$U = M(x \otimes y \otimes I_9) = (u_1 - u_5)I_9 + (u_2 - u_6)R + (u_3 - u_7)R^2 + (u_4 - u_8)R^3 + u_9 R^4.$$

Then

$$0 = M(x \otimes y \otimes z \otimes w) = M(x \otimes y \otimes I_9)(z \otimes w) = U(z \otimes w).$$

Now let

$$U_5 = u_9 I_9 \quad \text{and} \quad U_k = (u_k - u_{k+4})I_9 + U_{k+1}R \quad \text{for } k = 1, 2, 3, 4.$$

Then it can be verified that

$$U_1 = (u_1 - u_5)I_9 + ((u_2 - u_6)I_9 + ((u_3 - u_7)I_9 + ((u_4 - u_8)I_9 + (u_9 I_9)\, R)\, R)\, R)\, R = U.$$

For $k = 1, 2, 3, 4$, because R is singular, U_k is singular if and only if $u_k - u_{k+4} = 0$, or equivalently, $U_k = U_{k+1}R$. Furthermore, when U_k is singular,

$$\mathrm{Ker}(U_k R^{k-1}) = \mathrm{Ker}(U_{k+1} R R^{k-1}) = \mathrm{Ker}(U_{k+1} R^k).$$

Suppose at least one of $U_1, \ldots, U_5$ is nonsingular, say U_ℓ is nonsingular for some $1 \leqslant \ell \leqslant 5$ and $U_1, \cdots U_{\ell-1}$ are all singular. Then

$$\mathrm{Ker}(U) = \mathrm{Ker}(U_1) \subseteq \mathrm{Ker}(U_2 R) \subseteq \cdots \subseteq \mathrm{Ker}(U_\ell R^{\ell-1}) = \mathrm{Ker}(R^{\ell-1}) \subseteq \mathrm{Ker}(R^4).$$

But this is impossible since $U(w \otimes z) = 0$ while $\mathrm{Ker}(R^4)$ does not contain any nonzero element of $\mathbb{C}^3 \otimes \mathbb{C}^3$. Therefore, all $U_1, \ldots, U_5$ are singular. In this case, we have $u_k - u_{k+4} = 0$ for $k = 1, 2, 3, 4$ and $u_9 = 0$, or equivalently, $x \otimes y$ has the form $\begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_1 & u_2 & u_3 & u_4 & 0 \end{bmatrix}^T$, and contradiction again arrived. Thus, one can conclude that $\mathrm{Ker}(M)$ does not contain any nonzero element of $\mathbb{C}^3 \otimes \mathbb{C}^3 \otimes \mathbb{C}^3 \otimes \mathbb{C}^3$. Now take any 9×1 nonzero matrix N. Then the composition map $\phi : A \mapsto M\mathrm{vec}(A)N^T$ satisfies condition (B.5). In this case, $\mathrm{rank}\,(\phi(A)) \leqslant 1$ for all $A \in M_9$.

Remark B.1. For condition (1) of Theorem B.2, both M and N have size $m \times m$. In this case, any nonsingular matrices $M, N \in M_m$ satisfy case (1). But there exists singular matrices that satisfy the condition (1) too. For example, when $(n_1, n_2) = (2, 2)$ one can construct a rank three 4×4 matrix M with $\mathrm{Ker}(M) = \left\{ \begin{bmatrix} a & 0 & 0 & a \end{bmatrix}^T : a \in \mathbb{C} \right\}$, which does not contain any nonzero vector in $\mathbb{C}^2 \otimes \mathbb{C}^2$.

For condition (2) of Theorem B.2, the same observation as above follows if $n_1 = n_2$. If $n_1 < n_2$, M can be chosen to be any $m \times n_1^2$ matrix with full column rank, i.e., $\mathrm{rank}\,(M) = n_1^2$. Similarly, N can be chosen to be any $m \times n_2^2$ matrix with full column rank if $n_1 > n_2$.

Finally, it has to point out that the partial transpose and realignment are two useful concept in the study of separable problem, which is one of the most important

problem in quantum information science. Although it have been showed that the general characterization of separable states is NP-hard [19], researchers are interested in finding effective criterion to determine separability of a quantum state. A quantum state (density matrix) X is PPT (positive partial transpose) if X^{PT_1} (or equivalently X^{PT_2}) is positive semi-definite. One of the classical and popular criteria is PPT criterion introduced by Peres [48]. The PPT criterion states that if X is separable, then X is PPT and these two conditions are equivalent if $m = n_1 n_2 \leqslant 6$ [21]. Another strong criterion is CCNR criterion [6, 50], which confirmed that $\|X^R\|_1 \leqslant 1$ if X is separable. It has to note that researchers also studied preservers on separable states, see [2, 12, 23]. In particular, the authors in [12] studied linear maps that send the set of separable states onto itself in multipartite system.

B.3 Proof of the main results

In this section, we will present the proof of Theorem B.1. The proof relies on the structure result of Westwick [57, Theorem 3.4] on preservers of nonzero decomposable tensors, and we restate this result as follows.

Theorem B.3. *Let $U_1, \ldots, U_p$ and $W_1, \ldots, W_q$ be finite dimensional vector spaces over a field F with $\dim(U_i) \geqslant 2$ and define $U = \bigotimes_{i=1}^{p} U_i$ and $W = \bigotimes_{j=1}^{q} W_j$. Suppose $f : U \to W$ is a linear map sending nonzero decomposable tensors into nonzero decomposable tensors. Then there is a partition $\{S_1, \ldots, S_q\}$ of $\{1, \ldots, p\}$ (S_j can be an empty set) and linear functions $f_j : \bigotimes_{i \in S_j} U_i \to W_j$ sending nonzero decomposable tensors to nonzero vectors, such that*

$$f(x_1 \otimes \cdots \otimes x_p) = \bigotimes_{j=1}^{q} f_j \left(\otimes_{i \in S_j} x_i \right).$$

Here, f_j is defined to be a nonzero constant function, i.e., $f_j(\cdot) = w_j$ for some nonzero $w_j \in W_j$, if $S_j = \varnothing$.

We will prove the following equivalent version of Theorem B.1.

Theorem B.4. *Let $n_1, \ldots, n_k$ be positive integers larger than or equal to 2 and let $m = \prod_{i=1}^{k} n_i$. Suppose $\phi : M_m \to M_m$ is a linear map. Then*

$$\operatorname{rank}(\phi(A_1 \otimes \cdots \otimes A_k)) = 1 \quad \textit{whenever} \quad \operatorname{rank}(A_1 \otimes \cdots \otimes A_k) = 1 \quad \textit{for all } A_i \in M_{n_i} \tag{B.6}$$

if and only if there are two subsets K_1, K_2 of $K = \{1, \ldots, k\}$, a $m \times m_1 m_2$ matrix M and a $m \times m^2/(m_1 m_2)$ matrix N with $m_t = \prod_{i \in K_t} n_i$ or $m_t = 1$ if $K_t = \varnothing$, $t = 1, 2$, satisfying

$$\operatorname{Ker}(M) \cap \left(\bigotimes_{i \in K_1} \mathbb{C}^{n_i} \otimes \bigotimes_{j \in K_2} \mathbb{C}^{n_j} \right) = \{0\} \quad \textit{and} \quad \operatorname{Ker}(N) \cap \left(\bigotimes_{i \notin K_1} \mathbb{C}^{n_i} \otimes \bigotimes_{j \notin K_2} \mathbb{C}^{n_j} \right) = \{0\} \tag{B.7}$$

such that

$$\phi\left(x_1 y_1^T \otimes \cdots \otimes x_k y_k^T\right) = M \left(\bigotimes_{i \in K_1} x_i \otimes \bigotimes_{j \in K_2} y_j \right) \left(\bigotimes_{i \notin K_1} x_i \otimes \bigotimes_{j \notin K_2} y_j \right)^T N^T \tag{B.8}$$

for all $x_i, y_i \in \mathbb{C}^{n_i}$. Furthermore, for any given subsets K_1, K_2 of K, there always exists some M and N that satisfy the above kernel condition, except the case $k = 2$, $K = \{1, 2\}$, $2 \in \{n_1, n_2\}$, and either $K_1 = K_2 = K$ or $K_1 = K_2 = \varnothing$.

Proof. The necessary part is clear. For the sufficient part, define a linear map $f : \mathbb{C}^{m^2} \to \mathbb{C}^{m^2}$ such that

$$f\left(\bigotimes_{i=1}^{k} (x_i \otimes y_i) \right) = \operatorname{vec}\left(\phi\left(\bigotimes_{i=1}^{k} x_i y_i^T \right) \right) \quad \text{for all} \quad x_i, y_i \in \mathbb{C}^{n_i},$$

and by linearity, extend the definition of f to all vectors in $\mathbb{C}^{m^2}$. Recall that $\operatorname{vec}(A) = x \otimes y$ if $A = xy^T$ is rank one. As ϕ satisfies (B.6), the map f will send all nonzero vectors of the form $\bigotimes_{i=1}^{k} (x_i \otimes y_i)$ to some nonzero vectors of the form $u \otimes v \in$

$\mathbb{C}^m \otimes \mathbb{C}^m$, i.e., f sends nonzero decomposable elements of $\bigotimes_{i=1}^{k} \mathbb{C}^{n_i} \otimes \mathbb{C}^{n_i}$ to nonzero decomposable elements of $\mathbb{C}^m \otimes \mathbb{C}^m$. Applying Proposition B.3 ([57, Theorem 3.4]) with $p = 2k$ and $q = 2$, there are two partitions $\{K_1, \overline{K_1}\}$ and $\{K_2, \overline{K_2}\}$ of $K = \{1, \ldots, k\}$, and linear maps $f_1 : \mathbb{C}^{m_1 m_2} \to \mathbb{C}^m$ and $f_2 : \mathbb{C}^{m^2/(m_1 m_2)} \to \mathbb{C}^m$, where m_t is defined as in statement of the theorem, such that

$$f\left(\bigotimes_{i=1}^{k}(x_i \otimes y_i)\right) = f_1\left(\bigotimes_{i\in K_1} x_i \otimes \bigotimes_{j\in K_2} y_j\right) \otimes f_2\left(\bigotimes_{i\in \overline{K_1}} x_i \otimes \bigotimes_{j\in \overline{K_2}} y_j\right).$$

As f_1 and f_2 are linear, there exist a $m \times m_1 m_2$ matrix M and a $m \times m^2/(m_1 m_2)$ matrix N such that $f_1(z) = Mz$ and $f_2(w) = Nw$. Thus, ϕ has the form as described in (B.8). Further, $f_1(z) \neq 0$ for all $z \in \bigotimes_{i\in K_1} \mathbb{C}^{n_i} \otimes \bigotimes_{j\in K_2} \mathbb{C}^{n_j}$ and $f_2(w) \neq 0$ for all $w \in \bigotimes_{i\notin K_1} \mathbb{C}^{n_i} \otimes \bigotimes_{j\notin K_2} \mathbb{C}^{n_j}$ as $\overline{K_j} = K\backslash K_j$, and hence, M and N satisfy the condition (B.7). The last statement will be confirmed by Proposition B.3. □

Now the equivalence of Theorems B.1 and B.4 can be seen as follows.

Proof of Theorem B.1. Suppose ϕ satisfies the rank condition (B.3). Then Theorem B.4 implies that ϕ has the form (B.8) with M and N satisfying (B.7). Set $P_1 = K_1\backslash K_2$, $P_2 = K_2\backslash K_1$, $P_3 = K_1 \cap K_2$, and $P_4 = K\backslash(K_1 \cup K_2)$. First, there exists a permutation matrix Q_x such that for any $x_i, y_i \in \mathbb{C}^{n_i}$,

$$\begin{aligned} Q_x\left(\bigotimes_{i\in P_1} x_i \otimes \bigotimes_{j\in P_2} y_j \otimes \bigotimes_{k\in P_3}(x_k \otimes y_k)\right) &= \left(\bigotimes_{i\in P_1} x_i \otimes \bigotimes_{i\in P_3} x_i \otimes \bigotimes_{j\in P_2} y_j \otimes \bigotimes_{j\in P_3} y_j\right) \\ &= \left(\bigotimes_{i\in K_1} x_i \otimes \bigotimes_{j\in K_2} y_j\right). \end{aligned}$$

Similarly, there exists another permutation matrix Q_y such that for any $x_i, y_i \in \mathbb{C}^{n_i}$,

$$\begin{aligned} Q_y \left(\bigotimes_{j\in P_1} y_j \otimes \bigotimes_{i\in P_2} x_i \otimes \bigotimes_{k\in P_4} (x_k \otimes y_k) \right) &= \left(\bigotimes_{i\in P_2} x_i \otimes \bigotimes_{i\in P_4} x_i \otimes \bigotimes_{j\in P_1} y_j \otimes \bigotimes_{j\in P_4} y_j \right) \\ &= \left(\bigotimes_{i\notin K_1} x_i \otimes \bigotimes_{j\notin K_2} y_j \right). \end{aligned}$$

Now for any rank one matrix $A_i = x_i y_i^T$ with $x_i, y_i \in \mathbb{C}^{n_i}$, $i = 1, \ldots, k$,

$$\begin{aligned} & \phi\left(A_1 \otimes \cdots \otimes A_k\right) = \phi\left(x_1 y_1^T \otimes \cdots \otimes x_k y_k^T\right) \\ = \;& M \left(\bigotimes_{i\in K_1} x_i \otimes \bigotimes_{j\in K_2} y_j \right) \left(\bigotimes_{i\notin K_1} x_i \otimes \bigotimes_{j\notin K_2} y_j \right)^T N^T \\ = \;& MQ_x \left(\bigotimes_{i\in P_1} x_i \otimes \bigotimes_{j\in P_2} y_j \otimes \bigotimes_{k\in P_3} (x_k \otimes y_k) \right) \left(\bigotimes_{j\in P_1} y_j \otimes \bigotimes_{i\in P_2} x_i \otimes \bigotimes_{k\in P_4} (x_k \otimes y_k) \right)^T Q_y^T N^T \\ = \;& MQ_x \left(\bigotimes_{i\in P_1} x_i \otimes \bigotimes_{j\in P_2} y_j \otimes \bigotimes_{k\in P_3} (x_k \otimes y_k) \right) \left(\bigotimes_{j\in P_1} y_j^T \otimes \bigotimes_{i\in P_2} x_i^T \otimes \bigotimes_{k\in P_4} (x_k \otimes y_k)^T \right) Q_y^T N^T \\ = \;& MQ_x \left(\left(\bigotimes_{i\in P_1} x_i \right) \left(\bigotimes_{i\in P_1} y_i^T \right) \otimes \left(\bigotimes_{j\in P_2} y_j \right) \left(\bigotimes_{j\in P_2} x_j^T \right) \otimes \bigotimes_{k\in P_3} (x_k \otimes y_k) \otimes \bigotimes_{k\in P_4} (x_k \otimes y_k)^T \right) Q_y^T N^T \\ = \;& MQ_x \left(\bigotimes_{i\in P_1} x_i y_i^T \otimes \bigotimes_{j\in P_2} y_j x_j^T \otimes \bigotimes_{k\in P_3} (x_k \otimes y_k) \otimes \bigotimes_{k\in P_4} (x_k \otimes y_k)^T \right) Q_y^T N^T \\ = \;& MQ_x \left(\bigotimes_{i\in P_1} x_i y_i^T \otimes \bigotimes_{j\in P_2} (x_j y_j^T)^T \otimes \bigotimes_{k\in P_3} \mathrm{vec}(x_k y_k^T) \otimes \bigotimes_{k\in P_4} \mathrm{vec}^T(x_k y_k^T) \right) Q_y^T N^T \\ = \;& MQ_x \left(\bigotimes_{i\in P_1} A_i \otimes \bigotimes_{j\in P_2} (A_j)^T \otimes \bigotimes_{k\in P_3} \mathrm{vec}(A_k) \otimes \bigotimes_{k\in P_4} \mathrm{vec}^T(A_k) \right) Q_y^T N^T. \end{aligned}$$

By linearity, the equality holds for any matrix $A_i \in M_{n_i}$ and hence we have (B.4). Finally, the kernel condition can be easily reduced from (B.7). □

Next we show that the matrices M and N in Theorem B.4 (equivalently, Theorem

B.1) always exist, except for two special cases, namely, when $k = 2$, $K = \{1, 2\}$, $2 \in \{n_1, n_2\}$, and $K_1 = K_2 = K$ or $K_1 = K_2 = \varnothing$. For simplicity, we focus on the existence of M. For positive integers $p_1, \ldots, p_r$, denote by $\mathcal{E}(p_1, \ldots, p_r)$ the collection of subspaces $\mathcal{S}$ of $\mathbb{C}^{p_1 \cdots p_r}$ such that

$$\mathcal{S} \cap (\mathbb{C}^{p_1} \otimes \cdots \otimes \mathbb{C}^{p_r}) = \{0\}.$$

The subspace $\mathcal{S}$ is called a completely entangled subspace in [46]. In the same paper, the author also obtained the maximum dimension of $\mathcal{S}$ in $\mathcal{E}(p_1, \ldots, p_r)$ as follows.

Proposition B.2. [46, Theorem 1.5] *Let $p_1, \ldots, p_r$ be positive integers. Then*

$$\max_{\mathcal{S} \in \mathcal{E}(p_1, \ldots, p_r)} \dim \mathcal{S} = \prod_{i=1}^{r} p_i - \sum_{i=1}^{r} p_i + r - 1.$$

It has to mention that an explicit construction for maximum completely entangled subspace for bipartite case ($r = 2$) was also given in [46]. Based on the above proposition, we can deduce the following result which showed that the matrix M always exists, except for one special case.

Proposition B.3. *Let $n_1, \ldots, n_k$ be positive integers larger than or equal to 2, $K = \{1, \ldots, k\}$, and $K_1, K_2 \subseteq K$. Define $m = \prod_{i \in K} n_i$ and $m_t = \prod_{i \in K_t} n_i$ for $t = 1, 2$. Then there always exists a $m \times m_1 m_2$ matrix M such that*

$$\mathrm{Ker}(M) \cap \left(\bigotimes_{i \in K_1} \mathbb{C}^{n_i} \otimes \bigotimes_{j \in K_2} \mathbb{C}^{n_j} \right) = \{0\},$$

except the case when $K_1 = K_2 = K = \{1, 2\}$ and $2 \in \{n_1, n_2\}$.

Proof. If $m \geqslant m_1 m_2$, then any $m \times m_1 m_2$ matrix with full column rank, i.e., $\mathrm{rank}\,(M) = m_1 m_2$ will satisfy the kernel condition. Let us assume that $m < m_1 m_2$. Notice that $\mathrm{Ker}(M)$ is a subspace of $\mathbb{C}^{m_1 m_2}$. By Proposition B.2, the maximum

dimension of subspace of $\mathbb{C}^{m_1 m_2}$ which does not contain any nonzero element of $\bigotimes_{i\in K_1}\mathbb{C}^{n_i}\otimes\bigotimes_{j\in K_2}\mathbb{C}^{n_j}$ is equal to

$$\begin{aligned} d(K_1,K_2) &:= m_1m_2-\sum_{i\in K_1}n_i-\sum_{j\in K_2}n_j+|K_1|+|K_2|-1 \\ &= m_1m_2-\sum_{i\in K_1}(n_i-1)-\sum_{j\in K_2}(n_j-1)-1. \end{aligned}$$

On the other hand, $\dim\mathrm{Ker}(M)\geqslant m_1m_2-m$ for all $m\times m_!m_2$ matrices and the equal holds when M has full row rank, i.e., $\mathrm{rank}\,(M)=m$. Therefore, the $m\times m_1m_2$ matrix M satisfying the kernel condition will always exist when $d(K_1,K_2)\geqslant m_1m_2-m$, or equivalently,

$$m\geqslant\sum_{i\in K_1}(n_i-1)+\sum_{j\in K_2}(n_j-1)+1. \tag{B.9}$$

Notice that for any positive integers $a_1,\ldots,a_k$,

$$\prod_{j=1}^{k}(a_j+1)\geqslant\sum_{1\leqslant i<j\leqslant k}a_ia_j+\sum_{j=1}^{k}a_j+1\geqslant\sum_{j=1}^{k}a_j+\sum_{j=1}^{k}a_j+1=2\sum_{j=1}^{k}a_j+1\quad\text{if}\quad k\geqslant 3.$$

Assume $k\geqslant 3$ and take $a_j=n_j-1$ in the above equation, we have

$$m=\prod_{i\in K}n_i\geqslant 2\sum_{i\in K}(n_i-1)+1\geqslant\sum_{i\in K_1}(n_i-1)+\sum_{j\in K_2}(n_j-1)+1.$$

Therefore, the matrix M exists when $k\geqslant 3$. For $k=2$,

$$m=\prod_{j=1}^{2}n_j=2\sum_{j=1}^{2}(n_j-1)+\prod_{j=1}^{2}(n_j-2)\geqslant\sum_{i\in K_1}(n_i-1)+\sum_{j\in K_2}(n_j-1)+0,$$

and the equality holds if and only if $K_1=K_2=K=\{1,2\}$ and at least one of n_i is equal to 2. In all other cases, the above inequality is strict, and therefore, the inequality (B.9) holds. Finally, suppose $K_1=K_2=K=\{1,2\}$ and $2\in\{n_1,n_2\}$. We may assume $n_1=2$, then

$$d(K_1,K_2)=4n_2^2-2n_2-1<4n_2^2-2n_2\leqslant\dim\mathrm{Ker}(M)$$

for any $(2n_2) \times (2n_2)(2n_2)$ matrix M. Therefore, there is no matrix M satisfying the kernel condition in this case. $\square$

After we obtained the above result, it has come to our attention that Lim [42] has already given a necessary and sufficient condition for the existence of linear maps preserving nonzero decomposable tensor for any algebraically close field, see [42, Proposition 2.8]. This existence condition is actually equivalent to the inequality (B.9) in our proof. Also a similar conclusion on linear maps on matrix space is obtained in a recent work of Lim in [41] too.

Finally, we apply Theorem B.1 to obtain the following corollaries, which generalize the results of Zheng et al. [62] and Lim [43].

Corollary B.1. *Let $n_1, \ldots, n_k$ be positive integers larger than or equal to 2 and let $m = \prod_{i=1}^{k} n_i$. Suppose $\phi : M_m \to M_m$ is a linear map. If*

$$\operatorname{rank}(\phi(A_1 \otimes \cdots \otimes A_k)) = 1 \quad \textit{whenever} \quad \operatorname{rank}(A_1 \otimes \cdots \otimes A_k) = 1 \quad \textit{for all } A_i \in M_{n_i},$$

and there is a matrix $X_1 \otimes \cdots \otimes X_k$ with $X_i \in M_{n_i}$ and $\operatorname{rank} X_i > 1$ for $i = 1, \ldots, k$ such that

$$\operatorname{rank}(\phi(X_1 \otimes \cdots \otimes X_k)) = \operatorname{rank}(X_1 \otimes \cdots \otimes X_k),$$

then ϕ has the form

$$\phi(A_1 \otimes \cdots \otimes A_k) = M(\psi_1(A_1) \otimes \cdots \otimes \psi_k(A_k))N^T$$

for all $A_i \in M_{n_i}$ with $i = 1, \ldots, k$, where ψ_j is the identity map or the transpose map for $j = 1, \ldots, k$, and $M, N \in M_m$ satisfy

$$\operatorname{Ker}(M) \cap (\mathbb{C}^{n_1} \otimes \cdots \otimes \mathbb{C}^{n_k}) = \{0\} \quad \textit{and} \quad \operatorname{Ker}(N) \cap (\mathbb{C}^{n_1} \otimes \cdots \otimes \mathbb{C}^{n_k}) = \{0\}.$$

Proof. By Theorem B.1, ϕ has the form (B.4) with partition $\{P_1, P_2, P_3, P_4\}$ as defined in the theorem. Notice that $\operatorname{rank}(\operatorname{vec}(A)) = 1$ for any matrix A. Suppose

$P_3 \cup P_4 \neq \varnothing$. Then

$$\begin{aligned} \text{rank } (\phi(X_1 \otimes \cdots \otimes X_k)) &\leqslant \left(\prod_{j \in P_1 \cup P_2} \text{rank } (X_j) \right) \left(\prod_{j \in P_3 \cup P_4} \text{rank } (\text{vec}(X_j)) \right) \\ &= \left(\prod_{j \in P_1 \cup P_2} \text{rank } (X_j) \right) < \text{rank } (X_1 \otimes \cdots \otimes X_k), \end{aligned}$$

which contradicts the assumption. So $P_3 \cup P_4 = \varnothing$ and ϕ has the asserted from. □

Corollary B.2. *Let $n_1, \ldots, n_k$ be positive integers larger than or equal to 2 and let $m = \prod_{i=1}^{k} n_i$. Suppose $\phi : M_m \to M_m$ is a linear map. Then*

$$\text{rank } (\phi(A_1 \otimes \cdots \otimes A_k)) = 1 \quad \textit{whenever} \quad \text{rank } (A_1 \otimes \cdots \otimes A_k) = 1 \quad \textit{for all } A_i \in M_{n_i},$$

and $\phi(X_1 \otimes \cdots \otimes X_k)$ is nonsingular for some $X_1 \otimes \cdots \otimes X_k$ with $X_i \in M_{n_i}$ if and only if there exist nonsingular matrices $M, N \in M_m$ such that

$$\phi(A_1 \otimes \cdots \otimes A_k) = M \left(\psi_1(A_1) \otimes \cdots \otimes \psi_k(A_k) \right) N \quad \textit{for all} \quad A_i \in M_{n_i},\ i = 1, \ldots, k,$$

where ψ_j, $j = 1, \ldots, k$ is either the identity map or the transpose map.

Proof. The sufficient part is clear. For the necessary part, by Theorem B.4 and a similar argument as in the proof of Corollary B.1, one can show that $P_3 \cup P_4 = \varnothing$ and M and N are both nonsingular. Then the result follows. □

Bibliography

[1] D. Aharonov and M. Ben-Or. Fault tolerant quantum computation with constant error. *SIAM J. Comput.*, 38:1207–1282, 1997.

[2] E. Alfsen and F. Shultz. Unique decompositions, faces, and automorphisms of separable states. *Journal of Mathematical Physics*, 51:251–253, 2010.

[3] C. Bennett, J. S. D.P. Divincenzo, and W. Wootters. Mixed state entanglement and quantum error correction. *Phys. Rev. A*, 54:3824–3851, 1996.

[4] A. Calderbank, E. Rains, P. Shor, and N. Sloane. Quantum error correction via codes over gf(4). *IEEE Trans. Inf. Th.*, 44:1369–1387, 1998.

[5] A. Calderbank and P. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, 1996.

[6] K. Chen and L. Wu. A matrix realignment method for recognizing entanglement. *Quantum Inf. Comput*, 3:193–202, 2003.

[7] R. Cleve and D. Gottesman. Efficient computations of encodings for quantum error correction. *Phys. Rev. A*, 56:76–82, 1997.

[8] A. Fošner, Z. Huang, C.-K. Li, Y.-T. Poon, and N.-S. Sze. Linear maps preserving the higher numerical range of tensor product of matrices. *Linear and Multilinear Algebra*, 62:776–791, 2014.

[9] A. Fošner, Z. Huang, C.-K. Li, and N.-S. Sze. Linear maps preserving ky fan norms and schatten norms of tensor product of matrices. *SIAM J. Matrix Anal. Appl.*, 34:673–685, 2013.

[10] A. Fošner, Z. Huang, C.-K. Li, and N.-S. Sze. Linear maps preserving numerical radius of tensor product of matrices. *J. Math. Anal. Appl.*, 407:183–189, 2013.

[11] A. Fošner, Z. Huang, C.-K. Li, and N.-S. Sze. Linear preservers and quantum information science. *Linear and Multilinear Algebra*, 61:1377–1390, 2013.

[12] S. Friedland, C.-K. Li, Y.-T. Poon, and N.-S. Sze. The automorphism group of separable states in quantum information theory. *Journal of Mathematical Physics*, 52:042203, 2011.

[13] G. Frobenius. Uber die darstellung der endlichen gruppen durch linear substitutionen. *Sitzungsber Deutsch. Akad. Wiss. Berlin*, pages 994–1015, 1897.

[14] F. Gaitan. *Quantum Error Correction and Fault Tolerant Quantum Computing*. CRC Press, New York, 2008.

[15] D. Gottesman. Class of quantum error-correcting codes saturating the quantum hamming bound. *Phys. Rev. A*, 54:1862–1868, 1996.

[16] D. Gottesman. Stabilizer codes and quantum error correction. *Ph.D. thesis, California Institute of Technology, Padadena, CA*, 1997.

[17] D. Gottesman. Quantum error correction and fault-tolerance. *Encyclopaedia of Mathematical Physics*, 2006:196–201, 2006.

[18] U. Gungordu, C.-K. Li, M. Nakahara, Y.-T. Poon, and N.-S. Sze. Recursive encoding and decoding of the noiseless subsystem for qudits. *Phys. Rev. A*, 89:042301, 2014.

[19] L. Gurvits. Classical deterministic complexity of edmonds problem and quantum entanglement. In *in Proceedings of the 35th ACM Symposium on Theory of Computing, ACM Press, New York*.

[20] A. Guterman, C.-K. Li, and P. Šemrl. Some general techniques on linear preserver problems. *Linear Algebra Appl.*, 315:61–81, 2000.

[21] M. Horodecki, P. Horodecki, and R. Horodecki. Separability of mixed states: necessary and sufficient conditions. *Phys. Lett. A*, 223:1–8, 1996.

[22] Z. Huang, S. Shi, and N.-S. Sze. Linear rank preservers of tensor products of rank one matrices. *Linear Algebra Appl.*, 508:255-271, 2016.

[23] N. Johnston. Characterizing operations preserving separability measures via linear preserver problems. *Linear Multilinear Algebra*, 59:1171–1187, 2011.

[24] J. Kempe, D. Bacon, D. Lidar, and K. Whaley. Theory of decoherencefree fault-tolerant universal quantum computation. *Phys. Rev. Lett.*, 63:42307–42335, (2001.

[25] A. Kitaev. Quantum computation: algorithms and error correction. *Russian Math. Surv.*, 52:1191–1249, 1997.

[26] A. Kitaev. Fault-tolerant quantum computation by anions. *Annals of Physics*, 303:2–30, 2003.

[27] A. Klappenechker and M. Rotteler. Clifford code constructions of operator quantum error correcting codes. *arXiv:quant-ph/0604.161v2*.

[28] A. Klappenechker and M. Rotteler. On the structure of nonstabilizer clifford codes. *Quant. Inf.& Comp.*, 4:152–160, 2004.

[29] E. Knill and R. Laflamme. Theory of quantum error-correcting codes. *Phys. Rev. A*, 55:900–911, 1997.

[30] E. Knill and R. Laflamme. Theory of quantum error correction for general noise. *Phys. Rev. Lett.*, 84:2525–2528, 2000.

[31] D. Kribs, R. Laflamme, and D. Poulin. Unified and generalized approach to quantum error correction. *Phys. Rev. Lett.*, 94:180501, 2005.

[32] D. Kribs, R. Laflamme, D. Poulin, and M. Lesosky. Operator quantum error correction. *Quant. Inf.& Comp.*, 6:383–399, 2006.

[33] C.-K. Li, M. Nakahara, Y.-T. Poon, and N.-S. Sze. Maximal noiseless code rates for collective rotation channels on qudits. *Quantum Information Processing*, 14:4039–4055, 2015.

[34] C.-K. Li, M. Nakahara, Y.-T. Poon, N.-S. Sze, and H. Tomita. Efficient quantum error correction for fully correlated noise. *Phys. Lett. A*, 375:3255–3258, 2011.

[35] C.-K. Li, M. Nakahara, Y.-T. Poon, N.-S. Sze, and H. Tomita. Recursive encoding and decoding of noiseless subsystem and decoherence free subspace. *Phys. Lett. A*, 84:044301, 2011.

[36] C.-K. Li, M. Nakahara, Y.-T. Poon, N.-S. Sze, and H. Tomita. Recovery in quantum error correction for general noise without measurement. *Quant. Inf.& Comp.*, 12:149–158, 2012.

[37] C.-K. Li and S. Pierce. Linear preserver problems. *Amer. Math. Monthly*, 108:591–605, 2001.

[38] C.-K. Li, Y.-T. Poon, and N.-S. Sze. Linear preservers of tensor product of unitary orbits, and product numerical range. *Linear Algebra Appl.*, 438:3797–3803, 2013.

[39] X. Li, F. Deng, and H. Zhou. Faithful qubit transmission against collective noise without ancillary qubits. *Appl. Phys. Lett.*, 91:144101, 2007.

[40] D. Lidar, D. Bacon, and K. Whaley. Concatenating decoherence-free subspaces with quantum error correcting codes. *Phys. Rev. Lett.*, 82:4556–4559, 1999.

[41] M. Lim. A note on rank one non-increasing linear maps on tensor products of matrices. *preprint.*

[42] M. Lim. Additive preservers of non-zero decomposable tensors. *Linear Algebra Appl.*, 428:239–253, 2008.

[43] M. Lim. A note on linear preservers of certain ranks of tensor products of matrices. *Linear and Multilinear Algebra*, 63:1442–1447, 2015.

[44] M. Marcus and B. Moyls. Linear transformations on algebras of matrices. *Canad. J. Math.*, 11:61–66, 1959.

[45] M. Nakahara and T. Ohmi. *Quantum Computing, From Linear Algebra to Physical Realization*. CRC Press, New York, 2008.

[46] K. Parthasarathy. On the maximal dimension of a completely entangled subspace for finite level quantum systems. *Pro. Indian Acad. Sci. (Math. Sci.)*, 114:365–374, 2004.

[47] A. Peres. Quantum theory: Concepts and methods. 1995.

[48] A. Peres. Separability criterion for density matrices. *Phys. Rev. Lett.*, 77:1413–1415, 1996.

[49] D. Poulin. Stabilizer formalism for operator quantum error correction. *Phys. Rev. Lett.*, 95:230504, 2005.

[50] O. Rudolph. A separability criterion for density operators. *J. Phys. A: Math. Gen.*, 33:3951, 2000.

[51] B. Schumacher and M. Nielsen. Quantum data processing and error correction. *Phys. Rev. A*, 54:2629–2635, 1996.

[52] P. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:2493–2496, 1995.

[53] P. Shor. Fault-tolerant quantum computation. *arXiv:quant-ph/9605011*, pages 722–725, 1996.

[54] A. Steane. Error correcting codes in quantum theory. *Phys. Rev. Lett*, 77:793–797, 1996.

[55] A. Steane. Multiple particle interference and quantum error correction. *Proc. Roy. Soc. London. Ser. A*, 452:2551–2577, 1996.

[56] L. Viola, E. Fortunato, M. Pravia, E. Knill, R. Laflamme, and D. Cory. Experimental realization of noiseless subsystems for quantum information processing. *Science*, 293:2059–2063, 2001.

[57] R. Westwick. Transformations on tensor spaces. *Pacific J. Math.*, 23:613–620, 1967.

[58] R. Westwick. Transformations on tensor spaces ii, linear multilinear algebra. *Linear and Multilinear Algebra*, 40:81–92, 1995.

[59] R. Westwick. Decomposability under field extensions. *Linear and Multilinear Algebra,*, 41:251–253, 1996.

[60] J. Xu, B. Zheng, and A. Fošner. Linear maps preserving rank of tensor products of rank-one hermitian matrices. *Journal of the Australian Mathematical Society*, 98:407–428, 2015.

[61] C. Yang and J. Gea-Banacloche. Three-qubit quantum error-correction scheme for collective decoherence. *Phys. Rev. A*, 63:022311, 2001.

[62] B. Zheng, J. Xu, and A. Fošner. Linear maps preserving rank of tensor products of matrices. *Linear and Multilinear Algebra*, 63:366–376, 2015.