

An introduction to quantum error correction

Dheera Venkatraman

Term paper for 6.453, Fall 2006

1 Introduction

Communication at high speeds, long distances, and in unknown environments often requires combatting noise that may affect or destroy data before it has reached its intended destination. When designing robust, practical communication systems, it is important to take such effects into account and engineer a system to be as immune to noise as possible. Classical communications often relies upon various error detection and correction schemes, from the simple parity check to a variety of more sophisticated correction algorithms, designed to ensure nearly error-free transmission of data. Classical error-correction systems often employ redundancy and checksums to accomplish error correction; however, error correction in quantum communication channels is complicated by the fact that a qubit's state is affected by measurement. Furthermore, one may only have a single copy of each qubit to work with in quantum algorithms: in quantum cryptography algorithms such as BB84, for instance, it does not make sense to re-transmit qubits prior to establishing a key. In this paper, we explore the problem of quantum error correction and present some of the simple algorithms that have been proposed to correct errors of various types in a channel of qubits.

2 Classical error detection and correction

Classical bits take on one of exactly two states, 0 or 1, and can be reliably measured and duplicated at any time. Assuming a bit is not lost, a noisy channel may either flip a bit at probability p or leave the bit intact at probability $1 - p$; there is no other possibility. The

presence of noisy channels in real-world classical communications leads to the necessity for error detection, to ensure that data can be reliably transmitted between two parties. Here we provide a brief background of basic classical error detection and error correction techniques.

2.1 The parity check

The simplest form of classical error detection is the *parity check*, which may identify a single error in a string of classical bits. It involves taking a string of bits and adding a *parity bit* whose value is equal to the sum of all the other bits, modulo 2. For example, if the string 10101 is to be transmitted, a parity bit of $(1 + 0 + 1 + 0 + 1) \bmod 2 = 1$ can be transmitted at the end of the string. The receiving party can then ensure that the parity bit matches the data, and if a single bit error occurs anywhere (including the parity bit itself), the party can discard the data and request that the sender re-transmit the data. The main advantage to the parity check is its simplicity and efficiency (requiring only one additional bit for a potentially large number of transmitted bits) to verify a channel for errors. It is widely used today for many systems in which p is sufficiently low, making a catastrophic two-bit error nearly impossible. However, the parity check has a major drawback: if an error is detected, it requires a re-transmit of the data and cannot offer a correction mechanism on the receiving end. We now turn to the subject of error correction, in which the receiving end is able to recover the lost data in limited cases, by studying the simple repetition code.

2.2 The repetition code

Classical error correction is an extensively developed field with sophisticated methods to correct for bit errors with the least number of additional transmitted bits. However, for the purposes of this paper, and exploring simple quantum error correction mechanisms, we consider how a very basic classical error correction scheme might function. The repetition code employs redundancy: a single bit is duplicated a number of times and sent through a channel as a group. The receiving end then simply takes the majority value of the group [1]. For example, a single bit (such as 1), can be duplicated three times (111), passed through a

noisy channel that might flip its second bit (101), and corrected by simply taking the value of the majority of bits (1). This simple code will function as long as no more than one error occurs within the group. With the simple repetition code in hand, we now turn to the subject of quantum error correction and see that this is not possible to implement with qubits, but a very similar result can be accomplished.

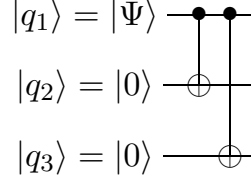
3 Quantum error correction

3.1 The bit flip code

While in a classical channel it is simple to measure and duplicate the bit being sent in order to create redundancy, this is not possible in a quantum channel due to the no-cloning theorem, which states that an qubit with a completely arbitrary state cannot be duplicated. Thus, a classical-style parity check is ruled out since a re-transmit of data may be impossible if there is only one chance to observe a qubit and no method to duplicate it. A single qubit also cannot be duplicated three times as seen previously in the repetition code, and any measurement of the qubit would result in its collapsing into the measurement basis, destroying some of the information in the qubit. However, it is possible to achieve similar results to the repetition code by using entanglement and syndrome measurements, in a process known as the *three-qubit bit flip code*. We begin with a state

$$|\Psi\rangle = a|0\rangle + b|1\rangle \tag{1}$$

with arbitrary complex a and b , that is to travel through a channel that may flip a qubit (change $|1\rangle$ to $|0\rangle$ and $|0\rangle$ to $|1\rangle$) with probability p . The first step of the procedure is to entangle the qubit with two other qubits using two CNOT gates with $|0\rangle$ inputs [1]:



The resulting combined state of the three qubits after the CNOT gate will then be

$$|\Psi'\rangle = a|000\rangle + b|111\rangle \quad (2)$$

. Note that this is very different from cloning a state (impossible for arbitrary states due to the no-cloning theorem). Instead, the result is merely a tensor product of three qubits that will be simultaneously projected into $|000\rangle$ or $|111\rangle$, if it were to be measured in that basis.

We then pass all three mutually entangled qubits through separate similarly constructed channels, with the assumption that each channel may flip its qubit with probability p . For instance, if the first qubit were to be flipped, the result would be $|\Psi'_e\rangle = a|100\rangle + b|011\rangle$. To diagnose bit flips in any of the three possible qubits, *syndrome diagnosis* is performed on the final state. This involves the use of four projection operators whose values can be measured without disturbing the state [1]:

$$P_0 = |000\rangle\langle 000| + |111\rangle\langle 111| \quad (3)$$

$$P_1 = |100\rangle\langle 100| + |011\rangle\langle 011| \quad (4)$$

$$P_2 = |010\rangle\langle 010| + |101\rangle\langle 101| \quad (5)$$

$$P_3 = |001\rangle\langle 001| + |110\rangle\langle 110| \quad (6)$$

If upto one bit flip has occurred in the communication channel in any of the three qubits, the measurement of the four projection operators above can reveal the *error syndrome* that occurred. For example, if the first qubit is flipped, we obtain $|\Psi'_e\rangle$ as defined above. One

conveniently finds that [1]

$$\langle \Psi'_e | P_0 | \Psi'_e \rangle = 0 \quad (7)$$

$$\langle \Psi'_e | P_1 | \Psi'_e \rangle = 1 \quad (8)$$

$$\langle \Psi'_e | P_2 | \Psi'_e \rangle = 0 \quad (9)$$

$$\langle \Psi'_e | P_3 | \Psi'_e \rangle = 0 \quad (10)$$

revealing that the error syndrome corresponding to P_1 occurred. Note that since they are in orthogonal subspaces, the four syndromes can be sequentially measured without destroying the state. The only information that is obtained from the syndrome measurement is the basis that $|\Psi'\rangle$ is currently in (of four orthogonal possibilities), which determines which bit, if any, was flipped. One can then proceed to restore the three qubit product state to its original state $|\Psi'\rangle$ by applying a bit flip to the appropriate qubit. For example, to repair $|\Psi'_e\rangle$ one would apply $\hat{\sigma}_x \otimes \hat{I} \otimes \hat{I}$ to the three-qubit state, where \hat{I} is the identity matrix and $\hat{\sigma}_x$ is the x -direction Pauli spin matrix, which is also the bit flip operator:

$$\hat{\sigma}_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (11)$$

The correct three-qubit state $|\Psi'\rangle$ is then restored. from which the original $|\Psi\rangle$ can be recovered [1].

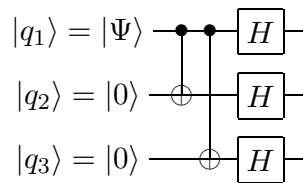
The bit flip code can be used to reliably transmit a qubit when a maximum of one error occurs for the three channels used. This emulates the behavior of the classical 3-bit repetition code. As with the classical scenario, the probability that exactly zero or one error occurs (the cases in which the bit is ultimately repaired and transmitted correctly) is

$$P_s = (1 - p)^3 + 3p(1 - p)^2 \quad (12)$$

If $p < 0.5$, then $1 - P_s < p$, that is, the error probability after the implementation of the bit flip scheme is less than the error probability of using a single channel and simply transmitting the qubit unprotected, making the scheme useful. However, the bit flip code is severely limited in that it *only* corrects for bit flip errors, while a quantum system can realistically introduce a continuum of error possibilities.

3.2 The phase flip code

While the bit flip scheme can be extremely effective in accomplishing its task, it is important to note that unlike classical bit errors where there is only one error possibility (the bit flip), there is a continuum of other error possibilities in qubit logic. Here, we explore and isolate the possibility of a channel to create a phase flip, in which the relative sign between the $|0\rangle$ and $|1\rangle$ states is flipped at probability p for each transmitted qubit. For example, a qubit in the state $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ may be transformed into $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ by an imperfect channel. This can be corrected for in a procedure almost identical to the bit flip code, except by simply rewriting the same $a|0\rangle + b|1\rangle$ state in the $|\pm\rangle$ basis and using a version of a CNOT gate that operates in the $|\pm\rangle$ basis to entangle the state with two other qubits. An equivalent result is obtained by simply using the bit flip setup and placing Hadamard gates after the qubits [1]:



The result is a conversion of the original state

$$|\Psi\rangle = \alpha |+\rangle + \beta |-\rangle \tag{13}$$

into the entangled state

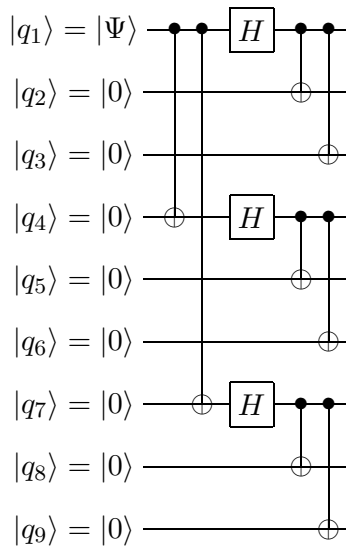
$$|\Psi'\rangle = \alpha |+++ \rangle + \beta |-- \rangle \tag{14}$$

where α and β are determined from a and b by the basis conversion. From here, it is easy to see that a method similar to the bit flip code can be applied to achieve similar results for phase flip correction [1].

3.3 Putting it all together: The Shor Code

Provided that the probability of an error is small, we have now seen to perform error correction over channels that may have either bit flip errors or phase flip errors. Realistically, neither code is practically useful on its own since imperfect channels may act on a state with any unitary matrix to create an error. However, since any arbitrary error on a *single* qubit can be expressed in terms of a combination of bit flip, phase flip, and combined bit-phase flip operations, it is possible to combine both the bit flip and phase flip codes in a unique way, referred to now as the *Shor code*, to correct for arbitrary qubit errors.

The first step of the Shor code is to set up qubits 1, 4, and 7 for the phase flip code, placing the product state of those three qubits as defined by equation 14. Following this, we then set up independent bit flip code systems on each of the three qubit lines generated by the phase flip code setup, creating a total of nine qubits that need to be transmitted [2]. A quantum circuit of the entire procedure is shown below.



Note that the overall effect of the system is to transform the original state $|\Psi\rangle = a|0\rangle + b|1\rangle$ into the product state of 9 qubits $|\Psi'\rangle = a|0_S\rangle + b|1_S\rangle$, where [2]

$$|0_S\rangle = (|000000000\rangle + |000000111\rangle + |000111000\rangle + |000111111\rangle + |111000000\rangle + |111000111\rangle + |111111000\rangle + |111111111\rangle)/\sqrt{8} \quad (15)$$

$$|1_S\rangle = (|000000000\rangle - |000000111\rangle - |000111000\rangle + |000111111\rangle - |111000000\rangle + |111000111\rangle + |111111000\rangle - |111111111\rangle)/\sqrt{8} \quad (16)$$

If a bit flip error occurs on a single qubit, it is easy to see that a bit flip code applied to a subset of the qubits will reverse the error. Namely, if the qubits were numbered from 1 to 9, independent syndrome analysis on the sets of states $\{1,2,3\}$, $\{4,5,6\}$, and $\{7,8,9\}$ will correct for such errors. As is seen from the diagram above, the 9 qubits can be treated as three independent bit flip correction systems, after the Hadamard gates.

With a little more insight it is possible to see that an arbitrary phase flip error on a single qubit of the 9 qubits can be repaired [1]. This is done by identifying the group of 3 qubits ($\{1,2,3\}$, $\{4,5,6\}$, or $\{7,8,9\}$) the phase flip error occurred in. This can be accomplished by doing syndrome analysis. We begin by considering each of the three groups of qubits above to be a "big" qubit, based on the product states $|000\rangle$ and $|111\rangle$ for each group. We then employ the basis transformation

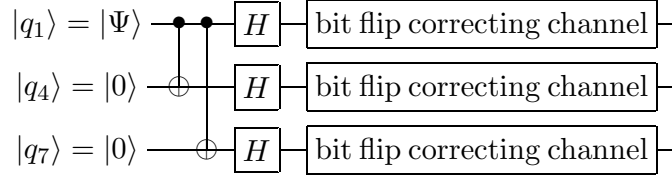
$$|+\rangle = |000\rangle + |111\rangle \quad (17)$$

$$|-\rangle = |000\rangle - |111\rangle \quad (18)$$

and perform phase flip syndrome analysis by using the three "big" qubits (also note that $|+\rangle \neq |+++ \rangle$). We then combine the three groups, considering that the three "big" qubits are part of a larger product state, expressible as $\alpha|+++\rangle + \beta|---\rangle$. Syndrome analysis can then detect and repair the overall phase of whichever "big" qubit, if any, a phase flip occurred. With the assumption of upto one qubit being affected of the nine, a phase flip in any single qubit of a three-qubit group or "big" qubit is equivalent to an overall phase flip

of the group, so this procedure is capable of repairing a single qubit phase flip error.

An alternative way to visualize a single phase flip being corrected is to abstract out and black-box the three bit-flip systems in the diagram above and treat each black-box as a working single qubit channel of its own. The remaining system is just a phase flip-correcting setup as described earlier:



We have shown that a bit flip or phase flip error in a single qubit of the nine qubits being transmitted can be repaired. It is also evident that the two procedures can be done in sequence to cure both a bit flip and a phase flip in a single qubit, as the black-boxed bit flip mechanisms will automatically repair single bit flip errors, transparently to the phase flip mechanism. The functionality of the bit flip syndrome analysis is also immune to phase flip errors and will still repair a flipped bit, leaving the phase error intact. The phase flip mechanism can then repair such a single phase flip that may have occurred.

It turns out that the two error correcting capabilities (bit flip and phase flip) are sufficient to correct for *any* arbitrary error to a single qubit [2], as we will demonstrate. Consider an arbitrary error to be an arbitrary unitary transform \hat{U} to the qubit. That is, $\hat{U}|\psi\rangle = |\psi_e\rangle$ where $|\psi\rangle$ is the original state of the single qubit (of the nine in the Shor code) that is being affected. Such an arbitrary unitary matrix can be expressed in the form

$$\hat{U} = C_0\hat{I} + C_1\hat{\sigma}_x + iC_2\hat{\sigma}_y + C_3\hat{\sigma}_z \quad (19)$$

where C_0 , C_1 , C_2 , and C_3 are complex coefficients, \hat{I} is the identity, and the Pauli spin matrices are given by

$$\hat{\sigma}_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \hat{\sigma}_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \hat{\sigma}_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (20)$$

When \hat{U} is equal to the identity operator \hat{I} , the state is unaffected. However, it is easy to see by inspection that $\hat{U} = \hat{\sigma}_x$ corresponds to a bit flip error, $\hat{U} = \hat{\sigma}_z$ corresponds to a phase flip error, and $\hat{U} = i\hat{\sigma}_y$ corresponds to both a bit flip and a phase flip (note that $\hat{\sigma}_z\hat{\sigma}_x = i\hat{\sigma}_y$). Since an arbitrary matrix \hat{U} can be expressed in terms of these matrices, an arbitrary error is a superposition of these four actions on the state (no action, bit flip, phase flip, and combined bit-phase flip). Since it has been already shown that the Shor code can correct for any of these four complete and orthogonal actions, the application of the Shor code to an arbitrary unitary matrix transformation to a single qubit will cause it to collapse into a single one of these four actions on the state [1], resulting in the error correction system working as intended.

Of course, the Shor code is only designed to work in the case that upto a single qubit of nine is affected. Summing the probabilities of exactly zero and exactly one error occurring, the probability that a qubit will be correctly transmitted in a Shor code system with arbitrary errors of probability p per qubit is

$$P_s = (1 - p)^9 + 9p(1 - p)^8 \tag{21}$$

If p is too high, there is a high probability that more than one qubit error may occur in the nine qubits needed for transmission, causing the Shor code to potentially fail. However, as long as the communication channels can be built to have a low enough p , the Shor code's use of nine qubits succeeds in improving the success rate of transmission, handling arbitrary errors. In particular, numerical calculations will show that if p is less than approximately 3%, $1 - P_s < p$, making the Shor code useful.

3.4 The CSS codes

The Shor code requires nine qubits to be transmitted to detect errors in a single qubit. However, it is possible to correct for arbitrary errors using a smaller number of transmitted qubits. The Calderbank-Shor-Steane (CSS) codes, which we shall explore in this section,

provide a theoretical foundation to more easily identify such possibilities for a more efficient error correction system.

The CSS codes are derived from the classical linear codes [3]. Classical linear codes are specified by a generator matrix G or a parity check matrix H with binary values such that $HG = 0$ [1]. The generator matrix G is multiplied by a vector of binary information to obtain the codewords for transmission. For example, the classical repetition code is given by [4]

$$G = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad (22)$$

where it is clear that multiplying G by a single bit will yield a vector containing that bit repeated three times. The width k of G determines the number of bits that are encoded at a time (in this case, one), and the height n of G determines the number of bits in the codeword. Such a code is called a $[n, k]$ code. The parity check matrix H is then used to determine the error syndrome. If x is the data to be transmitted, Gx is the encoded string which is transmitted (the codeword). Let $v \approx Gx$ be the received codeword after a noisy channel. If $Hv = 0$, then the transmission was correct; otherwise, $H(v + e)$ will provide a unique code for the classical error syndrome of corresponding to the error e that has occurred [4]. The repetition code is also part of a larger class of linear codes called the Hamming codes, which are capable of correcting upto one bit error per codeword [4].

The CSS codes extend the classical linear codes to correct for arbitrary qubit errors. A valid CSS code requires two classical linear codes, $C_1(G_1, H_1)$ which is $[n, k_1]$ and $C_2(G_2, H_2)$ which is $[n, k_2]$, for some $k_1 < k_2$. It must also be the case that $C_1 \subset C_2$, where the sets C_1 and C_2 are defined as the set of possible codewords generated by that code. A CSS code, denoted $CSS(C_1, C_2)$ encoding $k_1 - k_2$ qubits into n qubits can then be defined [3].

For a CSS code encoding $k_1 - k_2$ qubits, we map the first $2^{k_1 - k_2}$ binary numbers (starting with 0) to codewords in C_1 . Denote a mapping from binary number j to a codeword as $x_j \in C_1$. Note that a set of states indexed by binary numbers would form a complete

orthonormal basis for these $k_1 - k_2$ qubits. We denote such states as $|j\rangle$ for binary number j (where $0 \leq j \leq k_1 - k_2 - 1$). Such a mapping must also satisfy the condition $x_i \oplus x_j \notin C_2$ [5], where \oplus here represents the bit-wise modulo 2 sum (i.e. bit-wise exclusive OR).

The quantum codeword for state $|j\rangle$ (not the classical codeword for binary number j) is denoted here as $|x_j \oplus C_2\rangle$ and given by [5]

$$|j\rangle \longrightarrow |x_j \oplus C_2\rangle \equiv \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x_j \oplus y\rangle \quad (23)$$

where by $|x_j \oplus y\rangle$ we mean the n -qubit product state corresponding to the binary number $x_j \oplus y$. The original requirement that $x_i \oplus x_j \notin C_2$ implies [5]

$$\langle x_i \oplus C_2 | x_j \oplus C_2 \rangle = \delta_{ij} \quad (24)$$

giving an orthonormal basis to encode any superpositions of qubits into. In the *Steane code*,

$$H_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \quad (25)$$

and

$$H_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (26)$$

which have the unique property that $H_1 = G_2^T$ and $H_2 = G_1^T$ when the generator matrices G_1 and G_2 are calculated. It can also be verified that this choice gives $C_2 \subset C_1$ if the possible codewords are listed [1]. Since C_1 is a classical $[n = 7, k_1 = 4]$ code and C_2 is a classical $[n = 7, k_2 = 3]$ code, the Steane code encodes $4 - 3 = 1$ qubits into 7 qubits, permitting a maximum of one qubit error since C_1 and C_2 are in fact Hamming codes [1]. In this case, since $k_1 - k_2 = 1$, we enumerate codewords x_j for $0 \leq j \leq (2^{k_1 - k_2} - 1) = 1$. We select

$x_0 = 0000000$ and $x_1 = 1111111$, noting that $x_0 \oplus x_1 \notin C_2$ as is required. Following equation 23 gives the Steane codewords [5]:

$$\begin{aligned} |x_0 \oplus C_2\rangle &= (|0000000\rangle + |0001111\rangle + |0110011\rangle + |1010101\rangle \\ &\quad + |0111100\rangle + |1011010\rangle + |1100110\rangle + |1101001\rangle)/\sqrt{8} \end{aligned} \quad (27)$$

$$\begin{aligned} |x_1 \oplus C_2\rangle &= (|1111111\rangle + |1110000\rangle + |1001100\rangle + |0101010\rangle \\ &\quad + |1000011\rangle + |0100101\rangle + |0011001\rangle + |0010110\rangle)/\sqrt{8} \end{aligned} \quad (28)$$

from which a qubit $|\Psi\rangle = a|0\rangle + b|1\rangle$ can be encoded into $a|x_0 \oplus C_2\rangle + b|x_1 \oplus C_2\rangle$ through a network of CNOT and Hadamard gates. Error detection in CSS codes such as the Steane code is accomplished by using the error-correcting properties of the classical codes C_1 and C_2 [1]. By assuming a length n bit flip error vector e_1 and phase flip error vector e_2 (where 0 indicates no flip and 1 indicates a flip for each vector position), an incorrectly transmitted codeword for a basis state $|j\rangle$ may be written as [1]

$$|(x_j \oplus C_2)_e\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x_j \oplus y) \cdot e_2} |x_j \oplus y \oplus e_1\rangle \quad (29)$$

where only bit flips, phase flips, and combined bit-phase flips are considered for each bit (as explained with the Shor code above, a code that is shown to correct for these errors will correct for arbitrary errors). One then introduces an *ancilla* state containing $n - k_2$ (the width of G_2 , thus the height of H_1) qubits initially prepared in the $|0\rangle$ state and constructs a quantum circuit that implements a computation of the parity check matrix H_1 using CNOT and Hadamard gates to evaluate the product of H_1 and the faulty state, placing the result in the ancilla states. This results in the combined state of the system and the ancilla states [1]

$$|(x_j \oplus C_2)_e\rangle |\psi_{ancilla}\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x_j \oplus y) \cdot e_2} |x_j \oplus y \oplus e_1\rangle |H_1(x_j \oplus y \oplus e_1)\rangle \quad (30)$$

$$= \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x_j \oplus y) \cdot e_2} |x_j \oplus y \oplus e_1\rangle |H_1 e_1\rangle \quad (31)$$

The resulting measurement of the $(n - k_2)$ -digit binary number H_1e_1 , determined by measurement of the ancilla, will be a unique code for the error syndrome describing which bits were flipped, if any [1]. After fixing any bit flips, the phase flips may be approached in a similar manner by first applying the Hadamard operator to every qubit, computing H_2e_2 in a new ancilla, measuring the ancilla and correcting any phase flips, and applying the Hadamard operator again to every qubit to restore the state [5]. The mathematics of the phase flip correction is detailed by Nielsen and Chuang [1].

4 Conclusions

We have demonstrated the reasons that classical encoding schemes for error detection and correction cannot be directly applied to quantum mechanical channels with arbitrary qubits and have shown the structure of basic quantum error correction techniques. The simple bit flip and phase flip codes demonstrate how a single type of error can be corrected, and are combined in the Shor code to correct for arbitrary qubit errors. The framework for the CSS codes has also been discussed, opening up a large class of quantum error correcting codes, including the Steane code which uses only 7 qubits (instead of the Shor code's 9) to perform arbitrary error correction on one qubit. Another more complex code not discussed here achieves the quantum Hamming bound of 5 qubits, the minimum possible integral number of qubits needed to correct arbitrary errors in a single qubit using the linear techniques [1]. Some examples of further efforts in quantum error correction lie in the stabilizer formalism [6] and the topological error-correcting codes introduced by Kitaev [7]. Michael Ben-Or and Dorit Aharonov also asserted, in what is now known as the Threshold Theorem, that if the error rate of quantum gates is low enough, it is possible to correct for all errors and achieve fault-tolerant quantum computing with a constant error rate, regardless of the complexity of the system [8]. Thus, combining the developments of quantum error correction with well-designed low-error elements may show a promise for practical, sophisticated quantum computing systems in the future.

References

- [1] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- [2] P. Shor, *Scheme for reducing decoherence in qunatum computer memory*. PR A, 52 2493 (1995).
- [3] A. Calderbank and P. Shor, *Good quantum error-correcting codes exist*. PR A, 54 1098 (1996).
- [4] J. Boileau and D. Gottesman, Lecture notes for C&O 639 on Quantum Error Correction, 22-January-2004.
- [5] J. Watrous, Lecture notes for CPSC 519/619, University of Calgary, 23-March-2006.
- [6] D. Gottesman, *Stabilizer codes and quantum error correction*. Dissertation, 21-May-1997.
- [7] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, *Topological quantum memory*. Journal of Mathematical Physics 43 (2002).
- [8] D. Aharonov and M. Ben-Or, *Fault Tolerant Quantum Computation with Constant Error*. quant-ph/9611025.