Solution Set #1

CO 639: Quantum Error Correction
Instructor: Daniel Gottesman

Problem 1. Uncorrectable Errors and the Nine-Qubit Code

a) The nine-qubit code can correct one bit flip error in each set of three qubits, plus one phase error. Also, two phase errors in the same set of three qubits act the same on the codewords, so do nothing to the state (the product is in the stabilizer). Thus, the code can correct X_2X_7, X_5Z_6, and Z_5Z_6. X_1X_3 cannot be corrected because it involves two bit flip errors in the same set of three, and Y_2Z_8 cannot be corrected because it involves two phase flip errors on different sets of three (the bit flip part of Y_2 can be corrected, however).

b) For X_1X_3, the error correction procedure notes that qubit number 2 is the misfit, and "corrects" it by performing the bit flip operation X_2. Thus, the net effect is to flip all of the first three qubits. Thus, the encoded $|\overline{0}\rangle$ state does not change (as $|000\rangle + |111\rangle$ becomes $|111\rangle + |000\rangle$), but the encoded $|\overline{1}\rangle$ state becomes $-|\overline{1}\rangle$ (as $|000\rangle - |111\rangle$ becomes $|111\rangle - |000\rangle$). That is, $\alpha|\overline{0}\rangle + \beta|\overline{1}\rangle$ becomes $\alpha|\overline{0}\rangle - \beta|\overline{1}\rangle$; the logical operation is an encoded Z.

   For Y_2Z_8, we can write $Y_2 = iX_2Z_2$. The code can correct X_2, but the residual phase error Z_2Z_8 cannot be corrected. (The factor i is an overall phase, which has no physical significance and does not count as an error.) The correction procedure notes that the phase on the middle block of three is different, and tries to fix it with a Z_5, say, making the overall error a $Z_2Z_5Z_8$. Thus, $|000\rangle + |111\rangle$ becomes $|000\rangle - |111\rangle$ on all three blocks and vice-versa, changing $|\overline{0}\rangle$ into $|\overline{1}\rangle$. This is a logical X operation: $\alpha|\overline{0}\rangle + \beta|\overline{1}\rangle$ becomes $\beta|\overline{0}\rangle + \alpha|\overline{1}\rangle$.

Problem 2. Maximally Entangled States and QECCs

a) To test if a three-qubit state is maximally entangled, we only need to check that the density matrix of any single qubit of the state is the identity. For the GHZ state, this is clearly true. The stabilizer of the GHZ state is generated by $\{X \otimes X \otimes X, Z \otimes Z \otimes I, I \otimes Z \otimes Z\}$.

b) First, note that
$$\langle\psi|E|\psi\rangle = \mathrm{Tr}(\rho E), \tag{1}$$
   where ρ is the density matrix of $|\psi\rangle$ on the qubits where E acts. For $\mathrm{wt}\, E \leq n/2$, by the definition of a maximally entangled state, $\rho = I$. Thus, $\mathrm{Tr}(\rho E) = \mathrm{Tr}\, E = \delta_{E,I}$ (since the Pauli matrices X, Y, Z have trace 0). Therefore, $|\psi\rangle$ is an $[[n, 0, d]]$ code with $d = \lfloor n/2 \rfloor + 1$.

c) Using the arguments in the last part, we see that $\mathrm{Tr}(\rho E) = \delta_{E,I}$ whenever $\mathrm{wt}\, E \leq n/2$. The Pauli operators form a basis for the set of matrices, so we can expand $\rho = \sum_{P \in \mathcal{P}} c_P P$, where $\mathcal{P}$ is the Pauli group, and the $\{c_P\}$ are real numbers (real, not complex, since ρ is Hermitian). Then $\mathrm{Tr}(\rho E) = c_E$, and we find $c_E = \delta_{E,I}$, so $\rho = I$. Thus, the state is maximally entangled.

d) The 5-qubit code is a nondegenerate $[[5, 1, 3]]$ stabilizer code. Note that for a stabilizer code of distance d, when $\mathrm{wt}\, E < d$ and $|\psi\rangle$ is a codeword, $\langle\psi|E|\psi\rangle$ is either 0 (when E anticommutes with some element M of the stabilizer, thereby taking $|\psi\rangle$ to an orthogonal eigenspace of M) or 1 (when E is in the stabilizer). For a nondegenerate stabilizer code, it is therefore always 0 (except for $E = I$). Thus, any valid codeword of the $[[5, 1, 3]]$ code will be a maximally entangled state — for instance, the encoded $|\overline{0}\rangle$ state (which is written out in ket notation in problem 6).

e) If we had just taken the $k = 0$ limit of the usual QECC conditions, we would have gotten

$$\langle \psi | E | \psi \rangle = C_E, \tag{2}$$

which is completely vacuous — all states $|\psi\rangle$ satisfy this condition.

Problem 3. Other Forms and Consequences of the QECC Conditions

First, there is a typo in this problem, which is probably obvious: eqs. (3) and (4) should both read $E_a^\dagger E_b$, not $E_a^\dagger E_b^\dagger$.

a) Let $|\psi\rangle = \sum_i \alpha_i |i\rangle$, with $\sum_i |\alpha_i|^2 = 1$. This notation is for the unencoded states, but the same relation immediately follows for the encoded states: $|\overline{\psi}\rangle = \sum_i \alpha_i |\overline{i}\rangle$. From eq. (3) of the problem set, it follows that

$$\begin{aligned} \langle \overline{\psi} | E_a^\dagger E_b | \overline{\psi} \rangle &= \sum_{i,j} \alpha_j^* \alpha_i \langle \overline{j} | E_a^\dagger E_b | \overline{i} \rangle && (3) \\ &= \sum_{i,j} \alpha_j^* \alpha_i C_{ab} \delta_{ij} && (4) \\ &= C_{ab}. && (5) \end{aligned}$$

Conversely, let us start from eq. (4) of the problem set and consider the states $|\overline{i}\rangle \pm |\overline{j}\rangle$ (for $i \neq j$). Then

$$\begin{aligned} \langle \overline{i} | E_a^\dagger E_b | \overline{i} \rangle + 2\,\mathrm{Re} \langle \overline{i} | E_a^\dagger E_b | \overline{j} \rangle + \langle \overline{j} | E_a^\dagger E_b | \overline{j} \rangle &= C_{ab} && (6) \\ \langle \overline{i} | E_a^\dagger E_b | \overline{i} \rangle - 2\,\mathrm{Re} \langle \overline{i} | E_a^\dagger E_b | \overline{j} \rangle + \langle \overline{j} | E_a^\dagger E_b | \overline{j} \rangle &= C_{ab}, && (7) \end{aligned}$$

so

$$2\,\mathrm{Re} \langle \overline{i} | E_a^\dagger E_b | \overline{j} \rangle = 0. \tag{8}$$

Similarly, by considering $|\overline{i}\rangle \pm i|\overline{j}\rangle$, we find that

$$2\,\mathrm{Im} \langle \overline{i} | E_a^\dagger E_b | \overline{j} \rangle = 0. \tag{9}$$

The remaining condition,

$$\langle \overline{i} | E_a^\dagger E_b | \overline{i} \rangle = C_{ab}, \tag{10}$$

is just part of eq. (4) without any manipulation.

If $|\overline{\psi}\rangle$ were only required to run over basis states in eq. (4) of the problem set, the condition would not work. For instance, we can choose a stabilizer with n generators, and define $|\overline{i}\rangle$ to be the state with error syndrome i (an n-bit vector). All of these 2^n states have the same C_{ab}, even if we let E_a and E_b be arbitrary Pauli errors (of any weight) but the code clearly cannot correct all these errors, since Pauli operators map the states with various syndromes into each other.

b) Let $\rho(|\psi\rangle)$ be the density matrix of $|\overline{\psi}\rangle$ on some particular set of $d-1$ qubits. Eq. (4) of the problem set says that

$$\mathrm{Tr}(\rho(|\psi\rangle)E) = \mathrm{Tr}(\rho(|\phi\rangle)E) \tag{11}$$

whenever E is Pauli matrix of weight $d-1$ or less. As in the solution to problem 2c, we can expand the density matrices in terms of the Pauli operators: $\rho(|\psi\rangle) = \sum c_P(|\psi\rangle)P$ and $\rho(|\phi\rangle) = \sum c_P(|\phi\rangle)P$, and the above equation then becomes simply the statement that $c_E(|\psi\rangle) = c_E(|\phi\rangle)$ for all E of weight $d-1$ or less, which are all that are required to expand a density matrix on $d-1$ qubits. Therefore $\rho(|\psi\rangle) = \rho(|\phi\rangle)$.

c) We can just take the argument of the previous part backwards: $\rho(|\psi\rangle) = \rho(|\phi\rangle)$, therefore $c_E(|\psi\rangle) = c_E(|\phi\rangle)$, therefore $\mathrm{Tr}(\rho(|\psi\rangle)E) = \mathrm{Tr}(\rho(|\phi\rangle)E)$, and therefore we have eq. (4) from the problem set, which from part a is equivalent to the usual QECC conditions.

Problem 4. Correcting X and Z, but not Y

a) This problem was badly phrased to make it harder than I had intended (although there is actually a solution to the harder version of the problem). It was sufficient to find a code, as given below, that corrects X and Z errors, but cannot distinguish Y errors from each other. The code below can correct a Y error on, say, the first qubit, if you know that is the only place a Y error can occur.

We can choose as generators of this stabilizer:

$$\begin{array}{cccccccc} X & X & X & X & X & X & X & X \\ Y & Y & Y & Y & Y & Y & Y & Y \\ Y & Y & Y & Y & I & I & I & I \\ Y & Y & I & I & Y & Y & I & I \\ Y & I & Y & I & Y & I & Y & I \end{array} \tag{12}$$

Basically we are using the first generator to tell if the error is an X or a Z, and the last 4 generators to see where the error is. The last 4 generators give a version of the classical Hamming code (it is called the "extended" Hamming code because it adds an extra bit), and allow a binary search on the 8 bits to locate the error. Since the last 4 generators are made of just Ys they can identify the location of an X or Z error, but tell us nothing about Y errors, with which they commute. This code can detect a Y error, but we will know nothing about where it is. The distance of this code is 2: $Y \otimes Y$ (on any two qubits) will commute with all the generators of the stabilizer, and is outside the stabilizer. Any one-qubit Pauli operator anticommutes with some generator of the stabilizer (Y or Z on any qubit anticommutes with the first generator, X on any qubit anticommutes with the second generator).

b) Since the code corrects X_a and Z_a, by the error-correction conditions,

$$\langle \bar{j} | X_a Z_a | \bar{i} \rangle = C \delta_{ij}, \tag{13}$$

so the code can detect the error Y_a. Therefore the code has distance at least 2. Of course, if the distance were 3 or more, the code could actually correct all single-qubit Y errors, so in fact the distance of the code must be exactly 3.

Problem 5. Combining Stabilizer Codes

a) Clearly the $M_i \otimes I_{n_2}$ commute with each other, since the M_i's do, and the $I_{n_1} \otimes N_j$ commute with each other similarly. But $M_i \otimes I_{n_2}$ also commutes with $I_{n_1} \otimes N_j$, since they are each I where the other is nontrivial, so we have an Abelian group and a stabilizer code $S = S_1 \oplus S_2$ using $n = n_1 + n_2$ qubits. There are $n_1 - k_1$ M generators and $n_2 - k_2$ N generators, for a total of $r = n_1 + n_2 - (k_1 + k_2)$, so the number of encoded qubits is $k = n - r = k_1 + k_2$.

Suppose E is an error which is not detected by the code S_1. That is, E commutes with all generators of S_1, but is not in S_1. Then, clearly, $E \otimes I_{n_2}$ commutes with the generators of S, but is not in S, and is therefore not detected by S either. Conversely, if E is detected by S_1, either $E \in S_1$, in which case $E \otimes I_{n_2} \in S_1$, or E anticommutes with some generator M_i of S_1, in which case $E \otimes I_{n_2}$ also anticommutes with the generator $M_i \otimes I_{n_2}$ of S. Similarly, if F is or is not detected by S_2, then $I_{n_1} \otimes F$ is or is not detected by S as well.

The distance of S_1 is d_1 and the distance of S_2 is d_2. That means that there exist errors E and F of weight d_1 and d_2, respectively, which are not detected by S_1 and S_2. Thus, there is a Pauli operator of weight $d = \min(d_1, d_2)$ which is not detected by S. Conversely, any operator of weight less than d can be written $E \otimes F$, where E has weight less than d_1 (and is therefore detected by S_1) and F has weight less than d_2 (and is therefore detected S_2). We have four cases:

- E and F anticommute with some generator of S_1 and S_2, respectively. In this case, clearly the product is detected by both the M generators and the N generators, and is therefore detected by S.

- $E \in S_1$ and F anticommutes with a generator of S_2. In this case, $E \otimes F$ is detected by an N generator, so is still detected by S.
- $F \in S_2$ and E anticommutes with a generator of S_1. In this case, $E \otimes F$ anticommutes with an M generator, so is detected by S.
- $E \in S_1$ and $F \in S_2$. In this case, $E \otimes I_{n_2}$ and $I_{n_1} \otimes F$ are both in S, which is closed under multiplication, so $E \otimes F \in S$ as well.

In all 4 cases, S detects the error $E \otimes F$. That is, S detects all errors of weight less than d and fails to correct at least one error of weight d. The distance of the code is thus exactly $d = \min(d_1, d_2)$.

b) Now we have generators $M_i \otimes N_i$. These commute with each other, since $[M_i, M_j] = 0$ and $[N_i, N_j] = 0$, so we have a stabilizer code S. This code has $n = n_1 + n_2$ total qubits and $r = n_1 - k_1 = n_2 - k_2$ total generators. Therefore, $k = (n_1 + n_2) - (n_1 - k_1) = n_2 + k_1 = n_1 + k_2$.

As in part a, if E is not detected by S_1, then $E \otimes I_{n_2}$ is not detected by S (and similarly if F is not detected by S_2). If E anticommutes with something in S_1, then E anticommutes with something in S. However, if $E \in S_1$, it does not follow that $E \otimes I_{n_2} \in S$, and even if both E and F anticommute with generators of S_1 and S_2, it need not follow that $E \otimes F$ anticommutes with any generator of S: if E and F anticommute with exactly the same numbered generators of S_1 and S_2 (that is, they have the same error syndrome), then $E \otimes F$ will commute with $M_i \otimes N_i$.

Therefore, the distance of this code is at most $\min(d_1, d_2)$, but it could be much smaller. For instance, if S_1 has a generator of weight 1, then S would have distance 1: For instance, let S_1 and S_2 each be a 6-qubit code with the 5-qubit code generators on the first 5 qubits, plus an extra generator Z on the sixth qubit (i.e., it is the 5-qubit code with an extra $|0\rangle$ qubit appended). Then Z on the sixth qubit commutes with all generators, but is no longer in the stabilizer (although $Z_6 Z_{12}$ would be if we order the generators in the same way for both codes). The code thus has distance 1.

Even if we insist that S_1 and S_2 be nondegenerate, the distance of S could still be as low as 2: If there are single-qubit errors E and F with the same error syndrome for S_1 and S_2, respectively, then $E \otimes F$ will commute with every element of S, but will not be in S. The distance of S could not be 1 in this case, since any single-qubit operator would anticommute with either S_1 or S_2, but not both.

c) There is no reason M_i needs to commute with N_j, so in general, they need not define a QECC. If it happens that they do commute, then $k = n - (2n - k_1 - k_2) = k_1 + k_2 - n$. (If $n > k_1 + k_2$, it is impossible for all M_i to commute with all N_j.) The distance of such a code, if it exists, is at least $\max(d_1, d_2)$, as any operator that anticommutes with or is in S_1 or S_2 also anticommutes with or is in the combined stabilizer S. It could in some cases be larger, even much larger, as errors which fail to be detected by one code could be picked up instead by the other. The CSS construction is an example of this: Each code, by itself, only corrects bit or phase flip errors, and is unable to correct general errors, and therefore has distance 1 as a quantum code. Together, however, they have a much larger distance.

Problem 6. Ket Representation of Stabilizer Codes

a) The generators of the 5-qubit code are

$$\begin{array}{ccccc} X & Z & Z & X & I \\ I & X & Z & Z & X \\ X & I & X & Z & Z \\ Z & X & I & X & Z \end{array} \tag{14}$$

and we can take the encoded Z operator to be $\overline{Z} = Z \otimes Z \otimes Z \otimes Z \otimes Z$. The encoded $|\overline{0}\rangle$ state therefore has a stabilizer with 5 generators: the 4 generators of the whole code, plus $\overline{Z}$. The projector on $|\overline{0}\rangle$ is $\sum M$, where the sum is taken over M in the 5-generator stabilizer. Perhaps the easiest way to write down this state is find a state which is not annihilated by the projector (such as $|00000\rangle$) and just see what we get when the projector acts on it:

$$
\begin{aligned}
|\overline{0}\rangle &= \frac{1}{4}(|00000\rangle + |10010\rangle + |01001\rangle - |11011\rangle + |10100\rangle - |00110\rangle - |11101\rangle - |01111\rangle \\
&\quad +|01010\rangle - |11000\rangle - |00011\rangle - |10001\rangle - |11110\rangle - |01100\rangle - |10111\rangle + |00101\rangle). \quad (15)
\end{aligned}
$$

The terms of this sum are generated by all possible elements of the stabilizer acting on $|00000\rangle$.

Similarly, the encoded $|\overline{1}\rangle$ state is a +1-eigenstate of $-\overline{Z}$. We can use the same procedure as above, but now $|00000\rangle$ is annihilated by the projection operator, so let us instead start with $|11111\rangle$ and do the same procedure:

$$
\begin{aligned}
|\overline{1}\rangle &= \frac{1}{4}(|11111\rangle + |01101\rangle + |10110\rangle - |00100\rangle + |01011\rangle - |11001\rangle - |00010\rangle - |10000\rangle \\
&\quad +|10101\rangle - |00111\rangle - |11100\rangle - |01110\rangle - |00001\rangle - |10011\rangle - |01000\rangle + |11010\rangle). \quad (16)
\end{aligned}
$$

It is perhaps worth noting that every basis ket appears in either $|\overline{0}\rangle$ or $|\overline{1}\rangle$, but the code is still able to correct all single-qubit bit flips because of the varying phases.

b) Conceptually, we want to apply the procedure from the previous part backwards. The presence of the i phases implies that we will have Ys in the stabilizer. We first note that we are looking for 2 generators.

To find the first, match the first two kets and the last two kets in each codeword. Each pair is related by bit flips on the first two qubits and a phase. Thus, the possibilities are X or Y on the first qubit, X or Y on the second qubit, and I or Z on the third qubit. To find the correct phase, note that the first of each pair gets a phase i when it becomes the second, but the second of each pair gets a phase $-i$ when it becomes the first. The first kets of each pair all have even parity for the last two qubits, while the second kets of each pair have odd parity for the last two qubits. That suggests we pick X for the first qubit, Y for the second qubit, and Z for the third qubit, and indeed, these codewords are fixed under $X \otimes Y \otimes Z$.

To find the second generator, we follow a similar thought process, but instead match the first and third kets and the second and fourth kets in each codeword. This tells us we will need bit flips on the second and third qubits. We again get factors of i going to the right and factors of $-i$ going to the left, and we note that the first and second kets have even parity on the first two qubits, while the third and fourth kets have odd parity on the first two qubits. That suggests we should choose $Z \otimes Y \otimes X$ as the second generator, which indeed works.

If we had paired the first and fourth kets and the second and third kets, we would have found the operator $Y \otimes I \otimes Y$, which is also in the stabilizer (as the product of the two generators above).