Ph219a/CS219a

Solutions 5
Feb 5, 2014

Problem 5.1

(a) The derivation of the Gilbert-Varshamov (GV) bound for CSS codes follows closely the argument discussed in class for the general GV bound, except that we want to include the specific property that CSS codes correct **X** and **Z** errors separately. This requires us to deal with **X** and **Z** operators separately.

Let n be the block size of the code, and take n_X to be the number of **X**-type stabilizer generators, and n_Z to be the number of **Z**-type generators, with $n_X + n_Z < n$ for a nontrivial code space. For fixed n_X and n_Z, let $\mathscr{S}$ be the list of all stabilizers S with n_X **X**-type generators and n_Z **Z**-type generators[1]. Let $S_X \subseteq S$ be the set of operators generated by the **X**-type generators only, and $S_Z \subseteq S$ be the set generated by the **Z**-type generators only. For each stabilizer $S \in \mathscr{S}$, we first estimate the number of *dangerous* errors (errors that are not correctable by the code), i.e. Pauli operators that are in $S^\perp \backslash S$.

Now, an **X**-type error (call it E_X) is dangerous iff $E_X \notin S_X$ and E_X commutes with S_Z. The number of **X**-type operators that commute with S_Z is[2] 2^{n-n_Z}, and the number of operators in the set S_X is 2^{n_X} (since the set is generated by n_X **X**-type generators), so that the number of nontrivial **X**-type errors corresponding to a given $S \in \mathscr{S}$ is $2^{n-n_Z} - 2^{n_X}$. Similarly, a **Z**-type error (E_Z) is dangerous for $S \in \mathscr{S}$ iff $E_Z \ni S_Z$ and E_Z commutes with S_X. Again, the number of such **Z**- type errors is $2^{n-n_X} - 2^{n_Z}$. Note that the number of dangerous errors of the **X** and **Z** type are independent of the particular choice of S.
It follows from the Clifford group symmetry that each nontrivial (ie. $\neq I$) **X**-type operator E_X is dangerous for the same number (say N_X) of stabilizers $S \in \mathscr{S}$. Then, equating the total number of dangerous **X**-type errors for all $S \in \mathscr{S}$, to the total number of nontrivial **X**-type

[1] Note that this is not the same as listing all stabilizers with $n_X + n_Z$ generators. The generators must be either **X**-type or **Z**-type for it to be a CSS code.

[2] The total number of **X**-type operators is 2^n since each of the n positions in the block can be either **X** or **I**. These are subject to n_Z independent commutation constraints with the generators of S_Z, so this number reduces to 2^{n-n_Z}.

operators that are dangerous for the elements of $\mathscr{S}$, we have the following identity -

$$\begin{aligned} |\mathscr{S}| \times \left(\begin{array}{c} \text{No. of dangerous } \mathbf{X} \text{ type} \\ \text{errors for each } S \in \mathscr{S} \end{array} \right) &= \left(\begin{array}{c} \text{No. of nontrivial} \\ X\text{-errors} \end{array} \right) \\ &\quad \times \left(\begin{array}{c} \text{No. of } S \in \mathscr{S} \text{ that} \\ \text{each error is dangerous for} \end{array} \right) \\ \Rightarrow |\mathscr{S}|(2^{n-n_Z} - 2^{n_X}) &= (2^n - 1)N_X \\ \Rightarrow \frac{N_X}{|\mathscr{S}|} &= \frac{2^{n-n_Z} - 2^{n_X}}{2^n - 1} \end{aligned} \tag{1}$$

A similar argument for the **Z**-type errors gives the following identity

$$\frac{N_Z}{|\mathscr{S}|} = \frac{2^{n-n_X} - 2^{n_Z}}{2^n - 1} \tag{2}$$

where N_Z is the number of $S \in \mathscr{S}$ that each nontrivial **Z**-type operator is dangerous for.

Let $\mathcal{E}^X$ denote the set of **X** errors we want to correct and $\mathcal{E}^Z$ denote the set of **Z** errors we seek to correct. Then, we can define the sets

$$\begin{aligned} \mathcal{E}^{X(2)} &:= \{E_a^\dagger E_b : E_a, E_b \in \mathcal{E}^X\} \\ \mathcal{E}^{Z(2)} &:= \{E_a^\dagger E_b : E_a, E_b \in \mathcal{E}^Z\} \end{aligned} \tag{3}$$

For each nontrivial operator E in $\mathcal{E}^{X(2)}$ and $\mathcal{E}^{Z(2)}$, if we delete from $\mathscr{S}$ all the stabilizers for which E is dangerous, we would have deleted at most $(|\mathcal{E}^{X(2)}| - 1)N_X + (|\mathcal{E}^{Z(2)}| - 1)N_Z$ stabilizers. Therefore, in order for good codes to exist, the number of stabilizers in $\mathscr{S}$ must be at least

$$\begin{aligned} |\mathscr{S}| &> (|\mathcal{E}^{X(2)}| - 1)N_X + (|\mathcal{E}^{Z(2)}| - 1)N_Z \\ \Rightarrow 1 &> (|\mathcal{E}^{X(2)}| - 1)\frac{N_X}{|\mathscr{S}|} + (|\mathcal{E}^{Z(2)}| - 1)\frac{N_Z}{|\mathscr{S}|} \\ &= (|\mathcal{E}^{X(2)}| - 1)\frac{2^{n-n_Z} - 2^{n_X}}{2^n - 1} + (|\varepsilon^{Z(2)}| - 1)\frac{2^{n-n_X} - 2^{n_Z}}{2^n - 1} \end{aligned} \tag{4}$$

which gives the desired GV bound for CSS codes:

$$(|\varepsilon^{X(2)}| - 1)(2^{n-n_Z} - 2^{n_X}) + (|\mathcal{E}^{Z(2)}| - 1)(2^{n-n_X} - 2^{n_Z}) < 2^n - 1 \tag{5}$$

(b) Suppose we want a CSS code that corrects t_X **X**-type errors and t_Z **Z**-type errors. Then, the error sets are

$$\begin{aligned} \mathcal{E}^X &= \{\mathbf{X}\text{-type errors of wt. } \leq t_X\} \Rightarrow \mathcal{E}^{X(2)} = \{E_X : \mathrm{wt}(E_X) \leq 2t_X\} \\ \mathcal{E}^Z &= \{\mathbf{Z}\text{-type errors of wt. } \leq t_Z\} \Rightarrow \mathcal{E}^{Z(2)} = \{E_Z : \mathrm{wt}(E_Z) \leq 2t_Z\} \end{aligned} \tag{6}$$

Therefore,

$$\begin{aligned} |\mathcal{E}^{X(2)}| - 1 &= \sum_{j=1}^{2t_X} \left(\begin{array}{c} n \\ j \end{array} \right) \approx \left(\begin{array}{c} n \\ 2t_X \end{array} \right) \approx 2^{nH_2(2t_X/n)} \\ |\mathcal{E}^{Z(2)}| - 1 &= \sum_{j=1}^{2t_Z} \left(\begin{array}{c} n \\ j \end{array} \right) \approx \left(\begin{array}{c} n \\ 2t_Z \end{array} \right) \approx 2^{nH_2(2t_Z/n)} \end{aligned} \tag{7}$$

where we first simplify to the large n limit and then use Stirling's approximation. $H_2 = -x\log_2 x - (1-x)\log_2(1-x)$ is the binary entropy function.

Using Eqn.(32) in the GV bound derived in part(a), we get (in the limit as $n \to \infty$),

$$\begin{aligned} 2^{nH_2(2t_X/n)}(2^{n-n_Z} - 2^{n_X}) + 2^{nH_2(2t_Z/n)}(2^{n-n_X} - 2^{n_Z}) &\lesssim 2^n \\ \Rightarrow 2^{nH_2(2t_X/n)-n_Z} + 2^{nH_2(2t_Z/n)-n_X} &\lesssim 1 \end{aligned} \tag{8}$$

since $2^{n_X}/2^n \to 0$ and $2^{n_Z}/2^n \to 0$ as $n \to \infty$ (since $n_X, n_Z < n$).
Suppose , for some $\epsilon > 0$, we choose n_X and n_Z as follows

$$n_X = (1+\epsilon)nH_2\left(\frac{2t_X}{n}\right) \quad n_Z = (1+\epsilon)nH_2\left(\frac{2t_Z}{n}\right) \tag{9}$$

where we assume $2t_X/n$, $2t_Z/n$ small enough so that $H_2(.) < 1/2$, and ϵ is chosen small enough so that $n_X + n_Z < n$. In fact, we can take $\epsilon \to 0$ as $n \to \infty$. Then,

$$2^{n\epsilon H_2(2t_X/n)} + 2^{n\epsilon H_2(2t_Z/n)} \lesssim 1 \tag{10}$$

This is clearly satisfied for large enough values of n. Therefore,

$$\begin{aligned} k &= n - (n_X + n_Z) \\ \frac{k}{n} &= 1 - \frac{n_X}{n} - \frac{n_Z}{n} \end{aligned} \tag{11}$$

which implies, in the limit as $\epsilon \to 0$ and $n \to \infty$

$$\frac{k}{n} = 1 - H_2\left(\frac{2t_Z}{n}\right) - H_2\left(\frac{2t_X}{n}\right) \tag{12}$$

Problem 5.2

(a) Rather than prove that C_1 satisfies all the axioms of a vector space, we instead show that there is an equivalence between vectors in C_1 and the vectors of coefficients of a polynomial of degree m. More formally, given $\{x_0, x_1, ..., x_{n-1}\}$, there is an isomorphism between C_1 and the set $\mathbb{F}_p^{m+1}$ defined as

$$\mathbb{F}_p^{m+1} := \{\vec{a} = (a_0, a_1, ..., a_m)\,,\; a_i \in \mathbb{F}_p\} \tag{13}$$

We can define two operations on this set: addition mod p, and scalar multiplication over the field $\mathbb{F}_p$. Note that $\mathbb{F}_p^{m+1}$ is closed under addition mod p, has an identity element $\vec{0}$, and every element has an inverse: $(\vec{a})^{-1} = (a_0^{-1}, a_1^{-1}, ..., a_m^{-1})$, since $\mathbb{F}_p$ is a field. Other vector space axioms are also clearly satisfied, so $\mathbb{F}_p^{m+1}$ is a vector space over $\mathbb{F}_p$.
Now consider the following isomorphism $\phi : \mathbb{F}_p^{m+1} \to C_1$

$$\phi(\vec{a}) = (f_{\vec{a}}(x_{n-1}), f_{\vec{a}}(x_{n-2}), ..., f_{\vec{a}}(x_0)) \tag{14}$$

where $f_{\vec{a}}$ denotes the polynomial $a_0 + a_1 x + ... + a_m x^m$. ϕ is clearly closed under vector addition and scalar multiplication. Corresponding to every vector in $\mathbb{F}_p^{m+1}$, there exists a vector in C_1

and vice-versa. Since C_1 is isomorphic to a vector space $\mathbb{F}_p^{m+1}$, it is also a vector space.

(b) A nonzero polynomial of degree m has at most m zeros over $\mathbb{F}_p$, ie. for any $f(x)$ defining a vector in C_1 there exist at most m distinct x_i's for which $f(x_i) = 0$. Therefore, since $x_0, x_1, ..., x_{n-1}$ are all distinct, each vector $(f(x_{n-1}), ..., f(x_0)) \neq 0$ in C_1 has at most m zero entries, and thus has weight at least $n - m$. This implies $d_1 \geq n - m$.

(c) C_2 is isomorphic to $\mathbb{F}_p^m$ in a similar way as discussed in part(a). $\mathbb{F}_p^m$ is a vector space, using the arguments in (a) (with $m \to m-1$), so C_2 is also a vector space. It is clearly a subspace of C_1 since it consists of those (and only those) vectors in C_1 with f such that the mth degree coefficient is zero.

(d) We want a degree $m-1$ polynomial f such that $f(z_i) = y_i \; \forall \; i$, i.e. we want to fit an $(m-1)$-degree polynomial through m distinct points. This is just Lagrange interpolation - through any 2 points, there is a unique line, through any 3 points, there is a unique quadratic,..., through any m points, there is a unique polynomial of degree $m-1$. f can be constructed explicitly as

$$\begin{aligned} f(z) &= \frac{(z-z_2)(z-z_3)...(z-z_m)}{(z_1-z_2)(z_1-z_3)...(z_1-z_m)} y_1 + \frac{(z-z_1)(z-z_3)...(z-z_m)}{(z_2-z_1)(z_2-z_3)...(z_2-z_m)} y_2 \\ &\quad + ... + \frac{(z-z_1)(z-z_2)...(z-z_{m-1})}{(z_m-z_1)(z_m-z_2)...(z_m-z_{m-1})} y_m \\ &= \sum_{i=1}^{m} y_i \left(\prod_{j \neq i} \frac{z-z_j}{z_i-z_j} \right) \end{aligned} \tag{15}$$

Notice that only the ith term in the sum is nonzero (equals 1) for $f(z_i)$, and so $f(z_i) = y_i \; \forall \; i$ as desired. Furthermore, the denominators are well defined since the z_i's are all distinct, and f is of degree $m-1$ since each term in the sum consists of a product of exactly $m-1$ factors of $(z - z_j)$ in the numerator.

(e) The dual code of C_2 is defined as

$$C_2^{\perp} = \{ \vec{u} := (u_{n-1}, ..., u_1, u_0) \, , \; u_i \in \mathbb{F}_p, \; \vec{u}.\vec{v} = 0 \; \forall \; \vec{v} \; \in \; C_2 \} \tag{16}$$

Pick m different indices from 0 to $n-1$ and write this set as $\mathcal{I} := \{i_\alpha\}_{\alpha=1}^m, i_1 < i_2 < ... < i_m$. Consider, $\forall \; \vec{v} \in C_2$, the projection $\vec{v} = (f(x_{n-1}), ..., f(x_0)) \mapsto \vec{v}|_{\mathcal{I}} := (f(x_{i_m}), ..., f(x_{i_\alpha}), ..., f(x_{i_1})) \in \mathbb{F}_p^m$. Let $z_\alpha := x_{i_\alpha} \; \forall \; \alpha = 1, \ldots, m$. Recall that all the x_i's are distinct. Now, pick any vector $\vec{y} = (y_m, y_{m-1}, \ldots, y_1)$ in $\mathbb{F}_p^m$. From part (d), we know that there exists a polynomial of degree $m-1$, (call it f_y) such that

$$f_y(z_\alpha) = y_\alpha, \; \forall \; \alpha = 1, 2, \ldots, m \tag{17}$$

so $\vec{y}$ is exactly the projection of the vector $(f_y(x_{n-1}), \ldots, f_y(x_{i_m}), \ldots, f_y(x_{i_1}), \ldots, f_y(x_0)) \in C_2$ into $\mathbb{F}_p^m$. Since $\vec{y}$ was an arbitrary vector in $\mathbb{F}_p^m$, the projection of C_2 according to $\mathcal{I}$ into $\mathbb{F}_p^m$ is

the whole space $\mathbb{F}_p^m$. This is true for any choice of the projection indices $\mathcal{I}$.

Now, suppose a nonzero vector $\vec{u} \in C_2^\perp$ has at most m nonzero components. Then, $\vec{u}$ can be thought of as a vector $\vec{u}' \in \mathbb{F}_p^m$, by discarding $n-m$ zero components. Let us choose $\mathcal{I}$ such that it contains the indices of all the nonzero components of $\vec{u}$. From our argument above, there exists a nonzero $\vec{v} \in C_2$ such that $\vec{v}|_{\mathcal{I}} = \vec{u}'$, and so $\vec{v}|_{\mathcal{I}}.\vec{u}' = \vec{v}.\vec{u} \neq 0$. Hence, there cannot be such a vector $\vec{u}$ in $C_2^\perp$, i.e. every nonzero vector in $C_2^\perp$ must have at least $m+1$ nonzero components, which implies $d_2 \geq m+1$.

(f) Recall the definition of a coset: $\vec{u}, \vec{v} \in C_1$ belong to the same coset of C_2 iff $\vec{u} - \vec{v} \in C_2$. C_1 consists of vectors constructed from degree m polynomials, so $\vec{u}, \vec{v} \in C_1$ can differ by a vector in C_2, which is constructed from degree $m-1$ polynomials, iff $\vec{u}$ and $\vec{v}$ have the same coefficient for x_m. Since the coefficients are chosen from $\mathbb{F}_p$, there are exactly p distinct possibilities for the coefficient of x_m, and hence there are exactly p distinct cosets. Alternatively, you could have recalled Lagrange's theorem: the number of distinct cosets of C_2 in C_1 is $|C_1|/|C_2| = |\mathbb{F}_p^{m+1}|/|\mathbb{F}_p^m| = p^{m+1}/p^m = p$. Therefore, the number of encoded qu$p$its is $\log_p(\text{dim. of code space}) = \log_p(\text{number of distinct cosets}) = \log_p p = 1$.

(g) We want to correct $t = (d-1)/2$ errors , and for a CSS code, $d = min(d_1, d_2)$, so we require $d_1 \geq 2t+1$ and $d_2 \geq 2t+1$. From parts (b) and (e), we know $d_1 \geq n-m$ and $d_2 \geq m+1$, hence it suffices to impose

$$\begin{aligned} d_1 &\geq n-m \geq 2t+1 \Rightarrow n \geq m+2t+1 \\ d_2 &\geq m+1 \geq 2t+1 \Rightarrow m \geq 2t \end{aligned} \tag{18}$$

Suppose we take $m = 2t$, then $n \geq 4t+1$, so we can also take $n = 4t+1$ as required by the question. Since $n \leq p$, such a code can only be constructed for $p \geq 4t+1$ and p prime.

Problem 5.3

(a) Given the stabilizer generators $M_X = X^{nr_1}$ and $M_Z = Z^{nr_2}$ (where n, r_1 and r_2 are positive integers), for any $|j\rangle$, we have,

$$\begin{aligned} [M_X, M_Z]\,|j\rangle &= (X^{nr_1}Z^{nr_2} - Z^{nr_2}X^{nr_1})\,|j\rangle \\ &= X^{nr_1}(w^j)^{nr_2}\,|j\rangle - Z^{nr_2}\,|j + nr_1(\text{mod}d)\rangle \\ &= [w^{jnr_2} - (w^{j+nr_1})^{nr_2}]\,|j + nr_1(\text{mod}d)\rangle \\ &= w^{jnr_2}[1 - (w^{nr_1r_2})^n]\,|j + nr_1(\text{mod}d)\rangle \\ &= w^{jnr_2}[1 - (w^d)^n]\,|j + nr_1(\text{mod}d)\rangle \ (\text{since} d = nr_1r_2) \\ &= 0 \ \forall\ j = 0, 1, ..., d-1 \end{aligned} \tag{19}$$

since $w^d = e^{2\pi i} = 1$. Thus, $[M_X, M_Z]\,|j\rangle = 0$ for all values of j, verifying that $[M_X, M_Z] = 0$.

(b) Using $ZX = wXZ$, we can work out commutation relations between powers of X and Z -

$$Z^k X^l = wZ^{k-1} XZX^l = w^k XZ^k X^{l-1} = w^{kl} X^l Z^k \tag{20}$$

for any pair of positive integers k and l. Therefore,

$$\begin{aligned} M_X(X^a Z^b) &= X^a X^{nr_1} Z^b = \bar{w}^{bnr_1}(X^a Z^b) M_X \\ M_Z(X^a Z^b) &= Z^{nr_1} X^a Z^b = w^{anr_2}(X^a Z^b) M_Z \end{aligned} \tag{21}$$

(c) We want the generators of normalizer group $S^\perp$, which consists of all Pauli operators $X^a Z^b$ that commute with M_X and M_Z. From our answers in part(b), we know that $X^a Z^b$ commutes with both M_X and M_Z iff a and b satisfy:

$$\begin{aligned} w^{anr_1} &= 1 \Rightarrow \exp\left(2\pi i \frac{anr_2}{d}\right) = \exp\left(2\pi i \frac{a}{r_1}\right) = 1 \\ \Rightarrow a &= kr_1, \;\; k \in \mathbb{Z} \end{aligned} \tag{22}$$

and

$$\begin{aligned} \bar{w}^{bnr_1} &= 1 \Rightarrow \exp\left(-2\pi i \frac{bnr_1}{d}\right) = \exp\left(-2\pi i \frac{b}{r_2}\right) = 1 \\ \Rightarrow b &= lr_2, \;\; l \in \mathbb{Z} \end{aligned} \tag{23}$$

Therefore, $S^\perp = \{X^{kr_1} Z^{lr_2} : \; k, l \in \mathbb{Z}\}$, and the two generators are $\tilde{X} := X^{r_1}$ and $\tilde{Z} := Z^{r_2}$. They satisfy

$$\tilde{Z}\tilde{X} = Z^{r_2} X^{r_1} = w^{r_1 r_2} X^{r_1} Z^{r_2} = \tilde{w}\tilde{X}\tilde{Z} \tag{24}$$

with $\tilde{w} := w^{r_1 r_2} = \exp(2\pi i/n)$. Note that this is the same as the commutation relation between the generalized Paulis X and Z, (eqn. (12) of HW 7) with $d \to n$, $X \to \tilde{X}$, $Z \to \tilde{Z}$ and $w \to \tilde{w}$. Since the operators in the normalizer group act as the logical operations on the codespace, the commutation relation leads us to suspect that there must be one encoded qu$n$it. We verify this by explicitly constructing the code subspace.

Recall that the codespace is the simultaneous $+1$ eigenspace of the stabilizer generators. Since the action of M_X on a computational basis state $|j\rangle$ is to shift it by nr_1, the $+1$ eigenstates of M_X must be uniform superpositions of the states from the cosets

$$\mathcal{C}_j := \{|j + mnr_1 \,(\mathrm{mod}\, d)\rangle \;,\; m = 0, 1, ...r_2 - 1\} \;\forall\; j = 0, 1, ..., nr_1 - 1 \tag{25}$$

Thus the $+1$ eigenspace of M_X is spanned by the orthonormal states

$$|\psi_j\rangle = \frac{1}{\sqrt{\mathcal{C}_j}} \sum_{m=0}^{r_2-1} |j + mnr_1\rangle = \frac{1}{\sqrt{r_2}} \sum_{m=0}^{r_2-1} |j + mnr_1\rangle \tag{26}$$

Within this eigenspace, to find the $+1$ eigenstates of M_Z, we look at how M_Z acts on $|\psi_j\rangle$:

$$M_Z |\psi_j\rangle = \frac{1}{\sqrt{r_2}} \sum_{m=0}^{r_2-1} (w^{j+mnr_1})^{nr_2} |j + mnr_1\rangle = w^{jnr_2} |\psi_j\rangle \tag{27}$$

Thus $|\psi_j\rangle$ is a $+1$ eigenstate of M_Z iff $w^{jnr_2} = 1$, ie. $j = kr_1$, where k takes on integer values. Since $j = 0, 1, ...nr_1 - 1$ there are exactly n values of j that satisfy this condition. Thus, the common $+1$ eigenspace of M_X and M_Z is spanned by n vectors $|\psi_j\rangle$, showing that the codespace is indeed n-dimensional. In other words, this code has one encoded qu$n$it.

(d) The distance d_X of the code for X errors is given by the minimum weight of the X-type operators in $S^\perp \backslash S$. Therefore, $d_X = \text{wt}(\tilde{X}) = r_1$. This code can therefore correct an amplitude shift (X errors) of up to $|a| = \frac{d_X - 1}{2} = \frac{r_1 - 1}{2}$. Similarly, $d_Z = \text{wt}(\tilde{Z}) = r_2$, so that the largest phase shift (Z error) this code can correct is $|b| = \frac{d_Z - 1}{2} = \frac{r_2 - 1}{2}$.

Problem 5.4

(a) We can recall the action of the Hadamard ($\mathbf{H} = \mathbf{H}^{-1}$) and the phase gates ($\mathbf{P}$) on the Pauli operators from an earlier Hw problem, where we had shown

$$\begin{aligned} \mathbf{HXH}^{-1} &= \mathbf{Z}\ ,\ \mathbf{HYH}^{-1} = -\mathbf{Y}\ ,\ \mathbf{HZH}^{-1} = \mathbf{X} \quad (28) \\ \mathbf{PXP}^{-1} &= \mathbf{Y}\ ,\ \mathbf{PYP}^{-1} = -\mathbf{X}\ ,\ \mathbf{PZP}^{-1} = \mathbf{Z} \quad (29) \end{aligned}$$

From the definition of the normalizer group, this immediately shows that $\mathbf{H}, \mathbf{P} \in \mathcal{C}_1$. To show $\Lambda(\mathbf{X}) \in \mathcal{C}_2$, it suffices to see how the CNOT gate acts by conjugation on the generators of $\mathcal{P}_2$: $\{\mathbf{XI}, \mathbf{IX}, \mathbf{ZI}, \mathbf{IZ}\}$. Using $\Lambda(\mathbf{X}) = |0\rangle\langle 0| \otimes \mathbf{I} + |1\rangle\langle 1| \otimes \mathbf{X}$ and noting that $(\Lambda(\mathbf{X}))^{-1} = \Lambda(\mathbf{X})$, we have,

$$\begin{aligned} \Lambda(\mathbf{X})(\mathbf{XI})\Lambda(\mathbf{X}) &= \mathbf{XX} \in \mathcal{P}_2\ ,\ \Lambda(\mathbf{X})(\mathbf{IX})\Lambda(\mathbf{X}) = \mathbf{IX} \in \mathcal{P}_2 \\ \Lambda(\mathbf{X})(\mathbf{ZI})\Lambda(\mathbf{X}) &= \mathbf{ZI} \in \mathcal{P}_2\ ,\ \Lambda(\mathbf{X})(\mathbf{IZ})\Lambda(\mathbf{X}) = \mathbf{ZZ} \in \mathcal{P}_2 \quad (30) \end{aligned}$$

Since all the generators of $\mathcal{P}_2$ are mapped to elements of the group itself under the action of $\Lambda(\mathbf{X})$, we see that $\Lambda(\mathbf{X}) \in \mathcal{C}_2$.

(b) We know that under the action of $\mathbf{H}$,

$$\begin{aligned} &\mathbf{X} \to \mathbf{Z}\ ,\ \mathbf{Z} \to \mathbf{X}\ ,\ \mathbf{Y} \to -\mathbf{Y} \\ \Rightarrow\ &\mathbf{XYZ} \to -\mathbf{ZYX} = -\mathbf{YXZ} = \mathbf{XYZ} \quad (31) \end{aligned}$$

Similarly under the action of the phase gate $\mathbf{P}$,

$$\begin{aligned} &\mathbf{X} \to \mathbf{Y}\ ,\ \mathbf{Y} \to -\mathbf{X}\ ,\ \mathbf{Z} \to \mathbf{Z} \\ \Rightarrow\ &\mathbf{XYZ} \to -\mathbf{YXZ} = \mathbf{XYZ} \quad (32) \end{aligned}$$

Thus, $\mathbf{H}$ and $\mathbf{P}$ are permutations of $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ that preserve the inner product $\mathbf{XYZ} = i\mathbf{I}$. Furthermore, (ignoring phases) $\mathbf{H}$ is a permutation of $(\mathbf{X}, \mathbf{Z})$, and $\mathbf{P}$ permutes $(\mathbf{X}, \mathbf{Y})$, which together generate all 3! permutations of 3 objects. When we keep track of phase we see that $\mathbf{H}$ and $\mathbf{P}$ preserve $\mathbf{XYZ} = i\mathbf{I}$. Note that since $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ are Hermitian, and conjugation

by a unitary operator preserves Hermiticity, no imaginary phases can be reached by application of the Clifford group. After each permutation, we therefore only need to check we can additionally impliment any phases which preserve $\mathbf{XYZ} = i\mathbf{I}$, which corresponds to making one of the following four replacements: $\{\mathbf{X},\mathbf{Y},\mathbf{Z}\} \mapsto \{\mathbf{X},\mathbf{Y},\mathbf{Z}\}$, $\{\mathbf{X},\mathbf{Y},\mathbf{Z}\} \mapsto \{\mathbf{X},-\mathbf{Y},-\mathbf{Z}\}$, $\{\mathbf{X},\mathbf{Y},\mathbf{Z}\} \mapsto \{-\mathbf{X},\mathbf{Y},-\mathbf{Z}\}$, $\{\mathbf{X},\mathbf{Y},\mathbf{Z}\} \mapsto \{-\mathbf{X},-\mathbf{Y},\mathbf{Z}\}$. The second and fourth of these replacements can be made with $\mathbf{HP}^2\mathbf{H}$ and $\mathbf{P}^2$ respectively, which can be used to generate the other two. Thus there are 24 elements of the Clifford group on one qubit, all of which can be implimented by $\mathbf{H}$ and $\mathbf{P}$.

(c) Using eqns.(1) and (2), it follows that $\Lambda(\mathbf{Z}) = \mathbf{H}_2\Lambda(\mathbf{X})\mathbf{H}_2$ (where $\mathbf{H}_2$ denotes a Hadamard gate acting on the target qubit), and $\Lambda(\mathbf{Y}) = \mathbf{P}_2\Lambda(\mathbf{X})\mathbf{P}_2$.
Now, consider $\Lambda(\sigma)\mathbf{Z}_1\Lambda(\sigma)$. When the control qubit is $|0\rangle$, the target is left unchanged by both the $\Lambda(\sigma)$ gates. When the control qubit is $|1\rangle$, $\Lambda(\sigma)\mathbf{Z}_1\Lambda(\sigma)\,|1\rangle\,|.\rangle = \Lambda(\sigma)\mathbf{Z}_1\sigma_2\,|1\rangle\,|.\rangle = -(\sigma_2)^2\,|1\rangle\,|.\rangle = \mathbf{Z}_1\,|1\rangle\,|.\rangle$, since $\sigma^2 = \mathbf{I}$. Similarly, we can analyze the action of $\Lambda(\sigma)\mathbf{X}_1\Lambda(\sigma)$ as follows:

$$\begin{aligned}\Lambda(\sigma)\mathbf{X}_1\Lambda(\sigma)\,|0\rangle\,|.\rangle &= \Lambda(\sigma)\,|1\rangle\,|.\rangle = \mathbf{X}_1\sigma_2\,|0\rangle\,|.\rangle \\ \Lambda(\sigma)\mathbf{X}_1\Lambda(\sigma)\,|1\rangle\,|.\rangle &= \Lambda(\sigma)\mathbf{X}_1\sigma_2\,|1\rangle\,|.\rangle = \Lambda(\sigma)\sigma_2\,|0\rangle\,|.\rangle = \mathbf{X}_1\sigma_2\,|1\rangle\,|.\rangle \end{aligned} \tag{33}$$

Thus we have shown that

$$\Lambda(\sigma)\mathbf{Z}_1\Lambda(\sigma) = \mathbf{Z}_1 \quad \Lambda(\sigma)\mathbf{X}_1\Lambda(\sigma) = \mathbf{X}_1\sigma_2 \tag{34}$$

(d) First, note that $\mathbf{UX}_1\mathbf{U}^{-1}$ anticommutes with $\mathbf{UZ}_1\mathbf{U}^{-1}$ -

$$(\mathbf{UX}_1\mathbf{U}^{-1})(\mathbf{UZ}_1\mathbf{U}^{-1}) = \mathbf{UX}_1\mathbf{Z}_1\mathbf{U}^{-1} = -\mathbf{UZ}_1\mathbf{X}_1\mathbf{U}^{-1} = (\mathbf{UZ}_1\mathbf{U}^{-1})(\mathbf{UX}_1\mathbf{U}^{-1}) \tag{35}$$

implying that $\mathbf{UX}_1\mathbf{U}^{-1}$ and $\mathbf{UZ}_1\mathbf{U}^{-1}$ must differ in their action on atleast one of the $n+1$ qubits. Suppose they differ in their action on the first qubit, we can write

$$\mathbf{U} : \mathbf{X}_1 \to \mathbf{M}_1\mathbf{M} \ , \ \mathbf{Z}_1 \to \mathbf{N}_1\mathbf{N} \tag{36}$$

where $\mathbf{M}_1 \neq \mathbf{N}_1 \in \mathcal{P}_1$. In this case, we can simply choose $\mathbf{W}$ to be an element of $\mathcal{C}_1$ that acts such that $\mathbf{W} : \mathbf{M}_1 \to \mathbf{X}_1$, $\mathbf{N}_1 \to \mathbf{Z}_1$.
On the other hand, suppose $\mathbf{M}_1 = \mathbf{N}_1$. Since $\mathbf{UX}_1\mathbf{U}^{-1}$ and $\mathbf{UZ}_1\mathbf{U}^{-1}$ anti-commute, they have to act differently on atleast one qubit. Suppose they act differently on the kth qubit, ie. $\mathbf{M}$ and $\mathbf{N}$ are such that $\mathbf{M}_k \neq \mathbf{N}_k$. Now we choose $\mathbf{W}$ as follows: we choose an operator in $\mathcal{C}_1$ to act on the kth position such that $\mathbf{W}_k : \mathbf{M}_k \to \mathbf{X}_k$, $\mathbf{N}_k \to \mathbf{Z}_k$. Then, we perform a CNOT with the kth qubit as the control and the first qubit as the target so that the first qubit is left unchanged when the operator in the kth position is $\mathbf{Z}$, or gets transformed by an $\mathbf{X}$ operator when the kth position is $\mathbf{X}$. Now that we can distinguish between the two cases, we can transform the first qubit with an appropriate operator in $\mathcal{C}_1$ so that $\mathbf{M}_1 \to \mathbf{X}_1$ and $\mathbf{N}_1 \to \mathbf{Z}_1$.

(e) Analyzing the action of $\mathbf{V}$ on $\mathbf{X}_1$, we have,

$$\begin{aligned}
\mathbf{V}:\mathbf{X}_1 &\equiv \Lambda(\mathbf{M})\mathbf{H}_1\Lambda(\mathbf{N})\mathbf{H}_1\mathbf{W}\mathbf{U}:\mathbf{X}_1 \\
&\hookrightarrow \Lambda(\mathbf{M})\mathbf{H}_1\Lambda(\mathbf{N})\mathbf{H}_1:\mathbf{X}_1\mathbf{M} \quad (\text{since}\mathbf{W}\mathbf{U}:\mathbf{X}_1 \rightarrow \mathbf{X_1}\mathbf{M}) \\
&\hookrightarrow \Lambda(\mathbf{M})\mathbf{H}_1\Lambda(\mathbf{N}):\mathbf{Z}_1\mathbf{M} \\
&\hookrightarrow \Lambda(\mathbf{M})\mathbf{H}_1:\mathbf{Z}_1\mathbf{M} \\
&\hookrightarrow \Lambda(\mathbf{M}):\mathbf{X}_1\mathbf{M} \\
&\hookrightarrow \mathbf{X}_1\mathbf{M}^2=\mathbf{X}_1 \qquad (37)
\end{aligned}$$

Where we used the fact that $\Lambda(\mathbf{N})\mathbf{Z}_1\mathbf{M}\Lambda(\mathbf{N}) = \mathbf{N})\mathbf{Z}_1\Lambda(\mathbf{N})\mathbf{M} = \mathbf{Z}_1\mathbf{M}$ since $\mathbf{M}$ and $\mathbf{N}$ commute and $\Lambda(\mathbf{N})$ can be written as the consecutive action of $\Lambda(\sigma)$, with the control on the first qubit and σ a single qubit pauli as studied in part (c). Similarly, we can show $\mathbf{V}:\mathbf{Z}_1 \rightarrow \mathbf{Z}_1$.

(f) We wish to show that $\mathbf{V}\mathbf{P}\mathbf{V}^\dagger \in \mathcal{P}_n$ for any $\mathbf{P} \in \mathcal{P}_n$ (has the identity on the first qubit but any pauli on the other n qubits). If this is true then $\mathbf{V} \in \mathcal{C}_n$. Note that any pauli $\mathbf{Q} \in \mathcal{P}_{n+1}$ is also in $\mathcal{P}_n$ if and only if $X_1\mathbf{Q} = \mathbf{Q}X_1$ and $Z_1\mathbf{Q} = \mathbf{Q}Z_1$. Also note that $\mathbf{V}$, $\mathbf{V}^\dagger$ and $\mathbf{P}$ commute with both X_1 and Z_1 since $\mathbf{V}X_1\mathbf{V}^\dagger = X_1$, $\mathbf{V}Z_1\mathbf{V}^\dagger = Z_1$ and $\mathbf{P}$ has no support on the first qubit. Therefore $\mathbf{V}\mathbf{P}\mathbf{V}^\dagger$ commutes with both X_1 and Z_1, implying $\mathbf{V}\mathbf{P}\mathbf{V}^\dagger \in \mathcal{P}_n$ as desired.
Since $\mathbf{V} = \Lambda(\mathbf{M})\mathbf{H}_1\Lambda(\mathbf{N})\mathbf{H}_1\mathbf{W}\mathbf{U}$, and $\Lambda(\mathbf{M}), \Lambda(\mathbf{N})$ and $\mathbf{W}$ can all be constructed from $\mathbf{H}$, $\mathbf{P}$ and $\Lambda(\mathbf{X})$, it follows that $\mathbf{U}$ can also be constructed from $\mathbf{H}$, $\mathbf{P}$ and $\Lambda(\mathbf{X})$.