

# Project Outline: Hybrid Codes for Hybrid Secret Sharing Schemes

Andrew Nemec

August 2025

## 1 Hybrid Quantum-Classical Codes

Imagine we have a quantum channel, and we want to send both quantum and classical information across it. One approach is to encode the quantum and classical information in two separate codes, and then transmit using a time-sharing approach. Another option is to encode quantum and classical information together in a *quantum-classical hybrid code*, often just called a hybrid code.

The way to think about this is as a collection  $\{\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_{M-1}\}$  of orthogonal quantum codes, and then if we want to send the quantum state  $|\psi\rangle$  and the classical message  $a$ , we encode  $|\psi\rangle$  into the quantum code  $\mathcal{C}_a$  and transmit the encoded state. That is, we index our possible quantum codes using the classical message.

If our original code is a stabilizer code, the subspaces orthogonal to the code  $\mathcal{C}$  are those that have been transformed by a destabilizer  $T_i$ , that is some Pauli element that commutes with all the logical operators but anticommutes with at least one stabilizer generator. We call these  $T_i$  the *transition operators*, or the *classical logical operators*. We can write these orthogonal subspaces as

$$\mathcal{C}_i = T_i \mathcal{C} = \{T_i |\psi\rangle \mid |\psi\rangle \in \mathcal{C}\}.$$

Note that we can always choose  $T_0 = I$  so that  $\mathcal{C} = \mathcal{C}_0$  is one of the codes. If the  $T_i$  form a group  $\mathcal{T}$ , then  $M = 2^m$  and we call the code a hybrid stabilizer code.

An equivalent way to define hybrid stabilizer codes is in terms of the stabilizer group. Pick a set of generators for the transition operators:

$$\mathcal{T} = \langle T_1, T_2, \dots, T_m \rangle,$$

so that an arbitrary transition operator is  $T_a = T_1^{a_1} T_2^{a_2} \dots T_m^{a_m}$ . Then we can pick a set of stabilizer generators

$$\mathcal{S} = \langle S_1, S_2, \dots, S_{n-k} \rangle,$$

where  $m \leq n - k$ , such that  $S_i$  and  $T_i$  anticommute for  $1 \leq i \leq m$ , and  $S_i$  and  $T_j$  commute for  $i \neq j$ .

Recall that the projector  $P$  onto a stabilizer code  $\mathcal{C}$  can be written in terms of both an orthonormal basis for the code  $\{|c_i\rangle\}$  as well as in terms of its stabilizer group  $\mathcal{S}$ :

$$P = \sum_{i=1}^{2^k} |c_i\rangle\langle c_i| = \frac{1}{|\mathcal{S}|} \sum_{S \in \mathcal{S}} S = \prod_{i=1}^{n-k} \frac{I + S_i}{2}.$$

The projector  $P_i$  onto the orthogonal space  $\mathcal{C}_i$  is given by

$$P_a = \sum_{i=1}^{2^k} T_a |c_i\rangle\langle c_i| T_a^\dagger = T_a P_0 T_a^\dagger = T_a \left( \prod_{i=1}^{n-k} \frac{I + S_i}{2} \right) T_a^\dagger = \prod_{i=1}^{n-k} \frac{I + (-1)^{a_i} S_i}{2},$$

so each orthogonal code has the same stabilizers, but the phases on some have changed. In particular, note that the classical information is encoded in these phases.

One way to create a hybrid code is to start with an  $\llbracket n, k + m, d \rrbracket$  and use  $m$  logical qubits to transmit classical information, giving us an  $\llbracket n, k : m, d \rrbracket$  hybrid code. This is easy, as it guarantees the minimum distance  $d$  does not decrease. To do this, we treat the logical  $Z$  operators on our now logical classical bits as stabilizers in which we encode the classical information, and the corresponding logical  $X$  operators as the translation operators. However, the interesting cases are when there exists an  $\llbracket n, k : m, d \rrbracket$  hybrid code but no  $\llbracket n, k + m, d \rrbracket$  quantum code.

The set of errors that can be corrected by a hybrid code is completely determined by the hybrid Knill-Laflamme condition.

**Theorem 1.1 (Hybrid Knill-Laflamme Condition)** *A hybrid code with orthogonal projectors  $P_0, P_1, \dots, P_{M-1}$  can correct all errors  $E_i, E_j \in \mathcal{E}$  if and only if*

$$P_a E_i^\dagger E_j P_b = \lambda_{i,j}^{(a)} \delta_{a,b} P_a.$$

Note that in the hybrid case the  $\lambda_{i,j}^{(a)}$  depend on the classical information.

## 2 Codeword Stabilized Codes

Codeword stabilized quantum codes are constructed in a similar way to hybrid codes, in that we start with a stabilizer code (usually just a single stabilizer state, i.e., a  $\llbracket n, 0, d \rrbracket$  code) and then expand the code using codeword operators  $T_i$  that act identically to the translation operators of hybrid codes. The key difference is that for codeword stabilized codes we don't require the operators to form a group, and the code they define satisfies the normal Knill-Laflamme condition for quantum codes.

**Theorem 2.1** *A quantum code with projector  $P$  can correct all errors  $E_i, E_j \in \mathcal{E}$  if and only if*

$$P E_i^\dagger E_j P = \lambda_{i,j} P.$$

Note here that  $\lambda_{i,j}$  depend only on  $E_i, E_j$  and not on any particular encoded state, in contrast to hybrid codes, where  $\lambda_{i,j}^{(a)}$  depends on the classical message  $a$ . Also, since our codeword operators do not necessarily form a group, the dimension of our code is not necessarily a power of two and therefore not a stabilizer code. Therefore, you might see the notation  $((n, K, d))$  denoting that the code is not a stabilizer code and is  $K$ -dimensional. For stabilizer codes, we have  $K = 2^k$ , so the parameters  $[[n, k, d]]$  and  $((n, 2^k, d))$  would be equivalent.

### 3 Project Goals

The goals of this project are to:

- Determine if there are “good” hybrid codes that outperform stabilizer codes over the amplitude damping and related channels (phase damping, generalized amplitude damping, etc.). You should look to the exhaustive searches for hybrid codes and genetic searches for stabilizer codes for inspiration.
- Develop upper bounds on the performance of hybrid codes over amplitude damping channels using linear programming (I will help you with this later).