

M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem

TDK DOLGOZAT

Kvantumos hibajavítás a felületi kódban

Koherens hibák és kiolvasási hibák vizsgálata

Készítette:

Márton Áron

Fizika BSc 3. évfolyam
BME Természettudományi kar

Konzulens:

Dr. Asbóth János

Egyetemi docens
BME Elméleti Fizika Tanszék

2021. október 27.

Kivonat

A gyakorlatban jól használható kvantumszámítógépekhez elengedhetetlen a kvantum hibajavító kódok használata, amelyekkel meg lehet védeni a kvantumbiteket a környezeti zajtól. A legígéretesebbnek tartott kvantum kód az úgynevezett felületi kód, ami véletlenszerű bit-flip, illetve fázishibákkal szemben is bizonyítottan hatékonyan védi a kvantuminformációt. Az úgynevezett koherens hibák – ahol a környezet valamilyen kis unitér transzformációval hat a kvantumbitekerekre – hatása viszont nehezen modellezhető a felületi kódban numerikusan.

A koherens hibák egy osztályára Bravyi és szerzőtársai [5] a Fermionikus Lineáris Optika elméleti eszközeit használó szimulációs eljárást dolgoztak ki. Munkájukban azonban nem vették figyelembe a kiolvasási hibák hatását, amelyek a mai kvantumszámítógép-prototípusokban nagyon jelentősek.

Dolgozatomban a felületi kód hibajavítási hatékonyságát vizsgáltam a koherens hibák és kiolvasási hibák egyidejű fellépése esetén, Bravyiék szimulációs eljárását kiterjesztve. Meghatároztam a koherens és kiolvasási hibák együttes hibaküszöbét: ez közelítően megegyezik az inkoherens hibák és kiolvasási hibák együttes hibaküszöbével (ez utóbbi az irodalomból ismert). Ugyanakkor a szimulációkból kiderült, hogy a küszöb alatt, azaz kellően kis mértékű környezeti zaj mellett, a koherens hibák nagyobb logikai hibákhoz vezetnek, mint az inkoherens hibák, a kiolvasási hibák jelenlétében is.

A dolgozatomban bemutatott eredmények biztatóak lehetnek a felületi kód megvalósítását célzó projektek számára is.

Tartalomjegyzék

1. Bevezetés	3
2. Kvantumos hibajavító kódok	4
2.1. Stabilizátorformalizmus	4
2.2. Kvantumos hibajavítás	6
2.3. Felületi kód	7
2.4. A felületi kód kvantumos logikai áramköri megvalósítása	9
2.5. A minimális súlyú kapcsolódások elvén működő (MWPM) de- kódoló	10
2.6. Hibás szindrómák dekódolása	13
2.7. Hibaküszöb	15
3. Koherens hibák	18
3.1. Kvantumos hibacsatornák	18
3.2. Általános hibák a felületi kódban	20
3.3. Koherens forgatások	21
3.4. Kiolvasási hibák és koherens forgatások	23
4. A szimulációs módszer	24
4.1. Tiszta Gaussi állapotok	24
4.2. Fermionikus Lineáris Optika	27
4.3. A C4 kód	28
4.4. A felületi kód Majorana hálózata	30
4.5. A végállapot-eloszlás mintavételezése	33
4.6. A szimuláció erőforrásigénye	36
5. Eredmények	37
5.1. Koherens hibák, tökéletes kiolvasás	37
5.2. A koherens hibák és a kiolvasási hibák együttes hatása	42
6. Összefoglalás	46
7. Köszönetnyilvánítás	47
A. Függelék: A logikai forgatási szög meghatározása kiolvasási hibák jelenlétében	51
B. Függelék: A Fermionikus Lineáris Optika lépései	52
C. Függelék: A Felületi kód szimulálási idejének lerövidítése	53

1. Bevezetés

Az elmúlt évtizedekben hatalmas előrelépések történtek a kvantumszámítógépek megvalósításának irányában [2, 28], már több mint 50 kvantumbitet tartalmazó prototípusokat sikerült megvalósítani a kutatóknak. A gyakorlatban is jól használható kvantumalgoritmusok megvalósításához azonban ennél jóval több kvantumbitre van szükség.

Elméleti szempontból az egyik legfontosabb probléma nem a kvantumbitek száma, hanem a kvantumkapuk pontossága, hiszen ez is elengedhetetlen feltétele a jól használható kvantumszámítógépeknek. A megfelelő, $10^{-10} - 10^{-15}$ -es pontosságú kvantumszámítás a jelenlegi kvantumbitekkel elérhetetlennek tűnik, ennek oka, hogy a fizikailag megvalósított különböző típusú kvantumbitek elég érzékenyek a környezeti zajokra.

A probléma megoldására egy széles körben elterjedt javaslat az úgynevezett kvantum hibajavító kódok és ezen belül is a felületi kód (surface code) [11, 18] használata. Itt több fizikai kvantumbitben tároljuk el azt az információt, amit elég lenne egy kvantumbitben tárolnunk, azonban így bizonyos mértékig ki tudjuk javítani a kvantumszámítógépben fellépő hibákat.

A dolgozatom első fejezetében bemutatok egy a kvantum hibajavító kódok leírására alkalmas általános formalizmust, majd rátérek a felületi kód részletes tárgyalására.

A felületi kód hatékonyan véd a véletlenszerű Pauli-hibákkal szemben, azonban a valóságban másfajta hibák is érhetik a qubiteket. A dolgozat második fejezetében egy ilyen a véletlenszerű Pauli-hibáktól eltérő hibamodell mutatok be, az úgynevezett koherens hibákat, ebben a hibamodellben az egyes qubitekre, valamilyen untér transzformációval hat a környezet.

A Pauli-hibák szimulációja hatékonyan megvalósítható, azonban a bonyolultabb hibamodelleknél már erősen korlátozva van a felületi kód szimulálhatósága. Bravyi és társai megmutatták [5], hogy a koherens hibák egy csoportja hatékonyan szimulálható a Fermionikus Lineáris Optika eszköztárával, a dolgozat harmadik fejezetében ezt a szimulációs módszert mutatom be. A fejezetben azt is megmutatom, hogyan terjesztettem ki a szimulációs módszert egy olyan esetre, amikor a koherens hibák mellett kiolvasási hibák is megjelennek a felületi kódban.

A dolgozat utolsó fejezetében a numerikus eredményeimet mutatom be. Először saját szimulációmmal reprodukálom az irodalomból már ismert [5, 26] tökéletes kiolvasások melletti koherens hibák esetére meghatározom a felületi kód hibaküszöbét és azt is megmutom, hogy a koherens hibák a hibaküszöb alatti fizikai hibaráták esetén veszélyesebbek, mint az inkoherens hibák. Ezután rátérek az új eredményeimre, azaz a koherens hibák és a kiolvasási hibák

együttes viselkedésének bemutatására. Erre a hibamodellre elsőként meghatározom a felületi kód hibaküszöbét, és megmutatom, hogy a hibaküszöb alatti fizikai hibaráták esetén a koherens hibák a kiolvasási hibák figyelembe vételével is veszélyesebbek mint inkohereus társaik.

2. Kvantumos hibajavító kódok

A kvantuminformáció a klasszikus információval ellentétben nagyon sérülékeny ennek oka, hogy a kvantuminformáció hordozói, a qubitek, meglehetősen érzékenyek a külső zajokra. A mindennapokban hasznos kvantumszámítógépek megvalósítása felé vezető úton az egyik legnagyobb feladat egy olyan módszer kidolgozása, amely leküzdí a fizikai qubitek sérülékenységéből fakadó akadályokat.

A dolgozat ezen fejezetében bemutatom, hogy ha ügyesen kódoljuk be egy fizikai qubit információját egy sok fizikai qubitből álló erősen összefonódott rendszerbe, akkor hatékonyabban megvédhetjük a kvantuminformációt. Az ilyen ügyes bekódolásokat nevezzük kvantum hibajavító kódoknak.

2.1. Stabilizátorformalizmus

A sok qubitből álló rendszerek állapotának leírása igen hosszadalmas és gyakran bonyolult feladat, ezért a kvantum hibajavító kódokat egy sokkal elegánsabb, de talán kevésbé megszokott formalizmus keretében fogjuk tárgyalni, ez a stabilizátorformalizmus. Először is vezessük be a szokásos módon a Pauli-mátrixokat,

$$\hat{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \hat{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; \quad \hat{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}; \quad \hat{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (1)$$

Legyen egy n fizikai qubitből álló rendszerünk, és vezessük be az egyqubit Pauli-operátorokat:

$$\hat{P}_j \in \hat{X}_j, \hat{Y}_j, \hat{Z}_j, \quad (2)$$

ezek az n qubit Hilbert-terén hatnak, az j -edik qubiten \hat{P} -vel a többin \hat{I} -vel. Definiáljuk az egyqubit Pauli-operátorok és az identitás által generált csoportot, ez a Pauli-csoport, melynek jelölése \mathcal{P} . A Pauli-csoport operátorait egyszerűen Pauli-operátoroknak fogom nevezni.

A stabilizátorformalizmus legfontosabb lépése az \mathcal{S} stabilizátorcsoport definiálása, ez általánosságban bármilyen kommutatív részcsoportha lehet

a Pauli-csoportnak, de csak akkor kapunk értelmes hibajavító kódot, ha a stabilizátorcsoport generátorainak száma nem ugyanannyi, hanem kevesebb, mint ahány qubit van a rendszerben, legyen ez k . A stabilizátorcsoport segítségével definiálhatunk egy úgynevezett *logikai alteret*, vagy *kódteret* \mathfrak{L} , amelyben az összes állapot a stabilizátorok sajátállapota $+1$ -es sajátértékkel,

$$\hat{S}|\psi_L\rangle = |\psi_L\rangle \quad \forall \hat{S} \in \mathcal{S} \quad \forall |\psi_L\rangle \in \mathfrak{L}. \quad (3)$$

A stabilizátorok ügyes megválasztásával elérhetjük, hogy a logikai altér védett legyen a lokális Pauli-hibákkal szemben. Ezzel elérhetjük, hogy a logikai altérben tárolt kvantuminformáció védett legyen a külső zajokkal szemben, ami a kvantumos hibajavító kódok ígérete volt.

A kvantumos hibajavító kódok leírását egy sokkal kevésbé formális, és a szokásos fizikai megközelítéshez közelebb álló módon is megtehetjük. Ebben a leírásmódban a logikai altér fogalmához egy sokkal kézzelfoghatóbb jelentéstartalmat tudunk társítani. Először is definiáljunk egy Hamilton-operátort a következő formában:

$$\hat{H} = -J \sum_{j=1}^k \hat{g}_j, \quad (4)$$

itt $\hat{g}_1, \hat{g}_2, \dots, \hat{g}_k$ a stabilizátorcsoport generátorait jelölik. A Hamilton-operátor által definiált rendszer egzaktul megoldható lesz, hiszen tudjuk, hogy a stabilizátorgenerátorok páronként kommutálnak. A rendszer alapállapot energiájához tartozó megoldás egy 2^{n-k} -szorosán degenerált altér lesz, ami minden stabilizátornak a $+1$ -es sajátértékhez tartozó sajátaltère lesz. Ez azt jelenti, hogy a logikai altérben található állapotok lesznek a rendszer alapállapotai,

$$\forall |\psi_L\rangle \in \mathfrak{L}: \quad \hat{H}|\psi_L\rangle = -kJ|\psi_L\rangle \quad \dim \mathfrak{L} = 2^{n-k}. \quad (5)$$

A stabilizátorformalizmus keretei között csak a Pauli-operátorokat tudjuk kezelni, ezeket is 3 kategóriába sorolhatjuk be.

1. A stabilizátorcsoport operátorai.
2. Azok az operátorok, amik nem elemei a stabilizátorcsoportnak, de kommutálnak az összes stabilizátorral.
3. Azok az operátorok, amik legalább 1 stabilizátorral antikommutálnak.

A harmadik kategóriába eső operátorok reprezentálják a hibákat a kódban, ezt egy későbbi fejezetben (2.2) részletesen tárgyalom. Azok az operátorok, amelyek nem stabilizátorok, de kommutálnak minden stabilizátorral, lesznek

a kód *logikai operátorai*. Ezen operátorokon keresztül lehet a logikai altérbe kódolt kvantuminformációval dolgozni. Gondolhatunk úgy a 2^{n-k} dimenziós logikai altérre, hogy m logikai qubit van bekódolva a kódunkba:

$$m = n - k. \quad (6)$$

Ilyenkor mindig tudunk találni m darab logikai \hat{X}^L és \hat{Z}^L operátort, amik teljesítik a következő kommutációs relációkat:

$$\{\hat{X}_j^L, \hat{Z}_j^L\} = 0 \quad [\hat{X}_j^L, \hat{Z}_k^L] = 0 \quad j \neq k, \quad (7)$$

amennyiben találtunk m darab *kódolt logikai operátort* $\hat{X}_1^L, \hat{X}_2^L, \dots, \hat{X}_m^L, \hat{Z}_1^L, \hat{Z}_2^L, \dots, \hat{Z}_m^L$, meggyőződhetünk arról, hogy a kódolt logikai operátorok és a stabilizátorok szorzatai is megfelelő logikai operátorként hatnak a logikai altéren,

$$\hat{S} \hat{X}_j^L |\psi_L\rangle = \hat{X}_j^L |\psi_L\rangle \quad \hat{S} \hat{Z}_j^L |\psi_L\rangle = \hat{Z}_j^L |\psi_L\rangle \quad \forall \hat{S} \in \mathcal{S}. \quad (8)$$

A kvantum hibajavító kódokat 3 paraméterrel tudjuk jellemezni, a fizikai qubitek n számával, a logikai qubitek m számával és az úgynevezett d *kódtávolsággal*. Az utolsó mennyiség azt méri, hogy hány qubiten hat a legkevesebb qubiten ható logikai operátor, azaz a legkisebb súlyú logikai operátor súlya,

$$d = \min_{\hat{S}, \hat{L}} |\hat{S}\hat{L}| \quad \hat{S} \in \mathcal{S}, \quad L \in \{\hat{X}_j^L, \hat{Z}_j^L\}. \quad (9)$$

A kódtávolság hivatott mérni a kód méreteit, később látni fogjuk, hogy ez a hibajavítás szempontjából fontos mennyiség.

2.2. Kvantumos hibajavítás

A stabilizátorformalizmus lehetőséget ad a hibák egy típusának kezelésére, ezek a Pauli-hibák. Ebben a hibamodellben minden olyan Pauli-operátort hibának tekintünk, ami legalább 1 stabilizátorral antikommútál, az ilyen Pauli-operátorokat \hat{E} -vel jelöljük.

$$\hat{S} \hat{E} |\psi_L\rangle = -\hat{E} |\psi_L\rangle \quad \exists \hat{S} \in \mathcal{S}. \quad (10)$$

Ezek a hiba-operátorok jól reprezentálják a kvantumbitek lokális meghibásodását, azaz azt az esetet, amikor a qubiteken egy bizonyos valószínűséggel Pauli-hibák történhetnek. Természetesen ezek is vezethetnek olyan Pauli-operátorokhoz, amik nem lesznek hiba-operátorok, de ennek a valószínűsége kicsi, ezek már nem lesznek lokális hibák.

A hiba-operátorok kicsit jobban megvilágítják a stabilizátorok jelentőségét hiszen, ha ismételten megmérjük a stabilizátorok generátorait, akkor tudjuk detektálni a hibákat anélkül, hogy a mérésekkel logikai altérbe kódolt kvantuminformációt bántanánk.

A stabilizátorgenerátorok mért értékeinek tárolásához be kell vezetnünk egy nagyon fontos mennyiséget ez a *szindróma* $s = \{s_j\}$. Itt a szindróma j -edik tagja s_j megegyezik az j -edik stabilizátorgenerátor \hat{g}_j mért értékével. A kvantum hibajavító kódok hibajavító mechanizmusa a következő fázisokra bontható:

- Történik valamilyen hiba \hat{E} a kódban, de ezt a hibát mi nem ismerjük.
- Megmérjük a stabilizátorgenerátorokat és mért értékekből felépítjük a szindrómát, s -t.
- s szindrómából generálunk egy \hat{C}_s korrekciós operátort, amely jó esetben kijavítja a kezdeti \hat{E} hibát.

Az egész hibajavító mechanizmus legfontosabb kérdése, hogy az ismert szindrómából, hogyan tudunk olyan korrekciós operátort generálni, amelyik kijavítja a kódon történt hibát. A későbbiekben részletesen tárgyalok egy algoritmust (2.5), amelyik megoldja ezt a problémát, de addig meg kell elégednünk a korrekciós operátor egy absztrakt definíciójával. Minden olyan Pauli-operátor megfelelő korrekciós operátor, amelyik a kód hibás állapotát visszaviszi a logikai altérbe, azaz

$$\hat{C}_s \hat{E} |\psi_L\rangle \in \mathcal{L}. \quad (11)$$

Ez a kritérium 2 módon teljesülhet, a $\hat{C}_s \hat{E}$ operátor vagy egy stabilizátor, vagy egy logikai operátor. Abban az esetben, ha stabilizátorra javítottuk a hiba hatását a hibajavítás sikeres, azonban ha logikai operátorra javítottuk a hiba hatását, akkor egy logikai hiba történt, és a hibajavítás sikertelen.

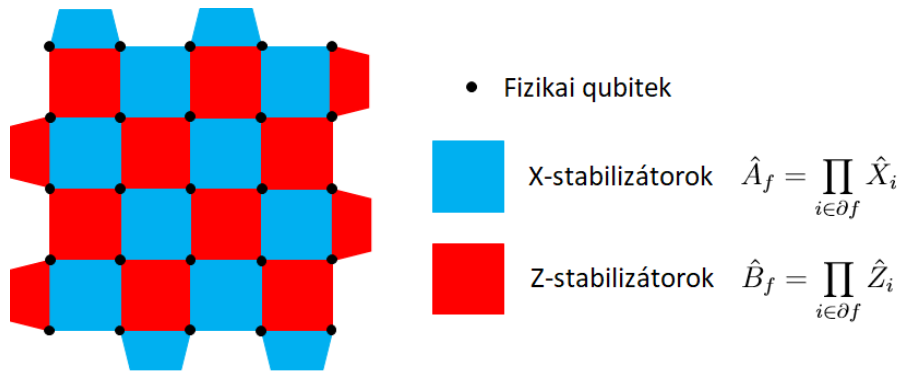
Azt a folyamatot, amiben a szindrómából korrekciós operátort generálunk *dekódolásnak*, az algoritmust pedig, ami a dekódolást végzi, *dekódolónak* hívjuk.

2.3. Felületi kód

Most, hogy már megismerkedtünk a kvantum hibajavító kódok tárgyalásához szükséges stabilizátorformalizmussal a következőkben egy konkrét kódot fogunk megnézni, ez a felületi kód (Surface code) [8, 11, 18]. A felületi kód egy nagyon fontos példa, hiszen a dolgozat későbbi részében ennek a kódnak a tulajdonságait fogjuk behatóan tanulmányozni, másrészt ez az egyik

olyan kód, amiben nagy potenciált látnak a kutatók a kvantumszámítógépek tényleges hibajavítására [22].

A felületi kódban egy négyzetrács rácspontjaiban d^2 fizikai qubit helyezkedik el. A kódban kétféle stabilizátorgenerátort definiálhatunk: az X-stabilizátorokat, melyeket \hat{A} -val jelölök, és a Z-stabilizátorokat, melyeket \hat{B} -vel. (A továbbiakban ezeket egyszerűen stabilizátoroknak fogom nevezni, azonban ezek csak a stabilizátorcsoport generátorai.) Ezek a stabilizátorok a négyzetrács lapjain helyezkednek el felváltva, beleértve bizonyos "éllapokat is".



1. ábra. Példa a felületi kódokra: Az 5-ös kódtávolságú felületi kód, a fekete pontok a fizikai qubitek, a piros és kék lapok a stabilizátorok, beleértve bizonyos éllapokat is.

Az X- és Z-stabilizátorok pontos kifejezése az f lapon a következőképpen írható fel:

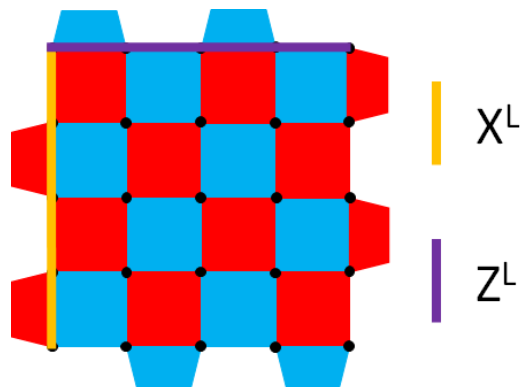
$$\hat{A}_f = \prod_{i \in \partial f} \hat{X}_i \quad \hat{B}_f = \prod_{i \in \partial f} \hat{Z}_i, \quad (12)$$

ahol a ∂f az f lap csúcsainak halmazát jelöli. A stabilizátorokra gondolhatunk úgy, mint X- és Z-paritás operátorokra a megfelelő lapok csúcsaiban lévő qubiteken. A négyzetrács külső határoló élei mentén elhelyezett éllapok csúcsain csak 2 qubit helyezkedik el, ezen felül fontos, hogy nem minden éllap felel meg stabilizátornak, lent és fent csak X-stabilizátorok, bal és jobb oldalon pedig csak Z-stabilizátorok helyezkednek el.

Ezzel a kiosztással a stabilizátorok páronként kommutálnak, hiszen a különböző típusú stabilizátorok vagy 0, vagy 2 qubiten osztoznak. Egy d kódtávolságú kódban ezzel a kiosztással $d^2 - 1$ stabilizátor és d^2 fizikai qubit helyezkedik el, így a bekódolt logikai qubitek száma 1.

A következő lépés, hogy megtaláljuk a logikai qubithez tartozó kódolt logikai X- és Z-operátorokat. Bármely, X-operátorokból álló, egyenes X-vonal, amely összeköti a kód alját a tetejével nyilvánvalóan kommutál minden Z-stabilizátorral a kód belsejében, hiszen vagy 0, vagy 2 qubiten osztoznak. Kommutációs problémák egyedül a vonal végein léphetnek fel, azonban a kód felépítéséből adódóan a Z-stabilizátorok hiányoznak az alsó és a felső élapokon, így kód alját és tetejét összekötő X-vonalak megfelelő kódolt logikai X-operátorok. Hasonló úton eljuthatunk a kódolt logikai Z-operátorokhoz is, ezek a kód bal és jobb oldalát összekötő egyenes Z vonalak. Praktikus okokból a kód bal oldalát és a tetejét választjuk kódolt logikai operátornak, de más választások is jók lennének.

$$\hat{X}^L = \prod_{j \in LEFT} \hat{X}_j \quad \hat{Z}^L = \prod_{j \in TOP} \hat{Z}_j. \quad (13)$$



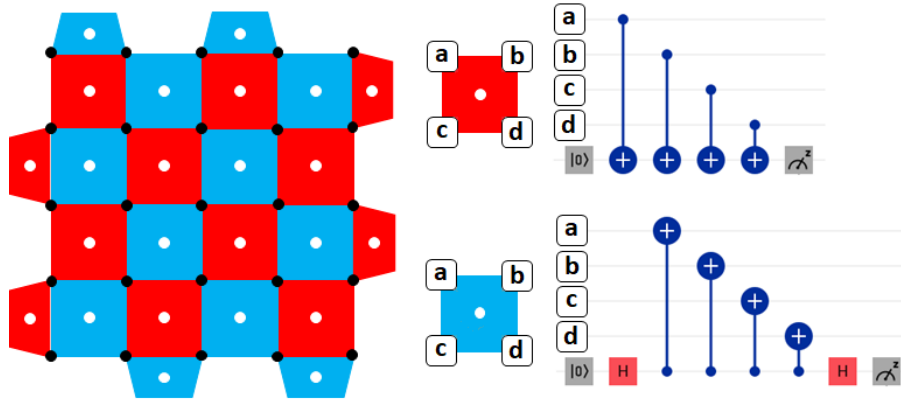
2. ábra. A felületi kód kódolt logikai operátorai. \hat{X}^L (narancs) egy X-vonal a kód bal oldalán, \hat{Z}^L (lila) pedig egy Z-vonal a kód tetején.

A dolgozatban a felületi kódot végig csak egy kvantumos memóriaként tárgyalom, azaz egy olyan protokollként, ami a kódolt kvantuminformációt eltárolja és megvédi a külső zajoktól. Valójában egy kvantumos memóriánál sokkal többre van szükségünk, hogy elérjük a hibatűrő univerzális kvantumszámítást [3]. Megmutatható, hogy felületi kód ügyes trükkök [6, 16] segítségével képes erre, de ennek részletes tárgyalása túlmutat a dolgozat keretein.

2.4. A felületi kód kvantumos logikai áramköri megvalósítása

A felületi kód fizikai realizációja első ránézésre igen nehéz feladatnak tűnik, hiszen a stabilizátorok méréséhez több-qubites méréseket kell kiviteleznünk.

A standard eljárás ilyenkor az *ancilla qubitek* alkalmazása, ezek a segédqubitek valamilyen alapállapotban vannak inicializálva, majd ügyes összefonásokkal elérhetjük, hogy a több-qubites mérések egy az ancillán végzett mérésre redukálódjanak.



3. ábra. A felületi kód az ancilla qubitekkel (fehér pontok) együtt. A stabilizátor mérés áramkörei a különböző fajta stabilizátorokhoz, Z-stabilizátor mérés esetén az ancilla qubitek $|0\rangle$ állapotban vannak inicializálva, majd összefonva a lap sarkain található adat qubitekkel CNOT kapuk segítségével és végül ezek lesznek kimérve. Az X-stabilizátorok esetén az áramkör ugyanaz Hadamard bázisban. Értelemszerűen az éllapokon lévő stabilizátoroknál az áramkör csak 2 adatqubitet és 1 ancillát érint.

Ilyen módon stabilizátor mérések visszavezethetőek az ancilla qubitek megmérésére. Minden ilyen mérés után az ancilla qubitet újra inicializálni kell a $|0\rangle$ állapotba és a mérés kezdődhet előlről.

2.5. A minimális súlyú kapcsolódások elvén működő (MWPM) dekódoló

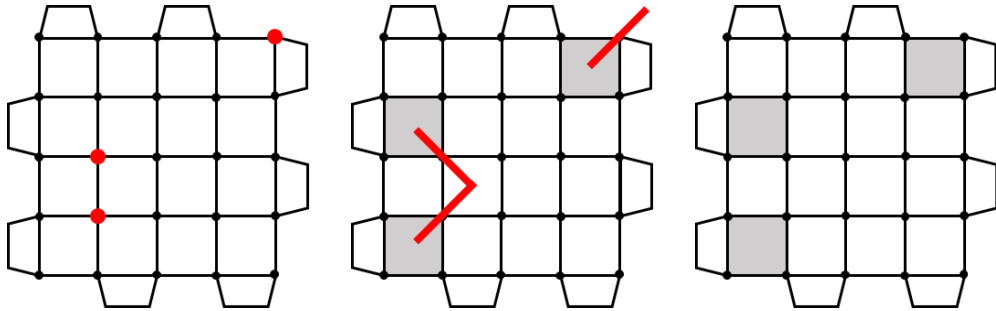
A következőkben egy dekódoló eljárással fogunk behatóan foglalkozni, ez a minimális súlyú kapcsolódások elvén működő dekódoló (Minimum Weight Perfect Matching decoder), a továbbiakban MWPM dekódoló [10]. A hibajavítás során a valamilyen \hat{E} Pauli-hiba történik a felületi kódban és ezt detektáljuk a stabilizátorok mérésével, a mérési eredményeket pedig s szindrómában tároljuk. Mivel a szindrómák nem tartoznak egyértelműen egy hibához ezért a detektálás sem egyértelmű. Ilyenkor logikus eljárás lehet azt a legvalószínűbb hiba osztályt megkeresni, ami az adott szindrómát okozza, itt azonos hibaosztályba tartoznak azok a hibák, amelyeket ugyanaz a korrekciós

operátor ki tud javítani, azaz egy hibaosztályon belüli hibák szorzatai stabilizátorok lesznek. Ilyenkor az adott szindrómához tartozó legvalószínűbb hibaosztályt kell megkeresni és egy olyan korrekciós operátort választani, ami kijavítja a hibaosztályba eső hibákat (Maximum likelihood decoder) [8]. Sajnos ez az optimális eljárás exponenciálisan skálázódik a qubitek számával, így ez a dekódolás nagy méreteknél már nem használható. A nagy méretű kódokhoz egy másik, nem optimális, de jól működő eljárást kell választanunk.

A mért szindrómát úgy is felfoghatjuk, mint összepárosítatlan pontok a felületi kód lapjain ott, ahol a stabilizátor mérési eredménye -1 lett, tehát ha ezeket a pontokat összepárosítjuk (az azonos típusú stabilizátorokat egymással), akkor vissza fogunk térni a logikai altérbe, hiszen az összepárosító operátor biztosan pontosan azokkal a stabilizátorokkal fog antikommutálni, amelyeket -1 -nek mértünk. Az összepárosító operátor megválasztásánál azt az operátort kell választanunk, ami a legnagyobb valószínűséggel egyezik meg a kódban történt \hat{E} hibával. (Fontos, hogy itt már nem hibaosztályokról van szó, hiszen azok kezelése nagyon költséges.) Az ilyen összepárosító operátort úgy tudjuk megtalálni, hogy az egyes qubiteket súlyozzuk az ott előforduló hiba valószínűsége alapján. (Ezt általános egyqubites hibavalószínűségeknél is meg lehet tenni, de most csak az az eset érdekes, amikor a hiba valószínűsége minden qubiten egyforma, ilyenkor a súlyok azonosak.) A megfelelő korrekciós operátor ilyenkor a legkisebb súlyú összepárosító operátor lesz. Ezt a korrekciós operátort találja meg az MWPM dekódoló.

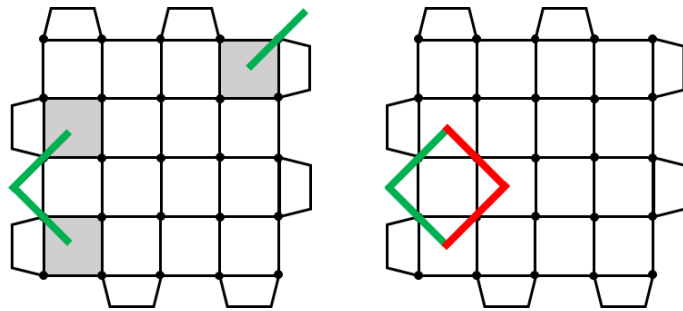
Nézzük meg a dekódoló működését egy konkrét példán keresztül is.

Az egyszerűség kedvéért tételezzük fel, hogy csak Z-hibák történtek a kódban, hiszen az X-hibákat az algoritmus ugyanolyan módon, a Z-hibáktól teljesen függetlenül detektálja és javítja. Néhány qubit elszenvedett egy Z-flippet, és ez a hiba operátor antikommutál bizonyos X-stabilizátorokkal, ezeket -1 -nek mérjük, majd a mérési eredményekből felépítjük a szindrómát.



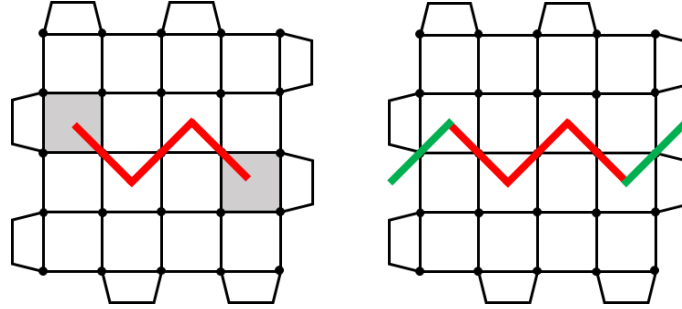
4. ábra. A hibadetektálás. A piros pontok és vonalak reprezentálják a Z-hibákat 3 qubiten, a szürke lapok azokat a stabilizátorokat jelölik, amikkel antikommútál a hiba, azaz ahol -1 -et mérünk. Az utolsó képen a szindróma láthat, valójában csak ezt ismerjük.

Az MWPM dekódoló pontosan ezeket a szürke lapokat próbálja meg összekötni a legkisebb súlyú Z vonallal. Az algoritmus működése szempontjából fontos még, hogy a kód bal és jobb széle is úgy működik, mint a szürke lapok, tehát ezeket nem csak összepárosítani, hanem a szélre "kihúzni" is lehet, ez a kód határfeltételeiből adódik. (X-hibák esetén értelemszerűen az alsó és felső határ játszik ilyen szerepet.) Összefoglalva tehát az algoritmus eltárolja a szürke lapok pontos helyét és kiszámolja azt a legrövidebb utat, ami összeköti őket (beleértve a szélre kihúzást is).



5. ábra. Az MWPM algoritmus alapján számolt korrekciós operátor (zöld) és a korrekciós operátor hiba (piros) szorzata. Ez utóbbi kommutál az összes stabilizátorral, sőt maga is egy Z-stabilizátor.

Abban az esetben, ha a hiba túl sok qubitet érint, lehetséges, hogy a dekódolás sikertelen lesz, ilyenkor a korrekciós operátor és a hiba szorzata egy logikai Z-operátorként fog hatni a kódra.



6. ábra. Sikertelen dekódolás. A hiba (piros) és a korrekció (zöld) szorzata egy logikai Z hibához vezet.

Összefoglalhatjuk a dekódoló sikerességét sokkal formálisabb keretek között is, tehát van valamilyen Z-hibánk \hat{E} , megmérjük a stabilizátorokat és felépítjük a szindrómát s -t, majd az MWPM dekódolóval meghatározzuk a korrekciós operátort \hat{C}_s -t. Amennyiben a dekódolás sikeres volt a kód végső állapota meg fog egyezni a kezdeti állapotával, azaz

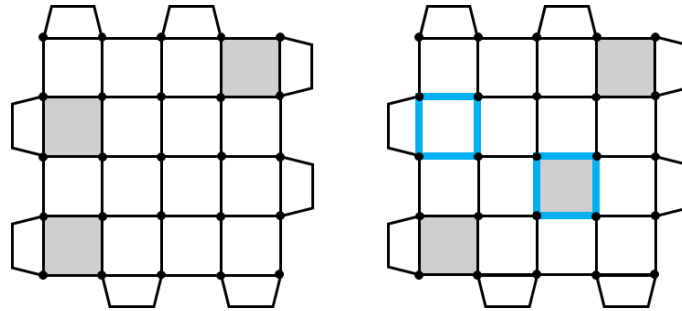
$$\hat{C}_s \hat{E} |\psi_L\rangle = |\psi_L\rangle, \quad (14)$$

Abban az esetben, ha a dekódolás sikertelen a végső állapot elszenved egy logikai Z-hibát:

$$\hat{C}_s \hat{E} |\psi_L\rangle = \hat{Z}^L |\psi_L\rangle. \quad (15)$$

2.6. Hibás szindrómák dekódolása

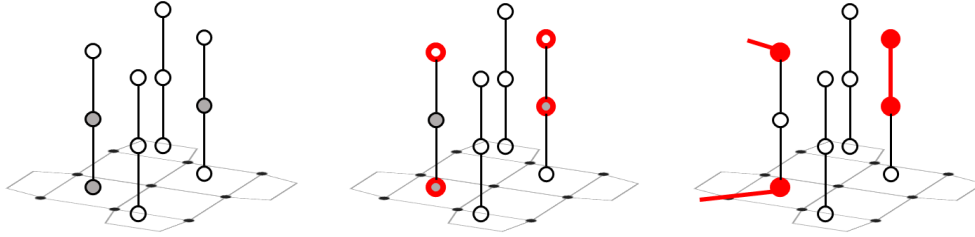
A Pauli-hibákon kívül a felületi kódban fellépő mérési hibákat is kezelniük kell valahogyan. Szerencsére ezt meg tudjuk tenni, vezessük be a *kiolvasási hibákat* (ezeket fenomenológikus hibáknak is szokták nevezni [21], és megfigyelhetőek az ancilla qubiteken történő Pauli-hibáknak), amiket a következőképpen definiálhatunk. Tegyük fel, hogy van egy tökéletes szindrómánk ami tárolja a tényleges mérési eredményeket, azonban amikor ezt megpróbáljuk kiolvasni bizonyos mért értékek megváltoznak, +1-ről -1-re és fordítva. A dekódoló algoritmushoz már csak ez a hibás szindróma jut el.



7. ábra. Tökéletes szindróma (bal) és hibás szindróma (jobb). A kék körök jelölik azokat a lapokat, ahol kiolvasási hibák történtek, azaz ahol a mérési eredmény megváltozott.

Hibás mérések esetén a megfelelő dekódoláshoz többkörös méréseket kell végeznünk, hogy meg tudjuk különböztetni a mérési hibákat a valódi hibáktól. A leoptimálisabb eljárás során d mérési kör eredményeiből felépíthetünk egy 3D-s *szindróma gráfot*, ami alapján egy a mérési hibákat is figyelembe vevő korrekciós operátort tudunk előállítani.

A 3D-s szindróma gráfban a stabilizátor mérési eredmények időbeli változását tároljuk el a 3D-s tér időben. Ebben a képben gondolhatunk úgy a kiolvasási hibákra, mint az időben történt hibákra, a valódi hibákra pedig úgy, mint térben történő hibákra. A megfelelő korrekcióhoz csak össze kell párosítanunk a stabilizátor mérési eredmények időbeli változását jelölő pontokat 3D-ben, az MWPM algoritmus segítségével (a megfelelő határfeltételek figyelembevételével).



8. ábra. A 3-mas kódtávolságú felületi kód dekódolása. Az első képen a kód felett lévő pontok reprezentálják a mért stabilizátorértékeket, a szürke pontok: -1 , fehér pontok: $+1$. A második képen a 3D-s szindróma gráf látható, itt a piros körök jelölik a stabilizátor mérések eredményeinek időbeli változását, magyarul, azt a pontot, ahol mást mértünk, mint az előző körben. Végül a dekódolt szindróma gráfot láthatjuk a korrekciós operátorral (piros), a térbeli korrekciók Pauli- (jelen példában Z) operátorokat jelölnek, az időbeli korrekció pedig csak azt jelöli, hogy a dekódoló szerint egy mérési hiba történt ott, így figyelmen kívül lehet hagyni.

Egy formálisabb nyelven megfogalmazva d kör stabilizátor mérés miatt d körben történhetnek hibák a kódban $\hat{E}_1, \hat{E}_2, \dots, \hat{E}_d$, ezek d darab hibás szindrómához fognak vezetni s'_1, s'_2, \dots, s'_d . Ezekből a hibás szindrómákból fel tudjuk építeni a 3D-s szindróma gráfot és az MWPM algoritmus segítségével kiszámolhatjuk az ehhez tartozó korrekciós operátort $\hat{C}_{s'}$ -t, ahol s' -ben tároljuk mind a d darab hibás szindrómát, így a végállapot a következő lesz:

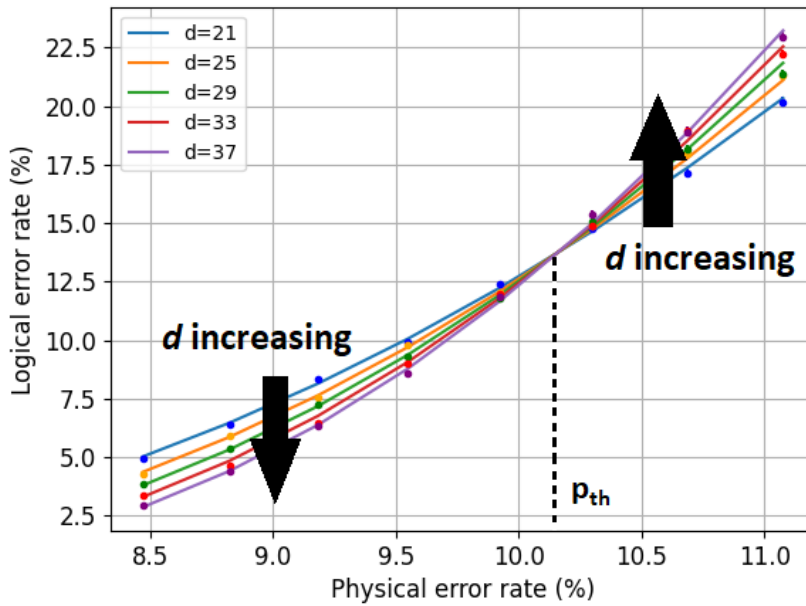
$$\hat{C}_{s'} \hat{E}_d, \hat{E}_{d-1}, \dots, \hat{E}_1 |\psi_L\rangle. \quad (16)$$

Ez a végállapot sajnálatos módon nem mindig lesz a logikai altérben és ez nem jó hír. Különböző megközelítések léteznek a dekódolás jóságának meghatározására ebben az esetben [8, 25], az egyik legegyszerűbb, ha azt képzeljük, hogy az utolsó mérési kör mentes a mérési kiolvasási hibáktól, ilyenkor a korrekció biztosan visszavisz a logikai altérbe.

2.7. Hibaküszöb

Korábban láthattuk, hogy a felületi kód képes kijavítani a hibák egy részét, azonban léteznek olyan hibák is, amelyek logikai hibát okoznak a felületi kódban a hibajavítási procedúra után. A felületi kód akkor hasznos, ha a kód méretének növelésével tudjuk csökkenteni a logikai hibák előfordulásának valószínűségét. Ezt a viselkedést a legegyszerűbb hibamodell segítségével fogjuk megvizsgálni. Itt minden egyes qubiten történhet egy fajta Pauli-hiba (vagy Z vagy X) valamilyen p fizikai hiba valószínűséggel. Az MWPM dekódoló hatékonyságának kvantifikálásához szükségünk van egy új mennyiségre,

a logikai hiba valószínűsége P_L , ez azt mutatja, hogy adott fizikai hiba valószínűsénél mekkora az esélye annak, hogy logikai hibát okoz a dekódolás. A probléma teljes tárgyalásához meg kell találnunk a $P_L(p,d)$ függvényt, hiszen P_L kódtávolságtól való függése az érdekes számunkra. A Pauli-hibák klasszikus viselkedése miatt a hibajavítás hatékonyan szimulálható, így numerikusan meghatározhatjuk a $P_L(p,d)$ függvényt.



9. ábra. A logikai hibaráta a fizikai hibaráta függvényében, különböző kódtávolságok (d) esetén. A pontok: szimulációs eredmények, melyeket 50.000 kör alapján lett meghatározva. A szimulációkat Oscar Higgott MWPM implementációjával végeztem, a PyMatching-gel [15]. A folytonos vonalak egyes illesztési eljárással [27] kapott görbék.

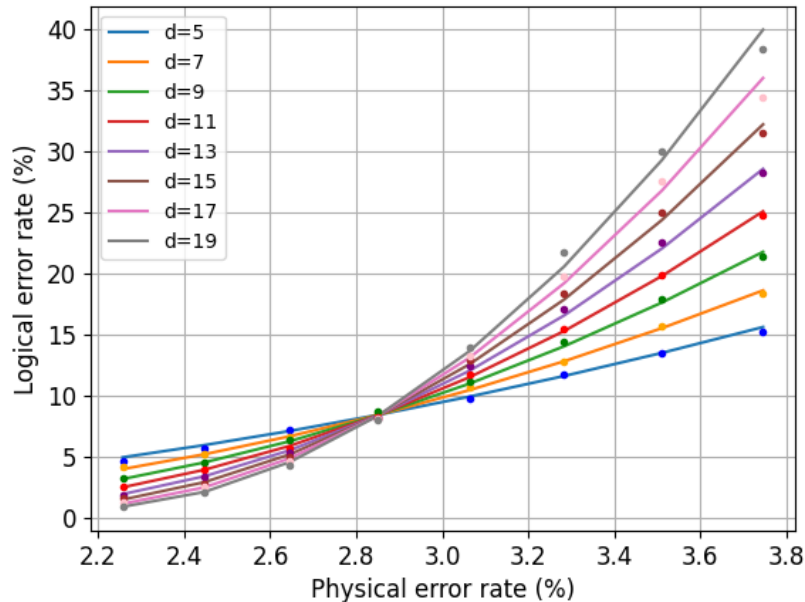
Az ábrán (9) láthatjuk a felületi kód egyik legérdekesebb tulajdonságát, méghozzá azt, hogy kis fizikai hibarátnál megéri növelni a kód méreteit, egyre kisebb a logikai hibaráta, ahogyan d nő. Azonban van egy kritikus fizikai hibaráta, ahol ez a trend megfordul és már nem éri meg felskálázni a kódot, ezt a kritikus értéket hívják *hibaküszöbnek*.

A hibaküszöböt (az illesztés hibájával együtt) erre a hibamodellre numerikusan meghatároztam:

$$p_{th} = 10.17\% \pm 0.04\% \quad (17)$$

A hibaküszöb meghatározásához egy okos illesztési eljárást alkalmaztam [27]. Megmutatható, hogy a fizikai hibaráta és a kódtávolság egy paraméterbe gyurható össze, és a logikai hibaráta már csak ettől a paramétertől fog függeni. Az illesztés során ezt a függvényt másodfokban közelítettem, de ennek pontos bemutatása túlmutat a dolgozat keretein.

Hasonló viselkedést tapasztalhatunk akkor, ha a kiolvasási hibákat is megengedjük a szimulációkban. Ebben a hibamodellben egy típusú Pauli-hiba történhet p valószínűséggel és az egyszerűség kedvéért a kiolvasási hiba valószínűségének is ugyanezt a p értéket választottam, ahogy ez az irodalomban is elterjedt.



10. ábra. A logikai hibaráta a fizikai hibaráta p függvényében. Pontok: szimulációs eredmények, melyeket a PyMatching segítségével végeztem el. Minden pont 50.000 kör alapján lett meghatározva, ahol egy körben a dekódoló egy $d \times d \times d$ méretű 3D-s szindróma gráfot dekódolt. A folytonos vonalak az illesztett görbék.

Itt is meghatároztam a hibaküszöböt, ami egy jelentősen alacsonyabb érték, mint amikor tökéletes méréseket feltételezünk.

$$p_{th} = 2.85 \pm 0.01\% \quad (18)$$

Az itt ismertetett hibaküszöbök értékeit már többen is meghatározták [23, 27], ezek az értékek jó egyezést mutatnak az általam meghatározott értékekkel. (A kis különbségek abból adódhatnak, hogy a felületi kódot különböző határfeltételekkel és stabilizátor kiosztásokkal is lehet definiálni, ezek a véges méretek esetén apró különbségeket okozhatnak.)

3. Koherens hibák

A Pauli-hibák könnyen kezelhetők a stabilizátorformalizmus keretei között, és így szimulálásuk is hatékonyan megvalósítható, azonban a valóságban előforduló hibák jóval bonyolultabbak, mint a Pauli-zaj. Az általánosan előforduló hibák szimulálását nem lehet hatékonyan elvégezni, ha el lehetne nem lenne értelme kvantumszámítógépet építeni.

Ebben a fejezetben egy nagyon specifikus és potenciálisan veszélyes hibamodellt fogunk megvizsgálni, ez a koherens hibák modellje. Azt is megvizsgáljuk, hogyan lehet ezt a hibamodellt összeegyeztetni a kiolvasási hibákkal.

3.1. Kvantumos hibacsatornák

Először is vezessünk be egy pár mennyiséget, amik nagyon hasznosak lesznek a hibák leírásában. A *kvantumos hibacsatornák* [20] olyan folyamatokat reprezentálnak, amiben a rendszer sűrűségoperátora valamilyen változáson megy keresztül. Legáltalánosabban a következőképpen írhatjuk fel a folyamatot:

$$\varepsilon(\hat{\rho}) = \sum_j \hat{K}_j \hat{\rho} \hat{K}_j^\dagger. \quad (19)$$

Itt a \hat{K}_j *Kraus-operátorok* kielégítik a következő feltételt:

$$\sum_j \hat{K}_j^\dagger \hat{K}_j = 1. \quad (20)$$

Az olyan többqubites rendszereknél, amilyen a felületi kód is, nagyon sokféle hibacsatornát tudunk elképzelni, azonban a továbbiakban csak az egyqubites hibacsatornák kompozíciójaként előálló hibákkal fogunk foglalkozni. Ezzel a megszorítással kizárjuk a hibák egy széles családját, a korrelált hibákat, amik bizonyos esetekben elég fontosak és egyáltalán nem elhanyagolhatóak [19].

$$\varepsilon = \varepsilon_1 \circ \varepsilon_2 \circ \dots \circ \varepsilon_n, \quad (21)$$

itt ε_j az n -qubites rendszer j -edik qubitjén ható hibacsatorna.

Nézzünk meg egy fontos, már jól ismert példát, a véletlenszerű Pauli-hibákat. Az egyqubitese hibacsatornák a következő alakban írhatóak ilyenkor:

$$\varepsilon(\hat{\rho}) = p_0\hat{\rho} + p_x\hat{X}\hat{\rho}\hat{X} + p_z\hat{Z}\hat{\rho}\hat{Z} + p_y\hat{Y}\hat{\rho}\hat{Y}, \quad (22)$$

ahol a (20)-as egyenletből tudjuk, hogy:

$$p_0 + p_x + p_z + p_y = 1. \quad (23)$$

A hibacsatornát leíró absztrakt kifejezés mögött egy nagyon egyszerű kép bújik meg. Véletlenszerű Pauli-hibák történhetnek az adott qubiten, \hat{X} hiba p_x , \hat{Y} p_y és \hat{Z} p_z valószínűségekkel. Annak a valószínűsége, hogy nem történik hiba p_0 .

Később látni fogjuk, hogy bizonyos esetekben nagyon hasznos lehet a hibacsatornákat Pauli hibacsatornákkal közelíteni, ezért bevezetjük a *Pauli-pödrés* közelítést (Pauli twirling approximation) [9]. Egy egyqubitese hibacsatorna Pauli-pödrött verziójának definíciója a következő:

$$\varepsilon_{twirl}(\hat{\rho}) = \frac{1}{4} \sum_{\hat{P} \in \{\hat{I}, \hat{X}, \hat{Y}, \hat{Z}\}} \hat{P} \varepsilon(\hat{P}\hat{\rho}\hat{P}) \hat{P}. \quad (24)$$

A definíció mögött megbúvó gondolat nagyon egyszerű, pusztán annyit csinálunk, hogy a (19)-es egyenletben definiált Kraus-operátorokat kifejtjük a Pauli-bázisban, így a hibacsatorna kifejezésében lesznek "kevert tagok", ahol a sűrűség mátrix bal és jobb oldalán nem ugyanaz az operátor áll. A Pauli-pödrés közelítésben ezeket a tagokat elhagyjuk.

A kvantuminformáció-elmélet egyik fontos kérdése, hogyan tudjuk számszerűsíteni a különböző kvantumállapotok hasonlóságát. Két állapot távolságára bevezethetjük a *hűség* (fidelity) fogalmát,

$$F(\hat{\rho}, \hat{\sigma}) = \left(\text{Tr} \sqrt{\sqrt{\hat{\rho}} \hat{\sigma} \sqrt{\hat{\rho}}} \right)^2. \quad (25)$$

Abban az esetben ha nem két állapot távolságát szeretnénk számszerűsíteni, hanem egy kvantum hibacsatornát (leképezést) szeretnénk jellemezni, érdemes a legrosszabb esetet figyelembe vennünk, azaz minimalizálnunk a leképezés előtti és utáni állapotok hűségét az összes állapotra. Praktikus okokból nem a hűség minimát, hanem a hűtlenséget (infidelity) maximumát vezettem be, ezt a továbbiakban P_L *logikai hibaráta* fogom nevezni,

$$P_L(\varepsilon) = \max_{\hat{\rho}} (1 - F(\hat{\rho}, \varepsilon(\hat{\rho}))). \quad (26)$$

A másik fontos mérték, ami két kvantumállapot hasonlóságát hívatott mérni a *trace-norma* (trace distance), ennek definíciója a következő:

$$\|\hat{\rho} - \hat{\sigma}\|_1 = \text{Tr}\left(\sqrt{(\hat{\rho} - \hat{\sigma})^\dagger(\hat{\rho} - \hat{\sigma})}\right). \quad (27)$$

Itt is megtehetjük, hogy két állapot helyett egy kvantumos hibacsatorna jellemzésére optimalizáljuk ezt a mértéket, azonban itt a konvenció szerint nem csak egyszerű maximalizálást szokás bevezetni a lehetséges állapotokra, hanem egy nagyobb térre való kiterjesztés is. Ennek, akkor van jelentősége, ha az n qubites rendszerünk össze van fonódva a környezetével, ilyenkor egy $2n$ qubites rendszerré történő kiterjesztés, azaz n ancilla qubit bevezetése segíthet a rendszer kezelésében. Ezt a mértéket gyémánt normának (diamond norm) szokás nevezni,

$$\|\varepsilon\|_\diamond = \max_{\hat{\rho}} \|(\varepsilon \otimes \mathbb{1})(\hat{\rho}) - (\mathbb{1} \otimes \mathbb{1})(\hat{\rho})\|_1. \quad (28)$$

A két különböző mérték más-más aspektusból közelíti meg a kvantumos hibacsatornák "jóságát", később látni fogjuk, hogy a felületi kódban előforduló koherens hibák esetén ez a különbség szembeötlő lesz.

3.2. Általános hibák a felületi kódban

A Pauli-hibák tárgyalásánál nagy könnyítés volt számunkra, hogy a stabilizátor mérések nem befolyásolták a felületi kód állapotát, hiszen minden elképzelhető Pauli-hiba az egyes stabilizátorok valamilyen sajátállapotába viszi a kódot. Általános egyqubites hibák esetén ez a tulajdonság már nem lesz igaz, így a stabilizátormérések visszahatását is figyelembe kell vennünk.

Vezessük be az általános stabilizátorokat, ha a lapokon \hat{A} , b lapokon pedig \hat{B} stabilizátor helyezkedik el

$$\hat{S}_f = \hat{A}_f \quad f \in a \quad \hat{S}_f = \hat{B}_f \quad f \in b. \quad (29)$$

Ezek után be tudjuk vezetni az s szindrómához tartozó projektort,

$$\hat{\Pi}_s = \prod_{\forall f} \frac{1}{2}(1 + s_f \hat{S}_f), \quad (30)$$

ahol s_f az s szindrómában tárolt mérési eredménye \hat{S}_f -nek (értéke ± 1). Ezen projektorok segítségével már könnyen felírhatjuk a felületi kód sűrűségoperátorát a hibajavító procedúra után, ahol s szindrómát mértünk, tetszőleges ε hibacsatorna esetén.

$$\hat{\rho}_L^s = \frac{1}{p(s)} \hat{C}_s \hat{\Pi}_s \varepsilon(\hat{\rho}_L^0) \hat{\Pi}_s \hat{C}_s, \quad (31)$$

itt a kód kezdetben ρ_L^0 logikai állapotban van. $p(s)$ annak a valószínűsége, hogy s szindrómát mértünk a stabilizátorok mérésénél,

$$p(s) = \text{Tr}(\hat{\Pi}_s \varepsilon(\hat{\rho}_L^0)). \quad (32)$$

A hibák jellemzése szempontjából a mért szindrómák nem fontosak, ezért vezessük be a *logikai hibacsatorna* fogalmát, amiben a különböző szindrómákhoz tartozó kezdőállapot végállapot átlagolva vannak a szindrómák valószínűségével súlyozva.

$$\varepsilon_L(\hat{\rho}_L^0) = \sum_{\forall s} p(s) \hat{\rho}_L^s. \quad (33)$$

3.3. Koherens forgatások

A koherens hiba alatt azt értjük, amikor minden egyes qubiten valamilyen szisztematikus unitér művelet megy végbe, de ez qubitenként eltérő is lehet. Az egyqubites unitér műveletet forgatásnak is nevezhetjük, az egyqubites hibacsatorna általános alakja a következő:

$$\varepsilon_j^{coh}(\hat{\rho}) = \hat{U}_j \hat{\rho} \hat{U}_j^\dagger \quad \hat{U}_j^\dagger \hat{U}_j = \hat{I}_j. \quad (34)$$

A teljes kódon értelmezett hibacsatorna pedig értelemszerűen az egyqubites hibacsatornák kompozíciója.

$$\varepsilon^{coh} = \varepsilon_1^{coh} \circ \varepsilon_2^{coh} \circ \dots \circ \varepsilon_n^{coh}. \quad (35)$$

Egy qubiten történő koherens hiba első ránézésre nem tűnik veszélyesnek a felületi kód szempontjából hiszen, ha a forgatás kicsi akkor a stabilizátormérések során a meghibásodott qubit nagy valószínűséggel visszavetítődik a hiba előtti állapotába, és egy kis valószínűséggel bevetítődik egy Pauli-hibával leírható állapotba. A koherens hibák nagy veszélye abban rejlik, hogy az összes qubit ugyanabba az irányba forog, ráadásul minden egyes körben ugyanúgy.

A felületi kódban fellépő koherens hibákat már többen vizsgálták, pusztán elméleti módszerekkel [17], kis kódok általános szimulációjával [12, 25], közelítő eljárásokkal [7, 14] és egy okos szimulációs eljárással, ami hatékonyan alkalmazható nagy kódokra is [5, 26]. A munkám során én is ezt az utóbbi szimulációs módszert alkalmaztam, azonban itt a hatékonyság komoly megszorításokkal jár, így csak olyan koherens hibákat tudunk vizsgálni, amik a Z-tengely körüli forgatásoknak felelnek meg. (Ez az X-tengely is lehet a

lényeg, hogy vagy az egyik, vagy a másik.) Ilyenkor az egyes qubitek hibacsatornája a következő:

$$\varepsilon_j(\rho) = e^{i\theta_j \hat{Z}_j} \hat{\rho} e^{-i\theta_j \hat{Z}_j}, \quad (36)$$

itt a θ_j *fizikai forgatási szögek* általános esetben minden qubiten különbözhetnek, a szimulációs módszer megengedi ezt és ki is használja, ezért a leírásban inhomogén θ_j eloszlást is megengedünk. Valójában viszont csak azt az esetet vizsgáljuk, amikor fizikai forgatási szögek minden qubiten azonosak. A különböző forgatási szögek okozta hibák általános leírása még egy feltérképezetlen, nyitott probléma.

A Z-tengely körüli koherens forgatások okozta hiba operátora a következő alakban írható fel:

$$\hat{E}_{coh}(\theta) = \prod_{j=1}^n e^{i\theta_j \hat{Z}_j}. \quad (37)$$

A felületi kód s mért szindrómához tartozó végállapota a következőképpen áll elő:

$$|\phi_L^s\rangle = \frac{1}{\sqrt{p(s)}} \hat{C}_s \hat{\Pi}_s \hat{E}_{coh} |\psi_L^0\rangle, \quad (38)$$

ahol s szindróma mérésének valószínűsége:

$$p(s) = \langle \psi_L^0 | \hat{E}_{coh}^\dagger \hat{\Pi}_s \hat{E}_{coh} | \psi_L^0 \rangle. \quad (39)$$

A $|\phi_L^s\rangle$ végállapotnak több meglepő és nagyon fontos tulajdonsága van. Egyrészt $p(s)$ független a kiinduló $|\psi_L^0\rangle$ logikai állapottól, másrészt a végállapot mindig felírható a kezdeti állapot egy Z-tengely körüli elforgatásaként, azaz:

$$|\phi_L^s\rangle = e^{i\theta_s \hat{Z}^L} |\psi_L^0\rangle. \quad (40)$$

Itt a θ_s *logikai forgatási szög* szintén független a kiinduló állapottól, csak a mért szindróma függvénye. (Az állítások bizonyítása kicsit hosszadalmas, de megtalálható Bravyiék cikkének függelékében [5].)

A végállapotra vonatkozó központi állítások (40) felhasználásával kifejezhetjük a koherens Z-forgatásokat leíró logikai hibacsatornát jellemző, már korábban bevezetett (25,28) mennyiségeket. A logikai hibaráta és a gyémánt normára a következő kifejezéseket kapjuk:

$$P_L(\varepsilon_L^{coh}) = \sum_{\forall s} p(s) \sin^2(\theta_s) \quad ||\varepsilon_L^{coh}||_\diamond = 2 \sum_{\forall s} p(s) |\sin(\theta_s)|. \quad (41)$$

3.4. Kiolvasási hibák és koherens forgatások

A felületi kódban a koherens hibák mellett a már korábban bevezetett (2.6) kiolvasási hibák is megjelenhetnek, ilyenkor a hibajavítást jellemző logikai hibacsatorna leírása kicsit körülményesebb. A koherens Z-forgatások okozta hibák viselkedését tökéletes kiolvasások mellett, már alaposan feltérképezték [5, 26]. Az általam vizsgált hibamodellben azonban a koherens hibák mellett kiolvasási hibák is megjelenhetnek a felületi kódban, így ebben a fejezetben ezek leírását tekintem át.

A hibajavítás során d stabilizátor mérési kör adataiból felépített három dimenziós szindróma gráfot kell dekódolnunk, ezért egy teljes hibajavító körben d körben történhet hiba és d stabilizátormérés történik. Minden egyes stabilizátormérési körben s^j tökéletes szindrómát mérjük, azonban nem ezt olvassuk ki, (a j -edik körben mért s^j szindrómában tárolt k -edik stabilizátormérés eredménye s_k^j). A kód hibajavítás előtti állapota a következő alakba írható:

$$|\psi_{err}^{\underline{s}}\rangle = \frac{1}{\sqrt{p(\underline{s})}} \hat{\Pi}_{s^d} \hat{E}_{coh} \hat{\Pi}_{s^{d-1}} \hat{E}_{coh} \dots \hat{\Pi}_{s^1} \hat{E}_{coh} |\psi_L^0\rangle, \quad (42)$$

ahol $p(\underline{s})$ annak a valószínűsége, hogy mért tökéletes szindróma az j -edik körben s^j .

$$p(\underline{s}) = \langle \psi_L^0 | \hat{E}_{coh}^\dagger \hat{\Pi}_{s^1} \dots \hat{E}_{coh}^\dagger \hat{\Pi}_{s^{d-1}} \hat{E}_{coh}^\dagger \hat{\Pi}_{s^d} \hat{E}_{coh} \hat{\Pi}_{s^{d-1}} \hat{E}_{coh} \dots \hat{\Pi}_{s^1} \hat{E}_{coh} | \psi_L^0 \rangle. \quad (43)$$

A szindróma kiolvasások közben a stabilizátor mérési eredmények p valószínűséggel megváltoznak, így kapjuk meg a kiolvasott hibás szindrómákat (\underline{s}'), majd ezekből felépítjük a 3D-s szindróma gráfot és megadjuk a korrekciós operátort az MWPM dekódoló segítségével $\hat{C}_{\underline{s}'}$. Ilyenkor az \underline{s} tökéletes szindrómákhoz és \underline{s}' hibás szindrómákhoz tartozó végállapot a következő alakban írható fel:

$$|\phi^{\underline{s}, \underline{s}'}\rangle = \hat{C}_{\underline{s}'} |\psi_{err}^{\underline{s}}\rangle. \quad (44)$$

Megmutatható (lásd: függelék), hogy (44)-es kifejezést a következő alakban lehet felírni:

$$|\phi^{\underline{s}, \underline{s}'}\rangle = \hat{C}_{\underline{s}'} \hat{C}_{s^d} e^{i\theta^* \hat{Z}^L} |\psi_L^0\rangle, \quad (45)$$

itt $\theta^*(\underline{s})$ forgatási szög csak a tökéletes szindrómák függvénye, és C_{s^d} a d -edik tökéletes szindrómához tartozó korrekciós operátor.

A $|\phi^{\underline{s},\underline{s}'}\rangle$ állapot nem feltétlenül lesz a logikai altéren, ezért feltételezzük, hogy a d -edik szindróma mérést tökéletesen olvassuk ki,

$$s^d = s'^d. \quad (46)$$

Ilyenkor a $\hat{C}_{\underline{s}'}\hat{C}_{s^d}$ szorzat vagy az identitás, vagy pedig egy logikai Z-operátor, tehát a végső állapot a következő alakban áll elő:

$$|\phi_L^{\underline{s},\underline{s}'}\rangle = e^{i\theta_L \hat{Z}^L} |\psi_L^0\rangle, \quad (47)$$

ahol $\theta_L(\underline{s},\underline{s}')$ logikai forgatási szög a $\hat{C}_{\underline{s}'}$ korrekciós operátoron keresztül függ a hibás szindrómáktól.

$$\begin{aligned} \theta_L = \theta^* & \quad \leftarrow \hat{C}_{\underline{s}'}\hat{C}_{s^d}|\psi_L\rangle = |\psi_L\rangle \\ \theta_L = \theta^* + \frac{\pi}{2} & \quad \leftarrow \hat{C}_{\underline{s}'}\hat{C}_{s^d}|\psi_L\rangle = \hat{Z}^L|\psi_L\rangle. \end{aligned} \quad (48)$$

A hibamodell logikai hibacsatornájának (ε^{coh+m}) jellemzésére szolgáló metrikák, a gyémánt norma (28) és a logikai hibaráta (26) a következőképpen írható fel:

$$\begin{aligned} P_L(\varepsilon_L^{coh+m}) &= \sum_{\forall \underline{s},\underline{s}'} p(\underline{s})p(\underline{s} \rightarrow \underline{s}') \sin^2(\theta_L) \\ \|\varepsilon_L^{coh+m}\|_{\diamond} &= 2 \sum_{\forall \underline{s},\underline{s}'} p(\underline{s})p(\underline{s} \rightarrow \underline{s}') |\sin(\theta_L)|. \end{aligned} \quad (49)$$

4. A szimulációs módszer

A következő fejezetben egy látszólag, az eddigiektől teljesen független téma tárgyalásába kezdünk, fermionikus rendszereket fogunk vizsgálni. Ezen belül is speciális fermionikus állapotokkal, az úgynevezett *tiszta Gaussi állapotokkal* fogunk foglalkozni. Ezen speciális állapotok klasszikusan hatékonyan szimulálhatók a *Fermionikus Lináris Optika* (FLO) [4, 24] eszköztárával.

A fejezet második felében a felületi kód fermionikus rendszerként való leírásáról lesz szó és arról, hogyan lehet a koherens Z-forgatások hatását tiszta Gaussi állapotokon keresztül végigkövetni, azaz hatékonyan szimulálni a koherens hibákat a felületi kódban.

4.1. Tiszta Gaussi állapotok

Legyen egy N fermionból álló rendszerünk és vezessük be a fermionikus keltő \hat{a}^\dagger és eltüntető \hat{a} operátorokat, melyek a következő antikommutációs relációt teljesítik:

$$\{\hat{a}_i, \hat{a}_j\} = 0 \quad \{\hat{a}_i, \hat{a}_j^\dagger\} = \delta_{ij}. \quad (50)$$

A rendszer Hilbert-terében felírhatunk egy bázist a betöltési szám reprezentációban:

$$|n_1, n_2, \dots, n_N\rangle = (\hat{a}_1^\dagger)^{n_1} \dots (\hat{a}_N^\dagger)^{n_N} |0\rangle \quad n_i \in \{0, 1\}, \quad (51)$$

ahol a $|0\rangle$ állapot a fermionikus vákuum,

$$\hat{a}_i |0\rangle = 0 \quad \forall i. \quad (52)$$

A keltő és eltüntető operátorok helyett térjünk át a Majorana operátorokra, melyeket a következő módon tudjuk definiálni:

$$\hat{c}_{2j} = \hat{a}_j + \hat{a}_j^\dagger \quad \hat{c}_{2j-1} = i(\hat{a}_j - \hat{a}_j^\dagger) \quad j = 1, \dots, N. \quad (53)$$

A Majorana operátorok központi szerepet játszanak a FLO szimulációkban, ezen operátorok antikommutációs relációi a következők:

$$\{\hat{c}_j, \hat{c}_k\} = 2\delta_{jk}. \quad (54)$$

Vizsgáljuk azokat a fermionikus rendszereket, amik egy kvadratikus Hamilton-operátorral írhatóak le. (Itt most a Majorana operátorok szerinti kvadrati-kusságot nézzük, de könnyen belátható, hogy ez ekvivalens a fermionikus operátorok szerinti kvadratikusával.)

$$\hat{H} = i \sum_{j,k=1}^{2N} A_{jk} \hat{c}_j \hat{c}_k. \quad (55)$$

Ezen Hamilton-operátoron keresztül tudjuk definiálni a tiszta Gaussi állapotokat, ugyanis minden olyan állapot tiszta Gaussi állapot, amelyik egy kvadratikus Hamilton-operátor alapállapota.

A Hamilton-operátor hermitikuságából és az antikommutációs relációkból (54) következik, hogy a A_{jk} mindig választható egy valós antiszimmetrikus mátrixnak. A Hamilton-operátort a következő alakban is felírhatjuk:

$$\hat{H} = i \underline{\hat{c}}^T \underline{A} \underline{\hat{c}}, \quad (56)$$

ahol $\underline{\hat{c}}$ a Majorana operátorokból felépített $2N$ dimenziós vektor.

A valós antiszimmetrikus mátrixok, mint \underline{A} mindig egy úgynevezett blokk-diagonális alakra hozhatóak egy \underline{R} valós ortogonális forgatással.

$$\underline{R} \underline{A} \underline{R}^T = \bigoplus_{n=1}^N \begin{bmatrix} 0 & \alpha_n \\ -\alpha_n & 0 \end{bmatrix}, \quad (57)$$

így a Hamilton-operátor a következő alakban áll elő:

$$\hat{H} = i\hat{c}^T \underline{R}^T \underline{A} \underline{R} \hat{c}, \quad (58)$$

ahol az elforgatott Majorana vektor már transzformált Majorana operátorokból épül fel:

$$\hat{c}' = \underline{R} \hat{c} = (\hat{c}'_1, \hat{c}'_2, \dots, \hat{c}'_{2N})^T. \quad (59)$$

Ezeket az operátorokat is jogosan nevezhetjük Majorana operátoroknak, hiszen teljesítik a megfelelő antikommutációs relációkat.

$$\{\hat{c}'_j, \hat{c}'_k\} = 2\delta_{jk}. \quad (60)$$

A Gaussi állapotok, azaz a Hamilton-operátor alapállapota ilyen módon a $i\hat{c}'_{2n}\hat{c}'_{2n-1}$ elforgatott Majorana párok sajátállapotai lesznek.

$$i\hat{c}'_{2n}\hat{c}'_{2n-1}|GS\rangle = |GS\rangle \quad n = 1, 2, \dots, N. \quad (61)$$

A Gaussi állapotok azért lényegesek, mert teljesen karakterizálja őket egy $2N \times 2N$ -es mátrix, az úgynevezett *kovarianciamátrix*. Ebben a tulajdonságban rejlik a hatékony szimulálhatóság kulcsa, hiszen ha a rendszerünk Gaussi állapotokon keresztül fejlődik, akkor nem kell a teljes állapotvektort fejlesztenünk, csak a kovarianciamátrixot. A kovarianciamátrix definíciója a következő:

$$M_{jk}(|\psi\rangle) = \langle \psi | i\hat{c}_j \hat{c}_k | \psi \rangle - i\delta_{jk}. \quad (62)$$

Láthatjuk, hogy a kovarianciamátrixot bármilyen $|\psi\rangle$ állapotra lehet definiálni, azonban a Gaussi állapotok kovariancia mátrixa különleges. Könnyen belátható, hogy a $|GS\rangle$ Gaussi állapothoz tartozó \underline{R} mátrix a következő blokk-diagonális alakba transzformálja a kovariancia mátrixot:

$$\underline{R} \underline{M}(|GS\rangle) \underline{R}^T = \bigoplus_{n=1}^N \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \quad (63)$$

Ez azt jelenti, hogy a Gaussi állapotok kovariancia mátrixa ortogonális lesz. (Belátható, hogy ez ekvivalensen jó definíciója a tiszta Gaussi állapotoknak és ez egy könnyen ellenőrizhető tulajdonság.)

Korábban már szó esett arról, hogy a tiszta Gaussi állapotokat tökéletesen karakterizálja a kovariancia mátrixuk, ezt fejezi ki Wick-tétele [24], ami arra vonatkozik, hogy a tetszőlegesen hosszú Majorana láncok várható értéke kifejezhető a kovariancia mátrix segítségével:

$$i^p \langle GS | \hat{c}_{j_1} \hat{c}_{j_2} \dots \hat{c}_{j_{2p}} | GS \rangle = \text{Pf}(M(|GS\rangle)_{j_1, j_2, \dots, j_{2p}}). \quad (64)$$

Itt bal oldalon a kovarianciamátrix egy részmatrixának Pfaff-polinomja [13] jelenik meg, hogy ez pontosan mit jelent azt nem kívánom részletezni, hiszen a tételt, csak egy egyszerű esetre a 4 egység hosszú Majorana-láncokra fogom alkalmazni.

$$\langle \hat{c}_p \hat{c}_q \hat{c}_r \hat{c}_s \rangle = \langle \hat{c}_p \hat{c}_q \rangle \langle \hat{c}_r \hat{c}_s \rangle - \langle \hat{c}_p \hat{c}_r \rangle \langle \hat{c}_q \hat{c}_s \rangle + \langle \hat{c}_p \hat{c}_s \rangle \langle \hat{c}_q \hat{c}_r \rangle. \quad (65)$$

4.2. Fermionikus Lineáris Optika

A FLO szimulációs módszer olyankor alkalmazható, ha egy rendszer a fejlődése során csak tiszta Gaussi állapotokon megy keresztül, ilyenkor elegendő a rendszer kovariancia mátrixát fejleszteni, hiszen abban a minden információ benne van. Egy N fermionikus módust tartalmazó rendszer esetén $\mathcal{O}(N)$ nagyságrendű fejlesztési lépést $\mathcal{O}(N^3)$ idő alatt tudunk szimulálni.

Az FLO szimulációknál vannak bizonyos megengedett lépések, amelyek egy Gaussi állapotot Gaussi állapotba visznek. A szimulációk során csak ezeket a megengedett lépéseket tudjuk végrehajtani, melyek a következők:

1. Majorana párok által stabilizált állapotok inicializálása.
2. Kvadratikus Hamilton-operátorok által indukált időfejlődés.
3. Majorana párok mérése.

A Majorana párok által stabilizált Gaussi állapotok kovariancia mátrixa igen egyszerűen felírható. Legyen N darab (p,q) pár, az ehhez tartozó Majorana párok stabilizálják az adott Gaussi állapotot,

$$i\hat{c}_p \hat{c}_q |GS\rangle = |GS\rangle. \quad (66)$$

Ilyenkor a kovarianciamátrix a következő alakban áll elő:

$$M_{j,k} = \sum_{\forall(p,q)} \delta_{jp} \delta_{kq} - \delta_{jq} \delta_{kp}. \quad (67)$$

A kovarianciamátrix létrehozásának ideje $\mathcal{O}(N)$ nagyságrendben skálázódik.

Egy tiszta Gaussi állapot kvadratikus Hamilton-operátor által indukált időfejlődése a következő alakban írható fel:

$$|GS(t)\rangle = \exp\left(t \sum_{j,k} A_{jk} \hat{c}_j \hat{c}_k\right) |GS(0)\rangle. \quad (68)$$

Itt fontos, hogy az időfejlődést leíró Hamilton-operátor nem feltétlenül ekvivalens azzal a Hamilton-operátorral, aminek a $|GS(0)\rangle$ állapot az alapállapota.

A továbbiakban csak egy speciális esettel fogunk foglalkozni, amikor a Hamilton-operátor csak egy Majorana párt tartalmaz, később ezek fogják reprezentálni a koherens hibákat. (Az itt bemutatott formulák az általános esetre is könnyen átültethetőek.) Ilyenkor egy tetszőleges időpontban az adott állapot a következő:

$$|GS'\rangle = \exp(\theta \hat{c}_p \hat{c}_q) |GS\rangle. \quad (69)$$

Ilyenkor a kovarianciamátrix következő lesz:

$$M'_{jk} = \langle GS | \exp(-\theta \hat{c}_p \hat{c}_q) i \hat{c}_j \hat{c}_k \exp(\theta \hat{c}_p \hat{c}_q) |GS\rangle - i \delta_{jk}. \quad (70)$$

Ebben az esetben a kovarianciamátrix transzformációját végrehajtó algoritmus futásideje szintén $\mathcal{O}(N)$ nagyságrendben skálázódik.

Az utolsó tiszta Gaussi állapotot eredményező operáció a Majorana párok mérése. Ilyenkor egy projektív mérést képzelünk el, amely a következő állapotot hozza létre a (p,q) Majorana párok mérése esetén:

$$|GS'\rangle = \frac{\sqrt{2}}{\sqrt{\langle GS | (1 + i \hat{c}_p \hat{c}_q) |GS\rangle}} \frac{1}{2} (1 + i \hat{c}_p \hat{c}_q) |GS\rangle. \quad (71)$$

Ezt az eredményt akkor kapjuk, ha (p,q) pár mérése $+1$ lett. A -1 -es mérési eredmény nyilvánvalóan a (q,p) pár $+1$ -es mérési eredményének felel meg. A mérés utáni állapot kovarianciamátrixa a következő lesz:

$$M'_{jk}(|GS'\rangle) = \frac{\langle GS | (1 + i \hat{c}_p \hat{c}_q) i \hat{c}_j \hat{c}_k (1 + i \hat{c}_p \hat{c}_q) |GS\rangle}{2 \langle GS | (1 + i \hat{c}_p \hat{c}_q) |GS\rangle} - i \delta_{jk}. \quad (72)$$

A kovarianciamátrix kiszámolása ebben az esetben $\mathcal{O}(N^2)$ nagyságrendű időt vesz igénybe.

A felületi kód esetében a geometriai adottságok lehetővé teszik, hogy az általános esetben $\mathcal{O}(N^3)$ nagyságrendben skálázódó futási időt $\mathcal{O}(N^2)$ -re csökkentjük. Ennek pontos leírása és a kovarianciamátrix transzformálását leíró algoritmusok a dolgozat függelékében találhatóak meg.

4.3. A C4 kód

A felületi kód FLO szimulációjának egyik kulcsmozzanata a kód fermionikus rendszerként történő leírása, ehhez a felületi kód fizikai qubitjeit két-két fermionikus módusból álló úgynevezett C4 kódokba kódoljuk. Bár ez a fajta

bekódolás pusztán a szimulációkban segít minket, mégis erős fizikai motívációja van, hiszen ha a fizikai qubitek $|0\rangle$ és $|1\rangle$ állapotait megfeleltetjük egy kettős kvantumpöttyön elhelyezkedő fermion állapotainak, (a $|0\rangle$ állapot amikor a fermion a baloldali, az $|1\rangle$ állapot, amikor a jobboldali pöttyön van), akkor minden állapot eleget fog tenni a fermionszám-megmaradásnak.

A két kvantumpöttyhöz, amin a fermion elhelyezkedhet, tartoznak fermionikus keltő és eltüntető operátorok $\hat{a}_1, \hat{a}_2, \hat{a}_1^\dagger, \hat{a}_2^\dagger$. Vezessük be a C4 kód Majorana operátorait a következő módon:

$$\begin{aligned} \hat{c}_1 &= i(\hat{a}_2 - \hat{a}_2^\dagger) & \hat{c}_2 &= \hat{a}_1 + \hat{a}_1^\dagger \\ \hat{c}_3 &= i(\hat{a}_1 - \hat{a}_1^\dagger) & \hat{c}_4 &= \hat{a}_2 + \hat{a}_2^\dagger. \end{aligned} \quad (73)$$

Ez a definíció nem konzisztens a Majorana operátorok általános (53)-es definíciójával, de ez csak azért van, hogy a számozás később egyszerűbb legyen.

A fizikai qubitek általános állapota a fermionikus módusok betöltési szám reprezentációjában a következő alakban áll elő:

$$|\psi\rangle_{C4} = \alpha|01\rangle + \beta|10\rangle, \quad (74)$$

ezt az alakot nevezhetjük a C4 kód kódterének is, hiszen csak ez az alak kódol valódi tartalmat.

Bevezethetjük a fermionikus módusokhoz tartozó paritás operátorokat \hat{p} , amelyek várható értéke 1, ha a módus nincs betöltve, és -1 , ha be van,

$$\hat{p}_1 = 1 - 2\hat{a}_1^\dagger\hat{a}_1 \quad \hat{p}_2 = 1 - 2\hat{a}_2^\dagger\hat{a}_2. \quad (75)$$

Könnyen beláthatjuk, hogy a fizikai qubit tetszőleges, (74) szerinti állapota -1 -es sajátértékkel sajátállapota a paritás operátorok szorzatának. Ilyen módon bevezethetjük az \hat{S} operátort, ami stabilizálja C4 kódot, azaz ellenőrzi a fermionszám paritását. (Ez a jelen esetben megegyezik magával a fermionszámmal, azaz a \hat{S} operátor saját alterében csak olyan állapotok vannak, ahol 1 fermion van a rendszerben. A C4 kód akkor is működne, ha nem a páratlan fermionszámú alter lenne az \hat{S} stabilizátor sajátaltere, hanem a páros, ilyenkor a fermionszám nem maradna meg, csak a paritás.)

$$\hat{S}|\psi\rangle_{C4} = -\hat{p}_1\hat{p}_2|\psi\rangle_{C4} = |\psi\rangle_{C4}. \quad (76)$$

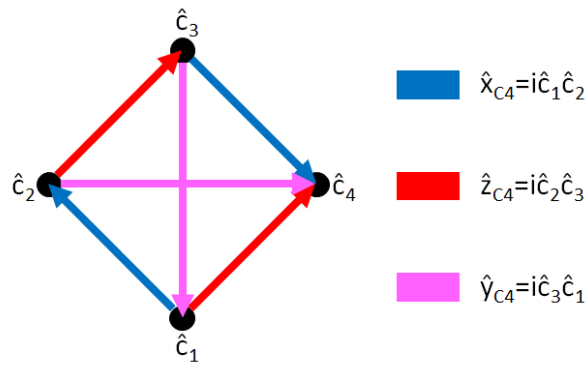
A C4 kód \hat{S} stabilizátora a Majorana fermionokkal a következő alakban írható fel:

$$\hat{S} = -\hat{c}_1\hat{c}_2\hat{c}_3\hat{c}_4. \quad (77)$$

A fizikai qubitek Pauli-operátorait is könnyen megtalálhatjuk, ezek a Majorana operátorokkal kifejezve a következőknek választhatók:

$$\hat{X}_{C4} = i\hat{c}_1\hat{c}_2 \quad \hat{Z}_{C4} = i\hat{c}_2\hat{c}_3 \quad \hat{Y}_{C4} = i\hat{c}_3\hat{c}_1. \quad (78)$$

Fontos észrevétel, hogy C4 kód Pauli-operátorai kommutálnak az \hat{S} stabilizátorral és az $\hat{S}\hat{X}_{C4}$, $\hat{S}\hat{Z}_{C4}$, $\hat{S}\hat{Y}_{C4}$ operátorok is megfelelő Pauli-operátorokként hatnak a C4 kód kódterén.

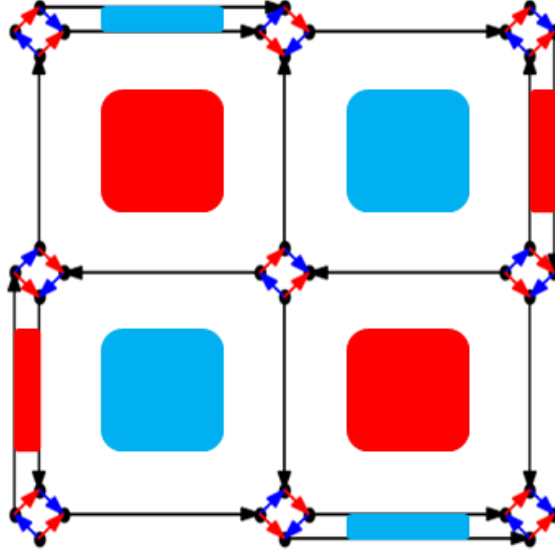


11. ábra. A C4 kód sematikus ábrája. A Majorana párokat nyilak reprezentálják, amik az első Majorana operátortól a második felé mutatnak. A piros nyilak az \hat{X}_{C4} és $\hat{S}\hat{X}_{C4}$ operátorok, a kék nyilak a \hat{Z}_{C4} és $\hat{S}\hat{Z}_{C4}$ operátorok és a rózsaszín nyilak pedig a \hat{Y}_{C4} és $\hat{S}\hat{Y}_{C4}$ operátorok.

4.4. A felületi kód Majorana hálózata

A következő feladat, hogy a teljes felületi kódot felépítsük a C4 kódokba kódolt fizikai qubitekből. Ehhez meg kell találnunk a felületi kód stabilizátorainak és logikai operátorainak Majorana-reprezentációját.

Első lépésként a C4 kódok orientációját úgy állítjuk be, hogy a felületi kód lapjainak szélén elhelyezkedő Majorana párok egységesen X-, vagy Z-operátorok legyenek, az adott lapon definiált stabilizátor típusának függvényében. Ez gyakorlatilag annyit jelent, hogy minden második C4 kódot 90° -kal el kell forgatni a (11)-es ábrához képest.



12. ábra. A 3-mas kódtávolságú felületi kód Majorana hálózata. A fekete pontok reprezentálják a Majorana fermionokat, a kék, illetve piros nyilak a C4 kódok Z és X-operátorait, a kék és piros négyzetlapok a felületi kód Z- és X-stabilizátorait, a fekete nyilak pedig az éloperátorokat.

A következő lépés a felületi kód stabilizátorainak reprezentálása a Majorana operátorok segítségével. Könnyen beláthatjuk, hogy a stabilizátorok az adott lap határán elhelyezkedő Majorana párok szorzatai lesznek, hiszen az elrendezés úgy van kialakítva, hogy például egy X-stabilizátor lap határán csak \hat{X}_{c_4} vagy $\hat{S}\hat{X}_{c_4}$ operátorok legyenek (Z-stabilizátorokra ugyanez igaz).

Bevezethetjük az úgynevezett *éloperátorokat*, amelyek a felületi kód élein definiált Majorana párok. Ezek nem tartoznak egyik C4 kódhoz sem és egy adott e élen a következőképpen írhatjuk fel őket:

$$\hat{L}_e = i\hat{c}_p\hat{c}_q, \quad (79)$$

ahol \hat{c}_p és \hat{c}_q az adott él két végpontja. Az éloperátort irányát, a Majorana operátorok sorrendjét egy kézenfekvő feltétel határozza meg, még hozzá az, hogy a stabilizátorok legyenek egyenlőek az adott lap élein definiált éloperátorok szorzatával,

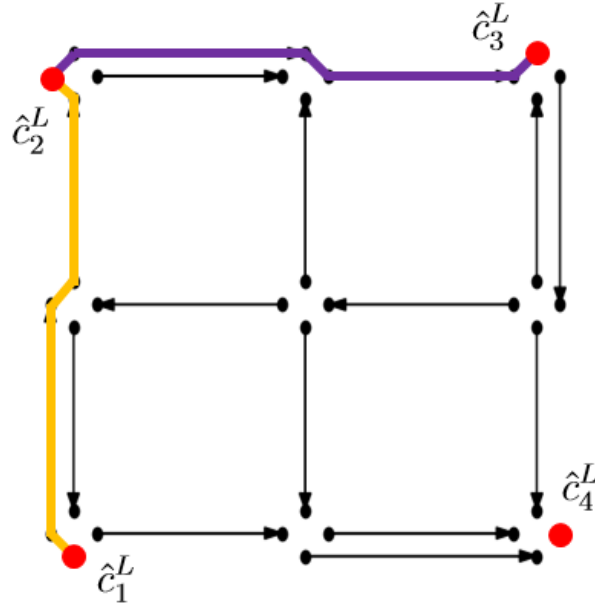
$$\hat{S}_f = \prod_{e \in \partial f} \hat{L}_e. \quad (80)$$

Ezt feltételt többféle éloperátor kiosztás is meg tudja valósítani, ezek közül az egyik látható a (12)-es ábrán. Itt 3 kódtávolságú kód látható, de

minden nagyobb kód is felépíthető ebből az alapegységből. A kiosztás a következő: a horizontális éloperátorok egy adott sorban egy irányba mutatnak, de soronként változik ez az irány, ugyanez igaz a vertikális éloperátorokra is. Az éllapok külső éloperátorai pedig lent és fent megegyeznek az éllap másik éloperátorának irányával, bal és jobb oldalon pedig eltérnek attól.

Megfigyelhetjük, hogy minden Majorana fermionhoz csatlakozik egy éloperátor, kivéve 4 darab a sarkokban elhelyezkedő Majorana fermiont. Ezen kimaradó Majorana operátorok segítségével definiálhatjuk a felületi kód logikai operátorait. A logikai X-operátor a felületi kód bal oldalán elhelyezkedő X-operátorok szorzata, a logikai Z-operátor pedig a kód tetején lévő Z-operátoroké. Ezt átírva az éloperátorok és a kimaradó $\hat{c}_1^L, \hat{c}_2^L, \hat{c}_3^L, \hat{c}_4^L$ Majorana operátorok nyelvére a következő kifejezéseket kapjuk:

$$\hat{X}^L = i\hat{c}_1^L\hat{c}_2^L \prod_{e \in LEFT} \hat{L}_e \quad \hat{Z}^L = i\hat{c}_2^L\hat{c}_3^L \prod_{e \in TOP} \hat{L}_e. \quad (81)$$



13. ábra. A 3-mas kódtávolságú felületi kód éloperátorai és a logikai operátorok. A logikai X-operátor (narancs), a logikai Z-operátor (lila). A logikai C4 kód Majorana operátorai (piros pontok).

Az éloperátorok által érintett $4n-4$ Majorana fermionból álló rendszernek definiálhatjuk egy kitüntetett állapotát, az úgynevezett *link-állapotot* $|\phi_{link}\rangle$, amelyik az összes éloperátor által stabilizált állapot,

$$\hat{L}_e|\phi_{link}\rangle = |\phi_{link}\rangle \quad \forall e. \quad (82)$$

Ez az állapot egy tiszta Gaussi állapot, hiszen Majorana párok által van stabilizálva, és a felületi kód összes stabilizátorának sajátállapota +1-es sajátértékkel, hiszen a stabilizátorok az éloperátorok szorzatai.

Sajnos azonban van egy nagy gond ezzel az állapottal, nem a C4 kódok alterében van, így nem sajátállapota a C4 kódok stabilizátorainak.

A következő feladat a felületi kódban tárolt logikai információ megtalálása, ehhez vizsgáljuk meg a $|\phi_{link}\rangle$ állapotból kimaradt 4 Majorana fermiont. Ezekből alkothatunk egy úgynevezett *Logikai C4 kódot*, ami ugyanúgy egy C4 kód operátor kiosztásával rendelkezik, de a Majorana fermionok nincsenek egy helyre lokalizálva, a felületi kód 4 sarkában helyezkednek el. Az így definiált logikai C4 kód tetszőleges állapota $|\psi_{C4}^L\rangle$ is tiszta Gaussi állapot.

A felületi kód egy tetszőleges logikai állapota a következő alakban írható fel:

$$|\psi_L\rangle = 2^{(n-1)/2} \prod_{j=1}^n \frac{1}{2} (1 + \hat{S}_j) |\phi_{link}\rangle \otimes |\psi_{C4}^L\rangle, \quad (83)$$

itt \hat{S}_j a j-edik qubit C4 kódjának stabilizátora. A logikai állapot kifejtésében egy normáló faktoron kívül megjelenik a C4 kódok alterébe történő projektálás. Könnyen beláthatjuk, hogy ez az állapot már mind a C4 kód stabilizátorok, mind a felületi kód stabilizátorok sajátállapota lesz, valamint a (81)-os egyenletben definiált logikai operátorok is megfelelő módon hatnak az állapoton, hiszen a logikai operátorok felcserélnek mind a C4 stabilizátorokkal, mind az éloperátorokkal, így végső soron csak a logikai C4 kód állapota változik.

Fontos észrevétel ugyanakkor, hogy sajnos a felületi kód logikai állapotai nem lesznek tiszta Gaussi állapotok hiszen a C4 kódok alterébe történő projektálás nem tartozik a megengedett műveletek közé.

4.5. A végállapot-eloszlás mintavételezése

A szimulációs módszer lényege, hogy a felületi kód koherens hibákhoz tartozó hibajavítás utáni végállapotait (40) mintavételezzük a lehetséges szindrómák valószínűsége alapján. Ezután a kiolvasási hibákat már könnyen be tudjuk építeni a szimulációba, hiszen a (45)-ös egyenletben a felületi kód végállapota csak a $\hat{C}_{s'}$ korrekciós operátoron keresztül függ a kiolvasási hibáktól és ennek a megkonstruálása már hatékony lesz.

A koherens hibák és kiolvasási hibák szimulációja a következő lépésekben foglalható össze:

- \underline{s} tökéletes szindrómák generálása a $p(\underline{s})$ eloszlás alapján.

- A koherens hibákhoz tartozó $\theta^*(\underline{s})$ forgatási szög kiszámítása az generált \underline{s} szindrómák alapján.
- Az \underline{s} tökéletes szindrómákból \underline{s}' hibás szindrómák generálása a $p(\underline{s} \rightarrow \underline{s}')$ eloszlás alapján.
- θ_L logikai forgatási szög kiszámítása a (48)-es képlet alapján.

Az első két pont tulajdonképpen csak a koherens hibák szimulálását jelenti, hiszen \underline{s} d darab szindrómát koherens hibák esetén mért szindrómát tartalmaz, θ^* pedig megkonstruálható az egyes s^j szindrómákhoz tartozó θ_{s^j} logikai forgatási szögekből.

A szindrómák generálása elég nehéz feladat az FLO szimuláció keretei közt, hiszen egyrészt a felületi kód logikai állapotai nem tiszta Gaussi állapotok, illetve a felületi kód stabilizátorainak mérése nem egy megengedett művelet, hiszen 2-nél hosszabb Majorana láncok mérését jelentené. Erre a problémára az jelent megoldást, hogy az X-stabilizátorok helyett a fizikai qubitek X-operátorait mérjük meg, hiszen a szindrómák mintavételezése visszavezethető az X-operátorok mérési eredményeinek mintavételezésére.

Vezessük be az $m = \{m_j\}$ szindrómákat, ahol m_j megegyezik a j -edik qubit \hat{X}_j operátorának mért eredményével. Minden m szindrómához tartozik egy s szindróma, de egy s szindrómához több m szindróma is tartozik a következő módon:

$$s = \delta(m) : \quad s_f = \prod_{j \in \partial f} m_j, \quad (84)$$

ahol s_f az s szindróma értéke a felületi kód f lapján, amin egy X-stabilizátor helyezkedik el. Ilyenkor s szindróma mérésének valószínűsége a következő alakban írható fel az m szindrómák mérési valószínűségével:

$$p(s) = \sum_{\forall m: \delta(m)=s} p(m). \quad (85)$$

Ez azt jelenti, hogy ha $p(m)$ eloszlás szerint mintavételezünk egy m szindrómát, majd kiszámoljuk az ahhoz tartozó $\delta(m) = s$ szindrómát, az ekvivalens azzal, mintha egy s szindrómát mintavételeznénk $p(s)$ eloszlás szerint.

Fontos megjegyezni, hogy ez csak olyan esetekben működik, ahol a Z-stabilizátorokat nem kell megmérnünk, azaz a hiba felcserél ezekkel. Olyan esetekben, ahol minden stabilizátort meg kell mérnünk a mintavételezés már nem vezethető vissza a fizikai qubitek operátorainak mérésére.

Az m szindrómák mintavételezése egy Monte Carlo algoritmus alapján működik, ahol sorba rendezzük a fizikai qubiteket és a j -edik mérési eredményt, m_j -t a $p(m_j|m_1, m_2, \dots, m_{j-1})$ feltételes valószínűség alapján sorsoljuk ki.

$$p(m_j|m_1, m_2, \dots, m_{j-1}) = \frac{p(m_1, m_2, \dots, m_j)}{p(m_1, m_2, \dots, m_{j-1})}, \quad (86)$$

itt legyen $p(m_A) = p(m_1, m_2, \dots, m_j)$ annak a valószínűsége, hogy az első j qubitnél m_A rész-szindrómát mértünk, azaz $p(m)$ egy marginális eloszlása.

$p(m_A)$ a következő alakban írható:

$$p(m_A) = \langle \psi_L^0 | \hat{E}_A^{\dagger coh} \hat{\Pi}_{m_A}^x \hat{E}_A^{coh} | \psi_L^0 \rangle, \quad (87)$$

ahol a hiba és az X-mérések csak az első j qubiten hatnak.

$$\hat{E}_A^{coh} = \prod_{k=1}^j e^{i\theta_k \hat{Z}_k} \quad \hat{\Pi}_{m_A}^x = \prod_{k=1}^j \frac{1}{2}(1 + m_k \hat{X}_k). \quad (88)$$

Írjuk fel a kezdeti állapotot a (83)-as egyenlet alapján és ez legyen a logikai $|+L\rangle$ állapot, hiszen $p(m_A)$ független a kiinduló logikai állapottól és ezt könnyű inicializálni.

$$p(m_A) = 2^{n-1} \langle \phi_{link}, +_{C4}^L | \hat{\Omega} \hat{E}_A^{\dagger coh} \hat{\Pi}_{m_A}^x \hat{E}_A^{coh} \hat{\Omega} | \phi_{link}, +_{C4}^L \rangle, \quad (89)$$

itt $\hat{\Omega}$ a C4 kódok alterébe projektáló projektor. Természetesen itt is definiálhatjuk az első j qubitra ható projektort.

$$\hat{\Omega} = \prod_{j=1}^n \frac{1}{2}(1 + \hat{S}_j) \quad \hat{\Omega}_A = \prod_{k=1}^j \frac{1}{2}(1 + \hat{S}_k). \quad (90)$$

Több átalakítás után (a Bravyi cikk függelékében ez is megtalálható [5]) ezt a kifejezést a következő alakra hozhatjuk:

$$p(m_A) = \nu \langle \phi_{link}, +_{C4}^L | \hat{O}_1^\dagger \dots \hat{O}_j^\dagger \hat{O}_j \dots \hat{O}_1 | \phi_{link}, +_{C4}^L \rangle, \quad (91)$$

ahol $\nu = 2^j$, ha $j < n$ és $j = n$ esetén pedig $\nu = 2^{n-1}$. Az \hat{O} operátorok pedig a következő alakban írhatók:

$$\hat{O}_k = \frac{1}{2}(1 + m_k \hat{X}_k) \frac{1}{2}(1 + m_k \hat{S}_k \hat{X}_k) e^{i\theta_k \hat{Z}_k}. \quad (92)$$

Fontos észrevétel, hogy mind az $\hat{X}_k, \hat{S}_k \hat{X}_k$ és \hat{Z}_k operátorokat Majorana párok valósítják meg a felületi kód Majorana hálózatában, így az \hat{O}_k operátorok hatása a kezdeti tisztán Gaussi $|\phi_{link}, +_{C4}^L\rangle$ állapoton megengedett műveletek, és így az állapot végig tisztán Gaussi marad.

A következő feladat a meglévő s szindróma alapján kiszámolni a θ_s logikai forgatási szöget. Belátható, hogy ez kiszámolható speciális, $p(m)$ szindróma valószínűségként, inhomogén fizikai forgatási szögeket feltételezve.

Ennek a kiszámításához a következő képletek segítenek hozzá, (ezen képletek levezetését most nem részletezem, de a Bravyi cikk [5] függelékében megtalálható.)

$$\begin{aligned}\tan^2(\theta_s) &= \frac{p_-}{p_+} = \frac{\langle +_L | \hat{E}_-^\dagger \hat{\Pi}_0 \hat{E}_- | +_L \rangle}{\langle +_L | \hat{E}_+^\dagger \hat{\Pi}_0 \hat{E}_+ | +_L \rangle} \\ \tan^2(\theta_s - \pi/4) &= \frac{q_-}{q_+} = \frac{\langle Y_L | \hat{E}_-^\dagger \hat{\Pi}_0 \hat{E}_- | Y_L \rangle}{\langle Y_L | \hat{E}_+^\dagger \hat{\Pi}_0 \hat{E}_+ | Y_L \rangle},\end{aligned}\tag{93}$$

itt $\hat{\Pi}_0$ abba az állapotba projektál, ahol minden stabilizátormérés eredménye $+1$, $|Y_L\rangle$ az \hat{Y}^L operátor sajátállapota, \hat{E}_- és \hat{E}_+ definíciói pedig a következők:

$$\hat{E}_- = \hat{Z}^L \hat{C}_s \hat{E}^{coh} \quad \hat{E}_+ = \hat{C}_s \hat{E}^{coh}.\tag{94}$$

A (93)-as egyenlet átrendezésével a következő összefüggéseket kapjuk:

$$\sin(2\theta_s) = \frac{q_+ - q_-}{q_+ + q_-} \quad \cos(2\theta_s) = \frac{p_+ - p_-}{p_+ + p_-}.\tag{95}$$

Ez alapján a logikai forgatás szöge egy π faktor pontosságáig meghatározható.

4.6. A szimuláció erőforrásigénye

Az általam írt python kódot, ami megvalósítja a koherens hibák és a kiolvasási hibák szimulációját, elérhetővé tettem [1]. A kódban a korrekciós operátorok előállítását a PyMatching csomaggal [15] végeztem el. A szimulációkat egy 24 magos clusteren futattam.

A tökéletes kiolvasások meletti koherens hibák és a kiolvasási hibákat is figyelembe vevő python scriptek futási idejét, néhány jellemző kódtávolság esetén az alábbi táblázatokban foglaltam össze:

Kódtávolság	5	7	9	11	13	15	17	19
Futási idő (s)	0.10	0.23	0.44	0.7	0.8	1.3	1.5	1.9

1. táblázat. A koherens hibák szimulációjához tartozó python script futási ideje. A feladat egy θ_s logikai forgatási szög mintavételezése a $p(s)$ eloszlásból.

Kódtávolság	5	7	9	11	13	15	17	19
Futási idő (s)	0.6	1.8	4.0	7.2	9.8	19.1	26.0	39.8

2. táblázat. A koherens hibák és kiolvasási hibák szimulációjához tartozó python script futási ideje. A feladat egy θ_L logikai forgatási szög mintavételezése a $p(\underline{s})p(\underline{s} \rightarrow \underline{s}')$ eloszlásból.

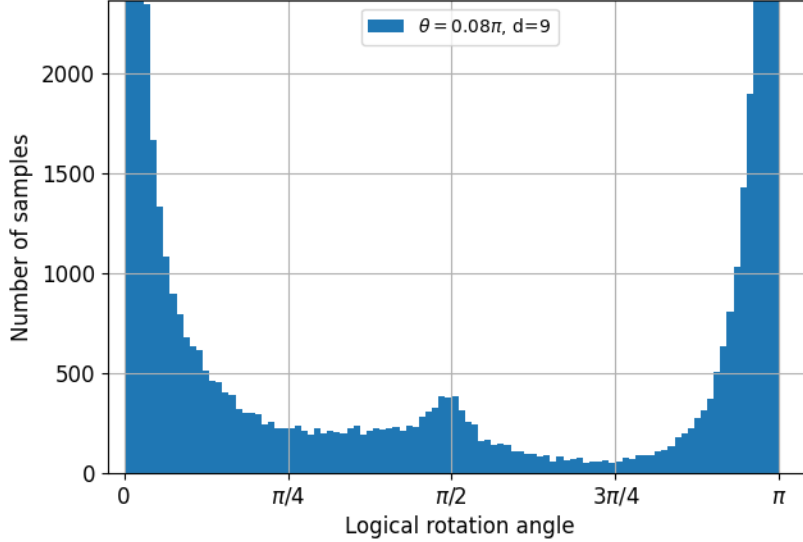
5. Eredmények

A koherens hibák hatását tökéletes kiolvasások mellett már részletesen megvizsgálták [5, 26], azonban a legfontosabb eredményeket reprodukáltam a saját kódom segítségével és néhány a korábbi munkákból kimaradt szempontot is megvizsgáltam. A fejezet első részében ezeket az eredményeket mutatom be, hiszen ezek elengedhetetlenek a koherens hibák viselkedésének pontos megértéséhez.

A fejezet második részében pedig a koherens hibák és a kiolvasási hibák együttes hatását vizsgáló szimulációim eredményeit ismertetem, különös tekintettel azokra az aspektusokra, ahol a kiolvasási hibák jelenléte nagyban befolyásolja a koherens hibák hatását.

5.1. Koherens hibák, tökéletes kiolvasás

A koherens hibák hatásának szimulációja a következő sematikus képpel illusztrálható: a szimuláció bemenete a koherens Z -forgatásokhoz tartozó fizikai forgatási szög (θ) és a felületi kód kódtávolsága d . Ilyenkor a szimuláció kimenete egy $p(s)$ eloszlás szerint generált θ_s logikai forgatási szög.



14. ábra. A θ_s eloszlása $\theta = 0.08\pi$, $d=9$ értékeknél. Az ábrán a 0 és π körüli csúcsok le vannak vágva, ezek valójában sokkal magasabbak. Az adatok 96.000 Monte Carlo kör eredményeit tartalmazzák.

Az ábrán jól láthatjuk, hogy a 0 és a π értékek körüli logikai forgatási szögek a leggyakoribbak, ezek annak felelnek meg, amikor a felületi kód sikeresen kijavította a hibát és nem forog el a kezdeti állapot. Ezen felül azt is megfigyelhetjük, hogy a $\pi/2$ -es érték körül egy kisebb csúcs alakul ki, ez a logikai Z-hibának felel meg. Egy szemléletes képpel élve azt mondhatjuk, hogy a felületi kód bizonyos fokig "kimossa" a koherenciát a környezeti zajból, azaz a fizikai szinten megjelenő koherens hibákat a hibajavítási procedúra a logikai szinten bizonyos valószínűséggel előforduló Z hibákká alakítja. Természetesen ez egy túlzó kép, azonban ezt a koherencia "kimosást" matematikailag is megvizsgálhatjuk.

Vezessük be a *koherenciaráta* fogalmát, amelyet a logikai ε_L hibacsatornára a következő módon definiálunk:

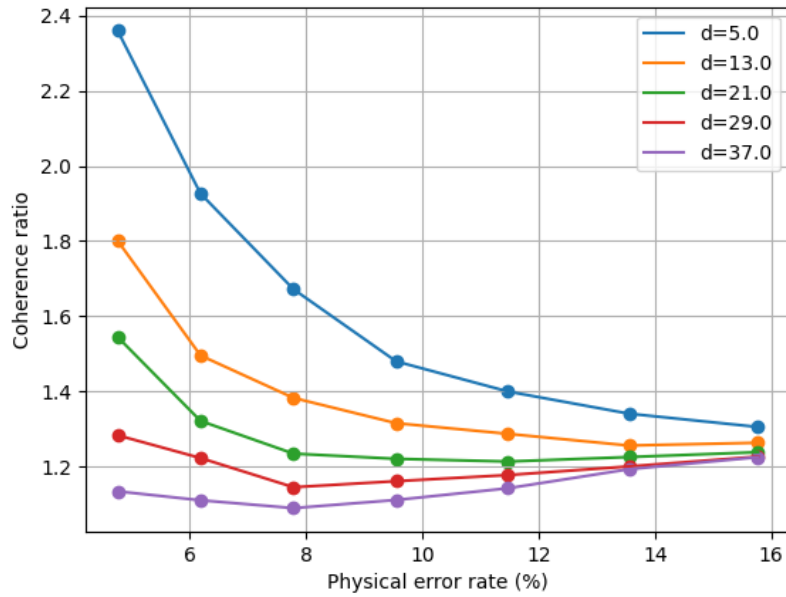
$$K(\varepsilon_L) = \frac{\|\varepsilon_L\|_{\diamond}}{\|\varepsilon_L^{twirl}\|_{\diamond}}. \quad (96)$$

Ez a mennyiség azt hivatott mérni, hogy a valódi hibacsatorna mennyire van távol a Pauli-pödört verziójától, azaz mennyire alakítja át a hibákat Pauli-hibákká.

A koherenciaráta a mi esetünkben a (49) egyenletek alapján következőképpen néz ki:

$$K(\varepsilon_L^{coh}) = \frac{\sum_s p(s) |\sin(\theta_s)|}{\sum_s p(s) \sin^2(\theta_s)}, \quad (97)$$

ezt a kifejezést megvizsgálva láthatjuk, hogy ez mindig egy 1-nél nagyobb szám, és abban az esetben, ha θ_s csak $\{0, \pi/2, \pi\}$ értékeket vesz fel, a koherenciaráta pontosan 1.



15. ábra. A koherencia ráta a fizikai hibaráta függvényében különböző kód-távolságok esetén. Az ábrán látható koherenciarátákhoz tartozó gyémánt normákat 5.000 Monte Carlo kör alapján határoztam meg.

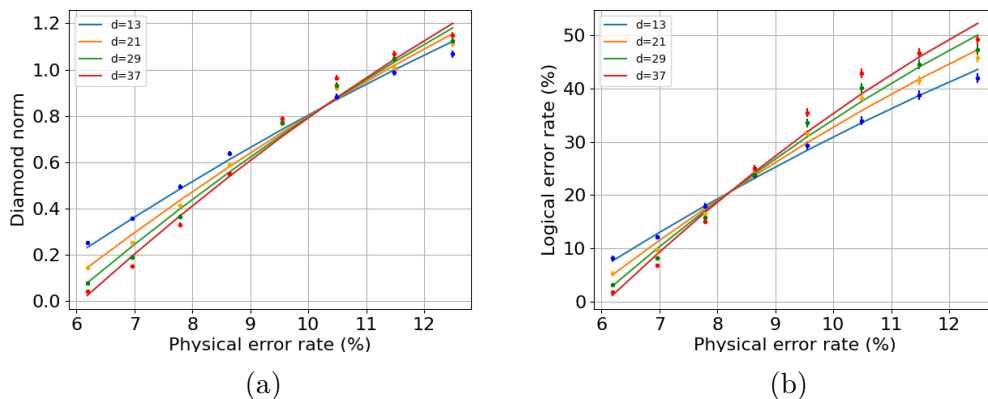
Az ábrán a koherenciarátát nem a θ fizikai forgatási szög függvényében ábrázoltam, hanem a $\sin^2(\theta)$ fizikai hibaráta függvényében. Ez a mennyiség a Pauli-hibákkal való összevetés során lesz igazán hasznos, hiszen ez pontosan a fizikai hibacsatornák Pauli-pödört verziójának a p fizikai hiba valószínűsége.

Az is jól látszik a (15)-ös ábrán, hogy az egyre nagyobb kódméretnek esetén a koherenciaráta egyre kisebb, így a koherencia egyre jobban "kimosódik" a logikai hibákból és ez nagy fizikai hibák esetén egy állandó, 1-nél kicsit nagyobb, értékhez tart.

A koherens hibák hatásának karakterizálásához nagyon fontos lépés a hibaküszöb megállapítása. Ehhez először is definiálnunk kell, hogy a koherens

hibák esetén, hogyan mérjük a hibajavítás hatékonyságát. Ezt megtehetjük a P_L logikai hibaráta és a $\|\varepsilon_L^{coh}\|_\diamond$ gyémánt norma (41) segítségével is. Mindkét mérték visszaadja a Pauli-hibacsatorna esetén a szokásos logikai hiba valószínűsége, ezért jó kiterjesztései lehetnek ennek, bonyolultabb hibák esetére. Első ránézésre a logikai hibaráta tűnik a megfelelő mértéknek, azonban az irodalomban sokszor a gyémánt normát használják [5, 26]. A felületi kód szempontjából a hibajavítás hatékonyságának kvantifikálása igen fontos feladat, azonban a megfelelő mérték használata nem egyértelmű, mindkét mérték használható, én éppen ezért mindkét esetet megvizsgáltam.

A különböző méretű kódok segítségével elvégzett szimulációk alapján mindkét esetre meghatároztam a hibaküszöb értékét, az inkoherens esethez hasonlóan (9) itt is a már korábban hivatkozott illesztési eljárással [27] állapítottam meg a pontos értékeket. (Az illesztett görbék itt nem pontosan illeszkednek a szimulációs pontokra, ennek talán az az oka, hogy az illesztés alapjául szolgáló elméleti leírás a koherens hibák esetén nem pontosan úgy működik, mint az inkoherens esetben. Ez további érdekes elméleti kérdéseket vet fel, de ezek túlmutatnak a dolgozat keretein.)



16. ábra. A koherens hibák által okozott logikai hibacsatorna gyémánt normája (16a) és logikai hibaráta (16b) a fizikai hibaráta függvényében. Az ábrákon lévő pontokat 5.000 Monte Carlo kör alapján állapítottam meg.

Az illesztés alapján a p_{th}^d gyémánt norma esetén és a p_{th}^i logikai hibaráta esetén meghatározott hibaküszöbök pontos értéke a következőnek adódott:

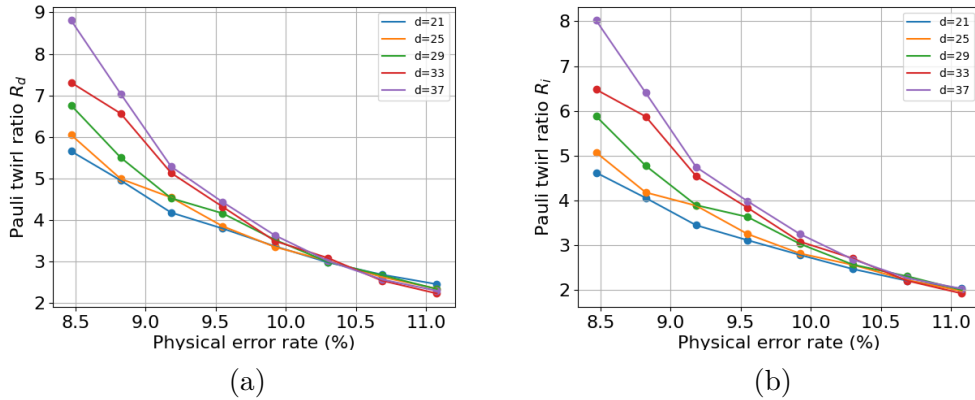
$$p_{th}^d = 10.28\% \pm 0.25\% \quad p_{th}^i = 8.20\% \pm 0.12\%. \quad (98)$$

Az hibaküszöbök értékei igen érdekes viselkedést mutatnak, megfigyelhetjük, hogy a gyémánt norma esetén jóval magasabb a hibaküszöb értéke, mint a

logikai hibaráta esetén. A p_{th}^d érték hibahatáron belül megegyezik a fizikai hibacsatornák Pauli-pödört verziójánál meghatározott hibaküszöbvel (17). Ez a korábbi munkákban [5, 26] is egy központi állítás, azonban logikai hibarátánál meghatározott p_{th}^i hibaküszöb alacsony értéke kétségeket vethet fel a koherens hibák viselkedésével kapcsolatban.

A szimulációk eredményéből az derül ki, hogy a hibaküszöb a koherens hibák esetén nagyságrendileg azonos az inkohereus esettel (a logikai hibarátát vizsgálva számottevően alacsonyabb). A hibaküszöb alatti értékek esetén hibajavítási hatékonyságot is érdemes megvizsgálni, hiszen a koherens és inkohereus hibák közti különbségek itt lehetnek igazán fontosak. Ehhez a $P_L(\varepsilon^{coh})$ logikai hibarátát és a $\|\varepsilon_L^{coh}(\varepsilon^{coh})\|_\diamond$ gyémánt normát érdemes az inkohereus esettel összevetni, azaz a fizikai hibacsatornák Pauli-pödört verziójánál számolt $P_L(\varepsilon_{twirl}^{coh})$ logikai hibarátával és $\|\varepsilon_L(\varepsilon_{twirl}^{coh})\|_\diamond$ gyémánt normával. Így bevezethetjük a R_d és R_i Pauli-pödörös arányokat, ami a koherens hibák és a hozzájuk tartozó inkohereus hibák okozta hibajavítási hatékonyságokat veti össze.

$$R_d = \frac{\|\varepsilon_L^{coh}(\varepsilon^{coh})\|_\diamond}{\|\varepsilon_L(\varepsilon_{twirl}^{coh})\|_\diamond} \quad R_i = \frac{P_L(\varepsilon^{coh})}{P_L(\varepsilon_{twirl}^{coh})}. \quad (99)$$



17. ábra. A Pauli-pödörös arányok, a gyémánt norma (17a) és a logikai hibaráta (17b) esetén a fizikai hiba ráták függvényében. Az ábrákon látható pontokat 5.000 Monte Carlo kör eredményei alapján határoztam meg.

Az ábrákon jól látszik, hogy a két mérték között nem tapasztalható jelentős különbség, hasonlóan viselkednek. Mindkét ábrán jól látható, hogy a kis fizikai hibaráta esetén a koherens hibák jelentősen veszélyesebbek is lehetnek, mint az inkohereus hibák. A koherens hibákhoz tartozó logikai hibaráta sokszorosai is lehetnek az inkohereus hibákhoz tartozóknak és ez az

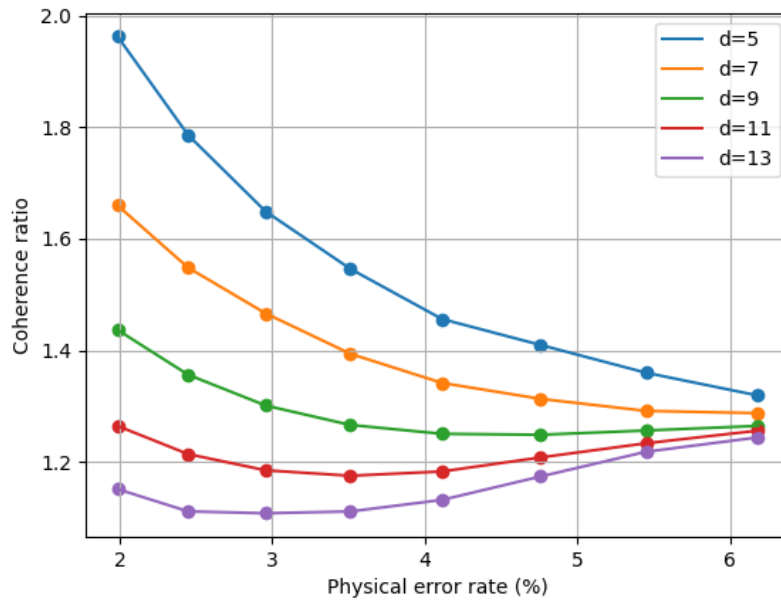
arány a kódtávolság növelésével egyre jobban nő. Ez azt jelenti, hogy bár a koherens hibák a felületi kódra jellemző hibaküszöb értékeket nem tolják jelentősen lejjebb, de a hibaküszöb alatti tartományban nagy kódok esetén jelentősen veszélyesebbek, mint az inkohereus hibák.

5.2. A koherens hibák és a kiolvasási hibák együttes hatása

A kiolvasási hibák megjelenése megváltoztatja a szimuláció paramétereit, ebben az esetben a d kódtávolságon kívül két független bemeneti változónk lesz: a koherens hibákra jellemző θ fizikai forgatási szög és p a kiolvasási hiba valószínűsége. Ilyenkor a logikai hibát jellemző $\theta_L(\underline{s}, \underline{s}')$ logikai forgatási szög a tökéletes és a hibás szindrómáktól is fog függeni és a szimulációban a $p(\underline{s})p(\underline{s} \rightarrow \underline{s}')$ eloszlás alapján mintavételezzük.

Az általam vizsgált esetben $p = \sin^2(\theta)$ volt kiolvasási hiba valószínűsége, így csak egy bemeneti paraméter (θ) határozta meg a problémát. (A $p = \sin^2(\theta)$ választás mögött közepesen helytálló fizikai megfontolások rejlenek, hiszen az áramköri szinten megjelenő koherens hibák Pauli-pödört közelitésénél a kiolvasási hibák ezzel a valószínűséggel jelennek meg.) Természetesen a kimerítő vizsgálat az lenne, ha egy adott kódtávolságnál tetszőleges (θ, p) párra meghatároznánk a logikai forgatási szög eloszlását, azonban ez az irány még egyelőre kidolgozás alatt áll.

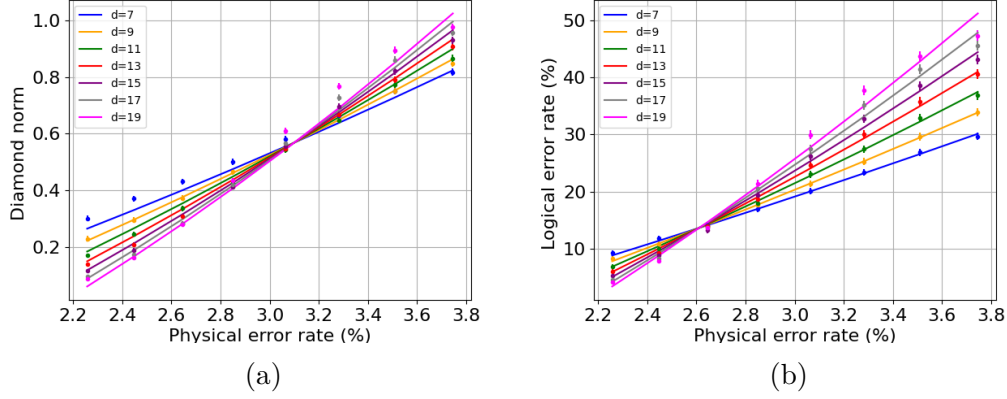
A kiolvasási hibák figyelembevételével kapott szimulációs eredmények esetén érdemes megvizsgálni a felületi kód viselkedését a koherens hibáknál tárgyalt szempontok alapján. Először is nézzük meg az itt tapasztalt koherencia rátát (96).



18. ábra. A koherencia ráta a fizikai hibaráta $\sin^2(\theta)$ függvényében (itt ez a kiolvasási hiba valószínűség is) különböző kódtávolságok esetén. A z ábrán látható pontokat $5.000 \times 100 \times d$ Monte Carlo kör alapján határoztam meg, azaz minden kód esetén 1 hibajavító körben d -kör koherens hibát szimuláltam és minden tökéletes szindrómához 100 hibás szindrómát generáltam.

Az ábrán jól látható a koherens esethez hasonló viselkedés (15), azzal a különbséggel, hogy a koherenciaráta itt már sokkal kisebb fizikai forgatási szögek esetén is kis értékeket vesz fel. Ez alapján a koherencia "kimosódása" a logikai hibákból itt is tetten érhető, talán még erősebben, mint kiolvasási hibák nélkül. Ennek oka, hogy itt d kör koherens hiba és mérés történik a logikai forgatási szög megállapítása előtt, illetve, hogy a kiolvasási hibák eleve inkohereusnak tekinthetők.

A hibamodellre jellemző hibaküszöböket itt is kétféleképpen állapítottam meg, a logikai hibaráta és a gyémánt normát is ábrázolva. A hibaküszöbök megállapítása itt is a korábban már használt illesztés alapján történt [27].



19. ábra. A koherens és kiolvasási hibákhoz tartozó logikai hibacsatorna gyémánt normája (19a) és logikai hibaráta (19b) a fizikai hibaráta függvényében.

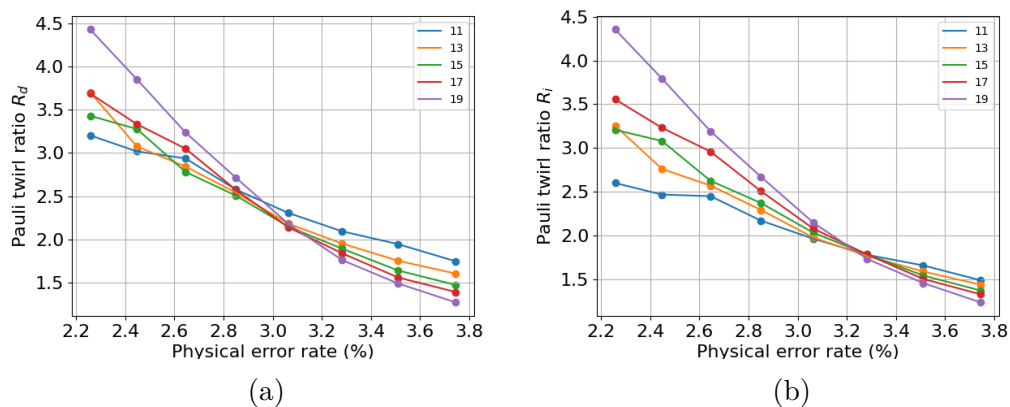
Az illesztések alapján a koherens és kiolvasási hibákhoz tartozó küszöbértékek a következőknek adódtak:

$$p_{th}^d = 3.10\% \pm 0.05\% \quad p_{th}^i = 2.62\% \pm 0.02\%. \quad (100)$$

Ezek az értékek a koherens hibákhoz hasonló viselkedést mutatnak, hiszen itt is egy magasabb hibaküszöb érték tartozik a gyémánt normához, mint a logikai hibarátához. Az inkoherens esettel történő összevetés során azonban egy érdekes különbséget figyelhetünk meg, a tökéletes kiolvasásokkal szemben itt az inkoherens esetnél meghatározott hibaküszöb (18) a két érték közé esik és mind három érték viszonylag közel van egymáshoz. A koherens hibák esetén nem, de a kiolvasási hibák figyelembevételével kijelenthetjük, hogy a hibaküszöb értéke nem lesz jelentősen kisebb az inkoherens esethez képest. (Sőt a gyémánt norma esetén még kicsit magasabb is.)

A csak koherens hibákhoz hasonlóan itt is érdemes a hibajavítás hatékonyságát összevetnünk az fizikai hibacsatornák Pauli-pödört verziójával, azaz a kiolvasási hibákat is figyelembe vevő inkoherens esettel.

Itt is ábrázoltam mindkét, R_d , R_i Pauli-pödrés arányt.



20. ábra. A Pauli-pödrés arányok, a gyémánt norma (20a) és a logikai hibaráta (20b) esetén a fizikai hiba ráták függvényében. Az ábrázolt pontokat $5.000 \times 100 \times d$ Monte Carlo kör alapján határoztam meg.

Az ábrákon jól látható, hogy kis fizikai hibaráta esetén a Pauli-pödrés arányok egyre magasabbak, tehát itt is igaz lesz a tökéletes kiolvasások mellett tett megállapítás, hogy hibaküszöb alatti fizikai hibaráták esetén a koherens hibák lényegesen veszélyesebbek, mint az inkohereus hibák.

6. Összefoglalás

A dolgozatomban röviden bemutatam az egyik legelterjedtebb kvantum hibajavító kód, a felületi kód, működését. A felületi kódban megjelenő lehetséges hibák közül a koherens hibákat (kis unitér transzformációk a qubiteken) és a kiolvasási hibákat részletesen tárgyaltam.

Ezután összefoglaltam, egy Bravyi és szerzőtársai által kifejlesztett módszert [5], amely a Fermionikus Lineáris Optika elméleti eszközeit használva, alkalmas a felületi kódban megjelenő koherens hibák hatékony szimulációjára.

Megmutattam, hogyan lehet ezt a módszert kiterjeszteni, úgy, hogy alkalmas legyen a koherens hibák mellett megjelenő kiolvasási hibák hatékony szimulálására is.

A szimulációs módszer ismertetése után bemutatam az eredményeimet, melyeket a felületi kódban megjelenő koherens és kiolvasási hibák numerikus vizsgálatával kaptam. Megmutattam, hogy a koherenciát a felületi kód hibajavító mechanizmusa hatékonyan "kimossa" a logikai hibacsatornából, a kiolvasási hibák jelenlétében hasonlóan, mint a tisztán koherens hibák esetén.

Meghatároztam a hibamodellre jellemző hibaküszöb értékét a *logikai hibaráta* és a *gyémánt norma* használatával, ezeket összevettem az inkohereus esettel. Eredményül azt kaptam, hogy a hibaküszöb értékek mindhárom esetben közel azonosak, így a koherens hibák, ebben az értelemben, nem veszélyesebbek, mint az inkohereus (Pauli-pödört) társaik.

Mindemellett szimulációim eredményei rámutattak arra is, hogy a koherens hibák a hibaküszöb alatti fizikai hibaráta esetén a kiolvasási hibák jelenlétében is sokkal veszélyesebbek, mint az inkohereus hibák.

A munkám során rengeteg újabb kérdés vetődött fel, amelyek további vizsgálódásra érdemesek. A felületi kód viselkedését érdemes lehet a hibamodell általánosításával is megvizsgálni, mikor a koherens hibákat és a kiolvasási hibákat két független paraméter írja le, ezen felül az ancilla qubiteken megjelenő koherens hibák vizsgálata is érdekes irány, azonban ez túlmutat a Fermionikus Lineáris Optika által nyújtott szimulációs lehetőségeken. A hibajavítás hatékonyságát jellemző mértékek karakterizálása is egy olyan probléma, amivel a szakirodalomban nem találkoztam és a koherens hibák esetén jól látható kérdéseket vet fel. Végül, de nem utolsó sorban a Fermionikus Lineáris Optika különböző problémákra történő optimalizálása is egy érdekes kihívásokat rejtő feladat, ilyen problémák lehetnek például a felületi kódon kívüli más kvantum hibajavító kódok, vagy összetettebb hibamodellek vizsgálata.

7. Köszönetnyilvánítás

A dolgozatom alapjául szolgáló kutatást az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatta a Kvantuminformatika Nemzeti Laboratórium keretében, valamint a Nemzeti Kiválósági Program keretében, a "Kvantumbitek előállítása, megosztása és kvantuminformációs hálózatok fejlesztése" című, 2017-1.2.1-NKP-2017-00001. számú projekt részeként, és a "Topologikusan védett állapotoktól a topologikus kvantumszámítógépig" című, FK 124723 számú pályázat részeként.

Szeretném megköszönni a QWorld szervezetnek, hogy részt vehettem QIntern 2021 nyári gyakornoki programjukban, ahol lehetőségem nyílt más diákokkal együtt dolgozni, és bemutatni az eredményeim egy részét.

Emellett szeretném megköszönni témavezetőmnek, Dr. Asbóth Jánosnak a rengeteg időt és energiát, amit nekem szentelt, a hasznos tanácsai, ötletei és kiváló témavezetése nélkül a dolgozat nem valósulhatott volna meg.

Hivatkozások

- [1] Flo simulation of the surface code - python script. https://github.com/martonaron88/Surface_code_FLO.git.
- [2] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.
- [3] Adriano Barenco, Charles H Bennett, Richard Cleve, David P DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical review A*, 52(5):3457, 1995.
- [4] S Bravyi and R König. Classical simulation of dissipative fermionic linear optics. *Quantum Information and Computation*, 12(11-12):1–19, 2012.
- [5] Sergey Bravyi, Matthias Englbrecht, Robert König, and Nolan Peard. Correcting coherent errors with surface codes. *npj Quantum Information*, 4(1):1–6, 2018.
- [6] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Physical Review A*, 71(2):022316, 2005.
- [7] Andrew S Darmawan and David Poulin. Tensor-network simulations of the surface code under realistic noise. *Physical review letters*, 119(4):040502, 2017.
- [8] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. Topological quantum memory. *Journal of Mathematical Physics*, 43(9):4452–4505, 2002.
- [9] Joseph Emerson, Marcus Silva, Osama Moussa, Colm Ryan, Martin Laforest, Jonathan Baugh, David G Cory, and Raymond Laflamme. Symmetrized characterization of noisy quantum processes. *Science*, 317(5846):1893–1896, 2007.
- [10] Austin G Fowler. Minimum weight perfect matching of fault-tolerant topological quantum error correction in average $o(1)$ parallel time. *arXiv preprint arXiv:1307.1740*, 2013.

- [11] Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):032324, 2012.
- [12] Daniel Greenbaum and Zachary Dutton. Modeling coherent errors in quantum error correction. *Quantum Science and Technology*, 3(1):015007, 2017.
- [13] Howard E Haber. Notes on antisymmetric matrices and the pfaffian. *matrix*, 1:3, 2015.
- [14] Shigeo Hakkaku, Kosuke Mitarai, and Keisuke Fujii. A sampling-based quasi-probability simulation for fault-tolerant quantum error correction on the surface codes under coherent noise. *arXiv preprint arXiv:2105.04478*, 2021.
- [15] O Higgott. Pymatching: A python package for decoding quantum codes with minimum-weight perfect matching. *arXiv preprint arXiv:2105.13082*, 2021.
- [16] Clare Horsman, Austin G Fowler, Simon Devitt, and Rodney Van Meter. Surface code quantum computing by lattice surgery. *New Journal of Physics*, 14(12):123011, 2012.
- [17] Joseph K Iverson and John Preskill. Coherence in logical quantum channels. *New Journal of Physics*, 22(7):073066, 2020.
- [18] A Yu Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, 2003.
- [19] Naomi H Nickerson and Benjamin J Brown. Analysing correlated noise on the surface code using adaptive decoding algorithms. *Quantum*, 3:131, 2019.
- [20] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.
- [21] Christopher A Pattison, Michael E Beverland, Marcus P da Silva, and Nicolas Delfosse. Improved quantum error correction using soft information. *arXiv preprint arXiv:2107.13589*, 2021.
- [22] KJ Satzinger, Y Liu, A Smith, C Knapp, M Newman, C Jones, Z Chen, C Quintana, X Mi, A Dunsworth, et al. Realizing topologically ordered states on a quantum processor. *arXiv preprint arXiv:2104.01180*, 2021.

- [23] Ashley M Stephens. Fault-tolerant thresholds for quantum error correction with the surface code. *Physical Review A*, 89(2):022321, 2014.
- [24] Barbara M Terhal and David P DiVincenzo. Classical simulation of noninteracting-fermion quantum circuits. *Physical Review A*, 65(3):032325, 2002.
- [25] Yu Tomita and Krysta M Svore. Low-distance surface codes under realistic quantum noise. *Physical Review A*, 90(6):062320, 2014.
- [26] F Venn and B Béri. Error-correction and noise-decoherence thresholds for coherent errors in planar-graph surface codes. *Physical Review Research*, 2(4):043412, 2020.
- [27] Chenyang Wang, Jim Harrington, and John Preskill. Confinement-higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory. *Annals of Physics*, 303(1):31–58, 2003.
- [28] Yulin Wu, Wan-Su Bao, Sirui Cao, Fusheng Chen, Ming-Cheng Chen, Xiawei Chen, Tung-Hsun Chung, Hui Deng, Yajie Du, Daojin Fan, et al. Strong quantum computational advantage using a superconducting quantum processor. *arXiv preprint arXiv:2106.14734*, 2021.

A. Függelék: A logikai forgatási szög meghatározása kiolvasási hibák jelenlétében

Induljunk ki a (44)-es egyenletből

$$|\phi^{\underline{s}, \underline{s}'}\rangle = \frac{1}{\sqrt{p(\underline{s})}} \hat{C}_{\underline{s}'} \hat{\Pi}_{s^d} \hat{E}_{coh} \hat{\Pi}_{s^{d-1}} \hat{E}_{coh} \dots \hat{\Pi}_{s^1} \hat{E}_{coh} |\psi_L^0\rangle. \quad (101)$$

Mindegyik hibajavítás után beszúrhatjuk az adott körben érzékelt tökéletes szindrómához tartozó korrekciós operátor négyzetét, hiszen ez úgyis 1-et ad eredményül. Ez annak felel meg, mintha minden mérési kör után kijavítanánk a hibát, de rögtön vissza is csinálnánk a javítást.

$$|\phi^{\underline{s}, \underline{s}'}\rangle = \frac{1}{\sqrt{p(\underline{s})}} \hat{C}_{\underline{s}'} \hat{C}_{s^d} \hat{C}_{s^d} \hat{\Pi}_{s^d} \hat{E}_{coh} \dots \hat{C}_{s^1} \hat{C}_{s^1} \hat{\Pi}_{s^1} \hat{E}_{coh} |\psi_L^0\rangle. \quad (102)$$

A j -edik körben megjelenő hiba, mérés és korrekció a következő alakra hozható

$$\frac{1}{\sqrt{p(s^j)}} \hat{C}_{s^j} \hat{\Pi}_{s^j} \hat{E}_{coh} \hat{C}_{s^{j-1}} = \frac{1}{\sqrt{p(s^j)}} \hat{C}_{s^j} \hat{\Pi}_{s^j} \hat{U}_j(s^j), \quad (103)$$

ahol U_j felfogható egy a j -edik körben megjelenő koherens hibaként, ahol a fizikai forgatási szögek nem azonosak az összes qubiten. A (103)-as egyenlet tovább írható, a (40)-es egyenlet felhasználásával, hiszen a (103)-as egyenletben szereplő operátor mindig egy logikai állapoton fog hatni.

$$\frac{1}{\sqrt{p(s^j)}} \hat{C}_{s^j} \hat{\Pi}_{s^j} \hat{U}_j = e^{i\theta_j(s^j, s^{j-1}) \hat{Z}^L}. \quad (104)$$

Fontos megjegyezni, hogy a j -edik mérési körhöz tartozó logikai forgatási szög a j -edik és a $j-1$ -edik körben mért szindrómától is függ U_j -n keresztül.

Bevezethetjük ezeknek a θ_j szögeknek az összegét θ^* -ot:

$$\theta^* = \sum_{j=1}^d \theta_j, \quad (105)$$

ennek a segítségével pedig már fel tudjuk írni a (45)-ös egyenletet:

$$|\phi^{\underline{s}, \underline{s}'}\rangle = \hat{C}_{\underline{s}'} \hat{C}_{s^d} e^{i\theta^* \hat{Z}^L} |\psi_L^0\rangle. \quad (106)$$

B. Függelék: A Fermionikus Lineáris Optika lépései

A FLO megengedett lépéseihöz bizonyos, a kovarianciamátrixot érintő, mátrixtranszformációk tartoznak. Ezeket a mátrixtranszformációkat végrehajtó algoritmusokat fogom ismertetni a függelék ezen részében.

A (67)-es egyenlet által definiált kovarianciamátrix létrehozásához nem írtam le külön egy algoritmus, itt egyszerűen végig kell menni az N darab (p,q) páron és beírni az $M_{pq} = 1, M_{qp} = -1$ értékeket.

A (70)-es egyenletből kiderül, hogy a kovarianciamátrix elemei közül csak azok változnak meg, ahol az $i\hat{c}_j\hat{c}_k$ nem cserél fel az $e^{\theta\hat{c}_p\hat{c}_q}$ operátorral. Ez akkor fordul elő, amikor $j = p$, vagy $j = q$, vagy $k = p$, vagy $k = q$. Ilyenkor a Majorana operátorokra a következő transzformációs szabályok írhatók fel:

$$\begin{aligned} e^{-\theta\hat{c}_p\hat{c}_q}\hat{c}_pe^{\theta\hat{c}_p\hat{c}_q} &= \cos(2\theta)\hat{c}_p + \sin(2\theta)\hat{c}_q \\ e^{-\theta\hat{c}_p\hat{c}_q}\hat{c}_qe^{\theta\hat{c}_p\hat{c}_q} &= \cos(2\theta)\hat{c}_q - \sin(2\theta)\hat{c}_p. \end{aligned} \quad (107)$$

Ezen szabályok alapján a kovarianciamátrixot transzformáló algoritmus a következő lesz:

1. Algorithm Rotation (M, θ, p, q)

```

M'_{p,:} ← M_{p,:} cos(2θ) − M_{q,:} sin(2θ)
M'_{q,:} ← M_{q,:} cos(2θ) + M_{p,:} sin(2θ)
M'_{:,p} ← M_{:,p} cos(2θ) − M_{:,q} sin(2θ)
M'_{:,q} ← M_{:,q} cos(2θ) + M_{:,p} sin(2θ)
M'_{p,p} ← 0
M'_{q,q} ← 0
M'_{p,q} ← M_{p,q} cos²(2θ) − M_{q,p} sin²(2θ)
M'_{q,p} ← −M_{p,q} cos²(2θ) + M_{q,p} sin²(2θ)
return M'

```

A (72)-es egyenlet kifejtésénél fel kell használnunk a 4 hosszúságú Majorana láncokra vonatkozó összefüggést (65). Az átalakítások után könnyen beláthatjuk, hogy a következő algoritmus fogja megvalósítani a megfelelő mátrixtranszformációt.

2. Algorithm Measurement (M,p,q)

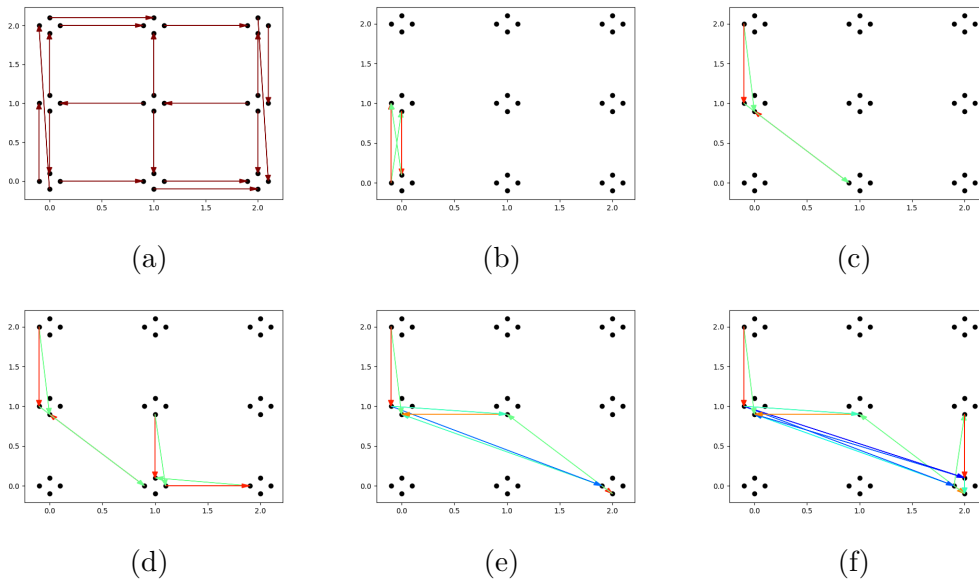
$p \leftarrow (1/2)(1 + M_{p,q})$ \triangleright +1 mérési eredmény valószínűsége.
if $p \neq 0$ **then**
 $\mathbf{K} \leftarrow p$ -th row of matrix M
 $\mathbf{L} \leftarrow q$ -th row of matrix M
 $M' \leftarrow M + (1/2p)(\mathbf{L}\mathbf{K}^T - \mathbf{K}\mathbf{L}^T)$
 $M'_{p,:} \leftarrow 0$
 $M'_{q,:} \leftarrow 0$
 $M'_{:,p} \leftarrow 0$
 $M'_{:,q} \leftarrow 0$
 $M'_{p,q} \leftarrow 1$
 $M'_{q,p} \leftarrow -1$
 return (M',p)
end if

C. Függelék: A Felületi kód szimulálási idejének lerövidítése

A felületi kód FLO szimulációját jelentősen lerövidíti, ha nem a teljes kovarianciamátrixot transzformáljuk az egyes lépésekben, csak annak egy nem triviális részmatrixát. Ezt akkor tehetjük meg, ha a (92)-es egyenletben definiált transzformációkat egyesével qubitról-qubitra hajtjuk végre, hiszen az egyes Majorana párok megmérése "kicsatolja" azokat a kovarianciamátrixból.

A felületi kód kovarianciamátrixát ábrázolhatjuk egy gráfként is, ahol a felületi kódot alkotó Majorana operátorok a csúcsok és az őket összekötő élek -1 és 1 között súlyozva vannak. (A p,q csúcsok közötti él felel meg az $M_{p,q}$ elemnek a kovarianciamátrixban.) A felületi kód inicializált $|\phi_{link,+}^L\rangle$ kiinduló állapota esetén minden Majorana fermion pontosan egy másik Majorana fermionnal van összekötve (1-es súlyozással). A szimuláció során a forgatások új éleket hoznak be a gráfba, viszont a mérések bizonyos éleket "kiiktatnak" a gráfból, ez adja meg a lehetőséget arra, hogy a lehetséges $4n(4n - 1)/2$ él helyett csak sokkal kevesebb éllel számoljunk a szimuláció során.

A szimuláció során a teljes kovarianciamátrix helyett elegendő 3 másik mennyiséget figyelemmel kísérnünk. A kovarianciamátrix triviális éleit, ezek azok az élek, amelyek értéke 1 (vagy -1). Azoknak a Majorana fermionoknak a sorszámát, amelyeket nem triviális él köt össze, ezek az aktív Majoranák. Végül pedig a kovarianciamátrixnak az aktív Majoranákat tartalmazó részmatrixát.



21. ábra. A 3-as kódtávolságú kód kovarianciamátrix gráfjai. A (21a) ábrán a kiinduló állapot triviális élei láthatóak, a későbbi ábrákon a triviális élek már nem láthatóak. A (21b) ábrán az első forgatás utáni nem triviális élek látszanak, (a színek az élek súlyát reprezentálják, de ezek konkrét értéke nem igazán fontos). A (21c) ábrán már az első \hat{X} és $\hat{S}\hat{X}$ mérések utáni nem triviális élek látszanak, jól látható, hogy a mérések új nem triviális éleket is behoznak, azonban az első qubitet "kicsatolják" a rendszerből, triviális élek sem csatlakoznak ezekhez a Majoranákhoz. A további ábrákon a további forgatások és mérések hatása látható.