

UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA



*Corso di Laurea Triennale in
Ingegneria Elettronica*

*Codici per la correzione d'errore nelle comunicazioni
quantistiche*

Error correction codes in quantum communications

Relatore:

PROF. FRANCO CHIARALUCE

Correlatore:

DOTT. MASSIMO BATTAGLIONI

Laureando:

ALESSIO BALDELLI

ANNO ACCADEMICO 2021-2022

*"If computers that you build are quantum,
then spies everywhere will all want 'em.
Our codes will all fail,
and they'll read our e-mail,
till we get crypto that's quantum, and daunt 'em."
Jennifer and Peter Shor*

Sommario

I codici per la correzione degli errori quantistici (Quantum Error Correction Codes - QECCs) possono essere costruiti dal noto paradigma di codifica classico sfruttando l'isomorfismo intrinseco tra il dominio classico e quello quantistico, superando anche le sfide imposte dalle stringenti leggi della fisica quantistica. Partendo da tali presupposti, questo lavoro di tesi fornisce approfondimenti sulla dualità tra la teoria classica e quantistica della codifica, cercando quindi di colmare il divario tra loro. In particolare, si esaminerà la ricca storia dei codici classici e di quelli quantistici. Si fornisce, pertanto, una guida semplice e dettagliata per la costruzione dei QECCs basati sui cosiddetti stabilizzatori, a partire da codici classici binari e quaternari arbitrari, come esemplificato dai codici Calderbank-Shor-Steane (CSS) dual-containing e non-dual-containing, dai codici non CSS e dai codici entanglement-assisted (EA). Infine, si applicano queste considerazioni a due popolari famiglie di codici classici, vale a dire i codici Bose-Chaudhuri-Hocquenghem (BCH) e i codici convoluzionali, fornendo degli esempi di progettazione dettagliati sia per le loro versioni classiche che per quelle quantistiche.

Indice

1	Rudimenti teorici fondamentali di meccanica quantistica	7
1.1	Superposition ed Entanglement	7
1.1.1	<i>Superposition</i>	7
1.1.2	<i>Entanglement</i>	11
1.1.3	<i>Considerazioni finali</i>	13
1.2	Implicazioni della meccanica quantistica sulla teoria dell'informazione e nelle comunicazioni	14
1.3	Decoerenza quantistica	17
1.3.1	<i>Canale di smorzamento dell'ampiezza</i>	17
1.3.2	<i>Canale di smorzamento di fase</i>	24
1.3.3	<i>Canale di Pauli</i>	26
2	Panoramica storica dei codici di correzione degli errori classici e quantistici	33
2.1	Teoria della codifica classica	33
2.1.1	<i>Obiettivi di progetto</i>	33
2.1.2	<i>Codici per la correzione degli errori</i>	39
2.2	Teoria della codifica quantistica	47
2.2.1	<i>Obiettivi di progetto</i>	47
2.2.2	<i>Codici per la correzione degli errori</i>	51
3	Transizione dal dominio classico al dominio quantistico	60
3.1	Premessa	60
3.2	Teorema di non clonazione	61
3.3	Operazione di misura del qubit	64
3.4	Natura degli errori quantistici	67
4	Formalismo Stabilizzatore	74
4.1	Progettazione di codici tramite il formalismo stabilizzatore	74
4.2	Classificazione dei modelli di errore	81
5	Isomorfismo dal dominio quantistico a quello classico	84
5.1	Isomorfismo dal dominio di Pauli al dominio binario	84
5.2	Isomorfismo dal dominio di Pauli al dominio quaternario	89
5.3	Conclusioni	93

6	Tassonomia dei codici stabilizzatori	95
6.1	Codici Calderbank-Shor-Steane	95
6.1.1	<i>Esempio Operativo</i>	98
6.2	Codici Non CSS	100
6.2.1	<i>Esempio Operativo</i>	100
6.3	Codici Entanglement-Assisted	101
6.3.1	<i>Esempio operativo</i>	103
7	Esempi di progetto	106
7.1	Codici Bose-Chaudhuri-Hocquenghem	106
7.1.1	<i>Codici classici Bose-Chaudhuri-Hocquenghem</i>	106
7.1.2	<i>Esempio Operativo: codice classico BCH</i>	109
7.1.3	<i>Codici quantistici Bose-Chaudhuri-Hocquenghem</i>	112
7.1.4	<i>Esempio Operativo: codice QBCH</i>	112
7.2	Codici Convolutionali	116
7.2.1	<i>Codici classici Convolutionali</i>	116
7.2.2	<i>Esempio Operativo: codice classico Convolutionale</i>	117
7.2.3	<i>Codici quantistici Convolutionali</i>	118
7.2.4	<i>Esempio Operativo: codice quantistico Convolutionale</i>	121
	Bibliografia	127

Ringraziamenti

Ringrazio il Professore Franco Chiaraluce e il Dottor Massimo Battaglioni per avermi concesso l'opportunità di svolgere il tirocinio e la tesi in questo ambito così affascinante.

Ringrazio la mia famiglia per essermi stata sempre accanto, anche nei momenti di difficoltà e di stress; vi voglio bene.

Ringrazio i miei cari amici di sempre per avermi supportato in questo percorso e i miei fedeli colleghi di corso Giovanni e Federico per avermi aiutato durante questo lungo viaggio.

Ringrazio Camilla per avermi dato tanta forza per superare gli ostacoli di questi anni.

Ringrazio la mia passione per non avermi fatto mollare.

Capitolo 1

Rudimenti teorici fondamentali di meccanica quantistica

In questa prima sezione vengono discusse ed approfondite le conoscenze teoriche di base di meccanica quantistica, indispensabili per la comprensione dei fenomeni di comunicazione quantistica.

1.1 Superposition ed Entanglement

Per la teoria dell'informazione classica un generico bit può assumere il valore 0 (valore logico basso) o 1 (valore logico alto). Invece, un qubit (quantum bit o bit quantistico) è caratterizzato dalla sovrapposizione (superposition) tra un possibile stato $|0\rangle$ ed un possibile stato $|1\rangle$, fintanto che non viene misurato o "osservato". In particolare, con la seguente notazione di Dirac $|\cdot\rangle$ (chiamata anche Ket) [46], si indica uno stato quantistico che caratterizza il generico qubit, nell'istante di osservazione.

1.1.1 *Superposition*

Il fenomeno della superposition, conosciuto anche con il nome di principio di sovrapposizione, è il primo postulato della meccanica quantistica. Esso afferma che due o più stati quantistici possono essere sommati, cioè sovrapposti, dando come risultato un altro stato quantistico parimenti valido. Dunque ogni stato quantistico può essere rappresentato come somma di due o più altri stati distinti. Tornando al principale oggetto della presente trattazione, si osserva pertanto che un qubit può essere, allo stesso tempo, sia nello stato $|1\rangle$ che nello stato $|0\rangle$ (superposition). Pertanto, lo stato sovrapposto del qubit si descrive come:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle. \quad (1.1)$$

In particolare si osserva che α e β sono due coefficienti complessi, che possono assumere qualsiasi valore tale che valga la seguente relazione, che ne lega i moduli: $|\alpha|^2 + |\beta|^2 = 1$. Il loro significato è quello di determinare la probabilità con cui il qubit può collassare in uno dei due stati. In particolare

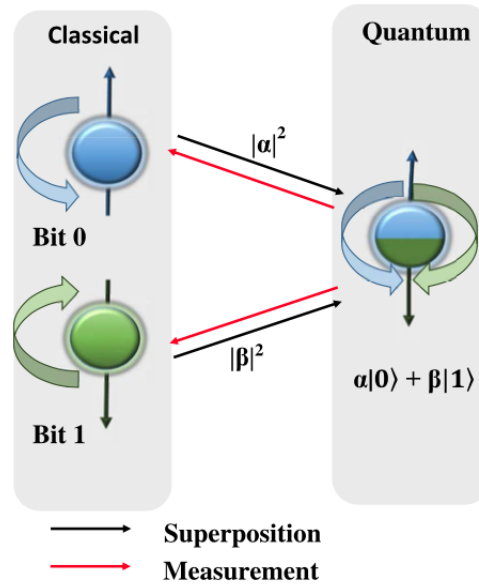


Figura 1.1: Realizzazione di un bit classico e di un qubit usando la rappresentazione mediante lo spin di un elettrone [12].

si ha che, $|0\rangle$ è caratterizzato da una probabilità pari a $|\alpha|^2$, mentre $|1\rangle$ da $|\beta|^2$. Si sottolinea ancora una volta che, quando si misura un qubit, esso dà come risultato della misurazione sempre e solo $|0\rangle$ o $|1\rangle$, così come accade per il bit classico.

Ad esempio, un qubit può essere nello stato

$$|\psi\rangle = \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle, \quad (1.2)$$

che, quando viene misurato, può dare sia il risultato $|0\rangle$ che il risultato $|1\rangle$ con una probabilità pari al cinquanta per cento, essendo infatti che $|\alpha|^2 = |\beta|^2 = |1/\sqrt{2}|^2 = 0.5$. Si farà riferimento a questo esempio considerato notevole, poiché i due stati sono equiprobabili, anche nel seguito della trattazione [116].

Supponendo di rappresentare un generico bit classico utilizzando un elettrone, si associa il verso dello spin (dell'elettrone) ad uno dei suoi due stati, secondo il seguente mapping:

$$\begin{cases} 0 \rightarrow |\uparrow\rangle \text{ (spin-up)} \\ 1 \rightarrow |\downarrow\rangle \text{ (spin-down)}. \end{cases}$$

Naturalmente, sarebbe stato valido anche il viceversa (vedere Fig. 1.1).

Pertanto, per quanto sopra, per gli stati del qubit, vale:

$$\begin{cases} |0\rangle \rightarrow |\uparrow\rangle \text{ (spin-up)} \\ |1\rangle \rightarrow |\downarrow\rangle \text{ (spin-down)}. \end{cases}$$

Dunque un qubit esiste come superposition dei due stati, ma collassa ad un singolo valore quando viene misurato.

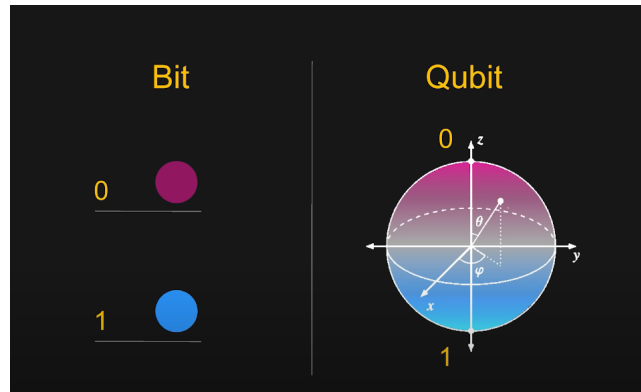


Figura 1.2: Confronto tra i due possibili stati di un bit tradizionale e rappresentazione con la sfera di Bloch dell'informazione contenuta in un qubit [106].

Appare evidente da quest'ultima interpretazione che le quantità α e β descrivono, nell'esempio considerato, l'orientazione dello spin nello spazio tridimensionale. Dalla Fig. 1.1 risulta che l'inclinazione dell'asse di spin, dovuta al valore di α e β , ha significato solo per un qubit e non per un bit: è una possibile rappresentazione grafica della superposition.

In alternativa, lo stato di superposition quantistico può essere facilmente spiegato rappresentando il generico qubit come la posizione di un punto su una sfera, come appare in Fig. 1.2.

Approfondimento: Sfera di Bloch La sfera rappresentata in Fig. 1.2 è conosciuta in meccanica quantistica come "Sfera di Bloch", in onore del fisico tedesco Felix Bloch. È una rappresentazione degli stati "puri" di un sistema quantistico a 1 qubit. La sfera di Bloch è geometricamente una sfera di raggio unitario i cui punti sulla superficie sono in corrispondenza biunivoca con gli stati "puri" del qubit. Il generico stato sovrapposto $|\psi\rangle$ del qubit viene rappresentato nella sfera di Bloch come segue:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{j\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle, \quad (1.3)$$

dove $0 \leq \theta \leq \pi$ e $0 \leq \phi < 2\pi$. I parametri ϕ e θ identificano univocamente un punto di coordinate (x, y, z) sulla sfera unitaria nello spazio euclideo \mathbb{R}^3 tramite le seguenti espressioni:

$$\begin{cases} x = \sin(\theta) \cos(\phi) \\ y = \sin(\theta) \sin(\phi) \\ z = \cos(\theta) \end{cases} .$$

Per ulteriori approfondimenti, si consultino [116], [2].

Infatti, un qubit riesce a contenere una quantità d'informazione tale da

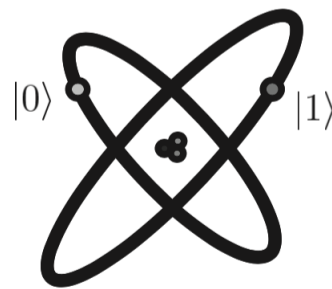


Figura 1.3: Qubit rappresentato da un atomo con due livelli elettronici [116].

essere rappresentata come un punto su una sfera. Banalmente, per comprendere a fondo la grande differenza in termini di capacità di informazione tra un generico qubit ed un bit classico, ci si può servire di questo esempio. Si immagini di dover dichiarare la propria posizione sul pianeta Terra avendo a disposizione soltanto un bit (*caso 1*) oppure un qubit (*caso 2*).

Caso 1. Ad esempio, un'informazione che si può fornire è se ci si trovi sull'emisfero boreale (nord) o su quello australe (sud), in base allo stato assunto dal bit, previo opportuno mapping, dato che il generico bit può assumere solo due stati differenti. Se ne riporta uno a titolo di esempio:

$$\begin{cases} 0 \rightarrow \text{emisfero nord} \\ 1 \rightarrow \text{emisfero sud} \end{cases}$$

naturalmente, sarebbe stato valido anche il mapping opposto.

Caso 2. Viceversa questa volta, supponendo di modellare il pianeta Terra come una sfera, si è in grado di indicare con precisione assoluta la propria posizione sul globo (un punto), in ragione delle coordinate sferiche che caratterizzano la rappresentazione del qubit. Quindi si osserva che i possibili stati che potenzialmente può assumere un qubit sono infinitamente più numerosi rispetto a quelli del bit tradizionale.

Concretamente, i ricercatori hanno sviluppato più modalità per ottenere sperimentalmente un qubit; per esempio attraverso le due differenti polarizzazioni di un fotone, o servendosi dell'allineamento di uno spin nucleare in un campo magnetico uniforme oppure tramite i due stati di un elettrone nell'orbita di un singolo atomo, come mostrato in Fig. 1.3 [116]. Nel modello dell'atomo, infatti, l'elettrone può esistere sia nello stato detto di terra (ground state), chiamato anche stato fondamentale, sia in quello chiamato eccitato (excited state), che corrispondono agli stati $|0\rangle$ e $|1\rangle$, rispettivamente, del generico qubit. Naturalmente, di qui in avanti, si presta molta attenzione al significato di queste interpretazioni del qubit, dato che è collegato al fenomeno della superposition e alla natura intrinsecamente probabilistica dei sistemi quantistici, piuttosto che a questioni prettamente "fisiche".

Dunque, terminata questa doverosa premessa, si può affermare che un qubit è interpretabile come un vettore bi-dimensionale a coefficienti complessi che

appartiene allo spazio vettoriale di Hilbert \mathcal{H}_2 , dove la dimensione è data, appunto, dai due possibili stati. Pertanto un possibile sistema composto da N qubit è rappresentato da un vettore 2^N -dimensionale, descritto come segue:

$$\alpha_0 |00\dots 0\rangle + \alpha_1 |00\dots 1\rangle + \dots + \alpha_{2^N-1} |11\dots 1\rangle, \quad (1.4)$$

dove $\alpha_i \in \mathbb{C}$ e $\sum_{i=0}^{2^N-1} |\alpha_i|^2 = 1$.

Appare ora chiaro che un sistema di N qubit è caratterizzato dalla superposition contemporanea di tutti i 2^N possibili valori, che conferisce, a tali sistemi quantistici, la proprietà intrinseca conosciuta come *parallelismo* [12], [116].

1.1.2 Entanglement

L'entanglement, o correlazione quantistica, che letteralmente significa "groviglio", "intreccio", è un fenomeno quantistico che presuppone un'azione immediata a distanza, che coinvolge almeno due entità. Per esempio si immagini di prendere in considerazione un evento nello spazio che viene influenzato da un altro evento nello spazio, arbitrariamente distante, in un tempo infinitesimo. A tal proposito è celebre, per descrivere questo fenomeno, la seguente definizione del 1930 di Albert Einstein: "spooky action at distance" che letteralmente significa "azione spettrale (nel senso di spaventosa) a distanza" [49]. Il motivo di questa definizione, apparentemente poco razionale, è spiegato nel seguito.

Secondo un importante postulato della meccanica quantistica [116]; *lo spazio di stato di un sistema fisico composto è dato dal prodotto tensoriale degli spazi di stato dei sistemi fisici che lo compongono*. In particolare, ai fini della trattazione, si considera come sistema fisico composto un sistema di N qubit, dove ogni qubit è uno dei sistemi fisici componenti, mentre, per spazio di stato, si intende il generico stato $|\psi\rangle$. Allora lo stato congiunto del macrosistema, composto da N qubit, risulterebbe:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_N\rangle, \quad (1.5)$$

dove $|\psi_i\rangle$ indica lo stato del generico i -esimo qubit e \otimes denota l'operatore di prodotto tensoriale.

Invece, nonostante quanto affermato dal postulato, che in generale è sempre valido in meccanica quantistica, tale formulazione non è assolutamente verificata per la definizione dello stato di un sistema composto da N qubit, nel caso in cui essi siano entangled tra loro [116]. Pertanto, di qui in avanti, si considerano dei particolari sistemi quantistici, cioè quelli composti da N qubit "aggrovigliati" (entangled) tra loro, per i quali quindi lo stato non può essere espresso come prodotto tensoriale dei singoli qubit.

Per comprendere meglio questo sorprendente fenomeno, si ipotizzi di considerare un sistema composto da soli 2 qubit. Richiamando quanto detto precedentemente sullo spin, si nota che in principio i bit quantistici in questione sono caratterizzati da un aggrovigliamento di possibilità di spin

(entanglement, appunto) e inoltre sono anche entangled tra loro. Quindi lo stato $|\psi\rangle$ del sistema, risulta:

$$|\psi\rangle = \alpha |00\rangle + \beta |11\rangle, \quad (1.6)$$

considerando naturalmente $\alpha, \beta \neq 0$.

Come motivato sopra, non è possibile pertanto scrivere il generico stato del sistema come prodotto tensoriale, poiché i due qubit sono entangled, infatti si ha:

$$\begin{aligned} \alpha |00\rangle + \beta |11\rangle &\neq (a_1 |0\rangle + b_1 |1\rangle) \otimes (a_2 |0\rangle + b_2 |1\rangle) \\ &= a_1 a_2 |00\rangle + a_1 b_2 |01\rangle + b_1 a_2 |10\rangle + b_1 b_2 |11\rangle, \end{aligned} \quad (1.7)$$

dove a_1 e b_1 sono i coefficienti complessi del primo qubit, mentre a_2 e b_2 quelli del secondo.

Particolarizzando poi la (1.6) con le stesse scelte nell'eq.(1.2), si ottiene il seguente stato:

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \quad (1.8)$$

dunque da questo esempio si può osservare che non ci sono singoli stati $|A\rangle$ e $|B\rangle$ dei qubit tali che $|\psi\rangle = |A\rangle |B\rangle$ [116].

Approfondimento: Stati di Bell Lo stato quantistico a due qubit riportato nell'eq.(1.8) è molto importante ed è conosciuto come Stato di Bell o Coppia EPR, in onore di Einstein, Podolsky e Rosen che ne studiarono le sorprendenti caratteristiche. È uno stato quantistico di due qubit che rappresenta l'esempio più semplice di entanglement quantistico. Nell'ambito del quantum computing, esso svolge un ruolo chiave per il teletrasporto quantistico (quantum teleportation) e per la codifica superdensa (superdense coding) [116].

Come è intuibile, lo stato di Bell ha la proprietà che, misurando il primo qubit, si possono ottenere due diversi risultati equiprobabili. Se la misurazione del primo qubit dà esito $|0\rangle$, con probabilità $1/2$, si trova lo stato, di post-misurazione, $|\phi'\rangle = |00\rangle$, mentre se l'esito fosse $|1\rangle$, anch'esso con probabilità $1/2$, si avrebbe lo stato $|\phi'\rangle = |11\rangle$, dopo la misurazione. Di conseguenza, una misurazione del secondo qubit dà sempre lo stesso risultato della misurazione del primo qubit. Cioè, i risultati della misurazione sono massimamente correlati; il fenomeno dell'entanglement quantistico è al suo culmine. Inoltre, si è scoperto che, nonostante si applichino prima alcune operazioni ai due qubit, se si eseguono nuove misurazioni sullo stato di Bell, esistono ancora interessanti correlazioni tra il risultato di una misurazione sul primo e sul secondo qubit [116].

In particolare, in base a quanto detto, esistono quattro diversi tipi di stati di Bell, cioè:

$$\begin{aligned} |\phi^+\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}}, & |\psi^+\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}}; \\ |\phi^-\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}}, & |\psi^-\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}}, \end{aligned} \quad (1.9)$$

dove la notazione $|\phi^\pm\rangle$ e $|\psi^\pm\rangle$ è utilizzata ad hoc proprio per indicare una di queste particolari configurazioni [116]. In particolare, il segno $-$ sta ad indicare un cambiamento di fase nello stato del qubit (o dei qubit, come in questo caso), cioè un'inversione del verso dello spin, in seguito verranno forniti ulteriori chiarimenti.

Effettivamente, a valle di quanto detto sopra, si può osservare che esiste una particolare correlazione che lega i due qubit (a causa dell'entanglement). Infatti dopo che uno dei due bit quantistici viene misurato e quindi ne viene stimato lo spin, cioè lo stato, si osserva che lo spin dell'altro, e quindi lo stato, è identico rispetto al primo, se e solo se le due particelle sono geograficamente separate. Questa caratteristica, in un ipotetico sistema composto da due elementi, permette di conoscere automaticamente l'entità dello stato di un qubit, dopo aver osservato lo stato dell'altro, se i due qubit sono entangled tra loro. Esplicitamente, se il primo qubit dell'eq.(1.6) collassa allo stato $|0\rangle$ dopo la misura, il che può accadere con probabilità $|\alpha|^2$, dunque il secondo qubit ha sicuramente stato $|0\rangle$. Allo stesso modo, se il primo qubit collassa allo stato $|1\rangle$, il che può verificarsi con una probabilità $|\beta|^2$, allora il secondo qubit è anch'esso caratterizzato dallo stato $|1\rangle$. È questo il senso profondo di correlazione quantistica, cioè il legame intrinseco che esiste tra i vari qubit che compongono il sistema.

Pertanto si dice che uno stato di un sistema composto avente questa proprietà (cioè che non può essere scritto come prodotto tensoriale di stati dei suoi sistemi componenti) è uno stato entangled. Per ragioni che sono ancora oggetto di discussione tra i ricercatori, gli stati entangled svolgono un ruolo cruciale nella computazione e nella teoria dell'informazione quantistica [116].

1.1.3 Considerazioni finali

Il fenomeno della superposition e dell'entanglement non hanno corrispondenti nel dominio classico, ma attraverso il loro studio è possibile sviluppare una nuova gamma di potenti paradigmi di calcolo e comunicazione sicura. Pertanto, a partire dagli anni 90' dello scorso secolo, sono stati formulati algoritmi che hanno il potenziale per risolvere problemi spesso considerati intrattabili, con una complessità computazionale sostanzialmente ridotta, come esemplificato dall'algoritmo di fattorizzazione di Shor [144] e dall'algoritmo di ricerca di Grover [70].

Questa straordinaria potenza di calcolo è dovuta al parallelismo quantistico, proprietà derivante dal fenomeno della superposition.

A questo punto è interessante, ai fini della trattazione, proporre un parallelismo tra una memoria tradizionale dotata di N bit e la sua controparte quantistica (con N qubit). Come è ben noto dalla teoria dell'informatica tradizionale, un generico registro a scorrimento ad N bit può memorizzare, in ogni cella di memoria, solo uno dei possibili 2^N valori (o stati). Mentre, a causa della superposition, un registro quantistico ad N qubit è in grado di memorizzare tutti i possibili 2^N valori (o stati) contemporaneamente. In

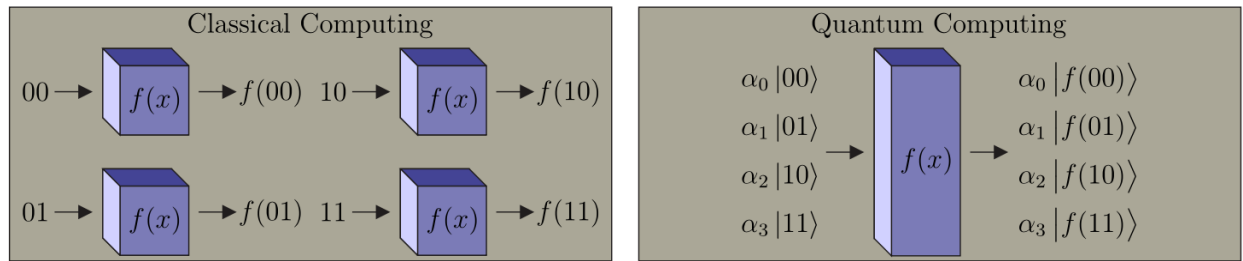


Figura 1.4: Confronto tra computazione classica e quantistica [12].

questo modo viene realizzato un modello di calcolo parallelo quantistico, che naturalmente risulta più prestazionale, a parità di bit/qubit, rispetto a quello realizzato con tecnologia tradizionale, equivalente alla presenza di più processori (o core) che agiscono simultaneamente.

Anche in questo ambito, si particolarizza il confronto di cui sopra analizzando prima un processo di calcolo classico a 2 bit e poi uno quantistico, con lo stesso numero di qubit.

Data una generica funzione binaria $f(x)$ tale che $f(x) : \{0, 1\} \rightarrow \{0, 1\}$, nel caso di computazione classica si calcola in serie $f(x)$ per tutte le possibili coppie $x \in \{00, 01, 10, 11\}$; quindi sono richieste quattro valutazioni (vedi Fig. 1.4). Mentre, in un sistema quantistico, si elaborano simultaneamente tutti i possibili valori di x , poiché il registro quantistico di 2 qubit è caratterizzato dalla superposition di tutti e quattro gli stati, vale a dire: $|\psi\rangle = \alpha_0 |00\rangle + \alpha_1 |01\rangle + \alpha_2 |10\rangle + \alpha_3 |11\rangle$. Dunque è sufficiente un'unica valutazione. Lo stato risultante, in termini della funzione $f(x)$, è dunque anch'esso dato dalla superposition di tutte e quattro le possibilità:

$$|\psi\rangle = \alpha_0 |f(00)\rangle + \alpha_1 |f(01)\rangle + \alpha_2 |f(10)\rangle + \alpha_3 |f(11)\rangle. \quad (1.10)$$

Come si può facilmente dedurre da quanto scritto sopra, peraltro, non è possibile leggere tutti e quattro i valori di $f(x)$, dal momento che il registro collasserà in uno dei quattro valori al momento della misurazione.

1.2 Implicazioni della meccanica quantistica sulla teoria dell'informazione e nelle comunicazioni

Si prevede che l'enorme capacità di elaborazione degli algoritmi di calcolo quantistico possa minacciare l'integrità della crittografia a chiave pubblica classica, che si basa sulla complessità computazionale delle funzioni matematiche sottostanti. In particolare, un importante schema di crittografia asimmetrica moderna, cioè l'RSA (dal nome dei suoi inventori, Rivest, Shamir e Adleman) si basa proprio sull'incapacità dei calcolatori di realizzare la scomposizione in fattori primi di numeri molto grandi in tempi utili; con i valori attualmente utilizzati, infatti, anche i più potenti super-computer

impiegherebbero anni [136]. Al contrario, i computer quantistici, muniti anche solo di un numero relativamente limitato di qubit, riescono a realizzare la fattorizzazione in tempi significativamente brevi. Pertanto, mentre la crittografia classica rischia di essere decrittata dall'informatica quantistica, le comunicazioni quantistiche supportano la diffusione sicura dei dati, dal momento che qualsiasi misura o osservazione mediante intercettazione perturba lo stato di superposition quantistico [116], [81]. Generalmente, nella comunicazione quantistica, i riceventi sono in grado di capire se un messaggio è stato decifrato oppure no perché le particelle quantistiche (qubit) che compongono il messaggio non possono essere osservate senza alterare le informazioni che contengono: i singoli fotoni una volta intercettati perdono infatti il loro contenuto informativo, per le leggi della meccanica quantistica. Se i fotoni che portano il messaggio arrivano nello stesso stato in cui sono stati inviati, significa che nessuno ha provato a intercettarlo. Viceversa se arrivano in uno stato diverso, vuol dire che le informazioni sono state captate da qualcuno o qualcosa e il messaggio non è più sicuro. A questo punto, generalmente, si procede con la ritrasmissione, ripetendo la procedura finché non si è certi di aver evitato un attacco [42].

Alcune delle principali applicazioni delle comunicazioni quantistiche sicure sono:

- le tecniche di distribuzione di chiavi quantistiche (QKD, Quantum Key Distribution) [171], [20];
- la comunicazione quantistica sicura e diretta (QSDC, Quantum Secure Direct Communication) [19], [29] e [168];
- la localizzazione quantistica incondizionata [168], spendibile per esempio per la futura automobile quantistica senza conducente (Quantum Car [102]) e per la geo-crittografia quantistica [103]. Tale verifica della posizione quantistica è già determinante per le transazioni bancarie sicure, così come per gli orologi quantistici ultra-precisi, utili per la sincronizzazione globale.
- Quantum Internet (Qinternet, vedi Fig. 1.5) [43].

Quest'ultima applicazione merita sicuramente una menzione speciale. In generale, le reti quantistiche, alla stregua di quelle classiche, permettono la trasmissione di informazioni quantistiche tra processori fisicamente separati. Esplicitamente, la Qinternet è concepita come una rete globale di sistemi quantistici eterogenei, che possono essere interconnessi, attraverso canali quantistici, realizzando sistemi più grandi. Per esempio, utilizzeranno la Qinternet i grandi Quantum computer [82], [110] e il sistema quantum key distribution (QKD), che è sicuro ed a lungo raggio. La QKD ha il compito di distribuire le chiavi di crittografia attraverso canali di comunicazione quantistica sia su fibra ottica terrestre che su collegamenti laser spaziali. Pertanto, si può osservare che una tecnologia come la rete di backhaul (o rete di ritorno), già molto usata nelle applicazioni telecomunicazionistiche tradizionali, essendo caratterizzata da una combinazione di canali wireless nello spazio libero e fibre ottiche, risulta particolarmente adatta per la Qinternet in virtù del parallelismo quantistico [43].

Andando più specificamente ad indagare il funzionamento di questo sistema, risulta utile capire le differenze, e quindi i possibili vantaggi, che esso

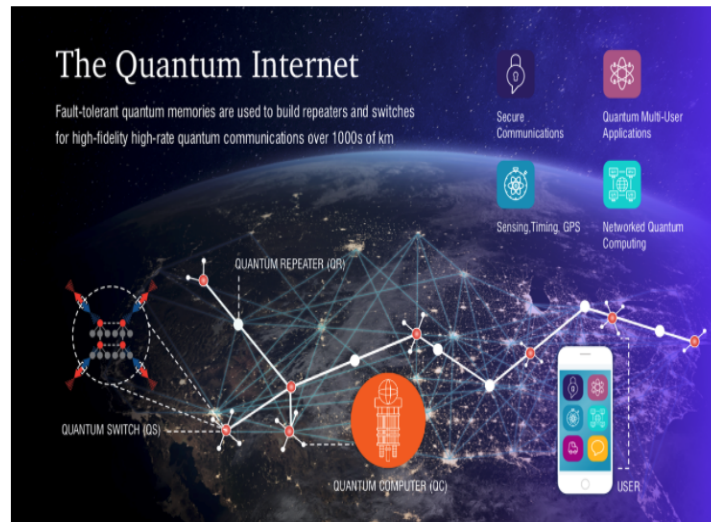


Figura 1.5: Rappresentazione della Qinternet e di alcune sue possibili applicazioni [151].

presenta rispetto all'internet tradizionale. Si osserva, per quanto detto precedentemente, che uno stato quantistico di N qubit richiede solo N usi del canale quantistico per trasmettere l'informazione completa, mentre sarebbero necessari 2^N usi del canale se si utilizzasse la trasmissione classica. Allo stesso modo, se ci sono k nodi quantistici, ognuno composto da N qubit, entangled tra loro, allora la loro capacità complessiva sarà quella di un sistema con kN qubit, dotato di uno spazio di stato 2^{kN} -dimensionale.

Al contrario, se i k nodi, di N qubit ciascuno, sono connessi classicamente, avranno uno spazio di stato effettivo di dimensione $k2^N$. Quindi, la connettività quantistica garantisce uno spazio di stato esponenzialmente più grande rispetto alla connettività classica.

Sfortunatamente, i canali quantistici, così come i sistemi quantistici della Fig. 1.5, presentano delle limitazioni, il che è un grosso ostacolo alla realizzazione pratica della Qinternet globale. Più specificamente, i qubit di un generico nodo, possono presentare alterazioni immesse sia dal canale di comunicazione che dai processi quantistici stessi [112]. Dunque, per quanto riguarda il canale di comunicazione quantistico, si ricorda che quest'ultimo introduce una certa attenuazione deleteria, misurata in dB/km, che limita significativamente il transmission rate attendibile, o equivalentemente, limita notevolmente la banda [158]. Le alterazioni dovute ai processi di calcolo quantistico, invece, sono causate dalle imperfezioni dell'hardware, come, per esempio, le non idealità dei quantum gates (porte quantistiche).

I sistemi di comunicazione basati su tecnologia quantistica supportano la trasmissione di informazioni sia classiche che quantistiche. Quando l'informazione da trasmettere è classica, si possono utilizzare tecniche tradizionali di correzione degli errori per contrastare l'impatto delle alterazioni quantistiche [13], [14]. In particolare, all'inizio, le informazioni classiche vengono codificate utilizzando un codice di correzione d'errore tradizionale. I bit così codificati, poi vengono mappati sui qubit, che infine sono trasmessi su

un canale di comunicazione quantistico. La mappatura dei bit classici in qubit può essere effettuata, ad esempio, dal cosiddetto protocollo di codifica super-denso [116], [13], [22].

Al contrario, per un sistema di comunicazione più generale, cioè che supporta la trasmissione di informazioni sia classiche che quantistiche, per un calcolo quantistico affidabile, bisogna ricorrere ai Quantum Error Correction Codes (QECCs). In particolare, proprio come i codici di correzione degli errori tradizionali, i QECCs sfruttano la ridondanza, questa volta nel dominio quantistico, per correggere le alterazioni quantistiche, consentendo quindi ai qubit di mantenere con alta probabilità i loro stati inalterati, per periodi più lunghi. I QECCs sono, pertanto, indispensabili per concepire un sistema di comunicazione quantistica che supporti la trasmissione dell'informazione ed il calcolo quantistico.

1.3 Decoerenza quantistica

In generale, per la teoria della decoerenza quantistica, o desincronizzazione della funzione d'onda, l'interazione fra i sistemi quantistici e l'ambiente esterno determina la perdita della coerenza della funzione d'onda.

La decoerenza dovuta all'ambiente costituisce generalmente una fonte importante di alterazioni quantistiche; in particolare tale fenomeno impedisce l'osservazione della superposition di stati per i sistemi macroscopici. Il carattere destabilizzante dell'ambiente riguarda, di fatto, tutto ciò che può essere influenzato dallo stato del sistema quantistico (e quindi può inavvertitamente "misurarlo"), per esempio: un singolo fotone, la vibrazione di una molecola, le particelle dell'aria.

In questa teoria l'ambiente non è semplice rumore: esso si comporta come uno strumento che osserva costantemente il sistema; infatti esso non può distinguere tra il contatto casuale e il contatto intenzionale di una misura. Il sistema perde così la coerenza perché lascia sfuggire informazione e quindi rivela al resto dell'universo il suo stato [180].

In termini semplici, la decoerenza ambientale può essere descritta come l'interazione indesiderata, o più specificamente il coinvolgimento, del qubit da parte dell'ambiente, che altera la sua superposition coerente di stati di base. Si può verificare tale decoerenza, ad esempio, durante la trasmissione o l'elaborazione quantistica (cioè il processo di calcolo), nonché nelle memorie quantistiche.

In questa sezione, si esaminano i canali quantistici della Fig. 1.6, che sono ampiamente utilizzati per la modellazione della decoerenza ambientale. In particolare, si va ad indagare la dualità tra i canali quantistici e i canali classici.

1.3.1 *Canale di smorzamento dell'ampiezza*

Uno dei possibili casi di decoerenza quantistica dovuta all'ambiente si verifica quando il qubit perde energia a causa della sua interazione con l'am-

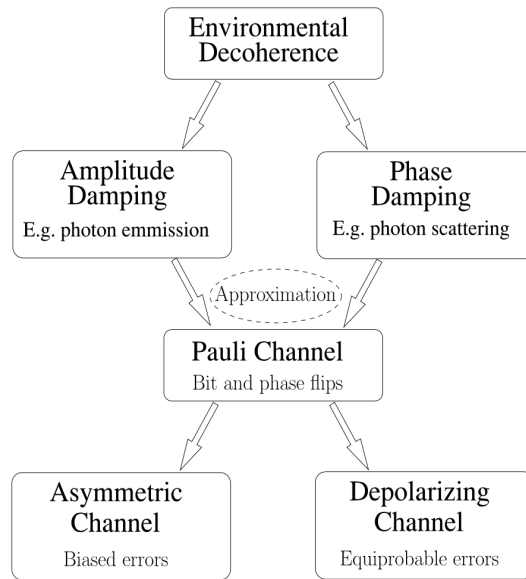


Figura 1.6: Modelli di canali quantistici [12].

biente; per esempio lo stato eccitato del qubit decade per via dell'emissione spontanea di un fotone. Questo processo di decoerenza può essere convenientemente modellato utilizzando un canale di smorzamento dell'ampiezza (Amplitude Damping Channel).

Si consideri quindi un qubit realizzato utilizzando un atomo a due livelli, come quello di Fig. 1.3, caratterizzato dallo stato fondamentale $|0\rangle$ e dallo stato eccitato $|1\rangle$ [116]. Invece l'ambiente, che è caratterizzato dallo stato di base $|0\rangle_E$, se non ci sono "fotoni liberi" oppure da $|1\rangle_E$ se ve ne sono, è inizialmente posto allo stato di vuoto $|0\rangle_E$ (vacuum state).

Supponendo di considerare un sistema costituito soltanto da un qubit e l'ambiente esterno (più semplicemente "ambiente"), si osserva che esiste una certa probabilità γ che il qubit decada dallo stato eccitato ($|1\rangle$) allo stato fondamentale ($|0\rangle$) e che quindi venga emesso un fotone nell'ambiente stesso. Tale cambiamento di stato del qubit è anche conosciuto come quantum jump, letteralmente salto quantistico [128] [41]. In questo modo l'ambiente compie una transizione dallo stato $|0\rangle_E$ (assenza di fotoni), allo stato $|1\rangle_E$ (presenza di un fotone), poiché l'ambiente acquisisce un fotone. Dunque, il canale di smorzamento dell'ampiezza caratterizza l'evoluzione del sistema risultante (qubit - ambiente) secondo la trasformazione unitaria (chiamata così perché preserva il prodotto vettoriale) che segue [128]:

$$|0\rangle |0\rangle_E \rightarrow |0\rangle |0\rangle_E; \quad (1.11)$$

$$|1\rangle |0\rangle_E \rightarrow \sqrt{1-\gamma} |1\rangle |0\rangle_E + \sqrt{\gamma} |0\rangle |1\rangle_E, \quad (1.12)$$

dove γ è, in particolare, la probabilità di smorzamento, o più specificamente la probabilità di perdere un fotone da parte del qubit.

In altri termini, l'eq.(1.11) implica che lo stato del qubit rimanga lo stesso, se si trova nello stato fondamentale $|0\rangle$, mentre potrebbe perdere un fotone,

con una probabilità pari a γ , quando è nello stato eccitato $|1\rangle$. In particolare, per descrivere questo ultimo caso, va considerata l'eq.(1.12) (perdita di un fotone da parte del qubit). Allora lo stato del qubit cambierà da $|1\rangle$ a $|0\rangle$, mentre quello dell'ambiente passerà da $|0\rangle_E$ a $|1\rangle_E$; così si ottiene lo stato $|0\rangle|1\rangle_E$ di eq.(1.12), che può verificarsi con probabilità $|\sqrt{\gamma}|^2 = \gamma$. Se invece, nel caso duale, il qubit nello stato eccitato $|1\rangle$ effettivamente non perde il fotone e quindi l'ambiente rimane privo di fotoni ($|0\rangle_E$), si ritrova giocoforza il caso di partenza con probabilità complementare ($|\sqrt{1-\gamma}|^2 = 1 - \gamma$) a quella descritta sopra. Pertanto, sulla base di quanto detto, si osserva che la somma delle due probabilità di cui sopra verifica la condizione di normalizzazione, infatti:

$$|\sqrt{\gamma}|^2 + |\sqrt{1-\gamma}|^2 = \gamma + 1 - \gamma = 1. \quad (1.13)$$

Sulla base delle equazioni precedenti, un generico qubit caratterizzato dalla superposition coerente dello stato eccitato e dello stato fondamentale $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, interagisce con l'ambiente come segue:

$$|\psi\rangle|0\rangle_E \rightarrow (\alpha|0\rangle + \beta\sqrt{1-\gamma}|1\rangle)|0\rangle_E + \beta\sqrt{\gamma}|0\rangle|1\rangle_E, \quad (1.14)$$

dove tale equazione è ottenuta combinando la (1.11) e la (1.12).

È opportuno ricordare che, generalmente, non si osserva mai un qubit isolato, infatti il generico qubit con stato $|\psi\rangle$, di norma, è correlato (entangled) con altri $N - 1$ qubit, che compongono un unico sistema quantistico di N qubit. Quindi, abbandonando per un momento il rigoroso formalismo della notazione usuale, i coefficienti α e β della (1.14), rappresentano gli stati degli altri $N - 1$ qubit del sistema, entangled con gli stati $|0\rangle$ e $|1\rangle$, rispettivamente, del qubit considerato, che è sottoposto a decoerenza. Inoltre si suppone che ogni qubit interagisca in modo indipendente con l'ambiente, quindi il processo di decoerenza associato è temporalmente e spazialmente incorrelato.

Si vuole ora spiegare il caso in cui il qubit emette un fotone nel passaggio dallo stato eccitato ($|1\rangle$) allo stato fondamentale ($|0\rangle$) e contestualmente, l'ambiente esterno assume lo stato $|1\rangle_E$. In effetti, stante l'ipotesi di cui sopra, è triviale osservare che lo stato iniziale del qubit è lo stato eccitato $|1\rangle$, infatti lo stato fondamentale non sarebbe mai potuto decadere, in quanto già privo di energia. Come appare evidente dal secondo addendo del secondo membro dell'eq.(1.14), in caso di salto quantistico il generico stato $|\psi\rangle$ del qubit collaserebbe a $|0\rangle$ con probabilità $|\beta\sqrt{\gamma}|^2 = \gamma\beta^2$.

D'altra parte, se non si rileva un fotone nell'ambiente esterno, esso si trova nello stato $|0\rangle_E$ e così il qubit decade nel seguente stato:

$$\alpha|0\rangle + \beta\sqrt{1-\gamma}|1\rangle, \quad (1.15)$$

a giustificare la quantità tra parentesi al primo termine del secondo membro della eq.(1.14). Normalizzando si riduce a:

$$\frac{\alpha}{\sqrt{1-\gamma\beta^2}}|0\rangle + \frac{\beta\sqrt{1-\gamma}}{\sqrt{1-\gamma\beta^2}}|1\rangle, \quad (1.16)$$

dove il coefficiente di normalizzazione $\sqrt{1 - \gamma\beta^2}$ è legato alla probabilità che non accada il salto quantistico, come poi sarà più chiaro nei passaggi successivi.

Approfondimento: notazione Bra-ket La notazione $\langle \cdot |$ (chiamata Bra) è legata alla sua duale $(| \cdot \rangle$, Ket) dalla seguente relazione:

$$c_a|a\rangle + c_b|b\rangle \leftrightarrow c_a^*\langle a| + c_b^*\langle b|, \quad (1.17)$$

dove il coefficiente c_a (c_b) è tale che il suo modulo al quadrato rappresenti la probabilità dello stato a (b) [3].

Se un sistema quantistico è un insieme statistico di stati puri, allora può essere descritto usando l'operatore di densità (chiamato anche matrice di densità) ρ , come segue:

$$\rho \equiv \sum_{i=1} p_i |\psi_i\rangle \langle \psi_i|, \quad (1.18)$$

dove $\sum_{i=1} p_i = 1$ e $0 < p_i \leq 1$ indica la probabilità di occorrenza dello stato i -esimo puro $|\psi_i\rangle$.

Approfondimento: stato puro e stato misto Un sistema quantistico il cui stato quantistico $|\psi\rangle$ è completamente noto, si dice che è (nel senso strettamente letterale del termine di "trovarsi") in uno stato *puro*. Quindi, lo stato puro è descritto da un singolo vettore di stato $|\psi\rangle$ e l'operatore di densità, chiamato anche matrice di densità, dell'eq.(1.18) si riduce a: $\rho = |\psi\rangle \langle \psi|$.

Altrimenti, l'operatore di densità ρ si trova in uno stato cosiddetto *misto*, in questo caso pertanto ρ è una mixture di un insieme di stati puri. È questo peraltro il caso più generale, descritto dall'eq.(1.18), cioè una somma probabilistica (non strettamente la superposition, nel senso specificato nella Sezione 1.1.1) di diversi stati puri $|\psi_i\rangle$. Ciò significa che non si conosce esattamente lo stato del sistema e può essere trovato nell' i -esimo stato puro $|\psi_i\rangle$ con una probabilità pari a p_i [116].

Analiticamente, esiste un semplice criterio per stabilire se uno stato è puro o misto. Uno stato si dice puro se verifica la seguente relazione:

$$\text{tr}(\rho^2) = 1, \quad (1.19)$$

altrimenti, se vale:

$$\text{tr}(\rho^2) < 1, \quad (1.20)$$

lo stato in analisi si dice misto.

Una precisazione: la terminologia "stato puro" è spesso usato in riferimento a un vettore di stato $|\psi\rangle$, per distinguerlo dall'operatore di densità ρ , il quale tipicamente viene usato per descrivere sistemi quantistici che non sono completamente noti. In particolare la matrice di densità ρ è usata come strumento per la descrizione dei singoli sottosistemi di un sistema quantistico composito [116].

La perdita di energia in un generico sistema quantistico descritto dall'eq.(1.18), può essere modellata utilizzando un canale di smorzamento di ampiezza N_{AD} , che mappa uno stato di ingresso, avente l'operatore di densità ρ , come segue [116]:

$$N_{AD}(\rho) = \mathbf{E}_0 \rho \mathbf{E}_0^\dagger + \mathbf{E}_1 \rho \mathbf{E}_1^\dagger. \quad (1.21)$$

Prima di procedere oltre è necessario spiegare il significato dell'operatore $(\cdot)^\dagger$. Data una matrice A ed indicando con A^T la sua trasposta e con l'asterisco $*$ l'operazione di coniugazione complessa di tutti i suoi elementi, la trasposta coniugata A^\dagger , è data da [118]:

$$A^\dagger = (A^T)^* = (A^*)^T. \quad (1.22)$$

In meccanica quantistica, l'*aggiunto* di un generico operatore A è definito come:

$$\langle \phi | A \psi \rangle = \langle A^\dagger \phi | \psi \rangle, \quad (1.23)$$

per tutti i vettori $|\psi\rangle$ e $|\phi\rangle$. Dunque, A risulta un operatore *auto-aggiunto* (self-adjoint) se vale: $A = A^\dagger$. Un operatore auto-aggiunto si può, inoltre, esprimere come:

$$A = \sum_{i=1}^n a_i P_i, \quad (1.24)$$

dove a_i è un autovalore di A e P_i è la corrispondente proiezione ortogonale sullo spazio degli autovettori con autovalore a_i [128].

Sempre continuando ad analizzare l'eq.(1.21), si definiscono i seguenti operatori, o matrici, di Kraus (chiamati anche operatori di errore), cioè \mathbf{E}_0 ed \mathbf{E}_1 , come segue:

$$\mathbf{E}_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \quad \mathbf{E}_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}. \quad (1.25)$$

In particolare, la matrice \mathbf{E}_0 risulta un operatore autoaggiunto poiché vale: $\mathbf{E}_0 = \mathbf{E}_0^\dagger$, mentre non si può certo affermare lo stesso per \mathbf{E}_1 . Dalla definizione degli operatori di Kraus, si può osservare che vale la seguente relazione:

$$\begin{aligned} \mathbf{E}_0^\dagger \mathbf{E}_0 + \mathbf{E}_1^\dagger \mathbf{E}_1 &= \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ \sqrt{\gamma} & 0 \end{pmatrix} \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1-\gamma \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \gamma \end{pmatrix} = \mathbf{I}, \end{aligned} \quad (1.26)$$

che può essere scritta in forma compatta come segue:

$$\sum_{k=0}^1 \mathbf{E}_k^\dagger \mathbf{E}_k = \mathbf{I}, \quad (1.27)$$

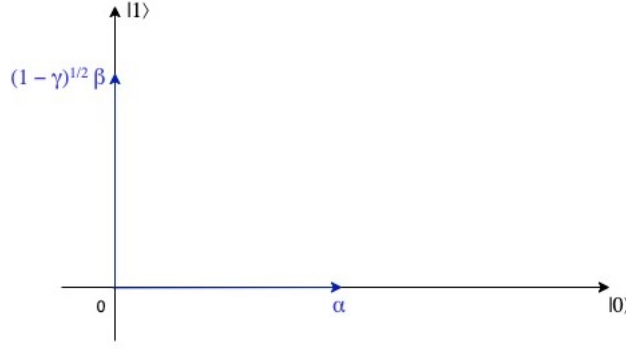


Figura 1.7: Rappresentazione grafica del vettore $\mathbf{E}_0 |\psi\rangle = \alpha |0\rangle + \sqrt{1-\gamma}\beta |1\rangle$.

dove \mathbf{I} è la matrice identità di dimensioni 2×2 . Per questo si dice che un generico canale quantistico N è un mapping lineare trace-persevering e completamente positivo, che mappa uno stato di input avente operatore di densità ρ . In generale, si ottiene:

$$N_{AD}(\rho) = \sum_{k=0}^1 \mathbf{E}_k \rho \mathbf{E}_k^\dagger. \quad (1.28)$$

Dunque, finita questa doverosa premessa, necessaria per affinare gli strumenti matematici da usare in seguito, si descrive lo stato decoerente di un generico qubit usando gli operatori di errore dell'eq.(1.25). Riprendendo il classico esempio di stato del qubit $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$, l'operatore di errore \mathbf{E}_0 agisce su $|\psi\rangle$ come segue:

$$\begin{aligned} \mathbf{E}_0 |\psi\rangle &= \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ \sqrt{1-\gamma}\beta \end{pmatrix} \\ &\equiv \alpha |0\rangle + \sqrt{1-\gamma}\beta |1\rangle, \end{aligned} \quad (1.29)$$

ritrovando lo stato del qubit espresso dall'eq.(1.15).

Supponendo ora di affidarsi all'interpretazione di $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$, come vettore composto da due elementi, gli stati $|0\rangle$ e $|1\rangle$ assumono il senso fisico di versori; si ottiene il grafico presente nella Fig. 1.7. Pertanto, da questo punto di vista, in notazione vettoriale il modulo corrisponde alla norma del vettore, cioè si ha: $\|\mathbf{E}_0 |\psi\rangle\|^2$. Dunque, per definizione di norma si ottiene la seguente relazione:

$$\|\mathbf{E}_0 |\psi\rangle\| = \sqrt{(|\alpha|)^2 + (\sqrt{1-\gamma}|\beta|)^2} = \sqrt{|\alpha|^2 + (1-\gamma)|\beta|^2}. \quad (1.30)$$

Elevando al quadrato l'eq.(1.30), si ha:

$$\|\mathbf{E}_0 |\psi\rangle\|^2 = |\alpha|^2 + (1-\gamma)|\beta|^2 = |\alpha|^2 + |\beta|^2 - \gamma|\beta|^2, \quad (1.31)$$

sapendo poi, come è stato dichiarato all'inizio, che $|\alpha|^2 + |\beta|^2 = 1$, si ottiene:

$$\|\mathbf{E}_0 |\psi\rangle\|^2 = 1 - \gamma|\beta|^2, \quad (1.32)$$

cioè:

$$|\mathbf{E}_0 |\psi\rangle|^2 = 1 - \gamma|\beta|^2 = 1 - \gamma\beta^2. \quad (1.33)$$

Dunque, lo stato corrotto dell'eq.(1.29) si verifica con probabilità di $1 - \gamma\beta^2$. In particolare, la moltiplicazione tra l'operatore \mathbf{E}_0 ed il generico stato del qubit $|\psi\rangle$, sta a significare che esiste una probabilità, pari a $|\sqrt{1 - \gamma}\beta|^2 = (1 - \gamma)|\beta|^2$, che il qubit si trovi nello stato $|1\rangle$. Dal punto di vista fisico infatti, tale espressione denota che il qubit rimane nello stato eccitato con quella certa probabilità, e quindi non perde il fotone. Tuttavia, l'eq.(1.29) contiene anche l'addendo $\alpha|0\rangle$ poiché, descrivendo il caso in cui non avviene la perdita del fotone, in principio il qubit potrebbe trovarsi nello stato fondamentale $|0\rangle$ con probabilità $|\alpha|^2$. In altre parole, \mathbf{E}_0 lascia lo stato $|0\rangle$ invariato, ma riduce la probabilità dello stato $|1\rangle$, poiché, come è facile osservare, vale: $\sqrt{1 - \gamma}\beta < \beta$. Fisicamente, questo accade perché il fotone non è stato perso nell'ambiente, e quindi, in questo caso, l'ambiente percepisce che è più probabile che il sistema sia nello stato $|0\rangle$, piuttosto che nello stato $|1\rangle$ [116].

Dopo la normalizzazione, l'espressione sopra riportata diventa:

$$\mathbf{E}_0 |\psi\rangle = \frac{\alpha}{\sqrt{1 - \gamma\beta^2}} |0\rangle + \frac{\beta\sqrt{1 - \gamma}}{\sqrt{1 - \gamma\beta^2}} |1\rangle, \quad (1.34)$$

che restituisce a sua volta l'eq.(1.16). Come si può osservare anche in questo caso, come nella eq.(1.16), la normalizzazione permette di ottenere la somma dei due moduli al quadrato unitaria. Infatti vale:

$$\begin{aligned} \left| \frac{\alpha}{\sqrt{1 - \gamma\beta^2}} \right|^2 + \left| \frac{\beta\sqrt{1 - \gamma}}{\sqrt{1 - \gamma\beta^2}} \right|^2 &= \frac{|\alpha|^2}{1 - \gamma|\beta|^2} + \frac{|\beta|^2(1 - \gamma)}{1 - \gamma|\beta|^2} \\ &= \frac{|\alpha|^2 + |\beta|^2(1 - \gamma)}{1 - \gamma|\beta|^2} \\ &= \frac{|\alpha|^2 + |\beta|^2 - \gamma|\beta|^2}{1 - \gamma|\beta|^2} \\ &= \frac{1 - \gamma|\beta|^2}{1 - \gamma|\beta|^2} = 1, \end{aligned} \quad (1.35)$$

poiché, come detto, vale $|\alpha|^2 + |\beta|^2 = 1$.

Analogamente al caso precedente, l'operatore di Kraus \mathbf{E}_1 modifica lo stato $|\psi\rangle$ come segue:

$$\mathbf{E}_1 |\psi\rangle = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \sqrt{\gamma}\beta \\ 0 \end{pmatrix} \equiv \sqrt{\gamma}\beta |0\rangle, \quad (1.36)$$

ritrovando il coefficiente moltiplicatore del secondo addendo dell'eq.(1.14). Ciò accade con probabilità pari a $|\mathbf{E}_1 |\psi\rangle|^2 = \gamma|\beta|^2$ e, come si può osservare, l'operatore \mathbf{E}_1 modifica la probabilità di occorrenza dello stato $|0\rangle$ del qubit. Non è un caso che $(\sqrt{\gamma}\beta)$ moltiplichino $|0\rangle$, infatti questa relazione, dal punto di vista fisico, indica che esiste una probabilità, pari a $|\sqrt{\gamma}\beta|^2 = \gamma|\beta|^2$,

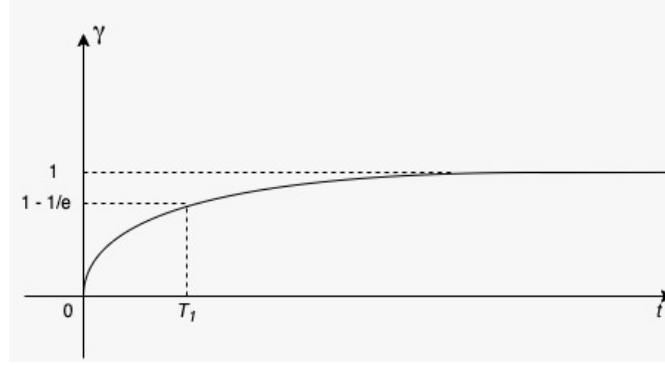


Figura 1.8: Andamento del tempo di rilassamento del qubit.

che il qubit passi, dallo stato eccitato $|1\rangle$, allo stato $|0\rangle$. In particolare, β è indispensabile in questa espressione, poiché descrive la probabilità che il qubit si trovasse nello stato $|1\rangle$ prima del salto quantistico, condizione necessaria per la perdita del fotone.

In definitiva, l'operatore \mathbf{E}_0 descrive il comportamento dello stato $|\psi\rangle$ del qubit qualora non ci sia perdita di energia (fotone), mentre \mathbf{E}_1 indica la situazione in cui si verifica il salto quantistico. Infatti, vale la seguente relazione:

$$|\mathbf{E}_0 |\psi\rangle|^2 + |\mathbf{E}_1 |\psi\rangle|^2 = (1 - \gamma|\beta|^2) + (\gamma|\beta|^2) = 1, \quad (1.37)$$

cioè la somma del modulo al quadrato della probabilità che accada il salto quantistico sommata alla probabilità che non succeda, rispetto naturalmente allo stato $|1\rangle$, restituisce l'unità, ritrovando l'eq.(1.35), come ci si aspetta in valutazione statistiche di questo tipo.

Osservazione Nei sistemi reali, la probabilità γ nell'istante t è caratterizzata dal tempo di rilassamento del qubit T_1 , come segue [62]:

$$\gamma = 1 - e^{-t/T_1}, \quad (1.38)$$

questo andamento è riportato graficamente in Fig. 1.8.

1.3.2 Canale di smorzamento di fase

Un'altra istanza di decoerenza ambientale, nota come dephasing o smorzamento di fase, caratterizza la perdita di informazione quantistica, da parte del qubit, senza perdita di energia. Può verificarsi ad esempio a causa della dispersione di fotoni durante il transito in una guida d'onda o a causa della perturbazione degli stati elettronici dei componenti, per effetto di cariche elettriche vaganti.

Gli operatori di errore del canale di smorzamento di fase N_{PD} sono definiti come segue [116]:

$$\mathbf{E}_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{pmatrix}, \quad \mathbf{E}_1 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{pmatrix}, \quad (1.39)$$

dove λ è la probabilità di scattering di un fotone, senza perdita di energia. In generale per *scattering* si intende il fenomeno fisico per cui la particella cambia traiettoria a causa della collisione quantistica con un'altra particella (o con un'onda).

Si può osservare che il comportamento di \mathbf{E}_0 dell'eq.(1.39) è simile all' \mathbf{E}_0 del canale di smorzamento dell'ampiezza, dunque vale:

$$\begin{aligned}\mathbf{E}_0 |\psi\rangle &= \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ \sqrt{1-\lambda}\beta \end{pmatrix} \\ &\equiv \alpha |0\rangle + \sqrt{1-\lambda}\beta |1\rangle.\end{aligned}\quad (1.40)$$

Pertanto, come nel caso precedente, questa espressione indica, dal punto di vista fisico, che il qubit rimane nello stato $|1\rangle$ con probabilità pari a $(1-\lambda)|\beta|^2$. Tuttavia, l'eq.(1.40) contiene anche l'addendo $\alpha|0\rangle$ poiché, descrivendo il caso in cui non avviene la perdita di informazione, in principio il qubit potrebbe trovarsi nello stato fondamentale $|0\rangle$ con probabilità $|\alpha|^2$. D'altro canto, l'operatore di errore \mathbf{E}_1 agisce su $|\psi\rangle$ come segue:

$$\mathbf{E}_1 |\psi\rangle = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 0 \\ \sqrt{\lambda}\beta \end{pmatrix} \equiv \sqrt{\lambda}\beta |1\rangle.\quad (1.41)$$

Ciò accade con una probabilità pari a $|\mathbf{E}_1 |\psi\rangle|^2 = \lambda|\beta|^2$ ed \mathbf{E}_1 è associato allo stato $|1\rangle$, diversamente rispetto al caso dell'amplitude damping channel. Dal punto di vista fisico, questa relazione spiega che esiste una probabilità, pari a $|\sqrt{\lambda}\beta|^2 = \lambda|\beta|^2$, che subisca lo scattering, a partire dallo stato $|1\rangle$. Al contrario di quanto accadeva nell'amplitude damping channel in questo caso, l'operatore \mathbf{E}_1 non cambia lo stato del qubit, da $|1\rangle$ a $|0\rangle$, ma ne riduce solo l'ampiezza, infatti vale che: $\sqrt{\lambda}\beta < \beta$ [116].

La probabilità λ è determinata sia sul tempo di rilassamento del qubit T_1 che sul tempo di dephasing T_2 , cioè si ha [62]:

$$\lambda = 1 - e^{-\frac{t}{T_1} - \frac{2t}{T_2}}.\quad (1.42)$$

Infine, si può osservare che l'eq.(1.38) e l'eq.(1.42) implicano che il qubit possa perdere la coerenza dello stato se il tempo di funzionamento (valido in fase di trasmissione o di elaborazione o di memorizzazione) t è equivalente al tempo di rilassamento T_1 e al tempo di dephasing T_2 . Dunque, se si fa tendere il tempo t al valore $T_1 = T_2 = T$, si osserva che il comportamento della probabilità di smorzamento γ è assimilabile a quello della probabilità di scattering λ ; entrambe tendono allo stesso valore. Si ottiene:

$$\lim_{t \rightarrow T} \gamma = \lim_{t \rightarrow T} 1 - e^{-\frac{t}{T}} = 1 - e^{-1} \simeq 0.63,\quad (1.43)$$

$$\lim_{t \rightarrow T} \lambda = \lim_{t \rightarrow T} 1 - e^{-\frac{t}{T} - \frac{2t}{T}} = 1 - e^{-1} \simeq 0.63.\quad (1.44)$$

Pertanto, T_1 e T_2 sono considerati due indicatori del tempo di vita di un qubit affidabile.

1.3.3 Canale di Pauli

La decoerenza ambientale può essere modellata al meglio combinando in un unico modello di canale l'amplitude damping channel ed il phase damping channel. Tuttavia, non è possibile simulare classicamente tali canali per un sistema composto da N qubit, poiché il sistema risultante avrebbe uno spazio di Hilbert di dimensione 2^N che sarebbe eccessivamente complesso. Pertanto, a causa di questo limite matematico, per facilitare la trattazione, tale combinazione può essere approssimata usando un cosiddetto canale di Pauli N_P , in onore del fisico elvetico Wolfgang Pauli che per primo lo ha teorizzato.

Un caso particolare di un canale di questo tipo è costituito da un singolo qubit che mappa uno stato di ingresso, avente l'operatore di densità ρ , dove gli operatori di Kraus sono semplicemente dati dalle matrici di Pauli, descritte di seguito [139]. Si ottiene:

$$N_P(\rho) = (1 - p_z - p_x - p_y)\mathbf{I}\rho\mathbf{I} + p_z\mathbf{Z}\rho\mathbf{Z} + p_x\mathbf{X}\rho\mathbf{X} + p_y\mathbf{Y}\rho\mathbf{Y}. \quad (1.45)$$

Esplicitamente \mathbf{I} , \mathbf{X} , \mathbf{Y} e \mathbf{Z} sono gli operatori (o matrici o porte) di Pauli ad un singolo qubit e sono definite come segue:

$$\mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad \mathbf{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \quad (1.46)$$

In particolare, \mathbf{X} modifica lo stato del generico qubit realizzando un'operazione di bit-flip (inversione del qubit), \mathbf{Z} opera un phase-flip (inversione della fase) e \mathbf{Y} agisce realizzando sia un bit-flip che un phase-flip. La matrice \mathbf{I} , invece non cambia lo stato del qubit. La Fig. 1.9 spiega l'effetto di questi operatori sulla sfera di Bloch. Ulteriori chiarimenti ed approfondimenti verranno forniti nel seguito di questa sezione.

Approfondimento: Operatori di Pauli In meccanica quantistica, le matrici di Pauli sono un insieme di matrici 2×2 complesse hermitiane unitarie, cioè delle matrici autoaggiunte che soddisfano la seguente relazione $A^\dagger A = AA^\dagger = \mathbf{I}$, dove A è una generica matrice ed \mathbf{I} è la matrice identità. Tipicamente, in letteratura possono trovarsi anche indicate dalla lettera greca σ , valendo le seguenti relazioni:

$$\sigma_0 \equiv \mathbf{I}, \quad \sigma_1 \equiv \sigma_x \equiv \mathbf{X}, \quad \sigma_2 \equiv \sigma_y \equiv \mathbf{Y}, \quad \sigma_3 \equiv \sigma_z \equiv \mathbf{Z}. \quad (1.47)$$

Queste matrici, inoltre, soddisfano le seguenti proprietà:

$$\sigma_0^2 = \sigma_1^2 = \sigma_2^2 = \sigma_3^2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \mathbf{I} \quad (1.48)$$

e

$$\begin{aligned} \sigma_1\sigma_2 &= i\sigma_3, & \sigma_2\sigma_1 &= i\sigma_3; \\ \sigma_2\sigma_3 &= i\sigma_1, & \sigma_3\sigma_2 &= i\sigma_1; \\ \sigma_3\sigma_1 &= i\sigma_2, & \sigma_1\sigma_3 &= i\sigma_2. \end{aligned} \quad (1.49)$$

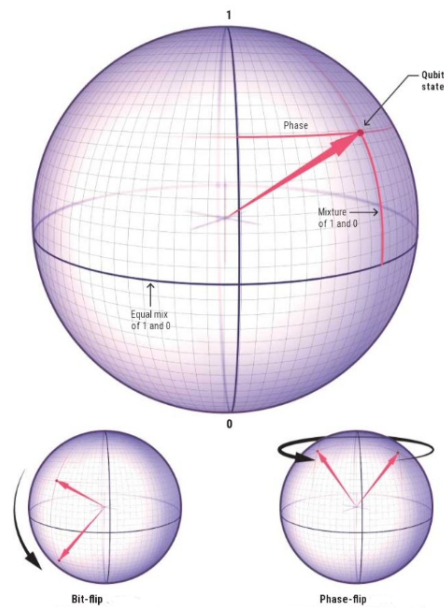


Figura 1.9: Generico stato del qubit, operazione di bit-flip e phase-flip sulla sfera di Bloch. *Bit-flip*: scambia lo stato $|0\rangle$ con lo stato $|1\rangle$; cambia la latitudine del qubit passando dall'emisfero nord a quello sud della sfera. *Phase-flip*: lo stato del qubit ruota di mezzo giro rispetto alla longitudine della sfera.

Inoltre, determinante e traccia di queste matrici risultano:

$$\det(\sigma_i) = -1, \quad \text{tr}(\sigma_i) = 0, \quad \forall i = 1, 2, 3, \quad (1.50)$$

pertanto si ricava semplicemente che gli autovalori delle tre matrici di Pauli sono ± 1 .

Infine $\sigma_1, \sigma_2, \sigma_3$, con l'aggiunta dell'identità, formano un insieme completo di matrici, ovvero una base B dello spazio delle matrici 2×2 hermitiane. Si ottiene:

$$B = c_0\sigma_0 + c_1\sigma_1 + c_2\sigma_2 + c_3\sigma_3, \quad (1.51)$$

dove c_0, c_1, c_2, c_3 sono i coefficienti complessi.

Si faccia poi riferimento allo schematico di Fig. 1.10 per comprendere il funzionamento di ognuna di queste matrici, a livello analitico e circuitale come porte logiche; in particolare, ogni operatore sarà descritto con dettaglio nel seguito.

In un canale di Pauli dunque, si hanno probabilità indipendenti p_x, p_y e p_z , tali che $p_x + p_y + p_z \leq 1$, per cui un qubit di input nello stato ρ sia sottoposto ad un errore di Pauli \mathbf{X} , \mathbf{Y} o \mathbf{Z} , rispettivamente. Queste probabilità si basano sul tempo di rilassamento del qubit (T_1) e sul tempo di dephasing

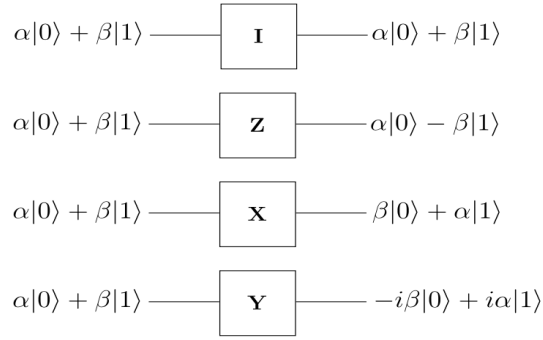


Figura 1.10: Schematico delle 4 porte logiche di Pauli [12].

(T_2), come riportato di seguito:

$$\begin{aligned}
 p_x = p_y &= \frac{1}{4}\gamma = \frac{1}{4}(1 - e^{-t/T_1}), \\
 p_z &= \frac{1}{4}(1 + e^{-t/T_1} - 2e^{-t/T_2}).
 \end{aligned} \tag{1.52}$$

Si può dunque osservare che il tempo T_1 influisce sugli errori di inversione del bit (bit-flip), di inversione della fase (phase-flip) e di inversione del bit e della fase (bit-and-phase flip). Al contrario, il tempo T_2 è relativo solo agli errori di inversione di fase. La probabilità che accada un errore di inversione di bit e quella che succeda un errore di inversione di bit e di fase sono associate alla probabilità di smorzamento γ (amplitude damping channel). Invece, la probabilità che si verifichi un errore di sola inversione di fase dipende sia da γ che dalla probabilità di scattering λ . Dato che $p_x = p_y$, gli errori di Pauli \mathbf{X} sono probabili (o improbabili) quanto gli errori \mathbf{Y} , poiché p_x , p_y e p_z rappresentano le probabilità di occorrenza degli errori di Pauli \mathbf{X} , \mathbf{Y} o \mathbf{Z} , rispettivamente.

In particolare poi si può notare che il rapporto tra p_z e p_x (o p_y) vale:

$$A = \frac{p_z}{p_x} = 1 + 2 \frac{1 - e^{t/T_1(1-T_1/T_2)}}{e^{t/T_1} - 1}, \tag{1.53}$$

il quale poi si riduce a $2(T_1/T_2) - 1$, se $t \ll T_1$. Tale rapporto A è estremamente importante, tanto che prende il nome di *parametro di asimmetria del canale*, poiché più il valore di A tende ad 1 e più il canale studiato è considerabile simmetrico.

Ora si analizza il funzionamento di ogni gate di Pauli che agisce sullo stato $|\psi\rangle$ di un generico qubit. Cominciando dall'operatore di identità \mathbf{I} , chiamato anche, più semplicemente, gate di ripetizione, si osserva che esso lascia intatto lo stato $|\psi\rangle$, come mostrato di seguito:

$$\mathbf{I}|\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \equiv \alpha|0\rangle + \beta|1\rangle. \tag{1.54}$$

L'operatore \mathbf{Z} invece, porta ad una inversione di fase (phase-flip) ed agisce

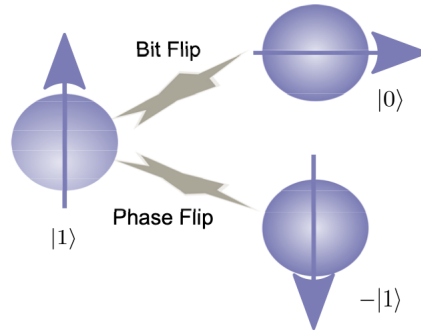


Figura 1.11: Bit-flip e phase-flip per lo stato di un generico qubit, rappresentati rispetto allo spin dell'elettrone. La polarizzazione verticale rappresenta lo stato $|1\rangle$, mentre la polarizzazione orizzontale rappresenta lo stato $|0\rangle$, il verso della freccia invece indica la fase [11].

come segue:

$$\mathbf{Z}|\psi\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \alpha \\ -\beta \end{pmatrix} \equiv \alpha|0\rangle - \beta|1\rangle, \quad (1.55)$$

infatti, rispetto a quanto accaduto con la matrice identità \mathbf{I} , in questo caso l'addendo $\beta|1\rangle$ risulta cambiato di segno, in ragione del $-$, che compare nell'elemento $z_{2,2}$ della matrice \mathbf{Z} ; questo funzionamento è ulteriormente spiegato in Fig. 1.11.

\mathbf{X} , diversamente, è un operatore di inversione di bit (bit-flip), risulta dunque analogo alla porta logica "NOT" tradizionale. Pertanto si ottiene:

$$\mathbf{X}|\psi\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix} \equiv \beta|0\rangle + \alpha|1\rangle, \quad (1.56)$$

come si può notare, infatti, il coefficiente β ora è associato allo stato $|0\rangle$ invece che a $|1\rangle$ e di conseguenza α moltiplica $|1\rangle$ al posto di $|0\rangle$. Di nuovo si guardi alla Fig. 1.11 per una rappresentazione schematica del fenomeno. Infine, \mathbf{Y} si ricava dal prodotto tra l'operatore \mathbf{X} e \mathbf{Z} , premoltiplicati per l'unità immaginaria i (in particolare, vale: $\mathbf{Y} = i\mathbf{XZ}$). Dunque, \mathbf{Y} è dato dalla combinazione tra un operatore di inversione di bit ed un operatore di inversione di fase, per questo agisce su $|\psi\rangle$ nel seguente modo:

$$\mathbf{Y}|\psi\rangle = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} -i\beta \\ i\alpha \end{pmatrix} \equiv -i(\beta|0\rangle - \alpha|1\rangle). \quad (1.57)$$

Pertanto, la matrice \mathbf{Y} viene chiamata operatore di inversione di bit e di fase; è responsabile appunto del fenomeno del bit-and-phase-flip.

Anche se apparentemente sembra incomprensibile, la moltiplicazione per l'unità immaginaria i è indispensabile. La natura della questione è puramente matematica ed ora verrà spiegata. Anzitutto, si osserva che l'operazione di bit-and-phase-flip si raggiunge, intuitivamente, moltiplicando \mathbf{X}

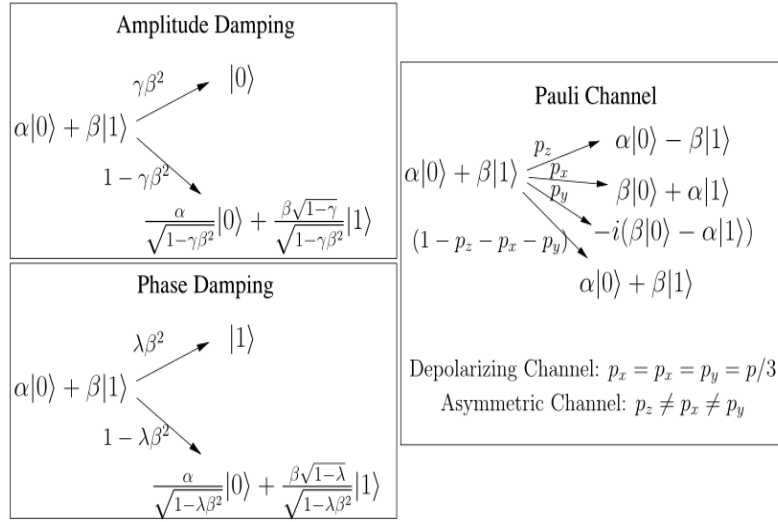


Figura 1.12: Interpretazione dei principali modelli dei canali quantistici argomentati [12].

per \mathbf{Z} , così si ottiene:

$$\xi = \mathbf{XZ} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad \xi' = \mathbf{ZX} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad (1.58)$$

dato che il prodotto tra matrici non è commutativo e quindi vanno calcolate entrambe le soluzioni. Effettivamente però, i due operatori ξ e ξ' , sebbene siano entrambe matrici hermitiane autoaggiunte, come \mathbf{I} , \mathbf{X} e \mathbf{Z} , non verificano entrambe le relazioni dell'eq.(1.50), in particolare quella sul determinante, dato che risulta $\det(\xi) = \det(\xi') = 1$, quindi non può essere utilizzata né ξ né ξ' per creare la base dello spazio. Questo problema pertanto viene superato premoltiplicando ξ per i ed ottenendo la matrice \mathbf{Y} di cui sopra. Infatti, data $\mathbf{Y} = i\xi = i\mathbf{XZ}$, sono verificati i requisiti sulla traccia e sul determinante dell'eq.(1.50), quindi \mathbf{Y} può far parte della base di cui sopra.

Quindi, riprendendo l'eq.(1.45) dalla quale si era partiti, si può notare che il canale di Pauli, durante il processo di decoerenza, effettivamente mappa lo stato di ingresso ($|\psi\rangle$) su di una combinazione lineare dello stato originale (operatore \mathbf{I}), dello stato di inversione di fase (operatore \mathbf{Z}), dello stato di inversione di bit (operatore \mathbf{X}) ed infine dello stato di inversione di bit e di fase (operatore \mathbf{Y}), pesati in base al valore delle probabilità p_z , p_x e p_y .

Richiamando le nozioni di tempo di rilassamento del qubit (T_1) e di tempo di dephasing (T_2), si nota che nella maggior parte dei sistemi reali, il valore di T_1 è superiore, di diversi ordini di grandezza, rispetto a quello di T_2 [166], [114]. Di conseguenza, questi sistemi quantistici si comportano come canali asimmetrici (si guardi A , parametro di asimmetria del canale, definito in precedenza) e dunque realizzano più inversioni di fase che inversioni di bit o inversioni di bit e fase.

Inoltre, una speciale categoria di canale di Pauli, conosciuta come *cana-*

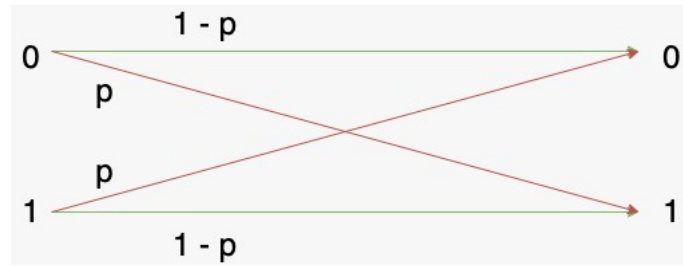


Figura 1.13: Canale binario simmetrico.

le depolarizzante (depolarizing channel) N_{DP} , modella lo scenario peggiore ipotizzando che tutti e tre gli errori siano equiprobabili, cioè $p_z = p_x = p_y = p/3$. Pertanto si tratta di un canale simmetrico ($A = 1$), diversamente da quello descritto sopra. In particolare, un canale depolarizzante è caratterizzato da una probabilità, detta di *depolarizzazione*, pari a $p/3$ di causare un errore di inversione di fase (operatore di Pauli \mathbf{Z}) o di inversione di bit (operatore di Pauli \mathbf{X}) o di inversione di bit e di fase (operatore di Pauli \mathbf{Y}). Riprendendo la notazione utilizzata per definire il canale di Pauli, il canale depolarizzante di cui sopra può essere rappresentato matematicamente nel seguente modo:

$$N_{DP}(\rho) = (1 - p)\rho + \frac{p}{3}(\mathbf{Z}\rho\mathbf{Z} + \mathbf{X}\rho\mathbf{X} + \mathbf{Y}\rho\mathbf{Y}), \quad (1.59)$$

dunque si può affermare che con probabilità $1 - p$ lo stato del qubit rimane immutato, mentre con probabilità di depolarizzazione pari a p si verifica sicuramente uno dei tre possibili errori. I suddetti modelli di canali quantistici sono riassunti nella Fig. 1.12.

Si può osservare che, producendo quattro possibili uscite, il canale di Pauli è considerato l'analogo quantistico del canale classico quaternario discreto. Tuttavia, mentre questo può determinare solo uno dei quattro possibili errori, l'errore causato dal canale di Pauli può essere interpretato come la superposition dei quattro possibili errori, vale a dire, in base agli operatori, \mathbf{I} , \mathbf{Z} , \mathbf{X} e \mathbf{Y} .

Il canale di Pauli può essere ulteriormente semplificato utilizzando due canali indipendenti: un canale di inversione del bit ed un canale di inversione della fase, che sono analoghi ai canali classici binari simmetrici, con probabilità di cross-over pari a $(p_x + p_y)$ e $(p_z + p_y)$, rispettivamente. In particolare, il canale classico binario simmetrico, riportato in Fig. 1.13, è caratterizzato da una sorgente binaria che emette solo due possibili simboli; per esempio il bit 0 ed il bit 1. Una volta emesso il simbolo, questo può arrivare a destinazione invariato oppure subire una distorsione e commutare da 0 a 1 o viceversa, con probabilità di cross-over pari a p , a causa della possibile insorgenza di errori. Pertanto p è la probabilità di errore, mentre la probabilità che il simbolo giunga corretto a lato ricevitore è pari a $1 - p$.

Dunque, nel caso del canale di bit-flip, la probabilità di incorrere in un errore è $(p_x + p_y)$, che è la somma della probabilità di inversione del bit e di inversione del bit e della fase del canale di Pauli. Analogamente, per

il canale di phase-flip calato nel dominio classico, la probabilità di errore coincide con $(p_z + p_y)$, cioè la somma della probabilità di phase-flip e di bit-and-phase-flip. In entrambi i casi è, giocoforza, presente p_y , che contiene sia il caso dell'inversione del bit che della fase e quindi può verificarsi in entrambi i modelli di canale.

Capitolo 2

Panoramica storica dei codici di correzione degli errori classici e quantistici

In questo capitolo verranno citati, e in alcuni casi genericamente esaminati, i principali esempi di codice della teoria classica e quantistica.

2.1 Teoria della codifica classica

2.1.1 *Obiettivi di progetto*

La teoria della codifica classica inizia grazie al lavoro pionieristico "A mathematical theory of communication", di Claude Shannon del 1948 [143]. In particolare, uno dei suoi contributi più significativi è quello legato al cosiddetto "Teorema di Codifica della Sorgente", per il quale vale che una generica sorgente con entropia $H(X)$ può essere rappresentata, senza perdita di informazione, utilizzando un numero medio di simboli, per segnale, pari a N , dove $N \geq H(X)$. Altrimenti, l'informazione è sicuramente distorta, per quanto complesse possano essere le tecniche di rappresentazione utilizzate [129]. Successivamente vennero elaborate, sulla base dei limiti teorici proposti da Shannon, sofisticate tecniche di codifica di canale che permettevano di ottenere una trasmissione affidabile, con un rate di codifica R inferiore alla capacità C di un generico canale AWGN a banda limitata (limite di Shannon: $R < C$). In questo modo è possibile trasmettere informazioni virtualmente esenti da errori (cioè senza distorsione); d'altra parte se $R > C$, non sarebbe possibile conseguire una trasmissione completamente affidabile, indipendentemente dal processo di elaborazione del segnale. La capacità C (bits/s) viene modellata come segue:

$$C = W \log_2 \left(1 + \frac{S}{N} \right), \quad (2.1)$$

dove W (Hz) è la larghezza di banda del canale, S (Watt) è la potenza del segnale e N (Watt) è la potenza di disturbo data dal rumore gaussiano

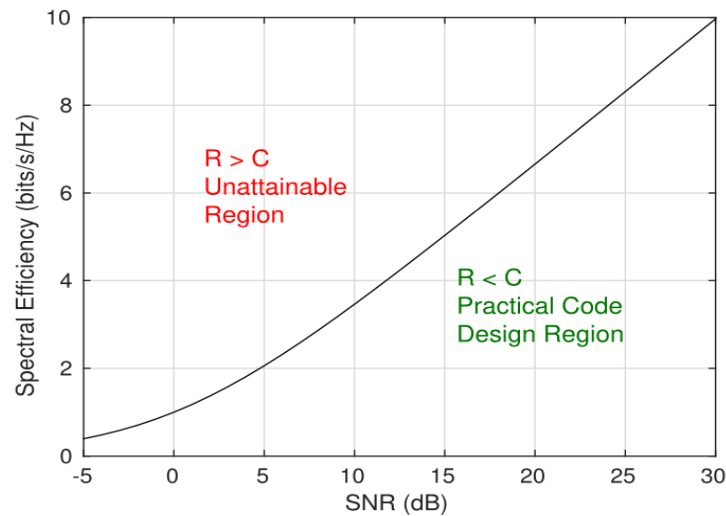


Figura 2.1: Limite di Shannon per la capacità di un generico canale AWGN [12].

incorrelato bianco e additivo (AWGN) che influenza il canale.

Tale relazione può essere scritta equivalentemente in termini di efficienza spettrale η , come rapporto tra la capacità C e la banda del canale W . Si ottiene:

$$\eta = \frac{C}{W} = \log_2 \left(1 + \frac{S}{N} \right) = \frac{1}{\ln(2)} \ln \left(1 + \frac{S}{N} \right) \approx 1.44 \ln \left(1 + \frac{S}{N} \right). \quad (2.2)$$

Di conseguenza, il limite di Shannon espresso dall'eq.(2.1) (e in modo equivalente dall'eq.(2.2)) quantifica i più alti rate di codifica (di trasmissione) possibili in grado di garantire una trasmissione senza errori, come illustrato in Fig. 2.1 [129].

Osservazione Inoltre, si può dedurre dall'eq.(2.1) che la velocità di trasferimento delle informazioni C di un sistema dipende dalla larghezza di banda del canale W e dal rapporto segnale-rumore (SNR) $\frac{S}{N}$ del sistema.

In particolare, si verifica che il valore di C cresce all'aumentare dell'SNR e della larghezza di banda del canale W . Dunque, quando l'SNR tende ad infinito, poiché la potenza di rumore N tende a zero, è possibile ottenere una velocità di trasmissione C , potenzialmente infinita, anche per una larghezza di banda del canale W ridotta. Tuttavia, per $\frac{S}{N} \rightarrow 0$ si ha che $\eta \rightarrow 1,44 \cdot \frac{S}{N}$, per gli sviluppi in serie di Taylor della funzione logaritmo naturale ($\ln(1+x) \rightarrow x - \frac{x^2}{2} + \frac{x^3}{3} + \dots + (-1)^{n-1} \frac{x^n}{n}$) [129].

In definitiva, l'obiettivo è quello di trovare un compromesso tra la larghezza di banda del canale W e l'SNR, in quanto, una larghezza di banda infinita non garantisce una velocità di trasmissione infinita, poiché la potenza del rumore aumenta anche all'aumentare della larghezza di banda [76].

L'opera pionieristica di Shannon, sebbene fosse priva di esempi espliciti di codice, negli anni successivi ha ispirato la ricerca della comunità scientifica

nell'ambito della progettazione di codici, nel rispetto del limite pratico da lui teorizzato [143]. Questo, a sua volta, ha evidenziato vari altri parametri da tenere in considerazione che, essendo in conflitto tra loro, bisogna far coesistere raggiungendo dei compromessi in fase di progettazione. Sono elencati di seguito:

- BER (Bit Error Rate, cioè il rapporto tra il numero di bit non ricevuti correttamente ed il numero di bit trasmessi);
- Coding Gain (Guadagno di codifica, quantifica la riduzione in termini di energia per bit ottenuta ad un certo BER, applicando il codice);
- Complessità di implementazione;
- Ritardo di trasmissione;
- Throughput effettivo (quantità di dati che viene effettivamente trasmessa in una certa unità di tempo);
- Larghezza di banda del sistema;
- Caratteristiche del canale.

Per esempio, date le particolari condizioni del canale, un codice può essere ottimizzato per ottenere un BER più basso o un guadagno di codifica più alto. Tuttavia, ciò di solito impone un aumento della complessità di decodifica e del ritardo di trasmissione, oppure una riduzione del throughput effettivo [76].

Il limite di Shannon dell'eq.(2.1) quantifica la capacità C di un canale Continuous-input Continuous-output Memoryless (CCMC), cioè un canale caratterizzato da un ingresso e da un'uscita continua, senza memoria, che può essere ottenuto solo con codici random infinitamente lunghi o con tecniche di trasmissione adeguate. Dal momento che i sistemi di comunicazione all'avanguardia trasmettono informazioni binarie (bit), sono stati ricavati diversi altri limiti per caratterizzare sia il trade-off *rate-versus-minimum-distance* (compromesso tra il rate e la distanza minima), che il trade-off *rate-versus-SNR* (compromesso tra il rate e l'SNR), che si rifà a quanto rappresentato nella Fig. 2.1. Queste soglie forniscono un limite superiore o inferiore al rate di codifica massimo $R_{max} = k/n$, data la distanza di Hamming, intesa come distanza minima d_{min} , o viceversa.

Approfondimento: Distanza di Hamming e distanza euclidea minima La *distanza di Hamming* rappresenta la distorsione; è una misura della distanza tra il segnale x e la sua rappresentazione \hat{x} :

$$d_H(x, \hat{x}) = \begin{cases} 0, & \text{se } x \neq \hat{x} \\ 1, & \text{se } x = \hat{x}, \end{cases} \quad (2.3)$$

ed è considerata una distanza "hard". Invece la distanza "soft" si calcola come:

$$d(x, \hat{x}) = (x - \hat{x}). \quad (2.4)$$

Per esempio, la distanza di Hamming d_{ij} tra due parole di codice v_i e v_j rappresenta il numero di posizioni per cui le due parole differiscono. In particolare, si definisce come *distanza minima di Hamming* la minima tra le distanze di Hamming, calcolate tra ogni possibile coppia di parole di codice.

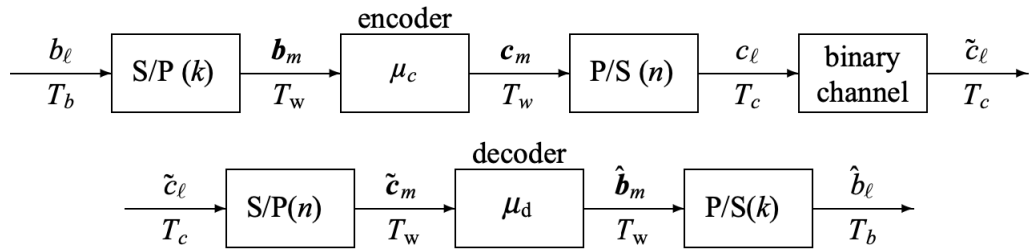


Figura 2.2: Schema a blocchi di un sistema con codifica a blocco binaria (n, k) [23].

Infine si definisce come *distanza minima euclidea* il quadrato della distanza minima d_{min}^2 .

Approfondimento: Codifica a Blocco In particolare, facendo riferimento alla teoria della codifica di canale e nello specifico alla *codifica a blocco*, k e n denotano, rispettivamente, il numero di bit di informazione e la lunghezza della sequenza codificata. I bit sono sempre k ; n si esprime in simboli ed è la lunghezza della parola di codice, con $n > k$, mentre la distanza di Hamming è definita come la minima distanza tra due qualsiasi parole di codice. Dunque, si andranno ad aggiungere $r = n - k$ simboli che non portano informazione (simboli di ridondanza), ma che sono necessari ai fini della rivelazione e della correzione degli errori [129]. Il codice risultante, nel caso di decodifica hard, è in grado di correggere fino a:

$$\begin{cases} t = \frac{d_{min}-1}{2}, & \text{se } d_{min} \text{ è dispari} \\ t = \frac{d_{min}-2}{2}, & \text{se } d_{min} \text{ è pari} \end{cases}$$

errori [129]. Dunque, per comprendere al meglio un'importante famiglia di codici a blocco, cioè i codici a blocco binari, si è riportata la Fig. 2.2, che ne tratteggia il funzionamento lato trasmettitore e lato ricevitore [23].

Inizialmente, la sequenza di simboli trasmessi (*messaggio di informazione*) è indicata con b_ℓ ed è caratterizzata da un rate di $1/T_b$, essendo T_b la durata di un bit. Poi viene frammentata in blocchi separati di k simboli (binari) ciascuno, chiamati *sequenze di informazione*, indicate con la notazione \mathbf{b}_m , caratterizzate ognuna da una durata $T_w = kT_b$. Ogni sequenza di informazione è poi associata ad una parola di codice di n simboli binari tramite il generico mapping:

$$\mu_c : \mathcal{A}^k \mapsto \mathcal{A}^n, \quad \mathcal{A} = \{0, 1\}. \quad (2.5)$$

All'uscita del codificatore (encoder) è presente la *parola di codice* (codeword), indicata con la notazione \mathbf{c}_m . L'insieme di tutte le possibili parole di codice è il *codice a blocco* $C(n, k)$ [23].

I simboli della parola di codice c_ℓ sono poi trasmessi in sequenza attraverso un canale binario al rate di $1/T_c$, con $T_c = T_w/n = kT_b/n$. All'uscita del canale, i *simboli ricevuti* $\{\tilde{c}_\ell\}$ in generale differiscono dai corrispondenti

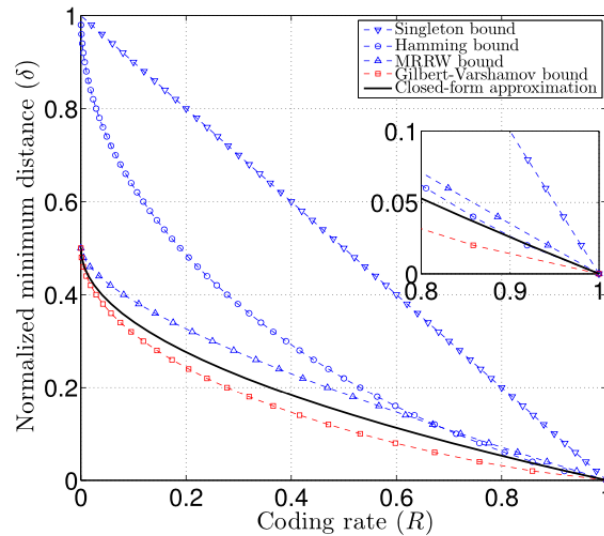


Figura 2.3: Rate (R) versus distanza minima normalizzata (δ) [12].

simboli trasmessi $\{c_\ell\}$ a causa degli errori introdotti dal canale. Al ricevitore, la sequenza $\{\tilde{c}_\ell\}$ è, dualmente al caso della trasmissione, suddivisa in "parole" $\{\tilde{c}_m\}$ costituite da n simboli binari ciascuna. Successivamente, il decodificatore, in base alla legge che lo caratterizza, decodifica le parole $\{\tilde{c}_m\}$ che riceve in ingresso e restituisce $\hat{b}_m = \mu_c^{-1}(\tilde{c}_m)$, ricostruendo, ove possibile, la parola di informazione c_m . Per un generico decoder, vale il seguente mapping:

$$\mu_d : \mathcal{A}^n \mapsto \mathcal{A}^k, \quad \mathcal{A} = \{0, 1\}, \quad (2.6)$$

dove k/n è il *code rate*. Infine la parola decodificata \hat{b}_m viene convertita in un segnale \hat{b}_ℓ , che deve essere il più fedele possibile al segnale di input iniziale b_ℓ , cioè il messaggio di informazione [23].

Approfondimento: Matrice Generatrice e Matrice di Parità In un codice a blocco lineare $C(n, k)$ si considerino le parole di codice corrispondenti alle sequenze di informazione $e_1 = (1000\dots 0)$, $e_2 = (0100\dots 0)$, $e_3 = (0010\dots 0)$, ..., $e_k = (0000\dots 1)$.

Esse possono essere contrassegnate da $g_1, g_2, g_3, \dots, g_k$, rispettivamente, dove ogni sequenza g_i è una sequenza binaria di lunghezza n . Dunque, ogni sequenza di informazione $x = (x_1, x_2, x_3, \dots, x_k)$ può essere scritta come:

$$x = \sum_{i=1}^n x_i e_i \quad (2.7)$$

e, quindi, la parola di codice corrispondente sarà:

$$c = \sum_{i=1}^n x_i g_i. \quad (2.8)$$

Dunque, si può definire la matrice generatrice per questo codice, come:

$$\mathbf{G} \triangleq \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{bmatrix}, \quad (2.9)$$

dunque, vale:

$$c = x\mathbf{G}. \quad (2.10)$$

Questo dimostra che qualsiasi combinazione lineare delle righe della matrice generatrice è una parola di codice. Per qualsiasi codice a blocco lineare, la matrice generatrice \mathbf{G} è una matrice di dimensioni $k \times n$ di rango k . Pertanto, la matrice generatrice di un codice descrive completamente il codice [129].

Per definizione un codice lineare a blocco C è un sottospazio lineare k -dimensionale dello spazio n -dimensionale. Dall'algebra lineare, è noto che se si considerano tutte le sequenze di lunghezza n che sono ortogonali a tutti i vettori di questo sottospazio lineare di dimensione k , il risultato sarà un sottospazio lineare $(n - k)$ -dimensionale chiamato complemento ortogonale del sottospazio k -dimensionale. Tale sottospazio $(n - k)$ -dimensionale definisce naturalmente un codice lineare $(n, n - k)$, che è noto come il duale del codice $C(n, k)$ iniziale. Il codice duale è indicato con C^\perp . Dunque, le parole di codice del codice originale C e del codice duale C^\perp sono ortogonali l'una rispetto all'altra. In particolare, se si denota con \mathbf{H} la matrice generatrice del codice duale C^\perp , che è una matrice di dimensioni $(n - k) \times n$, allora qualsiasi parola di codice del codice $C(n, k)$ è ortogonale a tutte le righe di \mathbf{H} ; cioè,

$$c\mathbf{H}^T = \mathbf{0}, \quad \forall c \in C, \quad (2.11)$$

dove \mathbf{H}^T indica la matrice trasposta di \mathbf{H} . La matrice \mathbf{H} , è chiamata *matrice di parità* (parity check matrix, conosciuta anche come PCM) del codice C . Poiché tutte le righe di \mathbf{G} sono parole di codice, si conclude che [129]:

$$\mathbf{G}\mathbf{H}^T = \mathbf{0}. \quad (2.12)$$

Nella Tabella 2.1 sono riportati sia i più popolari limiti di codifica dei codici a blocco di lunghezza finita, sia i limiti di codifica per i codici a blocco con comportamento asintotico ($n \rightarrow \infty$), mentre la Fig. 2.3 grafica solo i limiti asintotici. La distanza minima normalizzata $\delta = \frac{d_{min}}{n}$ è una distanza conforme a tutti i limiti teorici conosciuti, sia in contesti finiti (di lunghezza n) che in contesti asintotici ($n \rightarrow \infty$), che fornisce, in entrambi i casi, uno strumento pratico per la progettazione e la caratterizzazione di codici binari efficienti. Come si può osservare dalla Fig. 2.3, δ (asse verticale) è funzione del rate $R = k/n$ (asse orizzontale). In particolare, l'approssimazione in forma chiusa (Closed-Form Approximation), che è quella plottata in nero, si definisce proprio sulla base di δ e di n e consiste in un'espressione analitica invertibile siffatta [5]:

$$r(\delta, n) = (2\delta - 1)^2. \quad (2.13)$$

Tabella 2.1: Limiti del rate-versus-minimum-distance per i codici classici, dove H_2 indica l'entropia binaria [5], [39].

Classical Coding Bound	Finite-Length	Asymptotic
Singleton [149]	$\frac{k}{n} \leq 1 - \frac{d_{min}-1}{n}$	$\frac{k}{n} \leq 1 - \frac{d_{min}}{n}$
Hamming [74]	$\frac{k}{n} \leq 1 - \frac{1}{n} \log_2(\sum_{j=0}^{t=\lfloor \frac{d_{min}-1}{2} \rfloor} \binom{n}{j})$	$\frac{k}{n} \leq 1 - H_2(\frac{d_{min}}{2n})$
MRRW [109]		$\frac{k}{n} \leq H_2(\frac{1}{2} - \sqrt{\frac{d_{min}}{n}(1 - \frac{d_{min}}{n})})$
Plotkin [123]	$\frac{k}{n} \leq \frac{1}{n}(1 - \log_2(2 - \frac{n}{d_{min}}))$	
Gilbert-Varshamov (GV) [63]	$\frac{k}{n} \geq 1 - \frac{1}{n} \log_2(\sum_{j=0}^{d_{min}-1} \binom{n}{j})$	$\frac{k}{n} \geq 1 - H_2(\frac{d_{min}}{n})$

In altri termini, è una semplice funzione quadratica che soddisfa tutti i limiti asintotici [5], come per esempio; i limiti superiori MRRW [109], Hamming [74] e Singleton [149] (tracciati in blu), come anche il limite inferiore di Gilbert-Varshamov [63] (tracciato in rosso), riassunti nella Tabella 2.1. Questa curva serve per approssimare in maniera ottimale il trade-off tra il coding rate R e la distanza minima d_{min} .

Più specificamente, il limite di Singleton è un limite superiore lasco, sarebbe cioè la retta $\delta = R$ della Fig. 2.3. La teoria di questo limite dimostra che un codice di correzione degli errori di ordine q generico, dove con ordine q si intende un codice basato su q simboli, con $N = q^k$ parole di codice, ognuna di lunghezza $n = k + r$, non può avere una distanza minima d_{min} superiore a $r + 1$, dove $r = n - k$ [149]. Diversamente, il limite di Gilbert-Varshamov (GV bound) è il limite inferiore più aderente (nel senso di preciso) [63]. Infine, il limite di Hamming [74] fornisce un limite superiore stretto a rate di codifica elevati, mentre il limite MRRW (McEliece-Rodemich-Rumsey-Welch) è il limite superiore più aderente, utilizzato per i rate di codifica medio-bassi [109]. I limiti della Tabella 2.1 forniscono un insieme di distanze minime realizzabili, o più specificamente di distanze minime normalizzate δ , per il rate di codifica desiderato R .

2.1.2 Codici per la correzione degli errori

Nel 1950 Hamming concepì la prima famiglia di codici per la correzione degli errori classici [74]. Più specificamente, egli propose una famiglia di codici a blocco lineari binari, in grado di codificare $k = (2^r - 1 - r)$ bit di informazione in $n = (2^r - 1)$ simboli codificati, per $r \geq 2$. La parola di codice così risultante ha una distanza minima di Hamming pari a $d_{min} = 3$, quindi si possono correggere fino a $t = (d_{min} - 1)/2 = 1$ errore. I codici correttori di Hamming possono essere classificati come codici perfetti, secondo il limite di Singleton, dato che il rate di codifica associato $R = k/n = 1 - r/(2^r - 1)$ è il rate di codifica massimo ottenibile per $d_{min} = 3$ e per una lunghezza di blocco pari a $n = 2^r - 1$.

A seguito di questi sviluppi, nel 1954, Reed e Muller concepirono, indipendentemente, una classe di codici a blocco per la correzione di errori multipli, noti come codici Reed-Muller (RM) [130] e [111]. Reed ha inoltre introdotto, per gli stessi codici, un semplice decodificatore a decisione "hard" basato sulla logica a maggioranza [130]. Lo stesso anno, è stato sviluppato un algoritmo di decodifica a decisione "soft", noto come decodifica Wagner, per una classe speciale di codici RM [148].

I suddetti codici a blocco lineari si basavano principalmente sulla massimizzazione della distanza minima, per una determinata coppia (n, k) , o sulla massimizzazione equivalente del rate di codifica R , data la d_{min} e n . Per questi motivi, i codici risultanti dalle famiglie di codici di Hamming e Reed-Muller, supportano solo una gamma limitata di parametri di codice forniti dal trinomio (n, k, d_{min}) .

Al fine di progettare dei codici che offrissero una gamma più ampia di parametri, con una complessità di implementazione accessibile, Elias scoprì i codici convoluzionali (convolutional codes) nel 1955, e segnò l'inizio della cosiddetta era della *codifica probabilistica* [53].

I codici convoluzionali, sono una classe di codici usata frequentemente nelle applicazioni, poiché supportano le procedure di codifica e decodifica in maniera più efficiente. Infatti operano in una sliding-window, con conseguente latenza inferiore rispetto ai codici a blocco sopra indicati. In questo caso, lo schema di riferimento per il sistema è ancora quello della Fig. 2.2, con il mapping μ_c sostituito dal filtro $g(hT_w)$, rappresentato da una matrice di dimensioni $n \times k$. Questa famiglia di codici si chiama "convoluzionale" perché, come del resto espresso dall'eq.(2.14), la codifica è equivalente ad effettuare operazioni di convoluzione (discreta). La relazione ingresso-uscita per l'encoder è quindi:

$$c_m = \sum_{h=0}^{\nu} g(hT_w)b_{m-h}, \quad (2.14)$$

dove c_m è la parola di output, che dipende da b_m , che è la sequenza di informazione corrente, e da tutte le precedenti ν sequenze di informazione [23].

Successivamente, Viterbi ha proposto un algoritmo di stima della sequenza, a massima verosimiglianza (MLSE, Maximum Likelihood Sequence Estimation), o a distanza euclidea minima equivalente, per i codici convoluzionali [167]. Esplicitamente, l'algoritmo di Viterbi (VA) mira a trovare la sequenza di errore più probabile con una complessità di decodifica accettabile. Il VA è un algoritmo MLSE, dunque il BER risultante del sistema è circa uguale al BER minimo possibile, dato che solo un decodificatore complesso a massima verosimiglianza (ML, Maximum Likelihood) è in grado di valutare tutte le possibili sequenze valide. Per aggirare l'elevata complessità di quest'ultimo decodificatore, Bahl *et al.* hanno proposto l'algoritmo minimo di decodifica BER nel 1974, che è stato nominato algoritmo MAP (Maximum A Posteriori probability). È anche conosciuto come BCJR dai nomi dei suoi

inventori Bahl, Cocke, Jelinek e Raviv [18].

Proseguendo nell'ambito dei codici a blocco, Prange studiò i codici ciclici nel 1957 [131]. Un codice a blocco lineare si dice ciclico se lo spostamento ciclico dei simboli delle parole di codice genera parole di codice anch'esse legittime; pertanto le procedure di codifica e decodifica associate possono essere implementate in modo efficiente utilizzando registri a scorrimento. Ispirati da questi sviluppi, Hocquenghem [78], così come Bose e Ray-Chaudhuri [28], [27], hanno scoperto, indipendentemente, i codici della famiglia Bose-Chaudhuri-Hocquenghem (BCH), rispettivamente nel 1959 e nel 1960. In particolare, i codici BCH costituiscono una famiglia di codici a blocco ciclici in grado di correggere errori multipli; infatti codificano $k \geq (n - rt)$ bit di informazione in $n = (2^r - 1)$ simboli codificati, in modo che le parole di codice risultanti siano caratterizzate dalla massima distanza minima di Hamming possibile.

Nel 1960, Reed e Solomon elaborarono una versione M -aria dei codici BCH, denominata codici Reed-Solomon (RS), essi infatti operano su un alfabeto di simboli non binari costituito da $M = 2^m$ elementi, con $m > 1$ [132]. I parametri dei codici RS sono $n = 2^m - 1$ e $r = n - k = 2t$. Il legame tra r e t , in particolare, è estremamente favorevole, nel senso che corrisponde al migliore utilizzo possibile dei caratteri di ridondanza: per un dato r , non esistono infatti codici in grado di correggere un numero di errori $t > r/2$. I codici per i quali $t = r/2$ (ed i codici RS appartengono a questa categoria), pertanto rappresentano il miglior risultato conseguibile, sotto questo punto di vista, e prendono il nome di codici MDS (Maximum Distance Separable) [129].

L'anno successivo, nel 1961, Gorenstein e Zierler svilupparono lo schema di decodifica Peterson-Gorenstein-Zierler (PGZ) per i codici RS/BCH non binari. In seguito, Berlekamp e Massey hanno sviluppato l'algoritmo Berlekamp-Massey, ampiamente adottato per la decodifica dei codici ciclici RS/BCH [24], [54], [104], [105]. I codici RS hanno trovato diverse applicazioni pratiche a causa della loro capacità intrinseca di correggere sia errori casuali sia burst error, cioè più errori ravvicinati. In particolare, i codici RS sono ampiamente utilizzati per l'archiviazione su nastro magnetico e su disco, ambiti che sono suscettibili di burst [129]. Un'altra importante pietra miliare nella codifica è stata raggiunta con lo sviluppo dei cosiddetti codici non binari Redundant Residue Number System (RRNS) [170], [157] che sono anch'essi codici MDS e quindi presentano proprietà simili ai codici RS. A partire dal 1980, i codici di correzione degli errori sono stati introdotti con successo in vari sistemi di comunicazione, come quelli spaziali, satellitari e mobili. Tuttavia, le tecniche di correzione degli errori e le tecniche di modulazione sono state trattate in modo indipendente, dunque la ridondanza dei codici (richiesta per la correzione degli errori) ha inevitabilmente esteso il requisito della larghezza di banda, benché la dimensione della costellazione del segnale (data dalla modulazione numerica) fosse già stata fissata. Al fine di aggirare questo svantaggio della codifica, Ungerboeck ha inventato uno schema combinato di codifica e modulazione, basato su una rappresentazione a trellis (diagramma a traliccio) ed efficiente in termini di larghezza

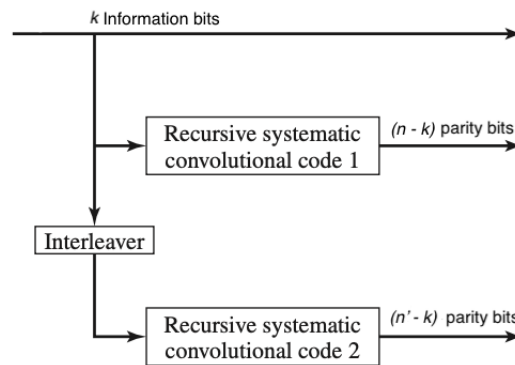


Figura 2.4: Schema a blocchi che rappresenta un generico codice turbo [129].

di banda, chiamato per questo Modulazione codificata a traliccio (TCM, Trellis-Coded Modulation) [163], [164], [165]. Infatti il TCM permette di assorbire i simboli di codifica ridondanti espandendo la dimensione della costellazione per ospitare più bit/simboli e quindi riesce a mantenere la larghezza di banda fissa. Questo schema di codifica e modulazione congiunto offre vantaggi prestazionali interessanti rispetto ai codici convoluzionali, pur incorrendo in una complessità di decodifica simile. Per esempio si utilizza per trasmettere informazioni con elevata efficienza su canali a banda limitata come le linee telefoniche.

Nel 1992 è stato concepito un altro schema di modulazione codificata chiamato Bit-Interleaved Coded Modulation (BICM). Veniva utilizzato per la trasmissione di dati su canali con fading (fading channels), ed utilizzava la tecnica dell'interleaving sui bit, in combinazione con il mapping di Gray [181], [34]. In particolare, in questo schema di codifica e modulazione congiunta, gli interleaver sono utilizzati all'uscita di un codice convoluzionale, con lo scopo di aumentare il guadagno risultante, sfruttando il fading. L'interleaver dispone i dati di un generico simbolo M -ario in maniera tale da migliorare le prestazioni del codice in termini di rilevazione e di correzione degli errori in ricezione, soprattutto nel caso di errori a burst.

Tuttavia, il BICM non supera le prestazioni della tecnica di codifica TCM sui canali AWGN, poiché presenta una distanza minima euclidea ridotta [12].

Sebbene la teoria della codifica classica avesse quasi cinque decenni di storia, solo nel 1993, grazie al contributo di Berrou *et al.*, venne elaborata una famiglia di codici (i codici turbo) che si basava sull'idea di operare arbitrariamente vicino al limite di Shannon, nozione che fino a quel momento era rimasta lontana dalla realizzazione [25], [26]. In particolare, il generico codice turbo si basa su una concatenazione parallela di codici Convoluzionali Ricorsivi e Sistemati (RSC, cioè Recursive Systematic Convolutional) con un interleaver tra di essi.

Approfondimento: Codici sistemati I codici caratterizzati da simboli in

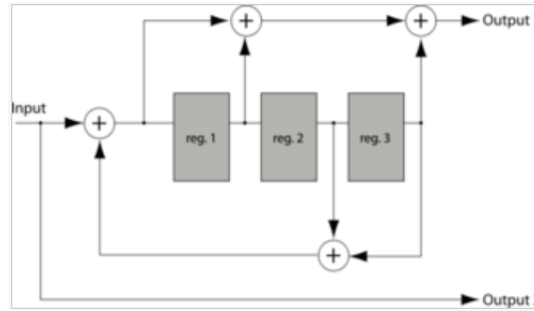


Figura 2.5: Esempio di codificatore ricorsivo [1].

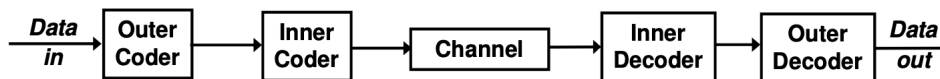


Figura 2.6: Schema a blocchi che realizza la concatenazione seriale.

output che includono i dati di input sono chiamati sistemati. Nel caso in cui essi sono anche ricorsivi, cioè usano un codificatore ricorsivo come quello di Fig. 2.5, si dicono RSC (anche conosciuti come codici pseudo-sistemati). Un esempio di codifica che usa i codici RSC sono i codici turbo.

Approfondimento: Concatenazione seriale e parallela Si parla di concatenazione seriale (o in serie) quando vengono realizzate due codifiche, l'una di seguito all'altra. Questa tecnica consente la realizzazione di codici con elevate capacità di correzione degli errori ed una complessità non eccessiva. È riportato un esempio di schema a blocchi che realizza tale modello in Fig. 2.6. I codici concatenati serialmente spesso vengono utilizzati congiuntamente ad un interleaver, in particolare esso viene posto tra il blocco Inner Coder ed il blocco Outer Coder di Fig. 2.6. Di conseguenza, nel caso in cui fosse necessario, il deinterleaver viene inserito tra l'Inner Decoder e l'Outer Decoder. Nello schema di concatenazione parallela, mostrato nella Fig. 2.7, la stessa sequenza di informazione viene immessa in due encoder sistemati distinti, in particolare è presente un interleaver prima del secondo codificatore. Dunque, solo i simboli di parità in uscita dal secondo codificatore vengono aggiunti in coda all'uscita del primo codificatore [23]. Un esempio di concatenazione parallela con interleaver è quella relativa al codice turbo mostrato in Fig. 2.4.

Tornando ai codici turbo, si osserva che il decodificatore è di tipo Soft-In Soft-Out (SISO) [18] ed utilizza la decodifica soft in modalità iterativa. "Soft-in" indica che i dati in entrata possono assumere valori diversi da 0 o 1, al fine di indicare l'affidabilità della trasmissione, allo stesso modo "Soft-out" si riferisce al fatto che ogni bit nell'uscita decodificata assume anche un valore che indica l'affidabilità. Tale decodifica soft si basa sulla distanza euclidea tra le sequenze di simboli ricevuti che compongono la costellazione e i corrispondenti vettori di proiezione, dunque si fonda sull'algoritmo

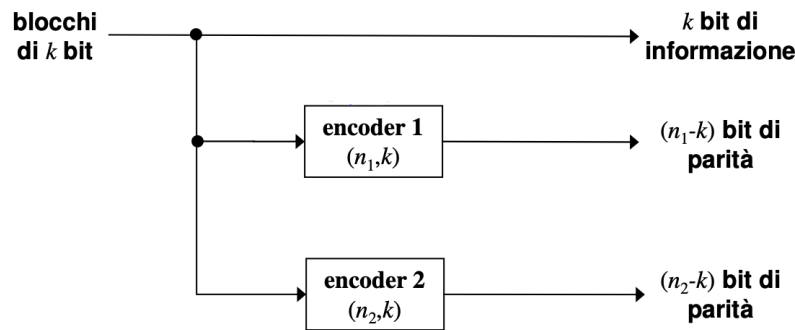


Figura 2.7: Schema a blocchi che realizza la concatenazione parallela.

MAP [18]. È opportuno ricordare in questa sede che l'algoritmo MAP supera di poco il VA in termini di BER ottenibile per i codici convoluzionali decodificati in modo non iterativo, richiedendo, allo stesso tempo, una complessità significativamente più elevata. Di conseguenza, la decodifica MAP è raramente utilizzata per decodificare i codici convoluzionali, almeno fino a quando sono stati inventati i codici turbo. Dato che i decodificatori turbo richiedono tecniche di decodifica soft bit-by-bit, essi richiedono una decodifica MAP complessa. Fortunatamente, la complessità dei turbo decoder può essere ridotta ricorrendo ai meno complessi decodificatori SISO, che utilizzano, ad esempio, l'algoritmo di Viterbi Soft-Output (SOVA) [71], l'algoritmo Log-MAP [89] e l'algoritmo Max-Log-MAP [137].

La rivoluzione dei codici turbo di Berrou ha innescato significativi sforzi di ricerca diretti verso la progettazione di codici iterativi, simili ad essi. In particolare, il 1995 ha visto la rinascita dei codici Low-Density Parity-Check (LDPC) [99], [100]. In effetti tali codici erano già stati proposti da Gallager nel 1962 [61] insieme a due modelli di decodificatore associati. Il primo era semplice, ma non garantiva prestazioni ottimali; al contrario il secondo era molto più complesso. Tale complessità fu considerata enorme per l'epoca e dunque i codici LDPC furono abbandonati per i decenni a venire. Questi codici sono noti per operare arbitrariamente vicino al limite di Shannon con parole di codice sufficientemente lunghe. Un codice LDPC binario è caratterizzato da una matrice di parità contenente in gran parte 0 e solo un piccolo numero di 1.

Esplicitamente un codice LDPC binario regolare è caratterizzato da una terna di parametri (n, λ, ρ) , dove n indica il numero di colonne della matrice di parità, λ rappresenta il (piccolo) numero fissato di 1 presenti in ogni colonna e ρ il (piccolo) numero fissato di 1 per ogni riga [61].

In particolare, un codice binario LDPC *regolare*, con colonna di peso λ e riga di peso ρ , consiste in un codice a blocco $C(n, k)$ (dove, come sempre, k sono i bit di informazione) caratterizzato da una matrice di parità \mathbf{H} , di dimensioni $r \times n$. In particolare, per *peso di Hamming* di un vettore, riga o colonna, si intende il numero di elementi non nulli, in questo caso si tratta degli 1. Dunque non sorprende che effettivamente in una matrice di soli 1 e 0, il valore della norma di un certo vettore riga o colonna eguagli il peso

1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1

Figura 2.8: Esempio di una matrice \mathbf{H} , con $n = 20$, $j = 3$ e $p = 4$ [61].

della stessa [61]. Nella PCM \mathbf{H} le righe h_i e le colonne h'_j soddisfano le seguenti relazioni:

$$\begin{aligned} \|h_i\|_H &= \rho, \quad \forall i = 1, \dots, r \\ \|h'_j\|_H &= \lambda, \quad \forall j = 1, \dots, n, \end{aligned} \quad (2.15)$$

cioè la norma di ogni riga vale sempre ρ , mentre la norma di ogni colonna è pari a λ [23]. Un esempio di matrice per un codice LDPC è riportato in Fig. 2.8.

Alcune proprietà generali di questi codici possono essere dedotte dalla definizione precedente, come un limite inferiore sulla distanza minima; in particolare si osserva che in un codice binario regolare LDPC, con peso di Hamming della colonna pari a λ , la distanza minima soddisfa la seguente relazione [23]:

$$d_{min} \geq \lambda + 1. \quad (2.16)$$

Nel corso degli anni sono state proposte molte varianti, ad esempio i codici LDPC irregolari [97], [135], i codici convoluzionali LDPC [57], i codici LDPC basati su protografi [161] e i codici LDPC spatially coupled (SC-LDPC, Spatially Coupled - Low Density Parity Check codes) [90].

La rivoluzione dei codici turbo ha portato anche ad altri schemi di codifica iterativi, che includono ad esempio i codici Turbo BCH [72], i codici Turbo Hamming [115], i codici BICM con decodifica iterativa (BICM-ID, Bit Interleaved Coded Modulation-Iterative Decoding) [94], la Turbo Trellis Coded Modulation (TTCM) [138], i codici turbo punturati [4] e i sistemi di codifica concatenata assistita (URC, Unity Rate Code) [48].

L'invenzione dell'EXtrinsic Information Transfer chart (EXIT charts) da parte di Ten Brink nel 2001 segna un'altra importante pietra miliare nel regno degli schemi concatenati che utilizzano la decodifica iterativa [160], [87]. Più specificamente, i diagrammi EXIT costituiscono uno strumento semi-analitico che aiuta la progettazione di sistemi iterativi con capacità prossime al limite imposto da Shannon [77], [51]. Quantitativamente, questi possono operare entro 1 dB dal limite di Shannon, si vedano ad esempio: i codici irregolari convoluzionali assistiti da schemi concatenati (IRCC, Irregular Convolutional Code) [162], il TTCM [113] e il BICM-ID [159].

Con l'aiuto di intensi sforzi di ricerca, la codifica turbo è stata commercializzata con successo in pochi anni ed è stata incorporata in vari sistemi standardizzati, come i sistemi di comunicazione mobile e i sistemi di trasmissione video [77]. In particolare, la codifica turbo è stata incorporata negli standard mobile 3G UMTS e 4G LTE [56], [55]. D'altra parte, a causa dell'elevata latenza associata ai codici turbo, essi sono stati rimpiazzati dai codici LDPC nei sistemi 5G NR. Di conseguenza, per superare questa difficoltà, è stato concepito da Maunder [107] un decodificatore per codici turbo completamente parallelo (FPTD, Fully-Parallel Turbo Decoder), che ne riduce significativamente la latenza associata. Nel corso degli anni, lo schema di codifica LDPC ha dimostrato di essere un notevole concorrente dei codici turbo, infatti è stato adottato anche da vari standard, per esempio WiMax IEE 802.16, che consente l'accesso di tipo wireless a reti di telecomunicazioni a banda larga, e DVB-S2 (Digital Video Broadcasting - Satellite second generation) che riguarda la trasmissione satellitare.

Nel 2009, i codici polari, concepiti da Arikan [8] hanno scatenato un'altra ondata di fermento all'interno della comunità scientifica nell'ambito della codifica, poiché costituiscono la prima classe di codici, per la codifica di canale, che raggiunge la capacità di canali senza memoria simmetrici, pur imponendo una relativamente modesta complessità di codifica e decodifica. I codici polari si basano su un kernel semplice, tale che i canali fisici siano polarizzati in canali virtuali, che risultano essere perfettamente privi di rumore o completamente randomici (e quindi a massima incertezza), a condizione che la lunghezza del blocco sia sufficientemente elevata [8]. Per lunghezze del blocco realistiche, i canali sono polarizzati in una serie di altri canali virtuali ad alta e a bassa affidabilità. Infine, i bit informativi vengono inviati attraverso i canali ad alta affidabilità, mentre i bit di ridondanza, chiamati anche "frozen bit", vengono trasmessi attraverso i canali a bassa affidabilità. Se le lunghezze dei blocchi di codice sono sufficientemente elevate, l'insieme dei canali virtuali ad alta affidabilità è equivalente alla capacità del canale raggiungibile. Al ricevitore, il decodificatore polare invoca un low-complexity successive cancellation decoding algorithm, che elabora in serie i bit ricevuti [117]. Pur avendo una bassa complessità di codifica e decodifica, i codici polari sono in grado di superare le performance dei codici LTE turbo e dei codici WiMax LDPC standardizzati, per lunghezze di blocco moderate [117]. Inoltre, il rate di codifica dei codici polari può essere variato in maniera quasi continua cambiando il numero dei frozen bit, rendendoli quindi ideali per tutti i possibili valori che può assumere il rate. Tuttavia, una limitazione importante dei codici polari è l'alta latenza (ritardo) associata al decodificatore, poiché elabora in sequenza l'informazione ricevuta. Nonostante ciò, questi codici hanno già trovato collocazione nel sistema 5G per le nuove comunicazioni mobile a banda larga, dove i codici polari e i codici LDPC sono stati scelti, rispettivamente, per i canali di controllo e per i canali dati.

Per concludere, i codici turbo classici, gli LDPC e i codici polari permettono di operare arbitrariamente vicino al limite di Shannon. Ad esempio, il codice turbo con rate $1/2$ di [26] opera entro 0.7 dB dal limite di Shannon

ad una lunghezza di blocco di 65 536 bit, mentre il codice irregolare LDPC di [135] con bit-rate pari ad $1/2$ supera le prestazioni dei codici turbo comparabili e funziona a soli 0.13 dB di distanza dalla capacità Shannon ad una lunghezza blocco di 106 bit.

Dunque, l'ambizione di raggiungere il limite di Shannon ha portato, nel corso degli anni, ad introdurre codici caratterizzati da lunghi ritardi di decodifica, che hanno a loro volta motivato la ricerca sulle architetture di decodifica parallele, ad esempio sui decodificatori LDPC completamente paralleli e sui FPTD, per i codici turbo [107], [121]. Alla luce di queste osservazioni si nota che la comunità scientifica ha inizialmente progettato codici praticamente irrealizzabili, nello spirito di raggiungere il limite di Shannon, e poi si è data un nuovo obiettivo di ricerca: ridurre i ritardi di decodifica. Pertanto, oggi la sfida ancora aperta dei ricercatori è quella di progettare codici che riescano a soddisfare i compromessi progettuali (design trade-off) desiderati, tra i vari parametri elencati nella Sezione 2.1.1. In modo esplicito, generalmente si ha bisogno di un codice che massimizzi il rate di codifica, nelle condizioni di canale date, e che riduca al minimo: il BER (Bit Error Rate) raggiungibile, la larghezza di banda del sistema, il ritardo e la complessità di implementazione. È inoltre auspicabile che il codice sia compatibile con tutti i possibili valori che può assumere il rate e che quindi sia in grado di operare in molteplici casi d'uso ed in diverse condizioni del canale.

2.2 Teoria della codifica quantistica

2.2.1 *Obiettivi di progetto*

Con circa sette decenni di ricca storia, la teoria della codifica classica è già abbastanza matura. Al contrario, la teoria della codifica quantistica è ancora nella fase del primo sviluppo, poiché l'implementazione della tecnologia quantistica non è stata commercializzata. Ciononostante i ricercatori hanno lavorato intensamente per progettare versioni quantistiche dei codici classici già esistenti. Dualmente alla teoria della codifica classica, i QECCs (Quantum Error Correction Codes) sono progettati per raggiungere la massima capacità del canale quantistico o, più precisamente, per rispettare l'hashing bound (limite di hashing) [96], [146], [45]. In particolare, l'hashing bound è solo un limite inferiore, poiché la capacità effettiva di un canale quantistico odierno può essere maggiore, a causa della natura *degenere* dei codici quantistici [47], [150].

La nozione di degenerazione implica che modelli di errore diversi possano produrre lo stesso stato quantistico degradato. Per esempio, si consideri lo stato di input $|\psi\rangle = |00\rangle + |11\rangle$, che durante la trasmissione, può essere modificato dall'errore \mathbf{IZ} o \mathbf{ZI} , indotto dal canale di Pauli. Si può osservare infatti che entrambi questi modelli di errore producono la stessa uscita dal canale, cioè $(|00\rangle - |11\rangle)$. Di conseguenza, i modelli di errore \mathbf{IZ} e \mathbf{ZI} sono

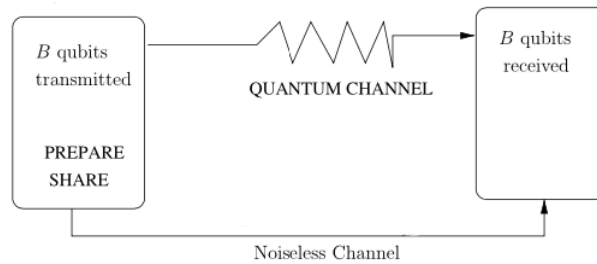


Figura 2.9: Schematico di un possibile sistema di comunicazione quantistico a B qubit che utilizza la codifica EA.

classificati come degeneri, come poi sarà ulteriormente discusso nel Capitolo 4. Allo stesso modo, il modello di errore \mathbf{ZZ} lascia lo stato $|\psi\rangle$ intatto, analogamente allo scenario privo di errori (\mathbf{II}); quindi \mathbf{ZZ} e \mathbf{II} (naturalmente, l'errore caratterizzato dalla doppia moltiplicazione per la matrice identità \mathbf{I} non provoca cambiamenti nello stato quantistico) sono anch'essi classificabili come errori degeneri.

Continuando nel confronto tra il dominio classico e quello quantistico, si osserva che per il primo vale il limite di Shannon dell'eq.(2.1), che determina la capacità del canale di comunicazione oltre la quale non si può operare con affidabilità (infatti vale: $R < C$), mentre per il secondo esiste l'hashing bound (o limite di hashing), il quale è completamente definito dalla probabilità di depolarizzazione p del canale (a tal proposito, si guardi la Sottosezione 1.3.3), come segue [21], [176]:

$$C_Q(p) = 1 - H_2(p) - p \log_2(3), \quad (2.17)$$

dove $H_2(p)$ indica la funzione di entropia binaria, calcolata per p .

In particolare, un codice quantistico casuale \mathcal{C} può presentare un Quantum Bit Error Ratio (QBER) arbitrariamente basso ad una probabilità di depolarizzazione pari a p , se il suo rate di codifica non supera $C_Q(p)$ di eq.(2.17), cioè l'hashing bound, e la parola di codice ha una lunghezza sufficientemente elevata. Il QBER è il rapporto tra i qubit non ricevuti correttamente e i qubit trasmessi, cioè si tratta della probabilità di errore sul qubit.

Il limite fornito dall'eq.(2.17) è valido solo per i codici quantistici non "assistiti". Esiste infatti anche una famiglia di codici quantistici "assistiti", detti "Entanglement-Assisted" (EA), che non ha un corrispettivo nel dominio classico [31], [32], [33], [33]. Differentemente dai codici quantistici non assistiti, questi si basano su qubit entangled (quindi intrecciati, nel senso della Sottosezione 1.1.2) e pre-shared, che aumentano naturalmente la capacità raggiungibile dal canale quantistico (questi concetti verranno approfonditi ulteriormente nei prossimi capitoli). È presente un esempio di implementazione nella Fig. 2.9. Dato che gli entangled qubit sono pre-condivisi, dal trasmettitore con il ricevitore autorizzato, su un canale non rumoroso, l'hashing bound EA associato è espresso dalla seguente relazione [176], [93]:

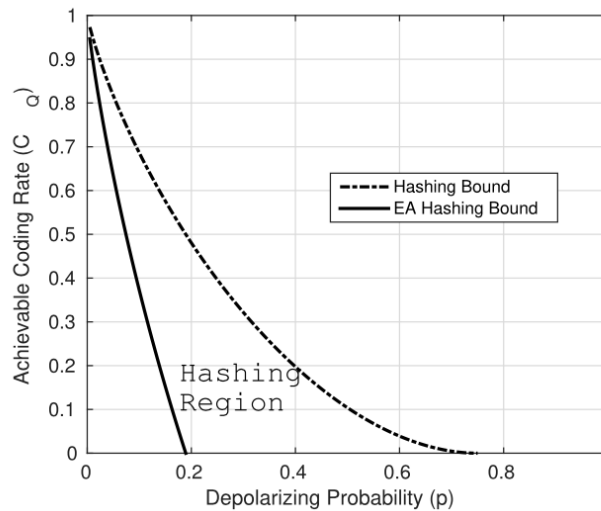


Figura 2.10: Sono rappresentati gli hashing bound per i codici quantistici non assistiti ($c = 0$) e massimamente entangled ($c = n - k$), espressi dalle eq.(2.17) e (2.19) [12].

$$C_Q(p) = 1 - H_2(p) - p \log_2(3) + E, \quad (2.18)$$

dove E denota il rate di consumo di entanglement (entanglement consumption rate), che è equivalente a $E = \frac{c}{n}$ per un codice con k qubit di informazione, n qubit codificati, e c qubit già condivisi, dove $0 \leq c \leq (n - k)$. Esplicitamente, con la dicitura "qubit codificati" si indicano i simboli nel dominio quantistico. In particolare, quando $c = 0$, l'eq.(2.18) si riduce al limite di hashing non assistito dell'eq.(2.17). Al contrario, quando c ha il valore massimo, pari a $(n - k)$, si ottengono i codici quantistici maximally-entangled e, di conseguenza, l'hashing bound maximally-entangled associato è [176], [93]:

$$C_Q(p) = 1 - \frac{H_2(p) - p \log_2(3)}{2}. \quad (2.19)$$

Quindi, come mostrato in Fig. 2.10, un codice quantistico EA può operare ovunque nella zona chiusa, denominata regione di hashing e delimitata dalle equazioni (2.17) e (2.19), che quantifica la capacità per $0 < c < (n - k)$.

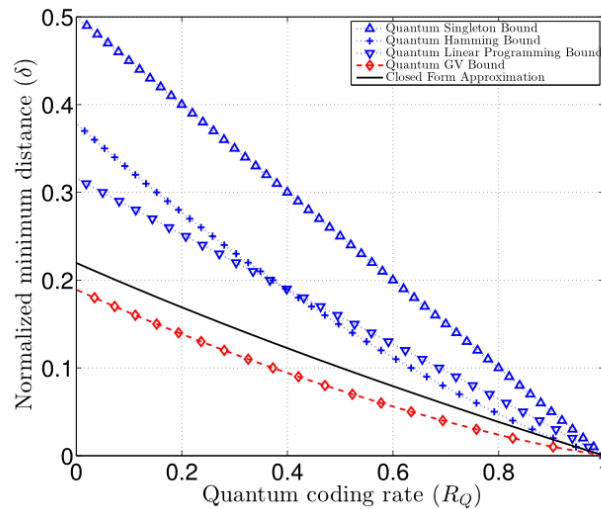
Inoltre, dualmente a quanto riportato nella Sezione 2.1.1, per la teoria della codifica classica, i parametri coinvolti nella progettazione dei QECCs (Quantum Error Correction Codes) sono illustrati di seguito:

- Coding Rate, cioè il tasso di codifica;
- QBER (Qubit Error Rate, cioè, come già detto, la probabilità di errore sul qubit);
- Tasso di consumo di entanglement (Entanglement Consumption Rate);
- Ritardo di trasmissione (transmission delay);
- Complessità di implementazione;
- Caratteristiche del canale.

Come nel caso della codifica classica, tali parametri, che influenzano la progettazione dei codici quantistici, sono in conflitto tra loro.

Tabella 2.2: Limiti del rate-versus-minimum-distance per i codici quantistici [39].

Quantum Coding Bound	Finite-Length	Asymptotic
Singleton [88]	$\frac{k}{n} \leq 1 - 2\left(\frac{d_{min}-1}{n}\right)$	$\frac{k}{n} \leq 1 - 2\left(\frac{d_{min}}{n}\right)$
Hamming [50]	$\frac{k}{n} \leq 1 - \frac{1}{n} \log_2\left(\sum_{j=0}^{\lfloor \frac{d_{min}-1}{2} \rfloor} \binom{n}{j} 3^j\right)$	$\frac{k}{n} \leq 1 - \left(\frac{d_{min}}{2n}\right) \log_2(3) - H_2\left(\frac{d_{min}}{2n}\right)$
Linear Programming [9]		$\frac{k}{n} \leq H_2(\tau) + \tau \log_2(3) - 1$
Gilbert-Varshamov (GV) [50]	$\frac{k}{n} \geq 1 - \frac{1}{n} \log_2\left(\sum_{j=0}^{d_{min}-1} \binom{n}{j} 3^j\right)$	$\frac{k}{n} \geq 1 - \left(\frac{d_{min}}{n}\right) \log_2(3) - H_2\left(\frac{d_{min}}{n}\right)$

**Figura 2.11:** Sono rappresentati i limiti asintotici dell'approssimazione in forma chiusa dell'eq.(2.20) del rate $R_Q = k/n$ in funzione della distanza minima normalizzata $\delta = d_{min}/n$ [12].

Dualmente alla teoria della codifica classica, i cui limiti sono riportati nella Tabella 2.1, la Tabella 2.2 riunisce i limiti di codifica nel dominio quantistico, che caratterizzano il compromesso tra rate e distanza minima (rate-versus-minimum-distance trade-off). Analogamente al caso classico, il limite quantistico di Singleton funge da limite superiore lasco, mentre il limite quantistico di Hamming come limite superiore più aderente e invece, quello di Gilbert-Varshamov (GV), come limite inferiore più aderente. Inoltre, Ashikhmin e Litsyn hanno esteso il classico approccio di programmazione lineare ai codici quantistici usando le identità di MacWilliams per migliorare il limite quantistico di Hamming [9]. Tuttavia, esisteva un ampio divario tra i limiti di codifica superiore e inferiore, fino a quando Chandra *et al.*, dualmente a quanto era stato fatto nel caso della codifica classica (si osservi a tal proposito la Tabella 2.1), concepirono un'espressione in forma chiusa per caratterizzare il trade-off rate-versus-minimum distance per i codici quantistici, cioè [39]:

$$R_Q(\delta) = \frac{32}{9}\delta^2 - \frac{16}{3}\delta + 1, \quad \forall 0 \leq \delta \leq 0.2197 \quad (2.20)$$

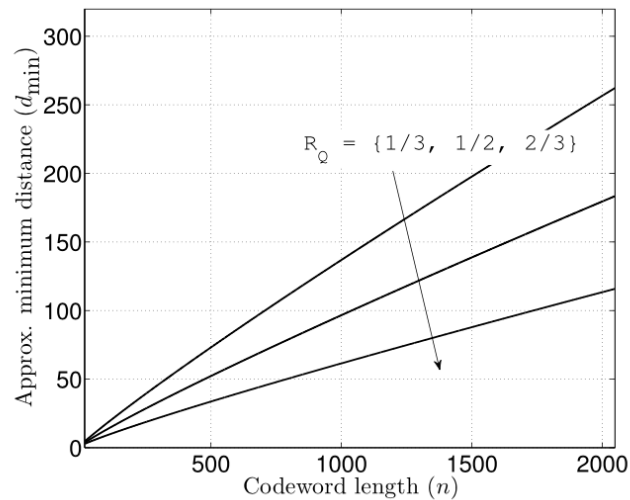


Figura 2.12: La crescita della distanza minima raggiungibile con l'aumento della lunghezza della parola di codice, sulla base dell'espressione dell'eq.(2.20) [12].

Come mostrato nel grafico di Fig. 2.11, la relazione di cui sopra, che si basa su una semplice funzione quadratica invertibile, soddisfa tutti i limiti di codifica noti. Come si può facilmente osservare, i limiti superiori sono tracciati in blu, mentre il limite inferiore è presentato in rosso. L'approssimazione in forma chiusa di eq.(2.20), inoltre, offre il vantaggio della semplicità e ha la seguente funzione inversa [39]:

$$\delta(R_Q) = \frac{3(\sqrt{2} - \sqrt{R_Q + 1})}{4\sqrt{2}}. \quad (2.21)$$

Tale relazione in forma chiusa suggerisce che è possibile creare un modello di codice la cui distanza minima d_{min} tende, crescendo linearmente con la lunghezza della parola di codice, al limite asintotico (si guardi Fig. 2.12) [39]. Dal momento che ad un certo rate di codifica quantistica R_Q , corrisponderà un unico valore costante δ , si può osservare dalla Fig. 2.12, che la distanza minima d_{min} aumenta quasi linearmente con la lunghezza della parola di codice; per questo motivo tale distanza è detta "distanza minima illimitata". Di conseguenza, è auspicabile progettare strutture di codice aventi proprio la distanza minima illimitata.

2.2.2 Codici per la correzione degli errori

Prima di calarci nel vivo della trattazione sui codici per la correzione degli errori nel dominio quantistico, dato che storicamente questi sono stati progettati a partire dai rispettivi modelli di codice mutuati dal dominio classico, è opportuno richiamare, in prima istanza, l'importante categoria dei *codici a ripetizione* classici, appartenente alla famiglia dei codici a blocco [129].

Approfondimento: Codici a Ripetizione Il codice a ripetizione con rate di codifica $R = 1/3$ è il più semplice (concettualmente) codice di correzione per un singolo errore, nel paradigma di codifica classico, e si basa sulla clonazione, cioè sulla ripetizione, dello stesso blocco di bit di informazione. In particolare, si tratta di un codice a blocco $(3, 1)$, quindi con $k = 1$ simboli di informazione, $n = 3$ simboli totali e dunque $r = n - k = 3 - 1 = 2$ simboli di ridondanza. Pertanto l'unico bit di informazione verrà "clonato" due volte e dunque esistono solo $2^k = 2$ possibili parole di codice: $X_1 = (000)$ e $X_2 = (111)$ e quindi può essere corretto $t = \lfloor \frac{n-1}{2} \rfloor = 1$ errore [129].

Sfortunatamente però, come dimostrato dal teorema di non clonazione, i qubit non possono essere clonati. Questo risultato non sorprende dato che tali elementi quantistici sono, per loro natura, a meno di una osservazione, caratterizzati dal generico stato quantistico $|\psi\rangle$ (superposition). Quindi sarebbe impossibile creare un codice a ripetizione di n qubit clonati, per esempio come quello di cui sopra, poiché lo stato di ogni qubit si conoscerebbe soltanto dopo un processo di misurazione (tale tematica sarà affrontata nel prossimo capitolo).

Per questo si credeva che i QECCs fossero irrealizzabili, fino a quando nel 1995 Shor riuscì ad elaborare il primo codice quantistico [145]. Il codice pionieristico di Shor è caratterizzato da un rate pari ad $1/9$ ed è in grado di correggere un singolo errore di tipo bit-flip, phase-flip oppure bit-and-phase-flip, nella prospettiva di usare il modello di canale di Pauli [145]. Motivati da questa svolta, Calderbank e Shor [36], e poi anche Steane [153], [152], hanno concepito indipendentemente un modello generalizzato per la strutturazione di codici quantistici, a partire dai codici classici lineari e binari, che costituisce la popolare famiglia dei codici CSS (Calderbank-Shor-Steane). Di fatto, la costruzione dei codici CSS si basa su una coppia di codici classici a blocco lineari binari C_1 e C_2 che soddisfano la relazione $C_1 \subset C_2$.

Successivamente, è stata introdotta una classe speciale di codici CSS, chiamata codici CSS dual-containing, che è stata derivata dai relativi codici binari dual-containing. Questi codici sono caratterizzati da $C_2 = C_1^\perp$, dove C_1^\perp è il codice duale di C_1 .

Seguendo questi principi, Steane [152] ha costruito un codice di correzione del singolo errore con rate pari ad $1/7$, partendo dal codice classico di Hamming [11], [44], [70], migliorando dunque quanto fatto da Shor. Con l'idea di migliorare ulteriormente il rate di codifica, Bennett *et al.* [21] e Laflamme *et al.* [92] hanno progettato, indipendentemente, un codice quantistico per la correzione di un singolo errore con rate ottimo pari ad $1/5$, considerando parole di codice con lunghezza più piccola possibile. Infatti, questo codice per la correzione degli errori che è in grado di correggere $t = 1$ errore con $k = 1$ qubit di informazione adopera, in totale, $n = 5$ qubit codificati (che è già un miglioramento rispetto al codice di Shor). Non esiste un codice per la correzione degli errori, in grado di correggere un errore, che coinvolge un numero di qubit minore di $n = 5$, in questo senso tale codice è considerato ottimo; cioè date queste prestazioni non può essere più piccolo di così.

In generale, un codice quantistico in grado di correggere t errori, si dice che

è caratterizzato da una distanza $d = 2t + 1$, poiché, data una certa parola di codice, sarebbe necessario andare a modificare $2t + 1$ singoli qubit, per ottenerne un'altra [88].

La costruzione dei codici CSS [57], [153], [36], [152], non sfrutta in modo efficiente i qubit ridondanti, poiché gli errori di bit-flip e phase-flip vengono corretti indipendentemente, concatenando una coppia di codici binari classici. Ai fini della progettazione di un codice ottimale, con lunghezza minima della parola di codice, come nel caso del codice con rate $1/5$ di cui sopra ([21], [92]), è importante correggere congiuntamente gli errori di inversione di bit (bit-flip) e di sfasamento (phase-flip). Così, durante la progettazione di tali codici ottimizzati, Gottesman elaborò, nel corso del suo Ph.D. ([65]), la teoria dei codici quantistici stabilizzatori (QSC, Quantum Stabilizer Codes) [64]. In particolare, Gottesman presentò un formalismo più generale, chiamato formalismo stabilizzatore (stabilizer formalism), in grado di facilitare la progettazione dei codici quantistici, a partire dai codici classici binari e quaternari. Rispetto ai CSS, tale formalismo impone un vincolo più rilassato, generalmente chiamato criterio del prodotto simplettico, sui codici classici sottostanti; quindi, i QECCs risultanti possono avere una struttura CSS o non CSS (chiamata anche struttura senza restrizioni). In altre parole, il criterio del prodotto simplettico (verrà trattato in maniera esauriente nel Capitolo 5) è il vincolo imposto alla PCM del codice (o codici) classico costituente, che assicura che il codice quantistico risultante sia un valido codice stabilizzatore. Inoltre, mentre i codici di tipo CSS correggono in modo indipendente gli errori di tipo bit-flip e phase-flip, i codici non-CSS correggono insieme (nel senso di "congiuntamente") gli errori di inversione del bit e di inversione della fase.

L'avvento del formalismo stabilizzatore ha avviato una grande rivoluzione nella storia della codifica quantistica, conducendo allo sviluppo delle seguenti famiglie di codici:

- i codici quantistici di Bose-Chaudhuri-Hocquenghem (QBCH, Quantum Bose Chaudhuri Hocquenghem) [154], [68], [35], [66], [155], [179];
- i codici torici (toric codes) [85], [86], [60],
- i codici quantistici di Reed-Muller (Quantum Reed Muller codes) [156];
- i codici quantistici di Reed-Solomon (QRS, Quantum Reed Solomon codes) [69];
- i codici quantistici Low-Density Parity-Check (QLDPC) [124], [98], [37], [38];
- i codici quantistici convoluzionali (QCC, Quantum Convolutional Codes) [119], [120], [59], [58];
- i codici quantistici turbo (QTC, Quantum Turbo Codes) [127], [126];
- i codici quantistici irregolari convoluzionali (QIRCC, Quantum Irregular Convolutional Codes) [11];
- i codici quantistici a rate unitario (QURC, Quantum Unity Rate Codes) [16];
- i codici quantistici polari di tipo CSS [133].

La ricerca in ambito Quantum negli ultimi tre decenni ha quindi investito, come mostrato sopra, nella progettazione delle controparti quantistiche

delle famiglie di codice classiche già esistenti. Fatta eccezione per i codici paralleli concatenati e per i codici a schema combinato di codifica e modulazione, quasi tutte le altre principali famiglie di codice del dominio classico hanno una controparte quantistica. Tra questi, i codici a blocco corti sono particolarmente importanti dal punto di vista dell'implementazione, poiché, essendo ancora la tecnologia quantistica molto "giovane", il fenomeno della decoerenza (Sezione 1.3) impedisce (ancora) un'implementazione efficiente di codici eccessivamente lunghi. Infatti, più una parola di codice è lunga e più è probabile che uno dei qubit che lo compone possa essere misurato dall'ambiente. Tuttavia, il desiderio di avvicinarsi al limite di hashing di Fig. 2.10 ha motivato i ricercatori a progettare codici QLDPC [98], [37], [38] e QTC [127], [126], che sfruttassero la tecnica della decodifica iterativa, dualmente alle loro controparti classiche. In particolare, nel dominio quantistico la natura sparsa della matrice di parità dei codici LDPC è particolarmente funzionale per raggiungere la decodifica fault-tolerant, poiché i qubit interagiscono solo con un numero limitato di altri qubit, essendoci fattivamente pochi '1', durante il processo di calcolo della sindrome (per ulteriori approfondimenti, si consulti [10]).

Approfondimento: Processo di calcolo della Sindrome La PCM di un codice classico consente di sviluppare un algoritmo di correzione degli errori, attraverso l'utilizzo della distanza minima di Hamming, basato sul concetto di *sindrome*. In particolare, dato un certo vettore di informazione ricevuto y , viene definita come sindrome il seguente vettore [23]:

$$s = y\mathbf{H}^T. \quad (2.22)$$

Esso consente di rivelare gli errori di y . Se s è un vettore nullo, si può essere in uno tra i due seguenti possibili casi:

- non si sono verificati errori;
- si sono verificati errori ma y è uguale ad una parola di codice differente da quella trasmessa, dunque gli errori non sono rivelabili. Pertanto, se la sindrome s contiene almeno un elemento non nullo si è in presenza di errori [23].

Un altro importante aspetto da osservare è che, siccome la matrice LDPC è sparsa, i codici QLDPC risultanti presentano un'elevata degenerazione. Tuttavia, il rigoroso criterio del prodotto simplettico, associato alla progettazione dei codici stabilizzatori, limita fortemente le prestazioni dei codici QLDPC. In particolare, a causa di tale criterio, la matrice QLDPC contiene molti cicli brevi (teoria della decodifica iterativa), con lunghezza pari a 4. Questo, a sua volta, degrada anche le prestazioni del decodificatore LDPC [10]. Purtroppo infatti, il decoder LDPC non è in grado di prevenire l'impatto di errori degeneri (si guardi a tal proposito la Sezione 2.2.1), poiché soffre del cosiddetto "errore di degenerazione simmetrica", che deriva proprio da questi cicli [10]. Al fine di migliorare le prestazioni del decoder LDPC, caratterizzato da codici basati sui cicli brevi di lunghezza 4 e dall'errore di degenerazione simmetrica, Poulin *et al.* hanno concepito efficaci metodi euristici, nello specifico il metodo "random perturbation" e "enhanced

feedback" [125]. Il primo metodo si basa principalmente sull'introduzione di perturbazioni casuali per innescare la convergenza di decodifica. Successivamente, questa tecnica di decodifica QLDPC è stata ulteriormente migliorata ed è stato così introdotto il secondo metodo [10], [169]. Nonostante questi sviluppi, le prestazioni dei codici QLDPC non sono ancora paragonabili a quelle dei codici LDPC classici [10].

Nel 2008, Poulin *et al.* hanno progettato le controparti quantistiche dei codici turbo [127], [126]. In particolare, mentre i codici turbo classici si basano, generalmente, sulla concatenazione parallela dei codici convoluzionali, i QTCs (Quantum Turbo Codes) si fondano sulla concatenazione in serie dei QCCs (Quantum Convolutional Codes) stessi [127], [126].

Rispetto ai codici QLDPC, i QTCs offrono parametri di codice più flessibili, come ad esempio: la lunghezza della parola di codice (frame), il rate di codifica e la lunghezza del vincolo. Inoltre, la decodifica iterativa dei QTCs tiene conto dell'impatto degli errori degeneri.

Approfondimento: Errori dati da codificatori catastrofici Un codificatore si dice *catastrofico* (catastrophic) se emette una sequenza codificata a peso finito in corrispondenza di una sequenza di input a peso infinito [12]. Si ricorda che il *peso* di una generica sequenza di codice x (indicato con $w(x)$) corrisponde al numero di elementi non nulli della sequenza stessa. Inoltre, nel caso dei codici convoluzionali, ed è proprio il caso in analisi, dato che i codici turbo sono derivati da codici convoluzionali, si considera proprio il peso dell'intera sequenza x per caratterizzare il codice. Di conseguenza, stante questo richiamo, un codice catastrofico può portare alla propagazione di errori cosiddetti *catastrofici*, poiché un numero finito di errori sulla sequenza codificata può produrre un numero infinito di errori sulla sequenza decodificata. Ciò implica a sua volta che i codici costitutivi di un codice concatenato devono essere non catastrofici per raggiungere la convergenza di decodifica [12].

Tuttavia, i QCCs basati sui codici stabilizzatori non possono essere contemporaneamente sia ricorsivi (si ricorda che, in generale, i codici turbo originali si basano sulla concatenazione parallela di codici RSC) sia non catastrofici [127], [126], [79]. Entrambe queste proprietà, però, sono essenziali per la costruzione di buoni codici turbo. In particolare, è necessario un codice interno ricorsivo per raggiungere una distanza minima illimitata, mentre entrambi i codici, componenti di un codice concatenato in serie, devono essere non catastrofici per garantire la convergenza della decodifica a un rate di errore infinitesimamente basso.

Pertanto, i QTCs di [127], [126], presentano una distanza minima limitata, in quanto si basano su QCCs non catastrofici e non ricorsivi. Allo scopo di progettare QTCs a capacità prossima all'hashing bound, Babar *et al.* hanno sviluppato schemi EXIT per il dominio quantistico [15], mentre è stata proposta una struttura di codice quantistico irregolare convoluzionale (QIRCC) e di codice quantistico a rate unitario (QURC, Quantum Unitary Rate Code), rispettivamente da L. Hanzo *et al.* [11] e da Z. Babar [16].

Recentemente inoltre, è stato concepito un decodificatore per codici quantistici turbo completamente parallelo (FPQTD, Full-Parallel Quantum Turbo Decoder) che riduce sostanzialmente la latenza di decodifica [17]. Come poi verrà ulteriormente spiegato nei capitoli successivi, i codici stabilizzatori devono soddisfare il rigoroso criterio del prodotto simplettico. Di conseguenza, non tutti i codici classici possono essere importati nel dominio quantistico. Questo criterio comporta caratteristiche di codice indesiderate, ad esempio i cicli (inevitabili) di lunghezza 4 dei codici QLDPC.

Al fine di superare le problematiche associate al criterio del prodotto simplettico, è stata sviluppata la menzionata teoria dei codici quantistici EA che, come detto, si basa sul pre-sharing di qubit entangled tra il trasmettitore e il ricevitore su un canale non rumoroso [31], [32], [33]. La nozione "EA" (Entanglement-Assisted) è stata adottata per quasi tutte le famiglie di codici; dando vita ai codici EA-QLDPC [80], EA-QCC [172] e EA-QTC [175], [176], riducendo così le problematiche derivanti dal criterio del prodotto simplettico. Per esempio, i codici EA-QLDPC possono essere progettati senza cicli di lunghezza 4, di conseguenza, la prestazione risultante è paragonabile a quella dei codici LDPC classici. Analogamente, un codice EA-QCC può essere contemporaneamente ricorsivo e non catastrofico [175], [176]. Di conseguenza, gli EA-QTC sono in grado di avere una distanza minima illimitata. Quindi, la famiglia dei codici quantistici EA ha portato finalmente le prestazioni dei codici quantistici in linea con quelle delle loro controparti classiche.

Anche i codici polari hanno attirato una notevole attenzione all'interno della comunità scientifica in ambito quantistico. Infatti, Wilde e Guha, ispirati dalla possibilità di raggiungere il modello dei codici polari di Arikan e quindi le loro efficienti strutture di codifica e decodifica, hanno dimostrato l'esistenza del fenomeno della polarizzazione dei canali quantistici per le informazioni classiche e quantistiche in [173] e [174], rispettivamente. Questi modelli di codice si basano sul pre-sharing di qubit entangled, attraverso un canale privo di rumore, tra il trasmettitore e il ricevitore, un modello di canale è rappresentato nella Fig. 2.9. Nonostante tali sforzi, il decodificatore elaborato per i codici quantistici polari non è riuscito ad eguagliare la complessità di decodifica di quello di Arikan, cioè il decodificatore per i codici polari classici. Dunque, tali codici quantistici ancora non sono in grado di ottenere le stesse prestazioni della loro controparte classica. Questo problema è stato poi affrontato da Renes *et al.* in [133], dove sono stati progettati codici polari quantistici, di tipo CSS, a partire dai codici polari classici. Dunque, tali codici quantistici sono dotati di codificatori e decodificatori più efficienti. Invece, i primi codici polari quantistici non assistiti, cioè non EA, sono stati concepiti recentemente in [134], il che segna un'altra importante pietra miliare nello sviluppo di questi codici.

In definitiva, analogamente alla codifica classica, la ricerca sulla codifica quantistica ha come obiettivo quello di avvicinarsi il più possibile al limite descritto dalla capacità del canale quantistico; cioè l'hashing bound. In [12], sono stati progettati codici quantistici basati su parole di codice di lun-

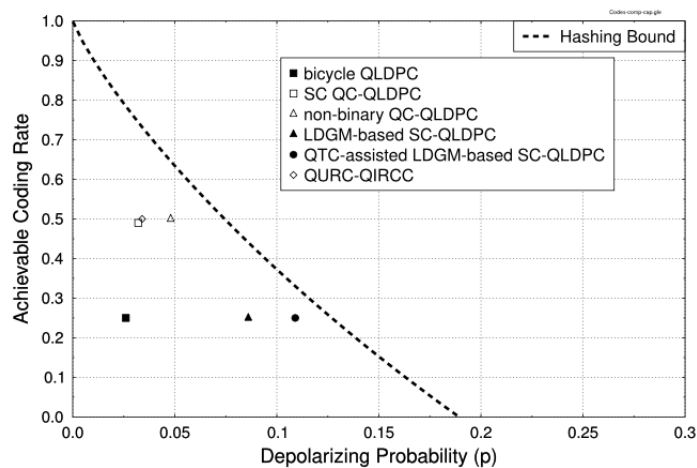


Figura 2.13: Prestazioni ottenibili a codeword error rate di 10^{-3} confrontato con l'Hashing bound [12].

ghezza elevata, come esemplificato dal codice QLDPC a doppio ciclo ($R = 0.25$, $n = 19\,014$) di [98], dal codice Spatially-Coupled Quasi-Cyclic (SC QC) QLDPC ($R = 0.49$, $n = 181\,000$) di [73], dal codice QC-QLDPC non binario ($R = 0.5$, $n = 20\,560$, $\text{GF}(2^{10})$) di [83] e [84], dal codice QLDPC basato sulla matrice LDGM (Low-Density Generator Matrix) ($R = 0.25$, $n = 76\,800$) di [6], dal codice SC-QLDPC (Spatially Coupled - Quantum Low Density Parity Check) basato su LDGM assistito da QTC ($R = 0.25$, $n = 821\,760$) (QTC-assisted LDGM-based SC-QLDPC code) di [108] e dal codice concatenato QURC-RCC ($R = 0.5$, $n = 139\,000$) di [16], le cui prestazioni sono confrontate con il limite di hashing, come mostrato in Fig. 2.13. Le parole di codice lunghe sono particolarmente dannose nel dominio quantistico, a causa dei brevi tempi di rilassamento e di dephasing dei qubit. Esplicitamente, se le parole di codice sono molto lunghe, i fotoni, che rappresentano i qubit all'interno del processo fisico della comunicazione quantistica, perderebbero la coerenza della funzione d'onda più velocemente di quanto possano essere corretti. Ciò significa che, in pratica, il sistema "non fa in tempo" a correggere la stringa di codice prima che essa perda il proprio contenuto informativo, a causa del fenomeno della decoerenza. In questo modo, l'informazione risultante sarebbe, almeno in parte, distorta, poiché ancora affetta da errori. Quindi, i codici quantistici che si basano su blocchi corti sono più efficienti da questo punto di vista, almeno fino a quando i tempi di rilassamento e dephasing dei qubit non diventeranno sufficientemente grandi, man mano che la tecnologia per la progettazione dell'hardware quantistico progredirà, migliorandosi.

Inoltre, nella ricerca per la progettazione delle controparti quantistiche dei codici classici conosciuti, sono stati proposti vari schemi EA, che impongono l'ulteriore aggiunta di qubit pre-shared attraverso un canale privo di rumore. Questo overhead però, deve essere ridotto al minimo per le implementazioni pratiche.

Riassumendo quanto detto finora, la Fig. 2.14 raccoglie tutte le più importanti famiglie di codici appartenenti alla storia della codifica classica e quantistica. In particolare, si nota che la codifica classica comincia con il lavoro pionieristico di Shannon sul finire degli anni 40', mentre quella quantistica ha il suo inizio nel 1995, con il codice di Shor.

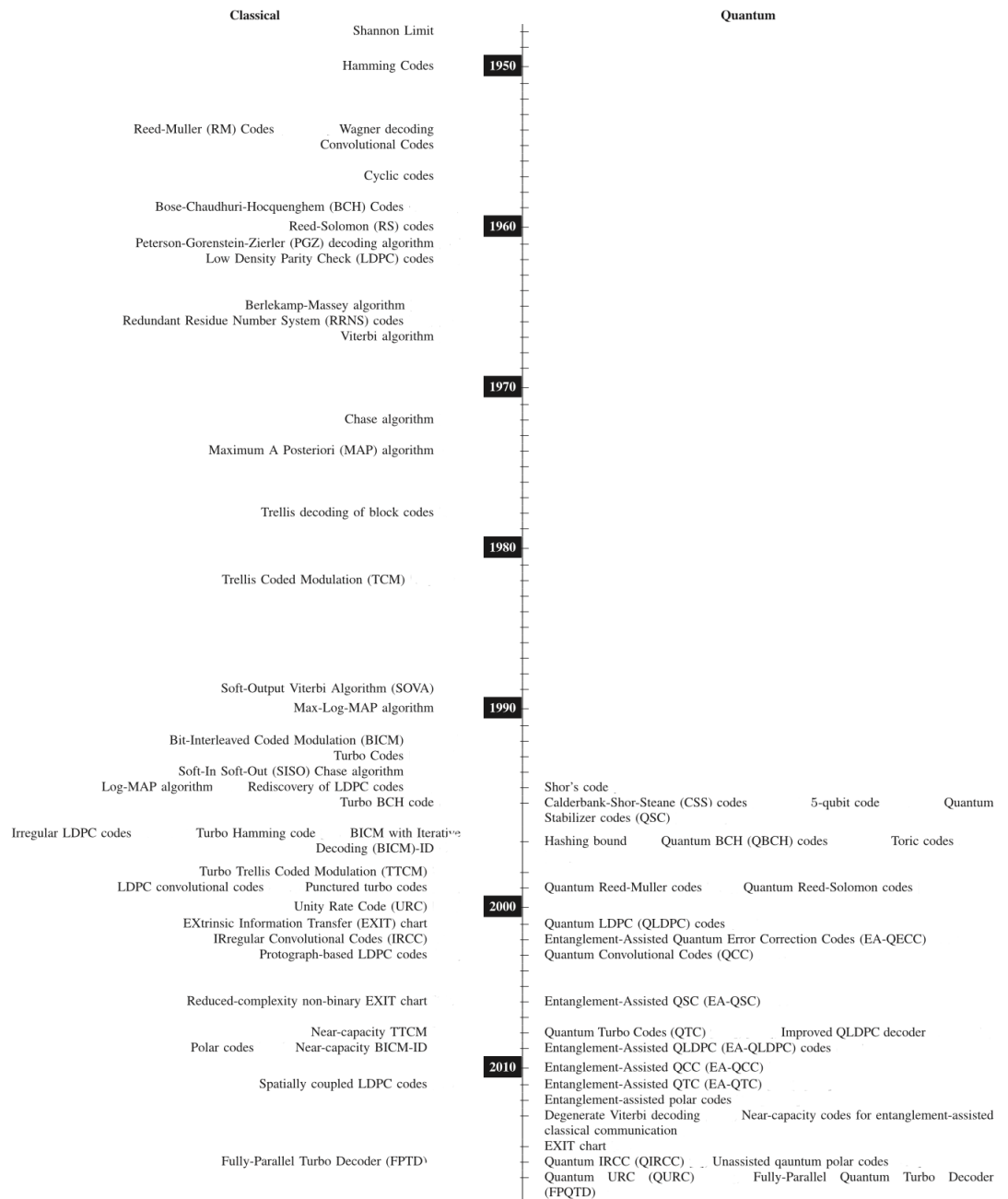


Figura 2.14: Principali risultati raggiunti nella storia della codifica classica e quantistica [12].

Capitolo 3

Transizione dal dominio classico al dominio quantistico

3.1 Premessa

Come si è visto nei capitoli precedenti, le leggi della meccanica quantistica rendono le tecniche di codifica quantistica intrinsecamente diverse dalle loro controparti classiche. Tuttavia, possono essere progettati, a partire dalle famiglie di codici classici già esistenti, codici quantistici efficienti, se si tengono in particolare considerazione le tre seguenti problematiche, che per naturali ragioni strutturali, non esistono nel dominio classico. Esse vengono elaborate e superate una per una, progettando la controparte quantistica del codice classico a ripetizione semplice, con rate pari ad $1/3$, che è in grado di correggere un singolo errore (si veda per chiarimenti *Approfondimento: Codici a Ripetizione* nella Sezione 2.2.2). La sintesi del ragionamento proposto è riassunta nella Fig. 3.9.

A. Teorema di non clonazione La maggior parte dei codici di correzione degli errori classici si basa su tecniche che combinano tra loro i bit, in particolare una classe di codici a blocco, quella dei codici a ripetizione, clona i bit stessi. Come spiegato in precedenza, vengono create copie multiple del bit di informazione per fornire ridondanza. Sfortunatamente, nel dominio quantistico, il teorema di non clonazione spiega che non esiste un operatore unitario che permette di copiare un qubit che si trova in uno stato arbitrario, come spiegato in [178].

B. Operazione di misura del qubit I codici classici si basano sulla misurazione dei valori dei bit ricevuti, attraverso l'intervento del decodificatore, che agisce con criterio hard o soft. Sfortunatamente, non è possibile misurare un qubit senza perturbarlo, il che comporterebbe inevitabilmente, al momento della misurazione, il collasso degli stati quantistici sovrapposti e quindi la perdita di informazione.

C. Natura degli errori quantistici I canali classici impongono solo errori

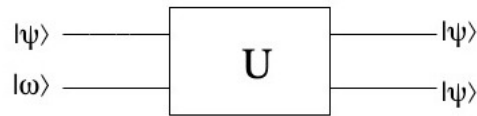


Figura 3.1: Schematico del circuito che rappresenta il funzionamento dell'operatore di copia \mathcal{U} .

di bit-flip e ne è un esempio il canale binario simmetrico citato nella Sezione 1.3.3. Al contrario, i canali quantistici determinano errori sia di bit-flip che di phase-flip, come accade per esempio nel canale di Pauli. Inoltre, la degradazione subita dalle particelle quantistiche si dice *continua*, poiché il qubit ricevuto può assumere un qualsiasi valore sulla sfera di Bloch; per chiarimenti si guardi l'apposito *Approfondimento* nella Sezione 1.1.1 oppure [116]. Naturalmente la caratterizzazione continua dei qubit si contrappone agli errori *discreti* che possono subire i bit classici.

3.2 Teorema di non clonazione

I codici quantistici sfruttano la ridondanza, nel dominio quantistico, senza clonare i qubit che portano l'informazione.

Nel dominio classico, invece, il codificatore di un codice a ripetizione a 3 bit copia ogni bit di informazione tre volte. In particolare, i bit di informazione 0 e 1 sono codificati come segue:

$$0 \rightarrow (000) \quad 1 \rightarrow (111). \quad (3.1)$$

Il processo di codifica dell'eq.(3.1) inevitabilmente non ha un equivalente quantistico, poiché in questo dominio non è consentita la clonazione di un qubit che si trova in uno stato arbitrario. Di seguito viene proposta una dimostrazione per assurdo che si basa sulla proposizione per cui tutti gli operatori unitari devono essere, per definizione, lineari. Sia allora \mathcal{U} un ipotetico operatore unitario di clonazione, chiamato, equivalentemente, operatore di copia. La proprietà di linearità impone all'operatore \mathcal{U} , dati due qubit generici negli stati $|\psi\rangle$ e $|\phi\rangle$, di verificare la seguente relazione:

$$\mathcal{U}(a|\psi\rangle + b|\phi\rangle) = a\mathcal{U}|\psi\rangle + b\mathcal{U}|\phi\rangle. \quad (3.2)$$

Questa dimostrazione pertanto vuole provare che \mathcal{U} è non lineare, cioè non rispetta il presupposto di partenza espresso dall'eq.(3.2) e quindi non esiste. Dato un certo qubit $|\psi\rangle$ che si trova in uno stato arbitrario e un qubit $|\omega\rangle$ noto, per semplicità si assuma $|\omega\rangle = |0\rangle$, entrambi posti in ingresso all'operatore \mathcal{U} , si ha in uscita una coppia di qubit nello stato arbitrario $|\psi\rangle$, come mostrato nella Fig. 3.1. Analiticamente, si ottiene:

$$\mathcal{U}(|\psi\rangle \otimes |\omega\rangle) = |\psi\rangle \otimes |\psi\rangle, \quad (3.3)$$

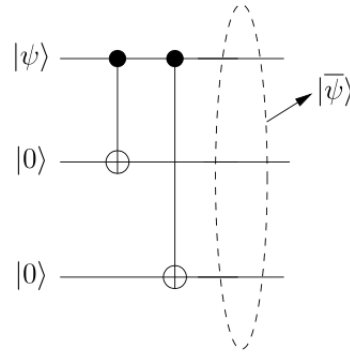


Figura 3.2: Circuito codificatore di un codice bit-flip a ripetizione a 3 qubit [12].

dove \otimes indica il prodotto tensoriale tra i due vettori. Si supponga poi che il generico stato arbitrario possa essere espresso (come di consueto) da $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, allora si ottiene:

$$\begin{aligned} |\psi\rangle \otimes |\omega\rangle &= |\psi\rangle \otimes |0\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle \\ &= \alpha|00\rangle + \beta|10\rangle. \end{aligned} \quad (3.4)$$

Dunque, per la proprietà di linearità dell'operatore di clonazione, si ha:

$$\mathcal{U}(\alpha|00\rangle + \beta|10\rangle) = \alpha\mathcal{U}|00\rangle + \beta\mathcal{U}|10\rangle. \quad (3.5)$$

A questo punto, si analizzano separatamente i due membri dell'equazione precedente, verificando se effettivamente sono uguali.

Per il membro di sinistra dell'eq.(3.5), data la definizione dell'operatore di copia, si ottiene:

$$\begin{aligned} \mathcal{U}(\alpha|00\rangle + \beta|10\rangle) &= (\alpha|00\rangle + \beta|10\rangle) \otimes (\alpha|00\rangle + \beta|10\rangle) \\ &= \alpha^2|0000\rangle + \alpha\beta|0010\rangle + \beta\alpha|1000\rangle + \beta^2|1010\rangle. \end{aligned} \quad (3.6)$$

Invece, per il membro di destra dell'eq.(3.5), applicando la definizione dell'operatore di clonazione e svolgendo i calcoli si trova:

$$\begin{aligned} \alpha\mathcal{U}|00\rangle + \beta\mathcal{U}|10\rangle &= \alpha(|00\rangle \otimes |00\rangle) + \beta(|10\rangle \otimes |10\rangle) \\ &= \alpha|0000\rangle + \beta|1010\rangle. \end{aligned} \quad (3.7)$$

Si conclude che i due membri dell'eq.(3.5) sono diversi, quindi nel dominio quantistico l'ipotetico operatore di clonazione è non lineare (c.v.d.) e pertanto \mathcal{U} non può esistere. Di conseguenza, $|\psi\rangle$ non può essere codificato in $|\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle$, seguendo l'esempio della codifica classica, contenuto nell'eq.(3.1).

Dunque, è stato progettato il codice a ripetizione a 3 qubit di tipo bit-flip per superare il vincolo di clonazione; cioè si clonano solo gli stati base $|0\rangle$ e $|1\rangle$, chiamati anche *basi computazionali*, piuttosto che lo stato sovrapposto $|\psi\rangle$. La codifica avviene allora come segue:

$$\begin{aligned} |0\rangle &\rightarrow |\bar{0}\rangle \equiv |000\rangle, \\ |1\rangle &\rightarrow |\bar{1}\rangle \equiv |111\rangle. \end{aligned} \quad (3.8)$$

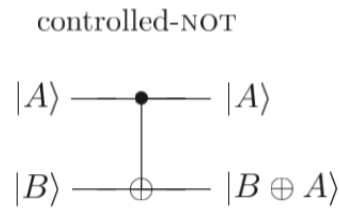


Figura 3.3: Gate CNOT quantistico a due qubit [116].

In particolare i due qubit ausiliari, chiamati così poiché servono per creare la ridondanza, nello stato $|0\rangle$ sono *entangled* con il qubit di informazione $|\psi\rangle$, come mostrato nel circuito di Fig. 3.2, attraverso l'utilizzo doppio della porta logica quantistica Controlled-NOT (CNOT), riportata dalla Fig. 3.3.

Approfondimento: Gate Quantistico Controlled-NOT Questa porta logica ha due qubit di input, che sono il qubit di controllo (control qubit) ed il qubit di destinazione (target qubit), rispettivamente. In output sono presenti ancora due qubit; in particolare il qubit di controllo rimane sempre inalterato, mentre il qubit di destinazione potrebbe variare. La rappresentazione circuitale è riportata in Fig. 3.3, dove la linea più in alto indica il qubit di controllo, mentre l'altra il qubit di destinazione. Il funzionamento è descritto nel seguito.

Se il qubit di controllo è impostato su 0, il qubit di destinazione rimane inalterato. Se il qubit di controllo, invece, è impostato su 1, il qubit di destinazione viene "flippato", cioè rovesciato. In formula, si ha:

$$\text{CNOT}(|\psi_0\rangle, |\psi_1\rangle) = |\psi_0\rangle \otimes |\psi_0 \oplus \psi_1\rangle, \quad (3.9)$$

dove $|\psi_0\rangle$ è il qubit di controllo, $|\psi_1\rangle$ è il qubit di destinazione, mentre \otimes indica il prodotto tensoriale e \oplus la somma modulo 2. Dunque si hanno 4 diversi possibili casi:

$$\begin{aligned} (i) |00\rangle &\rightarrow |00\rangle; & (ii) |01\rangle &\rightarrow |01\rangle; \\ (iii) |10\rangle &\rightarrow |11\rangle; & (iv) |11\rangle &\rightarrow |10\rangle. \end{aligned} \quad (3.10)$$

In altri termini, l'uscita può essere vista come un'operazione reversibile di una porta Exclusive-OR classica; quindi, la porta CNOT può essere considerata la controparte quantistica della porta XOR. Si tratta di una generalizzazione della porta XOR classica, poiché l'azione della porta quantistica CNOT può essere riassunta come $|A, B\rangle \rightarrow |A, A \oplus B\rangle$, che è esattamente ciò che fa la porta XOR. Diversamente dalla sua controparte classica, in cui i due ingressi sono combinati per produrre una singola uscita XOR irreversibile, il funzionamento di una porta CNOT è di fatto reversibile, poiché si possono ricostruire i due ingressi (controllo e destinazione) a partire dalle due corrispondenti uscite [116].

Di conseguenza, il codificatore di Fig. 3.2 replica gli stati $|0\rangle$ e $|1\rangle$ tre volte

Tabella 3.1: Look-Up Table per un codice classico a ripetizione con rate 1/3.

Syndrome	Error
s	e
(00)	(000)
(11)	(100)
(10)	(010)
(01)	(001)

nell'output codificato a 3 qubit $|\bar{\psi}\rangle$, che è dato da:

$$\begin{aligned} |\psi\rangle \otimes |0\rangle^{\otimes 2} &\rightarrow |\bar{\psi}\rangle = \alpha |\bar{0}\rangle + \beta |\bar{1}\rangle \\ &\equiv \alpha |000\rangle + \beta |111\rangle, \end{aligned} \quad (3.11)$$

dove l'apice $\otimes 2$ indica che lo stato $|0\rangle$ del qubit è stato replicato due volte.

3.3 Operazione di misura del qubit

La misurazione (o osservazione) del qubit è stata già citata più volte in questa trattazione, ma non è stato ancora spiegato di che cosa si tratti in maniera puntuale.

I codici quantistici devono stimare gli errori imposti dal canale senza misurare i qubit ricevuti. Nel dominio classico, lato ricevitore, il decodificatore di un codice a ripetizione a 3 bit legge i bit ricevuti e li decodifica, seguendo il criterio a maggioranza. Ad esempio, la sequenza ricevuta (011) è decodificata a 1, mentre (100) è decodificata a 0. Ciò richiede l'osservazione della sequenza ricevuta, che purtroppo non è possibile nel dominio quantistico. In particolare, se il qubit ricevuto ($|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$) viene misurato, esso collasserà, come visto, nello stato $|0\rangle$ o $|1\rangle$ con una probabilità di $|\alpha|^2$ o $|\beta|^2$, rispettivamente.

In alternativa, un codice a blocco lineare classico $C(n, k)$ può essere decodificato usando una PCM \mathbf{H} costituita da $(n - k) \times n$ elementi, in modo che tutte le parole di codice legittime e senza errori \bar{x} , diano:

$$\bar{x}\mathbf{H}^T = 0, \quad (3.12)$$

per ulteriori chiarimenti sulla matrice di parità \mathbf{H} si consulti l'*Approfondimento: Matrice Generatrice e Matrice di Parità* della sezione 2.1. Data una parola di codice ricevuta $y = \bar{x} + e$, dove e è il vettore che rappresenta l'errore indotto dal canale, il vettore di sindrome s , associato a y , è composto da $(n - k)$ simboli. Esso identifica univocamente e inequivocabilmente il vettore d'errore (se il numero di errori indotti dal canale rientra nella capacità di correzione degli errori del codice). È calcolato come segue:

$$s = y\mathbf{H}^T = (\bar{x} + e)\mathbf{H}^T = \bar{x}\mathbf{H}^T + e\mathbf{H}^T = e\mathbf{H}^T, \quad (3.13)$$

dato che la quantità $\bar{x}\mathbf{H}^T$ è nulla, come specificato dall'eq.(3.12).

Quindi, la sindrome può essere utilizzata, in linea di principio, per stimare

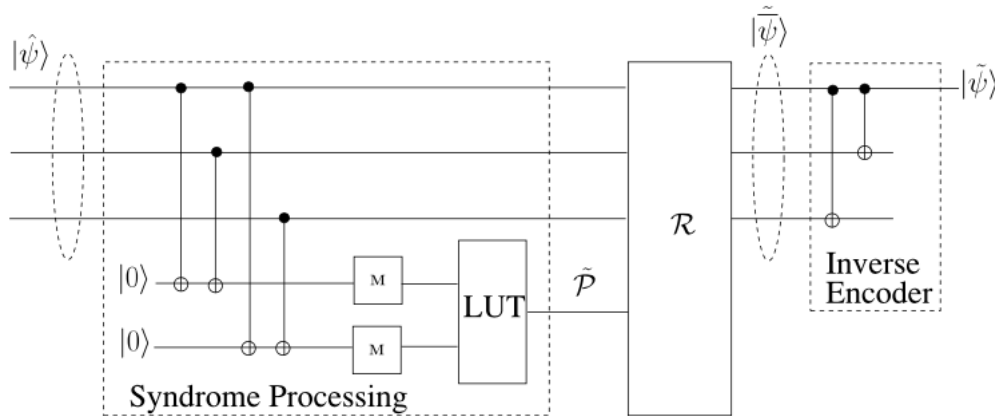


Figura 3.4: Circuito di decodifica del codice a ripetizione a 3 qubit di tipo bit-flip [12].

il vettore di errore e , utilizzando una Look-Up Table (LUT) già calcolata. Più esplicitamente, poiché un codice a blocco lineare $C(n, k)$ ha $(n - k)$ bit di parità, si possono avere $2^{(n-k)}$ sindromi uniche. Di conseguenza, è possibile stimare $2^{(n-k)}$ pattern di errore, composti ognuno da n simboli, che sono appunto pre-computati e memorizzati in una LUT.

Allo stesso modo, anche un codice classico a ripetizione a 3 bit può essere decodificato utilizzando la tecnica della sindrome basata sulla PCM. La PCM associata è data da:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \quad (3.14)$$

che produce un vettore di sindrome a valore zero, cioè completamente composto da zeri, per entrambe le parole di codice valide (111 e 000), mentre almeno uno dei due elementi della sindrome è 1 quando si verifica un singolo errore di bit-flip. La LUT risultante è riportata nella Tabella 3.1, che registra tutti gli errori di singolo bit-flip che possono essere stimati in un codice classico a ripetizione a 3 bit. Intuitivamente, la prima riga di \mathbf{H} confronta i primi due bit ricevuti di y . Se entrambi i bit sono uguali, il bit associato alla sindrome è 0, mentre se sono diversi, il bit è 1. Allo stesso modo, la seconda riga di \mathbf{H} confronta il primo e il terzo bit di y .

Operando in questo modo, un codice a ripetizione a 3 qubit di tipo bit-flip può essere decodificato utilizzando un decodificatore basato sulla sindrome, che confronta semplicemente i qubit senza in realtà conoscerne i valori specifici. Ciò si ottiene utilizzando due qubit ausiliari e le porte CNOT dell'eq.(3.9), come mostrato nel blocco "Syndrome Processing" di Fig. 3.4. In particolare, si può osservare nella Fig. 3.4 che il primo qubit ausiliario è invertito, se i primi due qubit sono diversi, mentre il secondo qubit ausiliario è invertito, quando il primo e il terzo qubit sono diversi. In altre parole, se $|\psi\rangle$ viene trasmesso, allora si potrebbe ricevere una delle seguenti quattro sequenze $|\hat{\psi}\rangle$, supponendo che durante la trasmissione accada solo

un singolo errore di bit-flip:

$$\begin{aligned}
(i) \quad & \alpha |000\rangle + \beta |111\rangle \xrightarrow{\mathbf{III}} \alpha |000\rangle + \beta |111\rangle; \\
(ii) \quad & \alpha |000\rangle + \beta |111\rangle \xrightarrow{\mathbf{XII}} \alpha |100\rangle + \beta |011\rangle; \\
(iii) \quad & \alpha |000\rangle + \beta |111\rangle \xrightarrow{\mathbf{IXI}} \alpha |010\rangle + \beta |101\rangle; \\
(iv) \quad & \alpha |000\rangle + \beta |111\rangle \xrightarrow{\mathbf{IIX}} \alpha |001\rangle + \beta |110\rangle.
\end{aligned} \tag{3.15}$$

Il processo di calcolo della sindrome opera su ciascuna delle possibili sequenze $|\hat{\psi}\rangle$ ricevute. Partendo dal caso più banale (i) dell'eq.(3.15), se tutti e tre i qubit rimangono uguali, come nel caso del vettore di errore **III**, i qubit ausiliari risultano dunque inalterati. Pertanto si può scrivere:

$$\begin{aligned}
\alpha |000\rangle + \beta |111\rangle \otimes |0\rangle^{\otimes 2} & \rightarrow \alpha |00000\rangle + \beta |11111\rangle \\
& = (\alpha |000\rangle + \beta |111\rangle) \otimes |00\rangle.
\end{aligned} \tag{3.16}$$

In secondo luogo, riferendosi al caso (ii), quando sia il primo che il secondo qubit, così come il primo e il terzo, sono diversi tra loro, entrambi i qubit ausiliari risultano invertiti. Si tratta di ciò che accade nel caso del vettore di errore **XII** e può essere formalizzato come segue:

$$\begin{aligned}
\alpha |100\rangle + \beta |011\rangle \otimes |0\rangle^{\otimes 2} & \rightarrow \alpha |10011\rangle + \beta |01111\rangle \\
& = (\alpha |100\rangle + \beta |011\rangle) \otimes |11\rangle.
\end{aligned} \tag{3.17}$$

Nel terzo evento (iii), quando il primo ed il secondo qubit sono diversi, ma il primo e il terzo sono identici, come nel caso del vettore di errore **IXI**, solo il primo qubit ausiliario viene invertito. Si ottiene:

$$\begin{aligned}
\alpha |010\rangle + \beta |101\rangle \otimes |0\rangle^{\otimes 2} & \rightarrow \alpha |01010\rangle + \beta |10110\rangle \\
& = (\alpha |010\rangle + \beta |101\rangle) \otimes |10\rangle.
\end{aligned} \tag{3.18}$$

Infine, come riportato in (iv), quando il primo ed il secondo qubit sono identici, ma il primo ed il terzo sono diversi, come nel caso del vettore di errore **IIX**, solo il secondo qubit ausiliario è invertito. Infatti si ha:

$$\begin{aligned}
\alpha |001\rangle + \beta |110\rangle \otimes |0\rangle^{\otimes 2} & \rightarrow \alpha |00101\rangle + \beta |11001\rangle \\
& = (\alpha |001\rangle + \beta |110\rangle) \otimes |01\rangle.
\end{aligned} \tag{3.19}$$

I qubit ausiliari delle eq.(3.16)-(3.19) sono misurati nel blocco M, di Fig. 3.4, per produrre la sindrome classica, che può assumere solo uno dei quattro valori possibili, cioè (00), (11), (10) e (01). La sindrome quindi, può essere usata per stimare l'errore \mathcal{P} usando la LUT della Fig. 3.4, già esplicitamente riportata nella Tabella 3.1. Successivamente, il codice trasmesso viene recuperato applicando l'operazione di recupero \mathcal{R} (vedere Fig. 3.4), che mira a correggere i bit-flip indotti dal canale, in base all'errore stimato $\tilde{\mathcal{P}}$. In particolare, nel contesto del codice a ripetizione a 3 qubit di tipo bit-flip, le porte quantistiche di Pauli **X** vengono utilizzate durante il processo di recupero dell'informazione, per contrastare l'impatto degli errore indotti dal

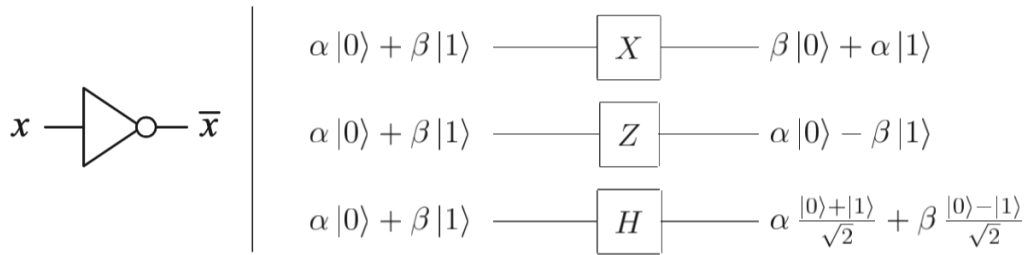


Figura 3.5: Confronto tra la porta logica "NOT" a singolo bit (sinistra) e gli operatori di Pauli e di Hadamard: **X** (bit-flip), **Z** (phase-flip) e **H** [116].

canale, riportati nella Tabella 3.1. Infine, la parola di informazione stimata $|\tilde{\psi}\rangle$, viene ripristinata, alimentando il codificatore inverso (inverse decoder in figura) con la parola di codice recuperata $|\tilde{\psi}\rangle$. Tale encoder di Fig. 3.4, è simile a quello presente in Fig. 3.2, cioè è sufficiente utilizzare il codificatore diretto da destra verso sinistra, per realizzare l'effetto inverso, mappando quindi i qubit codificati recuperati sui qubit di informazione.

A questo punto, è pertinente osservare che un codice classico a ripetizione è naturalmente sistematico. Di conseguenza, il bit d'informazione può essere estratto dalla parola di codice ricevuta senza utilizzare un'operazione di codifica inversa. Al contrario, il qubit d'informazione di un codice quantistico a ripetizione è *entangled* con i qubit ausiliari e quindi non può essere separato senza un encoder inverso. Ad esempio, se $|\tilde{\psi}\rangle = \alpha|000\rangle + \beta|111\rangle$, si devono applicare le due porte quantistiche CNOT del codificatore inverso della Fig. 3.2 e si ottiene:

$$\begin{aligned} \alpha|000\rangle + \beta|100\rangle &= (\alpha|0\rangle + \beta|1\rangle)|00\rangle \\ &\equiv |\tilde{\psi}\rangle|00\rangle, \end{aligned} \quad (3.20)$$

quindi si separa il qubit di informazione $|\tilde{\psi}\rangle$ dai qubit ausiliari $|00\rangle$.

3.4 Natura degli errori quantistici

I codici quantistici sono in grado di correggere gli errori quantistici di tipo bit-flip, phase-flip e bit-and-phase-flip.

Nel dominio classico, quando le sequenze di bit codificate (000) o (111) vengono trasmesse, uno 0 può essere invertito in un 1 e un 1 può essere invertito in uno 0. Di conseguenza, il canale di comunicazione classico impone, alle parole di codice trasmesse, solo errori discreti di bit-flip. Al contrario, quando un qubit viene trasmesso sul canale di depolarizzazione, può verificarsi un errore di inversione di bit, di inversione di fase e di inversione di bit e di fase, con la stessa probabilità $p/3$, come discusso nella Sezione 1.3.3. Un codice a ripetizione a 3 qubit di tipo phase-flip può essere progettato in modo analogo ad un codice a ripetizione a 3 qubit di tipo bit-flip, poiché l'inversione di fase e l'inversione di bit differiscono solo negli stati ($|0\rangle$ e $|1\rangle$). In

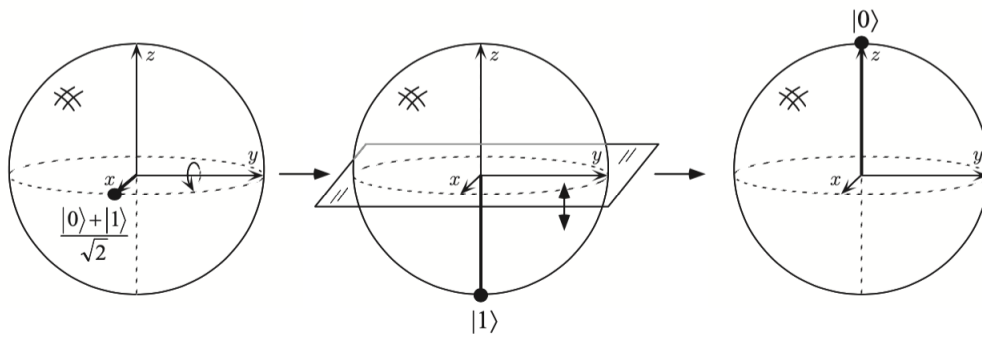


Figura 3.6: Visualizzazione del gate di Hadamard sulla sfera di Bloch, dato lo stato di ingresso $(|0\rangle + |1\rangle)/\sqrt{2}$ [116].

particolare, l'operazione di bit-flip capovolge la base computazionale $\{|0\rangle, |1\rangle\}$, mentre il phase-flip cambia la base di Hadamard, cioè il segno $\{|+\rangle, |-\rangle\}$. L'operazione di phase-flip è definita dalle seguenti due relazioni:

$$\begin{aligned} |+\rangle &\equiv H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}; \\ |-\rangle &\equiv H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \end{aligned} \quad (3.21)$$

dove H rappresenta l'operatore di Hadamard (Hadamard gate) che agisce su di un singolo qubit ed è specificato dalla seguente matrice [116]:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (3.22)$$

Approfondimento: Operatore di Hadamard Questo gate è talvolta descritto come una "radice quadrata della porta logica NOT", in quanto trasforma lo stato $|0\rangle$ in $(|0\rangle + |1\rangle)/\sqrt{2}$ (prima colonna di H), cioè uno stato intermedio tra $|0\rangle$ e $|1\rangle$, e trasforma $|1\rangle$ in $(|0\rangle - |1\rangle)/\sqrt{2}$ (seconda colonna di H), che è anch'esso "a metà strada" tra $|0\rangle$ e $|1\rangle$. A tal proposito, alcune importanti porte a singolo qubit sono mostrate in Fig. 3.5 e messe a confronto con il "NOT" classico [116].

Si noti, tuttavia, che $H^2 = \mathbf{I}$, cioè anche H è caratterizzata dalla stessa proprietà di \mathbf{X} , \mathbf{Z} , \mathbf{Y} e \mathbf{I} , come opportunamente investigato nell'apposito *Approfondimento: Operatori di Pauli* della Sezione 1.3.3. Quindi si conclude che applicando H due volte ad un certo stato quantistico non lo si modifica. Il gate di Hadamard è una delle porte quantistiche più utili e vale la pena provare a visualizzarne il funzionamento con la sfera di Bloch. L'operatore di Hadamard corrisponde pertanto ad una rotazione della sfera intorno all'asse y di 90° , seguita da una rotazione intorno all'asse x di 180° , come illustrato nella Fig. 3.6 [116].

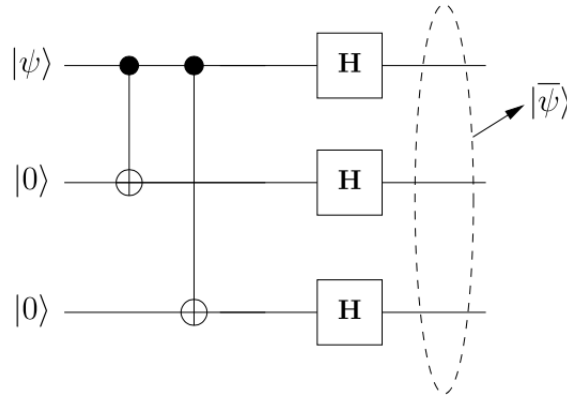


Figura 3.7: Circuito codificatore di un codice a ripetizione di tipo phase-flip a 3 qubit, dove il qubit d'informazione $|\psi\rangle$ è codificato in $|\bar{\psi}\rangle$ utilizzando due qubit ausiliari [12].

Pertanto, un errore di phase-flip (operatore di Pauli \mathbf{Z}) commuta la base di Hadamard come segue:

$$\begin{aligned}\mathbf{Z}|+\rangle &= |-\rangle; \\ \mathbf{Z}|-\rangle &= |+\rangle,\end{aligned}\tag{3.23}$$

mentre un errore di bit-flip (operatore di Pauli \mathbf{X}) agisce cambiando la base computazionale. Si ottiene infatti:

$$\begin{aligned}\mathbf{X}|0\rangle &= |1\rangle; \\ \mathbf{X}|1\rangle &= |0\rangle.\end{aligned}\tag{3.24}$$

Quindi, un codice a ripetizione di tipo phase-flip a 3 qubit protegge dai singoli capovolgimenti di fase (cioè può correggere fino ad un errore di phase-flip), replicando gli stati della base di Hadamard invece che copiare il qubit di informazione. Si opera come segue:

$$\begin{aligned}|0\rangle &\rightarrow |\bar{0}\rangle \equiv |+++ \rangle; \\ |1\rangle &\rightarrow |\bar{1}\rangle \equiv |-- - \rangle.\end{aligned}\tag{3.25}$$

Questo può essere ottenuto utilizzando il circuito di codifica di Fig. 3.7, che, come si può osservare, è simile a quello di Fig. 3.2, in cui i due qubit ausiliari sono entangled con il qubit di informazione $|\psi\rangle$, utilizzando circuiti CNOT e di Hadamard. In questo caso, però, sono presenti le porte di Hadamard (H) che trasformano la base computazionale ($|0\rangle$ o $|1\rangle$) nella base di Hadamard ($|-\rangle$ o $|+\rangle$). Di conseguenza, $|\psi\rangle$ è codificato come:

$$\begin{aligned}|\psi\rangle \otimes |0\rangle^{\otimes 2} &\rightarrow |\bar{\psi}\rangle = \alpha |\bar{0}\rangle + \beta |\bar{1}\rangle \\ &\equiv \alpha |+++ \rangle + \beta |-- - \rangle.\end{aligned}\tag{3.26}$$

Analogamente al decodificatore del codice a ripetizione a 3 qubit di tipo bit-flip, il decoder di un codice a ripetizione a 3 qubit di tipo phase-flip utilizza

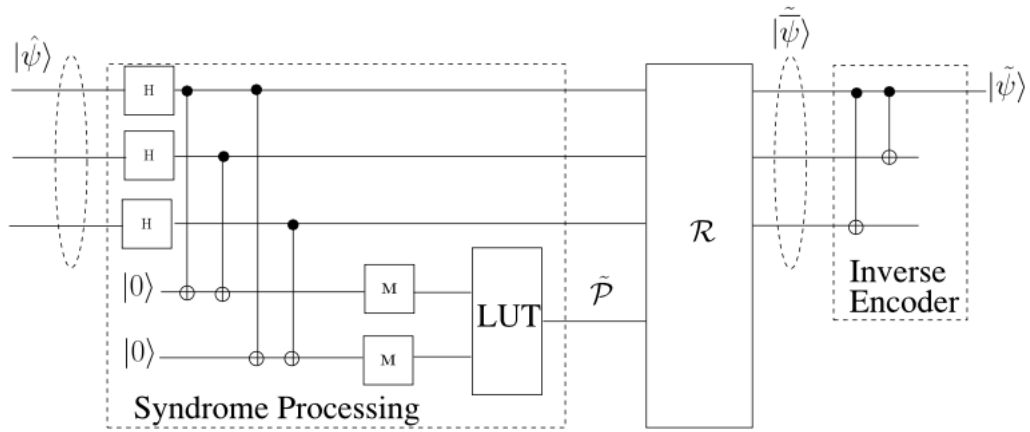


Figura 3.8: Circuito decodificatore di un codice a ripetizione di tipo phase-flip a 3 qubit [12].

due qubit ausiliari per calcolare il processo di sindrome a 2 bit associato. Il primo qubit nel processo di sindrome confronta la fase del primo e del secondo qubit, mentre il secondo confronta la fase del primo e del terzo qubit. Ciò può essere ottenuto utilizzando il circuito decodificatore mostrato in Fig. 3.8, che è lo stesso del codice a ripetizione a 3 qubit di tipo bit-flip, con la sola aggiunta dei gate di Hadamard che servono per trasformare le basi di Hadamard negli stati quantistici, cioè $|0\rangle$ e $|1\rangle$. In altre parole, si può osservare che gli operatori di Hadamard sono utilizzati all'ingresso e all'uscita del canale per trasformare le *inversioni di fase* nelle *inversioni di bit*. Quindi, sia gli errori di tipo bit-flip che quelli di tipo phase-flip possono essere corretti concatenando i codici a ripetizione a 3 qubit di tipo phase-flip e di tipo bit-flip. Essi, in realtà, costituiscono il codice con rate $1/9$ di Shor [145], in grado di correggere un singolo errore di tipo bit-flip, o phase-flip o, in alternativa, un errore di tipo bit-and-phase-flip, come detto nel paragrafo 2.2.2. In particolare, il qubit che porta informazione viene prima codificato utilizzando le basi di Hadamard in virtù dell'eq.(3.26). I tre qubit codificati risultanti sono poi codificati in modo indipendente usando il codice a ripetizione di tipo bit-flip descritto nell'eq.(3.11).

Osservazione A questo proposito, si osserva che l'ordine di concatenazione delle operazioni è molto importante. Se tale ordine fosse invertito, cioè, se si invocasse un codice a ripetizione di tipo bit-flip seguito da un codice a ripetizione di tipo phase-flip, allora il codice quantistico risultante codificherebbe gli stati in:

$$\begin{aligned} |\bar{0}\rangle &\rightarrow |+++ \rangle \otimes |+++ \rangle \otimes |+++ \rangle; \\ |\bar{1}\rangle &\rightarrow |-- - \rangle \otimes |-- - \rangle \otimes |-- - \rangle. \end{aligned}$$

Si tratta pertanto di un codice a ripetizione con rate $1/9$ di tipo phase-flip che è sicuramente efficace, ma d'altro canto, inevitabilmente, non è in grado di correggere errori di tipo bit-flip.

Quindi, gli stati di base sono mappati su tre blocchi a 3 qubit, come segue:

$$\begin{aligned}
|\bar{0}\rangle &\equiv \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \otimes \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \\
&\otimes \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle); \\
|\bar{1}\rangle &\equiv \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) \otimes \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle) \\
&\otimes \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle), \tag{3.27}
\end{aligned}$$

dove i tre qubit all'interno di un blocco sono le parole di un codice a ripetizione di tipo bit-flip, mentre i tre blocchi sono il risultato della codifica di un codice a ripetizione di tipo phase-flip. Svolgendo il prodotto tensore (\otimes) dell'eq.(3.27), si ottiene:

$$\begin{aligned}
|\bar{0}\rangle &\equiv \frac{1}{\sqrt{8}}(|000000000\rangle + |000000111\rangle + |000111000\rangle \\
&+ |000111111\rangle + |111000000\rangle + |111000111\rangle \\
&+ |111111000\rangle + |111111111\rangle); \\
|\bar{1}\rangle &\equiv \frac{1}{\sqrt{8}}(|000000000\rangle - |000000111\rangle - |000111000\rangle \\
&+ |000111111\rangle - |111000000\rangle + |111000111\rangle \\
&+ |111111000\rangle - |111111111\rangle). \tag{3.28}
\end{aligned}$$

Di conseguenza, lo stato codificato $|\bar{\psi}\rangle$ equivale a:

$$\begin{aligned}
\alpha |\bar{0}\rangle + \beta |\bar{1}\rangle &\equiv \frac{1}{\sqrt{8}}(\alpha + \beta)(|000000000\rangle + |000111111\rangle \\
&+ |111000111\rangle + |111111000\rangle) + \frac{1}{\sqrt{8}}(\alpha - \beta) \\
&\times (|000000111\rangle + |000111000\rangle + |111000000\rangle \\
&+ |111111111\rangle), \tag{3.29}
\end{aligned}$$

dove \times rappresenta l'operazione di prodotto di un vettore per uno scalare. Lo stato $|\bar{\psi}\rangle$ può essere decodificato concatenando i circuiti di decodifica di Fig. 3.4 e Fig. 3.8. In particolare, i tre blocchi, da 3 qubit ognuno, dell'eq.(3.27) sono prima decodificati in modo indipendente utilizzando il decoder a ripetizione a 3 qubit di tipo bit-flip di Fig. 3.4, ottenendo tre qubit d'informazione. Di conseguenza, i tre qubit risultanti costituiscono il codice ricevuto per il decodificatore a ripetizione a 3 qubit, di tipo phase-flip, di Fig. 3.8 e dunque vengono decodificati da questo.

Inoltre, come mostrato dall'eq.(1.59), nel canale di depolarizzazione il qubit ricevuto può trovarsi nella superposition di tutti e tre i possibili errori. In sostanza, un codice classico $C(n, k)$, progettato per proteggere un messaggio

di k bit di informazione, codificati in una parola di codice di n simboli, mira a ripristinare una delle 2^k parole di codice valide. Al contrario, poiché una parola di k qubit di informazione sia completamente descritta dai 2^k coefficienti complessi a valore continuo, i codici quantistici devono ripristinare tutti i 2^k coefficienti complessi [98] per ritrovare la parola cercata. Fortunatamente però, non vanno cercati i coefficienti complessi, azione che sarebbe piuttosto ambiziosa, ma è sufficiente misurare i qubit ausiliari, utilizzati per calcolare la sindrome. In particolare, sebbene i 2^k coefficienti siano a valore continuo, l'insieme di tutti gli errori che agiscono sui qubit d'informazione può essere discretizzato, a patto che il codice sia in grado di correggere errori discreti sia di tipo bit-flip, sia di tipo phase-flip così come di tipo bit-and-phase-flip. Per esempio, si supponga che possa verificarsi durante la trasmissione soltanto un singolo errore di inversione di bit. Allora, la parola di codice ricevuta, supponendo di avere un codice a ripetizione a 3 qubit, può essere espressa come segue:

$$|\hat{\psi}\rangle = p_0\mathbf{III}|\psi\rangle + p_1\mathbf{XII}|\psi\rangle + p_2\mathbf{IXI}|\psi\rangle + p_3\mathbf{IIX}|\psi\rangle, \quad (3.30)$$

dove p_0 è la probabilità che la trasmissione non sia affetta da errori, mentre p_i è la probabilità di incontrare un errore di tipo bit-flip sul qubit i -esimo, con $1 \leq i \leq 3$. Il processo di calcolo della sindrome di Fig. 3.4 rende entangled i due qubit ausiliari con $|\hat{\psi}\rangle$ dell'eq.(3.30), come segue:

$$\begin{aligned} |\hat{\psi}\rangle \otimes |0\rangle^{\otimes 2} \rightarrow p_0(\mathbf{III}|\bar{\psi}\rangle)|00\rangle + p_1(\mathbf{XII}|\bar{\psi}\rangle)|11\rangle \\ + p_2(\mathbf{IXI}|\bar{\psi}\rangle)|10\rangle + p_3(\mathbf{IIX}|\bar{\psi}\rangle)|01\rangle. \end{aligned} \quad (3.31)$$

Dunque si osserva che $|\hat{\psi}\rangle$ collassa in uno dei quattro stati sovrapposti, quando vengono misurati i qubit ausiliari. Lo stato risultante può quindi essere corretto in base alla specifica sindrome osservata.

Per riassumere quanto detto finora, si può guardare la Fig. 3.9. Essa rappresenta la transizione dal dominio classico a quello quantistico dei codici per la correzione degli errori [11]. Il problema viene scomposto in tre parti, in ragione dei tre macro-blocchi che compongono ogni sistema di comunicazione: il codificatore, il canale ed il decodificatore. Per ognuno di questi elementi viene proposto un confronto tra il dominio classico e quello quantistico.

Codificatore: gli encoder classici, per i codici a ripetizione, copiano i bit d'informazione. Purtroppo, come detto, non esiste alcun operatore di clonazione quantistico. Di conseguenza, nei codici quantistici i qubit d'informazione sono entangled con i qubit ausiliari, in modo che le informazioni vengano clonate negli stati di base.

Canale: le informazioni classiche possono riscontrare solo errori di tipo bit-flip, mentre i qubit possono sperimentare errori di tipo bit-flip, phase-flip e quindi anche bit-and-phase-flip. Gli errori aggiuntivi di inversione di fase, nel dominio quantistico, possono essere corretti usando la base di Hadamard $\{|+\rangle, |-\rangle\}$.

Decodificatore: i decoder classici misurano i bit ricevuti per stimare le informazioni trasmesse. Sfortunatamente, i qubit non possono essere

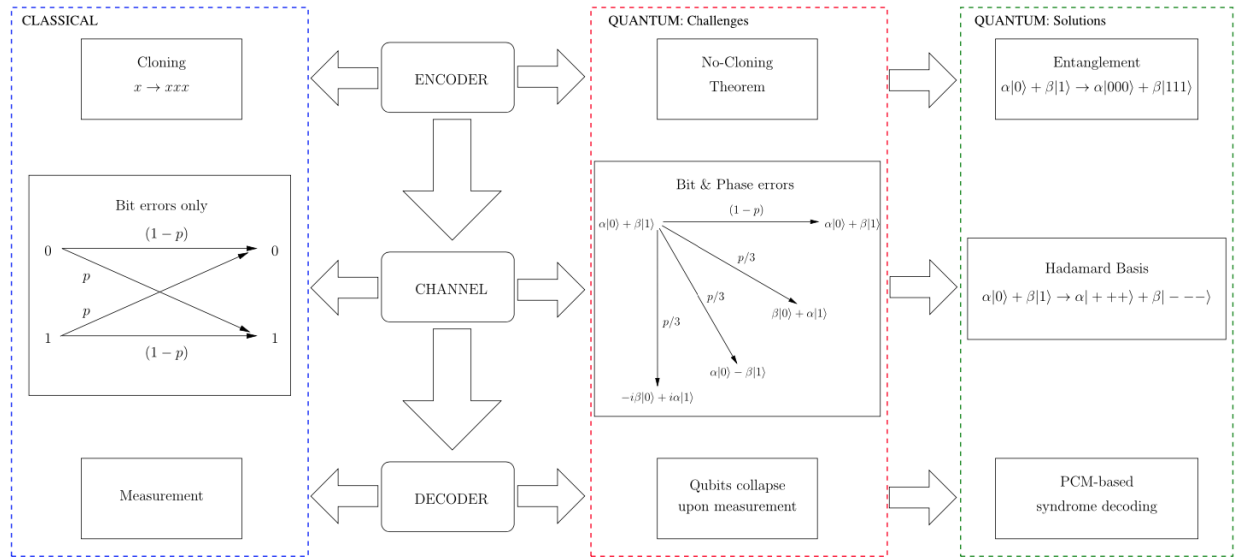


Figura 3.9: Transizione dal dominio classico a quello quantistico dei codici per la correzione degli errori [11].

misurati senza perturbare il loro stato quantistico sovrapposto. Pertanto, i codici quantistici utilizzano la decodifica della sindrome basata sulla PCM, cioè valutano i pattern di errore, indotti dal canale, senza in realtà osservare i qubit ricevuti.

Capitolo 4

Formalismo Stabilizzatore

La famiglia dei codici stabilizzatori quantistici (QSC - Quantum Stabilizer Code) si poggia sugli stessi principi di progettazione dei codici a ripetizione trattati nel Capitolo 3. In particolare, i QSCs si fondano sulla decodifica della sindrome basata sulla PCM dei codici classici a blocco lineari; quindi, dopo aver trovato l'errore indotto dal canale, si misurano i qubit della sindrome ausiliaria, invece che osservare i qubit ricevuti. Intuitivamente, il formalismo stabilizzatore [64], [65] può essere interpretato come il duale quantistico del paradigma classico di codifica lineare a blocco. Infatti, molti codici classici sfruttano la stessa infrastruttura di base dei codici a blocco lineari. Di conseguenza, il formalismo stabilizzatore fornisce un quadro teorico generale per la progettazione delle versioni quantistiche dei codici classici noti. Nella Sezione 4.1 si forniscono approfondimenti sulla dualità tra i QSCs e i codici classici a blocco lineari, mentre nella Sezione 4.2, si discute la classificazione dei modelli di errore sia per i QSCs che per i suddetti codici classici.

4.1 Progettazione di codici tramite il formalismo stabilizzatore

La Fig. 4.1 mostra il modello di un sistema di comunicazione quantistica basato su un QSC.

Un codice $C(n, k)$ a blocco lineare classico codifica la parola di informazione x , costituita da k bit, in una parola di codice \bar{x} , di n simboli, utilizzando $(n - k)$ simboli di parità inizializzati a zero ($\mathbf{0}^{n-k}$), come segue:

$$C = \{\bar{x} = (x : \mathbf{0}^{n-k})V\}, \quad (4.1)$$

dove V è una matrice di codifica invertibile di dimensioni $(n \times n)$. Allo stesso modo, un QSC generico $\mathcal{C}[n, k]$ codifica una parola di informazione $|\psi\rangle$, costituita da k qubit (chiamati *qubit logici*), in una parola di codice $|\bar{\psi}\rangle$, costituita da n qubit (chiamati *qubit fisici*), con l'aiuto di $(n - k)$ qubit ausiliari (noti anche come *qubit ancilla*), come segue:

$$\mathcal{C} = \{|\bar{\psi}\rangle = \mathcal{V}(|\psi\rangle \otimes |\mathbf{0}_{n-k}\rangle)\}, \quad (4.2)$$

4.1 Progettazione di codici tramite il formalismo stabilizzatore 75

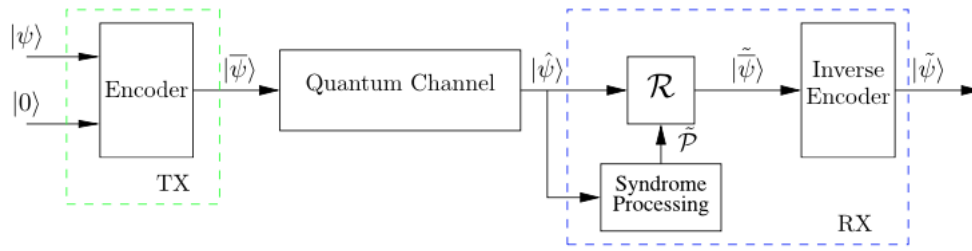


Figura 4.1: Schematico di un sistema di comunicazione quantistico che utilizza un QSC per la correzione degli errori [10].

dove \mathcal{V} è un codificatore a n qubit e \otimes denota il prodotto tensoriale.

N.B.: viene utilizzata la matrice V , invece che \mathbf{G} , per dualità con \mathcal{V} , del dominio quantistico. Si ricorda inoltre che si utilizzano le parentesi tonde (\cdot) per i codici classici, mentre le parentesi quadre $[\cdot]$ sono usate per i codici quantistici.

In particolare, i qubit ausiliari di un QSC sono analoghi ai bit di parità classici. I qubit codificati, che costituiscono la parola di codice $|\bar{\psi}\rangle$, vengono trasmessi nel canale quantistico di depolarizzazione (si veda a tal proposito la Sezione 1.3.3), che è caratterizzato dal vettore di errore del canale \mathcal{P} , considerando il caso generale ad n qubit. L'output potenzialmente affetto da errori, in uscita dal canale, che si indica con $|\hat{\psi}\rangle$, può quindi essere espresso come:

$$|\hat{\psi}\rangle = \mathcal{P} |\bar{\psi}\rangle. \quad (4.3)$$

Analogamente ai decodificatori dei codici a ripetizione a 3 qubit di tipo bit-flip e dei codici analoghi di tipo phase-flip, di Fig. 3.4 e Fig. 3.8, rispettivamente, il decoder di un QSC svolge un processo suddiviso in tre step per correggere gli errori di trasmissione, che include l'elaborazione della sindrome, il recupero degli errori (\mathcal{R}) e la fase di codifica inversa.

Pertanto, ora si riprende il processo di calcolo della sindrome per un codice a ripetizione a 3 qubit di tipo bit-flip, dal punto di vista del formalismo stabilizzatore. Si osserva infatti, dalla Fig. 3.4, che il primo bit della sindrome si calcola confrontando la base computazionale del primo e del secondo qubit, mentre il secondo bit della sindrome si ottiene confrontando, allo stesso modo, il primo ed il terzo qubit. Ciò equivale a misurare gli autovalori corrispondenti degli operatori di Pauli, a 3 qubit, $g_1 = \mathbf{ZZI}$ e $g_2 = \mathbf{ZIZ}$, che sono noti anche con il nome di *stabilizer generator*. In particolare gli stabilizer g_1 e g_2 sono relativi al codice a ripetizione a 3 qubit.

Osservazione Gli stabilizer generator basati sull'operatore \mathbf{Z} di Pauli sono utilizzati per confrontare i qubit, poiché sono in grado di rilevare errori di tipo bit-flip nelle basi computazionali. Pertanto, se i qubit che vengono confrontati hanno la stessa base computazionale, gli stabilizer generator basati sull'operatore \mathbf{Z} producono un autovalore pari a $+1$; altrimenti, se sono diversi, l'autovalore è -1 .

4.1 Progettazione di codici tramite il formalismo stabilizzatore 76

Per esempio, se la parola di codice ricevuta è valida, il che implica che sia il primo che il secondo qubit, così come il primo ed il terzo qubit, sono identici, come nell'eq.(3.16), si ha:

$$\begin{aligned} g_1[|\bar{\psi}\rangle] &= \mathbf{ZZI}(\alpha |000\rangle + \beta |111\rangle) = |\bar{\psi}\rangle; \\ g_2[|\bar{\psi}\rangle] &= \mathbf{ZIZ}(\alpha |000\rangle + \beta |111\rangle) = |\bar{\psi}\rangle. \end{aligned} \quad (4.4)$$

Dunque, quando viene ricevuta una parola di codice legittima, l'autovalore risultante è +1 sia per g_1 che per g_2 . Al contrario, se viene ricevuta una sequenza affetta da errori, cioè corrotta, del tipo $|\hat{\psi}\rangle = |100\rangle + \beta |011\rangle$, significa che sia il primo che il secondo qubit, così come il primo ed il terzo, sono diversi tra loro, come accade nell'eq.(3.17). Allora si ottiene:

$$\begin{aligned} g_1[|\hat{\psi}\rangle] &= \mathbf{ZZI}(\alpha |100\rangle + \beta |011\rangle) \\ &= -\alpha |100\rangle - \beta |011\rangle = -|\hat{\psi}\rangle; \\ g_2[|\hat{\psi}\rangle] &= \mathbf{ZIZ}(\alpha |100\rangle + \beta |011\rangle) \\ &= -\alpha |100\rangle - \beta |011\rangle = -|\hat{\psi}\rangle, \end{aligned} \quad (4.5)$$

dove sia g_1 che g_2 danno come autovalore -1 . Si ricorda poi, a partire dall'eq.(3.12) della Sezione 3.3, che la PCM di un codice classico a blocco lineare è progettata in modo da produrre un vettore di sindrome composto da tutti zeri (*all-zero*) per le parole di codice legittime, mentre si avrà un vettore di sindrome cosiddetto *non-zero*, in cui cioè possono anche comparire elementi diversi da zero, per le sequenze che non sono parole di codice. Queste particolari sequenze potranno essere corrette dal decodificatore solo se il numero di errori indotti dal canale rientra nelle capacità di correzione degli errori del codice. Il vettore della sindrome può risultare diverso da zero anche quando la sequenza ricevuta contiene un numero di errori che eccede le capacità correttive del codice; in questo caso però il decodificatore non è in grado di correggerli.

Allo stesso modo, gli stabilizer generator di un QSC devono essere progettati in modo da produrre un autovalore pari a +1 per le parole di codice legittime, ed un autovalore di -1 in presenza di errori. Quindi, continuando il confronto con il dominio classico, si nota che, come la PCM \mathbf{H} specifica completamente lo spazio di codice di un generico codice classico C , gli stabilizer generator definiscono lo spazio di codice di un QSC. Inoltre, si definisce *gruppo stabilizzatore completo* \mathcal{H} di un QSC l'insieme di tutti gli stabilizer generator ed i loro prodotti. Ad esempio, il gruppo stabilizzatore \mathcal{H} , del codice a ripetizione a 3 qubit di tipo bit-flip, è costituito sia dai generatori indipendenti g_1 e g_2 che dal loro prodotto, cioè \mathbf{IZZ} . Vale infatti:

$$g_1 \cdot g_2 = (\mathbf{ZZI}) \cdot (\mathbf{IZZ}) = \mathbf{Z} \cdot \mathbf{I} \quad \mathbf{Z} \cdot \mathbf{Z} \quad \mathbf{I} \cdot \mathbf{Z} = \mathbf{ZIZ}, \quad (4.6)$$

essendo $\mathbf{Z} \cdot \mathbf{Z} = \mathbf{I}$.

Gli autovalori +1 e -1 dell'eq.(4.5) sono mappati sui bit della sindrome classica 0 e 1, rispettivamente, quando gli operatori \mathbf{Z} costituenti sono realizzati

4.1 Progettazione di codici tramite il formalismo stabilizzatore 77

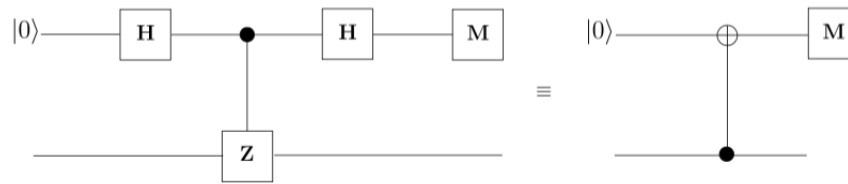


Figura 4.2: Circuito quantistico di misurazione dell'operatore \mathbf{Z} per la correzione di errori di tipo bit-flip [116], [12].

Tabella 4.1: Errori di inversione di bit a singolo qubit con gli autovalori associati per il codice a ripetizione a 3 qubit di tipo bit-flip [12].

$ \hat{\psi}\rangle = \mathcal{P} \bar{\psi}\rangle$	$g_1 \hat{\psi}\rangle$	$g_2 \hat{\psi}\rangle$	Syndrome (s)	$\hat{\mathcal{P}}$
$\alpha 000\rangle + \beta 111\rangle$	+1	+1	(00)	III
$\alpha 100\rangle + \beta 011\rangle$	-1	-1	(11)	XII
$\alpha 010\rangle + \beta 101\rangle$	-1	+1	(10)	IXI
$\alpha 001\rangle + \beta 110\rangle$	+1	-1	(01)	IIX

utilizzando il circuito quantistico di Fig. 4.2. Nell'immagine sono presenti due circuiti equivalenti, ed in particolare la rappresentazione a sinistra è più esplicativa, mentre quella a destra è più compatta e quindi più orientata all'implementazione [116]. In entrambi i circuiti della Fig. 4.2, il qubit superiore $|0\rangle$, rappresentato graficamente dalla linea posizionata in alto, è il qubit ausiliario utilizzato per calcolare la sindrome, mentre il qubit rappresentato dalla linea posta in basso è quello codificato, sottoposto all'azione dell'operatore \mathbf{Z} . Le sindromi risultanti sono elencate nella Tabella 4.1, insieme agli errori di tipo bit-flip associati al singolo qubit, agli autovalori e al modello di errore stimato $\hat{\mathcal{P}}$, che può essere calcolato utilizzando il metodo di decodifica della sindrome. Si consideri poi che vale $g_1 = \mathbf{ZZI}$ e $g_2 = \mathbf{ZIZ}$.

Analogamente al codice a ripetizione a 3 qubit di tipo bit-flip, la parola di codice nel caso di un codice a ripetizione a 3 qubit di tipo phase-flip è caratterizzata dagli stabilizer generator $g_1 = \mathbf{XXI}$ e $g_2 = \mathbf{XIX}$. Si può notare dunque che, mentre gli stabilizer generator analizzati precedentemente erano basati su operatori \mathbf{Z} di Pauli ed erano utilizzati per il rilevamento di errori di inversione di bit, gli stabilizer generator che si utilizzeranno ora sono basati su operatori \mathbf{X} di Pauli e sono invocati per confrontare i qubit nella base di Hadamard, poiché sono in grado di rilevarne gli errori, cioè, riescono a "scovare" errori di tipo phase-flip. Gli operatori \mathbf{X} associati possono essere realizzati utilizzando il circuito di Fig. 4.3. Anche questa volta, nell'immagine sono presenti due circuiti equivalenti, ed in particolare la rappresentazione a sinistra è più esplicativa dal punto di vista concettuale, mentre quella a destra è più adatta all'implementazione [116]. In entrambi i circuiti della Fig. 4.3, il qubit superiore $|0\rangle$, rappresentato graficamente dalla linea posizionata in alto, è il qubit ausiliario utilizzato per il calcolo

4.1 Progettazione di codici tramite il formalismo stabilizzatore 78

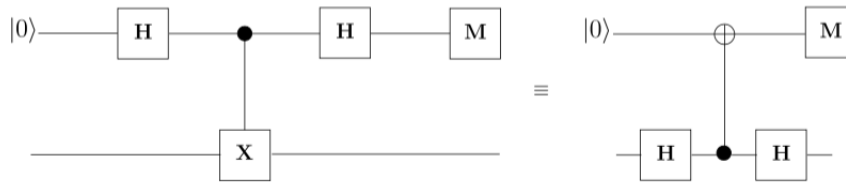


Figura 4.3: Circuito quantistico di misurazione dell'operatore \mathbf{X} per la correzione di errori di tipo phase-flip [116], [12].

della sindrome, mentre il qubit rappresentato dalla linea posta in basso è il qubit codificato sottoposto all'azione dell'operatore \mathbf{X} .

Si ricorda, dal Capitolo 3, che le parole di codice di Shor consistono di tre blocchi da 3 qubit ognuno, sicché i tre qubit all'interno di ogni blocco costituiscono la parola di codice, di un codice a ripetizione a 3 qubit di tipo bit-flip. Di conseguenza, gli errori di inversione del bit possono essere rilevati applicando, in modo indipendente, gli stabilizer generator, del codice a ripetizione a 3 qubit di tipo bit-flip, ai tre blocchi da 3 qubit ciascuno. Questo equivale a confrontare i tre qubit all'interno di ogni blocco. Ne risultano i seguenti sei stabilizer generator:

$$\begin{aligned}
 g_1 &= \mathbf{ZZIIIIII}; \\
 g_2 &= \mathbf{ZIZIIIII}; \\
 g_3 &= \mathbf{IIIZZIII}; \\
 g_4 &= \mathbf{IIIZIZII}; \\
 g_5 &= \mathbf{IIIIIZZI}; \\
 g_6 &= \mathbf{IIIIIZIZ},
 \end{aligned} \tag{4.7}$$

che permettono di rilevare i singoli errori di bit-flip che si verificano in ogni blocco da 3 qubit ciascuno. Al contrario, gli errori di inversione di fase (phase-flip) possono essere rilevati confrontando tra loro i blocchi stessi, utilizzando gli operatori \mathbf{X} di Pauli. In particolare, le informazioni riguardanti la fase di un blocco da 3 qubit vengono estratte applicando l'operatore \mathbf{XXX} ai tre qubit. Dunque, il codice da 9 qubit di Shor, che consiste in tre blocchi da 3 qubit ciascuno, può essere implementato usando i seguenti due stabilizer generator:

$$\begin{aligned}
 g_7 &= \mathbf{XXXXXXIII}; \\
 g_8 &= \mathbf{XXXIIIXXX},
 \end{aligned} \tag{4.8}$$

dove g_7 confronta la fase dei primi due blocchi, mentre g_8 confronta la fase del primo e del terzo blocco.

Sulla base delle considerazioni di cui sopra, il processo di decodifica in 3 step, di Fig. 4.1, può essere generalizzato come segue:

4.1 Progettazione di codici tramite il formalismo stabilizzatore 79

1) *Processo di calcolo della sindrome*: mentre lo spazio di codice C di un codice classico a blocco lineare è definito da una PCM \mathbf{H} che ha $(n - k)$ righe indipendenti, lo spazio di codice \mathcal{C} , associato ad un QSC, è descritto da $(n - k)$ operatori g_i indipendenti, per codici da n qubit, ognuno composto da n matrici di Pauli, con $1 \leq i \leq (n - k)$. Questi sono generalmente chiamati *stabilizer generator* oppure, più brevemente, *stabilizzatori* (stabilizer) o generatori di Pauli (Pauli generators). In particolare, gli stabilizzatori sono operatori *unici*, cioè non perturbano lo stato delle parole di codice legittime, infatti danno come risultato un autovalore pari a $+1$. Invece, gli stabilizzatori producono un autovalore di -1 per le parole di codice corrotte. Questo è equivalente ai valori 0 e 1 nella sindrome classica, rispettivamente, vale a dire gli elementi del vettore di sindrome espresso dall'eq.(3.13). In alternativa, si può dire che l'autovalore risultante è $+1$, quando l'errore \mathcal{P} , indotto dal canale, commuta con lo stabilizzatore g_i , mentre è -1 , quando l'errore non commuta con g_i . Tale proprietà può essere matematicamente espressa nel seguente modo:

$$g_i |\hat{\psi}\rangle = \begin{cases} |\bar{\psi}\rangle, & \text{se } g_i \mathcal{P} = \mathcal{P} g_i \\ -|\bar{\psi}\rangle, & \text{se } g_i \mathcal{P} = -\mathcal{P} g_i, \end{cases} \quad (4.9)$$

dove $|\hat{\psi}\rangle = \mathcal{P} |\bar{\psi}\rangle$. Gli autovalori risultanti possono essere mappati sulla sindrome classica s , utilizzando i circuiti quantistici di Fig. 4.2 e di Fig. 4.3. Quindi, l'insieme degli stabilizzatori costituisce la controparte quantistica della PCM classica. Tuttavia, gli stabilizzatori devono presentare la proprietà della *commutatività addizionale*, chiamato per semplicità criterio di commutatività. Per questa proprietà gli stabilizzatori devono formare coppie i cui due elementi sono mutuamente commutativi. In particolare, per una coppia di stabilizzatori g_1 e g_2 , si ha:

$$g_1 g_2 |\bar{\psi}\rangle = g_1 |\bar{\psi}\rangle = |\bar{\psi}\rangle, \quad (4.10)$$

ed allo stesso modo, vale:

$$g_2 g_1 |\bar{\psi}\rangle = g_2 |\bar{\psi}\rangle = |\bar{\psi}\rangle. \quad (4.11)$$

Dunque, risulta evidente che il criterio di commutatività non esiste nel dominio classico. Inoltre, il gruppo stabilizzatore associato \mathcal{H} , che contiene $(n - k)$ stabilizzatori g_i così come tutti i possibili prodotti tra i g_i , forma un sottogruppo abeliano \mathcal{G}_n .

Approfondimento: Gruppo di Abel In matematica ed in particolare in algebra astratta, un gruppo si dice *abeliano* (o *commutativo*) se è un gruppo la cui operazione binaria interna gode della proprietà commutativa. Dunque il gruppo (\mathcal{G}_n, \cdot) è abeliano se:

$$g_1 \cdot g_2 = g_2 \cdot g_1, \quad \forall g_1, g_2 \in \mathcal{G}_n, \quad (4.12)$$

dove \cdot sta ad indicare l'operazione binaria interna, commutativa, che in questo caso è il prodotto. Come spesso accade, il nome deriva dal matematico

4.1 Progettazione di codici tramite il formalismo stabilizzatore 80

norvegese Niels Henrik Abel, che per primo teorizzò questi principi [91], sulla scia del lavoro di Evariste Galois.

Il decoder di Fig. 4.1 elabora la sindrome della sequenza ricevuta $|\hat{\psi}\rangle$ attraverso gli stabilizzatori associati, che vengono implementati utilizzando dei qubit ausiliari. Analogamente ai decodificatori dei codici a ripetizione a 3 qubit ad inversione di bit e ad inversione di fase, visti in Fig. 3.4 e in Fig. 3.8, rispettivamente, i qubit ausiliari collassano nelle sindromi classiche al momento della misurazione, mappando quindi gli autovalori $+1$ e -1 sui bit classici 0 e 1, rispettivamente. I bit di sindrome risultanti vengono poi alimentati da una LUT o da un decodificatore classico della sindrome, basato su una PCM, per stimare il vettore di errore del canale $\tilde{\mathcal{P}}$ (questo argomento sarà analizzato più approfonditamente nel prossimo capitolo).

2) *Error Recovery* (\mathcal{R}): Il blocco di recupero degli errori (\mathcal{R}) di Fig. 4.1, ripristina il codice $|\tilde{\psi}\rangle$, potenzialmente privo di errori, utilizzando il modello di errore stimato $\tilde{\mathcal{P}}$. Naturalmente, se il numero di errori supera la capacità di correzione degli errori del codice, il processo di recupero risulta imperfetto. Tuttavia, in quest'ultimo caso, tale azione correttiva difettosa produrrà, in realtà, più errori di quanti ce ne fossero in origine.

3) *Codificatore inverso*: Infine, il codificatore inverso di Fig. 4.1 mappa la parola di codice recuperata $|\tilde{\psi}\rangle$ sulla parola di informazione stimata trasmessa $|\tilde{\psi}\rangle$. In particolare, mentre un codificatore tradizionale mappa le sequenze di informazione sulle parole di codice, un codificatore inverso lavora nella direzione inversa, mappando quindi le parole di codice sulle sequenze di informazione.

Si ricorda, dalle eq.(4.10) e (4.11), che gli $(n - k)$ stabilizer generator g_i di un QSC sono sempre commutativi tra loro. Ciò implica che gli operatori costitutivi, cioè le matrici di Pauli \mathbf{X} , \mathbf{Y} e \mathbf{Z} , devono essere selezionati in modo che tutti gli stabilizzatori risultanti commutino. Esplicitamente, si osserva che gli operatori (detti *non di identità*) \mathbf{X} , \mathbf{Y} e \mathbf{Z} sono intrinsecamente non commutativi tra loro. Ad esempio, si ha:

$$\mathbf{X}\mathbf{Y} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix} = i\mathbf{Z}, \quad (4.13)$$

allo stesso modo, vale:

$$\mathbf{Y}\mathbf{X} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} -i & 0 \\ 0 & i \end{pmatrix} = -i\mathbf{Z}. \quad (4.14)$$

Ciò implica che gli operatori $\mathbf{X}\mathbf{Y}$ e $\mathbf{Y}\mathbf{X}$ sono anti-commutativi, cioè si ottiene:

$$\mathbf{X}\mathbf{Y} = -\mathbf{Y}\mathbf{X}. \quad (4.15)$$

In altre parole, la proprietà di anti-commutatività si raggiunge quando, dati due operatori, il loro prodotto non ordinato dà sempre lo stesso risultato, a

meno del segno. Analogamente, si può facilmente dimostrare che:

$$\begin{aligned} \mathbf{YZ} = i\mathbf{X}, \quad \mathbf{ZY} = -i\mathbf{X} &\Rightarrow \mathbf{YZ} = -\mathbf{ZY}; \\ \mathbf{ZX} = i\mathbf{Y}, \quad \mathbf{XZ} = -i\mathbf{Y} &\Rightarrow \mathbf{ZX} = -\mathbf{XZ}. \end{aligned} \quad (4.16)$$

A causa della natura non commutativa degli operatori di Pauli \mathbf{X} , \mathbf{Y} e \mathbf{Z} , gli stabilizzatori devono essere progettati in modo che ci sia un numero pari di indici con operatori non di identità diversi tra loro.

Ad esempio, gli operatori di Pauli a 3 qubit \mathbf{ZZI} e \mathbf{XYZ} commutano, poiché sono caratterizzati da due indici con operatori non di identità diversi tra loro. In particolare, si tratta delle prime due posizioni, cioè degli indici $i = 1, 2$, in cui si ha: $\mathbf{Z} \neq \mathbf{X}$ e $\mathbf{Z} \neq \mathbf{Y}$. La terza posizione non viene conteggiata in questa valutazione, sebbene $\mathbf{I} \neq \mathbf{Z}$ (in questo caso sarebbero tre le disuguaglianze), poiché \mathbf{I} è la matrice identità.

Al contrario, gli operatori \mathbf{ZZI} e \mathbf{YZI} sono anti-commutativi, dato che è presente un unico indice, corrispondente alla prima posizione, in cui sono presenti operatori non di identità diversi: $\mathbf{Z} \neq \mathbf{Y}$. Mentre, nelle altre due posizioni si verifica l'uguaglianza tra le matrici: $\mathbf{Z} = \mathbf{Z}$ e $\mathbf{I} = \mathbf{I}$, sebbene quest'ultima non rientrasse nel conteggio.

In definitiva, se due stabilizer generator non commutano, essi risultano giocoforza anti-commutativi, poiché le singole matrici di Pauli di cui sono composti saranno sempre commutative o anti-commutative.

4.2 Classificazione dei modelli di errore

Sulla base delle considerazioni fatte nella sezione precedente, si può concludere che gli stabilizer generator svolgono lo stesso ruolo, nella correzione degli errori quantistici, della PCM classica \mathbf{H} , nella correzione degli errori nel dominio classico. Infatti, analogamente alla PCM, gli stabilizzatori producono dei bit di sindrome, che a loro volta servono per stimare gli errori dei canali quantistici. Più in particolare, l'insieme degli errori di un codice classico a blocco lineare C , avente una PCM \mathbf{H} , può essere oggetto della seguente classificazione:

1) *Modelli di errore rilevati*: Questi modelli di errore producono una sindrome non triviale, cioè $e\mathbf{H}^T \neq 0$, che potrebbe essere corretta dal codice.

2) *Modelli di errore non rilevati*: Questa classe di modelli di errore si traduce in una sindrome banale, cioè $e\mathbf{H}^T = 0$, che non può essere rilevata dal codice. Più specificamente, un errore non rivelato mappa la parola di codice trasmessa su un'altra parola di codice valida. Poiché il codice risultante si trova ancora nello spazio di codice C , cioè è considerato legittimo, non si innesca una sindrome di tipo non zero. Questi modelli di errore non rilevati derivano dalla distanza minima limitata del codice.

Analogamente ai modelli di errore classici succitati, i modelli di errore rilevati nel dominio quantistico sono anti-commutativi rispetto ad almeno uno

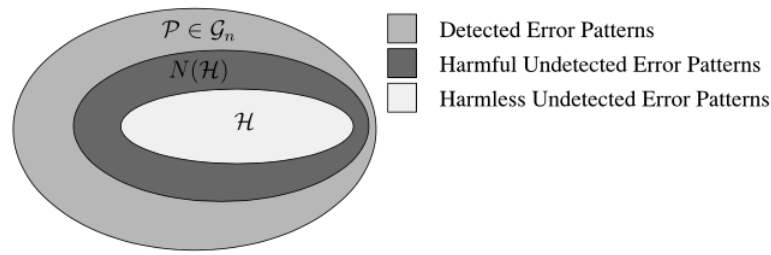


Figura 4.4: Classificazione dei modelli di errore per i codici stabilizzatori [12].

stabilizer generator, il che si traduce in una sindrome non triviale. Allo stesso modo, i modelli di errore quantistici non rilevati commutano con tutti gli stabilizer generator, producendo una sindrome completamente costituita da zeri. Questo insieme di modelli di errore di commutazione è noto anche come *centralizzatore* (o *normalizzatore*) del codice stabilizzatore, avente il gruppo stabilizzatore \mathcal{H} , ed è indicato con $C(\mathcal{H})$ (o $N(\mathcal{H})$). In particolare, il centralizzatore di un $[n, k]$ QSC è un sottospazio doppio costituito da n -uple di errori di Pauli $\mathcal{P} \in \mathcal{G}_n$, che sono ortogonali a tutti gli elementi del gruppo stabilizzatore \mathcal{H} . Inoltre, poiché \mathcal{H} è un gruppo abeliano, costituito da stabilizer generator mutuamente ortogonali, esso è contenuto nel centralizzatore; cioè si ha: $\mathcal{H} \subset N(\mathcal{H})$. Dato che gli stabilizer generator non modificano lo stato delle parole di codice valide, si può osservare che gli errori appartenenti al gruppo degli stabilizzatori, cioè quelli per cui vale: $\mathcal{P} \in \mathcal{H}$, non corrompono le parole di codice trasmesse e quindi possono essere classificati come modelli di errore innocui e non rilevati. Questa classe di errori non ha una controparte classica. Al contrario, i modelli di errore, che si trovano nel sottospazio $N(\mathcal{H}) \setminus \mathcal{H}$, appartengono alla categoria degli errori non rilevati dannosi, che mappano una parola di codice valida su un'altra. Quindi, come descritto nella Fig. 4.4, i modelli di errore quantistico possono essere classificati come segue:

- 1) *Modelli di errore rilevati:* Questi modelli di errore ricadono al di fuori del sottospazio normalizzatore, cioè soddisfano la seguente relazione: $\mathcal{P} \in \mathcal{G}_n \setminus N(\mathcal{H})$.
- 2) *Modelli di errore non rilevati dannosi:* Questa classe di modelli di errore è definita come $N(\mathcal{H}) \setminus \mathcal{H}$.
- 3) *Modelli di errore non rilevati innocui:* Questi errori cadono nel gruppo stabilizzatore \mathcal{H} .

La classe dei modelli di errore non rilevati innocui rende i codici quantistici *degeneri* [122]. In particolare, i modelli di errore \mathcal{P} e $\mathcal{P}' = g_i \mathcal{P}$ si dicono degeneri, poiché differiscono solo per gli elementi del gruppo stabilizzatore, che sono innocui. Di conseguenza, sia \mathcal{P} che \mathcal{P}' producono la

stessa uscita, come mostrato di seguito:

$$\mathcal{P}'[|\bar{\psi}\rangle] = g_i \mathcal{P}[|\bar{\psi}\rangle] = \mathcal{P} g_i[|\bar{\psi}\rangle]. \quad (4.17)$$

Sapendo poi che $g_i[|\bar{\psi}\rangle] = [|\bar{\psi}\rangle]$, si ottiene:

$$\mathcal{P}'[|\bar{\psi}\rangle] = \mathcal{P}[|\bar{\psi}\rangle]. \quad (4.18)$$

Attraverso queste considerazioni di natura matematica, si può osservare che i modelli di errore degeneri possono essere corretti dalla stessa operazione di recupero.

Si considerino i modelli di errore $\mathcal{P} = \mathbf{IIX}$ e $\mathcal{P}' = g_1 \mathcal{P} = \mathbf{ZZX}$, dove g_1 è lo stabilizzatore del codice a ripetizione a 3 qubit di tipo bit-flip definito nell'eq.(4.1). Quando questi modelli di errore vengono applicati al codice legittimo dell'eq.(3.11), si ottiene:

$$\begin{aligned} \mathbf{IIX}[\alpha |000\rangle + \beta |111\rangle] &= \alpha |001\rangle + \beta |110\rangle, \\ \mathbf{ZZX}[\alpha |000\rangle + \beta |111\rangle] &= \alpha |001\rangle + \beta |110\rangle. \end{aligned} \quad (4.19)$$

Dunque, \mathcal{P} e \mathcal{P}' sono considerati errori degeneri, dal momento che entrambi i modelli di errore producono la stessa parola di codice corrotta. Inoltre, la degenerazione aumenta la capacità raggiungibile dal canale, poiché le parole di codice non possono essere corrotte dagli schemi di errore innocui non rilevati. Si conclude pertanto che l'impatto delle degradazioni, che può determinare un generico canale quantistico è ridotto. Allo stesso modo, si può dire che la degenerazione consente ad un codice quantistico di contenere più informazioni rispetto al progetto classico corrispondente, poiché, in questo caso, si può operare ad un tasso di codifica più elevato.

Capitolo 5

Isomorfismo dal dominio quantistico a quello classico

Sulla base della dualità tra i QSC e i codici classici a blocco lineari, stabilita nel Capitolo 4, in questa sede si presenta l'*isomorfismo* tra questi due domini, che permette di costruire le versioni quantistiche dei codici classici conosciuti. In generale per isomorfismo si intende la corrispondenza biunivoca fra gli elementi di due insiemi; in questo caso si allude alla categoria dei QSC e dei codici classici a blocco lineari (gli insiemi) e ai codici che ne fanno parte (gli elementi).

In particolare, i QSC possono essere progettati a partire, tra gli altri, dai codici classici binari e quaternari, utilizzando le leggi di associazione (mapping) della Tabella 5.1, che permettono di passare dal dominio quantistico a quello classico, come descritto nelle successive Sezioni 5.1 e 5.2, rispettivamente. Inoltre, questo isomorfismo permette anche di utilizzare le classiche procedure di decodifica del vettore di sindrome, basato sulla PCM, per decodificare i QSC.

Osservazione Nel seguito della trattazione, il dominio quantistico viene rappresentato dal dominio di Pauli che, come osservato nella Sottosezione 1.3.3, è una sua possibile declinazione. Pertanto verrà descritto l'isomorfismo dal dominio di Pauli al dominio classico binario e quaternario.

5.1 Isomorfismo dal dominio di Pauli al dominio binario

Si ricorda, dal Capitolo 4, che gli stabilizzatori costituiscono le controparti quantistiche della PCM classica. Sulla base di questa dualità, i QSC possono essere descritti utilizzando una PCM binaria equivalente, che a sua volta aiuta a progettare i codici quantistici, a partire dalle famiglie di codici classici già esistenti. In particolare, i QSC possono essere caratterizzati completamente, nel formalismo binario, da una PCM \mathbf{H} binaria equivalente, derivata dagli stabilizer generator associati. Le righe della matrice \mathbf{H} corrispondono agli stabilizzatori, mentre gli operatori di Pauli \mathbf{I} , \mathbf{X} , \mathbf{Y} e \mathbf{Z} ,

Tabella 5.1: Isomorfismo dal dominio quantistico a quello classico [12].

Pauli	$(\mathbb{F}_2)^2$	$\mathbf{GF}(4)$
I	00	0
X	01	1
Y	11	$\bar{\omega}$
Z	10	ω
Moltiplicazione	Somma bit-a-bit	Addizione
Commutatività	Prodotto Simplettico	Traccia del prodotto scalare

che costituiscono gli stabilizzatori, sono mappati su una coppia di simboli binari, come segue:

$$\mathbf{I} \rightarrow (00), \quad \mathbf{X} \rightarrow (01), \quad \mathbf{Z} \rightarrow (10), \quad \mathbf{Y} \rightarrow (11). \quad (5.1)$$

A tal proposito si osservi ancora la Tabella 5.1, in cui con la notazione $(\mathbb{F}_2)^2$ si intende il dominio classico nel caso di due elementi. Quindi un 1 al primo indice rappresenta un operatore **Z**, mentre un 1 in seconda posizione rappresenta un operatore **X**. Infatti la matrice identità **I** è priva di 1, mentre l'operatore **Y** è composto da due 1 poiché combina gli effetti di **Z** e **X**, cioè bit-and-phase-flip. La PCM **H**, risultante dal mapping dal dominio di Pauli al dominio classico binario dell'eq.(5.1), cioè la legge di associazione che permette di passare dal dominio quantistico al dominio classico binario, può anche essere espressa come:

$$\mathbf{H} = (\mathbf{H}_z | \mathbf{H}_x), \quad (5.2)$$

dove \mathbf{H}_z e \mathbf{H}_x sono matrici binarie di dimensione $(n-k) \times n$, che corrispondono agli operatori **Z** e **X**, rispettivamente. Ciò significa che la matrice \mathbf{H}_z comprenderà solo gli 1 degli operatori **Z** e la matrice \mathbf{H}_x solo i valori 1 degli operatori **X**.

Si ricorda, che il codice a ripetizione a 3 qubit di tipo bit-flip si basa sugli stabilizzatori $g_1 = \mathbf{ZZI}$ e $g_2 = \mathbf{ZIZ}$. Di conseguenza, la PCM **H** ad esso associata è data da:

$$\mathbf{H} = \left(\begin{array}{ccc|ccc} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{array} \right), \quad (5.3)$$

dove \mathbf{H}_x è una matrice all-zero, cioè completamente composta da zeri, poiché g_1 e g_2 non contengono operatori di Pauli **X**. Inoltre, si può osservare che la matrice \mathbf{H}_z dell'eq.(5.3) è identica alla PCM **H** del codice a ripetizione classico, data dall'eq.(3.14). Quindi, entrambe producono vettori della sindrome identici, come già visto nella Tabella 3.1 e nella Tabella 4.1. Di seguito, a titolo dimostrativo, verranno illustrati i passaggi che portano alla costruzione della matrice **H** di cui sopra.

Come detto, dagli stabilizzatori g_1 e g_2 si ricavano le righe della matrice **H**. In particolare, nel dominio classico binario, si avrà:

$$\begin{aligned} g_1 &= 10\ 10\ 00, \\ g_2 &= 10\ 00\ 10, \end{aligned} \quad (5.4)$$

Tabella 5.2: Addizione in $(\mathbb{F}_2)^2$ [12].

+	00	01	10	11
00	00	01	10	11
01	01	00	11	10
10	10	11	00	01
11	11	10	01	00

partendo da:

$$\begin{aligned} g_1 &= \mathbf{ZZI}, \\ g_2 &= \mathbf{ZIZ}, \end{aligned} \quad (5.5)$$

in ragione del mapping dell'eq.(5.1), per cui $\mathbf{Z} = 10$ e $\mathbf{I} = 00$. Ora, per comporre le righe della matrice \mathbf{H} bisogna ricombinare le posizioni dei simboli binari negli stabilizzatori. In particolare, riferendosi alla prima riga, l'elemento della matrice di posizione $\mathbf{H}_{1,1}$ (o, equivalentemente $\mathbf{H}_{z_{1,1}}$) corrisponde con il primo indice del primo operatore di Pauli di g_1 , cioè 1. Così, il secondo bit della medesima matrice di Pauli, cioè 0, va posizionato in $\mathbf{H}_{1,4}$ (o, equivalentemente $\mathbf{H}_{x_{1,1}}$). Continuando la costruzione, si passa al secondo operatore di Pauli di g_1 , che è tra l'altro uguale al primo e cioè $\mathbf{Z} = 10$. Il primo elemento, un 1, va posizionato in $\mathbf{H}_{1,2}$ (o $\mathbf{H}_{z_{1,2}}$), mentre il secondo, uno 0, va posto in $\mathbf{H}_{1,5}$ (o $\mathbf{H}_{x_{1,2}}$). Infine, il terzo ed ultimo operatore di Pauli di g_1 , cioè l'identità $\mathbf{I} = 00$, avrà i suoi due elementi binari posizionati in $\mathbf{H}_{1,3}$ (o $\mathbf{H}_{z_{1,3}}$) e $\mathbf{H}_{1,6}$ (o $\mathbf{H}_{x_{1,3}}$), rispettivamente. Analogamente, si compone la seconda riga della matrice \mathbf{H} , a partire dallo stabilizzatore g_2 .

In altre parole, si supponga di usare la seguente notazione per indicare gli elementi binari degli stabilizer generator:

$$\begin{aligned} g_1 &= g_{1,1}g_{1,2}g_{1,3}g_{1,4}g_{1,5}g_{1,6}; \\ g_2 &= g_{2,1}g_{2,2}g_{2,3}g_{2,4}g_{2,5}g_{2,6}, \end{aligned} \quad (5.6)$$

avendo comunque a che fare con un codice a ripetizione a 3 qubit. Allora, la matrice associata \mathbf{H} sarebbe così composta:

$$\mathbf{H} = \left(\begin{array}{ccc|ccc} g_{1,1} & g_{1,3} & g_{1,5} & g_{1,2} & g_{1,4} & g_{1,6} \\ g_{2,1} & g_{2,3} & g_{2,5} & g_{2,2} & g_{2,4} & g_{2,6} \end{array} \right). \quad (5.7)$$

Allo stesso modo, la PCM del codice a ripetizione a 3 qubit di tipo phase-flip, è:

$$\mathbf{H} = \left(\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{array} \right), \quad (5.8)$$

dove si ha $g_1 = \mathbf{XXI}$ e $g_2 = \mathbf{XIX}$, mentre la matrice \mathbf{H} per il codice di Shor con rate $1/9$ di [145] è data dalla seguente relazione:

$$\mathbf{H} = \left(\begin{array}{cccccccc|cccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array} \right). \quad (5.9)$$

Quindi, un QSC con $[n, k]$ qubit, avente $(n - k)$ stabilizzatori, può essere caratterizzato da una PCM binaria di dimensioni $(n - k) \times 2n$.

Dunque, se si va ad estendere la generalizzazione proposta dall'eq.(5.6) e dall'eq.(5.7) al caso con $[n, k]$ qubit e $n - k$ stabilizzatori, di cui sopra, si trovano le seguenti relazioni:

$$\begin{aligned} g_1 &= g_{1,1_1} g_{1,1_2} g_{1,2_1} g_{1,2_2} \cdots g_{1,n_1} g_{1,n_2}; \\ g_2 &= g_{2,1_1} g_{2,1_2} g_{2,2_1} g_{2,2_2} \cdots g_{2,n_1} g_{2,n_2}, \\ &\vdots \\ g_{n-k} &= g_{n-k,1_1} g_{n-k,1_2} g_{n-k,2_1} g_{n-k,2_2} \cdots g_{n-k,n_1} g_{n-k,n_2}, \end{aligned} \quad (5.10)$$

dove il generico elemento g_{j,i_t} è caratterizzato dal pedice $j = 1, 2, \dots, n - k$ che indica la numerosità degli stabilizzatori (e delle righe della matrice \mathbf{H}), mentre il pedice $i = 1, 2, \dots, n$ rappresenta la quantità di operatori di Pauli che compongono lo stabilizer, cioè il numero di qubit codificati n . Infine il sottopedice $t = 1, 2$ indica l'elemento binario della matrice di Pauli che si sta designando, il primo o il secondo, rispettando il mapping dell'eq.(5.1). In alternativa, si possono anche sostituire i valori 1,2 del pedice t con z e x , in base alla regione della matrice designata, quindi si avrebbe $t = z, x$.

Alla luce di queste considerazioni, la matrice \mathbf{H} può essere espressa come segue:

$$\mathbf{H} = \left(\begin{array}{cccc|cccc} g_{1,1_z} & g_{1,2_z} & \cdots & g_{1,n_z} & g_{1,1_x} & g_{1,2_x} & \cdots & g_{1,n_x} \\ g_{2,1_z} & g_{2,2_z} & \cdots & g_{2,n_z} & g_{2,1_x} & g_{2,2_x} & \cdots & g_{2,n_x} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{n-k,1_z} & g_{n-k,2_z} & \cdots & g_{n-k,n_z} & g_{n-k,1_x} & g_{n-k,2_x} & \cdots & g_{n-k,n_x} \end{array} \right), \quad (5.11)$$

come previsto peraltro dall'eq.(5.2).

Inoltre, il rate di codifica classico equivalente R_c può essere determinato

come segue:

$$\begin{aligned} R_c &= \frac{2n - (n - k)}{2n} = \frac{n + k}{2n} \\ &= \frac{1}{2} \left(1 + \frac{k}{n} \right) = \frac{1}{2} (1 + R_Q), \end{aligned} \quad (5.12)$$

dove R_Q è il rate di codifica quantistica associato. Basandosi sull'eq.(5.12), il rate di codifica classico equivalente del codice quantistico a ripetizione a 3 qubit, con rate $1/3$, è pari a $2/3$, mentre quella del codice di Shor, con rate $1/9$, è di $5/9$.

Il formalismo binario dell'eq.(5.1) trasforma la moltiplicazione degli operatori di Pauli nella somma bit a bit (bit-wise addition) della corrispondente rappresentazione binaria. Per esempio, moltiplicare l'insieme degli operatori di Pauli $\{\mathbf{I}, \mathbf{X}, \mathbf{Z}, \mathbf{Y}\}$ per \mathbf{X} è equivalente alla seconda colonna della Tabella 5.2, se gli operatori di Pauli sono mappati su $(\mathbb{F}_2)^2$ secondo le espressioni dell'eq.(5.1). Allo stesso modo, la proprietà commutativa degli stabilizzatori, nel formalismo di Pauli, implica che le righe della matrice di parità \mathbf{H} devono essere ortogonali tra loro, rispettando il prodotto simplettico (indicato anche come "twisted product") nel formalismo binario. In particolare, se l' i -esima riga di \mathbf{H} è indicata come $\mathbf{H}_i = (\mathbf{H}_{z_i} | \mathbf{H}_{x_i})$ seguendo la notazione dell'eq.(5.2), la commutatività degli stabilizzatori g_i e $g_{i'}$ si trasforma nel prodotto simplettico delle righe \mathbf{H}_i e $\mathbf{H}_{i'}$, che viene calcolato come segue:

$$\mathbf{H}_i \star \mathbf{H}_{i'} = (\mathbf{H}_{z_i} \cdot \mathbf{H}_{x_{i'}} + \mathbf{H}_{z_{i'}} \cdot \mathbf{H}_{x_i}) \bmod 2, \quad (5.13)$$

dove \star indica il prodotto simplettico. Il prodotto simplettico risultante produce un valore pari a zero, se il numero degli operatori non di identità (\mathbf{X} , \mathbf{Y} o \mathbf{Z}) diversi, presenti negli stabilizzatori g_i e $g_{i'}$, è pari. In tal caso, soddisfacendo il criterio della commutatività. Inoltre, poiché tutti gli stabilizzatori devono essere commutativi, il prodotto simplettico deve essere zero per tutte le righe di \mathbf{H} . Cioè la PCM \mathbf{H} deve soddisfare la seguente relazione:

$$\mathbf{H}_z \mathbf{H}_x^T + \mathbf{H}_x \mathbf{H}_z^T = \mathbf{0} \bmod 2. \quad (5.14)$$

Questo, a sua volta, implica che qualsiasi coppia di codici binari classici, caratterizzati dalle PCM \mathbf{H}_z e \mathbf{H}_x , che soddisfano la relazione sul prodotto simplettico dell'eq.(5.14), può essere utilizzata per la costruzione di un QSC valido.

Il prodotto simplettico dell'eq.(5.14) può anche essere sfruttato per il calcolo della sindrome di un QSC nel dominio binario, ad esempio durante la decodifica della sindrome basata sulla PCM. In particolare, l'isomorfismo dal dominio di Pauli a quello binario dell'eq.(5.1) trasforma un errore di Pauli da n qubit ($\mathcal{P} \in \mathcal{G}_n$) in un vettore di errore effettivo P di lunghezza $2n$, come si può osservare dalla Fig. 5.1. In altre parole, analogamente al caso della matrice \mathbf{H} nell'eq.(5.2), il vettore che rappresenta l'errore effettivo P può essere espresso come $P = (P_z | P_x)$, dove P_z e P_x sono legati agli errori di Pauli \mathbf{Z} e \mathbf{X} , rispettivamente. Più precisamente, un 1 all'indice

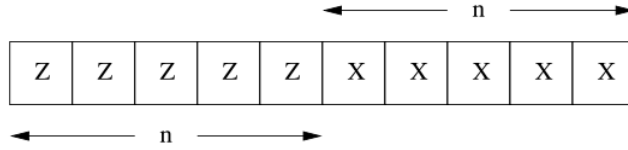


Figura 5.1: Rappresentazione dell'errore effettivo P , corrispondente all'errore di Pauli \mathcal{P} ad n qubit [12].

t -esimo di P_z denota un errore di Pauli \mathbf{Z} (phase-flip) sul qubit t -esimo, mentre un 1 all'indice t -esimo di P_x rappresenta l'occorrenza dell'errore di Pauli \mathbf{X} (bit-flip) sul qubit t -esimo. Allo stesso modo, l'errore di Pauli \mathbf{Y} (bit-and-phase-flip) sul t -esimo qubit produce un 1 all'indice t -esimo di P_z e di P_x . Infine, la sindrome di un QSC può essere calcolata nel formalismo binario utilizzando il prodotto simplettico e il vettore di errore effettivo P come segue:

$$s = \mathbf{H} \star P^T = (\mathbf{H}_z P_x^T + \mathbf{H}_x P_z^T) \bmod 2, \quad (5.15)$$

dove \mathbf{H}_z e \mathbf{H}_x sono utilizzati per correggere errori di bit-flip e di phase-flip, rispettivamente, come discusso in precedenza nel contesto dei codici a ripetizione a 3 qubit di tipo bit-flip e di tipo phase-flip, rispettivamente. La sindrome risultante assume quindi valori in $\{0, 1\}$. Pertanto, l'elaborazione della sindrome, nel dominio quantistico, può essere effettuata nel dominio binario utilizzando la PCM \mathbf{H} e l'errore effettivo P . Ciò implica, a sua volta, che il processo di decodifica quantistico è equivalente alla decodifica della sindrome, del codice classico equivalente, che si basa sulla PCM \mathbf{H} [98]. Tuttavia, poiché i codici quantistici sono degeneri, come discusso nel Capitolo 4, si può concludere che la decodifica quantistica mira a stimare l'*insieme di errori* più probabile, mentre la decodifica della sindrome classica stima l'*errore* più probabile.

5.2 Isomorfismo dal dominio di Pauli al dominio quaternario

Analogamente all'isomorfismo dal dominio di Pauli al dominio classico binario, l'isomorfismo da Pauli al dominio quaternario facilita la progettazione dei codici quantistici, partendo dai codici classici quaternari già esistenti. In particolare, gli operatori di Pauli \mathbf{I} , \mathbf{X} , \mathbf{Y} e \mathbf{Z} possono essere trasformati negli elementi del campo di Galois di ordine 4 ($\text{GF}(4)$), utilizzando il mapping riportato di seguito:

$$\mathbf{I} \rightarrow 0, \quad \mathbf{X} \rightarrow 1, \quad \mathbf{Z} \rightarrow \omega, \quad \mathbf{Y} \rightarrow \bar{\omega}, \quad (5.16)$$

dove 0 , 1 , ω e $\bar{\omega}$ sono proprio gli elementi di $\text{GF}(4)$.

Approfondimento: Campo di Galois Il campo di Galois (*Galois Field*,

Tabella 5.3: Addizione in GF(4) [12].

+	0	1	ω	$\bar{\omega}$
0	0	1	ω	$\bar{\omega}$
1	1	0	$\bar{\omega}$	ω
ω	ω	$\bar{\omega}$	0	1
$\bar{\omega}$	$\bar{\omega}$	ω	1	0

da cui l'acronimo GF), chiamato anche campo finito, è un campo che contiene un insieme finito di elementi. Un campo è un insieme in cui sono definite le operazioni di moltiplicazione, addizione, sottrazione e divisione e tali operazioni soddisfano le proprietà riassunte nelle tabelle 5.3, 5.4, 5.5 e 5.6. L'ordine di un campo di Galois corrisponde al numero di elementi che lo compongono; tale numero è necessariamente un numero primo o una potenza di un numero primo. Si può pertanto dimostrare che, per ogni numero primo p e per ogni numero intero positivo k , esistono campi di ordine p^k , tutti isomorfi tra loro. Inoltre, le regole di addizione e moltiplicazione per un campo di Galois di ordine p ($\text{GF}(p)$), con p numero primo, sono le stesse della somma e della moltiplicazione modulo p , mentre esistono altre leggi quando si considera un campo di ordine p^m ($\text{GF}(p^m)$), con $m > 1$, come nel caso in esame, in cui si ha: $p = 2$ e $m = 2$.

Si può osservare che l'operazione di moltiplicazione nel dominio di Pauli è equivalente all'operazione di addizione in $\text{GF}(4)$, mentre il criterio di commutatività (cioè il prodotto simplettico) nel dominio di Pauli è equivalente alla traccia del prodotto scalare in $\text{GF}(4)$; si guardi a tal proposito la Tabella 5.1 [65]. In particolare, l'operatore di traccia del prodotto scalare di $\text{GF}(4)$ mappa x su $(x + \bar{x})$, dove \bar{x} denota il coniugato di x , dunque vale $\text{Tr}(x) \triangleq x + \bar{x} = x + x^2$ [35], poiché l'operazione di coniugazione nel campo di Galois di ordine 4 è definita come $\bar{x} = x^2$ [35]. Pertanto, tale operazione non ha alcun impatto sugli elementi 0 e 1 di $\text{GF}(4)$, mentre gli elementi ω e $\bar{\omega}$ sono scambiati quando si calcola il rispettivo coniugato.

Le regole relative all'addizione e alla moltiplicazione in $\text{GF}(4)$ sono mostrate nelle tabelle 5.3 e 5.4, rispettivamente.

La somma degli elementi di $\text{GF}(4)$ è equivalente alla somma bit-a-bit (bit-wise) modulo 2 degli equivalenti modelli a 2 bit; si guardi a tal proposito *Approfondimento: Campo di Galois*. Quindi, la Tabella 5.3 può essere ottenuta mappando i modelli a 2 bit della Tabella 5.2, relativa al dominio classico binario, sui corrispondenti elementi di $\text{GF}(4)$.

Inoltre, l'operazione di moltiplicazione delle matrici di Pauli $\{\mathbf{I}, \mathbf{X}, \mathbf{Z}, \mathbf{Y}\}$ con \mathbf{X} è equivalente a sommare l'elemento 1 di $\text{GF}(4)$ (corrispondente a \mathbf{X} , secondo il mapping dell'eq.(5.16)) a ciascun elemento di $\text{GF}(4)$, come fatto nella seconda colonna della Tabella 5.3. D'altra parte, la relazione commutativa tra due elementi \hat{A} e \hat{B} di $\text{GF}(4)$ può essere stabilita utilizzando la

Tabella 5.4: Moltiplicazione in GF(4) [12].

\times	0	1	ω	$\bar{\omega}$
0	0	0	0	0
1	0	1	ω	$\bar{\omega}$
ω	0	ω	$\bar{\omega}$	1
$\bar{\omega}$	0	$\bar{\omega}$	1	ω

traccia del prodotto scalare, come segue:

$$\text{Tr}\langle \hat{A}, \hat{B} \rangle = \text{Tr}(\hat{A} \times \overline{\hat{B}}) = 0, \tag{5.17}$$

dove \langle, \rangle denota il prodotto scalare Hermitiano, \times , in questo caso, denota il prodotto scalare semplice, mentre $\overline{\hat{B}}$ è il coniugato di \hat{B} .

N.B.: Gli elementi di un generico GF(4) sono indicate con $\hat{\cdot}$, per esempio \hat{x} .

In particolare, si può dimostrare che $\text{Tr}(\hat{A} \times \overline{\hat{B}}) = 0$ se gli operatori di Pauli associati ad a ed a b commutano, mentre $\text{Tr}(\hat{A} \times \overline{\hat{B}}) = 1$ se tali matrici sono anti-commutative. Esplicitamente:

$$\begin{cases} \text{Tr}(\hat{A} \times \overline{\hat{B}}) = 0, & \text{gli operatori di Pauli associati commutano} \\ \text{Tr}(\hat{A} \times \overline{\hat{B}}) = 1, & \text{gli operatori di Pauli associati sono anti-commutativi.} \end{cases} \tag{5.18}$$

Nel caso in esame, si ha che $\text{Tr}(0) = \text{Tr}(1) = 0$, invece $\text{Tr}(\omega) = \text{Tr}(\bar{\omega}) = 1$. A titolo di esempio, si riporta la verifica per il caso $\text{Tr}(0) = 0$ (*Esempio 1*) e per $\text{Tr}(\bar{\omega}) = 1$ (*Esempio 2*).

Esempio 1

Posto che l'elemento 0 di GF(4) può essere espresso anche come moltiplicazione di 0 con 1, in accordo con la Tabella 5.4, bisogna verificare che $\text{Tr}(0 \cdot 1) = 0$. Facendo riferimento all'eq.(5.18), $\hat{A} = 0$ e $\overline{\hat{B}} = 1$, allora $\hat{B} = 1$, dato che, in questo caso, vale $\overline{\hat{B}} = (\hat{B})^2 = 1$. Sapendo poi che l'operatore di Pauli associato a 0 è \mathbf{I} e che quello associato a 1 è \mathbf{Y} , secondo il mapping dell'eq.(5.16), si osserva che $\mathbf{IY} = \mathbf{Y} = \mathbf{YI}$, in accordo con l'eq.(5.18), cioè i due operatori risultano commutativi.

Si può facilmente dimostrare che sarebbe stata valida anche qualsiasi altra coppia di elementi di GF(4) che, moltiplicati tra loro, avesse dato come risultato 0.

Esempio 2

Dato che l'elemento $\bar{\omega}$ di GF(4) può essere espresso, tra gli altri, come la moltiplicazione di sé stesso con 1, in accordo con la Tabella 5.4, bisogna verificare che $\text{Tr}(\bar{\omega} \cdot 1) = 1$. Facendo riferimento all'eq.(5.18), in questo caso particolare $\hat{A} = \bar{\omega}$ e $\overline{\hat{B}} = 1$, allora $\hat{B} = 1$, per quanto detto sopra. Poiché l'operatore di Pauli associato a $\bar{\omega}$ è \mathbf{Z} e che quello associato a 1 è \mathbf{Y} , secondo il mapping dell'eq.(5.16), si osserva che $\mathbf{ZY} = -\mathbf{YZ}$, in accordo con l'eq.(5.18), cioè le matrici di Pauli associate ai due elementi di GF(4)

Tabella 5.5: Prodotto scalare Hermitiano in GF(4) [12].

\langle, \rangle	0	1	ω	$\bar{\omega}$
0	0	0	0	0
1	0	1	ω	$\bar{\omega}$
ω	0	$\bar{\omega}$	1	ω
$\bar{\omega}$	0	ω	$\bar{\omega}$	1

Tabella 5.6: Traccia del prodotto scalare in GF(4) [12].

$\text{Tr}\langle, \rangle$	0	1	ω	$\bar{\omega}$
0	0	0	0	0
1	0	0	1	1
ω	0	1	0	1
$\bar{\omega}$	0	1	1	0

designati risultano mutuamente anti-commutative.

Anche in questo caso, sarebbe stata valida qualsiasi altra coppia di elementi di GF(4) il cui prodotto avesse dato come risultato $\bar{\omega}$.

Inoltre, sia il prodotto scalare Hermitiano che la traccia del prodotto scalare stesso tra gli elementi di GF(4) sono riportati nella Tabella 5.5 e nella Tabella 5.6, rispettivamente.

Se un QSC è caratterizzato dalla PCM classica $\hat{\mathbf{H}}$ nel dominio quaternario, allora il vincolo di commutatività degli stabilizzatori g_i e g'_i si trasforma nella traccia del prodotto scalare dell' i -esima e i' -esima colonna di $\hat{\mathbf{H}}$. In particolare, questa relazione può essere formulata come segue:

$$\hat{\mathbf{H}}_i \star \hat{\mathbf{H}}_{i'} = \text{Tr}\langle \hat{\mathbf{H}}_i, \hat{\mathbf{H}}_{i'} \rangle = \text{Tr}\left(\sum_{t=1}^n \hat{\mathbf{H}}_{it} \times \overline{\hat{\mathbf{H}}_{i't}}\right) = 0, \quad (5.19)$$

dove $\hat{\mathbf{H}}_{it}$ è l'elemento corrispondente alla riga i ed alla colonna t della matrice $\hat{\mathbf{H}}$.

Si può ora osservare che l'eq.(5.13) e l'eq.(5.19) sono in realtà equivalenti, poiché entrambe rispondono al requisito della commutatività. Dato $\mathbf{H}_i = (\mathbf{H}_{z_i}, \mathbf{H}_{x_i})$ e conoscendo il mapping dell'eq.(5.16), la matrice $\hat{\mathbf{H}}_i$ può essere espressa come:

$$\hat{\mathbf{H}}_i = \omega \mathbf{H}_{z_i} + \mathbf{H}_{x_i}. \quad (5.20)$$

Sostituendo poi l'eq.(5.20) nell'eq.(5.19), si ha:

$$\begin{aligned} \hat{\mathbf{H}}_i \star \hat{\mathbf{H}}_{i'} &= \text{Tr}\langle (\omega \mathbf{H}_{z_i} + \mathbf{H}_{x_i}), (\omega \mathbf{H}_{z_{i'}} + \mathbf{H}_{x_{i'}}) \rangle \\ &= \text{Tr}((\omega \mathbf{H}_{z_i} + \mathbf{H}_{x_i}) (\bar{\omega} \mathbf{H}_{z_{i'}} + \mathbf{H}_{x_{i'}})) \\ &= \text{Tr}(\mathbf{H}_{z_i} \mathbf{H}_{z_{i'}} + \omega \mathbf{H}_{z_i} \mathbf{H}_{x_{i'}} + \bar{\omega} \mathbf{H}_{x_i} \mathbf{H}_{z_{i'}} + \mathbf{H}_{x_i} \mathbf{H}_{x_{i'}}), \end{aligned} \quad (5.21)$$

ricordando che $\text{Tr}(1) = 0$ e che $\text{Tr}(\omega) = \text{Tr}(\bar{\omega}) = 1$. Pertanto, l'eq.(5.21) si riduce a:

$$\hat{\mathbf{H}}_i \star \hat{\mathbf{H}}_{i'} = \mathbf{H}_{z_i} \mathbf{H}_{x_{i'}} + \mathbf{H}_{x_i} \mathbf{H}_{z_{i'}} \pmod{2}, \quad (5.22)$$

coincidente con l'eq.(5.13). Di conseguenza, analogamente all'eq.(5.15), la sindrome nel caso quaternario è calcolata come:

$$s_i = \text{Tr}(\hat{s}_i) = \text{Tr}\left(\sum_{t=1}^n \hat{\mathbf{H}}_{it} \times \overline{\hat{P}_t}\right), \quad (5.23)$$

dove s_i è la sindrome corrispondente all' i -esima riga di $\hat{\mathbf{H}}$ e \hat{P}_t è l'elemento t -esimo di \hat{P} , che rappresenta l'errore determinato sul t -esimo qubit.

Qualsiasi codice classico lineare quaternario arbitrario, che è auto-ortogonale rispetto alla traccia del prodotto scalare (eq.(5.19)), può essere utilizzato per la costruzione di un QSC.

Poiché un generico codice lineare quaternario è un spazio vettoriale chiuso rispetto all'operazione di moltiplicazione tra gli elementi di $\text{GF}(4)$, questa condizione si riduce a soddisfare il prodotto scalare Hermitiano, piuttosto che la traccia del prodotto scalare stesso [35]. Questo può essere dimostrato come segue.

Sia C un codice lineare classico in $\text{GF}(4)$ con parole di codice u e v . Inoltre, si suppone che:

$$\langle u, v \rangle = \alpha + \beta\omega. \quad (5.24)$$

Per soddisfare il prodotto simplettico, bisogna avere:

$$\text{Tr}\langle u, v \rangle = 0. \quad (5.25)$$

Poiché $\text{Tr}(\omega) = 1$, l'eq.(5.25) è valida solo quando β è nullo nell'eq.(5.24). Inoltre, dato che il codice C è lineare in $\text{GF}(4)$, l'eq.(5.25) conduce ad avere:

$$\text{Tr}\langle u, \bar{\omega}v \rangle = 0, \quad (5.26)$$

che a sua volta implica che anche α dovrebbe essere zero nell'eq.(5.24). Quindi, per un codice classico lineare in $\text{GF}(4)$, il prodotto scalare Hermitiano dell'eq.(5.24) deve essere zero, quando la traccia del prodotto scalare dell'eq.(5.25) è nulla.

5.3 Conclusioni

Gli stabilizzatori possono essere mappati sulle rappresentazioni binarie o quaternarie equivalenti, come riassunto nella Tabella 5.1. Questi mapping, a loro volta, aiutano nella progettazione dei codici quantistici, a partire dai codici classici già esistenti, come sarà discusso ulteriormente nel prossimo capitolo. Inoltre, poiché un QSC può essere mappato su una PCM binaria o quaternaria classica equivalente, la decodifica della sindrome basata su una PCM classica può essere invocata durante il processo di decodifica quantistico. Più esplicitamente, il blocco di elaborazione della sindrome di Fig. 4.1 può essere ampliato, come mostrato in Fig. 5.2. Il processo inizia con il calcolo della sindrome della sequenza ricevuta $|\hat{\psi}\rangle$ utilizzando gli stabilizer generator, che collassano al valore binario 0 o 1 al momento

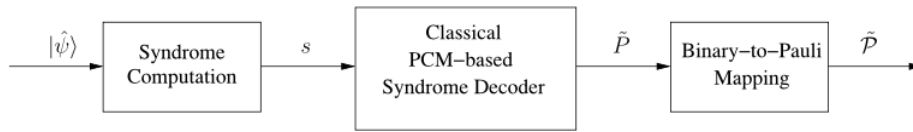


Figura 5.2: Schema a blocchi che rappresenta il processo di elaborazione della sindrome [12].

della misurazione. La sequenza della sindrome binaria s alimenta poi un decodificatore classico basato sulla PCM, che opera sull'equivalente PCM classica associata al QSC per stimare l'errore di canale equivalente \tilde{P} (o $\tilde{\hat{P}}$ nel dominio quaternario). Il decodificatore classico della sindrome basato sulla PCM della Fig. 5.2 è esattamente lo stesso decoder che si usa per un qualsiasi codice classico convenzionale, con l'eccezione delle seguenti due differenze:

1) In contrasto con la sindrome di un codice classico, che è data dal prodotto tra la PCM e il trasposto dell'errore di canale ($\mathbf{H}P^T$), la sindrome di un codice quantistico viene calcolata utilizzando il prodotto simplettico dell'eq.(5.15) (o la traccia del prodotto scalare dell'eq.(5.23)).

2) La decodifica classica convenzionale mira a stimare l'errore più probabile, data la sindrome osservata, mentre la decodifica quantistica ha l'obiettivo di stimare l'insieme di errori più probabile, che tiene conto della degenerazione dei codici quantistici, come discusso nel Capitolo 4.

Infine, il mapping dal dominio classico binario al dominio di Pauli dell'eq.(5.1) (o il mapping dal dominio classico quaternario al dominio di Pauli dell'eq.(5.16)) viene invocato per mappare l'errore stimato binario (o quaternario) sull'equivalente errore di Pauli $\tilde{\mathcal{P}}$.

Capitolo 6

Tassonomia dei codici stabilizzatori

L'isomorfismo tra il dominio classico e quello quantistico illustrato nel Capitolo 5 fornisce un solido quadro teorico per la costruzione dei codici quantistici, a partire dai codici classici già conosciuti. In particolare, i codici quantistici possono essere progettati a partire da una coppia di codici classici binari arbitrari, se soddisfano il criterio del prodotto simplettico, o da codici classici quaternari arbitrari, se soddisfano il prodotto scalare Hermitiano. Proseguendo la trattazione, in questo capitolo verrà presentata la tassonomia dei codici stabilizzatori, cioè la loro classificazione. Si osservi dunque la Fig. 6.1 che presenta le tre famiglie di QSC proposte nel seguito:

- Codici CSS, nella Sezione 6.1;
- Codici non CSS, nella Sezione 6.2;
- Codici EA, nella Sezione 6.3.

Tale figura pertanto mostra la struttura della PCM \mathbf{H} classica sottostante equivalente per ognuna di queste categorie di codice.

6.1 Codici Calderbank-Shor-Steane

I codici Calderbank-Shor-Steane (CSS) [153], [36], [152] sono una classe di codici stabilizzatori costruiti a partire da una coppia di codici classici binari. In particolare, la famiglia dei codici CSS può essere definita come segue.

Un codice CSS $[n, k_1 - k_2]$ può essere progettato a partire dai codici classici a blocco lineari binari $C_1(n, k_1)$ e $C_2(n, k_2)$, se lo spazio di codice di C_1 comprende lo spazio di C_2 ($C_2 \subset C_1$). Inoltre, se sia C_1 che il duale di C_2 , cioè C_2^\perp , presentano la stessa distanza minima di Hamming, pari a d_{min} , allora il codice CSS risultante presenta anch'esso una distanza minima di valore d_{min} . Pertanto, è in grado di correggere contemporaneamente fino a $(d_{min} - 1)/2$ errori di bit-flip ed altrettanti errori di phase-flip.

In altre parole, analogamente al codice di Shor, un codice CSS corregge in modo indipendente gli errori di bit-flip e di phase-flip. Più specificamente, il codice binario C_1 viene invocato per correggere le inversioni di bit, mentre il codice C_2^\perp viene utilizzato per la correzione delle inversioni di fase. Quindi,

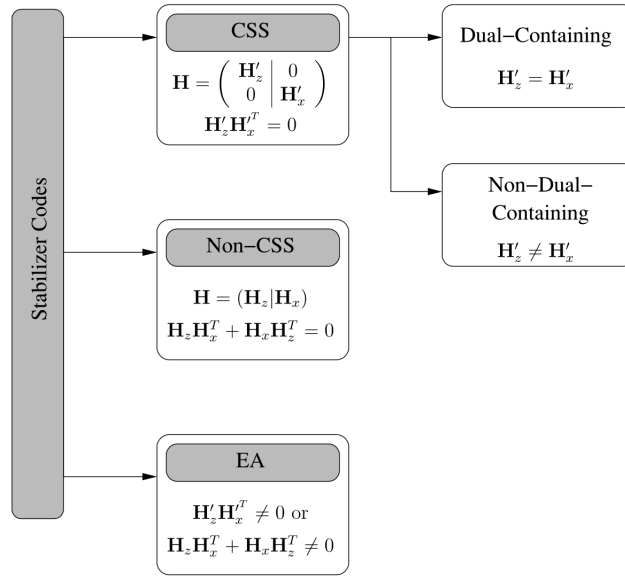


Figura 6.1: Tassonomia dei codici stabilizzatori (CSS: Calderbank-Shor-Steane, EA: Entanglement-Assisted) [12].

se \mathbf{H}'_z e \mathbf{H}'_x sono le PCM di C_1 e C_2^\perp , rispettivamente, allora il codice CSS risultante è caratterizzato dalla seguente PCM:

$$\mathbf{H} = [\mathbf{H}_z \mid \mathbf{H}_x] = \begin{pmatrix} \mathbf{H}'_z & | & \mathbf{0} \\ \mathbf{0} & | & \mathbf{H}'_x \end{pmatrix}, \quad (6.1)$$

dove si ha $\mathbf{H}_z = \begin{pmatrix} \mathbf{H}'_z \\ \mathbf{0} \end{pmatrix}$ e $\mathbf{H}_x = \begin{pmatrix} \mathbf{0} \\ \mathbf{H}'_x \end{pmatrix}$; dunque \mathbf{H}'_z e \mathbf{H}'_x sono matrici binarie di dimensioni $(n - k_1) \times n$ e $k_2 \times n$, rispettivamente. Inoltre, dato che vale $C_2 \subset C_1$, la condizione del prodotto simplettico nell'eq.(5.14) si riduce a:

$$\mathbf{H}'_z \mathbf{H}'_x{}^T = \mathbf{0}. \quad (6.2)$$

Quindi, il processo di progettazione di un QSC si riduce nel trovare una coppia di codici binari le cui PCM sono conformi al criterio del prodotto simplettico espresso dall'eq.(6.2). Poiché la PCM risultante dall'eq.(6.1) ha $(n - k_1 + k_2)$ righe, il codice quantistico codifica $(k_1 - k_2)$ qubit di informazione in n qubit codificati totali. Inoltre, se si ha $\mathbf{H}'_z = \mathbf{H}'_x$, allora il codice risultante è chiamato *dual-containing* (o auto-ortogonale), ed è caratterizzato da $\mathbf{H}'_z \mathbf{H}'_z{}^T = 0$, che è equivalente ad osservare che $C_1^\perp \subset C_1$. In particolare, nel caso dei codici CSS dual-containing, $C_2(n, k_2)$ è il codice duale di $C_1(n, k_1)$. Di conseguenza, si ha che $k_2 = (n - k_1)$ e i codici CSS dual-containing che ne risultano codificano $(k_1 - k_2) = (2k_1 - n)$ qubit di informazione in n qubit codificati. Si classificano ora i restanti modelli di codice CSS, in base alla Fig. 6.1, cioè quelli che hanno $\mathbf{H}'_z \neq \mathbf{H}'_x$, come codici CSS non-dual-containing.

Un codice CSS con $[n, k_1 - k_2]$ qubit, basato sui codici binari C_1 e C_2^\perp , è implementato trovando i sottoinsiemi unici di C_2 in C_1 , in modo che ciascuno

Tabella 6.1: Somma degli elementi di $Z_4 = \{0, 1, 2, 3\}$ modulo 4.

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Tabella 6.2: Somma bit-a-bit modulo 2.

+	0	1
0	0	1
1	1	0

dei $2^{k_1-k_2}$ stati sovrapposti possa essere mappato su un sottoinsieme unico di C_2 in C_1 .

Osservazione: sottoinsiemi unici Si supponga, a titolo di esempio, di lavorare con dei generici insiemi, invece che con gli spazi di codice. Sia dato allora l'insieme $C_1 = \{0, 1, 2, 3\}$ con $k_1 = 2$ e l'insieme $C_2 = \{0, 2\}$ con $k_2 = 1$, allora l'addizione modulo 4 tra ogni singolo elemento di C_1 con gli elementi di C_2 produce i seguenti sottoinsiemi:

$$\begin{aligned}
 0 + C_2 &\equiv \{0, 2\} = C_2; \\
 1 + C_2 &\equiv \{1, 3\} = 1 + C_2; \\
 2 + C_2 &\equiv \{2, 0\} = C_2; \\
 3 + C_2 &\equiv \{3, 1\} = 1 + C_2.
 \end{aligned} \tag{6.3}$$

Si consulti la Tabella 6.1 per osservare i risultati della somma modulo 4. Dunque, si osserva che ci sono due diversi sottoinsiemi unici di C_2 in C_1 , cioè, $\{0, 2\}$ e $\{1, 3\}$. Equivalentemente, si può dire che l'unione dei due sottoinsiemi unici $\{0, 2\}$ e $\{1, 3\}$ di C_2 costituisce l'insieme C_1 .

I sottoinsiemi unici di C_2 in C_1 , a loro volta, si derivano attraverso una somma bit-a-bit modulo 2 (Tabella 6.2) tra ogni parola di codice di C_1 e quelle dello spazio di codice di C_2 , si guardi, per capirne la ratio, l'esempio di cui sopra.

In particolare, se la parola di codice $x_1 \in C_1$ e la parola di codice $x_2 \in C_2$, l'operazione di somma normalizzata può essere formulata come segue:

$$|x_1 + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{x_2 \in C_2} |x_1 + x_2\rangle, \tag{6.4}$$

dove $+$ denota la somma modulo 2.

Poiché la cardinalità di C_1 è $|C_1| = 2^{k_1}$, mentre quella di C_2 è $|C_2| = 2^{k_2}$, si ottengono $|C_1|/|C_2| = 2^{k_1-k_2}$ sottoinsiemi unici di C_2 in C_1 . In altre parole, si hanno $2^{k_1-k_2}$ parole di codice diverse. Di conseguenza, ciascuno

Tabella 6.3: Sottoinsiemi unici di C_1^\perp in C_1 [12].

Sottoinsieme 1	Sottoinsieme 2
0000000	1111111
0111001	1000110
1011010	0100101
1100011	0011100
1101100	0010011
1010101	0101010
0110110	1001001
0001111	1110000

stato quantistico ortogonale, composto da $2^{k_1-k_2} \cdot (k_1 - k_2)$ qubit, può essere mappato su una sovrapposizione delle parole di codice del sottoinsieme unico.

6.1.1 Esempio Operativo

Si consideri la costruzione del codice di Steane [7, 1], il quale deriva dal codice classico dual-containing di Hamming (7, 4), che ha la seguente PCM:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}. \quad (6.5)$$

La PCM \mathbf{H} dell'eq.(6.5) produce $\mathbf{H}\mathbf{H}^T = \mathbf{0}$, quindi si presta essa stessa a costruire un codice CSS dual-containing. In particolare, C_1 è il codice di Hamming (7, 4), mentre C_2 è il suo codice duale, cioè $C_2 = C_1^\perp$, con parametri (7, 3). Dato che $\mathbf{H}\mathbf{H}^T = \mathbf{0}$, lo spazio di codice di C_2 è contenuto in quello di C_1 , cioè, si ha $C_2 \subset C_1$. Inoltre, sia C_1 che $C_2^\perp = C_1$ possono correggere un singolo errore. Di conseguenza, un codice CSS capace di correggere un singolo errore può essere costruito trovando i sottoinsiemi unici di C_1^\perp in C_1 , usando l'eq.(6.4), dato che $C_1^\perp = C_2$. Ciò si traduce in due sottoinsiemi unici, che sono elencati nella Tabella 6.3. Questi due sottoinsiemi, che contengono tutte le parole di codice possibili, partecipano a definire lo spazio di codice (7, 4) del codice di Hamming. I due stati ortogonali $|0\rangle$ e $|1\rangle$ della parola di informazione a singolo qubit sono quindi codificati come segue:

$$\begin{aligned} |\bar{0}\rangle &\equiv \frac{1}{\sqrt{8}}(|0000000\rangle + |0111001\rangle + |1011010\rangle + |1100011\rangle \\ &\quad + |1101100\rangle + |1010101\rangle + |0110110\rangle + |0001111\rangle), \\ |\bar{1}\rangle &\equiv \frac{1}{\sqrt{8}}(|1111111\rangle + |1000110\rangle + |0100101\rangle + |0011100\rangle \\ &\quad + |0010011\rangle + |0101010\rangle + |1001001\rangle + |1110000\rangle). \end{aligned} \quad (6.6)$$

In altre parole, $|\bar{0}\rangle$ e $|\bar{1}\rangle$ sono le sovrapposizioni, caratterizzate dallo stesso peso di Hamming, di tutte le parole di codice dei due sottoinsiemi della

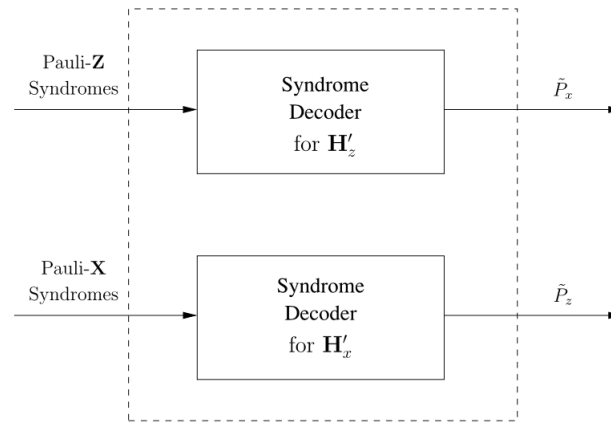


Figura 6.2: Decodificatore di sindrome per i codici quantistici di tipo CSS [12].

Tabella 6.3. Inoltre, \mathbf{H}'_z e \mathbf{H}'_x dello spazio di codice quantistico risultante sono equivalenti alla PCM binaria del codice di Hamming dell'eq.(6.5). Quindi, gli stabilizzatori associati, per il rilevamento degli errori di bit-flip e di phase-flip, del codice $[7, 1]$ di Steane sono i seguenti:

$$\begin{aligned}
 g_1 &= \mathbf{ZZIZZII}; \\
 g_2 &= \mathbf{ZIZZIZI}; \\
 g_3 &= \mathbf{IZZZIIZ}; \\
 g_4 &= \mathbf{XXIXXII}; \\
 g_5 &= \mathbf{XIXXIXI}; \\
 g_6 &= \mathbf{IXXXIIX}.
 \end{aligned} \tag{6.7}$$

Pertanto, la matrice che rappresenta il codice CSS di tipo dual-containing in esame è la seguente:

$$\mathbf{H}_{\text{css}} = \left(\begin{array}{c|c} \mathbf{H} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{H} \end{array} \right). \tag{6.8}$$

Si può così osservare che nell'eq.(6.7) così come nell'eq.(6.1), gli stabilizzatori di rilevamento degli errori di bit-flip e di phase-flip (o, equivalentemente, sindromi) di un codice quantistico di tipo CSS sono indipendenti. Pertanto, la stima del bit-flip e del phase-flip può essere effettuata, indipendentemente, da due decodificatori classici della sindrome separati, utilizzando le matrici \mathbf{H}'_z e \mathbf{H}'_x , rispettivamente, come illustrato in Fig. 6.2. Inoltre, quando viene utilizzato il decodificatore rappresentato in forma compatta nella Fig. 6.2, la prestazione dei codici CSS, osservata nell'ambito del canale di depolarizzazione espresso dall'eq.(1.59), è isomorfa rispetto alle stesse su due canali indipendenti di inversione di fase e di inversione di bit, dove ciascuno ha una probabilità di depolarizzazione pari a $2p/3$. Quindi le prestazioni, in termini di QBER, dei codici CSS possono essere approximate sommando tra loro i BER dei codici binari costituenti. In particolare, dato che p_e^x e p_e^z

sono i BER classici per \mathbf{H}'_x e \mathbf{H}'_z , rispettivamente, il codice CSS risultante presenta un QBER di:

$$\text{QBER} = p_e^x + p_e^z - p_e^x p_e^z \approx p_e^x + p_e^z, \quad (6.9)$$

che è equivalente a $2p_e^z$ per un codice CSS dual-containing con $\mathbf{H}'_x = \mathbf{H}'_z$.

6.2 Codici Non CSS

Si è osservato nella sezione precedente che i codici CSS correggono indipendentemente gli errori di bit-flip e di phase-flip. Questo, a sua volta, si traduce in un basso rate di codifica. Al contrario, i codici stabilizzatori non CSS sono in grado di sfruttare la ridondanza in modo più efficiente, poiché correggono gli errori di tipo bit-flip e di tipo phase-flip in modo congiunto. La PCM di un codice non CSS assume la struttura generale espressa nell'eq.(5.2). Di conseguenza, per progettare un codice stabilizzatore non CSS si possono utilizzare:

- una coppia di PCM binarie, conformi al criterio del prodotto simplettico espresso dall'eq.(5.14);
- una PCM classica quaternaria, che soddisfi il criterio della traccia del prodotto scalare, espresso dall'eq.(5.19).

Calderbank, Rains, Shor e Sloane hanno concepito una classe speciale di codici non CSS, chiamati codici Calderbank-Rains-Shor-Sloane (CRSS), che sono costruiti a partire dai codici quaternari classici e possono essere caratterizzati come descritto di seguito [35].

Un QSC con $[n, k]$ qubit può essere progettato, nel dominio quaternario, a partire da un codice classico auto-ortogonale (rispetto al prodotto scalare Hermitiano) lineare in $\text{GF}(4)$ a blocco $C(n, (n - k)/2)$. Inoltre, se il codice duale (chiamato anche ortogonale) $C^\perp(n, (n + k)/2)$ presenta una distanza minima di Hamming pari a d_{min} , allora il codice non CSS risultante presenta anch'esso una distanza minima di valore d_{min} . Pertanto, è in grado di correggere contemporaneamente fino a $(d_{min} - 1)/2$ errori di bit-flip così come $(d_{min} - 1)/2$ errori di phase-flip.

Sulla base di questo formalismo, la PCM del codice CRSS risultante è caratterizzata come segue:

$$\hat{\mathbf{H}} = \begin{pmatrix} \hat{\mathbf{H}}_c \\ \omega \hat{\mathbf{H}}_c \end{pmatrix}, \quad (6.10)$$

dove $\hat{\mathbf{H}}_c$ è la PCM del codice duale $C^\perp(n, (n + k)/2)$.

6.2.1 Esempio Operativo

Esiste un codice classico lineare in $\text{GF}(4)$ auto-ortogonale $C(5, 2)$, il cui codice duale $C^\perp(5, 3)$ è un codice di Hamming che ha la PCM $\hat{\mathbf{H}}_c$ data da [58]:

$$\hat{\mathbf{H}}_c = \begin{pmatrix} 0 & \bar{\omega} & \omega & \omega & \bar{\omega} \\ \bar{\omega} & 0 & \bar{\omega} & \omega & \omega \end{pmatrix}. \quad (6.11)$$

Di conseguenza, il codice quantistico $(5, 1)$ di Hamming può essere costruito come segue:

$$\hat{\mathbf{H}} = \begin{pmatrix} 0 & \bar{\omega} & \omega & \omega & \bar{\omega} \\ \bar{\omega} & 0 & \bar{\omega} & \omega & \omega \\ 0 & 1 & \bar{\omega} & \bar{\omega} & 1 \\ 1 & 0 & 1 & \bar{\omega} & \bar{\omega} \end{pmatrix}. \quad (6.12)$$

Utilizzando il mapping dal dominio quaternario classico al dominio di Pauli, in accordo con l'eq.(5.16), la PCM $\hat{\mathbf{H}}$ dell'eq.(6.12) viene mappata sugli stabilizer generator elencati di seguito:

$$\begin{aligned} g_1 &= \mathbf{IYZZY}; \\ g_2 &= \mathbf{YIYZZ}; \\ g_3 &= \mathbf{IXYYX}; \\ g_4 &= \mathbf{XIXYY}, \end{aligned} \quad (6.13)$$

che rappresentano appunto le righe di $\hat{\mathbf{H}}$.

Quindi, mentre un codice di tipo CSS che è in grado di correggere un singolo errore ha un rate di codifica pari a $1/7$, un codice non CSS, che parimenti è in grado di correggere fino ad un singolo errore, presenta un rate di codifica (maggiore, e quindi più favorevole) pari a $1/5$. I codici risultanti possono essere decodificati utilizzando un decodificatore classico basato sulla sindrome non binaria oppure un decodificatore basato sulla sindrome binaria che opera utilizzando la PCM binaria fornita dall'eq.(5.2), sfruttando la correlazione tra gli errori di tipo bit-flip e di tipo phase-flip, facilitandone così la decodifica congiunta. Questo, a sua volta, fornisce prestazioni di decodifica migliorate, anche se contestualmente aumenta la complessità di decodifica associata.

6.3 Codici Entanglement-Assisted

Si ricorda che i QSC possono essere costruiti a partire dai codici classici binari e quaternari, solo se i codici classici costituenti sono conformi al criterio del prodotto simplettico dell'eq.(5.14). Di conseguenza, mentre ogni QSC può avere una controparte nel dominio classico, non si può certo affermare il contrario; cioè che ogni codice classico abbia una versione quantistica basata sugli stabilizzatori. Inoltre, il rigoroso criterio del prodotto simplettico può comportare vari problemi di progettazione, come i cicli brevi, inevitabili nei codici QLDPC, e la non ricorsività, nel caso dei QCC non catastrofici. Al fine di superare queste limitazioni, è stato concepito il formalismo stabilizzatore entanglement-assisted di [31] e [32], che si basa su qubit entangled già condivisi con il ricevitore autorizzato su un canale privo di rumore. In particolare, il formalismo EA permette di trasformare un insieme di generatori di Pauli non commutativi in un insieme di generatori commutativi, che a loro volta costituiscono validi codici stabilizzatori. Di conseguenza, quando i codici classici sottostanti non soddisfano il criterio del prodotto

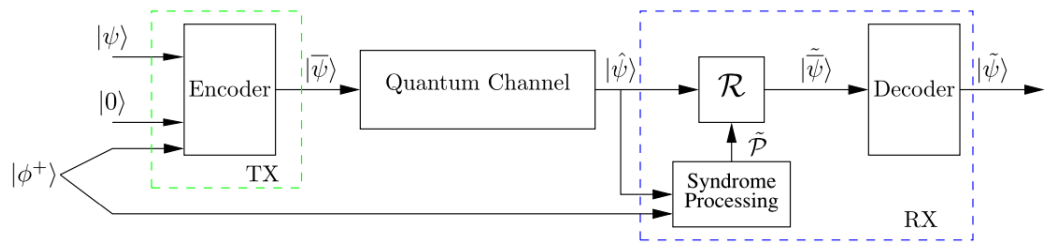


Figura 6.3: Schema a blocchi di un sistema di comunicazione quantistica basato su un codice stabilizzatore quantistico entanglement-assisted [12].

simplettico, viene invocato il formalismo EA per rendere commutativi gli stabilizzatori dati.

La Fig. 6.3 mostra il modello di un sistema di comunicazione quantistica basato su un codice stabilizzatore quantistico entanglement-assisted (EA-QSC, Entanglement-Assisted - Quantum Stabilizer Code). In particolare, un EA-QSC con $[n, k, c]$ qubit codifica una sequenza di informazione di k qubit $|\psi\rangle$ in una parola di codice, composta da n qubit $|\bar{\psi}\rangle$, utilizzando $(n - k - c)$ qubit ausiliari nello stato $|0\rangle$ e c qubit entangled già condivisi, chiamati più semplicemente *ebit* (entangled qubit). Gli ebit possono essere realizzati nello stato di Bell $|\phi^+\rangle$ e sono espressi come:

$$|\phi^+\rangle = \frac{|00\rangle^{T_X R_X} + |11\rangle^{T_X R_X}}{\sqrt{2}}, \quad (6.14)$$

si osservi a questo proposito *Approfondimento: Stati di Bell*, nella Sottosezione 1.1.2. L'eq.(6.14), che rappresenta una coppia di ebit, indica che il primo qubit è conservato presso il trasmettitore (T_X), mentre il secondo qubit, entangled rispetto al primo, viene inviato al ricevitore (R_X) prima che inizi la trasmissione effettiva, ad esempio durante le ore "non di punta", quando i canali sono poco utilizzati. Le notazioni T_X e R_X nell'eq.(6.14) vengono utilizzate per identificare la "metà" dell'ebit (cioè lo stato $|0\rangle$ o $|1\rangle$) del trasmettitore e del ricevitore, rispettivamente. Come detto, generalmente si presuppone che il pre-sharing degli ebit avvenga su un canale non rumoroso. Inoltre, come illustrato nella Fig. 6.3, il trasmettitore utilizza solo la "sua metà" degli ebit ($|0\rangle^{T_X}$ oppure $|1\rangle^{T_X}$) per codificare la sequenza di informazione $|\psi\rangle$ nella parola di codice $|\bar{\psi}\rangle$. Infine, le informazioni codificate vengono inviate su un canale quantistico rumoroso, a tal proposito è sempre valida la rappresentazione sintetica proposta dalla Fig. 2.9. Al ricevitore, la parola di codice ricevuta $|\hat{\psi}\rangle$ è pertanto affetta da rumore ed è combinata con la metà del ricevitore dei c ebit, durante il processo di decodifica. In particolare, gli stabilizzatori di un EA-QSC agiscono congiuntamente sulla parola di codice ricevuta $|\hat{\psi}\rangle$ e sugli ebit del ricevitore, per calcolare il vettore di sindrome. Questo poi viene utilizzato da un decodificatore basato sulla sindrome classica per stimare il modello di errore \tilde{P} , come precedentemente mostrato nella Fig. 5.2. Il resto del processo di elaborazione, al ricevitore, è identico a quello del QSC non assistito, presentato in Fig. 4.1.

Ora bisogna verificare se la matrice \mathbf{H} rispetta il criterio del prodotto simplettico, espresso nell'eq.(5.14), per il quale si ha che:

$$\mathbf{H}_z \mathbf{H}_x^T + \mathbf{H}_x \mathbf{H}_z^T = \mathbf{0} \text{ mod } 2. \quad (6.20)$$

Dopo aver calcolato le matrici trasposte \mathbf{H}_x^T e \mathbf{H}_z^T e sostituendo nell'equazione precedente, diventa:

$$\begin{aligned} & \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ & = \begin{pmatrix} 0 & 1 & 2 & 2 \\ 1 & 0 & 2 & 2 \\ 2 & 2 & 2 & 4 \\ 2 & 2 & 4 & 4 \end{pmatrix} \neq \mathbf{0} \text{ mod } 2. \end{aligned} \quad (6.21)$$

Sfortunatamente dunque, la PCM dell'eq.(6.19) non soddisfa il criterio del prodotto simplettico.

D'altra parte, la PCM \mathbf{H} può essere trasformata nei seguenti generatori di Pauli, utilizzando il mapping dal dominio di Pauli al dominio classico binario dell'eq.(5.1):

$$\mathbf{H}_Q = \begin{pmatrix} \mathbf{X} & \mathbf{Z} & \mathbf{X} & \mathbf{I} \\ \mathbf{X} & \mathbf{X} & \mathbf{I} & \mathbf{X} \\ \mathbf{Y} & \mathbf{Z} & \mathbf{Z} & \mathbf{X} \\ \mathbf{X} & \mathbf{Y} & \mathbf{Y} & \mathbf{Z} \end{pmatrix}, \quad (6.22)$$

dove il pedice Q sta ad indicare che la matrice \mathbf{H}_Q è l'equivalente, nel dominio quantistico, della PCM \mathbf{H} dell'eq.(6.19). Nello specifico, si ha a che fare con i seguenti generatori:

$$\begin{aligned} g_1 &= \mathbf{XZ XI} \rightarrow g_1 = 01100100; \\ g_2 &= \mathbf{XX IX} \rightarrow g_2 = 01010001; \\ g_3 &= \mathbf{YZ ZX} \rightarrow g_3 = 11101001; \\ g_4 &= \mathbf{XY YZ} \rightarrow g_4 = 01111110, \end{aligned} \quad (6.23)$$

dove \rightarrow indica il mapping dal dominio di Pauli al dominio classico binario, secondo l'eq.(5.1), che inevitabilmente risultano non commutativi, dato che la PCM \mathbf{H} risulta il criterio del prodotto simplettico.

In particolare, i primi due generatori (o righe) di \mathbf{H}_Q sono anti-commutativi, poiché il numero di indici corrispondenti ad operatori di Pauli non di identità, cioè \mathbf{X} , \mathbf{Z} o \mathbf{Y} , diversi tra loro, è pari a 1, poiché vale:

$$\begin{aligned} (i) & \mathbf{X} = \mathbf{X}; \\ (ii) & \mathbf{Z} \neq \mathbf{X}; \\ (iii) & \mathbf{X} \neq \mathbf{I}; \\ (iv) & \mathbf{I} \neq \mathbf{X}, \end{aligned} \quad (6.24)$$

dove la prima valutazione dà esito negativo, poiché i due operatori di Pauli sono uguali, mentre la terza e la quarta valutazione non sono valide in quanto \mathbf{I} è la matrice di identità. Dunque l'unica valutazione che dà esito positivo è la seconda. Pertanto quando il numero di indici è dispari, cioè 1, come in questo caso, i due generatori designati non sono commutativi.

Diversamente, con lo stesso metodo di cui sopra, si può dimostrare che tutti gli altri generatori (o righe) effettivamente commutano tra loro. In altre parole, solo gli operatori che agiscono sul secondo qubit anti-commutano, mentre gli operatori che agiscono individualmente su tutti gli altri qubit risultano essere commutativi. Per costruire i generatori dell'eq.(6.22) in modo tale che siano commutativi, le prime due righe della matrice \mathbf{H}_Q possono essere "aumentate" con una coppia di operatori non di identità anti-commutativi, come mostrato di seguito:

$$\mathbf{H}_Q = \left(\begin{array}{cccc|c} \mathbf{X} & \mathbf{Z} & \mathbf{X} & \mathbf{I} & \mathbf{Z} \\ \mathbf{X} & \mathbf{X} & \mathbf{I} & \mathbf{X} & \mathbf{X} \\ \mathbf{Y} & \mathbf{Z} & \mathbf{Z} & \mathbf{X} & \mathbf{I} \\ \mathbf{X} & \mathbf{Y} & \mathbf{Y} & \mathbf{Z} & \mathbf{I} \end{array} \right), \quad (6.25)$$

dove infatti vale $\mathbf{ZX} = -\mathbf{XZ}$. Come si può facilmente dimostrare, le matrici di identità \mathbf{I} possono essere aggiunte senza modificare il conteggio; in questa sede hanno dunque la funzione di elemento neutro. In questo modo si è ottenuta una matrice \mathbf{H}_Q effettivamente commutativa, cioè che rispetta il criterio il prodotto simplettico e che dunque può essere utilizzata per rappresentare un codice non CSS, come desiderato in partenza.

Si osserva pertanto che gli operatori a sinistra della barra verticale (|) agiscono sulle parole di codice trasmesse, composte da n qubit, mentre quelli a destra della barra verticale agiscono sugli ebit del ricevitore. Dunque, in questo esempio di progettazione, è richiesto solo un ebit poiché è presente solo una colonna a destra della barra verticale.

Capitolo 7

Esempi di progetto

Dall'analisi sviluppata nei capitoli precedenti si può concludere che il formalismo stabilizzatore è una struttura utile per sfruttare al meglio, ai fini del progetto di codici quantistici, le famiglie di codice classiche conosciute. In questo capitolo dunque, si estendono le considerazioni a due famiglie di codici di canale ampiamente utilizzate, cioè i codici BCH (Sezione 7.1) e i codici convoluzionali (Sezione 7.2), sottolineando, per ognuna di esse, la dualità tra la versione classica e quella quantistica.

7.1 Codici Bose-Chaudhuri-Hocquenghem

7.1.1 *Codici classici Bose-Chaudhuri-Hocquenghem*

I codici Bose-Chaudhuri-Hocquenghem (codici BCH) sono codici a blocco ciclici a distanza minima massima (per una data ridondanza) e sono in grado di correggere errori multipli. Un codice BCH classico, indicato come $BCH(n, k, d_{min})$, codifica $k \geq (n - mt)$ bit di informazione in parole di codice di n simboli, dove $n = 2^m - 1$, in modo che lo spazio di codice risultante abbia una distanza minima di Hamming dispari di valore d_{min} ; quindi tale codice è in grado di correggere fino a $t = (d_{min} - 1)/2$ errori.

Inoltre, i codici BCH possono essere sia sistematici che non sistematici, si guardi a tal proposito l'*Approfondimento: Codici sistematici* della Sezione 2.1.2. Tuttavia, è noto che i codici BCH sistematici hanno prestazioni superiori a quelle delle loro controparti non sistematiche [77]. Questo perché possono sfruttare la loro capacità di rilevamento degli errori per disabilitare le operazioni di decodifica, quando ciò comporterebbe la correzione dei simboli sbagliati, nei casi in cui si ha un numero di errori maggiore di t . In tali situazioni, il decodificatore sistematico BCH mantiene semplicemente la parte sistematica della parola di codice. Sfortunatamente, il decodificatore non sistematico non separa le informazioni e non ha segmenti di parità (cioè sequenze di bit di parità), quindi correggerebbe i simboli sbagliati anche quando è affetto da più di t errori. Questo causa, dopo la decodifica, ancora più errori rispetto a quelli che si hanno, inizialmente, all'uscita del canale.

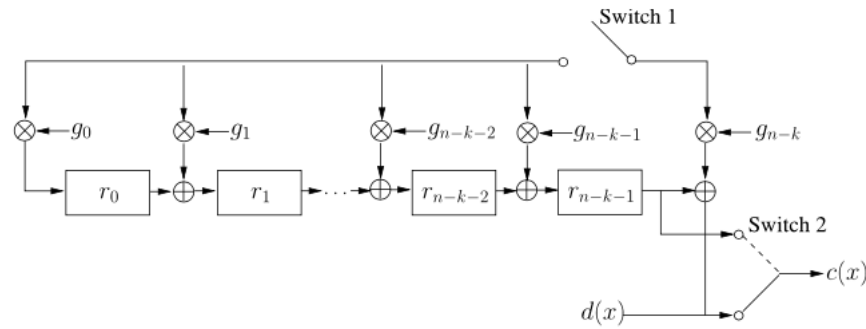


Figura 7.1: Schematico del generico codificatore sistemático BCH (n, k, d_{min}) [12].

Un codice binario BCH sistemático dunque, codifica i k bit di informazione in n simboli codificati, aggiungendo $(n - k)$ bit di parità al blocco di k bit di informazione. Dato il polinomio d'informazione $d(x)$, cioè il polinomio posto in ingresso al codificatore, i bit di parità sono calcolati a partire dai bit di informazione, utilizzando il polinomio generatore $g(x)$, che è dato da:

$$g(x) = g_0 + g_1x + g_2x^2 + \cdots + g_{n-k}x^{n-k}. \quad (7.1)$$

Come descritto in [77] e [75], il codificatore sistemático opera secondo il seguente schema:

- prima si moltiplica il polinomio d'informazione $d(x)$ per $x^{(n-k)}$;
- in questo modo si è spostato il polinomio $d(x)$ nella posizione di ordine più alto, della parola di codice $c(x)$;
- infine si collega ad essa il segmento di parità.

Esplicitamente, i bit di parità, rappresentati dal polinomio $p(x)$, sono definiti secondo il polinomio generatore $g(x)$, in modo che la parola di codice risultante $c(x)$ sia una parola di codice valida. Il processo complessivo di codifica sistemática può essere matematicamente riassunto come segue:

$$c(x) = x^{(n-k)} \cdot d(x) + p(x), \quad (7.2)$$

dove $p(x)$ è definito come:

$$p(x) = -\text{Rem} \left[\frac{x^{(n-k)} \cdot d(x)}{g(x)} \right], \quad (7.3)$$

dove l'operatore Rem indica il resto della divisione intera, ulteriori chiarimenti vengono forniti nel seguente *Approfondimento: Operazione di Resto*. Al fine di verificare che $c(x)$ sia una parola di codice valida, bisogna dimostrare che il resto della divisione tra $c(x)$ e il polinomio generatore $g(x)$ valga 0. Infatti, si ha:

$$\begin{aligned} \text{Rem} \left[\frac{c(x)}{g(x)} \right] &= \text{Rem} \left[\frac{x^{(n-k)} \cdot d(x) + p(x)}{g(x)} \right] \\ &= \text{Rem} \left[\frac{x^{(n-k)} \cdot d(x)}{g(x)} \right] + \text{Rem} \left[\frac{p(x)}{g(x)} \right] = 0, \end{aligned} \quad (7.4)$$

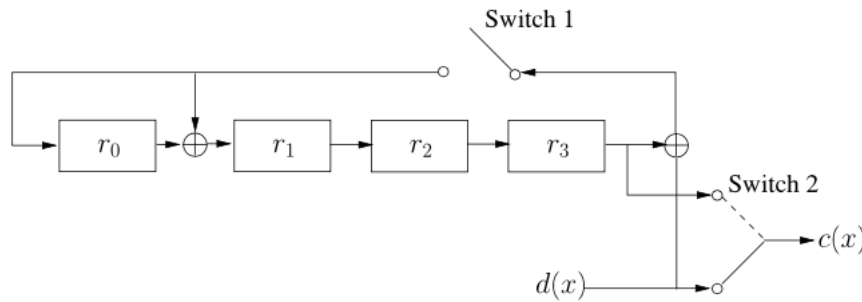


Figura 7.2: Codificatore sistematico BCH (n, k, d_{min}) [12].

poiché

$$\text{Rem} \left[\frac{p(x)}{g(x)} \right] = p(x) \quad (7.5)$$

e

$$\text{Rem} \left[\frac{x^{(n-k)} \cdot d(x)}{g(x)} \right] = -p(x), \quad (7.6)$$

in accordo con l'eq.(7.3).

Approfondimento: Operazione di Resto L'operazione $r = \text{Rem}(a, b)$ oppure $r = \text{Rem}[a/b]$ restituisce il resto dopo la divisione tra a e b , dove a è il dividendo e b è il divisore. Questa funzione è chiamata *Remainder*, da cui il diminutivo Rem.

In particolare, nel caso della divisione tra polinomi, un importante teorema dell'algebra afferma che, dati due polinomi $P(x)$ (polinomio dividendo) e $D(x)$ (polinomio divisore), è sempre possibile determinare due polinomi $Q(x)$ (polinomio quoziente) e $R(x)$ (polinomio resto) tali che:

$$P(x) = Q(x)D(x) + R(x), \quad \text{con } \text{gr}[R(x)] < \text{gr}[D(x)], \quad (7.7)$$

dove l'operatore $\text{gr}[\cdot]$ indica il grado del polinomio specificato.

Pertanto, anche in virtù della trattazione proposta, è interessante osservare che, se si ottiene $R(x) = 0$, allora la divisione polinomiale si dice esatta, altrimenti, se $R(x) \neq 0$, si dice non esatta.

Le corrispondenti moltiplicazioni e divisioni polinomiali delle eq.(7.2) e eq.(7.3), rispettivamente, possono essere realizzate attraverso operazioni basate su registri a scorrimento (shift register) a bassa complessità, come esemplificato qui di seguito.

Il codificatore di un codice BCH sistematico può essere implementato utilizzando i registri a scorrimento, come illustrato nella Fig. 7.1, dove \otimes denota l'operazione di moltiplicazione, mentre \oplus è l'operatore di somma modulo 2. In particolare, i bit di informazione $d(x)$ sono codificati nei bit $c(x)$, come segue:

1) L'interruttore 1 (switch 1) viene chiuso durante i primi k istanti temporali (o cicli di clock), consentendo quindi ai bit di informazione di $d(x)$

Tabella 7.1: Processo di codifica di un codice BCH(15, 11, 3) [12].

Indice	Input Bit	Stato $(r_0r_1r_2r_3)$ Binario	Stato $(r_0r_1r_2r_3)$ Decimale	Output Bit
0	-	0000	0	-
1	1	1100	12	1
2	0	0110	6	0
3	0	0011	3	0
4	0	0001	1	0
5	1	1100	12	1
6	1	1010	10	1
7	1	1001	9	1
8	0	0100	4	0
9	0	0010	2	0
10	1	1101	13	1
11	1	1010	10	1
12	-	0101	5	0
13	-	0010	2	1
14	-	0001	1	0
15	-	0000	0	1

di fluire nei $(n - k)$ registri a scorrimento, secondo le regole definite dal polinomio generatore $g(x)$. In particolare, i contenuti degli shift register dopo l'istante di tempo k -esimo costituiscono i bit di parità.

2) Contemporaneamente, l'interruttore 2 (switch 2) è in posizione bassa, in modo che i k bit di informazione $d(x)$ costituiscano i primi k bit di $c(x)$.

3) Dopo k istanti di tempo, l'interruttore 1 viene aperto, mentre l'interruttore 2 viene spostato nella posizione superiore. Questo ripulisce il contenuto dei registri a scorrimento, poiché i loro contenuti vengono spostati nell'uscita $c(x)$.

7.1.2 Esempio Operativo: codice classico BCH

Si consideri il codice BCH(15, 11, 3), che ha il seguente polinomio generatore:

$$\begin{aligned}
 g &= 23_{\text{ott}} \\
 &= 10011_{\text{bin}}, \\
 g(x) &= x^4 + x + 1.
 \end{aligned} \tag{7.8}$$

Osservazione Il polinomio generatore $g(x)$ è spesso rappresentato da un numero ottale, così che quando viene convertito nella notazione binaria, il bit più a destra costituisce il coefficiente di x^0 , cioè il coefficiente di grado zero.

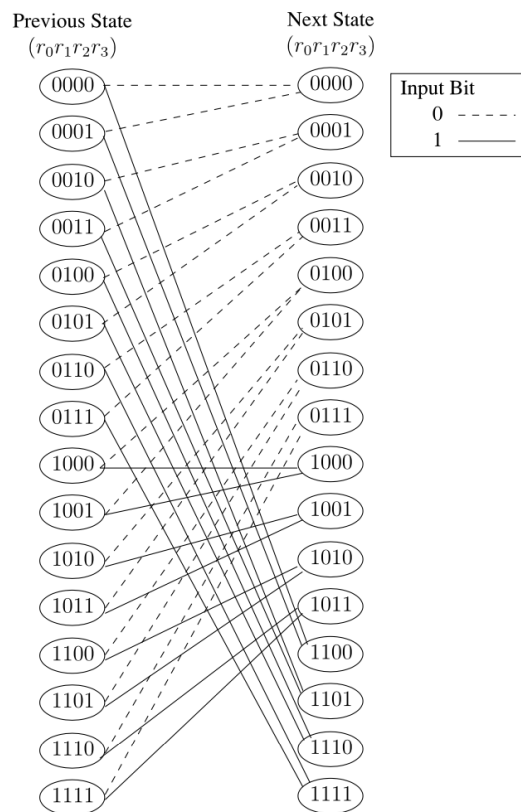


Figura 7.3: Diagramma di transizione di stato per un codice BCH(15, 11, 3) [12].

Il circuito di codifica associato, visibile nella Fig. 7.2, può essere facilmente derivato da quello di Fig. 7.1, sulla base dell'eq.(7.8) che caratterizza il polinomio generatore. Dall'eq.(7.8) si può inoltre osservare che i coefficienti possono avere solo valore pari a 1 o 0. Di conseguenza, il moltiplicatore è sostituito da un collegamento diretto, se il coefficiente corrispondente è 1, mentre non viene effettuata alcuna connessione, quando il coefficiente è 0. Si supponga quindi di utilizzare una sequenza di input a 11 bit $d = 11001110001$, che può anche essere rappresentata come $d(x) = 1 + x + x^4 + x^5 + x^6 + x^{10}$. Il relativo processo di codifica si compone delle seguenti parti:

1) Tutti i registri a scorrimento sono inizializzati allo stato zero. Durante i primi $k = 11$ istanti di tempo, quando cioè l'interruttore 1 è chiuso, i bit di ingresso di d fluiscono nel registro a scorrimento della Fig. 7.2. Gli stati risultanti sono riportati nella Tabella 7.1 per ogni istante di tempo; in particolare, con la notazione "indice" si intende l' i -esimo stato temporale, con $0 \leq i \leq 15$. Questa tabella rappresenta il processo di codifica di un codice BCH(15, 11, 3), con $d = 11001110001$ ($d(x) = 1 + x + x^4 + x^5 + x^6 + x^{10}$), che produce la parola di codice $c = 101011001110001$ ($c(x) = 1 + x^2 + x^4 + x^5 + x^8 + x^9 + x^{10} + x^{14}$).

2) Così, poiché l'interruttore 2 è posizionato in basso per i primi $k = 11$ istanti, i primi $k = 11$ bit codificati di $c(x)$ sono uguali ai $k = 11$ bit di informazione di $d(x)$.

3) Successivamente, poiché l'interruttore 1 viene aperto e l'interruttore

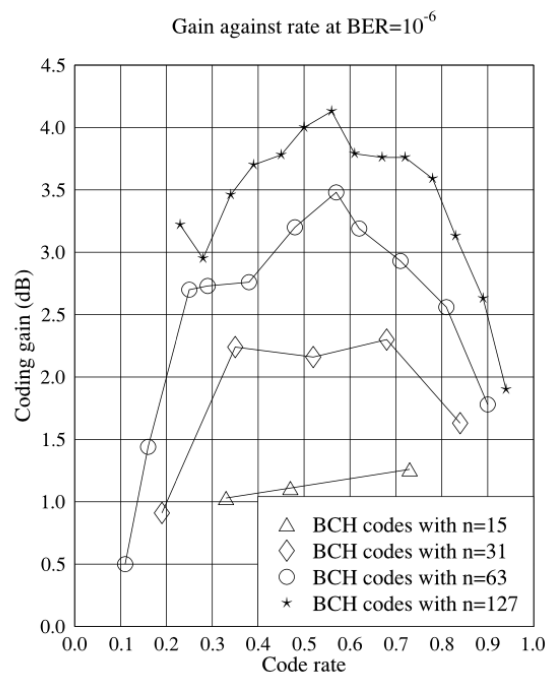


Figura 7.4: Guadagno di codifica rispetto al rate di codifica, per varie famiglie di codici BCH, con un BER di 10^{-6} sul canale AWGN [77]. *Nota:* è stato utilizzato per la decodifica l'algoritmo di Berlekamp-Massey [12].

2 viene spostato nella posizione superiore, i valori all'interno dei registri a scorrimento rappresentano i restanti $n - k = 15 - 11 = 4$ bit codificati di $c(x)$, come illustrato nella Tabella 7.1. Alla fine, tutti gli shift register vengono reimpostati allo stato iniziale, cioè zero.

In modo equivalente, il processo di codifica della Tabella 7.1 può anche essere rappresentato utilizzando il diagramma di transizione di stato della Fig. 7.3, che mostra tutte le possibili transizioni per il codificatore BCH della Fig. 7.2.

Nella sua forma concettualmente più semplice, la decodifica si basa su una semplice tabella di decodifica, che ha un totale di $2^{15} = 32768$ possibili input e $2^{11} = 2048$ parole di codice legittime. Poiché questo codice è caratterizzato da una distanza minima di Hamming di valore $d_{min} = 3$, la parola di codice corrotta ricevuta viene prontamente corretta in caso di un singolo errore, ma se la parola di codice legittima fosse sbagliata, nel caso dell'occorrenza di due errori, essa verrebbe comunque selezionata dal ricevitore. Il diagramma di transizione di stato della Fig. 7.3 facilita anche la decodifica a traliccio (trellis decoding) dei codici BCH [177]. Tuttavia, il numero di stati del traliccio aumenta esponenzialmente in ragione del fattore $(n - k)$, poiché il traliccio ha un totale di $2^{(n-k)}$ stati. Come strategia alternativa, l'algoritmo di Berlekamp-Massey [24], [54], [104] e [105] e l'algoritmo di Chase [40] sono ampiamente utilizzati per decodificare in modo efficiente i codici BCH.

La Fig. 7.4, invece, illustra l'andamento del guadagno di codifica rispetto al rate di codifica con un BER di 10^{-6} per codici BCH a rate diverso, e quindi capaci di correggere un numero diverso di errori, che utilizzano la stessa lunghezza del codice, cioè per $n = 15, 31, 63, 127$. Dalla Fig. 7.4 si può osservare che il guadagno di codifica aumenta all'aumentare del rate di codifica (cioè aumentando k) fino a raggiungere il valore massimo. In particolare, il guadagno massimo di codifica viene, in genere, ottenuto quando il rate è compreso tra 0.5 e 0.6.

7.1.3 Codici quantistici Bose-Chaudhuri-Hocquenghem

I codici quantistici BCH [154], [68], [35], [66], [155], [179], conosciuti come QBCH (Quantum Bose-Chaudhuri-Hocquenghem codes), possono essere derivati dai codici classici BCH binari dual-containing e dai codici classici BCH quaternari auto-ortogonali. In questa sezione, si descrive nel dettaglio la costruzione di un codice QBCH $[n, k']$ dual-containing, sulla base delle considerazioni avanzate nella Sezione 6.1.

Si ricorda, dalla Sezione 6.1, che se C è il codice classico specificato dalla PCM \mathbf{H} allora il suo duale è C^\perp , il cui spazio di codice è contenuto in quello di C , cioè si ha: $C^\perp \subset C$. Dunque il codice CSS dual-containing con $[n, k']$ qubit risultante, dove $k' = (2k - n)$, mappa ciascuno dei $2^{k'}$ stati sovrapposti di una sequenza d'informazione, avente k' qubit, su un sottoinsieme unico del codice duale C^\perp nello spazio di codice di C . I sottoinsiemi di C^\perp in C possono essere ottenuti sommando (modulo 2) una parola di codice legittima di C a tutte le parole di codice di C^\perp , come precedentemente mostrato nell'eq.(6.4). Tuttavia, solo quelle parole di codice di C generano un sottoinsieme unico di C^\perp , che non differisce da alcun elemento di C . In particolare, si dice che le parole di codice x_1 e x'_1 di C differiscono da un elemento di C^\perp se la somma bit-a-bit modulo 2 tra loro, produce una parola di codice di C^\perp cioè tale che, $x_1 + x'_1 = x_2$, dove $x_2 \in C^\perp$. Di conseguenza, tali parole di codice di C permettono di ottenere lo stesso sottoinsieme di C^\perp .

7.1.4 Esempio Operativo: codice QBCH

Ora si elabora questa idea costruendo il codice QBCH $[15, 7]$, in grado di correggere un singolo errore, a partire dal codice classico dual-containing BCH(15, 11) il cui codificatore è mostrato in Fig. 7.2 e la cui PCM è:

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}. \quad (7.9)$$

Il codificatore del codice QBCH $[15, 7]$ può essere derivato utilizzando il metodo proposto da MacKay *et al.* in [98], che procede come segue:

1) La PCM classica \mathbf{H} dual-containing viene prima trasformata nella matrice $\tilde{\mathbf{H}} = [\mathbf{I}_{(n-k)}|\mathbf{P}]$, attraverso operazioni elementari sulle righe e permutazioni delle colonne. In particolare, le operazioni elementari di riga includono le permutazioni delle righe e la somma tra due righe. La somma degli elementi delle righe è modulo 2. Poiché \mathbf{H} è una matrice $(n-k) \times n$, la matrice identità risultante $\mathbf{I}_{(n-k)}$ ha dimensioni $(n-k) \times (n-k)$, mentre \mathbf{P} è una matrice binaria di dimensioni $(n-k) \times k$.

In particolare, nel caso in esame si ha la matrice identità \mathbf{I}_4 , di dimensioni 4×4 poiché $n-k = 15-11 = 4$. Inoltre, si osserva che le prime 4 colonne della matrice \mathbf{H} costituiscono già una matrice identità 4×4 , pertanto, la matrice binaria \mathbf{P} è semplicemente la restante parte di \mathbf{H} , e cioè corrisponde a:

$$\mathbf{P} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad (7.10)$$

quindi, effettivamente, non è necessario applicare operazioni sulle righe e sulle colonne.

Per quanto detto dunque, per la PCM \mathbf{H} dell'eq.(7.9), si ottiene che $\tilde{\mathbf{H}} = \mathbf{H}$, cioè:

$$\tilde{\mathbf{H}} = \mathbf{H} = \left(\begin{array}{cccc|cccccccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{array} \right). \quad (7.11)$$

2) Come passo successivo, si applicano (nuovamente) le operazioni di riga a \mathbf{P} , in modo che sia ridotta a $\tilde{\mathbf{P}} = [\mathbf{I}_{(n-k)}|\mathbf{Q}]$, dove \mathbf{Q} è una matrice binaria di dimensioni $(n-k) \times k'$.

In particolare, bisogna ottenere, come prima, la matrice identità \mathbf{I}_4 , dunque questa volta bisogna effettivamente andare a modificare la matrice \mathbf{P} con le operazioni di riga. Pertanto, si ottiene:

$$\tilde{\mathbf{P}} = \left(\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{array} \right), \quad (7.12)$$

dove quindi

$$\mathbf{Q} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}. \quad (7.13)$$

3) Il codificatore associato può essere implementato in due fasi, come mostrato nella Fig. 7.5. Nella prima fase (Stage 1), la matrice \mathbf{Q} agisce sul secondo blocco di $(n-k) = 4$ qubit ausiliari (o qubit di parità), cioè $|0\rangle$ controllati dagli ultimi $k' = (2k-n) = 7$ qubit di informazione ($|q\rangle$),

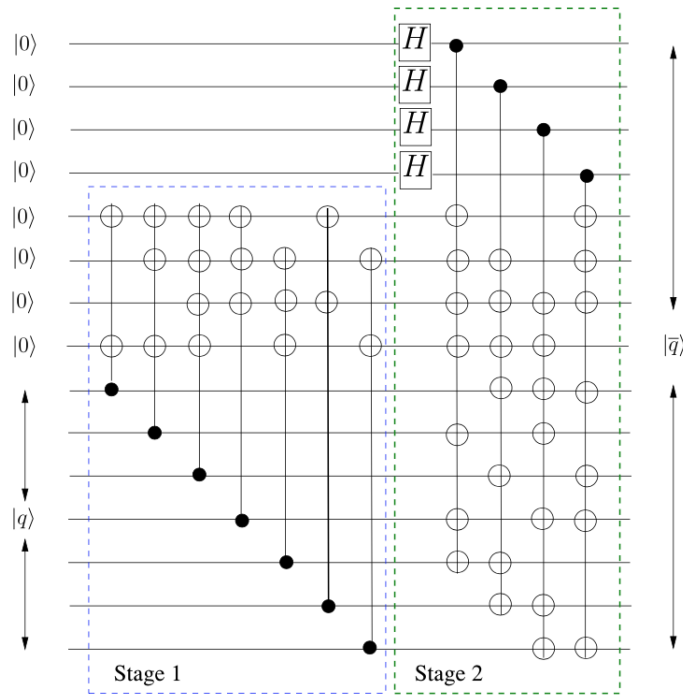


Figura 7.5: Codificatore per un codice QBCH[15, 7] [30].

che costituiscono la sequenza di informazione. In particolare, una porta Controlled NOT (CNOT) agisce sul qubit i -esimo del secondo blocco di $(n - k)$ qubit, che è controllato dal qubit d'informazione j -esimo. Inoltre, come è evidente, l'indice $1 \leq i \leq (n - k) = 4$ rappresenta le righe della matrice Q , mentre $1 \leq j \leq k' = (2k - n) = 7$ sta ad indicare le colonne della medesima matrice. Dunque, la presenza della porta CNOT che collega il qubit di parità i -esimo con il qubit di informazione j -esimo è determinata dal valore binario dell'elemento di pedice ij della matrice Q . In altre parole:

$$\begin{cases} Q_{ij} = 0 \rightarrow \text{assenza della porta CNOT} \\ Q_{ij} = 1 \rightarrow \text{presenza della porta CNOT.} \end{cases}$$

Quanto detto può essere formulato come segue:

$$|0\rangle^{\otimes(n-k)} |0\rangle^{\otimes(n-k)} |q\rangle \rightarrow |0\rangle^{\otimes(n-k)} |Qq\rangle |q\rangle, \quad (7.14)$$

dove l'apice $\otimes(n-k)$ indica che $|0\rangle$ viene replicato $(n - k)$ volte.

Gli stati risultanti costituiscono l'insieme delle parole di codice in \mathcal{C} , che non differiscono da alcun elemento di C^\perp e che quindi sono in grado di generare sottoinsiemi unici di C^\perp .

4) Nella seconda fase (Stage 2) si sommano le parole di codice di C^\perp con le parole di codice di \mathcal{C} , generate allo step precedente. Più specificamente, in questa sede si genera lo spazio di codice di C^\perp , secondo la PCM \tilde{H} . Per un codice classico che appartiene C^\perp , i primi $(n - k)$ bit sono i bit sistematici di informazione, che possono avere sia il valore 0 che il valore 1.

Tabella 7.2: Stabilizzatori del codice QBCH[15, 7] [12].

	Stabilizzatore
g_1	ZIIIZZZZIZIZZII
g_2	IIZIIIZZZZIZIZZI
g_3	IIIZIIIZZZZIZIZZZ
g_4	IIIZZZZIZIZZZIIZ
g_5	XIIIXXXXIXIXXII
g_6	IXIIIXXXXIXIXXI
g_7	IIXIIIXXXXIXIXX
g_8	IIIXXXXIXIXXIIIX

Per questo, i primi $(n - k) = 4$ qubit ausiliari subiscono una trasformazione di Hadamard, per generare lo spazio di codice completo del codice classico C^\perp . Per maggiori considerazioni sulla porta di Hadamard si guardi l'*Approfondimento: Operatore di Hadamard*, della Sezione 3.4. Infine, la matrice P agisce sugli ultimi $k = 11$ qubit, controllati, a loro volta, dai primi $(n - k) = 4$ qubit, generando quindi lo spazio di codice di C^\perp . In particolare, una porta CNOT agisce sul j -esimo qubit, che è controllato dall' i -esimo qubit, se $P_{ij} = 1$, dove, come nel caso precedente della matrice Q, l'indice $1 \leq i \leq (n - k) = 4$ rappresenta le righe della matrice P, mentre l'indice $1 \leq j \leq k = 11$ sta ad indicarne le colonne.

Gli stabilizzatori del codice QBCH[15, 7] sono costruiti utilizzando la PCM dell'eq.(7.9), sostituendo gli 1 con **Z** (o **X**), mentre gli 0 sono sostituiti con **I**, secondo l'isomorfismo dal dominio di Pauli al dominio classico binario, come spiegato nella Sezione 5.1. Gli stabilizer generator risultanti sono elencati nella Tabella 7.2. In particolare, dato che si sta progettando un codice QBCH CSS di tipo dual-containing, come specificato inizialmente, il modello per la costruzione della PCM è il seguente $\mathbf{H}_{\text{css}} = [\tilde{\mathbf{H}}_z \mid \tilde{\mathbf{H}}_x]$, dove $\tilde{\mathbf{H}}_z = \begin{pmatrix} \mathbf{0} \\ \mathbf{H}'_z \end{pmatrix}$ e $\tilde{\mathbf{H}}_x = \begin{pmatrix} \mathbf{0} \\ \mathbf{H}'_x \end{pmatrix}$ sono matrici binarie di dimensioni $n \times (n - k)$, come descritto nella Sezione 5.1. Dunque si osserva che $\mathbf{H}'_z = \mathbf{H}'_x$, poiché si tratta di un codice dual-containing. La PCM $\tilde{\mathbf{H}}$ dell'eq.(7.11) produce $\tilde{\mathbf{H}}\tilde{\mathbf{H}}^T = 0$, quindi si presta essa stessa a costruire un codice CSS di tipo dual-containing. Dunque, dato che vale $\mathbf{H}'_z = \tilde{\mathbf{H}}$, la matrice risultante è:

$$\mathbf{H}_{\text{css}} = \begin{pmatrix} \tilde{\mathbf{H}} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{H}} \end{pmatrix}, \quad (7.15)$$

a giustificazione del fatto che, all'inizio si era riportata solo la matrice $\tilde{\mathbf{H}}$, data la natura CSS di tipo dual-containing del codice. Per ulteriori approfondimenti, si faccia riferimento all'esempio di costruzione di un codice dual-containing nella Sottosezione 6.1.1

A causa della natura ciclica dei codici BCH, sia il codificatore della Fig. 7.5

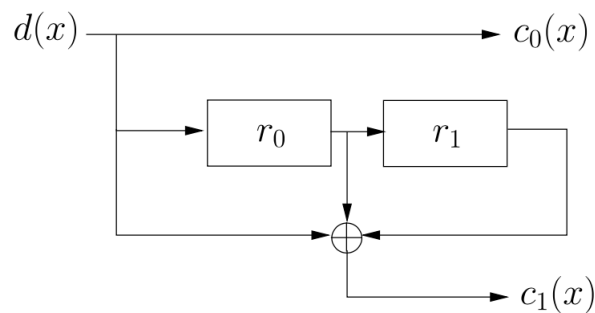


Figura 7.6: Schematico del codificatore del codice convoluzionale sistematico $(2, 1, 2)$ [12].

che gli stabilizer generator della Tabella 7.2 possono essere implementati utilizzando i registri a scorrimento quantistico, che a loro volta, rendono i codici QBCCH adatti a sistemi con simmetrie cicliche, ad esempio trappole a ioni circolari [67]. I valori della sindrome binaria, ottenuti utilizzando gli stabilizzatori della Tabella 7.2, alimentano poi un decodificatore di tipo Berlekamp-Massey classico, che stima l'errore più probabile.

7.2 Codici Convoluzionali

7.2.1 Codici classici Convoluzionali

Si ricorda che un codice a blocco $C(n, k)$, codifica ogni blocco di k bit di informazione, in modo indipendente, in n bit codificati.

Al contrario, un codice convoluzionale di parametri (n, k, m) , un esempio del quale, per un codice sistematico $(2, 1, 2)$ è riportato nella Fig. 7.6, codifica l'intera sequenza di informazione in una singola sequenza codificata. Più precisamente, ogni ingresso di k bit è codificato in n simboli, in modo che l'uscita codificata in ogni istante dipenda anche dai k bit di informazione ricevuti, negli m istanti di tempo precedenti. Il codice convoluzionale risultante ha quindi una memoria pari a m , o equivalentemente una lunghezza di vincolo di valore $(m + 1)$, che viene implementata utilizzando un opportuno numero di memorie. Inoltre, il codice è specificato da n polinomi generatori, che definiscono la topologia delle porte (modulo 2) per generare la sequenza codificata. In particolare, i polinomi generatori definiscono la connettività tra il valore corrente del bit di informazione e gli m bit di input precedenti.

Il codice convoluzionale sistematico $(2, 1, 2)$, il cui codificatore è rappresentato in Fig. 7.6, è specificato dai seguenti polinomi generatori:

$$\begin{aligned} g_0(x) &= 1; \\ g_1(x) &= 1 + x + x^2; \end{aligned} \quad (7.16)$$

per le generalità sui codici sistematici, si guardi l'*Approfondimento: Codici sistematici* della Sezione 2.1.2. In particolare, il codificatore della Fig. 7.6

Tabella 7.3: Processo di codifica di un codice convoluzionale sistematico con $(2, 1, 2)$ [12].

Indice	Input Bit d	State (r_0r_1) Binario	State (r_0r_1) Decimale	Output Bit c_0c_1
0	-	00	0	-
1	0	00	0	00
2	0	00	0	00
3	0	00	0	00
4	1	10	2	11
5	1	11	3	10
6	0	01	1	00
7	1	10	2	10
8	1	11	3	10
9	0	01	1	00
10	0	00	0	01

elabora $k = 1$ bit di informazione in $n = 2$ bit codificati, attraverso $m = 2$ registri di memoria r_0 e r_1 .

I polinomi generatori possono anche essere espressi come vettori binari, dove ogni bit indica la presenza o l'assenza di un collegamento. Di conseguenza, i polinomi generatori dell'eq.(7.16) si possono riportare anche in forma binaria, come segue:

$$\begin{aligned} g_0 &= (100); \\ g_1 &= (111). \end{aligned} \tag{7.17}$$

In particolare, il valore binario 1 o 0, posto nella posizione i -esima della sequenza binaria, rappresenta la presenza o l'assenza, rispettivamente, della potenza i -esima nel polinomio. Si può inoltre osservare che nell'eq.(7.17) g_0 ha un solo ingresso diverso da zero. Questo è causato dalla natura sistematica del codice.

7.2.2 Esempio Operativo: codice classico Convoluzionale

Si consideri ora una sequenza di input di 10 bit $d = 0011011000$, che può anche essere rappresentata in forma polinomiale come $d(x) = x^2 + x^3 + x^5 + x^6$. Questa sequenza di input è codificata in una sequenza di 20 bit, utilizzando il codificatore della Fig. 7.6, dove \oplus è l'operatore di somma modulo 2. Il processo di codifica associato è illustrato nella Tabella 7.3. In particolare, la Tabella 7.3 rappresenta il processo di codifica per il codice convoluzionale sistematico $(2, 1, 2)$, avente $d = 0011011000$ ($d(x) = x^2 + x^3 + x^5 + x^6$), che permette di ottenere la parola di codice $c = 01001010001011000000$, che può anche essere rappresentata in forma polinomiale come $c(x) = x + x^4 + x^6 + x^{10} + x^{12} + x^{13}$.

Inoltre, il registro a scorrimento è inizializzato allo stato all-zero, cioè tutte

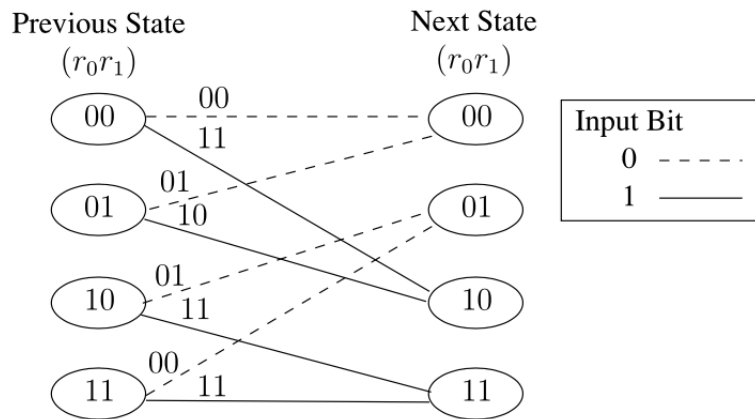


Figura 7.7: Diagramma di transizione di stato del codice convoluzionale sistematico (2, 1, 2) [12].

le celle di memoria sono impostate con valore nullo. Con ogni ciclo di clock, lo stato del registro r_0 viene aggiornato con il bit di informazione in entrata, mentre il suo valore precedente viene spostato sul registro successivo r_1 . Dunque, il bit di informazione in entrata d_i costituisce la parte sistematica c_0 della sequenza di bit codificata c , come mostra la Fig. 7.6, mentre l'uscita c_1 della somma modulo 2, di Fig. 7.6, rappresenta la parte di parità. In particolare, dalla Fig. 7.6 si mutuano le seguenti relazioni:

$$\begin{aligned} c_{0,i} &= d_i; \\ c_{1,i} &= d_i \oplus r_{0,i-1} \oplus r_{1,i-1}, \end{aligned} \quad (7.18)$$

dove il pedice i rappresenta la voce "indice" della Tabella 7.3.

Inoltre, analogamente ai codici BCH, l'operazione di codifica di un codice convoluzionale può anche essere realizzata utilizzando un diagramma di transizione di stato, come mostrato nella Fig. 7.7, per il codice convoluzionale (2, 1, 2) della Fig. 7.6.

Osservazione Le linee tratteggiate della Fig. 7.7 indicano le transizioni legittime a causa di un input di valore 0 (ingresso nullo), mentre le linee continue rappresentano un input con valore 1. Inoltre, le transizioni sono etichettate con i bit codificati $(c_0 c_1)$.

I codici convoluzionali utilizzano tecniche di decodifica a traliccio, ad esempio l'algoritmo di Viterbi [167] o l'algoritmo MAP [18], la cui complessità di decodifica è proporzionale al numero 2^m degli stati del traliccio.

7.2.3 Codici quantistici Convoluzionali

I codici quantistici convoluzionali (Quantum Convolutional Codes - QCC) possono essere progettati a partire dai codici classici convoluzionali, sfrut-

tando la loro natura di blocco semi-infinito. In altre parole, i codici convoluzionali possono essere rappresentati come codici a blocco lineari di lunghezza semi-infinita [95]. Questa equivalenza, a sua volta, permette di costruire le controparti quantistiche dei codici classici convoluzionali noti, basate sugli stabilizzatori.

Per prima cosa si elabora la struttura dei codici convoluzionali, costituita da blocchi di lunghezza semi-infinita, usando, in questo caso, un codice classico convoluzionale di parametri $(2, 1, m)$, avente i seguenti generatori:

$$\begin{aligned} g_0 &= (g_0^{(0)} g_0^{(1)} \dots g_0^{(m)}); \\ g_1 &= (g_1^{(0)} g_1^{(1)} \dots g_1^{(m)}). \end{aligned} \quad (7.19)$$

In sostanza, i polinomi generatori g_0 e g_1 descrivono le funzioni di risposta impulsiva del codificatore. Inoltre, esse sono convolute con la sequenza di input $[d = (d_0 d_1 d_2 \dots)]$, per produrre le sequenze di bit codificate di output $[c_0 = (c_0^{(0)} c_0^{(1)} c_0^{(2)} \dots)]$ e $[c_1 = (c_1^{(0)} c_1^{(1)} c_1^{(2)} \dots)]$, rispettivamente. Questo processo di codifica può essere matematicamente espresso come segue:

$$\begin{aligned} c_0 &= d \otimes g_0; \\ c_1 &= d \otimes g_1, \end{aligned} \quad (7.20)$$

dove \otimes rappresenta l'operazione di convoluzione discreta modulo 2. Il processo di convoluzione dell'eq.(7.20) può anche essere espresso nella seguente forma:

$$c_j^{(l)} = \sum_{i=0}^m d_{l-i} g_j^{(i)} = d_l g_j^{(0)} + d_{l-1} g_j^{(1)} + \dots + d_{l-m} g_j^{(m)}, \quad (7.21)$$

dove $j = 0, 1$, $l \geq 0$ e $u_{l-i} \triangleq 0$, $\forall l < i$. Infine, le due sequenze codificate c_0 e c_1 sono multiplate tra loro, ottenendo una unica sequenza codificata c , come segue:

$$c = (c_0^{(0)} c_1^{(0)} c_0^{(1)} c_1^{(1)} c_0^{(2)} c_1^{(2)} \dots). \quad (7.22)$$

Il processo di codifica dell'eq.(7.21) può anche essere rappresentato nella notazione matriciale, come segue:

$$c = d\mathbf{G}. \quad (7.23)$$

La matrice generatrice \mathbf{G} è costruita a partire dai polinomi generatori g_0 e g_1 come segue:

$$\mathbf{G} = \begin{pmatrix} g_{01}^{(0)} & g_{01}^{(1)} & \dots & g_{01}^{(m)} & 0 & 0 \\ 0 & g_{01}^{(0)} & g_{01}^{(1)} & \dots & g_{01}^{(m)} & 0 \\ 0 & 0 & g_{01}^{(0)} & g_{01}^{(1)} & \dots & g_{01}^{(m)} \\ 0 & 0 & 0 & \ddots & \dots & \ddots \end{pmatrix}, \quad (7.24)$$

dove $g_{01}^{(i)} \triangleq (g_0^{(i)} g_1^{(i)})$. La matrice \mathbf{G} , risultante dall'eq.(7.24), ha lunghezza semi-infinita, poiché la sequenza di input d può avere lunghezza arbitraria.

Inoltre, si può osservare che l' i -esima riga di \mathbf{G} si ottiene spostando la riga $(i - 1)$ -esima a destra di $n = 2$ posti, dato che si sta lavorando su un codice classico di parametri $(2, 1, m)$. In particolare, quando d è troncato per avere una lunghezza finita pari ad N , allora la matrice \mathbf{G} dell'eq.(7.24) è di dimensioni $(N \times 2(m + N))$.

In generale, per un codice convoluzionale, avente i parametri (n, k, m) , la matrice generatrice \mathbf{G} può essere espressa come:

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}^{(0)} & \mathbf{G}^{(1)} & \dots & \mathbf{G}^{(m)} & 0 & 0 \\ 0 & \mathbf{G}^{(0)} & \mathbf{G}^{(1)} & \dots & \mathbf{G}^{(m)} & 0 \\ 0 & 0 & \mathbf{G}^{(0)} & \mathbf{G}^{(1)} & \dots & \mathbf{G}^{(m)} \\ 0 & 0 & 0 & \ddots & \dots & \ddots \end{pmatrix}, \quad (7.25)$$

dove la matrice $\mathbf{G}^{(l)}$ è definita come:

$$\mathbf{G}^{(l)} = \begin{pmatrix} g_{1,1}^{(l)} & g_{1,2}^{(l)} & \dots & g_{1,n-1}^{(l)} \\ g_{2,1}^{(l)} & g_{2,2}^{(l)} & \dots & g_{2,n-1}^{(l)} \\ \vdots & \vdots & \dots & \vdots \\ g_{k,1}^{(l)} & g_{k,2}^{(l)} & \dots & g_{k,n-1}^{(l)} \end{pmatrix}, \quad (7.26)$$

dunque ha dimensioni $k \times (n - 1)$.

Anche la PCM \mathbf{H} di un codice convoluzionale può essere espressa come una matrice semi-infinita, simile alla matrice generatrice \mathbf{G} nell'eq.(7.25), come mostrato di seguito:

$$\mathbf{H} = \begin{pmatrix} \mathbf{H}^{(0)} & 0 & 0 & 0 & 0 & 0 \\ \mathbf{H}^{(1)} & \mathbf{H}^{(0)} & 0 & 0 & 0 & 0 \\ \mathbf{H}^{(2)} & \mathbf{H}^{(1)} & \mathbf{H}^{(0)} & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \\ \mathbf{H}^{(m)} & \mathbf{H}^{(m-1)} & \mathbf{H}^{(m-2)} & \dots & \mathbf{H}^{(0)} & 0 \\ 0 & \mathbf{H}^{(m)} & \mathbf{H}^{(m-1)} & \mathbf{H}^{(m-2)} & \dots & \mathbf{H}^{(0)} \\ 0 & 0 & \vdots & \vdots & \dots & \vdots \end{pmatrix}, \quad (7.27)$$

dove $\mathbf{H}^{(l)}$ è una sottomatrice di dimensioni $((n - k) \times n)$. La PCM \mathbf{H} nell'eq.(7.27) presenta una struttura a *banda di blocchi*, illustrata nella Fig. 7.8.

In particolare, se ogni riga delle sottomatrici $(\mathbf{H}^{(m)} \mathbf{H}^{(m-1)} \mathbf{H}^{(m-2)} \dots \mathbf{H}^{(0)})$ è vista come un blocco singolo, allora \mathbf{H} è caratterizzata da una struttura a banda di blocchi, in cui cioè ogni blocco è una versione spostata nel tempo del blocco precedente e i blocchi successivi hanno m sottomatrici sovrapposte. Questa struttura a banda di blocchi, che appare dopo i primi m blocchi, è espressa come:

$$h_{j,i} = [\mathbf{0}^{j \times n}, h_{0,i}], \quad 1 \leq i \leq (n - k), \quad 0 \leq j, \quad (7.28)$$

dove i denota l'indice della riga all'interno di un blocco, mentre j è l'indice del blocco stesso. Inoltre, $\mathbf{0}^{j \times n}$ è un vettore riga di dimensione $(j \times n)$,

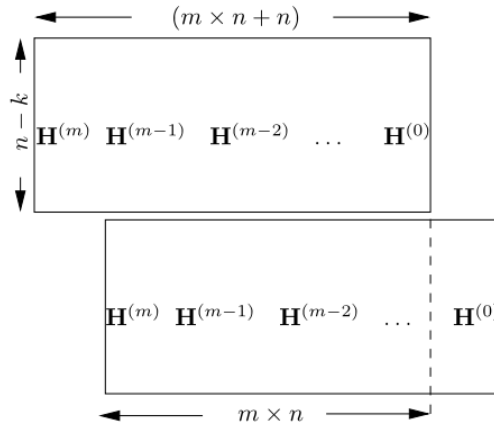


Figura 7.8: Rappresentazione della PCM \mathbf{H} classica semi-infinita, avente una struttura a banda di blocchi [12].

completamente costituito da zeri.

Dualmente rispetto all'eq.(7.28), il gruppo stabilizzatore \mathcal{H} di un QCC con $[n, k, m]$ qubit, può essere formulato come segue [120]:

$$\mathcal{H} = sp\{g_{j,i} = I^{\otimes jn} \otimes g_{0,i}\}, \quad 1 \leq i \leq (n - k), \quad 0 \leq j, \quad (7.29)$$

dove sp denota un gruppo simplettico.

7.2.4 Esempio Operativo: codice quantistico Convoluzionale

Esempio 1

Si progetta un QCC di tipo CSS con rate 1/3 [58], [59], a partire da un codice classico convoluzionale binario auto-ortogonale, cioè dual-containing, con rate 2/3, avente la seguente PCM:

$$\mathbf{H} = \left(\begin{array}{ccc|ccc|ccc|ccc|...} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & \dots \\ \dots & & & \dots & & & \dots & & & \dots & & & \dots \end{array} \right), \quad (7.30)$$

e una distanza minima d_{min} pari a 3. Si tratta di una matrice semi-infinita con struttura a banda di blocchi, dove i blocchi di $n = 3$ elementi sono shiftati di 1 posizione verso destra; dunque ogni singolo elemento è spostato di $n = 3$ posizioni verso destra. La matrice di parità corrispondente, che rappresenta il QCC nel dominio di Pauli, è descritta dall'eq.(6.1). Dato che si ricava il QCC a partire da un codice classico auto-ortogonale, cioè dual-containing, anche il QCC associato sarà dual-containing, quindi vale che:

$$\mathbf{H}_{css} = [\mathbf{H}_z \mid \mathbf{H}_x] = \left(\begin{array}{c|c} \mathbf{H}'_z & \mathbf{0} \\ \mathbf{0} & \mathbf{H}'_x \end{array} \right) = \left(\begin{array}{c|c} \mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{array} \right), \quad (7.31)$$

dove $\mathbf{H}'_z = \mathbf{H}'_x = \mathbf{H}$, dato che il codice classico di partenza è auto-ortogonale. In particolare, anche \mathbf{H}_{css} conserva la natura di matrice semi-infinita con

struttura a banda di blocchi.

Gli stabilizzatori corrispondenti di un QCC di tipo CSS, come visto nella Sezione 6.1, possono essere ottenuti sostituendo gli 1, dell'eq.(7.30), con gli operatori di Pauli \mathbf{X} e \mathbf{Z} , rispettivamente. Quindi, gli $n-k = 2$ stabilizzatori del QCC con $[3, 1]$ qubit risultante, sono:

$$\begin{aligned} g_{0,1} &= [\mathbf{XXX}, \mathbf{XII}, \mathbf{XXI}]; \\ g_{0,2} &= [\mathbf{ZZZ}, \mathbf{ZII}, \mathbf{ZZI}], \end{aligned} \quad (7.32)$$

secondo il mapping dal dominio classico binario al dominio di Pauli.

Questo codice è in grado di correggere $t = (d_{min} - 1)/2 = (3 - 1)/2 = 1$ errori, cioè un singolo errore. Infine, il gruppo stabilizzatore associato \mathcal{H} può essere costruito utilizzando l'eq.(7.29).

Esempio 2

In questo secondo esempio si mostra il progetto di un QCC non CSS di tipo CRSS, studiato da Forney in [59] e [58]. È costruito a partire dal codice classico convoluzionale quaternario con rate $2/3$ che è caratterizzato dalla seguente PCM classica:

$$\mathbf{H} = \left(\begin{array}{ccc|ccc|ccc|...} 1 & 1 & 1 & 1 & w & \bar{w} & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & w & \bar{w} & \dots \\ \dots & & & \dots & & & \dots & & & \dots \end{array} \right), \quad (7.33)$$

che è auto-ortogonale. Come prima operazione per calcolare gli stabilizzatori associati al QCC $[3, 1]$, si applica l'eq.(6.10), dunque si ottiene:

$$\mathbf{H}_{\text{crss}} = \begin{pmatrix} \mathbf{H} \\ w\mathbf{H} \end{pmatrix} = \left(\begin{array}{ccc|ccc|ccc|...} 1 & 1 & 1 & 1 & w & \bar{w} & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & w & \bar{w} & \dots \\ w & w & w & w & \bar{w} & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & w & w & w & w & \bar{w} & 1 & \dots \\ \dots & & & \dots & & & \dots & & & \dots \end{array} \right), \quad (7.34)$$

che, al pari di \mathbf{H} nell'eq.(7.33), è una matrice semi-infinita con struttura a banda di blocchi, dove i blocchi di $n = 3$ elementi sono shiftati di 1 posizione verso destra. Dunque, anche in questo caso, ogni singolo elemento è spostato di $n = 3$ posizioni verso destra. In particolare, la terza e la quarta riga della matrice \mathbf{H}_{crss} dell'eq.(7.34) sono ottenute moltiplicando la \mathbf{H} dell'eq.(7.33) con l'elemento di $\text{GF}(4)$ w . Dunque, data la natura della matrice \mathbf{H}_{crss} , anche gli stabilizzatori sono delle strutture di lunghezza minima, che si ripetono nella matrice. Pertanto, gli stabilizzatori $g_{0,i}$, per $1 \leq i \leq 2$, si ottengono mappando gli elementi di $\text{GF}(4)$ risultanti, della matrice \mathbf{H}_{crss} , sugli operatori di Pauli, secondo il mapping dal dominio classico quaternario al dominio di Pauli, come spiegato nella Sezione 5.2. Considerando allora, come strutture che si ripetono, i primi due blocchi da tre elementi della prima e della terza riga, $g_{0,1}$ e $g_{0,2}$ rispettivamente, cioè:

$$\begin{aligned} g_{0,1} &= (111, 1w\bar{w}); \\ g_{0,2} &= (www, w\bar{w}1), \end{aligned} \quad (7.35)$$

gli stabilizzatori risultanti nel dominio di Pauli sono:

$$\begin{aligned} g_{0,1} &= (\mathbf{XXX}, \mathbf{XZY}); \\ g_{0,2} &= (\mathbf{ZZZ}, \mathbf{ZYX}), \end{aligned} \quad (7.36)$$

secondo l'opportuno isomorfismo.

Nota: nella Sezione 5.2, come elementi di $\text{GF}(4)$ si usano $\{0, 1, \omega, \bar{\omega}\}$, mentre in questo esempio operativo si utilizzano $\{0, 1, w, \bar{w}\}$. Naturalmente le proprietà intrinseche di w e \bar{w} sono le stesse di ω e $\bar{\omega}$, rispettivamente.

Analogamente ad altri codici stabilizzatori, i valori della sindrome binaria, ottenuti utilizzando gli stabilizzatori del QCC, servono ad alimentare un decodificatore basato sulla sindrome classica.

Tuttavia, i codici classici convoluzionali impiegano generalmente l'algoritmo di decodifica di Viterbi [167] o l'algoritmo MAP [18], che opera su un traliccio, al fine di stimare la parola di codice più probabile. Al contrario, i QCC utilizzano il traliccio dell'errore basato sulla sindrome [140], [141], [142], [7] e [147] per stimare il modello di errore più probabile piuttosto che la parola di codice più probabile. Esplicitamente, a differenza di un codice classico convoluzionale, visto nella Fig. 7.7, che è costruito utilizzando il circuito di codifica, il traliccio basato sulla sindrome è costruito usando la PCM \mathbf{H} dell'eq.(7.27). Inoltre, il traliccio convenzionale, per esempio quello ottenuto utilizzando il diagramma di transizione di stato di Fig. 7.7, è noto come *traliccio del codice* (code trellis), poiché ogni possibile suo percorso è una parola di codice valida. Al contrario, ogni percorso del *traliccio dell'errore* (error trellis) è una sequenza di errore legittima per una data sindrome osservata. Pertanto, il traliccio del codice viene utilizzato per la decodifica delle parole di codice, mentre il traliccio che rappresenta l'errore viene utilizzato per la decodifica della sindrome. Tuttavia, entrambe le rappresentazioni del traliccio sono equivalenti, poiché ogni percorso nel traliccio d'errore corrisponde ad un percorso nel traliccio di codice. Inoltre in [122], è stato proposto un algoritmo di decodifica di Viterbi degenerare anche per i QCC, che tiene conto degli errori quantistici degeneri, migliorando quindi il loro il processo di decodifica.

Conclusioni

Conclusione e linee guida per la progettazione

I QECC (Quantum Error Correction Codes) sono essenziali per rappresentare le perturbazioni indesiderate dovute alla decoerenza quantistica. Sfortunatamente però, la teoria della codifica classica, che si è evoluta sviluppandosi nell'arco di sette decenni, non può essere applicata direttamente ai codici del dominio quantistico. In particolare, a differenza del generico bit classico, il qubit non può essere copiato e collassa al valore di un bit classico dopo essere stato misurato. Inoltre, mentre l'inversione di bit (bit-flip) è l'unico tipo di errore che si verifica durante la trasmissione dell'informazione su un canale classico, un canale quantistico può determinare sia errori di tipo bit-flip, sia errori di inversione di fase (phase-flip). Pertanto, non è possibile mappare direttamente i codici classici sulle loro controparti quantistiche.

Tuttavia, i codici quantistici possono essere progettati a partire dai codici classici esistenti sfruttando le delicate somiglianze tra questi due domini di codifica. In particolare, come descritto nella Sezione 1.3, la decoerenza quantistica può essere modellata usando il canale di depolarizzazione quantistico, che è considerato equivalente a una coppia di canali simmetrici binari, o più specificamente ad un canale classico quaternario, come spiegato nella Sezione 1.3.3.

Questa somiglianza ha permesso ai ricercatori di sviluppare le versioni quantistiche dei codici classici noti, come evidente dall'indagine condotta nel Capitolo 2. Al fine di fornire approfondimenti sulla transizione dalla teoria della codifica classica a quella quantistica, si sono iniziate le dovute analisi nel Capitolo 3 con un semplice codice a ripetizione a 3 qubit, che ha portato alla luce tre principi di progettazione fondamentali:

- l'operazione di copia dei codici classici è equivalente all'entanglement quantistico;
- la misurazione di un qubit può essere aggirata utilizzando le tecniche di decodifica della sindrome classica;
- gli errori di tipo phase-flip possono essere corretti utilizzando le basi di Hadamard.

Sulla base di questi principi di progettazione, nel Capitolo 4 si è sviluppato il formalismo stabilizzatore, che è, in sostanza, la controparte, nel dominio quantistico, dei codici classici a blocco lineari. Poiché molti codici classici si basano sul modello di base della codifica a blocco lineare, il formalismo

stabilizzatore ha permesso ai ricercatori di costruire codici quantistici, a partire dalle famiglie di codice classiche conosciute.

Nel Capitolo 5, si è studiata l'equivalenza tra la matrice di parità classica e quella quantistica, concentrandosi in particolare, sull'isomorfismo dal dominio di Pauli, come espressione del dominio quantistico, al dominio classico binario e sull'isomorfismo dal dominio di Pauli a quello classico quaternario. L'isomorfismo dal dominio di Pauli al dominio classico binario aiuta a progettare i codici quantistici, partendo da codici classici binari arbitrari, se soddisfano il criterio del prodotto simplettico, mentre l'isomorfismo dal dominio di Pauli al dominio classico quaternario permette di sfruttare i codici classici quaternari arbitrari, che soddisfano il criterio sul prodotto scalare Hermitiano.

Inoltre, sulla base di questo isomorfismo, si è presentata la tassonomia dei codici stabilizzatori nel Capitolo 6, vale a dire i codici dual-containing e non-dual-containing di Calderbank-Shor-Steane (CSS), i codici non CSS e i codici entanglement-assisted (EA), riassunti, per comodità, nella Tabella 7.4. Infine, nel Capitolo 7, si sono applicate le nozioni studiate nei capitoli precedenti a una coppia di famiglie di codici popolari nel mondo classico, cioè i codici BCH e i codici convoluzionali, per progettare le loro controparti quantistiche.

Tabella 7.4: Linee guida di progettazione per la costruzione di codici quantistici stabilizzatori [12].

Tipo di Codice e Codificatore	Matrice di parità	Criteri di Progetto	Esempio di Progetto	Esempio di Progetto
	PCM \mathbf{H}		<i>Classico</i>	<i>Quantistico</i>
Dual-containing CSS <i>Binario</i>	$\left(\begin{array}{c c} \mathbf{H}'_z & \mathbf{0} \\ \mathbf{0} & \mathbf{H}'_z \end{array} \right)$	$\mathbf{H}'_z \mathbf{H}'_z{}^T = 0$	Codice di Hamming (7, 4) Codice BCH (15, 11) Codice Convolutionale (3, 1, 2)	Codice di Steane [7, 1] (Sezione 7.2) Codice QBCH[15, 7] (Sezione 7.1) Codice QCC [3, 1, 2] (Sezione 7.2)
Non-dual-containing CSS <i>Binario</i>	$\left(\begin{array}{c c} \mathbf{H}'_z & \mathbf{0} \\ \mathbf{0} & \mathbf{H}'_x \end{array} \right)$	$\mathbf{H}'_z \neq \mathbf{H}'_x$ e $\mathbf{H}'_z \mathbf{H}'_x{}^T = 0$	Codice a Ripetizione (3, 1)	Codice di Shor [9, 1] (Sezione 5.1)
Non-CSS <i>Non binario</i>	$(\mathbf{H}_z \mathbf{H}_x)$	$\mathbf{H}_z \mathbf{H}_x{}^T + \mathbf{H}_x \mathbf{H}_z{}^T = 0$	Codice di Hamming non binario (5, 3) Codice Convolutionale non Binario (3, 1, 2)	Codice di Hamming [5, 1] (Sezione 6.2) Codice QCC [3, 1, 2] (Sezione 7.2)
EA <i>Binario e non binario</i>	$\left(\begin{array}{c c} \mathbf{H}'_z & \mathbf{0} \\ \mathbf{0} & \mathbf{H}'_z \end{array} \right)$ e $(\mathbf{H}_z \mathbf{H}_x)$	Minimizzare il numero degli ebit		EA-QSC arbitrario (Sezione 6.3)

Bibliografia

- [1] Pagina Wikipedia "Codici Convolutionali", 2021. Ultimo accesso 19/02/2022.
- [2] Pagina Wikipedia "Sfera di Bloch", 2021. Ultimo accesso 08/03/2022.
- [3] Pagina Wikipedia "Notazione bra-ket", 2022. disponibile in data 19/02/2022.
- [4] Ömer Acikel and William Ryan. Punctured turbo-codes for BP-SK/QPSK channels. *IEEE Transactions on Communications*, 47(9):1315–1323, 1999.
- [5] Jos Akhtman, Robert Maunder, Nicholas Bonello, and Lajos Hanzo. Closed-form approximation of maximum free distance for binary block codes. In *2009 IEEE 70th Vehicular Technology Conference Fall*, pages 1–3. IEEE, 2009.
- [6] Iryna Andriyanova, Denise Maurice, and Jean-Pierre Tillich. Spatially coupled quantum LDPC codes. In *2012 IEEE Information Theory Workshop*, pages 327–331. IEEE, 2012.
- [7] Meir Ariel and Jakov Snyders. Soft syndrome decoding of binary convolutional codes. *IEEE Transactions on Communications*, 43(2/3/4):288–297, 1995.
- [8] Erdal Arıkan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, 2009.
- [9] Alexei Ashikhmin and Simon Litsyu. Upper bounds on the size of quantum codes. *IEEE Transactions on Information Theory*, 45(4):1206–1215, 1999.
- [10] Zunaira Babar, Panagiotis Botsinis, Dimitrios Alanis, Soon Xin Ng, and Lajos Hanzo. Fifteen years of quantum LDPC coding and improved decoding strategies. *IEEE Access*, 3:2492–2519, 2015.
- [11] Zunaira Babar, Panagiotis Botsinis, Dimitrios Alanis, Soon Xin Ng, and Lajos Hanzo. The road from classical to quantum codes: A hashing bound approaching design procedure. *IEEE Access*, 3:146–176, 2015.

- [12] Zunaira Babar, Daryus Chandra, Hung Viet Nguyen, Panagiotis Botsinis, Dimitrios Alanis, Soon Xin Ng, and Lajos Hanzo. Duality of quantum and classical error correction codes: Design principles and examples. *IEEE Communications Surveys Tutorials*, 21(1):970–1010, 2019.
- [13] Zunaira Babar, Soon Xin Ng, and Lajos Hanzo. Near-capacity code design for entanglement-assisted classical communication over quantum depolarizing channels. *IEEE Transactions on Communications*, 61(12):4801–4807, 2013.
- [14] Zunaira Babar, Soon Xin Ng, and Lajos Hanzo. Exit-chart aided code design for symbol-based entanglement-assisted classical communication over quantum channels. In *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, pages 1–5. IEEE, 2014.
- [15] Zunaira Babar, Soon Xin Ng, and Lajos Hanzo. Exit-chart-aided near-capacity quantum turbo code design. *IEEE Transactions on Vehicular Technology*, 64(3):866–875, 2014.
- [16] Zunaira Babar, Hung Viet Nguyen, Panagiotis Botsinis, Dimitrios Alanis, Daryus Chandra, Soon Xin Ng, and Lajos Hanzo. Serially concatenated unity-rate codes improve quantum codes without coding-rate reduction. *IEEE Communications Letters*, 20(10):1916–1919, 2016.
- [17] Zunaira Babar, Hung Viet Nguyen, Panagiotis Botsinis, Dimitrios Alanis, Daryus Chandra, Soon Xin Ng, Robert Maunder, and Lajos Hanzo. Fully-parallel quantum turbo decoder. *IEEE Access*, 4:6073–6085, 2016.
- [18] Lalit Bahl, John Cocke, Frederick Jelinek, and Josef Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, 20(2):284–287, 1974.
- [19] Almut Beige, Berthold-Georg Englert, Christian Kurtsiefer, and Harald Weinfurter. Secure communication with single-photon two-qubit states. *Journal of Physics A: Mathematical and General*, 35(28):L407–L413, 2002.
- [20] Charles Henry Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *arXiv preprint arXiv:2003.06557*, 2020.
- [21] Charles Henry Bennett, David DiVincenzo, John Smolin, and William Wootters. Mixed-state entanglement and quantum error correction. *Physical Review A*, 54:3824–3851, Nov 1996.
- [22] Charles Henry Bennett and Stephen Wiesner. Communication via one-and two-particle operators on Einstein-Podolsky-Rosen states. *Physical Review Letters*, 69(20):2881, 1992.

- [23] Nevio Benvenuto and Michele Zorzi. *Principles of Communications Networks and Systems*. John Wiley & Sons, 2011.
- [24] Elwyn Ralph Berlekamp. On decoding binary Bose-Chadhuri-Hocquenghem codes. *IEEE Transactions on Information Theory*, 11(4):577–579, 1965.
- [25] Claude Berrou and Alain Glavieux. Near optimum error correcting coding and decoding: turbo-codes. *IEEE Transactions on Communications*, 44(10):1261–1271, 1996.
- [26] Claude Berrou, Alain Glavieux, and Punya Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Proceedings of ICC '93 - IEEE International Conference on Communications*, volume 2, pages 1064–1070 vol.2, 1993.
- [27] Raj Chandra Bose and Dwijendra Kumar Ray-Chaudhuri. Further results on error correcting binary group codes. *Information and Control*, 3(3):279–290, 1960.
- [28] Raj Chandra Bose and Dwijendra Kumar Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and Control*, 3(1):68–79, 1960.
- [29] Kim Boström and Timo Felbinger. Deterministic secure direct communication using entanglement. *Physical Review Letters*, 89(18):187902, 2002.
- [30] Panagiotis Botsinis, Zunaira Babar, Dimitrios Alanis, Daryus Chandra, Hung Nguyen, Soon Xin Ng, and Lajos Hanzo. Quantum error correction protects quantum search algorithms against decoherence. *Scientific reports*, 6(1):1–13, 2016.
- [31] Garry Bowen. Entanglement required in achieving entanglement-assisted channel capacities. *Physical Review A*, 66(5):052313, 2002.
- [32] Todd Brun, Igor Devetak, and Min-Hsiu Hsieh. Correcting quantum errors with entanglement. *Science*, 314(5798):436–439, 2006.
- [33] Todd Brun, Igor Devetak, and Min-Hsiu Hsieh. General entanglement-assisted quantum error-correcting codes. In *2007 IEEE International Symposium on Information Theory*, pages 2101–2105, 2007.
- [34] Giuseppe Caire, Giorgio Taricco, and Ezio Biglieri. Bit-interleaved coded modulation. *IEEE Transactions on Information Theory*, 44(3):927–946, 1998.

- [35] Robert Calderbank, Eric Rains, Peter Shor, and Neil James Sloane. Quantum error correction via codes over $\text{GF}(4)$. *IEEE Transactions on Information Theory*, 44(4):1369–1387, 1998.
- [36] Robert Calderbank and Peter Willstone Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54:1098–1105, Aug 1996.
- [37] Thomas Camara, Harold Ollivier, and Jean-Pierre Tillich. Constructions and performance of classes of quantum LDPC codes, 2005.
- [38] Thomas Camara, Harold Ollivier, and Jean-Pierre Tillich. A class of quantum LDPC codes: construction and performances under iterative decoding. In *2007 IEEE International Symposium on Information Theory*, pages 811–815, 2007.
- [39] Daryus Chandra, Zunaira Babar, Hung Viet Nguyen, Dimitrios Alanis, Panagiotis Botsinis, Soon Xin Ng, and Lajos Hanzo. Quantum coding bounds and a closed-form approximation of the minimum distance versus quantum coding rate. *IEEE Access*, 5:11557–11581, 2017.
- [40] David Chase. Class of algorithms for decoding block codes with channel measurement information. *IEEE Transactions on Information Theory*, 18(1):170–182, 1972.
- [41] Isaac Chuang, Debbie Leung, and Yoshihisa Yamamoto. Bosonic quantum codes for amplitude damping. *Physical Review A*, 56:1114–1125, Aug 1997.
- [42] Arturo Di Corinto. Hp Wolf Security, Che cos'è l'Internet quantistica e perché apre una nuova era per la sicurezza informatica, 2021. Ultimo accesso 19 Febbraio 2022.
- [43] Daniele Cuomo, Marcello Caleffi, and Angela Sara Cacciapuoti. Towards a distributed quantum computing ecosystem. *IET Quantum Communication*, 1(1):3–8, 2020.
- [44] David Deutsch. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.
- [45] Igor Devetak. The private classical capacity and quantum capacity of a quantum channel. *IEEE Transactions on Information Theory*, 51(1):44–55, 2005.
- [46] Paul Dirac. *The Principles of Quantum Mechanics*, (London, 1958). 208(5):48, 1963.

- [47] David DiVincenzo, Peter Willstone Shor, and John Smolin. Quantum-channel capacity of very noisy channels. *Physical Review A*, 57:830–839, Feb 1998.
- [48] Dariush Divsalar, Sam Dolinar, and Fabrizio Pollara. Serial concatenated trellis coded modulation with rate-1 inner code. In *Globecom '00 - IEEE. Global Telecommunications Conference. Conference Record*, volume 2, pages 777–782 vol.2, 2000.
- [49] Albert Einstein, Max Born, Hedwig Born, et al. Born-Einstein Letters, 1971. Disponibile come libro sul sito della Springer.
- [50] Artur Ekert and Chiara Macchiavello. Quantum error correction for communication. *Physical Review Letters*, 77(12):2585, 1996.
- [51] Mohammed El-Hajjar and Lajos Hanzo. Exit charts for system design and analysis. *IEEE Communications Surveys Tutorials*, 16(1):127–153, 2014.
- [52] Gabriele Elia. Speciale: Quantum technologies, 2020. Sito web del gruppo TIM, ultimo accesso 19 Febbraio 2022.
- [53] Peter Elias. Coding for noisy channels. *IRE Convention Record*, 3:37–46, 1955.
- [54] Berlekamp Elwyn Ralph. Algebraic coding theory. *McGraw-Hill, New York. MR*, 38:6873, 1968.
- [55] TS ETSI. LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding. *ETSI TS 136 212 V12.2.0 Release 12*, 2014.
- [56] TS ETSI. Universal mobile telecommunications systems (UMTS); Multiplexing and channel coding (FDD). *ETSI TS*, 125(212):v12, 2015.
- [57] Jimenez Felstrom and Kamil Zigangirov. Time-varying periodic convolutional codes with low-density parity-check matrix. *IEEE Transactions on Information Theory*, 45(6):2181–2191, 1999.
- [58] David Forney, Markus Grassl, and Saikat Guha. Convolutional and tail-biting quantum error-correcting codes. *IEEE Transactions on Information Theory*, 53(3):865–880, 2007.
- [59] David Forney and Saikat Guha. Simple rate-1/3 convolutional and tail-biting quantum error-correcting codes. In *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005*, pages 1028–1032, 2005.
- [60] Keisuke Fujii. *Quantum Computation with Topological Codes: from qubit to topological fault-tolerance*, volume 8. Springer, 2015.

- [61] Robert Gray Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1):21–28, 1962.
- [62] Joydip Ghosh, Austin Fowler, and Michael Geller. Surface code with decoherence: An analysis of three superconducting architectures. *Physical Review A*, 86(6):062318, 2012.
- [63] Edgar Gilbert. A comparison of signalling alphabets. *The Bell System Technical Journal*, 31(3):504–522, 1952.
- [64] Daniel Gottesman. Class of quantum error-correcting codes saturating the quantum Hamming bound. *Physical Review A*, 54:1862–1868, Sep 1996.
- [65] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction*. California Institute of Technology, 1997.
- [66] Markus Grassl and Thomas Beth. Quantum BCH Codes, 1999.
- [67] Markus Grassl and Thomas Beth. Cyclic quantum error-correcting codes and quantum shift registers. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 456(2003):2689–2706, 2000.
- [68] Markus Grassl, Thomas Beth, and Thomas Pellizzari. Codes for the quantum erasure channel. *Physical Review A*, 56:33–38, Jul 1997.
- [69] Markus Grassl, Willi Geiselmann, and Thomas Beth. Quantum Reed-Solomon Codes. In Marc Fossorier, Hideki Imai, Shu Lin, and Alain Poli, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, pages 231–244, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.
- [70] Lov Kumar Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM Symposium on Theory of Computing*, pages 212–219, 1996.
- [71] Joachim Hagenauer and Peter Adam Hoehner. A Viterbi algorithm with soft-decision outputs and its applications. In *1989 IEEE Global Telecommunications Conference and Exhibition 'Communications Technology for the 1990s and Beyond'*, pages 1680–1686 vol.3, 1989.
- [72] Joachim Hagenauer, Elke Offer, and Lutz Papke. Iterative decoding of binary block and convolutional codes. *IEEE Transactions on Information Theory*, 42(2):429–445, 1996.
- [73] Manabu Hagiwara, Kenta Kasai, Hideki Imai, and Kohichi Sakaniwa. Spatially coupled quasi-cyclic quantum LDPC codes. In *2011 IEEE International Symposium on Information Theory Proceedings*, pages 638–642. IEEE, 2011.

- [74] Richard Wesley Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950.
- [75] Lajos Hanzo, Tong Hooi Liew, and Bee Leong Yeap. *Turbo coding, turbo equalisation and space-time coding*. John Wiley & Sons, 2002.
- [76] Lajos Hanzo, Robert Maunder, Jin Wang, and Lie-Liang Yang. *Near-capacity variable-length coding: regular and EXIT-chart-aided irregular designs*, volume 20. John Wiley & Sons, 2011.
- [77] Lajos Hanzo and Soon Xin et al. Ng. *Turbo coding, turbo equalisation and space-time coding: EXIT-chart-aided near-capacity designs for wireless channels*, volume 22. John Wiley & Sons, 2011.
- [78] Alexis Hocquenghem. Codes correcteurs d’erreurs. chiffres (paris), 2, 147-156. *Mathematical Review*, 22:652, 1959.
- [79] Monireh Houshmand and Mark Wilde. Recursive quantum convolutional encoders are catastrophic: A simple proof. *IEEE Transactions on Information Theory*, 59(10):6724–6731, 2013.
- [80] Min-Hsiu Hsieh, Todd Brun, and Igor Devetak. Entanglement-assisted quantum quasicyclic low-density parity-check codes. *Physical Review A*, 79(3):032340, 2009.
- [81] Sandor Imre and Ferenc Balazs. *Quantum Computing and Communications: an engineering approach*. John Wiley & Sons, 2005.
- [82] Liang Jiang, Jacob Taylor, Anders Sørensen, and Mikhail Lukin. Distributed quantum computation based on small quantum registers. *Physical Review A*, 76(6):062323, 2007.
- [83] Kenta Kasai, Manabu Hagiwara, Hideki Imai, and Kohichi Sakaniwa. Non-binary quasi-cyclic quantum LDPC codes. In *2011 IEEE International Symposium on Information Theory Proceedings*, pages 653–657. IEEE, 2011.
- [84] Kenta Kasai, Manabu Hagiwara, Hideki Imai, and Kohichi Sakaniwa. Quantum error correction beyond the bounded distance decoding limit. *IEEE Transactions on Information Theory*, 58(2):1223–1230, 2011.
- [85] Alexei Yurievich Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191–1249, dec 1997.
- [86] Alexei Yurievich Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, 2003.

- [87] Jörg Kliewer, Soon Xin Ng, and Lajos Hanzo. Efficient computation of exit functions for nonbinary iterative decoding. *IEEE Transactions on Communications*, 54(12):2133–2136, 2006.
- [88] Emanuel Knill and Raymond Laflamme. Theory of quantum error-correcting codes. *Physical Review A*, 55(2):900, 1997.
- [89] Wolfgang Koch and Alfred Baier. Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference (applicable to digital mobile radio receivers). In *[Proceedings] GLOBECOM '90: IEEE Global Telecommunications Conference and Exhibition*, pages 1679–1684 vol.3, 1990.
- [90] Shrinivas Kudekar, Thomas Richardson, and Rüdiger Urbanke. Threshold saturation via spatial coupling: Why convolutional ldpc ensembles perform so well over the bec. *IEEE Transactions on Information Theory*, 57(2):803–834, 2011.
- [91] Hans Kurzweil and Bernd Stellmacher. *The Theory of Finite Groups: An Introduction*, volume 1. Springer, 2004.
- [92] Raymond Laflamme, Cesar Miquel, Juan Pablo Paz, and Wojciech Hubert Zurek. Perfect quantum error correcting code. *Physical Review Letters*, 77:198–201, Jul 1996.
- [93] Ching-Yi Lai, Todd Brun, and Mark Wilde. Dualities and identities for entanglement-assisted quantum codes. *Quantum Information Processing*, 13(4):957–990, 2014.
- [94] Xiaodong Li and James Ritcey. Bit-interleaved coded modulation with iterative decoding. In *1999 IEEE International Conference on Communications*, volume 2, pages 858–863 vol.2, 1999.
- [95] Shu Lin and Daniel Costello. *Error Control Coding*. Pearson, 2004.
- [96] Seth Lloyd. Capacity of the noisy quantum channel. *Physical Review A*, 55:1613–1622, Mar 1997.
- [97] Michael Luby, Michael Mitzenmacher, Amin Shokrollahi, Daniel Spielman, and Volker Stemann. Practical loss-resilient codes. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 150–159, 1997.
- [98] David John MacKay, Graeme Mitchison, and Paul McFadden. Sparse-graph codes for quantum error correction. *IEEE Transactions on Information Theory*, 50(10):2315–2330, 2004.
- [99] David John MacKay and Radford Neal. Good codes based on very sparse matrices. In *IMA International Conference on Cryptography and Coding*, pages 100–111. Springer, 1995.

- [100] David John MacKay and Radford Neal. Near Shannon limit performance of low density parity check codes. *Electronics letters*, 32(18):1645, 1996.
- [101] Robert Malaney. Location-dependent communications using quantum entanglement. *Physical Review A*, 81(4):042319, 2010.
- [102] Robert Malaney. The quantum car. *IEEE Wireless Communications Letters*, 5(6):624–627, 2016.
- [103] Robert Malaney. Quantum geo-encryption. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2016.
- [104] James Massey. Step-by-step decoding of the Bose-Chaudhuri-Hocquenghem codes. *IEEE Transactions on Information Theory*, 11(4):580–585, 1965.
- [105] James Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, 15(1):122–127, 1969.
- [106] Federico Mattei. Conferenza per TedX, 2019. Disponibile su YouTube, ultimo accesso 19 Febbraio 2022.
- [107] Robert Maunder. A Fully-Parallel Turbo Decoding Algorithm. *IEEE Transactions on Communications*, 63(8):2762–2775, 2015.
- [108] Denise Maurice, Jean-Pierre Tillich, and Iryna Andriyanova. A family of quantum codes with performances close to the hashing bound under iterative decoding. In *2013 IEEE International Symposium on Information Theory*, pages 907–911. IEEE, 2013.
- [109] Robert McEliece, Eugene Rodemich, Howard Rumsey, and Lloyd Welch. New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities. *IEEE transactions on Information Theory*, 23(2):157–166, 1977.
- [110] Christopher Monroe, Robert Raussendorf, Alex Ruthven, Kenneth Brown, Peter Maunz, Liwei Duan, and Jungsang Kim. Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects. *Physical Review A*, 89(2):022317, 2014.
- [111] David Eugene Muller. Application of boolean algebra to switching circuit design and to error detection. *Transactions of the IRE Professional Group on Electronic Computers*, (3):6–12, 1954.
- [112] Sreraman Muralidharan, Linshu Li, Jungsang Kim, Norbert Lütkenhaus, Mikhail Lukin, and Liang Jiang. Optimal architectures for long distance quantum communication. *Scientific reports*, 6(1):1–10, 2016.

- [113] Soon Xin Ng, Osamah Rashed Alamri, Yonghui Li, Jorg Kliewer, and Lajos Hanzo. Near-capacity turbo trellis coded modulation design based on exit charts and union bounds. *IEEE Transactions on Communications*, 56(12):2030–2039, 2008.
- [114] Hung Viet Nguyen, Zunaira Babar, Dimitrios Alanis, Panagiotis Botsinis, Daryus Chandra, Soon Xin Ng, and Lajos Hanzo. Exit-chart aided quantum code design improves the normalised throughput of realistic quantum devices. *IEEE Access*, 4:10194–10209, 2016.
- [115] Helmut Nickl, Joachim Hagenauer, and Frank Burkert. Approaching Shannon’s capacity limit by 0.2 dB using simple Hamming codes. *IEEE Communications Letters*, 1(5):130–132, 1997.
- [116] Michael Nielsen and Isaac Chuang. Quantum Computation and Quantum Information (Cambridge University Press, Cambridge, England). 2000.
- [117] Kai Niu, Kai Chen, Jiaru Lin, and Qiutong Zhang. Polar codes: Primary concepts and practical decoding algorithms. *IEEE Communications Magazine*, 52(7):192–203, 2014.
- [118] Ben Noble. *Applied linear algebra*. Prentice-Hall, 1988.
- [119] Harold Ollivier and Jean-Pierre Tillich. Description of a Quantum Convolutional Code. *Physical Review Letters*, 91:177902, Oct 2003.
- [120] Harold Ollivier and Jean-Pierre Tillich. Quantum convolutional codes: fundamentals. *arXiv preprint quant-ph/0401134*, 2004.
- [121] Naoya Onizawa, Takahiro Hanyu, and Vincent Gaudet. Design of High-Throughput Fully Parallel LDPC Decoders Based on Wire Partitioning. *IEEE Trans. Very Large Scale Integr. Syst.*, 18(3):482–489, Mar 2010.
- [122] Emilie Pelchat and David Poulin. Degenerate Viterbi decoding. *IEEE Transactions on Information Theory*, 59(6):3915–3921, 2013.
- [123] Morris Plotkin. Binary codes with specified minimum distance. *IRE Transactions on Information Theory*, 6(4):445–450, 1960.
- [124] Michael Postol. A proposed quantum low density parity check code. *arXiv preprint quant-ph/0108131*, 2001.
- [125] David Poulin and Yeojin Chung. On the iterative decoding of sparse quantum codes. *Quantum Info. Comput.*, 8(10):987–1000, Nov 2008.
- [126] David Poulin, Jean-Pierre Tillich, and Harold Ollivier. Quantum serial turbo-codes. In *2008 IEEE International Symposium on Information Theory*, pages 310–314, 2008.

- [127] David Poulin, Jean-Pierre Tillich, and Harold Ollivier. Quantum serial turbo codes. *IEEE Transactions on Information Theory*, 55(6):2776–2798, 2009.
- [128] John Preskill. Lecture notes for physics 229: Quantum information and computation. *California Institute of Technology*, 16(1):1–8, 1998.
- [129] John Proakis, Masoud Salehi, Ning Zhou, and Xiaofeng Li. *Communication systems engineering*, volume 2. Prentice Hall New Jersey, 1994.
- [130] Irving Stoy Reed. A class of multiple-error-correcting codes and the decoding scheme. *Transactions of the IRE Professional Group on Information Theory*, 4(4):38–49, 1954.
- [131] Irving Stoy Reed and Xuemin Chen. *Linear Cyclic Codes*, pages 139–187. Springer US, Boston, MA, 1999.
- [132] Irving Stoy Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of The Society for Industrial and Applied Mathematics*, 8:300–304, 1960.
- [133] Joseph Renes, Frédéric Dupuis, and Renato Renner. Efficient polar coding of quantum information. *Physical Review Letters*, 109(5):050504, 2012.
- [134] Joseph Renes, David Sutter, Frédéric Dupuis, and Renato Renner. Efficient quantum polar codes requiring no preshared entanglement. *IEEE Transactions on Information Theory*, 61(11):6395–6414, 2015.
- [135] Thomas Richardson, Amin Shokrollahi, and Rüdiger Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory*, 47(2):619–637, 2001.
- [136] Ronald Rivest, Adi Shamir, and Leonard Adleman. *A method for obtaining digital signatures and public key cryptosystems*. Routledge, 2019.
- [137] Patrick Robertson, Emmanuelle Villebrun, and Peter Adam Hoehner. A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain. In *Proceedings IEEE International Conference on Communications ICC '95*, volume 2, pages 1009–1013 vol.2, 1995.
- [138] Patrick Robertson and Thomas Worz. Bandwidth-efficient turbo trellis-coded modulation using punctured component codes. *IEEE Journal on Selected Areas in Communications*, 16(2):206–218, 1998.
- [139] Pradeep Kiran Sarvepalli, Andreas Klappenecker, and Martin Rötteler. Asymmetric quantum codes: constructions, bounds and performance. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2105):1645–1672, 2009.

- [140] Pieter Schalkwijk and Adrianus Vinck. Syndrome Decoding of Convolutional Codes. *IEEE Transactions on Communications*, 23(7):789–792, 1975.
- [141] Pieter Schalkwijk and Adrianus Vinck. Syndrome decoding of binary rate-1/2 convolutional codes. *IEEE Transactions on Communications*, 24(9):977–985, 1976.
- [142] Pieter Schalkwijk, Adrianus Vinck, and Karel Post. Syndrome decoding of binary-rate k/n convolutional codes. *IEEE Transactions on Information Theory*, 24(5):553–562, 1978.
- [143] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.
- [144] Peter Willstone Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.
- [145] Peter Willstone Shor. Scheme for reducing decoherence in quantum computer memory. *Physical Review A*, 52:R2493–R2496, Oct 1995.
- [146] Peter Willstone Shor. The quantum channel capacity and coherent information. In *Lecture Notes, MSRI Workshop on Quantum Computation*, 2002.
- [147] Vladimir Sidorenko and Victor Zyablov. Decoding of convolutional codes using a syndrome trellis. *IEEE Transactions on Information Theory*, 40(5):1663–1666, 1994.
- [148] Richard Silverman and Martin Balser. Coding for constant-data-rate systems-part I. A new error-correcting code. *Proceedings of the IRE*, 42(9):1428–1435, 1954.
- [149] Richard Singleton. Maximum distance q -nary codes. *IEEE Transactions on Information Theory*, 10(2):116–118, 1964.
- [150] Graeme Smith and John Smolin. Degenerate Quantum Codes for Pauli Channels. *Physical Review Letters*, 98:030501, Jan 2007.
- [151] Rachel Smith. BYU Electrical Computer Engineering faculty Ryan Camacho will help develop a new NSF Engineering Research Center for Quantum Networks, 2020. Ultimo accesso 08/03/2022.
- [152] Andrew Martin Steane. Error Correcting Codes in Quantum Theory. *Physical Review Letters*, 77:793–797, Jul 1996.
- [153] Andrew Martin Steane. Multiple-particle interference and quantum error correction. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 452(1954):2551–2577, 1996.

- [154] Andrew Martin Steane. Simple quantum error-correcting codes. *Physical Review A*, 54:4741–4751, Dec 1996.
- [155] Andrew Martin Steane. Enlargement of Calderbank-Shor-Steane quantum codes. *IEEE Transactions on Information Theory*, 45(7):2492–2495, 1999.
- [156] Andrew Martin Steane. Quantum Reed-Muller codes. *IEEE Transactions on Information Theory*, 45(5):1701–1703, 1999.
- [157] Nicholas Szabo and Richard Tanaka. *Residue arithmetic and its applications to computer technology*. New York: McGraw-Hill, 1967.
- [158] Masahiro Takeoka, Saikat Guha, and Mark Wilde. Fundamental rate-loss tradeoff for optical quantum key distribution. *Nature communications*, 5(1):1–7, 2014.
- [159] Ronald Tee, Robert Maunder, and Lajos Hanzo. Exit-chart aided near-capacity irregular bit-interleaved coded modulation design. *IEEE Transactions on Wireless Communications*, 8(1):32–37, 2009.
- [160] Stephan ten Brink. Convergence behavior of iteratively decoded parallel concatenated codes. *IEEE Transactions on Communications*, 49(10):1727–1737, 2001.
- [161] Jeremy Thorpe. Low-density parity-check (LDPC) codes constructed from protographs. *IPN Progress Report*, 42(154):42–154, 2003.
- [162] Michael Tüchler and Joachim Hagenauer. Exit charts of irregular codes. May 2002. Disponibile su CiteSeerX.
- [163] Gottfried Ungerboeck. Channel coding with multilevel/phase signals. *IEEE Transactions on Information Theory*, 28(1):55–67, 1982.
- [164] Gottfried Ungerboeck. Trellis-coded modulation with redundant signal sets Part I: Introduction. *IEEE Communications Magazine*, 25(2):5–11, 1987.
- [165] Gottfried Ungerboeck. Trellis-coded modulation with redundant signal sets Part II: State of the art. *IEEE Communications Magazine*, 25(2):12–21, 1987.
- [166] Lieven Vandersypen, Matthias Steffen, Gregory Breyta, Costantino Yannoni, Mark Sherwood, and Isaac Chuang. Experimental realization of Shor’s quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414(6866):883–887, 2001.
- [167] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

- [168] Chuan Wang, Fu-Guo Deng, Yan-Song Li, Xiao-Shu Liu, and Gui Lu Long. Quantum secure direct communication with high-dimension quantum superdense coding. *Physical Review A*, 71(4):044305, 2005.
- [169] Yun-Jiang Wang, Barry Sanders, Bao-Ming Bai, and Xin-Mei Wang. Enhanced Feedback Iterative Decoding of Sparse Quantum Codes. *IEEE Transactions on Information Theory*, 58(2):1231–1241, 2012.
- [170] Robert Watson and Wendy Hastings. Self-checked computation using residue arithmetic. *Proceedings of the IEEE*, 54(12):1920–1931, 1966.
- [171] Stephen Wiesner. Conjugate Coding. *SIGACT News*, 15(1):78–88, 1983.
- [172] Mark Wilde and Todd Brun. Entanglement-assisted quantum convolutional coding. *Physical Review A*, 81(4):042333, 2010.
- [173] Mark Wilde and Saikat Guha. Polar codes for classical-quantum channels. *IEEE Transactions on Information Theory*, 59(2):1175–1187, 2012.
- [174] Mark Wilde and Saikat Guha. Polar codes for degradable quantum channels. *IEEE Transactions on Information Theory*, 59(7):4718–4729, 2013.
- [175] Mark Wilde and Min-Hsiu Hsieh. Entanglement boosts quantum turbo codes. In *2011 IEEE International Symposium on Information Theory Proceedings*, pages 445–449, 2011.
- [176] Mark Wilde, Min-Hsiu Hsieh, and Zunaira Babar. Entanglement-assisted quantum turbo codes. *IEEE Transactions on Information Theory*, 60(2):1203–1222, 2013.
- [177] Jack Wolf. Efficient maximum likelihood decoding of linear block codes using a trellis. *IEEE Transactions on Information Theory*, 24(1):76–80, 1978.
- [178] William Wootters and Wojciech Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.
- [179] Lin Xiaoyan. Quantum cyclic and constacyclic codes. *IEEE Transactions on Information Theory*, 50(3):547–549, 2004.
- [180] Philip Yam. Il gatto di Schrodinger resuscita. *Le Scienze*, (348):90–97, 1997.
- [181] Epharaim Zehavi. 8-PSK trellis codes on Rayleigh channel. In *IEEE Military Communications Conference, "Bridging the Gap. Interoperability, Survivability, Security"*, pages 536–540 vol.2, 1989.

Elenco delle figure

1.1	Realizzazione di un bit classico e di un qubit usando la rappresentazione mediante lo spin di un elettrone [12].	8
1.2	Confronto tra i due possibili stati di un bit tradizionale e rappresentazione con la sfera di Bloch dell'informazione contenuta in un qubit [106].	9
1.3	Qubit rappresentato da un atomo con due livelli elettronici [116].	10
1.4	Confronto tra computazione classica e quantistica [12].	14
1.5	Rappresentazione della Qinternet e di alcune sue possibili applicazioni [151].	16
1.6	Modelli di canali quantistici [12].	18
1.7	Rappresentazione grafica del vettore $\mathbf{E}_0 \psi\rangle = \alpha 0\rangle + \sqrt{1-\gamma}\beta 1\rangle$	22
1.8	Andamento del tempo di rilassamento del qubit.	24
1.9	Generico stato del qubit, operazione di bit-flip e phase-flip sulla sfera di Bloch. <i>Bit-flip</i> : scambia lo stato $ 0\rangle$ con lo stato $ 1\rangle$; cambia la latitudine del qubit passando dall'emisfero nord a quello sud della sfera. <i>Phase-flip</i> : lo stato del qubit ruota di mezzo giro rispetto alla longitudine della sfera.	27
1.10	Schematico delle 4 porte logiche di Pauli [12].	28
1.11	Bit-flip e phase-flip per lo stato di un generico qubit, rappresentati rispetto allo spin dell'elettrone. <i>La polarizzazione verticale rappresenta lo stato $1\rangle$, mentre la polarizzazione orizzontale rappresenta lo stato $0\rangle$, il verso della freccia invece indica la fase</i> [11].	29
1.12	Interpretazione dei principali modelli dei canali quantistici argomentati [12].	30
1.13	Canale binario simmetrico.	31
2.1	Limite di Shannon per la capacità di un generico canale AWGN [12].	34
2.2	Schema a blocchi di un sistema con codifica a blocco binaria (n, k) [23].	36
2.3	Rate (R) versus distanza minima normalizzata (δ) [12].	37
2.4	Schema a blocchi che rappresenta un generico codice turbo [129].	42
2.5	Esempio di codificatore ricorsivo [1].	43
2.6	Schema a blocchi che realizza la concatenazione seriale.	43

2.7	Schema a blocchi che realizza la concatenazione parallela. . .	44
2.8	Esempio di una matrice \mathbf{H} , con $n = 20$, $j = 3$ e $p = 4$ [61]. . .	45
2.9	Schematico di un possibile sistema di comunicazione quantistico a B qubit che utilizza la codifica EA.	48
2.10	Sono rappresentati gli hashing bound per i codici quantistici non assistiti ($c = 0$) e massimamente entangled ($c = n - k$), espressi dalle eq.(2.17) e (2.19) [12].	49
2.11	Sono rappresentati i limiti asintotici dell'approssimazione in forma chiusa dell'eq.(2.20) del rate $R_Q = k/n$ in funzione della distanza minima normalizzata $\delta = d_{min}/n$ [12].	50
2.12	La crescita della distanza minima raggiungibile con l'aumento della lunghezza della parola di codice, sulla base dell'espressione dell'eq.(2.20) [12].	51
2.13	Prestazioni ottenibili a codeword error rate di 10^{-3} confrontato con l'Hashing bound [12].	57
2.14	Principali risultati raggiunti nella storia della codifica classica e quantistica [12].	59
3.1	Schematico del circuito che rappresenta il funzionamento dell'operatore di copia \mathcal{U}	61
3.2	Circuito codificatore di un codice bit-flip a ripetizione a 3 qubit [12].	62
3.3	Gate CNOT quantistico a due qubit [116].	63
3.4	Circuito di decodifica del codice a ripetizione a 3 qubit di tipo bit-flip [12].	65
3.5	Confronto tra la porta logica "NOT" a singolo bit (sinistra) e gli operatori di Pauli e di Hadamard: \mathbf{X} (bit-flip), \mathbf{Z} (phase-flip) e H [116].	67
3.6	Visualizzazione del gate di Hadamard sulla sfera di Bloch, dato lo stato di ingresso $(0\rangle + 1\rangle)/\sqrt{2}$ [116].	68
3.7	Circuito codificatore di un codice a ripetizione di tipo phase-flip a 3 qubit, dove il qubit d'informazione $ \psi\rangle$ è codificato in $ \bar{\psi}\rangle$ utilizzando due qubit ausiliari [12].	69
3.8	Circuito decodificatore di un codice a ripetizione di tipo phase-flip a 3 qubit [12].	70
3.9	Transizione dal dominio classico a quello quantistico dei codici per la correzione degli errori [11].	73
4.1	Schematico di un sistema di comunicazione quantistico che utilizza un QSC per la correzione degli errori [10].	75
4.2	Circuito quantistico di misurazione dell'operatore \mathbf{Z} per la correzione di errori di tipo bit-flip [116], [12].	77
4.3	Circuito quantistico di misurazione dell'operatore \mathbf{X} per la correzione di errori di tipo phase-flip [116], [12].	78
4.4	Classificazione dei modelli di errore per i codici stabilizzatori [12].	82

5.1	Rappresentazione dell'errore effettivo P , corrispondente all'errore di Pauli \mathcal{P} ad n qubit [12].	89
5.2	Schema a blocchi che rappresenta il processo di elaborazione della sindrome [12].	94
6.1	Tassonomia dei codici stabilizzatori (CSS: Calderbank-Shor-Steane, EA: Entanglement-Assisted) [12].	96
6.2	Decodificatore di sindrome per i codici quantistici di tipo CSS [12].	99
6.3	Schema a blocchi di un sistema di comunicazione quantistica basato su un codice stabilizzatore quantistico entanglement-assisted [12].	102
7.1	Schematico del generico codificatore sistematico BCH (n, k, d_{min}) [12].	107
7.2	Codificatore sistematico BCH (n, k, d_{min}) [12].	108
7.3	Diagramma di transizione di stato per un codice BCH(15, 11, 3) [12].	110
7.4	Guadagno di codifica rispetto al rate di codifica, per varie famiglie di codici BCH, con un BER di 10^{-6} sul canale AWGN [77]. <i>Nota: è stato utilizzato per la decodifica l'algoritmo di Berlekamp-Massey</i> [12].	111
7.5	Codificatore per un codice QBCH[15, 7] [30].	114
7.6	Schematico del codificatore del codice convoluzionale sistematico $(2, 1, 2)$ [12].	116
7.7	Diagramma di transizione di stato del codice convoluzionale sistematico $(2, 1, 2)$ [12].	118
7.8	Rappresentazione della PCM \mathbf{H} classica semi-infinita, avente una struttura a banda di blocchi [12].	121

Elenco delle tabelle

2.1	Limiti del rate-versus-minimum-distance per i codici classici, dove H_2 indica l'entropia binaria [5], [39].	39
2.2	Limiti del rate-versus-minimum-distance per i codici quantistici [39].	50
3.1	Look-Up Table per un codice classico a ripetizione con rate $1/3$	64
4.1	Errori di inversione di bit a singolo qubit con gli autovalori associati per il codice a ripetizione a 3 qubit di tipo bit-flip [12].	77
5.1	Isomorfismo dal dominio quantistico a quello classico [12].	85
5.2	Addizione in $(\mathbb{F}_2)^2$ [12].	86
5.3	Addizione in $\text{GF}(4)$ [12].	90
5.4	Moltiplicazione in $\text{GF}(4)$ [12].	91
5.5	Prodotto scalare Hermitiano in $\text{GF}(4)$ [12].	92
5.6	Traccia del prodotto scalare in $\text{GF}(4)$ [12].	92
6.1	Somma degli elementi di $\mathbb{Z}_4 = \{0, 1, 2, 3\}$ modulo 4.	97
6.2	Somma bit-a-bit modulo 2.	97
6.3	Sottoinsiemi unici di C_1^\perp in C_1 [12].	98
7.1	Processo di codifica di un codice BCH(15, 11, 3) [12].	109
7.2	Stabilizzatori del codice QBCH[15, 7] [12].	115
7.3	Processo di codifica di un codice convoluzionale sistematico con $(2, 1, 2)$ [12].	117
7.4	Linee guida di progettazione per la costruzione di codici quantistici stabilizzatori [12].	126