

UNIVERSITÀ DEGLI STUDI DI MILANO - BICOCCA

FACOLTÀ DI SCIENZE MATEMATICHE FISICHE E NATURALI



Corso di Laurea Triennale in Fisica

Teletrasporto su computer quantistico IBM

Relatori:

Dr. Andrea Giachero

Dr. Leonardo Banchi

Tesi di:

Dario Tona

Mat. 826150

Correlatore:

Dr. Marco Faverzani

Anno Accademico 2019/2020

*The task is not to see what has never been seen before, but to think what has
never been thought before about what you see everyday.*

E. Schrödinger

Sommario

Dall'intersezione della meccanica quantistica con l'informatica e con la teoria dell'informazione classica, sono recentemente nate nuove branche del sapere come l'informazione e la computazione quantistica. L'idea di utilizzare le leggi della meccanica quantistica per processare l'informazione è spesso attribuita a Richard Feynman, che durante una conferenza del 1981 dal titolo "Simulating Physics with Computers" spiegò come sia difficile per una macchina di Turing classica simulare in modo completo un sistema quantistico, e che per farlo sia necessaria una macchina *quantistica*.

Dal discorso di Feynman sono trascorsi 40 anni ed oggi i primi prototipi di *computer quantistici* sono già realtà. Sono macchine ancora caratterizzate da un forte rumore, ma ciò nonostante è già possibile utilizzarle per dei processi di calcolo.

Questi nuovi hardware hanno come unità elementare dell'informazione non i bit, ma i *qubit*, cioè sistemi quantistici a due livelli. Di conseguenza un qubit può trovarsi in una qualsiasi *sovrapposizione di stati* $|\psi\rangle = a|0\rangle + b|1\rangle$. È inoltre possibile creare l'*entanglement* tra due o più qubit ed utilizzarlo come potente strumento per la computazione. Le porte logiche classiche, invece, vengono sostituite con operatori unitari che prendono il nome di *quantum gates*.

Lo scopo della tesi è lo studio e l'implementazione del protocollo del *quantum teleportation* su un computer quantistico dell'IBM. L'azienda, infatti, rende disponibili alcuni dei propri device su un cloud, rendendo accessibile il *quantum computing* a chiunque.

Il quantum teleportation è un protocollo utile per la trasmissione di un generi-

co stato quantistico $|\psi\rangle$ senza il trasporto fisico del portatore dell'informazione. Affinché avvenga la trasmissione, il mittente deve avere un secondo qubit in uno stato massimamente entangled ad un qubit del ricevente. Il protocollo prevede l'applicazione, da parte del mittente, di porte quantistiche sia al proprio qubit entangled sia allo stato $|\psi\rangle$. Successivamente il mittente misura entrambi i propri stati sulla base $\{|0\rangle, |1\rangle\}$ e comunica i risultati al ricevente attraverso un canale di comunicazione classico. Sulla base dei risultati, il ricevente applica delle opportune operazioni al proprio qubit entangled riuscendo a trasformarlo nello stato $|\psi\rangle$. Si tratta quindi di "teletrasporto" di informazione e non di materia. Inoltre, è importante notare che lo stato $|\psi\rangle$ del mittente, a seguito della misura, decade in $|0\rangle$ o $|1\rangle$ e che quindi viene preservato il noto teorema di *no-cloning*.

Per l'implementazione sui computer, ho utilizzato il linguaggio di programmazione Python, unito a Qiskit, un framework open-source sviluppato dall'IBM appositamente per il quantum computing. Il codice è stato prima scritto per un simulatore quantistico chiamato *qasm simulator* e successivamente modificato per i computer quantistici reali soggetti al rumore. La principale modifica del codice si basa sul *deferred measurement principle*, che permette di posticipare le misure del protocollo alla fine del circuito. Con l'hardware attuale, infatti, risulta molto difficile effettuare misure all'interno del circuito.

Per verificare che lo stato $|\psi\rangle$ fosse stato effettivamente teletrasportato da un qubit all'altro del processore, ho applicato le opportune porte quantistiche per ritrasformare $|\psi\rangle$ in $|0\rangle$ e ho successivamente misurato lo stato sulla base $\{|0\rangle, |1\rangle\}$. Come era previsto, nel simulatore si ottiene il valore 0 con probabilità 100% mentre sui computer quantistici reali si ha una piccola probabilità di ottenere 1. In conclusione, lo stato $|\psi\rangle$ è stato teletrasportato con successo anche nel caso di computer quantistici reali con una discrepanza dal valore atteso di 4.6% che risulta però essere compatibile con la percentuale di errore dovuta al rumore introdotto dal dispositivo reale.

La tesi si suddivide in 5 capitoli:

- Introduzione al Quantum Computing e spiegazione teorica degli elementi

utilizzati nel codice.

- Descrizione teorica del protocollo del Quantum Teleportation.
- Implementazione sul simulatore quantistico Qasm.
- Implementazione su computer quantistico reale, risultati sul dispositivo Quito.
- Conclusioni.

Indice

Indice	5
1 Fisica Quantistica e Computazione	7
1.1 I Qubit	8
1.1.1 Misurare un qubit	9
1.1.2 La sfera di Bloch	9
1.1.3 Sistemi di più qubit	10
1.2 Quantum Gates	11
1.2.1 Porte a singolo qubit	12
1.2.2 Porte a più qubit	13
1.3 IBM Quantum Experience	14
2 Il <i>Quantum Teleportation</i>	15
2.1 Il protocollo	16
3 Esperimento sul simulatore Qasm	19
3.1 Funzioni preliminari	19
3.1.1 Creare uno stato casuale	19
3.1.2 Coppia EPR	20
3.2 Implementazione del protocollo	20
3.3 Verifica e risultati	22
4 Esperimento su computer quantistico	24
4.1 Dispositivi reali	24
4.1.1 Cenni sull'Hardware	26

<i>INDICE</i>	6
4.1.2 Dispositivi IBM	27
4.2 Modifiche del codice	28
4.3 Verifica e risultati	30
5 Conclusioni	33
6 Appendice	34
Bibliografia	40

Capitolo 1

Fisica Quantistica e Computazione

Dalla sua formulazione, all'inizio degli anni '20, la meccanica quantistica ha rivoluzionato il nostro modo di vedere il mondo. Dalle sue leggi abbiamo costruito nuove teorie fisiche con cui siamo stati capaci di descrivere in modo estremamente soddisfacente innumerevoli fenomeni naturali: dalla fusione nucleare alla descrizione di sistemi biologici.

Una delle più interessanti novità nella scienza moderna è la computazione quantistica, nata dall'unione della meccanica quantistica con l'informatica.

Tutti i giorni possiamo osservare intorno a noi i risultati della teoria dell'informazione classica, in particolare, i computer che utilizziamo quotidianamente si basano sui principi esposti da Alan Turing e successivamente implementati fisicamente con l'utilizzo delle leggi della meccanica classica.

La computazione quantistica nasce agli inizi degli anni '80 con il contributo di scienziati quali Paul Benioff, Jurij Manin e Richard Feynman. L'idea di base era semplice: riuscire a descrivere la natura con una macchina che utilizzasse il suo linguaggio, ovvero la meccanica quantistica. Se l'idea di base è semplice, non lo è affatto la sua implementazione, oggi giorno i primi *computer quantistici* esistono già, ma siamo ancora lontani dalla realizzazione di una macchina esente da rumore.

Tuttavia è già possibile utilizzare un computer quantistico per la com-

putazione, in particolare, in questa tesi viene implementato il protocollo del *quantum teleportation* prima su un simulatore e successivamente su computer quantistici messi a disposizione dall'IBM. In questo primo capitolo vengono esposti, da un punto di vista matematico, gli elementi principali del quantum computing insieme ai componenti utili all'implementazione del *quantum teleportation*.

1.1 I Qubit

A differenza dei computer classici, i computer quantistici hanno come unità elementare dell'informazione i qubit, cioè sistemi quantistici a due livelli. Esempi potrebbero essere lo spin di un elettrone o la polarizzazione di un singolo fotone.

Matematicamente un qubit è un vettore in uno spazio di Hilbert bidimensionale. Si denominano i due vettori di base con $|0\rangle$ e $|1\rangle$, utilizzando la notazione di Dirac. Lo stato di un qubit è quindi una combinazione lineare del tipo:

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

con a e b numeri complessi tali per cui $|a|^2 + |b|^2 = 1$.

Di conseguenza se gli unici valori che può assumere un bit sono 0 ed 1, un qubit può trovarsi in una qualsiasi *sovrapposizione di stati* della base $\{|0\rangle, |1\rangle\}$ chiamata *base computazionale*. L'informazione dei qubit è contenuta nei numeri complessi a e b , cioè 4 numeri reali di cui però uno è fissato per normalizzazione ed un altro, cioè la fase globale di $|\psi\rangle$, è inosservabile.

Nonostante i qubit abbiano un insieme continuo di possibili valori l'informazione contenuta in essi non è infinita, infatti, l'unico modo che abbiamo per estrarre l'informazione è attraverso la *misura*.

1.1.1 Misurare un qubit

Dalla meccanica quantistica sappiamo che è possibile associare a ciascuna osservabile fisica un operatore hermitiano:

$$O = \sum_i \lambda_i |e_i\rangle \langle e_i|$$

dove gli $|e_i\rangle$ sono gli autovettori ortonormali e gli λ_i i relativi autovalori reali. Il valore medio dell'osservabile è l'elemento di matrice di tale operatore nello stato del sistema:

$$\langle \psi | O | \psi \rangle = \sum_i \lambda_i |\langle e_i | \psi \rangle|^2 = \sum_i \lambda_i P_i$$

Effettuare una misura dell'osservabile O su un sistema quantistico in uno stato $|\psi\rangle$, causa il collasso dello stato del sistema in uno dei possibili autovettori $|e_i\rangle$ dell'operatore. Il risultato della misura è dato dall'autovalore λ_i associato allo stato finale del sistema $|e_i\rangle$. Ad ogni possibile risultato sarà associata una probabilità data dalla formula $P_i = |\langle e_i | \psi \rangle|^2$.

Nel quantum computing le osservabili utili alla computazione sono quelle esprimibili con le matrici di Pauli σ_z e σ_x , la prima ha infatti come autovettori $|0\rangle$ e $|1\rangle$ la seconda ha gli autovettori $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ e $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$ (Sezione 1.2.1).

1.1.2 La sfera di Bloch

È possibile avere una rappresentazione geometrica del qubit con la cosiddetta *sfera di Bloch*. Possiamo infatti visualizzare il qubit come un vettore su una sfera in \mathbb{R}^3 .

Si parametrizzano le ampiezze dello stato con tre numeri reali γ , δ e θ :

$$a = e^{i\gamma} \cos \frac{\theta}{2} \tag{1.1}$$

$$b = e^{i\delta} \sin \frac{\theta}{2} \tag{1.2}$$

Ignorando la fase globale si può scrivere lo stato come:

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \tag{1.3}$$

con $0 < \theta < \pi$ e $0 < \phi < 2\pi$. Utilizzando quindi le coordinate sferiche $(\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$ posso identificare lo stato come un punto su una sfera

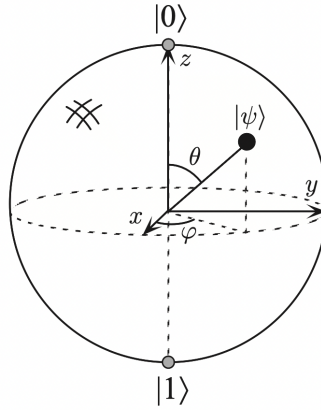


Figura 1.1: Sfera di Bloch [2]

La sfera di Bloch è un utile strumento per la computazione, permette di visualizzare gli effetti che le operazioni applicate in un calcolo hanno su un qubit.

1.1.3 Sistemi di più qubit

In generale un sistema quantistico di n qubit, vive in uno spazio di Hilbert dato dal prodotto tensoriale degli spazi di Hilbert bidimensionali dei singoli qubit $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \dots \otimes \mathcal{H}_n$. I vettori della base dello spazio vengono scritti in un unico ket $|x_1 x_2 \dots x_n\rangle$ con $x \in \{0, 1\}$.

Se consideriamo per semplicità un sistema di 2 qubit, uno in uno stato $|\psi\rangle_1$ e l'altro nello stato $|\phi\rangle_2$ il sistema composto è descritto dallo stato:

$$|\psi\rangle_1 \otimes |\phi\rangle_2 \quad (1.4)$$

In questi casi si dice che lo stato è *separabile*.

Tuttavia, in generale, lo stato del sistema è dato da una combinazione lineare dei vettori della base:

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle \quad (1.5)$$

con la normalizzazione $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$. Di conseguenza non è detto che sia sempre possibile scriverlo nella forma 1.4, in questi casi si parla di stati *entangled*.

Stati di Bell

I più importanti stati entangled a due qubit utilizzati per la computazione, ed in particolare per il quantum teleportation, sono gli *stati di Bell*:

$$\begin{aligned} |\phi_+\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}} \\ |\phi_-\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}} \\ |\psi_+\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}} \\ |\psi_-\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}} \end{aligned}$$

Si nota che, il risultato di una misura di uno dei due qubit in uno stato di Bell è indeterminato. Tuttavia misurando il primo qubit nella base computazionale, il risultato della misura dell'altro è correlato al risultato del primo. In particolare è uguale per gli stati $|\phi_i\rangle$ ed è opposto per gli stati $|\psi_i\rangle$.

Questa *azione a distanza* è stata molto dibattuta dai fisici, famoso è infatti il paradosso EPR da cui si conclude come la fisica quantistica sia incompatibile con il concetto classico di *realismo locale*.

1.2 Quantum Gates

Il modo che abbiamo per fare evolvere uno stato quantistico è attraverso l'applicazione di matrici unitarie al vettore di stato, in modo da preservare la normalizzazione. Queste operazioni vengono implementate con delle *quantum gates*, cioè l'analogo quantistico delle porte logiche dei circuiti classici. Quindi, se nella computazione classica l'unica porta diversa dall'identità applicabile ad un singolo bit, è la porta NOT (che trasforma 0 in 1 e viceversa), nella computazione quantistica ogni matrice unitaria è una potenziale porta quantistica. Inoltre, essendo le matrici unitarie, la operazioni sui qubit sono *sempre*

reversibili. Di seguito vengono espone le più importanti porte quantistiche a 1 e due qubit.

1.2.1 Porte a singolo qubit

Porte X, Y, Z

Tre porte quantistiche utilizzate spesso nelle computazioni sono le porte X, Y e Z che corrispondono rispettivamente alle matrici di Pauli σ_x σ_y σ_z . L'azione sul vettore $|\psi\rangle = a|0\rangle + b|1\rangle$ è quindi:

$$\begin{aligned} X|\psi\rangle &\equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} b \\ a \end{pmatrix} \\ Y|\psi\rangle &\equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = i \begin{pmatrix} -b \\ a \end{pmatrix} \\ Z|\psi\rangle &\equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ -b \end{pmatrix} \end{aligned}$$

Porta H o Hadamard-gate

Un'altra importante porta quantistica è la porta di Hadamard:

$$H|\psi\rangle \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} a+b \\ a-b \end{pmatrix} \equiv a|+\rangle + b|-\rangle$$

Questa porta trasforma la base computazionale negli stati $|+\rangle$ e $|-\rangle$, i quali sono gli autostati della porta X.

Porte Rotazionali

Le *porte rotazionali* permettono di compiere rotazioni intorno agli assi nella sfera di Bloch, corrispondendo quindi alle matrici di rotazione R_x R_y e R_z così definite:

$$R_x(\theta) = e^{-i\frac{\theta}{2}X} = \begin{pmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}$$

$$R_y(\theta) = e^{-i\frac{\theta}{2}Y} = \begin{pmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}$$

$$R_z(\theta) = e^{-i\frac{\theta}{2}Z} = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$$

È possibile dimostrare che per ogni trasformazione unitaria U di singolo qubit è sempre possibile trovare dei numeri reali α β γ e δ tali per cui:

$$U = e^{i\delta} R_z(\alpha) R_y(\beta) R_z(\gamma) \quad (1.6)$$

Rappresentano quindi un modo utile per decomporre un qualsiasi porta quantistica a singolo qubit.

1.2.2 Porte a più qubit

È possibile inoltre implementare porte quantistiche che agiscono su più qubit contemporaneamente. Le matrici a più qubit sono date dal prodotto tensoriale delle matrici a singolo qubit.

Un esempio di porta quantistica a 2 qubit è la CNOT-gate (o CX gate). Questa porta ha un qubit usato come controllo e l'altro come target: se il primo qubit è in $|1\rangle$ applica una NOT-gate al secondo mentre se il primo qubit è in $|0\rangle$ lascia il secondo inalterato. Questa porta è rappresentata dalla matrice:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

nella base $|00\rangle$ $|10\rangle$ $|01\rangle$ $|11\rangle$. Può essere dimostrato che le CNOT gate insieme alle gate di singolo qubit formano un insieme *universale*, il che significa che qualsiasi porta quantistica a n qubit può essere sempre decomposta con un'opportuna combinazione di porte CNOT e porte di singolo qubit.

In modo equivalente alla porta CX, esistono le rispettive porte controllate di Y e Z, cioè le porte quantistiche a due qubit CY e CZ.

1.3 IBM Quantum Experience

L'azienda IBM mette a disposizione alcuni dei propri computer quantistici su un cloud [6], in modo da permettere a tutta la comunità scientifica di realizzare esperimenti su vero "hardware quantistico". Sono disponibili sia *simulatori quantistici* che simulano il comportamento di un computer quantistico sia computer reali.

Il linguaggio di programmazione utilizzato per la computazione è Python, in particolare viene utilizzato il framework open-source Qiskit (sviluppato dall'IBM) con la quale è possibile interagire con i computer quantistici. In qiskit tutti i qubit sono inizializzati in $|0\rangle$, è quindi possibile applicare le varie porte quantistiche per la computazione. A fine circuito inoltre possono essere effettuate le misurazioni, salvando i risultati su bit classici.

Qiskit permette inoltre la visualizzazione grafica dei circuiti (figura 1.2)

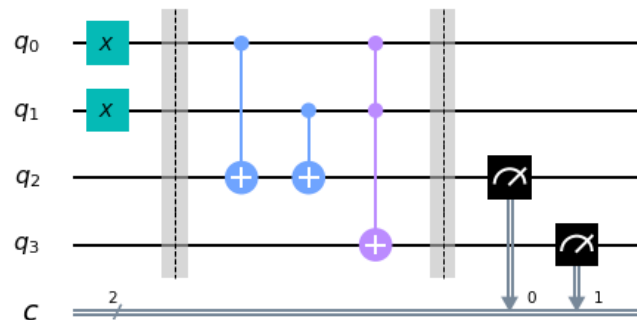


Figura 1.2: Esempio di circuito quantistico [7]

Negli hardware quantistici ogni circuito quantistico viene eseguito migliaia di volte, in modo da calcolare le probabilità associate ad ogni risultato e permetterci di confrontarle con le probabilità attese.

Capitolo 2

Il *Quantum Teleportation*

La parola *teletrasporto* è spesso associata, nel parlare comune, a film fantascientifici, in cui un oggetto o una persona viene fatta scomparire per poi ricomparire istantaneamente da qualche altra parte. Tuttavia, nella realtà, le cose son ben diverse. Nella scienza moderna la parola teletrasporto non sfida nessuna legge della fisica, teletrasporto infatti non è sinonimo di istantaneità. Inoltre non si tratta di teletrasporto di materia ma di *informazione*, in particolare di informazione quantistica.

I primi scienziati a parlare di teletrasporto quantistico sono stati C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters che in un paper del 1993 dal nome "Teleporting an Unknown Quantum State via Dual Classical and Einstein-Podolsky-Rosen Channels" [8] esposero un nuovo modo per trasferire l'informazione utilizzando degli stati di Bell.

Supponiamo di avere un *mittente*, che da qui in avanti chiameremo *Alice*, ed un *ricevente*, chiamato *Bob*, quello che si vuole fare è trasferire l'informazione in un qubit di Alice e farla avere a Bob, senza mandare il qubit stesso a Bob attraverso un canale di comunicazione quantistico. Siamo interessati al caso generale, cioè quando Alice non conosce lo stato $|\psi\rangle$ da trasferire, infatti se per esempio sapesse che $|\psi\rangle = |+\rangle$ potrebbe dire a Bob di applicare un Hadamard-gate ad un suo qubit. Ma, se il qubit è sconosciuto ad Alice, non c'è modo per lei di conoscerne il vettore di stato, come sappiamo l'unico modo per estrarre l'informazione in esso è attraverso una misura, che farebbe collassare la fun-

zione d'onda in un vettore della base. È inoltre impossibile per Alice fare una copia dello stato per il teorema di *no-cloning*.

Teorema di no Cloning

Non esiste un operatore unitario U su $\mathcal{H} \otimes \mathcal{H}$ tale per cui per tutti gli stati normalizzati $|\phi\rangle$ e $|e\rangle$ in \mathcal{H}

$$U(|\phi\rangle |e\rangle) = e^{i\theta} |\phi\rangle |\phi\rangle$$

Dimostrazione

Supponiamo per assurdo l'esistenza di un'operazione unitaria che permetta di copiare un qualsiasi stato e siano $|\psi\rangle$ e $|\phi\rangle$ due stati generici in \mathcal{H} , allora si avrà:

$$\langle\psi|\phi\rangle \langle e|e\rangle = \langle\phi|\langle e| U^\dagger U |\psi\rangle |e\rangle = e^{i(\alpha_1 - \alpha_2)} \langle\phi|\langle\phi|\psi\rangle |\psi\rangle = e^{i(\alpha_1 - \alpha_2)} \langle\phi|\psi\rangle^2$$

dato che $|e\rangle$ è normalizzato otteniamo $|\langle\phi|\psi\rangle| = |\langle\phi|\psi\rangle|^2$. Questa equazione è soddisfatta solo per stati ortonormali ma questo contraddice le ipotesi del teorema che prevedevano degli stati arbitrari.

2.1 Il protocollo

Nonostante le premesse, è comunque possibile per Alice riuscire a trasferire l'informazione del proprio qubit a Bob. La questione si può risolvere se i due condividono due qubit in uno stato di Bell. Questo può essere fatto in anticipo oppure si può fare affidamento su una sorgente che distribuisce i qubit entangled ad entrambi.

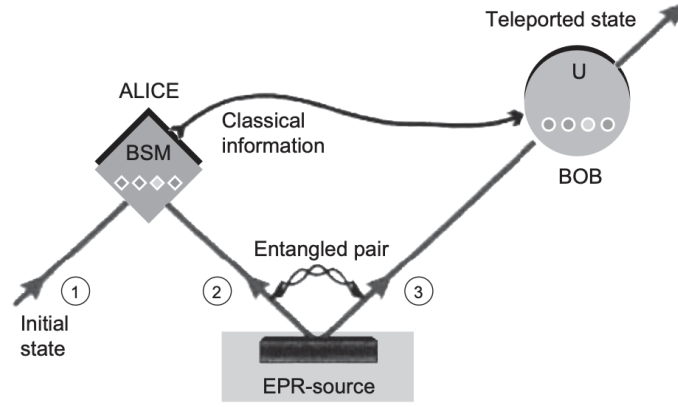


Figura 2.1: Schema teletrasporto quantistico [3]

Supponiamo quindi che Alice e Bob abbiano uno stato massimamente entangled condiviso, del tipo:

$$|\beta_{00}\rangle = \frac{|0_2 0_3\rangle + |1_2 1_3\rangle}{\sqrt{2}} \quad (2.1)$$

con il qubit 2 appartenente ad Alice e il qubit 3 a Bob. Alice vuole mandare a Bob uno stato a lei sconosciuto:

$$|\psi\rangle = \alpha |0_1\rangle + \beta |1_1\rangle$$

con $|\alpha|^2 + |\beta|^2 = 1$. Lo stato completo dei tre qubit del sistema sarà dato dal prodotto tra i due stati:

$$|\phi_a\rangle = |\psi\rangle |\beta_{00}\rangle = \frac{1}{\sqrt{2}} [\alpha |0_1\rangle (|0_2 0_3\rangle + |1_2 1_3\rangle) + \beta |1_1\rangle (|0_2 0_3\rangle + |1_2 1_3\rangle)] \quad (2.2)$$

Alice applica ai propri qubit una porta CNOT utilizzando lo stato $|\psi\rangle$ come controllo ed il proprio qubit entangled come target, ottenendo:

$$|\phi_b\rangle = \frac{1}{\sqrt{2}} [\alpha |0_1\rangle (|0_2 0_3\rangle + |1_2 1_3\rangle) + \beta |1_1\rangle (|1_2 0_3\rangle + |0_2 1_3\rangle)] \quad (2.3)$$

Dopodiché Alice applica una porta Hadamard allo stato $|\psi\rangle$. Si ha dunque:

$$|\phi_c\rangle = \frac{1}{2} [(\alpha(|0_1\rangle + |1_1\rangle)(|0_2 0_3\rangle + |1_2 1_3\rangle) + \beta(|0_1\rangle - |1_1\rangle)(|1_2 0_3\rangle + |0_2 1_3\rangle))] \quad (2.4)$$

Raggruppando i termini, lo stato può essere riscritto nel seguente modo:

$$\begin{aligned} |\phi_c\rangle = \frac{1}{2} [& |0_1 0_2\rangle (\alpha |0_3\rangle + \beta |1_3\rangle) + |0_1 1_2\rangle (\alpha |1_3\rangle + \beta |0_3\rangle) + \\ & + |1_1 0_2\rangle (\alpha |0_3\rangle - \beta |1_3\rangle) + |1_1 1_2\rangle (\alpha |1_3\rangle - \beta |0_3\rangle)] \end{aligned} \quad (2.5)$$

Quest'ultima equazione si divide naturalmente in quattro termini, possiamo notare che se Alice compie una misura dei suoi qubit sulla base computazionale ed ottiene come risultato 00 allora il qubit di Bob sarà nello stato $|\psi\rangle$. Tuttavia questo non avviene per tutti i possibili risultati 00, 01, 10 e 11 di Alice, ma a seconda dei risultati il qubit di Bob si troverà nei seguenti stati:

$$00 \longrightarrow (\alpha |0_3\rangle + \beta |1_3\rangle)$$

$$01 \longrightarrow (\alpha |1_3\rangle + \beta |0_3\rangle)$$

$$10 \longrightarrow (\alpha |0_3\rangle - \beta |1_3\rangle)$$

$$11 \longrightarrow (\alpha |1_3\rangle - \beta |0_3\rangle)$$

Quindi, Alice manda il risultato delle misure a Bob attraverso un canale di computazione classico, in questo modo Bob conosce lo stato del proprio qubit e può applicare le porte quantistiche che gli permettono di ottenere $|\psi\rangle$. Se il risultato è 00, Bob non deve far nulla. Se il risultato è 01 applica una X-gate. Se il risultato è 10, applica una Z-gate, infine se il risultato è 11, Bob trasforma il proprio stato in $|\psi\rangle$ applicando prima una X-gate e successivamente una Z-gate.

Ad un primo sguardo, sembrerebbe che il teletrasporto permetta la trasmissione di qubit in modo istantaneo. Questo ovviamente è falso, infatti affinché Bob ottenga lo stato $|\psi\rangle$ è necessaria una trasmissione di informazione attraverso un canale classico che è limitato dalla velocità della luce. Di conseguenza il teletrasporto non può avvenire a velocità maggiori. Si risolve quindi l'apparente paradosso.

Inoltre, sembrerebbe che il protocollo del teletrasporto quantistico violi il teorema di no-cloning, tuttavia possiamo notare che dopo la misura di Alice lo stato $|\psi\rangle$ decade in in uno stato della base computazionale $|0\rangle$ e $|1\rangle$ e che quindi lo stato $|\psi\rangle$ di Bob, è l'unico rimasto.

Un'altra importante osservazione da fare, è che non possiamo effettuare il teletrasporto di un qubit avendo solo due bit classici a disposizione, per questo protocollo di comunicazione, infatti, è necessario avere una coppia di qubit entangled (ebit).

Capitolo 3

Esperimento sul simulatore

Qasm

In questo capitolo viene descritta l'implementazione del protocollo del teletrasporto quantistico su un simulatore chiamato *Qasm simulator*. Si verifica inoltre il corretto funzionamento del programma per la successiva implementazione su computer quantistici reali. Chiaramente in questo caso il teletrasporto è simulato, non avviene nessuna trasmissione di informazione, ma, semplicemente il simulatore Qasm restituisce i risultati che ci si aspettano da un vero computer quantistico in assenza di rumore.

3.1 Funzioni preliminari

Come prima cosa ho scritto due funzioni preliminari utili all'implementazione del protocollo, la prima crea un generico stato $|\psi\rangle$ sul primo qubit, mentre la seconda crea l'entanglement tra il secondo ed il terzo qubit.

3.1.1 Creare uno stato casuale

In Qiskit tutti i qubit sono inizializzati nello stato $|0\rangle$. Per creare quindi uno stato generico $|\psi\rangle = a|0\rangle + b|1\rangle$ ho sfruttato il teorema (1.6) esposto nel primo capitolo: considerando una generica trasformazione che porta $|0\rangle$ in $|\psi\rangle$, questa può essere sempre scomposta in tre rotazioni Z-Y-Z, di cui però la prima

rotazione intorno a Z dà una fase globale irrilevante. Ho quindi applicato due gate rotazionali R_y e R_z al primo qubit, con angoli generati casualmente con la funzione `random.random()` di Python. Ho successivamente utilizzato qiskit per visualizzare lo stato sulla sfera di Bloch.

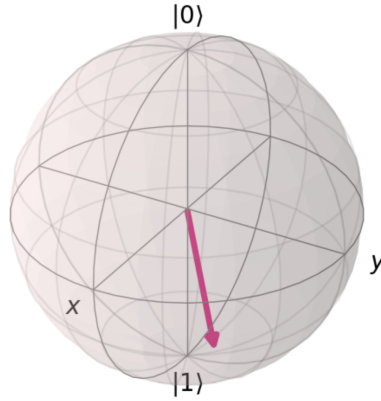


Figura 3.1: Stato casuale

3.1.2 Coppia EPR

Per creare uno stato massimamente entangled tra due qubit è stato necessario applicare una porta Hadamard ad uno dei due, trasformando lo stato $|0\rangle$ in $|+\rangle$, e successivamente applicare una CNOT utilizzando lo stato $|+\rangle$ come controllo e l'altro qubit come target. Ottenendo dunque lo stato di Bell necessario per il protocollo.

$$|\beta_{00}\rangle = \frac{|0_2 0_3\rangle + |1_2 1_3\rangle}{\sqrt{2}} \quad (3.1)$$

3.2 Implementazione del protocollo

Come prima cosa, nel codice è stato creato il circuito quantistico, che consta di un registro quantistico di tre qubit, i primi due assegnati ad Alice ed il terzo a Bob, e di un registro classico di tre bit utili per salvare i risultati delle misure. Sono state quindi richiamate le funzioni preliminari per creare lo stato da teletrasportare sul qubit "0" e creare l'entanglement tra gli altri due.

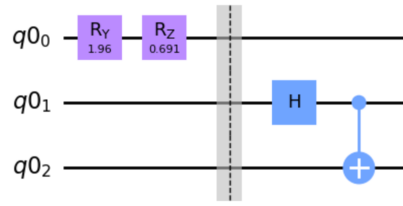


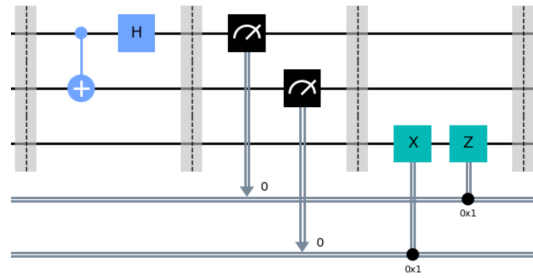
Figura 3.2: Funzioni preliminari

Dopo aver richiamato le funzioni preliminari si passa all'implementazione del protocollo vero e proprio, viene quindi creata una terza funzione in cui si applicano le operazioni di Alice sui propri qubit. Come descritto nella sezione 2.1, viene applicata una porta CNOT, con il primo qubit utilizzato come controllo ed il secondo come target. Successivamente viene applicata una porta Hadamard sullo stato da teletrasportare. Infine si crea una quarta funzione in cui vengono misurati i primi due qubit sulla base computazionale $\{|0\rangle, |1\rangle\}$. Adesso, il protocollo prevederebbe la comunicazione dei risultati a Bob attraverso un canale di comunicazione classico. Dato che in questo caso si sta teletrasportando uno stato da un qubit del processore ad un altro, quello che è stato fatto è simulare la trasmissione di informazione classica con l'utilizzo della funzione *c_if* di Qiskit. Questa funzione permette la lettura dei risultati dei registri classici in cui sono salvati i risultati delle misure, ed è stata utilizzata per aggiungere una condizione all'applicazione delle porte quantistiche di Bob.

```
def Bob_operations(circuit, qr, Alice_cr0, Alice_cr1) :
    circuit.x(qr[2]).c_if(Alice_cr1, 1)
    circuit.z(qr[2]).c_if(Alice_cr0, 1)

    circuit.barrier()
```

In questo modo, sulla base dei risultati, vengono applicate le porte X, Z o entrambe sul terzo qubit come da protocollo. Di seguito il circuito quantistico corrispondente:



Con queste operazioni finali di Bob, lo stato $|\psi\rangle$ dovrebbe adesso trovarsi nel terzo qubit.

3.3 Verifica e risultati

Per verificare che il teletrasporto sia effettivamente avvenuto e che quindi il protocollo sia stato implementato correttamente è stata scritta una funzione di verifica. Avendo salvato, in delle variabili, gli angoli generati casualmente per la creazione dello stato $|\psi\rangle$, è bastato applicare al terzo qubit delle porte rotazionali con angoli esplementari a quelli applicati al primo qubit prima dell'implementazione del protocollo. Ci si aspetta quindi che il terzo si trovi nello stato $|0\rangle$. Di seguito il circuito finale compreso di verifica:

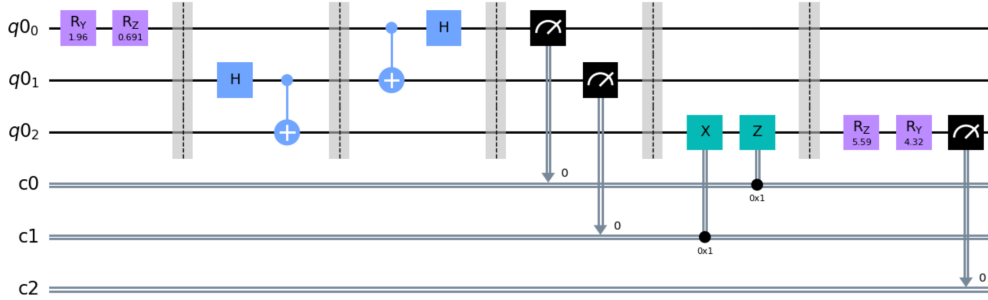


Figura 3.3: Circuito del teletrasporto quantistico con verifica

Il circuito è stato eseguito 1000 volte, nel grafico seguente è possibile osservare le probabilità delle misure effettuate:

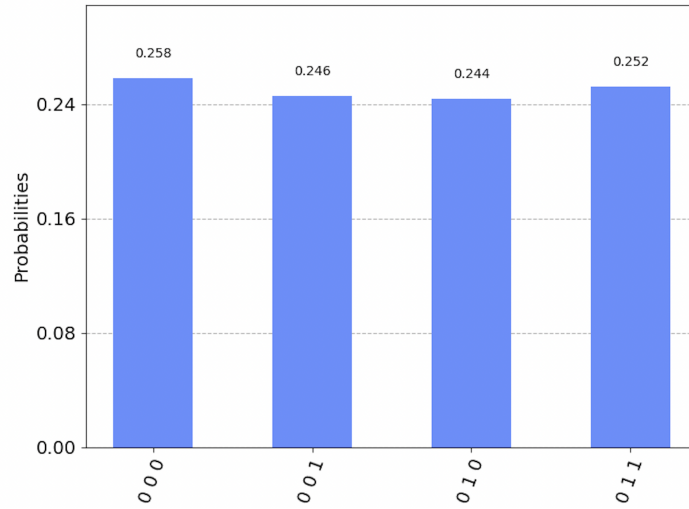


Figura 3.4: Risultati del simulatore Qasm

Come era previsto, la misura del terzo qubit dà 0 nel 100% dei casi, mentre per gli altri due si ha una probabilità di circa il 25% di ottenere 00, 10, 01 e 11. Per ogni bin è possibile associare un'osservabile $P_{xyz} = |xyz\rangle\langle xyz|$, con x , y e z numeri binari. Tale osservabile restituisce "1" se i tre numeri binari sono quelli dell'indice, 0 altrimenti. Le probabilità attese dell'istogramma possono essere scritte come valore di aspettazione sullo stato dell'osservabile P_{xyz} , cioè $Prob(xyz) = \langle P_{xyz} \rangle = 0.25$. Essendo P_{xyz} idempotente, si ha una varianza di tipo Bernoulli $V = Prob(xyz)(1 - Prob(xyz))$. Avendo eseguito il codice 1000 volte, si ha una deviazione standard $\sigma \simeq 0.01$

Capitolo 4

Esperimento su computer quantistico

Dopo l'analisi teorica del protocollo e la successiva implementazione sul simulatore, in questo capitolo viene descritta l'implementazione sui dispositivi reali e le limitazioni che ne derivano. Infatti, allo stato attuale della ricerca, la computazione quantistica non può prescindere dalle caratteristiche dell'hardware di questi nuovi dispositivi.

Attualmente esistono diversi sistemi fisici per la realizzazione di computer quantistici, ogni tecnologia presenta i suoi pro e i suoi contro. Di conseguenza le aziende attive in questo ambito di ricerca stanno procedendo in direzioni diverse, ma tutte hanno lo stesso obiettivo: la realizzazione di una macchina quantistica che rivoluzionerà il modo di processare l'informazione.

4.1 Dispositivi reali

Attualmente la realizzazione di un computer quantistico presenta numerose sfide tecnologiche, uno dei problemi principali è il controllo o la rimozione della *decoerenza* quantistica, cioè la perdita di informazione sullo stato dei qubit. La decoerenza è un processo irreversibile dato che agisce in modo non unitario ed è quindi da evitare. La causa principale della decoerenza è l'interazione del sistema con l'ambiente esterno, per questo motivo i processori

quantistici vengono isolati il più possibile. È possibile distinguere due tipi di decoerenza, il *rilassamento trasverso* cioè la perdita di informazione sulla fase relativa degli stati dei qubit, ed il *rilassamento longitudinale* cioè la tendenza dei qubit di decadere nello stato fondamentale. Ogni dispositivo è quindi caratterizzato da due parametri che indicano il tempo di coerenza dei qubit: T_2 per il rilassamento trasverso e T_1 per il rilassamento longitudinale. È quindi fondamentale che il tempo di applicazione delle porte logiche di un algoritmo non superi il tempo di decoerenza. Ogni porta quantistica ha bisogno infatti di un certo intervallo di tempo per essere implementata fisicamente, per esempio la scala di tempo di una misura è maggiore della scala di tempo delle porte, vedremo che questa sarà una delle limitazioni per l'implementazione del protocollo del teletrasporto quantistico.

Inoltre, ovviamente, è necessario riuscire ad applicare le porte quantistiche nel modo più preciso possibile, se da una parte quindi dobbiamo perturbare il sistema in modo molto preciso dall'altra bisogna evitare la decoerenza isolandolo.

È importante, inoltre, riuscire ad applicare almeno le porte quantistiche che formino un insieme *universale*, come quello descritto nella sezione 1.2.2.

Un altro problema fondamentale per la realizzazione dei processori è la *scalabilità*, cioè riuscire a controllare un numero sempre più alto di qubit senza modifiche consistenti delle tecnologie utilizzate.

L'insieme di tutte queste sfide, insieme a molte altre complicate, causa un rumore di fondo molto difficile da determinare ed eliminare. L'analisi del rumore e la sua rimozione è oggi uno degli ambiti della ricerca sul quantum computing più attivi ed importanti. Negli ultimi anni è stato possibile sviluppare teoricamente algoritmi quantistici *fault tollerant* con l'introduzione di operazioni di correzione, i quali richiedono però 1000-10000 qubit fisici per ogni qubit logico. Siamo perciò ancora lontani ad avere un computer quantistico totalmente esente da rumore.

4.1.1 Cenni sull'Hardware

Per i motivi esposti nel precedente paragrafo non esiste ancora un'unica tecnologia affermata per la realizzazione dei dispositivi. Alcuni esempi di tecnologie utilizzate sono:

- Dispositivi a semiconduttore, utilizzano quantum dots per il confinamento degli elettroni, in questo caso il qubit è dato dallo spin dell'elettrone stesso,
- Fotoni su chip a silicio, in questo caso per realizzare i qubit si può sfruttare la polarizzazione dei fotoni, oppure il *path encoding* o gli stati GVP.
- Ioni intrappolati, in questo caso si utilizzano i livelli energetici atomici di ioni isolati.
- Risonanza magnetica nucleare, con la quale è possibile applicare porte quantistiche ai qubit rappresentati dagli spin di nuclei di molecole immerse in un forte campo magnetico

Nel caso dei dispositivi dell'IBM si fa uso di circuiti elettronici a superconduttore. Nei superconduttori i portatori di carica non sono i singoli elettroni, ma coppie di elettroni, chiamate *coppie di Cooper*. Per realizzare i qubit si utilizza la *giunzione Josephson*, cioè una giunzione formata da due superconduttori separati da un sottile strato di isolante. Per effetto tunnel si crea una corrente che fluisce nella giunzione, tale corrente è utilizzata per creare induttanze non-lineari che permettono la creazione di oscillatori anarmonici, con i quali è possibile isolare solo due autostati dell'hamiltoniana, che saranno gli stati dei qubit. Si utilizzano dei *criostati* per mantenere questi tipi di processori a temperature inferiori ai 100mK, in modo da ridurre la decoerenza e permettere il corretto funzionamento degli elementi circuitali.

4.1.2 Dispositivi IBM

Come già anticipato nel primo capitolo, l'IBM mette a disposizione alcuni dei suoi dispositivi, ognuno dei quali ha caratteristiche tecnologiche diverse. Per rendere più agevole l'uso dei vari *backend*, l'azienda condivide una serie di parametri utili alla computazione. Il primo parametro è ovviamente il numero dei qubit disponibili, a cui si unisce la topologia, cioè il modo in cui i qubit sono distribuiti sul processore. La topologia è particolarmente importante perché ad essa viene associato, attraverso diverse gradazioni di colore, l'errore che si ha nell'interazione tra i diversi qubit. Il parametro sicuramente più importante è però il *quantum volume*, un parametro utile a quantificare l'efficienza generale del dispositivo. La prima definizione del volume quantistico è stata:

$$V_Q = \min[N, d(N)]^2 \quad \text{con} \quad d \simeq \frac{1}{N\epsilon_{eff}}$$

dove N è il numero totale di qubit, d è la profondità del circuito, dato dal numero totale di operazioni del circuito e ϵ_{eff} è l'errore medio di una gate a due qubit. Nel caso un algoritmo utilizzi un numero di qubit $n < N$ allora la formula è data da:

$$V_Q = \max_{n < N} \left\{ \min \left[n, \frac{1}{n\epsilon_{eff}(n)} \right] \right\}^2$$

Successivamente tale formula è stata modificata dai ricercatori dell'IBM:

$$\log_2 V_Q = \arg \max_{n < N} \{ \min[n, d(n)] \}$$

Dispositivi utilizzati

Per l'implementazione del protocollo sono stati utilizzati due backend a 5 qubit, caratterizzati da diversi errori e volumi quantistici. Il primo è chiamato *ibmqquito* ed il secondo *ibmq5-yorktown*. Di seguito una tabella con i parametri dei due dispositivi:

	<i>ibmq-quito</i>	<i>ibmq-5_yorktown</i>
Quantum Volume	16	8
Avg. CNOT Error:	1.122e-2	2.023e-2
Avg. Readout Error:	3.040e-2	5.822e-2
Avg. T1:	88.76 μs	50.95 μs
Avg. T2:	69.96 μs	40.75 μs

Come possiamo notare il dispositivo quito ha parametri migliori di yorktown, ha tempi di coerenza maggiori, errori medi minori, e quindi volume quantistico maggiore. Per l'implementazione del protocollo ci aspettiamo quindi risultati migliori su quito.

4.2 Modifiche del codice

Per l'implementazione del protocollo del teletrasporto quantistico su un dispositivo reale è stato necessario tenere in considerazione i limiti tecnologici dei computer quantistici.

Affinché avvenga il teletrasporto è necessario che Alice misuri i propri qubit in modo da poter comunicare i risultati a Bob. Questo è stato implementato senza particolari problemi sul simulatore Qasm, è bastato infatti misurare i due qubit e successivamente, sulla base dei risultati, applicare le corrette porte quantistiche utilizzando la funzione *c_if*. Tuttavia, non è possibile implementare questo processo su un dispositivo reale. Con l'hardware attuale risulta molto difficile dare delle istruzioni dopo aver misurato dei qubit all'interno del circuito quantistico, questo è dovuto al fatto che le scale di tempo per le misure e le porte quantistiche sono diverse. Di conseguenza non è possibile applicare un " *if quantistico* ". Ovviamente ci sono altri svantaggi che derivano dal fatto di non poter compiere misurazioni all'interno del circuito, in particolare, non è possibile il *riutilizzo* dei qubit dopo una misura.

Fortunatamente, però, nel nostro caso è possibile comunque riuscire ad implementare il protocollo, seppur con qualche compromesso. Il problema può essere risolto con delle misure *ritardate*.

Principle of deferred measurement

Le misurazioni possono sempre essere spostate da uno stadio intermedio di un circuito quantistico alla fine del circuito; se i risultati della misurazione vengono utilizzati in qualsiasi fase del circuito, le operazioni controllate classicamente possono essere sostituite da operazioni quantistiche condizionali [2].

Questo principio ci spiega che posporre le misure alla fine di un circuito quantistico non influenza la distribuzione di probabilità dei risultati. Il principio implica che la scelta di misurare un qubit durante o dopo un'operazione condizionata da quel qubit non ha effetti osservabili sui risultati previsti dalla computazione. Le *deferred measurements* ci permettono quindi di sostituire le misure interne al circuito e le gate successive con una gate controllata (Figura 4.1).

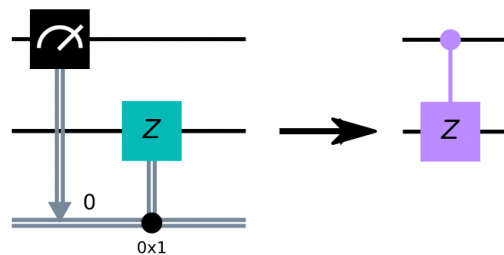


Figura 4.1: Applicazione del principio sulle misure posticipate [7]

Nel caso del teletrasporto quantistico però, è importante notare che stiamo facendo un compromesso: non stiamo più trasmettendo un qubit tra Alice e Bob utilizzando un canale di comunicazione classico. Perdiamo quindi il sostanziale vantaggio del protocollo. Tuttavia l'azione generale del nuovo codice non cambia, il qubit viene teletrasportato anche con questa modifica.

La funzione di Bob del codice descritta nel terzo capitolo viene quindi modificata in questo modo:

```
def Bob_operations_real_device(circuit, qr) :
    circuit.cx(qr[1], qr[2])
    circuit.cz(qr[0], qr[2])
```

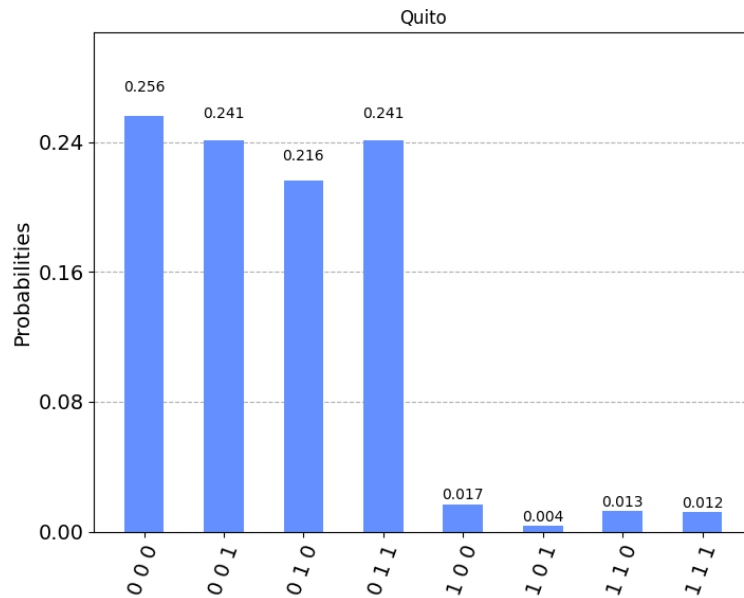



Figura 4.3: Risultati sul dispositivo Quito

Notiamo immediatamente che, a differenza del simulatore, si ha un piccola probabilità di ottenere 1 dalla misura del terzo qubit, non otteniamo una probabilità del 100% di avere 0 ma abbiamo una discrepanza del 4,6% dal valore atteso, tale discrepanza è infatti data dalla somma di tutte le piccole probabilità di avere 001 101 011 e 111. Questa probabilità è la diretta conseguenza del rumore di fondo causato dalle difficoltà tecnologiche descritte in precedenza.

Infine per avere un confronto qualitativo sulle differenti caratteristiche dell'hardware dei computer quantistici reali, è stato rilanciato l'esperimento anche sul backend Yorktown, Come visto in precedenza Yorktown ha un volume quantistico minore di Quito, ed in generale, tra tutti i dispositivi dell'IBM, è tra quelli aventi i parametri peggiori.

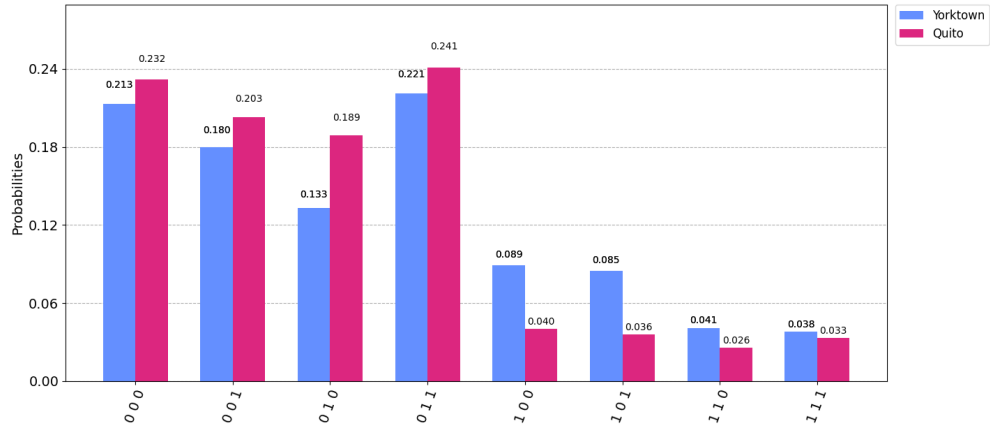


Figura 4.4: Confronto dei risultati di Yorktown e Quito

Dal grafico possiamo apprezzare la differenza di efficienza dei due dispositivi. Nel caso di Yorktown abbiamo una discrepanza dal valore atteso del 25%, compatibile però con il rumore maggiore a cui è soggetto il dispositivo.

Capitolo 5

Conclusioni

In questa tesi si sono introdotti gli elementi principali del *quantum computing* descrivendo i principi della meccanica quantistica che ne sono alla base e le potenzialità in termini computazionali. Sono stati inoltre descritti i limiti tecnologici a cui questo tipo di computazione deve sottostare a causa delle tecnologie attuali.

Il lavoro della tesi è stato lo studio e l'implementazione del protocollo del teletrasporto quantistico prima su un simulatore quantistico e dopo su dispositivi reali. Il protocollo spiega come sia possibile teletrasportare uno stato quantistico ignoto con l'utilizzo di un canale di comunicazione classico, riuscendo ad inviare l'informazione quantistica a distanze arbitrarie sfruttando la non-località dell'entanglement. È stato quindi implementato il protocollo con l'utilizzo del framework open-source Qiskit. Si è descritto nel dettaglio il codice scritto per il simulatore e le modifiche necessarie per l'implementazione sui dispositivi reali.

Il protocollo ha diverse applicazioni nel quantum computing, da esso derivano algoritmi come il *quantum gate teleportation* e il *superdense coding*. Inoltre, ad oggi, il protocollo apre la strada a future applicazioni, in particolare può rivelarsi molto utile per la creazione di reti di comunicazione quantistiche.

Capitolo 6

Appendice

Funzioni

```
from qiskit import *
from qiskit.extensions import Initialize
from qiskit.visualization import *
from qiskit.quantum_info import *
import numpy as np
import random
import math

#angoli casuali per creare psi
a = random.random()
b = random.random()

theta = a*2*np.pi
phi = b*np.pi

#creazione entangled
def Bell_pair(circuit, qr) :
    circuit.h(qr[1])
    circuit.cx(qr[1],qr[2])

    circuit.barrier()

#creazione stato casuale
```

```

def create_a_random_state1(circuit, qr) :
    circuit.ry(phi, qr[0])
    circuit.rz(theta, qr[0])

    plot_bloch_vector([1,phi,theta], title='psi',
        coord_type='spherical')

    circuit.barrier()

def Alice_operations(circuit, qr) :
    circuit.cx(qr[0],qr[1])
    circuit.h(qr[0])

    circuit.barrier()

def Alice_measurements(circuit, qr, Alice_cr0, Alice_cr1)
:
    circuit.measure(qr[0], Alice_cr0)
    circuit.measure(qr[1], Alice_cr1)

def Bob_operations(circuit, qr, Alice_cr0, Alice_cr1) :
    circuit.x(qr[2]).c_if(Alice_cr1, 1)
    circuit.z(qr[2]).c_if(Alice_cr0, 1)

    circuit.barrier()

def Bob_operations_real_device(circuit, qr) :
    circuit.cx(qr[1], qr[2])
    circuit.cz(qr[0], qr[2])

    circuit.barrier()

def checking(circuit, qr, Bob_cr) :
    #riporto lo stato a |0>

```

```

circuit.rz(2*np.pi-theta, qr[2])
circuit.ry(2*np.pi-phi, qr[2])

circuit.barrier()

```

Codice Simulatore

```

from qiskit import *
from funzioni import *
import matplotlib
import matplotlib.pyplot as plt
from qiskit.visualization import plot_histogram
from qiskit.quantum_info import *
from qiskit.extensions import Initialize

#loading my IBMQ account
provider = IBMQ.load_account()

#configurazione
Alice_cr0 = ClassicalRegister(1)
Alice_cr1 = ClassicalRegister(1)
Bob_cr = ClassicalRegister(1)
qr = QuantumRegister(3)
circuit = QuantumCircuit(qr, Alice_cr0, Alice_cr1, Bob_cr)

#trasformo il primo qubit in un generico stato |psi> che
    voglio teletrasportare
create_a_random_state1(circuit, qr)

#creo l'entanglement tra il secondo e terzo qubit
Bell_pair(circuit, qr)

#alice applica le sue porte e misura i suoi qubits
Alice_operations(circuit, qr)

Alice_measurements(circuit, qr, Alice_cr0, Alice_cr1)

```

```

circuit.barrier()

#bob applica l'operazione corretta
Bob_operations(circuit, qr, Alice_cr0, Alice_cr1)

#verifica
checking(circuit, qr, Bob_cr)
circuit.measure(qr[2], Bob_cr)

simulator = Aer.get_backend('qasm_simulator')
result = execute(circuit, backend = simulator, shots =
    1000).result()

counts = result.get_counts(circuit)

plot_histogram(counts)
print(counts)

circuit.draw(output = 'mpl')
plt.show()

```

Codice dispositivi reali

```

from qiskit import *
from funzioni import *
import matplotlib
import matplotlib.pyplot as plt
from qiskit.visualization import plot_histogram
from qiskit.quantum_info import *
from qiskit.extensions import Initialize
from qiskit.tools.monitor import job_monitor

#loading my IBMQ account
provider = IBMQ.load_account()

#configurazione
Alice_cr0 = ClassicalRegister(1)
Alice_cr1 = ClassicalRegister(1)

```

```

Bob_cr = ClassicalRegister(1)
qr = QuantumRegister(3)
circuit = QuantumCircuit(qr, Alice_cr0, Alice_cr1, Bob_cr)

create_a_random_state1(circuit, qr)

#creo l'entanglement tra il secondo e terzo qubit
Bell_pair(circuit, qr)

#alice applica le sue porte
Alice_operations(circuit, qr)

#bob applica l'operazione corretta
Bob_operations_real_device(circuit, qr)

#verifica
checking(circuit, qr, Bob_cr)

#misure finali
Alice_measurements(circuit, qr, Alice_cr0, Alice_cr1)
circuit.measure(qr[2], Bob_cr)

#risultati quito
realQc3 = provider.get_backend('ibmq_quito')
job3 = execute(circuit, backend = realQc3, shots = 1000)
job_monitor(job3)
reaults3 = job3.result()
counts3 = reaults3.get_counts(circuit)
plot_histogram(counts3, title = 'quito')

#risultati yorktown
realQc1 = provider.get_backend('ibmq_5_yorktown')
job1 = execute(circuit, backend = realQc1, shots = 1000)
job_monitor(job1)
reaults1 = job1.result()

```

```
counts1 = reasults1.get_counts(circuit)
plot_histogram(counts1, title = 'Yorktown')

legend = ['Yorktown', 'Quito']
plot_histogram([counts1, counts3], legend=legend)

circuit.draw(output = 'mpl')

plt.show()
```


Bibliografia

- [1] Forte Stefano, and Luca Rottoli. Fisica quantistica. Zanichelli, 2018.
- [2] Nielsen, Michael A., and Isaac L. Chuang. Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press, 2010.
- [3] Furusawa, Akira Loock, Peter. (2011). Quantum Teleportation and Entanglement: A Hybrid Approach to Optical Quantum Information Processing.
- [4] S.Olivares. Lecture Notes on Quantum Computing: https://sites.unimi.it/olivares/wp-content/uploads/2020/04/lectures_qc_Olivares_v4.0.pdf
- [5] Elias Fernandez-Combarro Alvarez. A practical introduction to quantum computing: from qubits to quantum machine learning and beyond: <https://indico.cern.ch/event/970903/>
- [6] IBM Quantum experience: <https://quantum-computing.ibm.com/>
- [7] Learn Quantum Computation using Qiskit: <https://qiskit.org/textbook>
- [8] Bennett, Charles H.; Brassard, Gilles; Crépeau, Claude; Jozsa, Richard; Peres, Asher; Wootters, William K. (29 March 1993). "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels". Physical Review Letters. 70 (13): 1895–1899.

- [9] Heim, B., Soeken, M., Marshall, S., Granade, C., Roetteler, M., Geller, A., ... Svore, K. (2020). Quantum programming languages. *Nature Reviews Physics*, 1-14.
- [10] Feynman, R. P. (2018). Simulating physics with computers. In *Feynman and computation* (pp. 133-153). CRC Press.
- [11] Gottesman, D., Chuang, I. L. (1999). Quantum teleportation is a universal computational primitive. arXiv preprint quant-ph/9908010.
- [12] Moll, Nikolaj; Barkoutsos, Panagiotis; Bishop, Lev S; Chow, Jerry M; Cross, Andrew; Egger, Daniel J; Filipp, Stefan; Fuhrer, Andreas; Gambetta, Jay M; Ganzhorn, Marc; Kandala, Abhinav; Mezzacapo, Antonio; Müller, Peter; Riesswe introd, Walter; Salis, Gian; Smolin, John; Tavernelli, Ivano; Temme, Kristan (2018). "Quantum optimization using variational algorithms on near-term quantum devices". *Quantum Sci. Technol.* 3 (3): 030503. arXiv:1710.01022.
- [13] Cross, Andrew W.; Bishop, Lev S.; Sheldon, Sarah; Nation, Paul D.; Gambetta, Jay M. (2019). "Validating quantum computers using randomized model circuits". *Phys. Rev. A.* 100 (3): 032328. arXiv:1811.12926.

Ringraziamenti

Desidero ringraziare i miei relatori per l'aiuto e la disponibilità: il Dott. Andrea Giachero dell'Università degli Studi di Milano-Bicocca, ed il Dott. Leonardo Banchi, dell'università degli Studi di Firenze. Ringrazio inoltre il mio correlatore, il Dott. Marco Faverzani.

Un grazie speciale ai miei colleghi, con cui ho trascorso dei bellissimi anni a Milano.

Ringrazio gli amici di sempre, per i momenti trascorsi insieme.

Grazie ad Emilia, Emanuele e Gianni per il sostegno e per le domeniche trascorse insieme.

Dedico questa tesi alla mia famiglia, senza la quale non avrei potuto raggiungere questo risultato.