Relazione sull'attività di laboratorio svolta presso il gruppo di ricerca di Informazione e Computazione Quantistica

Rodolfo Rocco

9 settembre 2015

Indice

Acronimi

ECT Extended Church-Turing Thesis

HOM Hong-Ou-Mandel

SPDC Spontaneous parametric down-conversion

QFT Quantum Fourier Trasform

PBS Beam-splitter polarizzatore

"[. . .] nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy." - Richard Feynman, 1982.

Sommario

In questo lavoro è descritta l'attività di laboratorio svolta presso il gruppo di ricerca di Informazione e Computazione Quantistica. Tale attività è consistita nella validazione di un chip integrato quantistico tramite verifica dell'effetto HOM. Nostro compito è stato scrivere il programma che ricostruisse la matrice unitaria caratterizzante il chip.

Nel Capitolo 1 introduciamo i fondamenti teorici alla base dell'esperimento. In particolare viene introdotto il modello computazionale che il chip implementa, che prende il nome di **boson-sampling**; è spiegato il protocollo scelto per certificare il *boson-sampler*, ovvero una legge di soppressione degli stati di output che è una generalizzazione dell'effetto HOM; infine è illustrata la tecnica di fabbricazione del chip integrato, sottolineano i vantaggi rispetto a soluzioni alternative.

Nel Capitolo 2 descriviamo l'esperienza di laboratorio, ovvero la fase di montaggio del chip e il processo di acquisizione dati.

Infine nel Capitolo 3 è discusso il programma che ricostruisce la matrice unitaria caratterizzante il chip in base all'algoritmo delineato in [1]. Inoltre è presentato un secondo programma che genera una matrice unitaria random. Dal confronto tra questa matrice e quella che il primo programma ricostruisce è possibile concludere la correttezza della nostra implementazione.

Capitolo 1

Fondamenti teorici e metodi sperimentali

1.1 La teoria della complessità computazionale

In questa sezione illustreremo brevemente i concetti chiave della teoria della complessità computazionale al fine di comprendere le limitazioni degli attuali computer classici.[1]

1.1.1 Introduzione

La teoria della complessità computazionale è interessata a classificare problemi diversi in base alla quantità di risorse, quali ad esempio tempo di esecuzione e memoria utilizzata, che un computer deve impiegare per risolverli.

Ciascuno di questi problemi è caratterizzato da un algoritmo. Solitamente una misura della complessità del problema è data dal tempo impiegato dal computer a eseguire l'algoritmo. Il tempo dipende ovviamente tanto dalla implementazione dell'algoritmo quanto dal computer sul quale questo gira; cionondimeno la teoria della complessità computazionale mira a fornire una misura della complessità del problema che sia indipendente da questi due fattori.

In aggiunta a ciò bisogna considerare due ulteriori variabili che possono influenzare il tempo d'esecuzione dell'algoritmo, ovvero la dimensione del problema (espressione ambigua sulla quale torneremo più avanti) e i dati specifici di input. Difatti è lecito attendersi che per input diversi il tempo di esecuzione varierà. L'approccio sarà quello di considerare il peggior scenario possibile al variare di tutti gli input x di una dimensione fissata n (ad esempio x potrebbe essere una matrice quadrata di cui n è l'ordine). In formula:

$$T(n) = \max_{|x|=n} t(x) \tag{1.1}$$

[1] Questa sessione è in grande parte desunta da [2].

Dove $T(n)$ è il tempo di esecuzione dell'algoritmo per input di dimensione n (ovvero la sua **complessità temporale**) mentre $t(x)$ è il tempo per lo specifico dato di input x.

La seconda delle variabili di cui tener conto è la dimensione del problema. Per tornare all'esempio della matrice di ordine n, se il problema consiste nel moltiplicare due matrici di ordine n non è difficile immaginare come l'algoritmo associato risulti indipendente dal valore di n. Apparentemente dunque la libertà di scelta di n rappresenta un'ambiguità: per stabilire la complessità del problema quale ordine n dovremo considerare?

Per rispondere alla domanda anzitutto introduciamo la seguente notazione, ispirata alla usuale notazione di Landau.

- $T(n)$ è dell'ordine di $g(n)$ e si scrive $T(n) = \Theta(g(n))$ se esistono c_1, $c_2 > 0$ e n_0 tali che $c_1 g(n) \leq T(n) \leq c_2 g(n)$ per tutti gli $n \geq n_0$.

L'ambiguità scompare se tutti i problemi che possono essere risolti da **algoritmi polinomiali**, ovvero algoritmi con complessità temporale $\Theta(n^k)$ per un qualche k[2], vengono raggruppati in una classe. Questi problemi sonno detti **trattabili**, mentre quelli la cui complessità temporale è ad esempio $\Theta(2^n)$, $\Theta(n!)$ e così via sono detti non trattabili.

1.1.2 Classi di problemi

Cominciamo con una più precisa formulazione delle conclusioni alle quali siamo giunti del paragrafo precedente.

Definizione. *Un problema decisionale*[3] *P è un elemento della casse* P *se e solo se può essere risolto in tempo polinomiale $T(n) = \mathcal{O}(n)$.*

Tutti i problemi della classe P sono trattabili. Passiamo ora alla definizione di una nuova classe. A tal fine dovremo considerare gli algoritmi cosiddetti **non deterministici**. Questi algoritmi contemplano istruzioni del tipo

goto both label 1, label 2

Ovverosia l'esecuzione del programma si divide in due rami: in un ramo essa riprende dal punto contrassegnato dalla etichetta 1, nell'altro da quello contrassegnato dalla etichetta 2. Ovviamente l'algoritmo si può ulteriormente ramificare ogni qual volta si presenti un'istruzione analoga. In virtù della crescita esponenziale dell'albero che descrive i processi che avvengono in parallelo, un algoritmo non deterministico è in grado di svolgere in tempi polinomiali un numero di calcoli che cresce esponenzialmente con il tempo.

Definizione. *Un problema decisionale P è un elemento della casse* NP[4] *se e solo se può essere risolto in tempo polinomiale da un algoritmo non deterministico.*

[2] Valori consueti per k sono $k = 1, 2, 3$. I problemi per cui $k > 10$ sono rari.
[3] Un problema decisionale è un problema al quale si può adeguatamente rispondere *sì/no.*
[4] $\mathcal{NP}$ sta per *nondeterministic polynomial.*

Una soluzione per un problema decisionale è raggiunta da un algoritmo non deterministico soltanto quando ciascuno dei rami è giunto a una soluzione *sì o no*. La soluzione complessiva è *sì* se anche un solo ramo ha soluzione *sì*. Va da sé che la soluzione è *no* solo quando la soluzione di tutti i rami è *no*.

Si dice che la soluzione del problema è raggiunta da un algoritmo non deterministico in tempo polinomiale se per il primo dei rami a riportare una soluzione *sì* vale $T(n) = \mathcal{O}(n)$, dove n è la dimensione del problema.

In considerazione del fatto che un algoritmo convenzionale non è altro che un algoritmo non deterministico nel quale non appaiono le summenzionate istruzioni **goto both**, è evidente che varrà $\mathsf{P} \subset \mathsf{NP}$. In generale i problemi appartenenti a NP non possono essere risolti in tempo polinomiale da una macchina deterministica, sarebbe a dire da una **macchina di Turing classica**, e per questo sono detti non trattabili. Tale difficoltà emerge dal tempo esponenziale necessario a esplorare lo spazio delle soluzioni. Tuttavia per alcuni dei problemi in NP un'intuizione matematica potrebbe suggerire una restrizione dello spazio delle soluzioni che permetta a una macchina di Turing di risolvere il problema in tempo polinomiale. In tal caso il problema è promosso alla classe P.

E' naturale domandarsi se per ogni problema in NP esista una "scorciatoia matematica" che lo renda trattabile, sarebbe a dire se $\mathsf{NP} = \mathsf{P}$. Questa domanda ad oggi non ha ancora una risposta.

E' interessante considerare una seconda definizione di problemi appartenenti alla classe NP. Anzitutto introduciamo il concetto di certificato. Il certificato è una stringa che permette di verificare tramite sostituzione la risposta a un problema. Un **certificato** è detto **succinto** se la sua dimensione è un $\mathcal{O}(n^k)$ dove n^k è la dimensione del problema.

Definizione. *Un problema decisionale P è un elemento della casse NP se e solo se per ogni ramo la cui risposta è sì esiste un certificato succinto che può essere provato in tempo polinomiale.*

A titolo d'esempio consideriamo il problema della fattorizzazione di un numero intero.

Problema. *Dato un numero intero $N > 0$ esistono due numeri interi p, $q > 1$ tali che $N = pq$?*

Questo problema appartiene alla classe NP. Difatti il numero di bit che compongono p e q è minore o uguale al numero di bit che compongono N e pertanto una qualunque soluzione può essere provata tramite moltiplicazione in tempo quadratico.

Diamo ora la definizione di **problema riducibile**, che tornerà utile poco più avanti nella discussione di una nuova classe di problemi.

Definizione. *Un problema P_1 è detto polinomialmente riducibile a un problema P_2 e si scrive $P_1 \leq P_2$ se, supposto che esista un algoritmo polinomiale per P_2, allora anche per P_1 esiste un algoritmo polinomiale.*

Come anticipato introduciamo una nuova classe di problemi: la classe dei problemi $\mathcal{NP}$-completi.

Definizione. *Un problema P è detto* NP*-completo se $P \in$ NP e $Q \leq P$ per tutti i problemi $Q \in$ NP.*

Si è soliti affermare che la classe dei problemi NP-completi raccolga i problemi più difficili di NP. Difatti se uno qualunque dei problemi NP-completi può essere risolto in tempo ottimale allora anche ogni problema in NP potrebbe esserlo; in altre parole NP = P. Questa congettura tuttavia sembra improbabile e ad oggi, come accennavamo, non è ancora stato trovato un algoritmo per risolvere in tempo polinomiale un problema NP-completo.

A questo punto è interessante studiare le relazioni tra le classi di problemi che abbiamo discusso. Abbiamo già visto che vale la relazione di inclusione P $\subset$ NP. Sappiamo dire qualcosa di più?

Il problema della fattorizzazione esaminato in precedenza appartiene a NP, in quanto non si è a conoscenza di un algoritmo polinomiale in grado di risolverlo. Allo stesso tempo non si conosce una riduzione che ci permetta di affermare che esso appartenga alla classe dei problemi NP-completi. L'esistenza del seguente teorema

Teorema. *Se* P $\neq$ NP *allora ci sono problemi appartenenti a* NP *che non sono* NP*-completi e che non appartengono a* P*.*

ci permette di delineare le due possibili mappe riportate in Figura 1.1

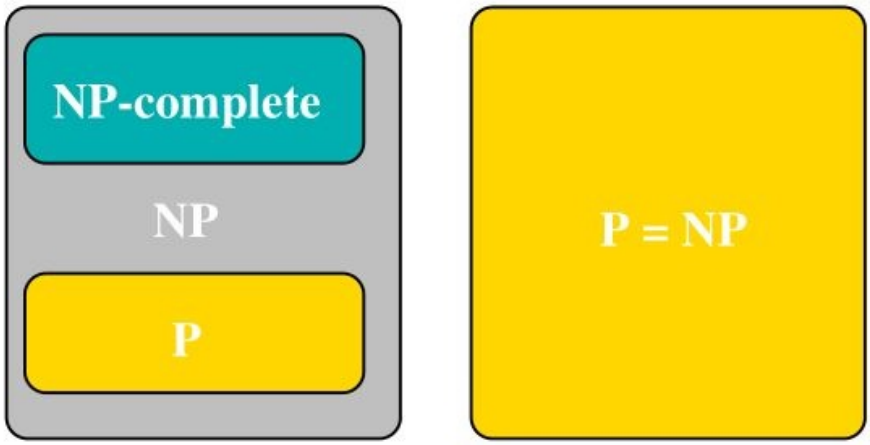

Figura 1.1: Le due possibili mappe per la classe NP, a seconda che valga l'ipotesi NP = P (mappa a destra) o meno. Immagine proveniente da [2].

1.2 Il computer quantistico e l'algoritmo di Shor

Nel 1982 Richard Feynman, nel discutere i limiti e le possibilità delle simulazioni al calcolatore di sistemi fisici, sottolineò quanto sarebbe stato difficile simulare

un sistema quantistico costituito da R particelle [3]. Una crescita lineare nel numero di particelle avrebbe infatti comportato una crescita esponenziale nel numero di elementi costituenti il calcolatore. L'argomentazione è molto generale ed è per questo riportata di seguito.

Anzitutto è necessario stabilire una regola in base alla quale una data simulazione possa essere ritenuta alla portata del calcolatore oppure no. Ciò non è affatto dissimile dal delineare una classe di problemi trattabili, come abbiamo fatto nella sezione precedente. Feynman propone che il numero di elementi che costituiscono il calcolatore debba essere proporzionale al volume del sistema che si intende simulare: se il volume del sistema raddoppia, le dimensioni del calcolatore non possono crescere esponenzialmente.

Poiché in meccanica quantistica le probabilità hanno un ruolo importante, può essere interessante approcciare il problema domandandosi inizialmente se un computer sia in grado di simulare efficacemente le probabilità, indipendentemente dal contesto fisico.

Consideriamo dunque un sistema costituito da R particelle. Per descrivere la probabilità della realizzazione di un certo evento dobbiamo conoscere la probabilità che le R particelle occupino le posizioni $x_1, x_2, \ldots, x_R$ al tempo t. Se il numero di posizioni che le particelle possono occupare è N avremo N^R configurazioni, a ciascuna delle quali è associata una probabilità. Più realisticamente, poiché in ogni punto dello spazio sarà presente informazione nella forma di campo elettrico, magnetico ecc., il numero di configurazioni sarà N^N. Pertanto, se raddoppiamo la dimensione del problema, il numero di configurazioni, e dunque la sua complessità (o il numero di componenti del computer, nella terminologia di Feynman), cresce esponenzialmente.

La difficoltà permane se il computer deve simulare un sistema quantistico; difatti la descrizione del comportamento di questo sistema sarà data dalla ampiezza di probabilità $\psi(x_1, x_2, \ldots, x_R, t)$ la quale, per utilizzare le parole di Feynman, "cannot be simulated with a normal computer with a number of elements proportional to R or proportional to N".

L'idea del computer quantistico consiste nel capovolgere la prospettiva di Feynman, ovvero nel domandarsi se un calcolatore "built of quantum mechanical elements which obey quantum mechanical laws" sia in grado di risolvere problemi non trattabili, in base alla definizione data nella sezione precedente, in tempo polinomiale.

La ragione per cui questa speranza appare fondata risiede nella similitudine che intercorre tra la complessità insita nella simulazione di un sistema quantistico e quella che caratterizza un problema non trattabile: entrambe crescono esponenzialmente al crescere della dimensione rispettivamente del sistema e del problema.

La supposizione secondo la quale un computer quantistico sarebbe stato in grado di risolvere problemi non trattabili in tempo polinomiale si rivelò essere fondata. Nel 1994 Shor [4] propose un algoritmo quantistico per la fattorizzazione di numeri interi, un problema che, come abbiamo visto nella precedente sezione, si crede non possa essere risolto da una macchina deterministica in

tempo polinomiale[5]. Successivamente vennero costruite macchine in grado di eseguire questo algoritmo; ad oggi il più grande numero che sia stato fattorizzato è 5615 [5].

1.3 Il boson-sampling

La ECT afferma che ogni sistema fisico può essere efficacemente simulato da una macchina di Turing. Tale ipotesi presenta una similitudine con l'affermazione secondo la quale ogni problema può essere risolto da una macchina deterministica in tempo polinomiale, ovvero $\mathsf{P} = \mathsf{NP}$. Poiché questa uguaglianza non è stata ancora dimostrata, ed anzi sembrerebbe plausibile supporre che sia falsa, anche la validità della ECT è stata messa in discussione. In effetti il successo incontrato dalla realizzazione sperimentale dell'algoritmo di Shor sembrerebbe confortare questa posizione. Tuttavia è importante osservare come difficoltà tecniche abbiano finora prevenuto la realizzazione di una macchina che fosse in grado di scomporre numeri al di fuori della portata di un calcolatore classico.

Il **boson-sampling** è un modello computazionale quantistico la cui realizzazione sperimentale risulta relativamente semplice. L'algoritmo che il boson-sampling implementa è classicamente difficile e la sua complessità cresce esponenzialmente al crescere del numero n di fotoni in ingresso al boson-sampler. Certificare il funzionamento di un boson-sampler nel limite di grandi n costituirebbe una prova sperimentale della falsità della ECT; al momento considerazioni fisiche sulle quali torneremo più avanti suggeriscono che tale limite non sia ottenibile. Cionondimeno il boson-sampling, nel limite in cui il sistema quantistico risulta più performante del calcolatore, rappresenta un valido modello computazione quantistico, ancorché non universale.

1.3.1 Il modello

Il boson-sampling è un modello computazionale quantistico che sfrutta bosoni non interagenti [6]. Questo modello può essere implementato usando una rete di elementi ottici, quali ad esempio beamsplitter, attraverso la quale far passare fotoni. I fotoni vengono infine misurati per determinarne il numero in ciascuna delle uscite (o modi) della rete.

Per comprendere meglio il modello possiamo pensare alla macchina di Galton, in Figura 1.2. Questa macchina consiste in un piano verticale sul quale sono fissati perpendicolarmente dei pioli in modo da formare una griglia. Da una fessura posta in cima al piano vengono fatte cadere delle biglie le quali, urtando i pioli, si dirigono verso destra o verso sinistra. Sul fondo sono collocati dei contenitori nei quali le biglie si depositano, una sopra l'altra, formando delle pile. L'input della macchina di Galton è la specifica disposizione dei pioli

[5] Il "si crede" è d'obbligo in quanto la non esistenza di un algoritmo classico polinomiale non è stata provata.

[6] Questa sezione deriva in gran parte da [6].

mentre l'output è fornito dal numero di biglie in ciascun contenitore. Se supponiamo che le biglie non interagiscano fra di loro la macchina di Galton descrive essenzialmente il modello che intendiamo studiare. Nel nostro caso le biglie saranno ovviamente sostituite da **bosoni non interagenti**, inoltre la disposizione degli elementi ottici che codifica il problema di interesse non sarà necessariamente regolare; in ultimo i bosoni, a differenza delle biglie, non partiranno dalla medesima locazione.

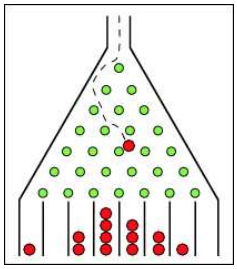

Figura 1.2: La macchina di Galton, solitamente usata per illustrare la distribuzione binomiale, può essere pensata come un rudimentale computer. Immagine proveniente da [6].

Poiché il boson-sampling non richiede alcuna esplicita interazione tra coppie di bosoni, esso elude quello che a lungo è stato il maggior ostacolo tecnologico alla realizzazione dei computer quantistici, ovvero come "far parlare" coppie di particelle non interagenti, come i fotoni, attraverso porte quantistiche. Questo aspetto del modello può sembrare paradossale: se non vi sono interazioni tra bosoni come si è in grado di produrre *entanglement*, dal quale derivano i vantaggi di un computer quantistico?

Come sappiamo scambiare due bosoni identici non ha effetti; pertanto quando scriviamo lo stato di n bosoni nella base dello spazio di Fock - la base ovvero consistente degli stati nella forma $|s_1, \ldots, s_m >$, dove s_i è il numero di bosoni nella i-esima posizione - anche se i bosoni non interagiscono "apparirà" esservi entanglement tra due posizioni qualunque.

Matematicamente, per trovare la probabilità associata a un qualunque output, risulta necessario calcolare il permanente di una matrice $n \times n$. Tale necessità appare anche nel modello classico: supponiamo di avere n biglie e che vi siano n contenitori sul fondo della macchina di Galton. Sia a_{ij} la probabilità che la biglia i-esima finisca nel j-esimo contenitore. Allora la probabilità che in ogni contenitore vi sia una biglia è data da

$$\sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)} \tag{1.2}$$

dove $\sigma(i)$ sono le possibili permutazioni dei contenitori contenenti ciascuno una delle n biglie. Definendo A come la matrice $A = (a_{ij})_{i,j \in [n]}$, la (1.2) è

proprio il permanente di A. Nel caso della macchina di Galton, e più generalmente nel caso classico, le probabilità a_{ij} saranno numeri reali non negativi; diversamente nel caso quantistico esse sono numeri complessi. Se i coefficienti sono reali non negativi è possibile approssimare il permanente in tempo polinomiale, mentre ciò non è possibile se la matrice è complessa, il che rende il boson-sampling un problema difficile da risolvere classicamente.

1.3.2 Il boson-sampling e la ECT

Un problema di boson-sampling può essere implementato costruendo una rete di elementi ottici al fine di indurre una trasformazione unitaria, caratterizzata da una matrice $m \times m$ che chiamiamo U, sullo stato degli n fotoni in ingresso. La riuscita dell'esperimento può essere appurata verificando che le probabilità degli stati finali coincidano con i permanenti delle sotto-matrici $n \times n$ di U. Per valori di n grandi è stato dimostrato [2] che non è possibile simulare classicamente l'esperimento.

A questo punto è naturale domandarsi quanto grande n debba essere perché si possa giungere a conclusioni interessanti in merito alla validità della ECT. In realtà il contenuto della ECT è rilevante nel limite di $n \to \infty$; allo stesso tempo per il boson-sampling non è stato ancora trovato un certificato che possa essere provato in tempo polinomiale. Dunque, se n è sufficientemente grande da impedire a un computer classico di risolvere il problema del boson-sampling, allo stato attuale non è possibile nemmeno verificare che il boson-sampler stia risolvendo il problema correttamente.

Bisogna notare che per $20 \leq n \leq 30$ un computer classico sarebbe certamente in grado di verificare i risultati del boson-sampler; allo stesso tempo il boson-sampler risulterebbe assai più efficiente del computer classico, che impiegherebbe milioni di operazioni per risolvere il problema. Pertanto, per quanto provare la falsità della ECT è formalmente impossibile, la riuscita di un esperimento di boson-sampling per i valori di n indicati rappresenterebbe un'importante evidenza sperimentale.

1.4 Validazione di un boson-sampler

Validare, o certificare, un boson-sampler significa verificarne il funzionamento tramite applicazione di un apposito protocollo. Scegliere un protocollo appropriato non è semplice. Calcolare classicamente la distribuzione campionaria è un problema la cui complessità cresce esponenzialmente con il numero di fotoni n. Allo stesso tempo la misura di osservabili efficacemente predicibili non sempre mette in risalto la capacità del boson-sampler di sfruttare l'interferenza bosonica, dalla quale deriva il vantaggio computazionale rispetto a dispositivi classici. Il protocollo deve dunque avere due caratteristiche [7]:

1. Deve essere **efficiente**, ovvero deve poter essere eseguito in tempo ottimale (le risorse impiegate devono scalare polinomialmente con n).

2. Deve essere **stringente**, ovvero non deve lasciare adito a interpretazioni alternative che escludano l'interferenza bosonica.

1.4.1 La matrice unitaria

Il nostro primo obiettivo consiste nell'introdurre formalmente il concetto di **matrice unitaria**[7]. Anzitutto immaginiamo di volere rappresentare le N realizzazioni di un possibile evento. Ad esempio potremmo servici del vettore $(p_1, p_2, \ldots, p_N)$, dove p_i è la probabilità associata alla i-esima realizzazione, ovvero un numero reale non negativo. Questo vettore dovrà soddisfare una proprietà, ovvero la sua norma dovrà essere pari a uno. Ovviamente è possibile definire più di una norma ma, poiché in questo caso le probabilità sono classiche, è naturale scegliere la norma 1:

$$\sum_{i=1}^{N} p_i = 1 \tag{1.3}$$

Per ragioni di semplicità consideriamo ora il vettore bidimensionale $(p, 1-p)$. Una qualunque operazione che mappi questo vettore in un altro vettore che soddisfi la norma 1 può essere rappresentata da una **matrice stocastica**. Una matrice stocastica è caratterizzata da due proprietà: i suoi coefficienti sono reali non negativi e la somma degli elementi di ciascuna colonna è pari a uno. A titolo d'esempio riportiamo l'operazione nota come *bit flip* applicata al nostro vettore.

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} p \\ 1-p \end{pmatrix} = \begin{pmatrix} 1-p \\ p \end{pmatrix} \tag{1.4}$$

Supponiamo ora di imporre la condizione di normalizzazione servendoci della norma 2, ovvero la norma euclidea. La matrice unitaria è la matrice più generale che mappa un vettore normalizzato secondo la norma euclidea in un altro vettore conservando la normalizzazione. In altre parole la matrice unitaria sta alla norma 2 come la matrice stocastica sta alla norma 1. Un vettore bidimensionale normalizzato secondo la norma 2 prende il nome di **qubit**. (α, β) è un qubit se $\alpha^2 + \beta^2 = 1$. Alternativamente possiamo utilizzare la usuale notazione dei ket $\alpha|0> + \beta|1>$.

1.4.2 Un protocollo efficiente ma non stringente

Introduciamo la notazione che utilizzeremo. Sia m il numero di modi della rete. Indichiamo con il vettore $\vec{j} = (j_1, \ldots, j_n)$ le porte attraverso le quali n bosoni entrano nella rete e con $\vec{k} = (k_1, \ldots, k_n)$ quelle attraverso cui escono $(n \leq m)$. Un evento sarà caratterizzato dal vettore $\vec{k}$ delle porte attraversate dai bosoni in uscita. Il test consiste nel campionare lo spazio degli eventi $\vec{k}$ e nel certificare la distribuzione ottenuta in base al protocollo stabilito. Sia

[7] Nel fare ciò seguiamo l'approccio presentato in [8].

U la matrice unitaria $m \times m$ associata alla trasformazione indotta dalla rete. La probabilità $P_B(\vec{j}, \vec{k}; U)$ che i bosoni attraversino le uscite $\vec{k}$ essendo stati preparati nei modi $\vec{j}$ è data da:

$$P_B(\vec{j}, \vec{k}; U) = |permanent(M)|^2 \tag{1.5}$$

Dove M è la sottomatrice di U contenente le righe e le colonne che corrispondo agli ingressi $\vec{j}$ e alle uscite $\vec{k}$.

Il protocollo che intendiamo discutere permette di distinguere un boson-sampler da un *uniform-sampler* [9]. Un uniform-sampler è un dispositivo che genera eventi $\vec{k}$ a partire da una distribuzione uniforme. Un modo per distinguere i due dispositivi consiste nel misurare il numero medio di particelle in ciascun modo.

$$\langle \hat{n}_k \rangle = \sum_{l=1}^{n} |U_{j_l,k}|^2 \tag{1.6}$$

avendo usato la notazione $M_{l,q} = U_{j_l,k_q}$. Il protocollo, che consiste appunto nella misura della osservabile (1.6), è efficiente in quanto non risulta necessario calcolare alcun permanente. Tuttavia la misura di osservabili di singola particella, come (1.6), pur riflettendo alcune proprietà della matrice U, risulta insensibile alla interferenza bosonica. In altre parole è possibile replicare i risultati attraverso un esperimento classico usando particelle distinguibili: preparando ogni particella in un modo j_l la probabilità di avere un evento $\vec{k}$ è $p_{j_l,k} = |U_{j_l,k}|^2$. Per questa ragione il protocollo non è stringente.

1.4.3 Certificazione attraverso le matrici di Fourier

In questa sezione presentiamo un protocollo efficiente e stringente [7]. Il protocollo sarà basato sulla misura di osservabili predicibili efficacemente e sensibili alla interferenza bosonica. Poiché le probabilità di eventi governati dalla interferenza bosonica sono difficili da predire, non useremo più, come nel caso precedente, una matrice unitaria U casuale, ma adotteremo la matrice di Fourier di dimensioni $m = n^p$

$$U_{l,q}^{Four} = \frac{1}{\sqrt{m}} \exp\left(i \frac{2\pi l q}{m}\right) \tag{1.7}$$

le cui proprietà di simmetria alleviano le difficoltà relative al calcolo del permanente. Inoltre si impone che gli stati iniziali siano ciclicamente simmetrici.

$$\vec{j} = (1, n^{p-1} + 1, 2n^{p-1} + 1, \ldots, (n-1)n^{p-1} + 1) \tag{1.8}$$

Ad esempio, nel caso di una rete a quattro modi con due fotoni, otteniamo:

$$\vec{j} = (1, 3) \tag{1.9}$$

ovverosia gli stati di Fock ammessi sono $(1, 0, 1, 0)$ e $(0, 1, 0, 1)$. La simmetria ciclica rimane intatta al passaggio dei fotoni attraverso la rete. Pertanto anche

le uscite dovranno essere cicliche, il che sopprime molti possibili eventi. La condizione che deve essere soddisfatta è

$$\mod \left(\sum_{l=1}^{n} k_l, n \right) = 0 \tag{1.10}$$

In altre parole, se il numero di bosoni è pari (dispari), la somma degli indici delle uscite deve essere pari(dispari). Tornando all'esempio di prima ($n = 2, m = 4$), le uscite ammesse sono $\vec{k} = (1, 3)$ e $\vec{k} = (2, 4)$ in quanto il numero di fotoni è pari e sommando gli indici si ottiene rispettivamente $1 + 3 = 4$ e $2 + 4 = 6$.

La (1.10) è una generalizzazione dell'effetto HOM. Il grado di violazione della legge di soppressione (1.10) è espresso da:

$$\nu = \mathcal{N}_{forbidden}/\mathcal{N}_{runs} \tag{1.11}$$

ovvero il rapporto tra il numero di eventi che violano la legge di soppressione e il numero di eventi totale. Un boson-sampler ideale sarà caratterizzato da $\nu = 0$, mentre un dispositivo fraudolento non sarà in grado di soddisfare la legge di soppressione, il che comporterà un certo numero di violazioni. Poiché gli eventi non permessi possono essere predetti efficacemente da un computer classico anche per un gran numero di fotoni ($n \approx 10^6$), è possibile discriminare i risultati di un boson-sampler da quelli di un dispositivo fraudolento. In particolare, nel caso di particelle distinguibili, per $n = 2$ e $m = 4$ varrà $\nu = 0.5$. Il protocollo descritto risulta dunque essere sia efficiente che stringente.

1.5 L'effetto HOM

L'effetto HOM permette di misurare l'intervallo temporale che separa due fotoni prodotti dalla Spontaneous parametric down-conversion (SPDC), e di conseguenza la lunghezza d'onda del pacchetto, con una precisione di 1fs [10].

1.5.1 L'esperimento

L'apparato sperimentale è presentato in Figura (1.3). Un fascio di luce coerente di frequenza ω_0 incide su un cristallo non lineare (in questo caso il diidrogenofosfato di potassio). Alcuni dei fotoni incidenti si separano in due fotoni di energia minore, che prendono il nome di **signal** e **idler**, le cui frequenze indichiamo con ω_1 e ω_2. Per la conservazione dell'energia dovrà valere

$$\omega_0 = \omega_1 + \omega_2 \tag{1.12}$$

I due fotoni sono riflessi dagli specchi M_1 e M_2 e attraversano il beamsplitter BS in modo tale che i fasci sovrimpressi interferiscano e vengano rilevati dai fotorivelatore $D1$ e $D2$. L'esperimento consiste nella misura delle coincidenze tra i due fotorilevatori quando il beamsplitter è fatto scorrere lungo una guida rettilinea, in direzione di un rivelatore o dell'altro. Chiamiamo $\pm c\delta\tau$ lo spostamento, piccolo, del beamsplitter.

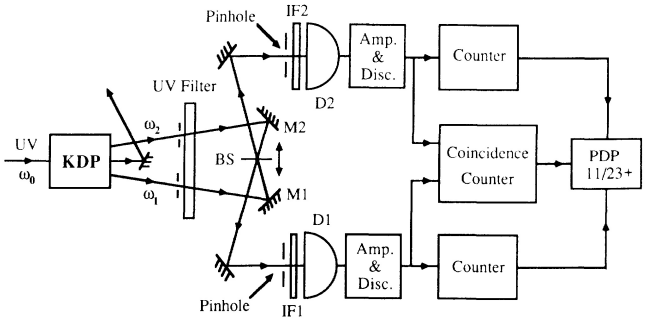

Figura 1.3: Apparato sperimentale dell'esperimento HOM [**?**].

La frequenza ω_0 è ben definita, tuttavia le frequenze dei fotoni signal e idler non lo sono. Lo scopo dei filtri passa banda $IF1$ e $IF2$ è proprio quello di determinare le due frequenze. I filtri sono dell'ordine di 5×10^{12}Hz, il che corrisponde a un tempo di coerenza per ciascun fotone dell'ordine di 100fs. Pertanto le ampiezze di probabilità dei fotoni interferiscono solo se si sovrappongono all'interno di questo intervallo. Da ciò si deduce che la precisione con la quale è possibile misurare l'intervallo che separa i due fotoni è determinata principalmente dalla frequenza dei filtri passa banda.

Introduciamo la nomenclatura per i modi della rete: quelli di ingresso saranno indicati con $01, 02$, quelli di uscita con $1, 2$. Supponiamo che la luce sia monocromatica. Se lo stato di input che rappresenta i fotoni generati dalla SPDC è lo stato di Fock $|1_{01}, 1_{02}>$ allora lo stato finale è dato da:

$$|\psi_{out}\rangle = (R-T)|1_1, 1_2> + i\sqrt{2RT}|2_1, 0_2> + i\sqrt{2RT}|0_1, 2_2> \tag{1.13}$$

dove R e T sono rispettivamente la riflettività e la trasmittività del beamsplitter, con $R+T=1$. Per un beamsplitter con $R=T=0.5$ il primo termine è zero a causa della interferenza distruttiva delle ampiezze di probabilità dei due fotoni. Pertanto idealmente non dovrebbero registrarsi coincidenze.

In pratica i fotoni risultanti dalla SPDC non sono mai monocromatici. Conviene dunque rappresentare lo stato dei due fotoni prodotti attraverso una combinazione lineare:

$$|\psi> = \int d\omega \phi(\omega_1, \omega_0 - \omega_1)|\omega_1, \omega_0 - \omega_1> \tag{1.14}$$

$\phi(\omega_1, \omega_2)$ è una funzione che "pesa" l'integrando e ha un picco per $\omega_1 = \omega_2 = \omega_0/2$. E' possibile dimostrare che il numero di coincidenze osservate e

dato da:

$$N_c = C\left[R^2 + T^2 - 2RT\frac{\int_{-\infty}^{\infty} g(\tau)g(\tau - 2\delta\tau)d\tau}{\int_{-\infty}^{\infty} g^2(\tau)d\tau}\right] \tag{1.15}$$

$G(\tau)$ è la trasformata di Fourier della funzione $\phi(\omega_0/2 + \omega, \omega_0/2 - \omega)$:

$$G(\tau) = \int \phi(\omega_0/2 + \omega, \omega_0/2 - \omega)e^{-i\omega\tau}d\omega \tag{1.16}$$

$g(\tau)$ è definita come $g(\tau) = G(\tau)/G(0)$. t è il tempo al quale un fotone viene rilevato da $D1$ e $t+\tau$ il tempo al quale il fotone successivo viene rilevato da $D2$. $N_c = C(R-T)^2 = 0$[8] quando $\delta\tau = 0$, ovvero quando il beamsplitter si trova in posizione centrale. Quando $\delta\tau$ supera apprezzabilmente il tempo di correlazione $g(\tau)$ risulta $N_c = (T^2 + R^2)$. Di conseguenza, graficando l'andamento di N_c in funzione dello spostamento del beamsplitter, dovrebbe essere possibile osservare una buca in corrispondenza di $\delta\tau = 0$. La buca è un effetto della interferenza tra le funzioni dei due fotoni, che sono in questo caso indistinguibili (Figura (1.4)).

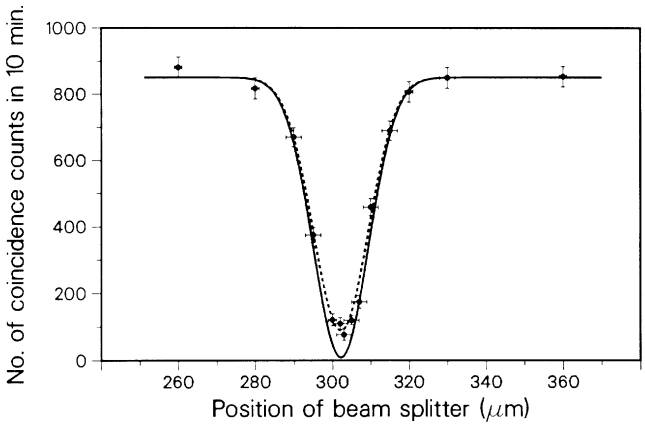

Figura 1.4: In figura è graficato il numero di coincidenze in funzione dello spostamento del BS. Quando il BS è in posizione centrale si osserva una buca, dovuta al fenomeno della interferenza tra le ampiezze di probabilità dei due fotoni [**?**].

La larghezza a metà altezza della buca fornisce infine una misura della larghezza di banda del pacchetto di fotoni.

[8] Ricordando che il beamsplitter è stato scelto in modo tale che $R = T$

1.6 Circuiti integrati quantistici

La costruzione di una rete di dispositivi ottici che soddisfi i requisiti di stabilità e accuratezza necessari al raggiungimento della sensibilità permessa dalla meccanica quantistica è assai difficoltosa [11].

Un approccio alternativo consiste nell'adottare circuiti ottici integrati. Un laser UV può essere utilizzato per iscrivere guide d'onda miniaturizzate in un chip costituito da strati di silicio drogati sopra un substrato di silicio. Questo tipo di circuiti è stato utilizzato in esperimenti nei quali i fotoni avevano una data polarizzazione. D'altro canto in molti casi d'interesse il grado di libertà sullo stato di polarizzazione gioca un ruolo importante. Per questa ragione è auspicabile l'utilizzo di un circuito che funzioni indipendentemente dallo stato di polarizzazione dei fotoni.

Le guide d'onda iscritte tramite laser UV sono affette da birifrangenza intrinseca. La birifrangenza è una conseguenza della tensione generata dal non perfetto allineamento dei reticoli costituenti gli strati di silicio drogato, i quali formano uno *stack* che poggia sul substrato di silicio. La birifrangenza causa la dispersione dei fasci di luce, dipendente dalla polarizzazione; di conseguenza il funzionamento del circuito non sarà più indipendente dalla polarizzazione.

In aggiunta a ciò bisogna osservare che la propagazione delle strutture birifrangenti può causare la decoerenza dei fotoni generati dalla SPDC.

Queste ragioni hanno fatto sì che si cercasse una tecnica alternativa per la fabbricazione di circuiti ottici integrati. La tecnica in questione consiste nell'usare impulsi infrarossi, la cui durata è dell'ordine del femtosecondo, focalizzati sul substrato del chip per indurre fenomeni di assorbimento non lineari basati sulla ionizzazione multifotonica e a valanga. Questi fenomeni portano alla formazione di plasma e all'assorbimento di energia in una piccola regione intorno al punto focale, causando una modificazione permanente e localizzata del *bulk*. Un'opportuna modifica dei parametri che caratterizzano il processo permette di incrementare in maniera continua l'indice di rifrazione; le guide d'onda sono prodotte traslando il substrato rispetto al laser.

L'approccio che abbiamo descritto ha diversi vantaggi.

- E' una tecnica *maskless*, il che permette di realizzare prototipi in tempi brevi.
- Le guide d'onda possono essere fabbricate in un solo passaggio.
- Può produrre chip che si estendono in tre dimensioni.
- Possono essere prodotte guide d'onda con un profilo trasverso circolare che permette la propagazione di fasci gaussiani con una qualunque polarizzazione.

Il vetro borosilicato è stato scelto quale substrato in quanto non è stata mai osservata la formazione di nanograte, che potrebbero causare birifrangenza. Inoltre gli impulsi laser inducono una diffusione termica isotropa e la fusione del

materiale intorno al punto focale, producendo una guida d'onda la cui sezione trasversale è circolare senza il bisogno di modellare il fascio laser. E' possibile ottenere guide d'onda con una perdita molto bassa traslando il substrato alla velocità d $1 - 5$cm/seec.

Per una lunghezza d'onda di 800nm le guide d'onda supportano un singolo fascio gaussiano di profilo circolare con un diametro di 8μm a $1/e^2$. La grandezza usata per caratterizzare la birifrangenza è di un ordine inferiore rispetto al valore del chip con substrato di silicio.

Le guide d'onda del chip in Figura (1.5) inizialmente hanno una distanza relativa di 250μm; nella regione dove le ampiezze si sovrappongono la distanza si riduce a 7μm. La distanza è la più piccola che prevenga allo stesso tempo il sovrapporsi delle guide d'onda. Questa configurazione è stata scelta per minimizzare la sensibilità a difetti di fabbricazione e per ottenere una regione di interazione la più corta possibile, dal momento che nel futuro dispositivi quantistici più complessi saranno costituiti da numerosi componenti collegati a cascata. La capacità del chip di preservare la polarizzazione dei fasci entranti è stata verificata misurando il grado di polarizzazione G e ottenendo $G \geq 99.8\%$.

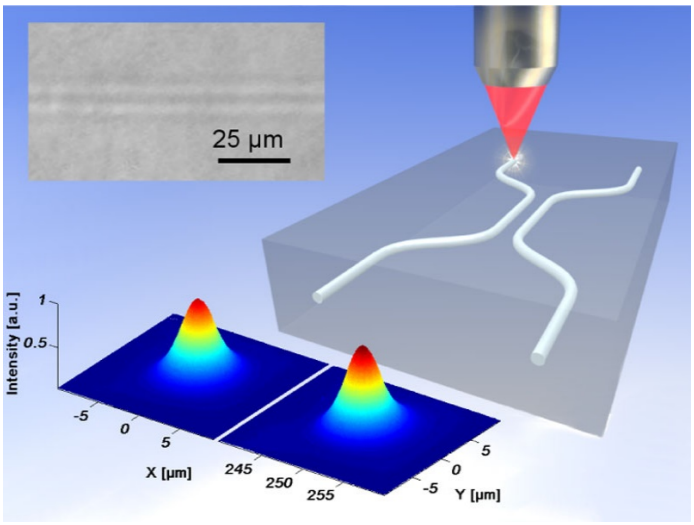

Figura 1.5: Un circuito integrato ottico, più esattamente un *directional coupler*, con substrato in vetro borosilicato. Nella parte in bassa è mostrato il profilo di intensità dei modi in uscita quando la luce è inviata in un solo ingresso. E' possibile apprezzare la forma simmetrica della gaussiana nonché lo splitting tra i due modi. Nella parte in alto è mostrata un'immagine al microscopio delle due guide d'onda nella regione di interazione. [11].

Capitolo 2

Esperienza di laboratorio

L'esperienza è consistita nella verifica della legge di soppressione di alcune particolari configurazioni delle uscite di un chip integrato quantistico. Il chip è un interferometro multimodale ($m = 4$) implementante una Quantum Fourier Trasform (QFT). La legge di soppressione, che è una generalizzazione dell'effetto HOM (1.5), per il quale valeva $m = 2$, è un effetto dell'interferenza bosonica. La verifica di tale legge rappresenta pertanto un protocollo stringente ed efficiente (1.4.3) che permette di certificare il corretto funzionamento del chip, a tutti gli effetti un boson-sampler.

Schematicamente, l'esperienza si articola in due parti. Nella prima è stato montato il chip e sono state eseguite le operazioni di taratura. Nella seconda si è proceduto all'acquisizione dati, ovverosia al conteggio delle coincidenze dei fotoni in uscita dal chip per ricavare le misure di visibilità grazie alle quali, insieme ai *coupling*, è stato possibile eseguire la tomografia del chip. Per tomografia intendiamo la ricostruzione della matrice unitaria caratterizzante il chip.

2.1 Apparato sperimentale

Anzitutto è bene osservare che durante la prima fase dell'esperienza può tornare utile (ad esempio per verificare l'allineamento del chip rispetto al *fiber array*) iniettare nel chip un fascio di luce, piuttosto che i fotoni generati dalla conversione parametrica. Di volta in volta nel seguito ci premureremo di spiegare le ragioni di tale scelta.

Negli altri casi i fotoni sono generati nel processo di conversione parametrica. Un laser emette a 785nm, viene convertito in un primo cristallo in un fascio a 392.5nm attraverso un processo di generazione di seconda armonica e infine va a pompare il cristallo di generazione delle coppie di fotoni. Ciascuno dei due fotoni attraversa un Beam-splitter polarizzatore (PBS), che separa la componente con polarizzazione orizzontale da quella con polarizzazione verticale. I fotoni viaggiano in delle fibre ottiche singolo modo. La polarizzazione dei fotoni all'uscita delle fibre può essere leggermente cambiata, questo a causa delle curve

che le fibre compiono. Ovviamente è importante che i fotoni abbiano la stessa polarizzazione, perché siano indistinguibili (e dunque perché interferiscano all'interno del chip). E' possibile agire su dei compensatori di polarizzazione alle quali le fibre sono attaccate.

I fotoni escono dalle fibre ottiche ed entrano in delle linee di ritardo. Infatti uno degli estremi può traslare lungo una guida rettilinea, controllato da un apposito software. Lo scopo di tale operazione consiste nel variare il grado di distinguibilità dei fotoni al fine di osservare la legge di soppressione, e dunque la "buca" (vd. Figura (1.4)), prevista dall'effetto HOM.

I fotoni attraversano le fibre collegate al fiber array ed entrano nel chip. All'uscita del chip si trova un secondo fiber array, le cui fibre sono collegate a dei fotorivelatori. Questi sono collegati a una scatola delle coincidenze, i cui conteggi sono inviati a un PC, attraverso cavi coassiali la cui lunghezza può essere variata al fine di compensare eventuali ritardi accumulati da uno fotone rispetto all'altro.

2.2 Montaggio del chip

Il chip presenta due strutture di tipo diverso: una guida rettilinea e la struttura che implementa la QFT. Per ciascun tipo vi è più di una struttura; un foglio fornito dal laboratorio dove il chip è stato fabbricato indica la posizione della struttura migliore rispetto a un'ablazione esterna. L'ablazione serve dunque a posizionare approssimativamente il chip rispetto al fiber array. Fatto ciò si inietta nel chip un fascio di luce e con una camera ccd si osservano i quattro *spot* in corrispondenza dei modi. Come si vedrà nel seguito, è possibile modificare l'allineamento del chip agendo su delle manopole che controllano i gradi di libertà della base su cui è poggiato. In questo modo è possibile traslare il chip parallelamente al fiber array in modo che la luce entri nella struttura con la minor dispersione.

2.3 Operazioni preliminari

In questa sezione descriveremo tutti gli accorgimenti che sono stati adottati prima di iniziare l'acquisizione dati al fine di garantire la riuscita dell'esperienza.

Per verificare che le polarizzazioni siano opportunamente compensante si pone un PBS davanti le uscite del chip e con un *power meter* si verifica che una delle due componenti sia nulla.

La base su cui poggia il chip è contraddistinta da sei gradi di libertà: tre rotazionali e tre traslazionali. In altre parole, se si immagina una terna costituita dai vettori che definiscono il piano del chip e un terzo vettore ad esso ortogonale, è possibile ruotare il chip intorno a ciascuno dei tre assi così definiti, o traslarlo lungo gli stessi, agendo su delle apposite manopole. In questo modo è possibile allineare il chip rispetto al fiber array, in modo che non vi sia dispersione dei fasci di luce. Per verificare che l'allineamento sia corretto si usa luce classica

e si copre alternativamente uno degli ingressi del chip, semplicemente ponendo un cartoncino all'interno di una linea di ritardo in modo da bloccare il laser. Dopodiché utilizzando la camera ccd è possibile verificare che l'intensità degli spot sia circa la stessa per i due ingressi. Se così non è si può agire sulle manopole per allineare il chip.

L'allineamento rispetto alle fibre collegate ai fotorilevatori non è altrettanto sensibile. Infatti queste fibre sono multi modo e il loro diametro e di circa 50μm. Un leggero spostamento del tavolo o anche un cambiamento di temperatura può disallineare il chip rispetto alle fibre singolo modo, di diametro 5μm, mentre nel caso delle multi modo il problema non si pone.

In generale è possibile dare una misura dell'attenuazione del fascio di luce facendo il rapporto, che prende il nome di efficienza, tra l'ampiezza misurata all'uscita del chip e all'interno della linea di ritardo[1][2]. Vari fattori possono contribuire a una perdita di efficienza. Uno di questi è la dispersione dovuta al disallineamento di una delle estremità della linea di ritardo, cui è possibile porre rimedio servendosi di una chiave a brugola.

Un altro aspetto di cui tener contro è il cambiamento di indice di rifrazione che si ha all'interfaccia tra il fiber-array e il chip, con conseguente dispersione per effetto della legge di Snell. Questo cambiamento può essere attenuato servendosi di un gel, che prende appunto il nome di *index matching*, il cui indice di rifrazione ha un valore intermedio tra quello delle fibre e quello dell'aria. Tuttavia si è constato che l'utilizzo dell'index matching non ha apportato alcun beneficio significativo e per questa ragione è stato accantonato.

In ultimo per provare a migliorare l'efficienza si possono pulire gli ingressi del chip utilizzando del metanolo.

In ogni caso è importante ricordare che una piccola dispersione si avrà comunque nel passaggio della luce all'interno del chip, causata dal *bending* delle guide d'onda e dalla loro non perfetta circolarità.

2.4 Acquisizione dati

Descriviamo brevemente il processo di acquisizione dati.

Per verificare la legge di soppressione dobbiamo variare la lunghezza di una delle linee di ritardo, traslando un suo estremo. Quando la distanza temporale che separa i fotoni entranti nel chip sarà minore del tempo di coerenza, allora i fotoni saranno indistinguibili e interferiranno. Il chip implementa una QFT il cui effetto è quello di sopprimere alcune configurazioni delle uscite. Tale effetto è frutto dell'interferenza bosonica pertanto, in corrispondenza delle configurazioni soppresse[3], dovrà essere possibile osservare una diminuzione (idealmente un azzeramento) del numero di coincidenze solo se il ritardo introdotto dalla

[1] Bisogna comunque ricordare che l'intensità di uno spot sarà solamente una frazione di quella che si avrebbe in assenza degli altri tre, cosa di cui bisogna tener conto nel calcolo della efficienza dividendo per la suddetta frazione.

[2] E' possibile collegare le fibre direttamente al power meter piuttosto che ai fotorilevatori per eseguire la misura dell'ampiezza all'uscita del chip.

[3] Ovvero per coppie di uscite non cicliche.

linea è tale da non rendere i fotoni distinguibili[4]. Nostro scopo sarà misurare il numero di coincidenze in funzione della lunghezza della linea di ritardo o, più esattamente, della sua variazione rispetto al valore al quale il numero di coincidenze è minimo.

Prima di fare ciò è bene assicurarsi il corretto allineamento della sorgente. Poiché il chip introduce inevitabilmente un minimo di dispersione, vorremmo iniettare i fotoni direttamente nelle fibre multimodo collegate ai fotorilevatori. Allo stesso tempo non è possibile spostare il chip senza perdere l'allineamento. La soluzione consiste nell'utilizzare degli specchi la cui base magnetica permette di fissarli alle linee di ritardo. Gli specchi riflettono i fotoni che viaggiano all'interno delle linee inviandoli a delle linee di ritardo collegate direttamente ai fotorilevatori tramite delle fibre multimodo, scavalcando in questo modo il chip.

Un programma permette di traslare un estremo della linea di ritardo; la lunghezza del singolo passo può essere impostata dall'utente. Un secondo programma restituisce i conteggi dei fotorilevatori collegati alla scatola delle coincidenze, nonché il numero di coincidenze contate. Per prima cosa bisogna trovare la lunghezza alla quale i fotoni interferiscono. L'operazione è eseguita manualmente, incrementando o decrementando la lunghezza della linea fino a quando non si registra una diminuzione del numero di coincidenze. Assumeremo che la posizione raggiunta dall'estremo della linea sotto queste condizioni sia la posizione zero. Fatto ciò è possibile stabilire un intervallo, centrato rispetto alla posizione zero, all'interno del quale far variare in maniera automatica lo spostamento. E' inoltre possibile fissare il tempo allocato alla acquisizione delle coincidenze per ogni singolo passo. In questo modo è stato ricostruito l'andamento del numero di coincidenze in funzione del ritardo tra i due fotoni, dal quale è possibile ricavare la visibilità, necessaria a eseguire la tomografia.

L'operazione è stata ripetuta per tutte le 36 combinazioni di coppie di ingressi e di coppie di uscite.

[4]In corrispondenza delle restanti configurazioni, ovvero per coppie di uscite cicliche, si registrerà al contrario un aumento del numero di coincidenze.

Capitolo 3

Tomografia del chip

In questo capitolo descriveremo il funzionamento del programma che ricostruisce la matrice unitaria implementata dal chip integrato quantistico. La ricostruzione, o **tomografia**, è resa possibile dalle misure di visibilità ottenute indirettamente dall'andamento del numero di coincidenze in funzione del ritardo temporale tra i fotoni. Inoltre è necessario usare le misure dei *coupling*. Ulteriori considerazioni in merito alla possibilità di non eseguire le misure per tutte le trentasei combinazioni di input e output verranno approfondite nel seguito.

Diamo anzitutto la definizione di visibilità. La **visibilità** è un parametro che varia tra 0 e 1 ed è definito come

$$\mathcal{V} = \frac{C - Q}{C} \tag{3.1}$$

dove C è il numero di coincidenze quando i fotoni sono distinguibili e Q è il numero quando non lo sono. Detto altrimenti, facendo riferimento alla Figura (1.4), C è il valore che il grafico assume in corrispondenza del *plateau* mentre Q è il minimo del grafico.

I **coupling** invece non sono altro che i *rate* che si registrano per ciascuna uscita iniettando un solo fotone nel chip.

Per verificare il funzionamento del programma che ricostruisce la matrice del chip è stato disegnato un secondo programma. Quest'ultimo genera una matrice unitaria *random*, della quale è in grado di fornire gli opportuni valori delle visibilità e dei coupling. Tali valori sono passati al primo programma, che ricostruisce la matrice random, da confrontare a questo punto con l'originale.

3.1 Algoritmo per la tomografia del chip

Dispositivi fotonici come il chip integrato quantistico della nostra esperienza sono tipicamente descritti da una matrice unitaria contenente informazioni sulle ampiezze e le fasi per le quali gli stati di input devono essere moltiplicati. La caratterizzazione della matrice unitaria è dunque molto importante.

L'algoritmo [1] che abbiamo implementato prevede che vengano iniettati coppie di fotoni all'interno del dispositivo per ottenere le misure di visibilità. I coupling possono invece essere ottenuti iniettando un solo fotone. Le informazioni sulle ampiezze sono ottenute dalle misure di singolo fotone mentre quelle sulle fasi dalle misure con due fotoni.

Prima di passare al caso generico per un numero di modi qualunque concentriamoci su un ipotetico dispositivo per il quale valga $m = 2$. Siano S_k i fotoni che vengono iniettati nell'ingresso k. Sia s_k l'efficienza, j la porta dalla quale esce il fotone S_k e $T_{j,k}$ la probabilità di trasmissione. In ultimo sia r_j l'efficienza con la quale i fotoni vengono rilevati e $R_{j,k}$ i rate dei conteggi (ovvero i coupling). Poiché $R_{j,k} = S_k s_k T_{j,k} r_j$, quattro misure dei rate sono sufficienti a imporre la seguente condizione su quattro valori di $T_{j,k}$, indipendentemente dalla efficienze.

$$\frac{T_{1,1} * T_{2,2}}{T_{1,2} * T_{2,1}} = \frac{R_{1,1} * R_{2,2}}{R_{1,2} * R_{2,1}} \tag{3.2}$$

Il caso di una matrice con due soli modi è interessante in quanto una matrice con m qualunque può essere costruita a partire da sottomatrici di ordine due. Pertanto per ora ci concentreremo su questa sottomatrice.

Sia S_2 la matrice unitaria che descrive il nostro ipotetico chip a due modi. Chiamiamo τ le ampiezze e $e^{i\alpha}$ i fattori di fase. Tenendo conto delle efficienze dei coupling e dei rilevatori, otteniamo la matrice P_2

$$P_2 = \begin{pmatrix} \sqrt{r_j} & 0 \\ 0 & \sqrt{r_g} \end{pmatrix} \begin{pmatrix} \tau_{j,k} e^{i\alpha_{j,k}} & \tau_{j,h} e^{i\alpha_{j,h}} \\ \tau_{g,k} e^{i\alpha_{g,k}} & \tau_{g,h} e^{i\alpha_{g,h}} \end{pmatrix} \begin{pmatrix} \sqrt{s_j} & 0 \\ 0 & \sqrt{s_g} \end{pmatrix} \tag{3.3}$$

Come la stessa notazione suggerisce righe diverse della matrice corrispondo a ingressi diversi mentre le colonne corrispondone alle uscite.

Se iniettiamo simultaneamente due fotoni nel dispositivo P_2 sappiamo che essi devono interferire. Pertanto, in base alla (1.5), possiamo concludere:

$$\begin{gathered} Q_{g,h,j,k} = |Per(P_2)|^2 = s_h s_k r_g r_j (\tau_{j,k}^2 \tau_{g,h}^2 + \tau_{g,k}^2 \tau_{j,h}^2) + \\ + s_h s_k r_g r_j \tau_{j,k} \tau_{j,h} \tau_{g,k} \tau_{g,h} \times 2\cos(\alpha_{j,k} - \alpha_{j,h} - \alpha_{g,k} + \alpha_{g,h}) \end{gathered} \tag{3.4}$$

La notazione adottata è la stessa della (3.1). Se invece introduciamo del ritardo tra i fotoni essi non interferiranno e avremo:

$$\begin{gathered} C_{g,h,j,k} = Per(|P_2|^2) = \\ = s_h s_k r_g r_j (\tau_{j,k}^2 \tau_{g,h}^2 + \tau_{g,k}^2 \tau_{j,h}^2) = R_{j,k} R_{g,h} + R_{g,k} R_{j,h} \end{gathered} \tag{3.5}$$

La visibilità sarà dunque data da

$$\begin{gathered} V_{g,h,j,k} = \frac{C_{g,h,j,k} - Q_{g,h,j,k}}{C_{g,h,j,k}} = \\ = -2\cos(\alpha_{j,k} - \alpha_{j,h} - \alpha_{g,k} + \alpha_{g,h}) \times \frac{\tau_{j,k}\tau_{j,h}\tau_{g,k}\tau_{g,h}}{\tau_{j,k}^2\tau_{g,h}^2 + \tau_{g,k}^2\tau_{j,h}^2} \end{gathered} \tag{3.6}$$

Come si evince dalla (3.6) la visibilità è insensibile alle efficienze.

Poiché le ampiezze τ sono le radici quadre delle probabilità di trasmissione T a partire dalla (3.2) è possibile imporre la seguente condizione (3.9)

$$X_{g,h,j,k} = \frac{R_{j,k}R_{g,h}}{R_{j,h}R_{g,h}} \tag{3.7}$$

$$x_{g,h,j,k} = \sqrt{X_{g,h,j,k}} \rightarrow \tag{3.8}$$

$$x_{g,h,j,k} = \frac{\tau_{j,k}\tau_{g,h}}{\tau_{j,h}\tau_{g,k}} \tag{3.9}$$

Da questa condizione deriva una relazione per gli angoli delle fasi:

$$y_{g,h,j,k} = x_{g,h,j,k} + x^{-1}_{g,h,j,k} \rightarrow \tag{3.10}$$

$$\cos(\alpha_{j,k} - \alpha_{j,h} - \alpha_{g,k} + \alpha_{g,h}) = -\frac{V_{g,h,j,k}y_{g,h,j,k}}{2} \tag{3.11}$$

Dal momento che il coseno è una funzione pari il segno dell'argomento non è noto. Tuttavia il segno di uno qualunque dei quattro angoli, ad esempio $\alpha_{g,h}$, può essere determinato se è noto il suo valore assoluto e se sono noti i valori e i segni dei restanti angoli. Anzitutto diamo un nome all'argomento del coseno: sia $\beta_{g,h,j,k} = |\alpha_{j,k} - \alpha_{j,h} - \alpha_{g,k} + \alpha_{g,h}|$. Allora il segno dell'angolo $\alpha_{g,h}$ può essere ricavato in questo modo:

$$\begin{aligned} sgn[\alpha_{g,h}] = sgn[|\beta_{g,h,j,k} - |\alpha_{j,k} - \alpha_{j,h} - \alpha_{g,k} - |\alpha_{g,h}||| - \\ -|\beta_{g,h,j,k} - |\alpha_{j,k} - \alpha_{j,h} - \alpha_{g,k} + |\alpha_{g,h}|||] \end{aligned} \tag{3.12}$$

Ovviamente per ricavare il segno è importante che la somma degli angoli di cui sono noti i segni non sia pari a zero.

A partire dalle equazioni (3.9), (3.11) e (3.12) è possibile ricostruire la matrice caratterizzante il chip, indipendentemente dalle efficienze. Per fare ciò è necessario tuttavia tener conto di alcune proprietà della matrice.

Due matrici M_a e M_b sono dette equivalenti se esistono due matrici diagonali unitarie D_1^U e D_2^U tali che $M_a = D_1^U M_b D_2^U$. Nel nostro caso le matrici diagonali rappresentano i fattori di fase, non noti, relativi alle porte di ingresso e di uscita dell'interferometro. In altre parole essi sono una realizzazione particolare delle fasi fluttuanti riportate in Figura (3.1)

Poiché le misure che ci interessano non sono sensibili a questi fattori di fase, è possibile sfruttare le matrice diagonali per ottenere una matrice equivalente a quella che caratterizza il chip la quale abbia però la proprietà di essere *real borderd*, ovvero gli elementi della prima riga e della prima colonna sono reali.

La seconda proprietà di cui abbiamo bisogno è che il segno dell'angolo $\alpha_{2,2}$ sia fissato, ad esempio $Im(M_2, 2) \geq 0 \rightarrow \alpha_{2,2} \geq 0$. Quale che esso sia non ha importanza dal momento che la statistica dei bosoni risulta simmetrica per il passaggio dalla matrice M alla sua complessa coniugata M^*. Tuttavia risulterà importante poter contare sul fatto che il segno di uno degli angoli sia noto *a priori*.

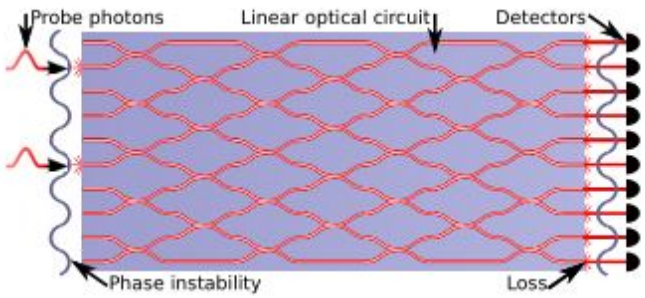

Figura 3.1: La tomografia del chip non dipende dalle fluttuazioni di fase

Tenendo conto di queste osservazioni scriviamo la matrice.

$$M = \begin{pmatrix} \tau_{1,1} & \tau_{1,2} & \dots & \tau_{1,m} \\ \tau_{2,1} & \tau_{2,2}e^{i\alpha_{2,2}} & \dots & \tau_{2,m}e^{i\alpha_{2,m}} \\ \vdots & \vdots & \ddots & \vdots \\ \tau_{m,1} & \tau_{m,2}e^{i\alpha_{m,2}} & \dots & \tau_{m,m}e^{i\alpha_{m,m}} \end{pmatrix} \tag{3.13}$$

Per trovare le ampiezze nella equazione (3.9) fissiamo $\tau_{j,k} = \tau_{1,1}$ e facciamo variare gli altri indici di conseguenza. L'ampiezza da determinare è $\tau_{g,h}$, la quale risulterà essere pertanto una funzione degli elementi della prima riga e dalla prima colonna.

Per trovare i valori in modulo degli angoli sfruttiamo il fatto che la matrice è real-bordered. Difatti, adottando la stessa notazione per cui $(j,k) = (1,1)$, l'argomento del coseno nella 3.11 è pari a $\alpha_{g,h}$.

Non rimane altro che da determinare i segni degli angoli. Ricordando che l'angolo $\alpha_{2,2}$ è definito positivo, i segni dei restanti angoli, ad esempio quelli della seconda colonna, possono essere trovati fissando $\alpha_{j,k} = \alpha_{2,1}$ e $h = j = 2$, e facendo variare l'indice g da 3 a m. Similmente per gli angoli delle righe.

3.2 Implementazione dell'algoritmo

In questa sezione illustriamo il programma che implementa l'algoritmo per la tomografia del chip. Prima di tutto è opportuno fare due precisazioni. La prima di queste consiste nel constatare che noi avevamo a disposizione tutti i coupling, e non solo quelli relativi alla prima riga e alla prima colonna della matrice che dobbiamo ricostruire. Inoltre, e questa è la seconda precisazione, abbiamo utilizzato i coupling normalizzati. Sarebbe a dire per ogni riga della matrice abbiamo normalizzati i quattro coupling corrispondenti a ciascuna colonna in modo tale che la loro somma fosse pari a uno.

Anzitutto riportiamo di seguito le misure che abbiamo utilizzato. Per cominciare abbiamo i coupling:

$$R = \begin{pmatrix} 0.232752 & 0.192008 & 0.335003 & 0.240237 \\ 0.276189 & 0.307241 & 0.213199 & 0.203371 \\ 0.193364 & 0.247673 & 0.260293 & 0.2984 \\ 0.258443 & 0.248083 & 0.268467 & 0.225007 \end{pmatrix} \tag{3.14}$$

Per quanto riguarda le misure di visibilità la proprietà della matrice di essere real bordered fa sì che, ai fini del calcolo degli angoli α in modulo, i valori realmente necessari siano soltanto $V_{g,h,11}$, con $g, h = [2, 3, 4]$. A scanso di equivoci ricordiamo che con j, g indichiamo gli ingressi del chip e con k, h le uscite.

$$V = \begin{pmatrix} -0.642 & 0.890 & 0.682 \\ 0.871 & -0.958 & 0.871 \\ 0.581 & 0.856 & -0.686 \end{pmatrix} \tag{3.15}$$

La matrice V che abbiamo riportato necessita di una piccola spiegazione. Poiché, come abbiamo detto, j e k sono fissati, e in particolare $j = k = 1$, per calcolare il modulo degli angoli è sufficiente considerare un solo elemento della matrice delle visibilità, che chiameremo *vis*. *vis* ha quattro indici, due per gli input e due per gli output; ciò vuol dire che ogni elemento di *vis* è esso stesso una matrice. Gli indici j, k selezionano dunque una particolare matrice, mentre gli indici g, h gli elementi della matrice selezionata. Di conseguenza V è la matrice selezionata da $j = k = 1$ di indici g, h necessariamente diversi da $j = k$ in quanto i fotoni vengono iniettati in porte diverse.

Passiamo ora ad esaminare il listato vero e proprio del programma.

Dopo aver inizializzato le matrici R e V la prima cosa che facciamo è costruire la matrice delle ampiezze.

$$\tau = \sqrt{R} \tag{3.16}$$

Ancora una volta sottolineiamo che avendo a disposizione tutti i coupling non è stato necessario servirsi della (3.9). Fatto questo calcoliamo $|\alpha|$ a partire dalla equazione (3.11).

```
x = Table[Sqrt[R[[1, 1]] * R[[g, h]] / (R[[1, h]] * R[[g, 1]])], {g, 2, 4}, {h, 2, 4}]
y = Table[x[[g, h]] + 1 / x[[g, h]], {g, 1, 3}, {h, 1, 3}]
a = Table[ArcCos[-V[[g, h]] * y[[g, h]] / 2], {g, 1, 3}, {h, 1, 3}]
```

Figura 3.2: In questo passaggio costruiamo la matrice a dei moduli degli angoli α, seguendo la (3.11)

La matrice a in realtà non include gli elementi della prima riga e dalla prima colonna (infatti è una matrice 3×3). Non è stato necessario calcolarli in quanto sappiamo già che, essendo M real bordered, essi sono nulli. Tuttavia per convenienza li aggiungiamo "manualmente" (vd. Figura (3.3)).

```
α = Table[0, {i, 4}, {j, 4}]
α[[2 ;; 4, 2 ;; 4]] = a
```

Figura 3.3: La matrice a è completata aggiungendo le fasi nulle della prima riga e della prima colonna. Questo stratagemma di aggiungere manualmente elementi banali di una matrice verrà adottato anche nel resto del programma, pertanto di volta in volta ci limiteremo a fare riferimento a questa Figura come esempio principale.

Ora dobbiamo calcolare i segni delle fasi. Cominciamo dalla seconda riga; sappiamo già $\alpha_{2,2} > 0$, quindi dobbiamo trovare i segni di $\alpha_{2,3}$ e $\alpha_{2,4}$. L'equazione che dobbiamo usare questa volta è (3.12), pertanto dobbiamo anzitutto costruire la matrice β il cui generico elemento è dato da $\beta_{g,h,j,k} = |\alpha_{j,k} - \alpha_{j,h} - \alpha_{g,k} + \alpha_{g,h}|$, che può essere ottenuto invertendo la relazione (3.11). Ovviamente non tutti gli indici sono liberi. Anzitutto dovremo sfruttare il fatto che la matrice M è real bordered, pertanto uno degli indici, **j** in questo caso, sarà pari a uno. Poiché siamo interessati ai segni degli angoli della **seconda riga**, la **colonna k** sarà pari al numero di riga, ovvero 2. Inoltre **k=g**.

La ragione per cui abbiamo introdotto queste condizioni sugli indici degli angoli è ovvia: l'argomento del coseno è costituito dalla somma di quattro angoli. Vogliamo che il segno di uno solo di questi sia indeterminato. La soluzione consiste dunque nello scegliere gli indici in modo tale che due di questi angoli appartengano alla prima riga, e siano dunque nulli, e il terzo sia $\alpha_{2,2}$, del quale conosciamo già il segno. Il quarto angolo è quello di cui vogliamo calcolare il segno.

```
b = Table[Abs[ArcCos[(-vis[[1, 2]][[2, h]] * U[[2, h]])/2]], {h, 3, 4}]
β = Table[0, {i, 1, 4}]
β[[3 ;; 4]] = b
```

Figura 3.4: Calcoliamo gli elementi della matrice β necessari alla determinazione del segno degli angoli $\alpha_{2,3}$ e $\alpha_{2,4}$. Come in Figura (3.3), per convenienza prima calcoliamo i suddetti elementi (sottomatrice b) e dopo li inseriamo nella matrice β. In questo caso tuttavia è più esatto parlare di vettore, piuttosto che di matrice: poiché l'algoritmo riportato in figura è valido soltanto per gli angoli della seconda riga, gli elementi di β che sono stati calcolati sono solo quelli necessari alla determinazione dei segni degli angoli di questa riga.

La matrice U in Figura (3.4) non è che una riscrittura di y, in analogia a quanto abbiamo fatto in (3.3).

```
X = Table[Sqrt[R[[1, g]] * R[[g, h]] / R[[1, h]] * R[[g, g]]], {g, 2, 4}, {h, 2, 4}]
y = Table[X[[g, h]] + 1 / X[[g, h]], {g, 1, 3}, {h, 1, 3}]
U = Table[0, {i, 4}, {j, 4}]
U[[2 ;; 4, 2 ;; 4]] = y
```

Figura 3.5: La matrice U non è altro che una forma conveniente di y, definita in (3.10)

L'ultimo passaggio per calcolare i segni degli angoli $\alpha_{2,3}$ e $\alpha_{2,4}$ consiste nell'implementare (3.12). A tal proposito si veda la Figura (3.6)

```
q = Table[0, {i, 4}]
w = Table[0, {i, 4}]
For[h = 3, h ≤ 4, h++,
 q[[h]] = Abs[-α[[2, 2]] - α[[2, h]] ];
 w[[h]] = Abs[-α[[2, 2]] + α[[2, h]] ];
 If[q[[h]] > Pi, q[[h]] = Abs[2 * Pi - q[[h]]],];
 If[w[[h]] > Pi, w[[h]] = Abs[2 * Pi - w[[h]]],];
 α[[2, h]] = α[[2, h]] * Sign[Abs[ β[[h]] - q[[h]]] - Abs[ β[[h]] - w[[h]]]]
]
```

Figura 3.6: In questo passaggio ci limitiamo a scrivere l'equazione (3.12), facendo attenzione al fatto che gli angoli devono essere compresi tra 0 e π

L'unico aspetto che vale la pena sottolineare nel calcolo in Figura (3.6) è l'utilizzo di if al fine di limitare il range degli angoli a $0 \leq \alpha \leq \pi$.

Nel calcolo dei segni degli angoli abbiamo utilizzato dei valori di visibilità non compresi tra quelli contenuti in V, pertanto, prima ancora di eseguire i passaggi che abbiamo descritto, è importante importarli in Mathematica (Figura (3.7)).

In questo modo abbiamo calcolato i segni degli angoli $\alpha_{2,3}$ e $\alpha_{2,4}$. Ora passiamo a calcolare quelli di $\alpha_{3,2}$ e $\alpha_{4,2}$. In realtà non dobbiamo aggiungere nulla a quanto detto in quanto ancora una volta il procedimento si fonda sulla conoscenza *a priori* del segno di $\alpha_{2,2}$, che è maggiore di zero. Quindi si tratta solo, nei passaggi che abbiamo descritto, di "scambiare" le righe con le colonne.

Gli ultimi angoli di cui ancora non si conoscono i segni sono: $\alpha_{3,3}, \alpha_{3,4}, \alpha_{4,3}, \alpha_{4,4}$. Anzitutto osserviamo che dobbiamo generalizzare la procedura che abbiamo fin qui illustrato. Infatti fino ad ora ci siamo occupati di coppie di angoli appartenenti alla stessa riga, o alla stessa colonna. Ora dobbiamo invece calcolare i segni degli angoli appartenenti alla sotto matrice 2×2 di indici g, h che possono assumere i valori $[3, 4]$. Questa necessità si riflette evidentemente nel calcolo

```
vis = Table[0, {j, 4}, {k, 4}, {g, 4}, {h, 4}]
vis[[1, 2]][[2, 3]] = 0.525
vis[[1, 2]][[2, 4]] = 0.882
vis[[2, 1]][[3, 2]] = 0.622
vis[[2, 1]][[4, 2]] = 0.916
vis[[2, 2]][[3, 3]] = -0.547
vis[[2, 2]][[3, 4]] =  0.898
vis[[2, 2]][[4, 3]] = 0.911
vis[[2, 2]][[4, 4]] = -0.931
```

Figura 3.7: Non è necessario importare le visibilità per ogni possibile combinazione di input e output ma solo quelle in Figura. La notazione utilizzata è $vis[[j,k][g,h]]$.

mostrato in Figura (3.8): a differenza di prima abbiamo due, e non uno, indici liberi, g e h.

```
b = Table[Abs[ArcCos[-vis[[2, 2]][[g, h]] * U[[g, h]] / 2]], {g, 3, 4}, {h, 3, 4}]
```

Figura 3.8: Ora b, e di conseguenza β, non sarà più un vettore ma una matrice 2×2.

Per calcolare i segni degli angoli in questione sfruttiamo il fatto che quelli degli angoli appartenenti alla seconda colonna e alla seconda riga sono stati già trovati. Il resto non è, come si diceva prima, che una generalizzazione al caso bidimensionale dei conti precedenti.

Avendo finalmente trovato tutti gli angoli non rimane che da costruire la matrice M del chip e verificare che sia unitaria.

```
Abs[M] // MatrixForm
```

$$\begin{pmatrix} 0.482444 & 0.438187 & 0.578795 & 0.490139 \\ 0.525537 & 0.554293 & 0.461734 & 0.450967 \\ 0.440038 & 0.497668 & 0.510189 & 0.54626 \\ 0.508373 & 0.498079 & 0.518138 & 0.474349 \end{pmatrix}$$

```
Arg[M] // MatrixForm
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.864281 & 2.77494 & -2.3347 \\ 0 & 2.6726 & -0.288896 & 2.66497 \\ 0 & -2.1933 & 2.62178 & 0.810706 \end{pmatrix}$$

Figura 3.9: Ampiezze e angoli delle fasi della matrice M caratterizzante il chip.

```
Re[M] // MatrixForm
Im[M] // MatrixForm
```

$$\begin{pmatrix} 0.482444 & 0.438187 & 0.578795 & 0.490139 \\ 0.525537 & 0.35984 & -0.431045 & -0.311954 \\ 0.440038 & -0.443933 & 0.489047 & -0.485379 \\ 0.508373 & -0.290415 & -0.449698 & 0.32682 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.421611 & 0.165527 & -0.325662 \\ 0 & 0.224938 & -0.14535 & 0.250613 \\ 0 & -0.404651 & 0.257369 & 0.343795 \end{pmatrix}$$

Figura 3.10: Parte reale e immaginaria della matrice M caratterizzante il chip.

$$\begin{pmatrix} 1. + 0.\,i & 0.00883176 - 0.120931\,i & 0.0629223 - 0.137273\,i & 0.0179099 - 0.140158\,i \\ 0.00883176 + 0.120931\,i & 1. + 0.\,i & 0.00128853 - 0.0135612\,i & 0.014588 + 0.0604832\,i \\ 0.0629223 + 0.137273\,i & 0.00128853 + 0.0135612\,i & 1. + 0.\,i & -0.0681966 - 0.0566887\,i \\ 0.0179099 + 0.140158\,i & 0.014588 - 0.0604832\,i & -0.0681966 + 0.0566887\,i & 1. + 0.\,i \end{pmatrix}$$

Figura 3.11: Come prima verifica della bontà del nostro programma possiamo moltiplicare M per la sua complessa coniugata e constatare che effettivamente il risultato è una matrice unitaria.

3.3 Verifica della correttezza della implementazione

Nella sezione precedente abbiamo mostrato che la matrice fornita dal nostro programma ha la proprietà di essere una matrice unitaria. Tuttavia ciò non è sufficiente a dimostrare che il programma funzioni correttamente, e dunque che la matrice descriva correttamente il chip quantistico. Per fare ciò abbiamo sviluppato un secondo programma, il quale genera casualmente una matrice unitaria, fornendo anche le visibilità e i coupling necessari alla sua ricostruzione. Questi dati vengono passati al primo programma, che a partire da essi ricostruisce la matrice: se essa coincide con quella originale allora bisogna supporre che il programma sia ben scritto.

Nel seguito discuteremo il listato del programma che genera la matrice random e daremo un esempio dell'intero procedimento di verifica.

Per prima cosa generiamo una matrice random con elementi complessi.

```
M = Table[RandomComplex[{-I * Pi, 1 + I * Pi}], {i, 4}, {j, 4}]
```

Figura 3.12: Generazione di una matrice random i cui elementi sono numeri complessi.

Fatto ciò dobbiamo rendere la matrice unitaria. Per farlo partiamo dalla **decomposizione QR**. Data una matrice M, attraverso questa decomposizione

è possibile ottenere due matrici Q e R tali che $M = QR$. Q ha la proprietà interessante di essere ortogonale: $QQ^T = I$. Per prima cosa vogliamo dunque ottenere la matrice Q associata alla matrice random che abbiamo generato.

```
A = QRDecomposition[M][[1]]
```

Figura 3.13: A partire dalla matrice random M costruiamo la matrice Q che ha la proprietà di essere ortogonale. Poiché $QRDecomposition$ restituisce due risultati, la matrice Q e la matrice R, noi dobbiamo selezionare solo il primo di questi, pertanto scriviamo $QRDecomposition[M][[1]]$.

La nostra matrice deve avere la proprietà molto importante di essere real bordered. A tal fine moltiplichiamo la matrice per un fattore di fase comune, ininfluente. L'operazione deve essere ripetuta due volte, per semplificare le fasi e della prima riga e della prima colonna.

```
For[i = 1, i ≤ 4, i++,
 A[[All, i]] = A[[All, i]] * e^(-i*Arg[A][[1,i]]);
 A[[i, All]] = A[[i, All]] * e^(-i*Arg[A][[i,1]])
]
```

Figura 3.14: Moltiplichiamo la matrice per un fattore di fase al fine di renderla real bordered.

Ora non ci rimane che ricavare i coupling e le visibilità di questa matrice. Per quanto riguarda i primi sarà sufficiente elevare al quadrato le ampiezze, mentre le seconde saranno ottenute invertendo l'equazione (3.11)

```
R = Table[τ[[i, j]]^2, {i, 4}, {j, 4}]
```

Figura 3.15: Generazione dei coupling della matrice random.

```
vis = Table[-2 * Cos[α[[j, k]] - α[[j, h]] - α[[g, k]] + α[[g, h]]] *
    τ[[j, k]] * τ[[j, h]] * τ[[g, k]] * τ[[g, h]]
  ------------------------------------------------, {j, 4}, {k, 4},
  τ[[j, k]]^2 * τ[[g, h]]^2 + τ[[g, k]]^2 * τ[[j, h]]^2
 {g, 4}, {h, 4}]
```

Figura 3.16: Generazione delle visibilità della matrice random invertendo l'equazione (3.11).

Ora non rimane altro che passare queste misure al programma che ricostruisce la matrice e verificare che l'output coincida effettivamente con la matrice random da noi costruita. Facciamo un esempio numerico.

Con il secondo programma generiamo la matrice random A che ha la proprietà di essere unitaria e real bordered. Facciamo notare che, se l'elemento $(2,2)$ di tale matrice ha fase negativa, essa deve essere scartata in quanto il programma che ricostruisce la tomografia è stato scritto assumendo che valga $\alpha_{2,2} > 0$.

```
MatrixForm[A]
A.Conjugate[Transpose[A]] // MatrixForm
```

```
data/matrice.dat
```

$$\begin{pmatrix} 0.45722 + 0.\,i & 0.0661735 - 1.38778\times10^{-17}\,i & 0.605152 + 6.245\times10^{-17}\,i & 0.648354 - 1.38778\times10^{-17}\,i \\ 0.183168 + 2.5804\times10^{-17}\,i & 0.320295 + 0.615982\,i & -0.58039 - 0.00633911\,i & 0.379856 - 0.0569528\,i \\ 0.339407 + 0.\,i & 0.412357 + 0.405387\,i & 0.35379 - 0.184473\,i & -0.611652 + 0.130806\,i \\ 0.801375 + 0.\,i & -0.28561 - 0.312487\,i & -0.362448 + 0.079579\,i & -0.197683 - 0.0423827\,i \end{pmatrix}$$

$$\begin{pmatrix} 1. + 0.\,i & 8.32667\times10^{-17} - 2.77556\times10^{-17}\,i & 5.55112\times10^{-17} - 2.77556\times10^{-17}\,i & -1.66533\times10^{-16} - 1.80411\times10^{-16}\,i \\ 8.32667\times10^{-17} + 2.77556\times10^{-17}\,i & 1. + 0.\,i & 1.66533\times10^{-16} + 6.245\times10^{-17}\,i & -1.38778\times10^{-16} - 6.93889\times10^{-18}\,i \\ 5.55112\times10^{-17} + 2.77556\times10^{-17}\,i & 1.66533\times10^{-16} - 6.245\times10^{-17}\,i & 1. + 0.\,i & 0. - 6.93889\times10^{-18}\,i \\ -1.66533\times10^{-16} + 1.80411\times10^{-16}\,i & -1.38778\times10^{-16} + 6.93889\times10^{-18}\,i & 0. + 6.93889\times10^{-18}\,i & 1. + 0.\,i \end{pmatrix}$$

Figura 3.17: Matrice random A, unitaria e real bordered, e il prodotto di A per la sua complessa coniugata.

Ampiezze e angoli delle fasi di A sono riportate in Figura (3.18)

```
MatrixForm[τ]
MatrixForm[α]
```

$$\begin{pmatrix} 0.45722 & 0.0661735 & 0.605152 & 0.648354 \\ 0.183168 & 0.694278 & 0.580425 & 0.384101 \\ 0.339407 & 0.578254 & 0.398995 & 0.625482 \\ 0.801375 & 0.423345 & 0.371081 & 0.202176 \end{pmatrix}$$

$$\begin{pmatrix} 0. & -2.09718\times10^{-16} & 1.03197\times10^{-16} & -2.14046\times10^{-17} \\ 1.40876\times10^{-16} & 1.0913 & -3.13067 & -0.148824 \\ 0. & 0.776875 & -0.480637 & 2.93091 \\ 0. & -2.31129 & 2.92546 & -2.93039 \end{pmatrix}$$

Figura 3.18: Ampiezze e angoli delle fasi della matrice random A.

Il programma che ricostruisce A, usato per la tomografia del chip, fornisce i valori in Figura (3.19).

Poiché i valori coincidono possiamo concludere che il programma ricostruisce correttamente la tomografia del chip.

```
Abs[M] // MatrixForm
```

$$\begin{pmatrix} 0.45722 & 0.0661735 & 0.605152 & 0.648354 \\ 0.183168 & 0.694278 & 0.580425 & 0.384101 \\ 0.339407 & 0.578254 & 0.398995 & 0.625482 \\ 0.801375 & 0.423345 & 0.371081 & 0.202176 \end{pmatrix}$$

```
Arg[M] // MatrixForm
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1.0913 & -3.13067 & -0.148824 \\ 0 & 0.776875 & -0.480637 & 2.93091 \\ 0 & -2.31129 & 2.92546 & -2.93039 \end{pmatrix}$$

Figura 3.19: Ampiezze e angoli delle fasi della matrice random ricostruita dal programma che esegue la tomografia del chip

Conclusioni

In questo lavoro abbiamo discusso l'attività di laboratorio svolta presso il gruppo di ricerca di Informazione e Computazione Quantistica. Lo scopo dell'esperienza è stata la certificazione di un circuito integrato quantistico implementante un particolare modello computazionale basato sull'interferenza bosonica, noto come boson-sampling. Il protocollo di certificazione scelto consiste nella verifica della legge di soppressione di alcune configurazioni di output. Tale legge è una generalizzazione dell'effettoHOM.

Abbiamo inoltre illustrato il programma da noi scritto che esegue la tomografia del chip: a partire dai rate e dalle misure di visibilità, ricavate dall'andamento del numero di coincidenze tra i fotoni in uscita del chip in funzione del ritardo all'ingresso, abbiamo ricostruito la matrice unitaria caratterizzante il chip.

Abbiamo verificato che la matrice fosse effettivamente unitaria. Inoltre abbiamo spiegato il funzionamento di un secondo programma, che produce una matrice unitaria random, le visibilità e i rate associati. Passando i dati della matrice random al primo programma abbiamo confrontato l'output con la matrice originale. In questo modo abbiamo potuto concludere che il programma funziona correttamente e dunque che la matrice fornita è effettivamente quella caratterizzante il chip.

Bibliografia

[1] A. Laing and J. L. O'Brien, "Super-stable tomography of any linear optical device," (2012) , arXiv:1208.2868.

[2] S. Mertens, "Computational Complexity for Physicists," (2000) , arXiv:cond-mat/0012185.

[3] R. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics* (1982) .

[4] P. W. Shor, "Algorithms for quantum computation: discrete log and factoring," *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (1994) .

[5] N. S. Dattani and N. S. Dattani, "Quantum factorization of 56153 with only 4 qubits," (2014) , arXiv:1411.6758.

[6] S. Aaronson and A. Arkhipov, "The Computational Complexity of Linear Optics," (2010) , arXiv:1011.3245.

[7] M. C. Tichy, K. Mayer, A. Buchleitner, and K. Mølmer, "Stringet and Efficient Assessment of Boson-Sampling Devices," *Physical Review Letters* (2014) .

[8] S. Aaronson, *Quantum Computing Since Democritus*. Cambridge University Press, 2013.

[9] N. Spagnolo *et al.*, "Efficient experimental validation of photonic boson sampling against the uniform distribution," (2013) , arXiv:1311.2622v2.

[10] C. K. Hong, Z. Y. Ou, and L. Mandel, "Measurement of Subpicosecond Time Intervals between Two Photons by Interference," *Physical Review Letters* (1987) .

[11] L. Sansoni, F. Sciarrino, G. Vallone, P. Mataloni, A. Crespi, R. Ramponi, and R. Osellame, "Polarization Entangled State Measurement on a Chip," *Physical Review Letters* (2010) .